UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

# Robot didáctico de bajo coste


Autor/es

Miguel Ángel Ezquerro Ezquerro

Director/es

CARLOS ELVIRA IZURRATEGUI y SILVANO NÁJERA CANAL

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA



Curso académico

2021-22

publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

UNIVERSIDAD DE LA RIOJA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

# TRABAJO DE FIN DE GRADO

**TITULACIÓN: Grado en Ingeniería Electrónica Industrial y Automática**

**CURSO: 2019/2020 CONVOCATORIA: JULIO**

**TÍTULO:**
**Low-cost didactic robot (Robot didáctico de bajo coste)**

**ESTUDIANTE:** Miguel Ángel Ezquerro Ezquerro

**TUTORES/AS:** Carlos Elvira Izurrategui
Silvano Nájera Canal

**DEPARTAMENTO: Ingeniería Eléctrica**

# LOW-COST DIDACTIC ROBOT

MIGUEL ÁNGEL EZQUERRO EZQUERRO

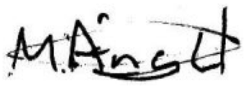
Undergraduate Final Year Project Report submitted in fulfilment for the requirements of the degree of Industrial and Automatic Electronic Engineering

**Higher Technical School of Industrial Engineering**
**University of La Rioja**

July 2022

# DECLARATION

I Miguel Angel Ezquerro Ezquerro declare and verify that this entitled “low-cost didactic robot” is the result of my own research except as cited in the references. This report has not been accepted for any degree and is not concurrently submitted in candidature for any other degree.

Signature: 

Name: Miguel Ángel Ezquerro Ezquerro

Date: 20 – July – 2022

# List of Documents

1 Memory
2 Annexes
3 Blueprints
4 Specifications
5 Measuring
6 Budget

UNIVERSIDAD DE LA RIOJA

*Escuela Técnica Superior de Ingeniería Industrial*

*Grado en Ingeniería Electrónica Industrial y Automática*

# LOW-COST DIDACTIC ROBOT

## **MEMORY**

Autor: Miguel Ángel Ezquerro Ezquerro

Tutor: Carlos Elvira Izurrategui

Date: 18 July 2022

# ABSTRACT

The present paper aims to expose the process of redesign, manufacturing, assembly and programming of a robotic arm with six degrees of freedom. The project is developed under an academic approach, presenting a real and cheap application case that facilitates the introduction of anyone in the fascinating world of robotics.

It will begin by treating the capabilities, limitations, improvements and working modes of the robot and its controllers, to later dig into each of the elements that compose it along with the various aspects of its development:

- Design and manufacture of the robot with its actuators and sensors.
- Configuration of the communication system with the robot and between its different actuators.
- Monitoring of the different variables of the robot.
- Finally, different didactic examples will be presented.

# RESUMEN

El presente trabajo tiene por objetivo mostrar el proceso de rediseño, manufactura, ensamblaje y programación de un brazo robótico de seis grados de libertad. El proyecto se desarrolla a su vez bajo un enfoque académico, presentando un caso de aplicación real y barato que facilita la introducción de cualquiera en el fascinante mundo de la robótica

Se comenzará tratando las capacidades, limitaciones, mejoras y modos de trabajo del robot y sus controladores, para posteriormente profundizar en cada uno de los elementos que lo componen junto con los diversos aspectos de su desarrollo:

- Diseño y fabricación del robot con sus actuadores y sensores.
- Configuración del sistema de comunicación con el robot y entre los diferentes actuadores del mismo.
- Monitorización de las diferentes variables del robot.
- Finalmente, se presentarán diferentes ejemplos didácticos.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

*All illustrations have been created exclusively for this paper.*

# List of Abbreviations

| | |
|---|---|
| 6-DOF | Six-degree-of-freedom |
| EMC | Electromagnetic Compatibility |
| FDM | Fused Deposition Material |
| HAT | Hardware Attached on Top |
| HTM | Homogeneous Transformation Matrices |
| IS | International Standard |
| MCU | Micro Controller Unit |
| PWM | Pulse Width Modulation |
| ROS | Robotics Operating System |
| RS | Reference System |
| RTOS | Real Time Operating System |
| spp. | Steps per second |
| TTL | Transistor-Transistor Logic |

# 1 INTRODUCTION

In defining the scope of our subject, we have to establish the genealogy of robotic mechanical systems. These are, obviously, a subclass of the much broader class of mechanical systems. Obviously, a mechanical system is a system composed of mechanical elements. If this system complies with the definition of dynamic system, then we end up with a dynamic mechanical system. Furthermore, a mechanical system can be either natural or engineered, we will find both of them in this paper. Engineered mechanical systems may be controlled or uncontrolled. Most engineering systems are controlled mechanical systems. In addition, a controlled mechanical system can be either robotic or nonrobotic. The second ones are systems supplied with primitive controllers, such as analog thermostats, servovalves, etc. Robotic mechanical systems, in turn, can be programmable, or intelligent. Programmable mechanical systems obey motion commands which are stored in a memory device or generated on-line. In either case, they need sensors, such as joint encoders, accelerometers, and dynamometers.

With all the information exposed we can conclude that a robotic mechanical system is composed of a few subsystems, namely, a mechanical subsystem composed in turn of both rigid and articulated bodies, although the system we will study here is composed only of the former; a sensing subsystem; an actuation subsystem; a controller; and an information-processing subsystem. Additionally, these subsystems communicate among themselves via interfaces, whose function consists basically of decoding the transmitted information from one medium to another.

Among all robotic mechanical systems, ***manipulators*** deserve special study for several reasons. One is that, even in their possible simplest forms, as robotic arms, they occur most frequently in industry, medicine, arts and education. Another is that the architecture of robotic arms constitutes the most straightforward of all robotic architectures, and consequently, appear as fraction of other, more complex robotic mechanical systems.

A manipulator, in general, is a mechanical system intended to object manipulation. Manipulating, in turn, means to move something with one hand, as the word comes from the Latin "*manus*", meaning hand. The essential idea behind the previous concept is that hands are among the organs that the human brain can control mechanically with the highest accuracy and versatility, as the work of an artist like Salvador Dalí, of an accomplished piano player, or of a surgeon can attest. Manipulators are the human creation that labour following the unachievable human versatility.

A manipulator is thus any device that helps a human being perform a manipulating task. Manipulators have existed ever since humanity created the first tool, but manipulators have been developed to the extent during the last decades and are now capable of actually mimicking motions of the human arm, and of the human hand, for that matter. In fact, during the Second World War, the need arose for manipulating probe tubes containing radioactive substances. This set up the first six-degree-of-freedom (6-DOF) manipulator.

# 2 MOTIVATION

Today the most useful and efficient robotic systems are the industrial robot manipulators which reduce near to zero the human presence in difficult or monotonous jobs, or where a human would otherwise be faced with hazardous conditions. This is the strongest incentive to science investigation; facilitate human life. New manipulators like FANUC or KUKA robots can do marvellous movements and follow complex trajectories with amazing dynamic control and collaborative competencies.

Every single engineer will come across a robotic manipulator, and it is essential to not only foresee these devices applications but understand its basements. In addition, the number of robot-human interactions will increase. When I went to the Advanced Factories Expo & Congress in Barcelona past march, It was repeatedly said that the key challenges of the present-day robotics are human-robot interaction and human-robot collaboration. To ensure this collaboration, Robots must be improved and the knowledge about them enriched and published.

# 3 LITERATURE SURVEY

In this paragraph the previous knowledge and precedents of Robotics uses and applications are described. The focus will be placed on industrial robotics applications, manipulators, proficiency robotics documentation and commercial individual user robots.

## 3.1 COMMON ROBOTICS APPLICATIONS

The robot plays the role of a slave in the nowadays industrial applications; some of them are workpiece and material handling, flexible fixturing, palletizing and parts feeding, and die casting (sometimes even under water). In this situation the role of industrial engineers is essential to ensure those machines safety and reliability. Pick and place robots particularly represent the most common application of robots in material handling, where tasks are often obnoxious or repetitive and potentially hazardous. Often the industrial robots are used in the labours when they execute point to point movements.

Robots are essential in tasks that human beings cannot develop not just because the corresponding risk or damage but the intricacy of the assignment. Special applications of the industrial robots are the following: maintenance and repair, quality assurance, inspection, and testing, robots in food, textile and clothing industry, and in construction. Quality assurance, inspection, and testing are commonly applied in the electronic industry, where electric parameters need to be tested during assembly of electronic circuits. In this particular situation the robot performs the necessary measurements on the object (dimensional, electric), while takes place the movements of grasping and placing the object into a new position. Another high accuracy tasks is the machines maintenance and repair using teleoperated and autonomous robots for instance in nuclear industry, highways, railways, power lines maintenance, and aircraft servicing. Controlled robots are also taking part in the food industry, where in addition to handling and wrapping up applications in food processing, they are used for the chores such as food preparation or even decorating dessert. The textile and clothing industry presents unusual problems due to of the limp nature of the workpieces, making handling of textiles or similar extremely complicate managing materials.

## 3.2 MAKING INFORMATION MORE ACCESIBLE

There are several Robotics Dissertation but most of them are from postgraduate studies, hard to understand in the lowest academic levels. Some of the most representative are listed in the bibliography, and this paper is supported by its disclosures. This dissertation has a didactic scope and is intended to facilitate and make complex robotics concepts accessible in its basic postulations and in its real-life implementation.

## 3.3 NIRYO ROBOT

The Niryo robot is a multi-platform open-source robot. With its three Steppers and its Dynamixel XL servomotors, it facilitates the understanding of robotics in a realistic environment like the industrial reality: its 6 axis allow it to develop all the movements required for specific tasks. Niryo One's philosophy is precisely reproduce complex handcrafted situations but still accessible. Indeed, the Niryo One manipulator has been designed to work in an autonomous way and under control of the freeware Niryo One Studio, which let you program the robot with different approaches, from the most intuitive to the most advanced.

The Niryo One robot is no longer available. In the official website you can only buy the new design, Niryo Ned 2. This model incorporates artificial vision and collaborative features.

In time, Niryo One robot cost to buy the amount of one thousand six hundred euros.

# 4 PROBLEM STATEMENT. OBJECTIVES AND ASSUMPTIONS

The aim of this project is to make several improvements in a commercial robot, an expensive 6-DOF manipulator. Furthermore, the prime target of this paper is to provide the reader with the tool to better understand fundamental concepts behind the analysis, control, and programming of robotic mechanical systems at large. Special attention has been given to the manufacturing cost reduction to streamline the acquisition of the presented device by educational institutions or individuals.

We start from the designed open licensed Niryo One Robot. The original design of the robot will be put under scrutiny, detecting grave design failures, calculating all the kinematics, thermodynamic analysis, and test functionality. Several design und functionality modifications will be implemented.

# 5 GEOMETRIC DESCRIPTION OF THE ROBOT

In this chapter we proceed to parametrize the whole robot geometry. To do this and to make the future analysis of the kinematic problem it is necessary to create a simplified model of the robot. The robot is a 6-axis robotic arm. That means 6 degrees of freedom for the robot.

The robot can be broken down into 7 parts, like a rope with nodes we can divide the robot between the 6 joints indicated in each change of colour:

*Base – **J1** – Shoulder – **J2** – Arm – **J3** – Elbow – **J4** – Forearm – **J5** – Wrist – **J6** – Hand*.

| | |
|---|---|
| 1- Base | **Orange** |
| 2- Shoulder | **Yellow** |
| 3- Arm | **Green** |
| 4- Elbow | **Cyan** |
| 5- Forearm | **Blue** |
| 6- Wrist | **Pink** |
| 7- Hand | **Red** |

*Illustration 1: Robot Parts*

## 5.1 HOMOGENEOUS TRANSFORMATION MATRICES

Homogeneous transformation matrices (*HTM*) enable us to combine rotation matrices (which have 3 rows and 3 columns) and displacement vectors[1] (which have 3 rows and 1 column) into a single matrix. They are an important concept of forward kinematics and enable us to study the rotation, translation, perspective orientation and scale of one ***RS*** viewed form another ***RS***.

*Notation:*

$$H_{01} = {}^{0}H_{1} \rightarrow \text{Reference system 1 seen from reference system 0}$$

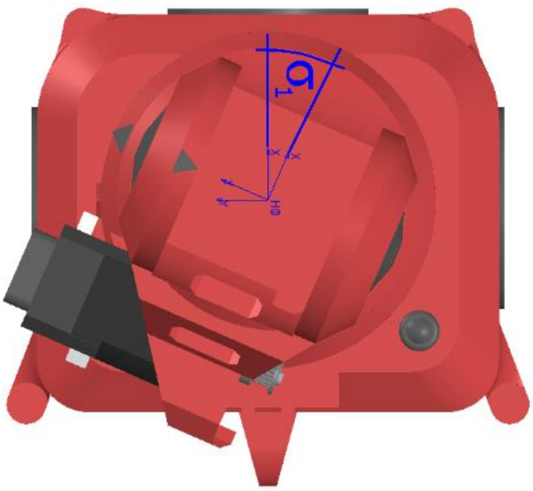
The different ***RS*** matrix must be ubicated in the strategic point where the Joint is ubicated and makes the permutation of the next or previous reference system orientation. We can see de six reference systems in blue:

*According to the sign criteria, twists are considered positive counterclockwise on the y, z, and x axes.*

*Illustration 3: Twist sign criteria*

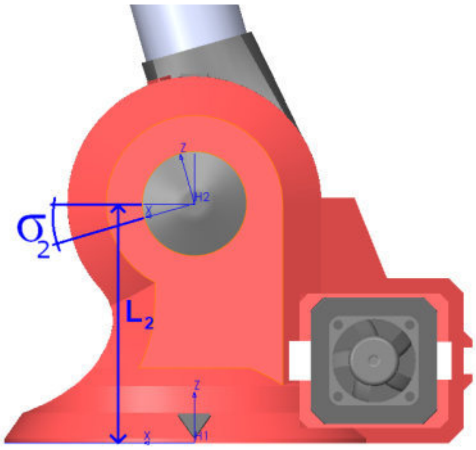
*Illustration 2: RS of the robot*

${}^{0}H_{1}$



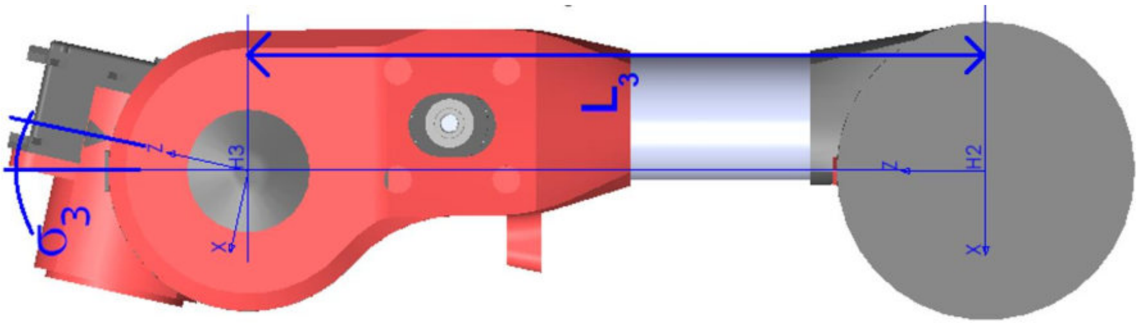
$$\begin{pmatrix} \cos(\sigma_1) & \sin(\sigma_1) & 0 & 0 \\ -\sin(\sigma_1) & \cos(\sigma_1) & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

${}^{1}H_{2}$



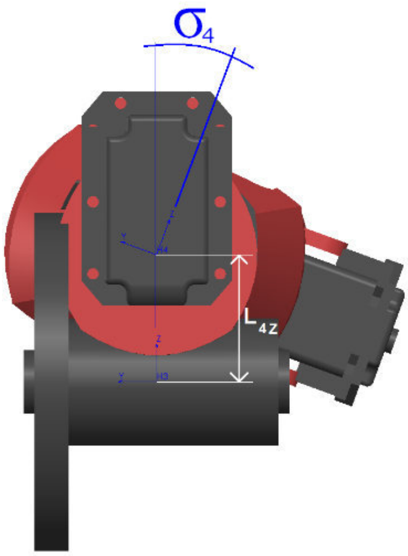

$$\begin{pmatrix} \cos(\sigma_2) & 0 & \sin(\sigma_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\sigma_2) & 0 & \cos(\sigma_2) & L_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

${}^{2}H_{3}$



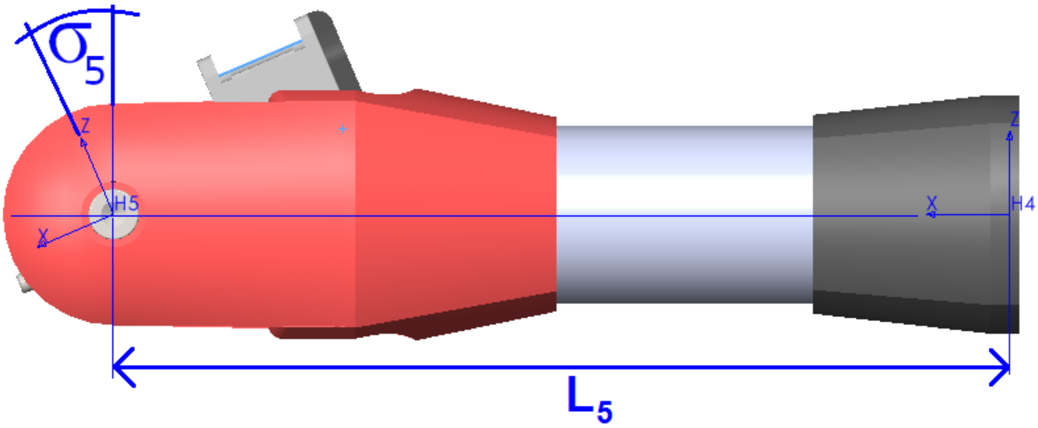

$$\begin{pmatrix} \cos(\sigma_3) & 0 & \sin(\sigma_3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\sigma_3) & 0 & \cos(\sigma_3) & L_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

${}^{3}H_{4}$



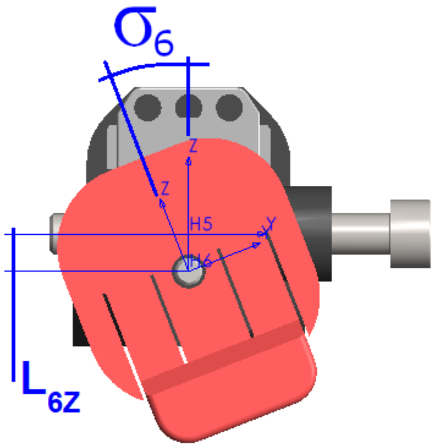

$$\begin{pmatrix} 1 & 0 & 0 & L_{4X} \\ 0 & \cos(\sigma_4) & -\sin(\sigma_4) & 0 \\ 0 & \sin(\sigma_4) & \cos(\sigma_4) & L_{4Z} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

${}^{4}H_{5}$




$$\begin{pmatrix} \cos(\sigma_5) & 0 & \sin(\sigma_5) & L_5 \\ 0 & 1 & 0 & 0 \\ -\sin(\sigma_5) & 0 & \cos(\sigma_5) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

${}^{5}H_{6}$




$$\begin{pmatrix} 1 & 0 & 0 & L_{6X} \\ 0 & \cos(\sigma_6) & -\sin(\sigma_6) & 0 \\ 0 & \sin(\sigma_6) & \cos(\sigma_6) & L_{6Z} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## 5.2 DENAVIT–HARTENBERG CONVENTION

In order to normalize the criteria used and to adapt our calculations to standard simulators and reference manuals it is very appealing the use of the Denavit-Hartenberg (***DH***) convention[2].

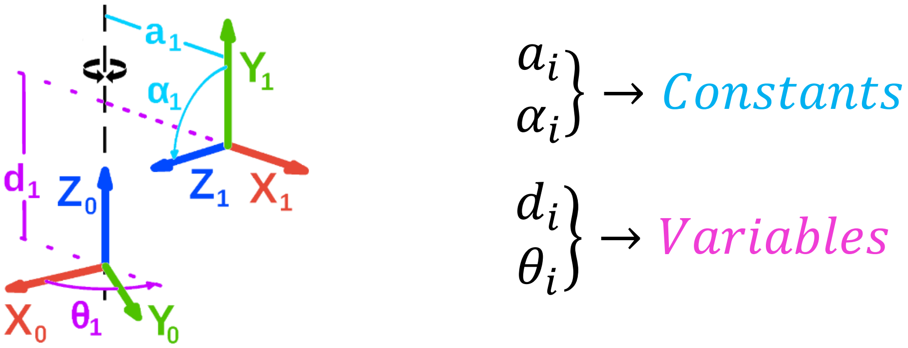



*Illustration 4: Denavit-Hartenberg Convention*

$$
{}^{i-1}H_i = \begin{pmatrix} Cos(\theta_i) & -Sin(\theta_i)\cdot Cos(\alpha_i) & Sin(\theta_i)\cdot Sin(\alpha_i) & a_i\cdot Cos(\theta_i) \\ Sin(\theta_i) & Cos(\theta_i)\cdot Cos(\alpha_i) & -Cos(\theta_i)\cdot Sin(\alpha_i) & a_i\cdot Sin(\theta_i) \\ 0 & Sin(\alpha_i) & Cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

Now the RS of the robot are relocated according to this convention. A simplified model is created to facilitate the visualization of the robot.

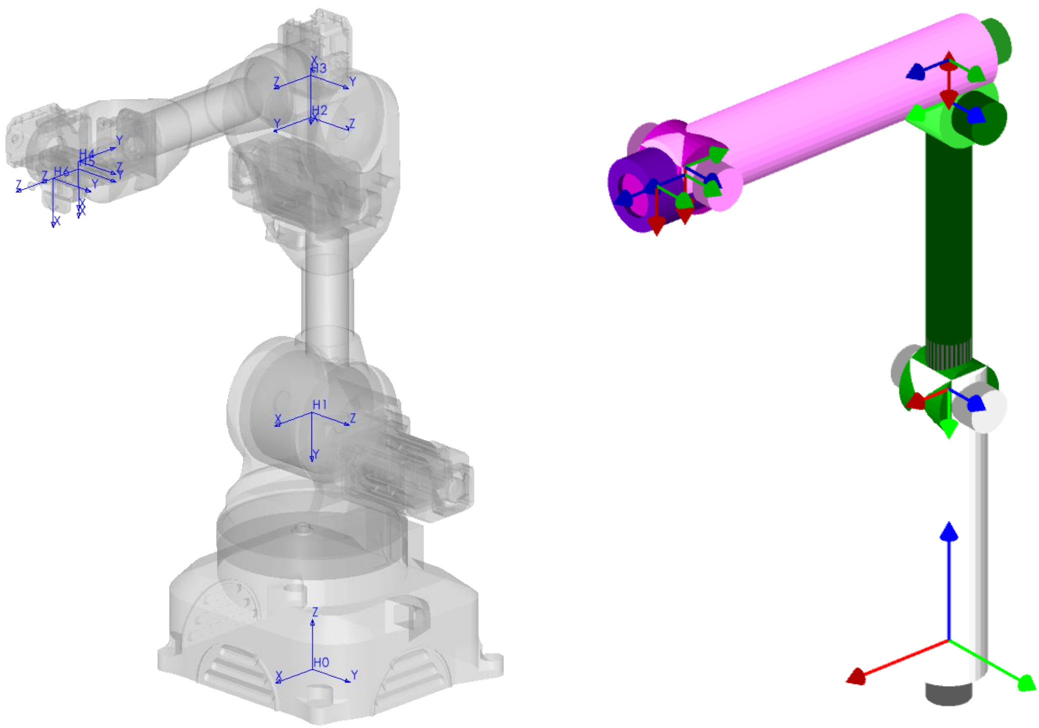
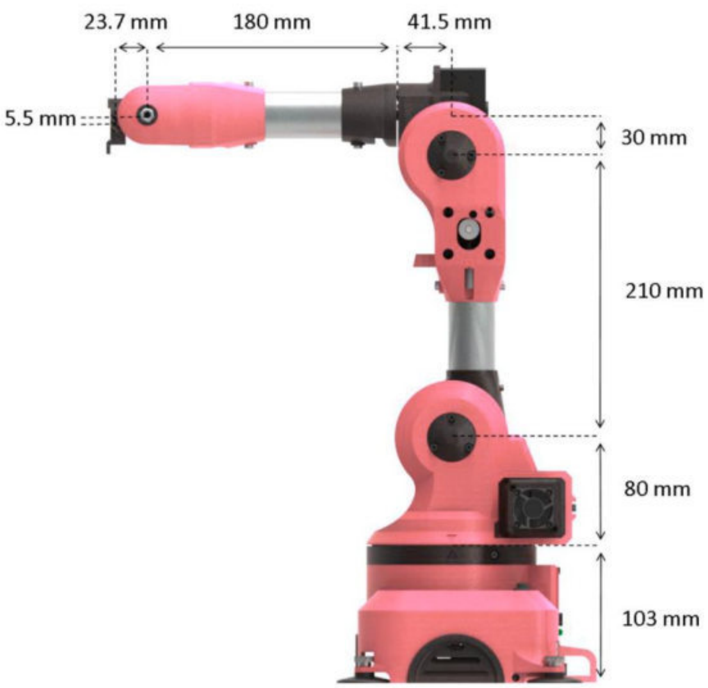


*Illustration 5: DH Robot*

### 5.2.1 MECHANICAL SPECIFICATIONS

These are the geometric specifications of the robot. All the lengths are essential to calculate the kinematics and workspace of the robot. The device is fully defined below:

**Dimensions**




**Max Rotation**

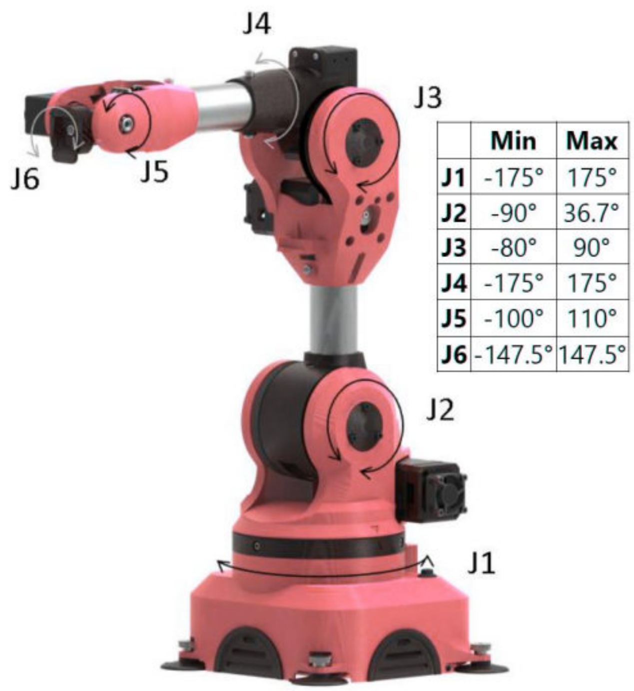



*Table 1: Mechanical Specifications*

With this parameter the DH table can be filled out.

| | $\theta_i$ | $d_i$ | $\alpha_i$ | $a_i$ |
|---|---|---|---|---|
| ***Link 1*** | $\theta_1$ | 0,183 | -90º | 0 |
| ***Link 2*** | $\theta_2 - 90º$ | 0 | 0 | 0,21 |
| ***Link 3*** | $\theta_3 + 180º$ | 0 | +90º | -0,03 |
| ***Link 4*** | $\theta_4$ | 0,2215 | -90º | 0 |
| ***Link 5*** | $\theta_5$ | 0 | +90º | 0,0055 |
| ***Link 6*** | $\theta_6$ | 0,0237 | 0 | 0 |

*Table 2: Robot DH Parameters*

Homogeneous transformation matrices:

$$^{0}H_1 = \begin{pmatrix} \cos(\theta_1) & -\cos(\alpha_1)\sin(\theta_1) & \sin(\alpha_1)\sin(\theta_1) & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\alpha_1)\cos(\theta_1) & -\sin(\alpha_1)\cos(\theta_1) & a_1\sin(\theta_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{1}H_2 = \begin{pmatrix} \cos(\theta_2 - 90) & -\sin(\theta_2 - 90)\cos(\alpha_2) & \sin(\theta_2 - 90)\sin(\alpha_2) & a_2\cos(\theta_2 - 90) \\ \sin(\theta_2 - 90) & \cos(\theta_2 - 90)\cos(\alpha_2) & -\cos(\theta_2 - 90)\sin(\alpha_2) & a_2\sin(\theta_2 - 90) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{2}H_3 = \begin{pmatrix} \cos(\theta_3 + 180) & -\sin(\theta_3 + 180)\cos(\alpha_3) & \sin(\theta_3 + 180)\sin(\alpha_3) & a_3\cos(\theta_3 + 180) \\ \sin(\theta_3 + 180) & \cos(\theta_3 + 180)\cos(\alpha_3) & -\cos(\theta_3 + 180)\sin(\alpha_3) & a_3\sin(\theta_3 + 180) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{3}H_4 = \begin{pmatrix} \cos(\theta_4) & -\cos(\alpha_4)\sin(\theta_4) & \sin(\alpha_4)\sin(\theta_4) & a_4\cos(\theta_4) \\ \sin(\theta_4) & \cos(\alpha_4)\cos(\theta_4) & -\sin(\alpha_4)\cos(\theta_4) & a_4\sin(\theta_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{4}H_5 = \begin{pmatrix} \cos(\theta_5) & -\cos(\alpha_5)\sin(\theta_5) & \sin(\alpha_5)\sin(\theta_5) & a_5\cos(\theta_5) \\ \sin(\theta_5) & \cos(\alpha_5)\cos(\theta_5) & -\sin(\alpha_5)\cos(\theta_5) & a_5\sin(\theta_5) \\ 0 & \sin(\alpha_5) & \cos(\alpha_5) & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{5}H_6 = \begin{pmatrix} \cos(\theta_6) & -\cos(\alpha_6)\sin(\theta_6) & \sin(\alpha_6)\sin(\theta_6) & a_6\cos(\theta_6) \\ \sin(\theta_6) & \cos(\alpha_6)\cos(\theta_6) & -\sin(\alpha_6)\cos(\theta_6) & a_6\sin(\theta_6) \\ 0 & \sin(\alpha_6) & \cos(\alpha_6) & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

HTM’s with numerical values:

$$^{0}H_1 = \begin{pmatrix} \cos(\theta_1) & 0 & -1 \cdot \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & 0.183 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{1}H_2 = \begin{pmatrix} \cos\left(\theta_2 - \frac{\pi}{2}\right) & -\sin\left(\theta_2 - \frac{\pi}{2}\right) & 0 & 0{,}21\cos\left(\theta_2 - \frac{\pi}{2}\right) \\ \sin\left(\theta_2 - \frac{\pi}{2}\right) & \cos\left(\theta_2 - \frac{\pi}{2}\right) & 0 & 0{,}21\sin\left(\theta_2 - \frac{\pi}{2}\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{2}H_3 = \begin{pmatrix} -1 \cdot \cos(\theta_3) & 0 & -1 \cdot \sin(\theta_3) & 0.03\cos(\theta_3) \\ -1 \cdot \sin(\theta_3) & 0 & \cos(\theta_3) & 0.03\sin(\theta_3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{3}H_4 = \begin{pmatrix} \cos(\theta_4) & 0 & -1 \cdot \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & \cos(\theta_4) & 0 \\ 0 & -1 & 0 & 0.2215 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{4}H_5 = \begin{pmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) & 0.0055\cos(\theta_5) \\ \sin(\theta_5) & 0 & -1 \cdot \cos(\theta_5) & 0.0055\sin(\theta_5) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{5}H_6 = \begin{pmatrix} \cos(\theta_6) & -1 \cdot \sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0.0237 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In repose pose:

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta_5 = \theta_6 = 0$$

$$^{0}H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0{,}183 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{3}H_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0{,}2215 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{1}H_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -0{,}210 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{4}H_5 = \begin{pmatrix} 1 & 0 & 0 & 0{,}0055 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{2}H_3 = \begin{pmatrix} -1 & 0 & 0 & 0{,}030 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{5}H_6 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0{,}0237 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Full HTM:

$$^{0}H_6 = \ ^{0}H_1 \cdot \ ^{1}H_2 \cdot \ ^{2}H_3 \cdot \ ^{3}H_4 \cdot \ ^{4}H_5 \cdot \ ^{5}H_6$$

$$^{0}H_6 \begin{pmatrix} 0 & 0 & 1 & 0{,}2452 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0{,}4175 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

All units are in International Standard (SI).

## 5.3 WORKSPACE

The robot workspace is basically defined as the space in which the robot operates. Every articulated robot has a defined workspace which contains the amount of room it is able to move around an area. This volume limit is made up of the infinite maximum reach points of the robot and never should be penetrated by a human being if the robot has not got specific safety measures. In the same way all the elements the robot should manipulate must be contained in the workspace.

In the following figure (created specifically for this dissertation) is shown the frontal and upper view of the robot workspace. Has been drawn moving the robot to the maximum reach positions. To make these drawings (which can be found in detail in blueprints section) the gripper two clamp is closed, reaching this way its maximum end advance value. Another assumption is that the robot will be ubicated over a solid floor and will not have any chance to reach a point under its base.

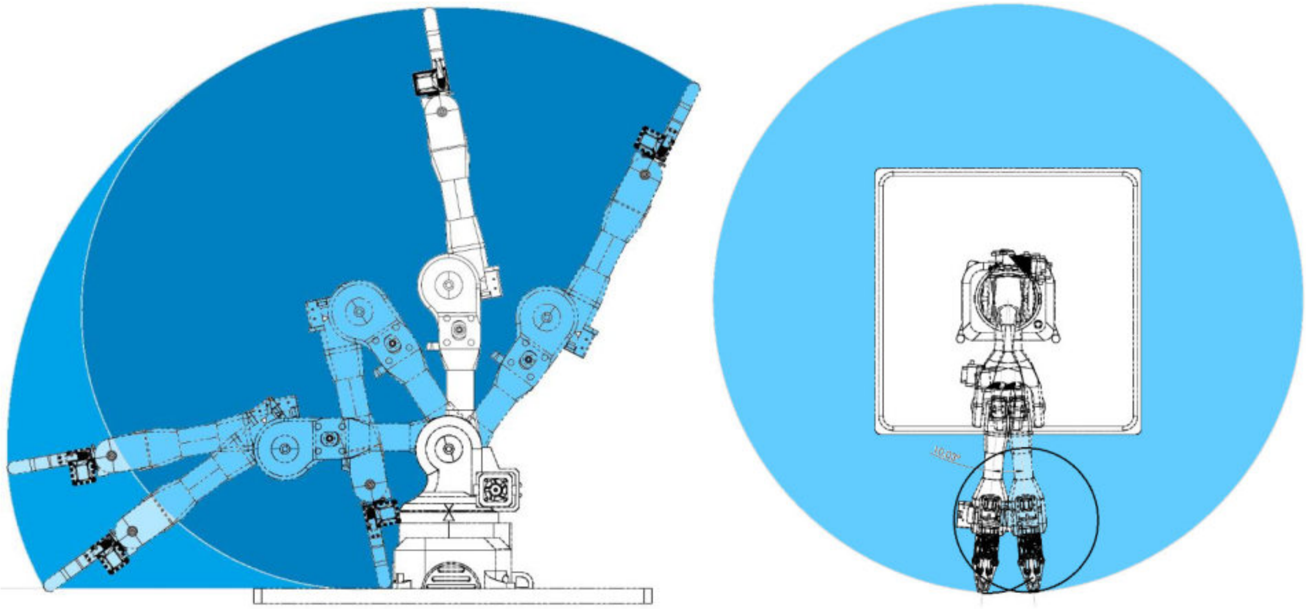


*Illustration 6: Workspace views*

There is a small volume portion that the robot cannot reach. It is a small rim defined by the maximum positive rotation angle for the second joint (36,7º described in the mechanical specifications) and the distance between the rotation axis of the fifth joint and the Second gripper clamp end (127 mm when the clamp is closed). We can pay no heed to this small unreachable space due to its reduced

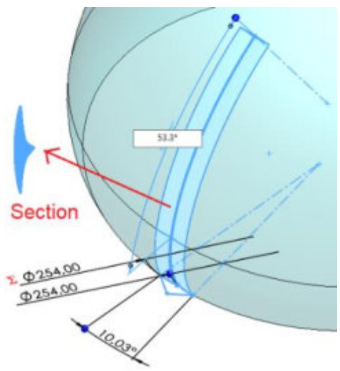



*Illustration 7: Unreachable backwards rim*

volume compared with the reachable workspace. This rim is shown in the following ilustration.

The workspace plays an important role in selecting a robot appropriate for a planned task. The kinematics of parallel robots is significantly different from the kinematics of serial manipulators and merits additional attention. This is not the case of study but proves the importance of the analysis of the workspace of an anthropomorphic robot.

With all this in mind we conclude that the workspace has an occupancy volume of 0,553 $m^3$ and an outer surface (ground round included) of 3,51 $m^2$. The robot can move along this area and will occasionally occupy every single point. This does not mean that the robot can reach every single portion of space with the gripper. There will be areas that the arm can occupy but cannot reach with the clamp. The robot ***dexterous workspace*** comprises all the points that can be reached with any arbitrary orientation of the robot end-effector. This volume will be always smaller than the workspace volume represented below:

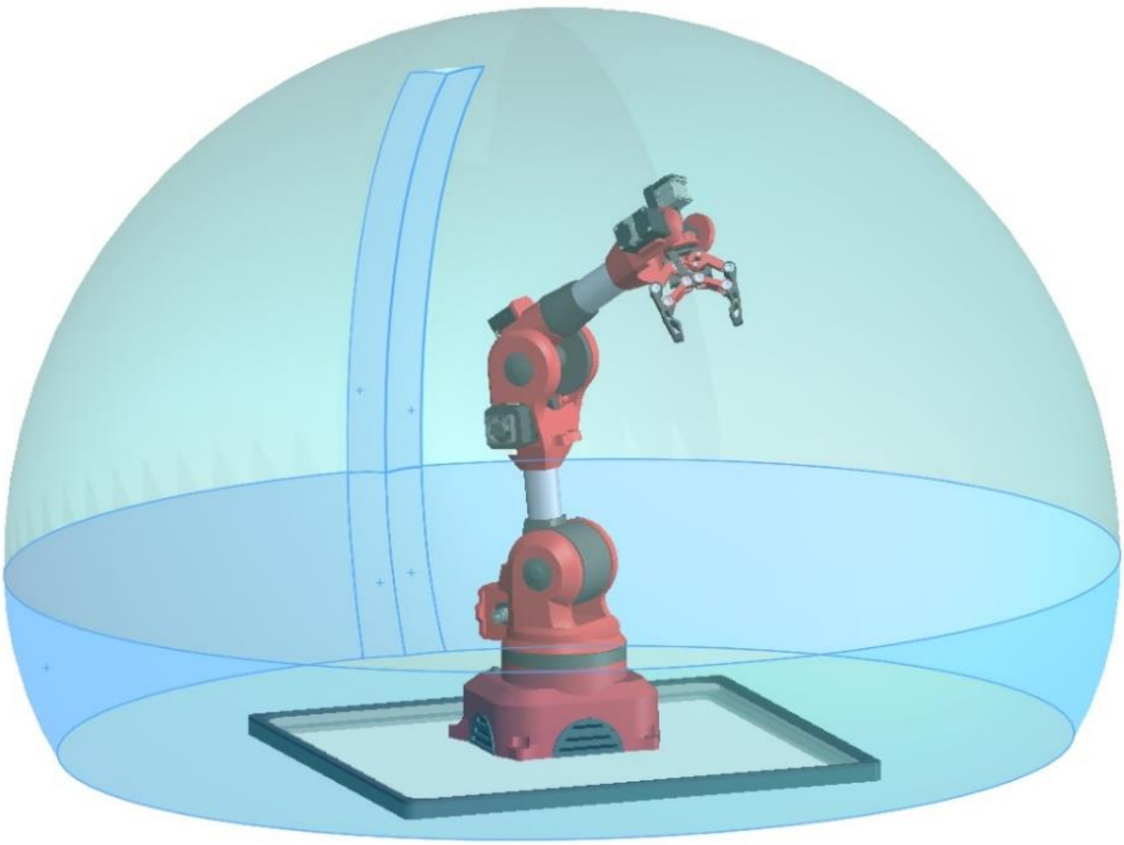


*Illustration 8: 3D workspace representation*

# 6 ACTUATORS AND SENSORS

The features of any engineered mechanical system (described in the introduction paragraph) are presented in terms of the power supply, power amplifier, servomotor[3] and transmission. In their control versatility perspective, two types of electric motors are used, namely, *electric servomotors* for actuating the joints of small and medium size (fourth, fifth, sixth joints and the gripper); and *electric steppers* for actuating the joints of large torque demand (first, second and third). Successively, *proprioceptive sensors* allow measurement of the rotation angles characterizing the internal state of the manipulator, namely, *encoders* and *resolvers* for joint position measurement, *tachometers* for joint velocity measurement. In industrial robots are also used *exteroceptive* transducers which are not incorporated in our robot. This type of sensors are presented including *force sensors* for end-effector force measurement, *distance sensors* for detection of objects in the workspace, and *vision sensors* for the measurement of the characteristic parameters of such objects, whenever the manipulator interacts with the environment. Artificial vision systems are also often used but will not be studied in this dissertation.

## 6.1 STEPPERS NEMA 17

*Illustration 9: NEMA 17 Stepper*

The stepper motor can be clearly defined as an incremental electromagnetic transducer that, when receiving an electrical pulse at the input, produces a rotation (angular increment) about an axis. This allows precise and efficient control of a movement on the motor shaft, achieving a constant and repeatable step. This device is excellent for applications that require an accurate control of the motion direction, speed, and even position of the rotating machine. The stepper motors are powered by a direct current source and controlled by a "driver" that allows the pulses received at the input to be converted into angular movements on the output axis. As the name suggests, stepper motors rotate in discrete steps, with each step corresponding to a pulse supplied to one of their stator windings.

NEMA 17 are a bipolar mixed or hybrid type motor. This device allows us to set the angular position (in multiples of the pitch angle value) of the rotor without a closed feedback loop. It is a simple and accurate open loop system. The precision of this motor is based on the intelligent arrangement of teeth on the rotor and stator. We go on to make a superficial analysis of its components and how it works. The stepper motor used can move a fraction of a degree per pulse. In turn, by varying the speed of the pulses, the motor can be made to advance very slowly, one step at a time, or to rotate gradually at high speeds. This kind of motors has been chosen due to its physical properties.

### 6.1.1 HOLDING TORQUE

Static properties are defined as those characteristics of the system that concern the system in equilibrium state, where the sum of all the forces that affect it is zero. This positioning motor allows, within its working regime, to establish a series of static balance positions, under certain excitation conditions of the windings. In this case, the rotation between two equilibrium states of the rotor corresponds to the so-called step.

The resistive torque or holding torque is the one necessary to produce a rotor movement of one step, when the motor being in a state of equilibrium (obviously within the running regime) goes to another. This data is specific to each particular engine and is provided by the manufacturer in the product's technical data sheets. When a torque that exceeds the holding torque (that which opposes rotation) is applied to the rotor shaft, the rotor will rotate. The holding torque is usually higher than the working torque (rotor in motion, explained in the next section) and acts as a brake to keep the rotor in one of the many stable or balanced positions. This static or resistive torque developed by the stepper motor is modified depending on the position of the rotor, the dimensions of the stator and the number of teeth on the rotor, as well as the value of the current flowing through the windings. The higher the circulating current, the higher the torque.

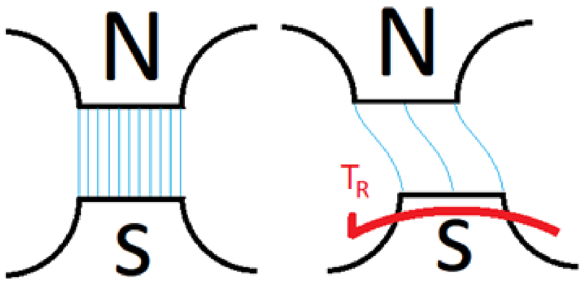


*Illustration 10: Holding torque*

Actually, in the equilibrium position, the rotor and stator teeth are perfectly aligned and no torque is produced. When the rotor is displaced slightly out of the equilibrium position, a force is developed between the teeth of the rotor and those of the stator, as a result of the magnetic attraction or repulsion between them; the sum of all teeth interactions results in a torque that tries to move the rotor to the equilibrium position. The system always tends to remain aligned, in balance.

### 6.1.2 DETENTION TORQUE

It is a unique torque of the permanent magnet stepper motor. It is due to the exclusive action of the rotor when no electric current circulates through the stator windings.

### 6.1.3 POSITION ERROR

In this kind of high demanding applications, a position error may arise due to a high mechanical load on the motor shaft, which may be subjected to an external torque produced by the robot linkers inertias and end charges. These types of loads can overcome the resistive torque produced by the motor and result in a small offset angle relative to the balance position. A very correct estimate of the static position error can be obtained by approximating the relationship between the static or resistive torque and the position of the rotor to a sinusoidal curve.

In general terms we can conclude that for a motor with ***n*** rotor teeth and a maximum resistive torque $T_P$, at a displacement $\theta$ from the equilibrium position; the torque developed by the motor will be given by the expression:

$$T = -T_P \cdot \sin(n \cdot \theta) \qquad [S1.1]$$

When the load torque $T_C$ applied to the rotor moves the rotor to a new equilibrium position $\theta_e$, it will equal the motor torque.

$$T\_C = T = -T\_P \cdot sin\ (n \cdot \theta\_e\ ) \quad [S1.2]$$

Finally, the position error will be given by the expression:

$$\theta_e = \frac{sin^{-1}\left(-\frac{T_C}{T_P}\right)}{n} \quad [S1.3]$$

As already indicated, this equation is a mere approximation of the real error. The deviation with respect to the real value increases as the rotor moves away from the original equilibrium position (motor without load), until reaching the maximum torque, from which it begins to decrease.

### 6.1.4 WORKING TORQUE

Dynamic properties are defined as those characteristics of the stepper that affect it outside of the equilibrium state described in the previous section. For this case, the motor will be spinning. The rotation movement of the shaft can never become strictly regular, since the rotor rotates through various equilibrium states, subtly and very briefly interlocking in each one of them. However, for step values less than 10º, we can consider that the rotation is continuous (constant or accelerated) and controlled (at a known specific speed).

The working or dynamic torque (working torque) is the maximum couple of forces (or torque) that the motor is capable of developing without losing step or what is the same: without the rotor stopping responding to the excitation impulses of the stator, with load or not. The typical stator excitation current waveform for a variable reluctance motor operating at low speed is rectangular in shape and a direct relationship between dynamic and resistive torques can be obtained. For more information, go to Annex I.

### 6.1.5 INERTIA EFFECT

When studying the whole robot dynamics, it is essential to keep in mind the different linkers moments of inertia. But also the joints inertias must be considered, despite all are usually forgotten. To understand Stepper inertias we start from the ideal assumption in which the motor indicated below operates without load and the rotor has a low inertia and almost zero friction with the bearings.

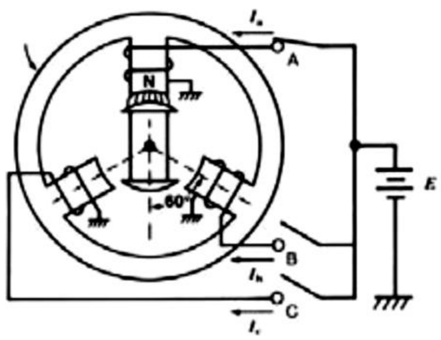



*Illustration 11: Stepper simplified schema*

Initially the rotor is aligned with pole 1 (0°). At the moment in switch A is opened and switch B is closed, the rotor, induced by the magnetic field generated by the coils, becomes influenced by an angular acceleration counterclockwise towards pole 2. After a few moments it picks up speed and soon becomes aligned with pole 2, where it should stop. However, at this instant the rotor, although it is no longer accelerated, is still moving with considerable speed and will miss the intended alignment. By doing this, the magnetic field of pole 2 will exert a force in the opposite direction to that previously described, trying to align the rotor with pole 2, slowing the rotor down and stopping it in an angle out-of-step. The rotor will stop aligned between poles 2 and 3 (certainly closer to 2) and start moving in the opposite direction always seeking equilibrium with pole 2. Once it has picked up speed, it will again pass the position of equilibrium with pole 2, where the magnetic field will exert a counterclockwise force again.

Thus, the rotor will oscillate like a simple pendulum around the center line of pole 2. The oscillations will however gradually decrease in amplitude due to the non-zero friction in the bearings and bearings. The oscillations continue and their amplitude gradually decreases until the motor has timed out. Speed can be given in revolutions per second, but for stepper motors it makes more sense to speak of degrees per second or steps per second. The velocity is momentarily zero at the instants of greatest amplitude. The speed is maximum whenever the rotor crosses the center line of pole 2. The oscillations last a relatively long time before the rotor stops. It is the same behavior of a simple pendulum.

This effect becomes more influential when without making any other changes we increase the inertia of the rotor by mounting a flywheel on the shaft connected to our joint by the transmission. Under this assumption in the harmonic motion described, both the period and the amplitude of the oscillations increase proportionally to the inertia.

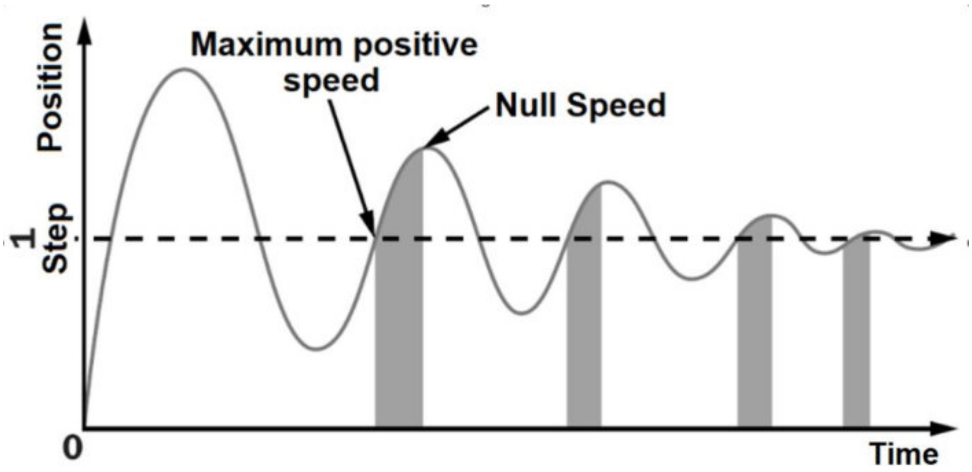



*Illustration 12: Motor response to a pulse type input*

Other dynamic effects like slew-rate, magnetic resonation is described in the first Addendum.

### 6.1.6 INSTALLATION

As can be seen in the illustrations, the motor consists of two protective covers which allow the anchorage of the entire system through the use of 4 threaded holes (M3) in the upper cover, the shaft output cover. The assembly is done in this way:

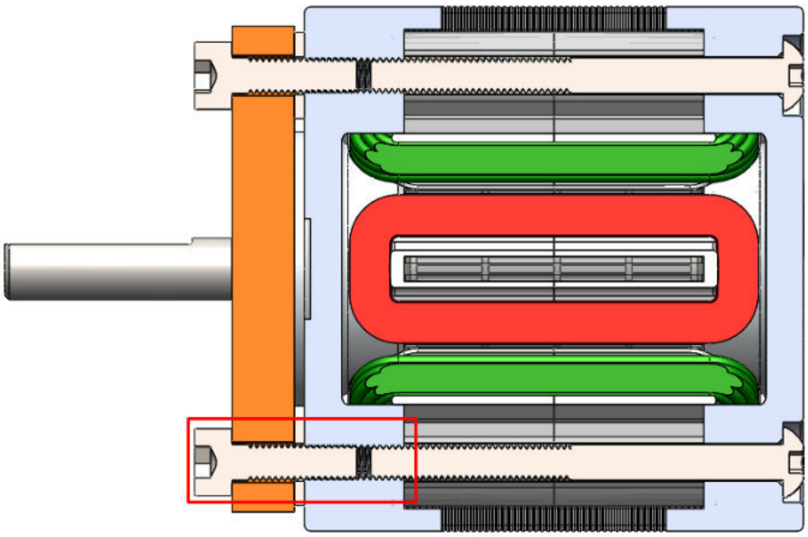


*Illustration 13: Assembly Nema 17*

The system is assembled by inserting the screws from the back and using a third of the threaded holes to screw the system. The rest of the 4 threaded cylinders (hollow) is used for the subsequent anchoring of the engine to the belt tensioners described in the following paragraph (indicated in orange).

## 6.2 TRANSMISSION

The joint movement of a manipulator demands low speeds with high torque rates. There is used a transmission in the three first joints to optimize the transfer of mechanical power from the motor to the joint. During this transfer, the power is dissipated as a result of friction and rubber belts deformations. The choice of proceeding to use transmission depends on the power requirements, the kind of desired motion, accuracy preliminary conditions, and the ubication of the motor with respect to the joint. The transmission even allows the outputs of the motor to be transformed both quantitatively (velocity and torque) and qualitatively (a rotational motion about the motor axis into a translational motion of the joint). In this case of application, the transmission transforms the spin movement quantitatively, in terms of speed and torque.

This system also allows the static and dynamic performance of a manipulator to be optimized, by reducing the effective loads when the motor is located upstream of the joint, like in our robot. Some motors are mounted to the base of the robot and they have to move the major mass percentage of the whole arm and need to apply high torque rates. The transmission system and it's achieved optimization are very useful to reduce the torque demand in each stepper and permit the installation of smaller and lighter motors, bringing the consequent energy efficiency rate increment and maximum load capability growth. It has been used a timing belts transmission system.

*Illustration 14: Transmission belts*

For the proper functioning of this mechanism is essential the use of the belts tensioners which by the easy tighten of a screw ensure the correct torque transmission and facilitate the assembly process. The stepper is tied down to the belt tensioner and both adjusted with a screw. The belt tensioner must be printed with a 100% infill density because gathers many axial and normal forces. Many replacements of this components should be also printed just in case. More information about this will be given forward.

In the Annex 1.6 paragraph is studied the steppers resolution, and in Annex 1.7 is described how to improve this stepper resolution. Well, the transmission system also improves the accuracy in joints one, two and three (powered by steppers) and eightfold it[4]. This is because the relation between the stepper timing number of teeth (16) and the number of belts fitting gaps (128) is 8. This means that a spin of 1,8 degrees in the stepper produces a rotation of only 0,225 degrees. This makes unnecessary the use of half step excitation sequence in the stepper. See the first Addendum, paragraphs 1.6 and 1.7.

## 6.3 DYNAMIXEL


*Illustration 15: Dynamixel XL 430*

Dynamixel is a reliable brand of servomotors for robotic projects. These servomotors, designed by the Korean company *ROBOTIS ©*, are among the most powerful and efficient on the market. Basically, this servomotor incorporates its position, velocity and torque control; and the hardware to connect the device to a Serial TTL net. In this device are both incorporated the actuator and the sensorics.

It has a MCU ARM CORTEX-M3 (72 [MHz], 32Bit) and a PID controller to communicate to the user via serial communication and to control the shaft movement.

### 6.3.1 PERFOMANCE

Here is a graph which describes the motor behaviour:

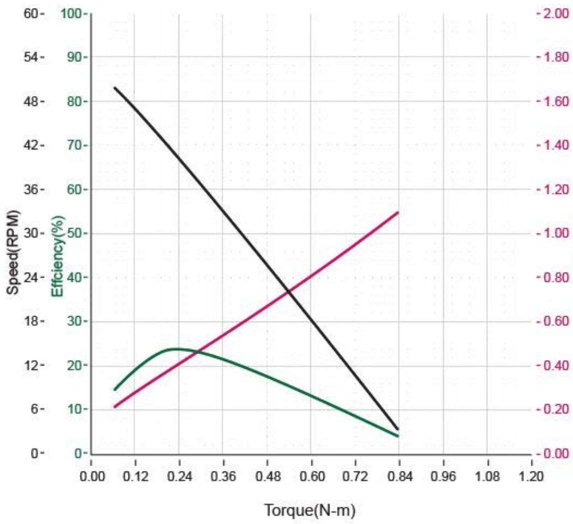



*Illustration 16: Perfomance curves Dynamixel XL 430*

The Max Torque and the Stall Torque of Performance Graph are different in measurement methods. Stall torque is a measured value of the momentary torque that it can reach. This is generally how RC servos are measured. When the motor moves, it rotates and the servo axis changes position, and in turn, produces a movement. The order consists of a small electrical impulse that indicates to the servo in which rotation position the motor should be placed. The Performance graph is also called as N-T curves, which is measured with the gradually increasing load. The actual motor operation environment is closer to the performance graph, not stall torque method. For this reason, the performance graph is broadly used in the industrial field. Generally, Max Torque of the Performance Graph is less than the Stall Torque.

### 6.3.2 CONTROL TABLE

The Control Table is a structure of data implemented in the device. Users can read a specific Data to get status of the device with Read Instruction packets and modify Data as well to control the device with WRITE Instruction Packets.

It is a structure that consists of different Data fields to store status or to control the device. Status of the device can be checked by reading a specific Data from the Control Table sending Read Instruction Packets. WRITE Instruction Packets otherwise enable users to control the device by changing a specific Data in the mentioned Control Table. A unique Address value is needed when accessing a specific Data in the Control Table with Instruction Packets. This is the specific Address in the Instruction Packet. The Control Table is divided into 2 differentiated Areas: RAM and EEPROM. Data in the RAM Area is not stored when rebooting the device. On the other hand, data in the EEPROM Area is maintained even when the device is powered disconnected from voltage. For good security measure data in the EEPROM Area can only be written to if Torque Enable (address 64) is cleared to ‘0’ (Torque OFF).

### 6.3.3 PROFILES

The Profile is a method to make velocity variations (accelerating and decelerating) a servomotor which reduces vibration, noise and load of the motor frequently induced when the controller makes abrupt changes in velocity and acceleration. It is also called ***Velocity Profile*** as it controls acceleration and deceleration supported on speed. DYNAMIXEL provides 3 different types of Profile. Profiles are usually made up by the combination of Profile Velocity (address 112) and Profile Acceleration (instruction 108, for more information, go to addendum 2.2). In the following image we can see the three different profiles. Step profile is never used in this case though.

| **Condition** | $V_{PRFL}(112) = 0$ | $(V_{PRFL}(112) \neq 0)$ & $(A_{PRF}(108) = 0)$ | $(V_{PRFL}(112) \neq 0)$ & $(A_{PRF}(108) \neq 0)$ |
|---|---|---|---|
| **Profile** | Step Instruction | Rectangular Profile | Trapezoidal Profile |

*Table 3: Select Profile*

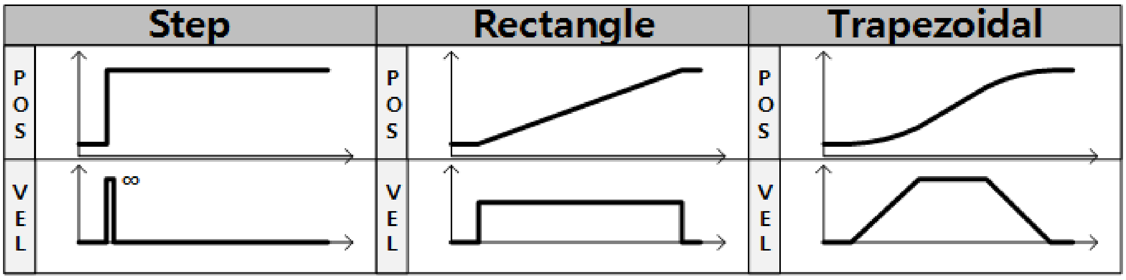



*Illustration 17: Velocity profiles*

When given target Position (116 instruction), DYNAMIXEL's profile derivates the position function and creates the desired velocity variation trajectory based on present velocity (initial velocity of the Profile). When DYNAMIXEL receives new updated target position while it is moving toward the previous Goal Position (address 116) and has not reached the previous dot, velocity smoothly varies for the new desired velocity trajectory. This is essential to maintain velocity modulus in a constant value. This method is called is called ***Velocity Override***.

Here is an example:

For a simple calculation, let's assume that the initial velocity of the Profile is '0'. Let's see the sequence in which Profile processes the new target position (116 address) in Position Control Mode.

1- An Instruction from the ROS is transmitted via DYNAMIXEL bus (RS485 TTL), then registered to Goal Position (address 116) (If Velocity-based Profile is selected).
2- Acceleration time is calculated from Profile Velocity (address 112) and Profile Acceleration (address 108) instructions.
3- Types of Profile are now decided based on the variables: Profile Velocity (address 112), Profile Acceleration (address 108) and total travel distance (the distance difference between desired position and present position).
4- The selected Profile type must be stored at Moving Status (address 123) by the ROS.
5- DYNAMIXEL is driven by the calculated desired trajectory from Profile.
6- Finally, desired velocity trajectory and desired position trajectory from Profile are stored at Velocity Trajectory (address 136) and Position Trajectory (address 140) respectively.

Velocity Control Mode only uses Profile Acceleration (108) where Step and Trapezoidal Profiles are supported as long as Velocity Override Profiles. Acceleration time(t1) can be calculated as below equation. If Time-based Profile is selected, Profile Velocity (address 112) is used to set the time span of the Profile(t3) (like in classic continuously acceleration movements calculus), while Profile Acceleration (address 108) sets accelerating time(t1) in millisecond. Profile Acceleration (108) will never exceed 50% of Profile Velocity (address 112) value.

Velocity-based Profile:

$$t_1 = 64 \cdot \frac{Profile\ Velocity\ (112)}{Profile\ Acceleration\ (118)} \qquad [D1.1]$$

Time-based Profile:

$$t_1 = Profile\ Acceleration\ (118) \qquad [D1.2]$$

The most used is the Trapezoidal, better known as SPLINE[5].

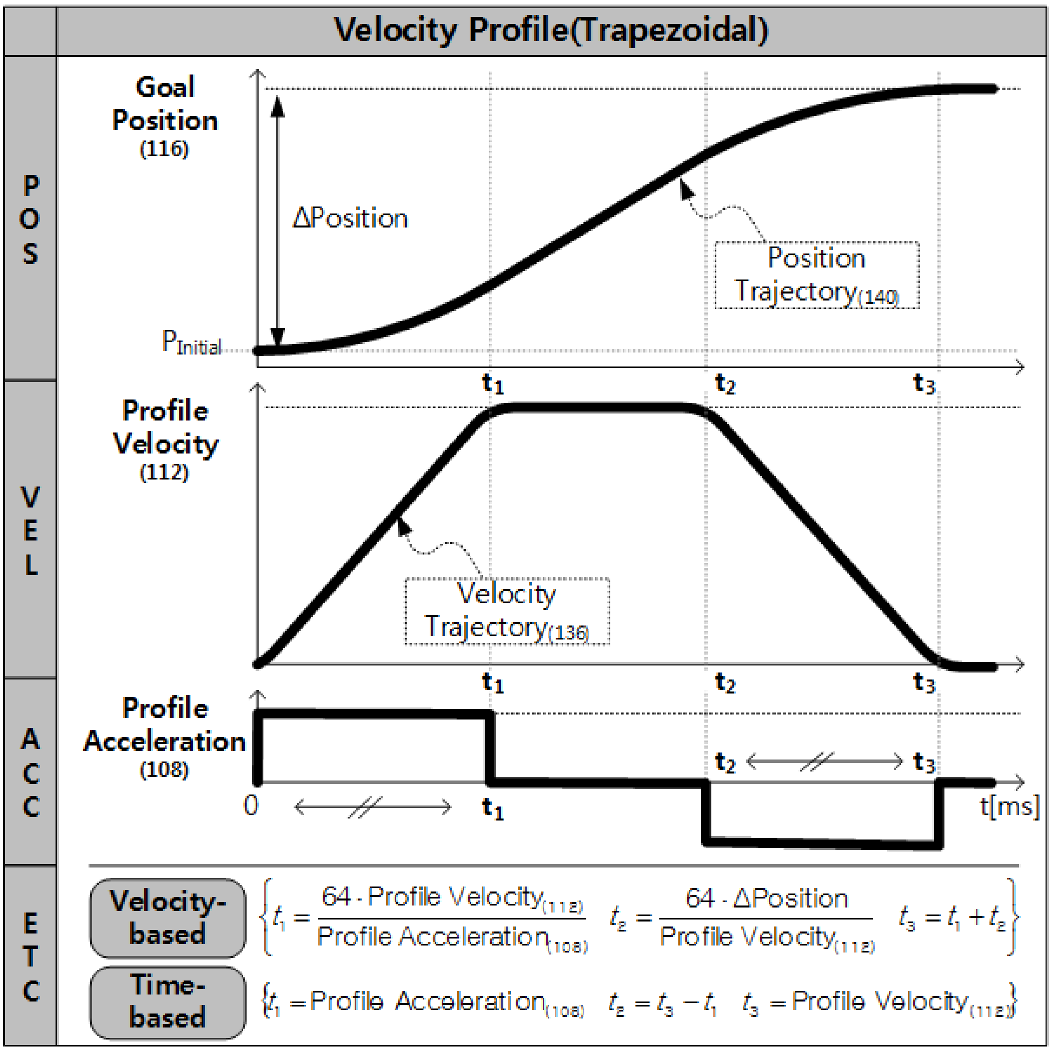



*Table 4: SPLINE equations*

### 6.3.4 OPERATING MODE

The change of operating mode (address 11) allow us to change between:

| Value | Operating Mode | Description |
|---|---|---|
| 1 | Velocity Control Mode | This mode controls velocity and ideal for wheel operation. |
| | (0° ~ 360°) | This mode is identical to the Wheel Mode(endless) from existing DYNAMIXEL. |
| 3(Default) | Position Control Mode | This mode controls position and identical to the Joint Mode. |
| | | Operating position range is limited by Max Position Limit(48) and Min Position Limit(52). |
| | | This mode is ideal for articulated robots that each joint rotates less than 360°. |
| 4 | Extended Position Control Mode | This mode controls position and identical to Multi-turn Mode. |
| | (Multi-turn) | 512 turns are supported(-256[rev] ~ 256[rev]) and ideal for multi-turn wrists or conveyer systems or a system that requires an additional reduction gear. |
| 16 | PWM Control Mode<br>(Voltage Control Mode) | This mode directly controls PWM output (Voltage Control Mode) |
| Bit 1(0x02) | - | Unused, always ‘0’ |
| Bit 0(0x01) | Normal/Reverse Mode | [0] Normal Mode: CCW(Positive), CW(Negative) |
| | | [1] Reverse Mode: CCW(Negative), CW(Positive) |

*Table 5: Operating Modes switch*

It is important to understand that switching Operating Mode will reset gains (PID, Feedfoward) (explained in subsequent paragraphs) properly to the selected Operating Mode. PWM is the abbreviation for Pulse Width Modulation that modulates PWM Duty to control motors. It changes pulse width to control average supply voltage to the motor and this technique is widely used in the motor control field. It is used Goal PWM (address 100) on PWM Control Mode in order to control supply voltage for DYNAMIXEL.

### 6.3.5 POSITION CODIFICATION

Present Position (instruction 132) represents 4 byte continuous range from -2,147,483,648 to 2,147,483,647 when Torque is turned off regardless of Operating Mode (address 11). However, Present Position (address 132) will be reset to an absolute position value within one full rotation in the following cases:

1- When Operating Mode (address 11) switches to Position Control Mode, Present Position (address 132) will be reset to an absolute position value within a full rotation.
2- When torque is turned on in Position Control Mode, Present Position (address 132) will be reset to an absolute position value within one full rotation.
3- Turning on the power supply or using Reboot Instruction.

#### 6.3.6 HOMING OFFSET

The Home Offset (address 20) sets the home position. The new offset value will be added to the Present Position (instruction 132).

$$Present\ position\ (132) = Actual\ Position + Homing\ Offset\ (20) \quad [D1.3]$$

#### 6.3.7 SHUTDOWN

The DYNAMIXEL can protect itself by detecting dangerous situations that could occur during the operation. In order be safer each Bit is automatically processed with the 'OR' logic when editing. In addition, REBOOT is the only method to reset Torque Enable (address 64) to '1' (Torque ON) after the emergency shutdown. Error register can be red in Dynamixel, called Error Status (address 70). The followings are possible error situations.

| Bit | Item | Description |
|---|---|---|
| Bit 7 | - | Unused, Always '0' |
| Bit 6 | - | Unused, Always '0' |
| Bit 5 | Overload Error(default) | Detects that persistent load that exceeds maximum output |
| Bit 4 | Electrical Shock Error(default) | Detects electric shock on the circuit or insufficient power to operate the motor |
| Bit 3 | Motor Encoder Error | Detects malfunction of the motor encoder |
| Bit 2 | Overheating Error(default) | Detects that internal temperature exceeds the configured operating temperature |
| Bit 1 | - | Unused, Always '0' |
| Bit 0 | Input Voltage Error | Detects that input voltage exceeds the configured operating voltage |

*Table 6: Error Status*

## 6.4 STEPPER DRIVERS

The Dynamixel include their own drivers, so in this paragraph we are going to focus on the Stepper Drivers, an specific PCB controller installed un the rear part of the motors.

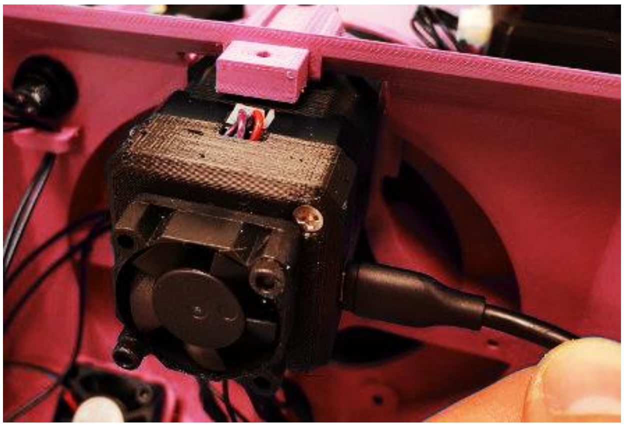


*Illustration 18: Stepper Driver*

There is used a specific board which has a magnetic sensor (AS5600) to measure the position of the stepper motor. The controller reads the spin of the magnetic field of the permanent magnets in the rotor’s stepper. The board drawings are shorn in blueprints appendix. This controller basically register the magnetic field orientation, which is the same as the rotor spin and sends the data to the ROS system via CAN bus interface.

This is a very safe system because if the stepper looses some steps or are forced the forced rotation is red and the ROS can apply the accurate offset to reduce the position error to zero. Every single rotation is captured, done by the robot or by external forces. Is the ROS who closes the feedback loop in the steppers control system, unlike Dynamixels which have their own closed loop control system. This control architectures are explained in the next paragraph.

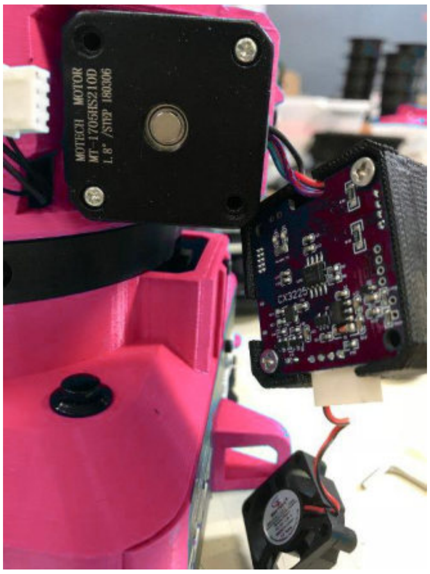


*Illustration 19: AS5600*

# 7 CONTROL ARCHITECTURE

In this paragraph we are going to study the different control loops used in the robot.

## 7.1 DYNAMIXEL PI CONTROLLER – VELOCITY

The Velocity control of Dynamixel actuators. The Velocity PI Gains (address 76, 78) indicate gains of Velocity Control Mode. We are going to abbreviate the velocity proportional Gain of DYNAMIXEL's internal controller[6] to $K_VP$ and that of the Control Table is abbreviated to $K_VP_{(TBL)}$.

| | Controller Gain | Conversion Equations |
|---|---|---|
| Velocity I Gain(address 76) | $K_VI$ | $K_VI = K_VI_{(TBL)} / 65,536$ |
| Velocity P Gain(address 78) | $K_VP$ | $K_VP = K_VP_{(TBL)} / 128$ |

*Table 7: Dynamixel PI parameters*

There has been created a block diagram describing the speed controller in Velocity Control Mode. When the instruction transmitted from the ROS is received by DYNAMIXEL, it takes following steps until driving the horn.

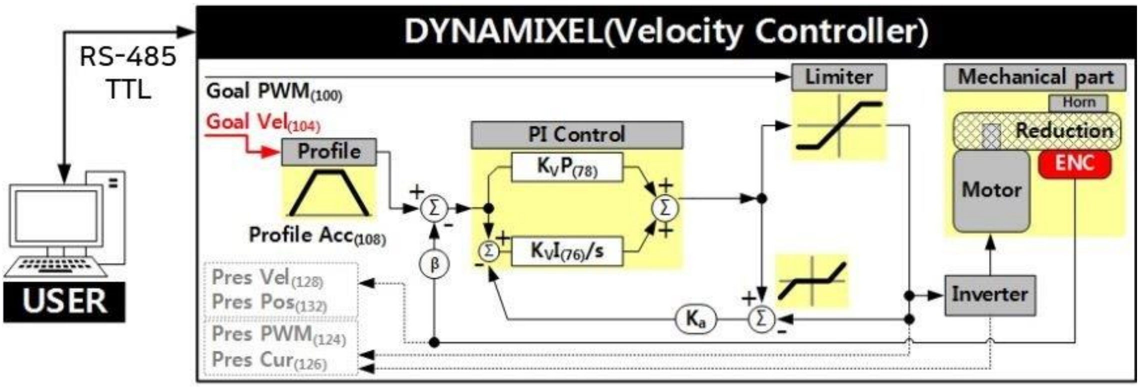



*Illustration 20: Dynamixel PI Velocity Controller*

1- The Instruction from the ROS is transmitted via TTL bus, then registered to Goal Velocity (address 104).

2- Goal Velocity (address 104) is converted to desired velocity trajectory by Profile Acceleration (address 108).

3- The desired velocity trajectory is directly stored in Velocity Trajectory (address 136).

4- PI controller calculates the PWM output for the motor based on the desired velocity trajectory.

5- Goal PWM (address 100) sets a limit on the calculated PWM output and decides the final PWM value.

6- Finally, PWM value is applied to the motor through an Inverter and the horn of DYNAMIXEL is driven.
7- Results are stored at Present Position (address 132), Present Velocity (address 128), Present PWM (address 124) and Present Load (address 126).

## 7.2 DYNAMIXEL PID CONTROLLER – POSITION

The Position PID Gain (address 80, 82, 84), and Feedforward first and second Gains (address 88, 90).

| | Controller Gain | Conversion Equations | Range |
|---|---|---|---|
| Position D Gain(80) | $K_PD$ | $K_PD = K_PD_{(TBL)} / 16$ | 0 ~ 16,383 |
| Position I Gain(82) | $K_PI$ | $K_PI = K_PI_{(TBL)} / 65,536$ | 0 ~ 16,383 |
| Position P Gain(84) | $K_PP$ | $K_PP = K_PP_{(TBL)} / 128$ | 0 ~ 16,383 |
| Feedforward 2nd Gain(88) | $K_{FF2nd}$ | $K_{FF2nd(TBL)} / 4$ | 0 ~ 16,383 |
| Feedforward 1st Gain(90) | $K_{FF1st}$ | $K_{FF1st(TBL)} / 4$ | 0 ~ 16,383 |

*Table 8: Dynamixel PID parameters*

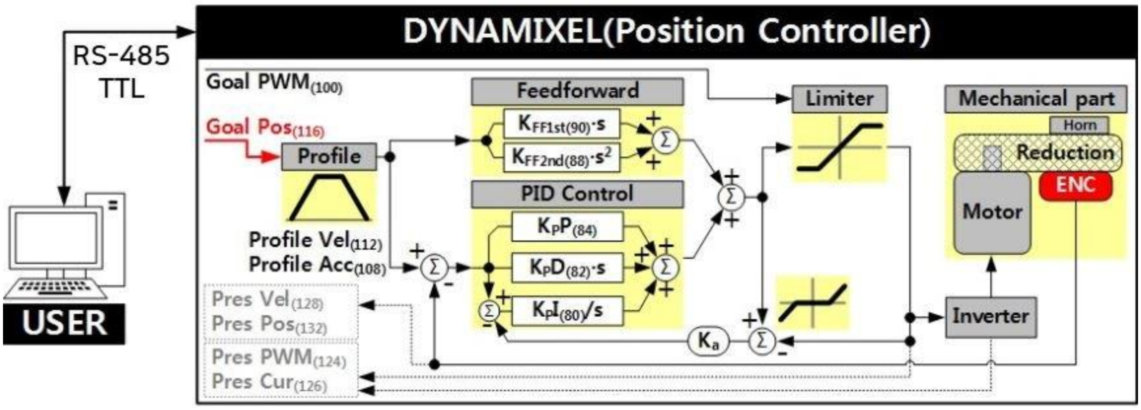



*Illustration 21: Dynamixel PID Controller*

The above figure is a block diagram describing the position controller in Position Control Mode and Extended Position Control Mode. When the instruction is received from the ROS the Dynamixel follows the following steps until driving the horn.

1- The instruction from the ROS is received via TTL bus and then registered to Goal Position (address 116).
2- Goal Position (address 116) is converted according to the desired position trajectory and the desired velocity trajectory by Profile Velocity (address 112) and Profile Acceleration (address 108)[6].
3- The calculated position trajectory and calculated velocity trajectory is stored at Position Trajectory (address 140) and Velocity Trajectory (address 136) respectively.

4- Now, the Feedforward and PID controller calculate PWM output for the motor based on desired trajectories.
5- Goal PWM (address 100) sets a limit on the calculated PWM output and decides the final PWM value.
6- Finally, the PWM value is applied to the motor through an Inverter, and the horn of DYNAMIXEL is driven.
7- Results are stored at Present Position (address 132), Present Velocity (address 128), Present PWM (address 124) and Present Load (address 126).

In case of using the PWM Control Mode, both Feedforward controller and PID controller are disactivated while Goal PWM (address 100) value is directly controlling the motor through an inverter. In this situation, ROS directly controls the supplying voltage to the motor.

## 7.3 STEPPERS CONTROL

As it has been previously said, the control of steppers is basically a simple loop in which the ROS stablish a communication with the steppers boards which gives to them the position of each stepper in real time. The ROS compares the target position with the real one and applies the correspondent offset in the setpoint to null de position error.

# 8 IMPROVEMENTS

As is has been said, the aim of this project is not only understand but improve the Niryo One robot design and functionalities. During the assembly and testing process some important issues have been detected.

## 8.1 WEAK POINTS

### 8.1.1 TEMPERATURE

In the Specifications, paragraph 1.4, we can see than the Raspberry Pi controller is design to work between 0°C and 85°C. Specifically, the CPU is qualified from -40°C to 85°C and the LAN is qualified from 0°C to 70°C. If we analyse the original design of the base:

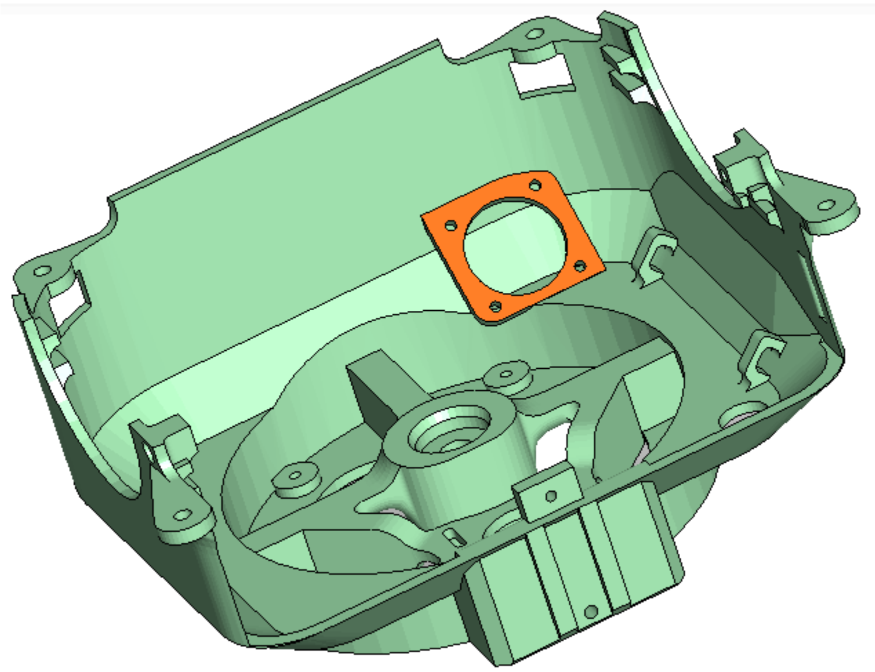


*Illustration 22: Base fan support*

As we can see there is a fan support to cool de controller. However, the fan when installed is fitted to de air and does not allow the air to flow freely, what is a grave design failure because that fan will only generate more heat[7].

### 8.1.2 STABILITY

The weigh distribution of the robot elevates a lot the centre of mass of it. It is not capable to support the movement al the limits of workspace nor the inertias at high speeds without falling. This issue makes the robot completely useless.

### 8.1.3 WRONG MANUFACTURING

In the original design and previous assemblies, the wrist plastic piece was broken due to fatigue because of a grave error in slicing configuration. The piece has to support lots of charges but is initially recommended to be printed with low density and squares infill pattern. This results in a weak component.

*Illustration 23: Broken piece*

## 8.2 BASE REFREIGERATION

The base CAD model has been redesigned to habilitate an opening in the wall next to the base fan. This modification ensures the air flow from the inside of the base to the outside in the point where the fan is most effective.

## 8.3 ROBOT STABILITY

This problem is much harder to solve. First of all, let's study the centre of mass of the robot.

### 8.3.1 CENTRE OF MASS STUDY

Let's study the initial centre of mass. In first place lets study the new centre of mass location when all plastic pieces have the same density (left) and then when we apply the new density distribution, in which the components dear the base have been printed with more infill density (right). Nevertheless the essential parts are printed solid.

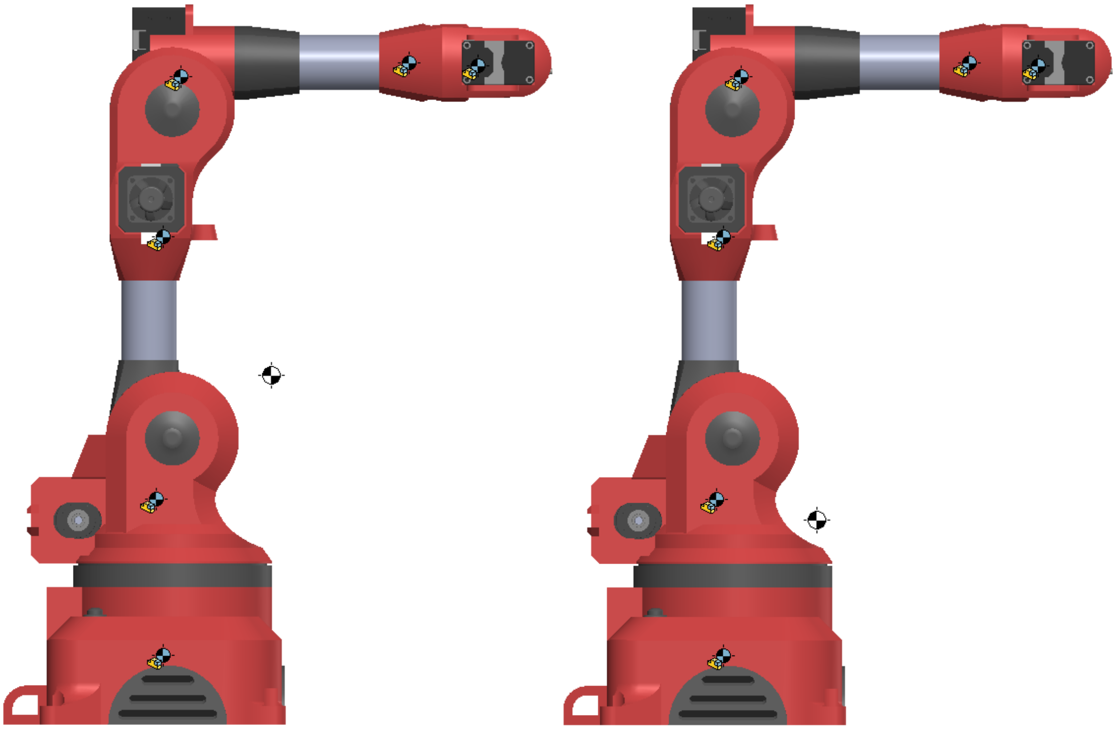


*Illustration 24: Centre of mass analysis*

Other measures to improve the stability is to double the wall base thickness, which makes the piece heavier and more resilient to charges and temperature.

The las measure to improve the robot stability is to design 4 moorings to affix the robot to a methacrylate plate as shown:

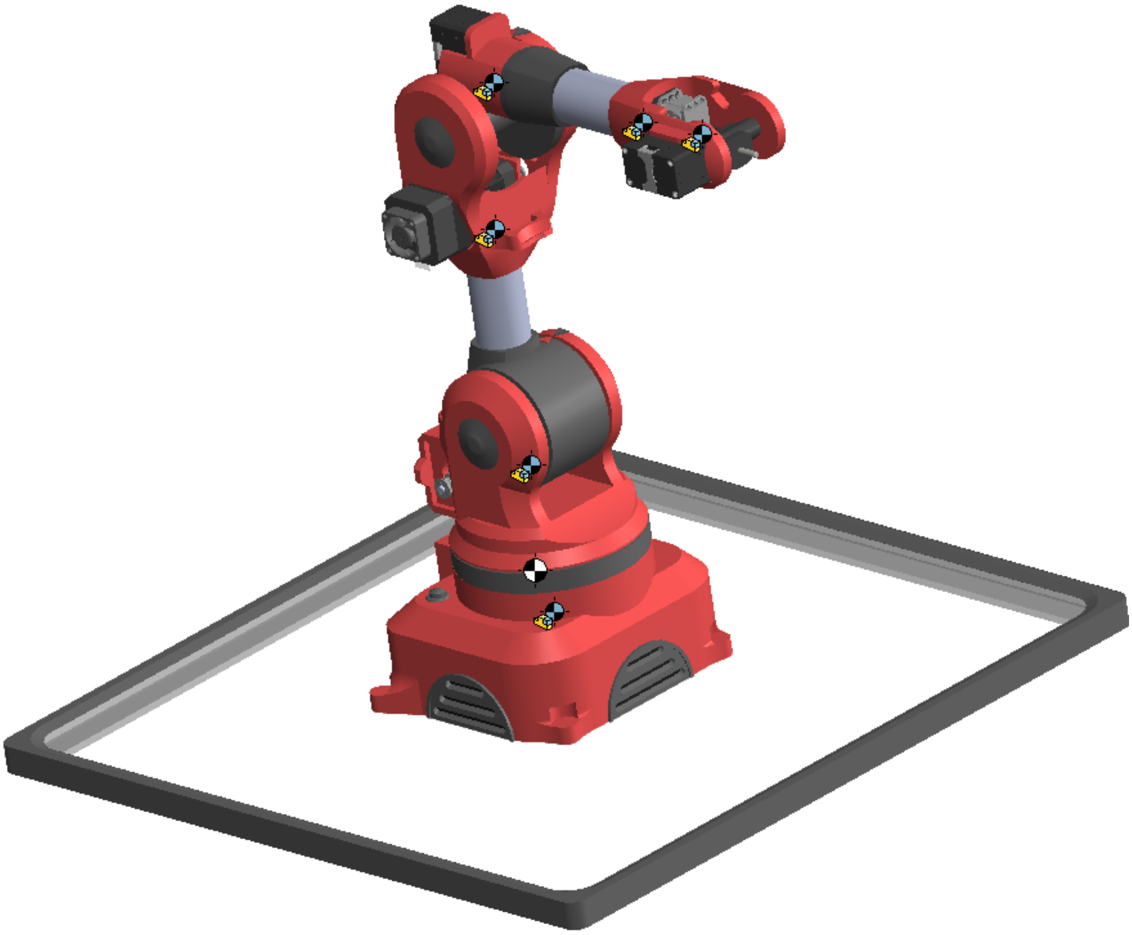


*Illustration 25: Base addon*

Once improved the design, we can start manufacturing the new robot.

According to the broken piece, the manufacturing correct configurations are exposed in the following paragraph.

# 9 MANUFACTURING

The robot was created with several components differentiated in three distinctive types: FDM (Fused Deposition Material) Plastic pieces, Actuators, electronics, screws and miscellaneous. In this paragraph the FDM manufacturing process[8] will be exposed. After the 3D modelling process, we proceed to export a file in "*.stl*" format to the slicing software; in this case, PRUSA Slicer. To print the different elements the Prusa MK3 3D printer has been used.

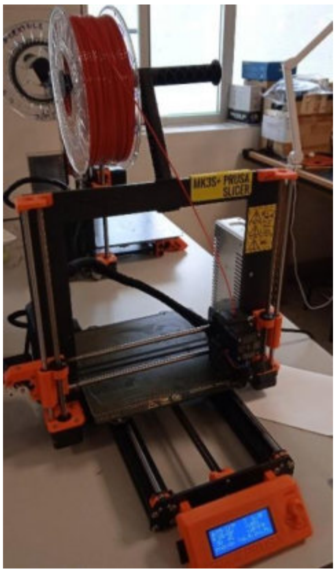


*Illustration 26: Prusa MK3*

It is not needed to go into details regarding the operation and handling of this device because it goes beyond the scope of this report, however, it is important to highlight that the different parameters of the device are essential to slice in an appropriate way and print the pieces. Obviously different devices and technologies can be used.

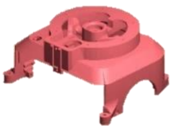
## 9.1 SLICING PROCESS

The slicing process consists in the transformation of the 3D model (the .stl previously generated file) into the necessary instructions to be sent to the 3D printer so that it interprets them and, as a cartesian robot, performs the relevant movements to manufacture the component; in our case, in PLA plastic by way of a process called FDM or Fused Deposition of Material. In the slicing program (Prusa Slicer), we must introduce a series of configurations referring to the device used and the finish surface quality of the component that we want to print. It is essential to choose wisely this key parameters:

| PARAMETER | DESCRIPTION |
|---|---|
| **LAYER HEIGHT** | Each layer thickness |
| **INFILL DENSITY** | Proportion of plastic and void in the inner side of the solid. From 0 (empty) to 100 (completely filled) |
| **FILL PATTERN** | Structure or drawing inside the solid |
| **SUPPORT MATERIAL** | Pillars and structures to support the possible cantilevers during the printing process |

*Table 9: Principal printing parameters description*

The criteria to choose the appropriate infill density is to put heavy pieces in the lowest part of the robot, near the base in order to facilitate the second and third joint (Seteppers) to rise up the rest of articulations. At the outermost part of the arm, we must install lightweight components. This is the reason why the last Dynamixels (Joint 6 and terminal effector) are smaller and lighter. Another criterion to choose the infill density is the number of predictable charges that a specific component will have to endure. If it is an essential part for the full robot functionality or will have to support lot of charges the infill density must be higher. There are much more parameters, but this are the crucial ones to manufacture a strong robot. In the next table we can see the accurate values for each parameter and piece.

| **Component** | **Parameter** | **Value** |
|---|---|---|
| Base | Layer height | 0,3 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Arm Top | Layer height | 0,2 mm |
|  | Fill density | 30% |
| | Fill pattern | Triangles |
| | Support material | Enabled |
| Belt tensioner | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Disabled |
| Air Routing | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Arm Bottom | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Elbow | Layer height | 0,2 mm |

| **Component** | **Parameter** | **Value** |
|---|---|---|
| Shoulder | Layer height | 0,2 mm |
|  | Fill density | 70% |
| | Fill pattern | Triangles |
| | Support material | Enabled |
| Tool Support | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Lines |
| | Support material | Enabled |
| Base bearing support | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Disabled |
| Shoulder Gearing | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Disabled |
| End Cap | Layer height | 0,2 mm |
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Forearm | Layer height | 0,2 mm |

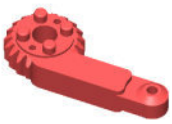
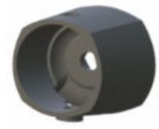
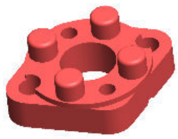
| Component | Parameter | Value |
|---|---|---|
|  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Forearm top  | Layer height | 0,2 mm |
| | Fill density | 30% |
| | Fill pattern | Triangles |
| | Support material | Enabled |
| Clamp  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Gripper 1 base  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Roc  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Gears  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| bottom  | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Wrist  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Drawer  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Clamp  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Rod  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |
| Rod  | Layer height | 0,2 mm |
| | Fill density | 100% |
| | Fill pattern | Linear |
| | Support material | Enabled |

*Table 10: Slicing components specifications*

With all these Printing settings also exist Filament settings and Printer settings. The first ones depend on the type of material we use to print and the surrounding conditions. For PLA we use this configuration:

| PARAMETER | DESCRIPTION | VALUE |
|---|---|---|
| **NOZZLE TEMPERATURE** | Temperature of the extruder Nozzle, the hole through which it comes out fused plastic | 220 ºC |
| **BASE TEMPERATURE** | Temperature of the printing bed on which the piece is printed | 60ºC |
| **DIAMETER** | Diameter of the plastic filament | 1,75 mm |
| **DENSITY** | Plastic Density | 1,24 g/cm$^3$ |
| **FAN SPEED** | Cooling layer fan speed, from 0 (no cooling) to 100 (max speed) | 100 |

*Table 11: Filament Settings*

The printer settings depend on the machine used for the FDM process. There is no need to focus on this because each machine specifies the appropriate settings or even give their own slicing profile or software.

## 9.1 3D PRINTING PROCESS

As already mentioned, in this process PLA plastic is melted and deposited in layers. To do this, the extruder that heats and melts the plastic is moved, with this movement and fused plastic ejection the nozzle draws with the fluid (at 220ºC) as if it were a pen; layer after layer until a three-dimensional object is obtained[9]. Drawn out plastic is introduced above and right in the nozzle, it is melted and printed with it in liquid state.

Regarding the extruder, it is shown below how the filament is introduced, propelled and melted (purple in the drawing).

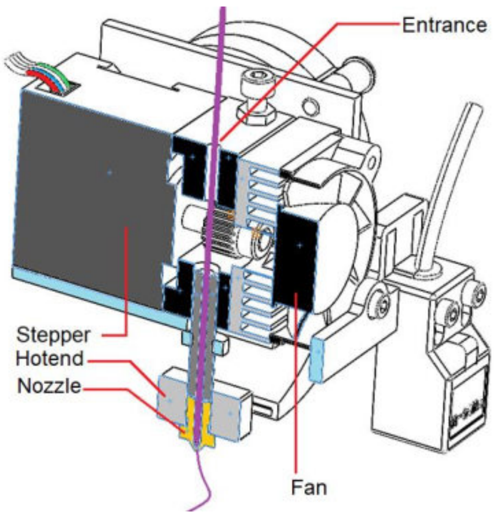



*Illustration 27: FDM Technology*

The entire extruder is anchored on a structure with mobile actuators that move it along a system of cartesian axes. A technology based on positioners (steppers) and belts are used that allow the extruder to be positioned in the cartesian coordinates on which the infrastructure of 3D printers are based. Here is what this technology can do:

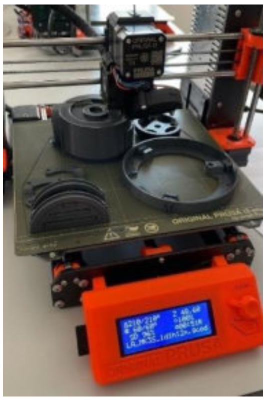


*Illustration 28: Print result*

## 9.2 PRINTS

In order to reduce the waste of material and the printing duration process it has been developed a series of 7 prints which took a total of 150 hours and 1.972 grams of plastic red and grey. In addition, during all these prints some replacements for important parts have been created.

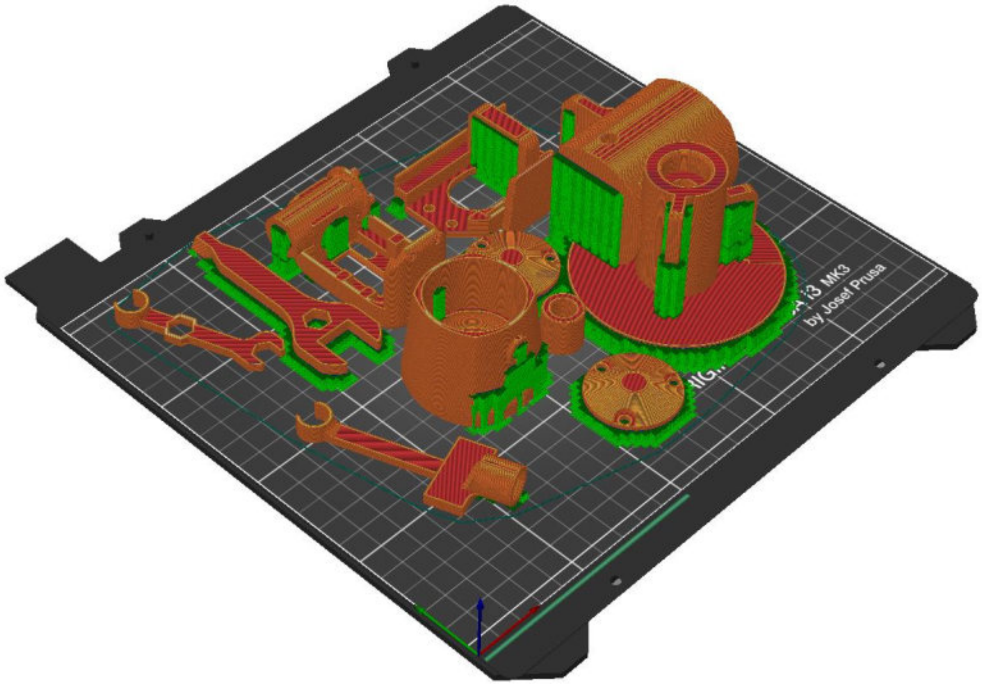


*Illustration 29: Printing process*

Two kilograms of plastic are needed to print the robot, but approximately the 12% of it is support material which must be removed before the robot assembly. In addition, there are some extra pieces for future replacements. The ***centre of mass study*** shown in the paragraph 10.3.1 DOES consider the variation of infill density of the different components and the consequent variation of mass and centre of mass dot coordinates.

### 9.2.1 ¿WHY PRUSA SLICER?

PrusaSlicer (formerly known as Slic3r Prusa Edition or Slic3r PE) is a slicer software based on the open-source project Slic3r. PrusaSlicer is an open-source, feature-rich, frequently updated tool which can export the perfect print files for (not only) Original Prusa 3D printer. The main feature for which this software was chosen was that is the recommended slicer for Prusa printers (the one has been used). Otherwise, this software allows to set specific printing properties for each part at the same impression process[10]. This feature allows us to configurate different infill density and pattern at different components in the same file. This facilitates the purpose to reduce timing, waste and resources in the printing process.

This software generates the "*.gcode*" file that the 3D printer controller interprets and executes. This file contains the information regarding the temperature and all the filament parameters described previously, as well as all the points through which the extruder nozzle must go through, depositing or not depositing plastic. Keep in mind that for the application case, the laminator generates a file in numerical control programming language (similar to CNC) with more than 400,000 instructions, which the printer will execute in real time. The slicer even makes all the relevant calculations and estimations of time, mass, meters of filament, etcetera.

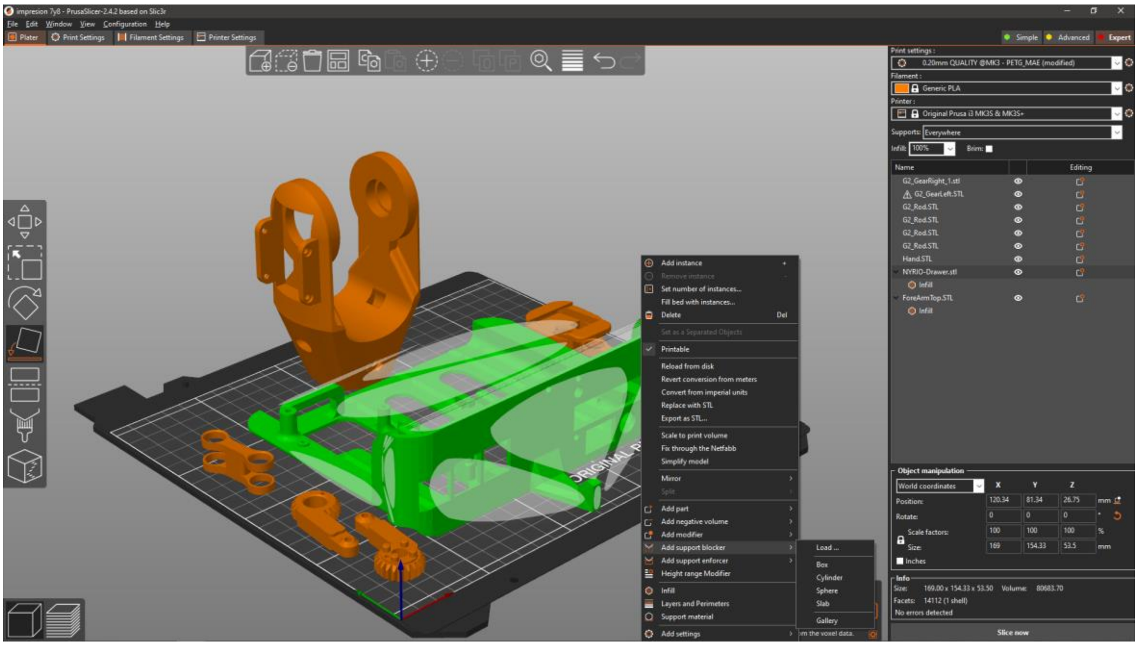


*Illustration 30: Prusa Slicer Software*

# 10 ASSEMBLY

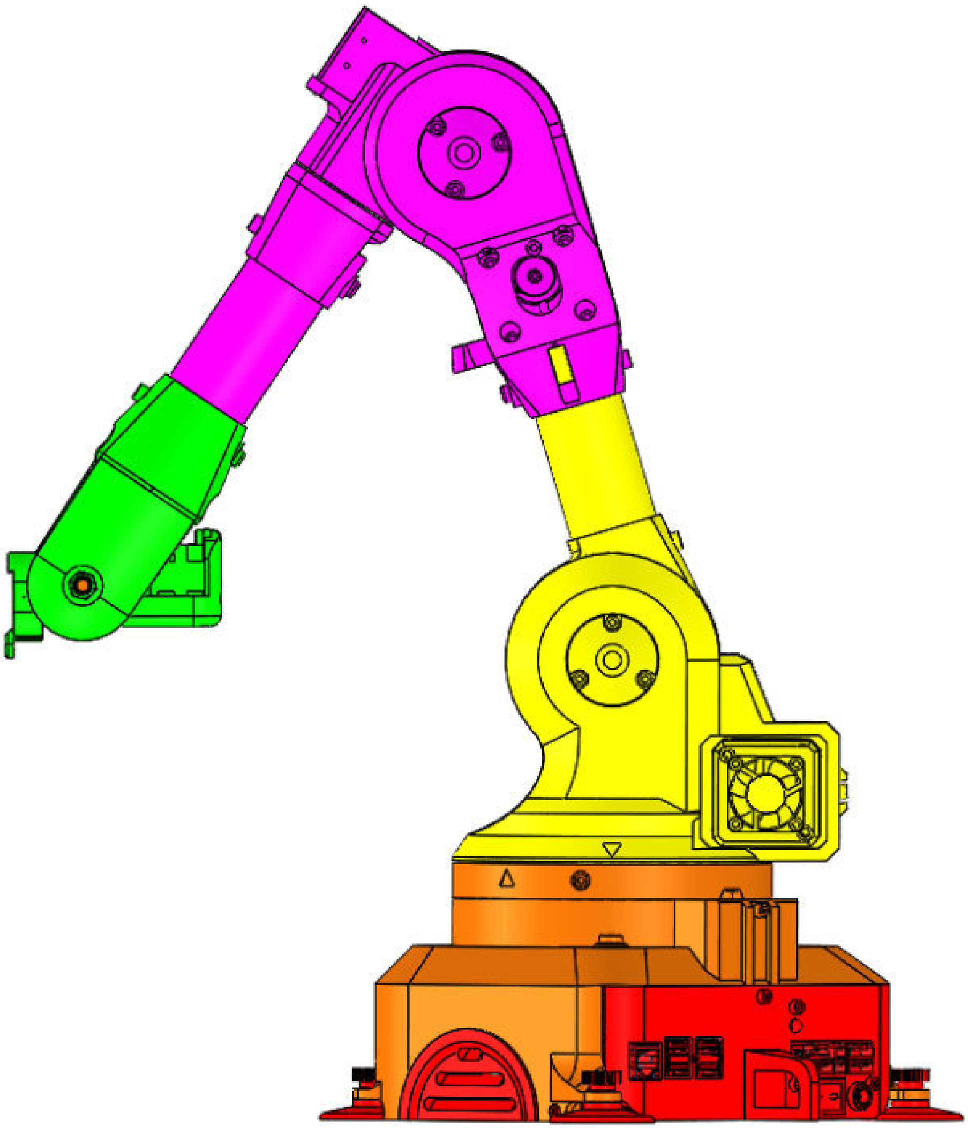


*Illustration 31 : Assembly Groups*

**BASE**

**JOINTS 1 & 2**

**JOINTS 3 & 4**

**JOINTS 5 & 6**

## 10.1 COMPONENTS

Here is shown every single component of the robot arranged by couples of joints.

### 10.1.1 BASE ASSEMBLY

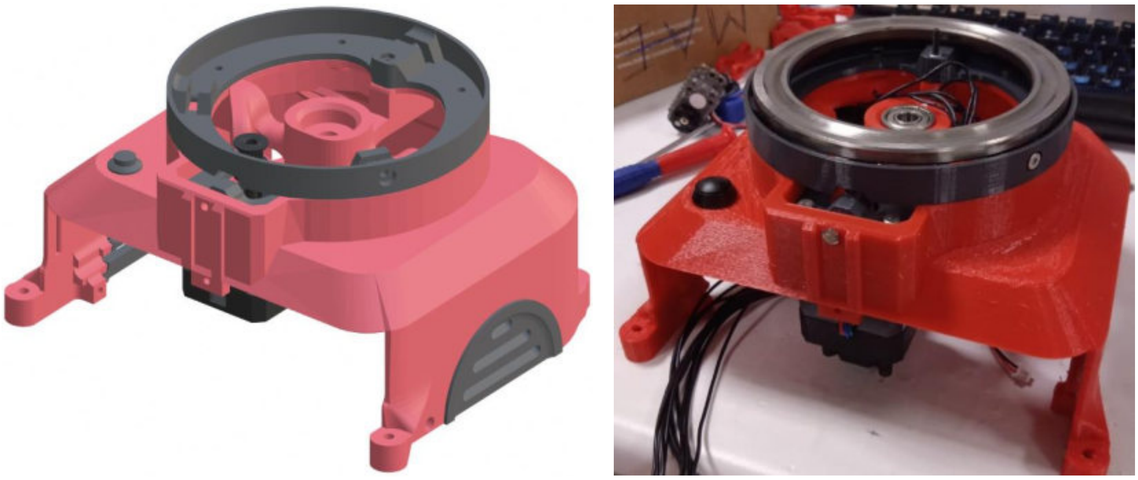
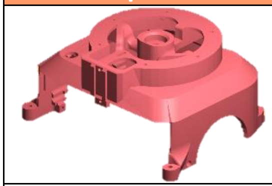


*Illustration 32: Base Assembly – no electronics*

These are de different components that shape the robot base.

| Component | Name | Count |
|---|---|---|
|  | Base | 1 |
|  | Belt tensioner | 1 |
|  | Base Bearing Support | 1 |
|  | Air Routing | 3 |
|  | NEMA17 STEPPER | 1 |
|  | Power button | 1 |

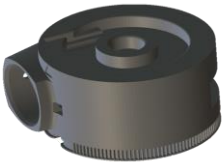
| Component | Name | Count |
|---|---|---|
|  | Fan | 1 |
|  | Bearing 8x22x7 | 1 |
|  | Bearing 3x10x4 | 3 |
|  | M3 Brake Nut | 2 |
|  | Screw M3 x 10 mm | 11 |
|  | Screw M3 x 16 mm | 8 |

*Table 12: Base assembly without electronics elements*

The assembly process is explained in the following video: https://youtu.be/3f-Ecz0IYLY

### 10.1.2 FIRST AND SECOND JOINT ASSEMBLY

This joint supports the rest of the articulations, also allows to rotate the robot in its base as we can see in the Illustration 7 and fold up the rest of the arm.

| Component | Name | Count |
|---|---|---|
|  | Shoulder | 1 |
|  | Shoulder Gearing | 1 |
|  | Arm Bottom | 1 |
|  | Belt tensioner | 1 |
|  | NEMA17 STEPPER | 1 |
|  | Half Bearing 90x120 mm | 1 |
|  | Belt | 2 |
|  | Thrust Bearing 8x19x7 | 1 |
|  | M3 Brake Nut | 1 |
|  | M4 Brake Nut | 1 |
|  | M8 Brake Nut | 1 |

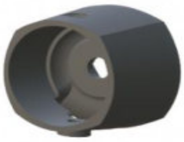
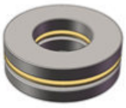
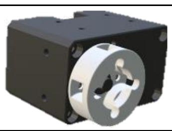
| Component | Name | Count |
|---|---|---|
|  *Illustration 33: Joints 1 & 2 Assembly* | | |
|  | End Cap | 2 |
|  | Bearing 8x22x7 | 1 |
|  | Fan | 1 |
|  | Aluminium tube | 1 |
|  | Spring | 1 |
|  | Screw M3 x 10 mm | 10 |
|  | Screw M3 x 16 mm | 6 |
|  | Screw M4 x 40 mm | 1 |
|  | Screw M8 x 55 mm | 1 |
|  | Screw M8 x 90 mm | 1 |

*Table 13: First & Second Joint elements*

The assembly process is explained in the following video: [https://youtu.be/f9BH28Sfflw](https://youtu.be/f9BH28Sfflw)

### 10.1.3 THIRD AND FOURTH JOINT ASSEMBLY

This pair of joints work like a human Elbow allowing to fold up and rotate the forearm.

| Component | Name | Count |
|---|---|---|
|  | Arm Top | 1 |
|  | Elbow | 1 |
|  | Forearm Bottom | 1 |
|  | Belt tensioner | 1 |
|  | NEMA17 STEPPER | 1 |
|  | Thrust Bearing 20x40x14 | 1 |
|  | Belt | 2 |
|  | Thrust Bearing 8x19x7 | 1 |
|  | M3 Brake Nut | 1 |
|  | M4 Brake Nut | 1 |
|  | M8 Brake Nut | 1 |
|  | Dynamixel XL 430 | 1 |

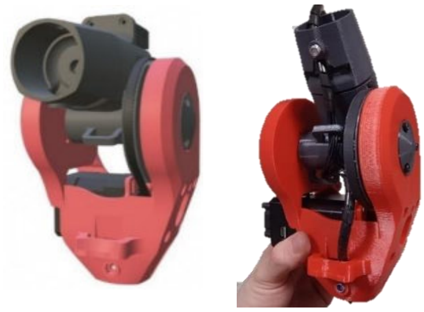
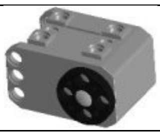
| Component | Name | Count |
|---|---|---|
|  *Illustration 34: Joints 3 & 4 Assembly* | | |
|  | End Cap | 2 |
|  | Bearing 8x22x7 | 2 |
|  | Fan | 1 |
|  | Aluminium tube | 1 |
|  | Bearing 3x10x4 | 2 |
|  | Screw M3 x 10 mm | 10 |
|  | Screw M3 x 16 mm | 2 |
|  | Screw M4 x 40 mm | 1 |
|  | Screw M8 x 40 mm | 1 |
|  | Screw M8 x 90 mm | 1 |

*Table 14: Third & Fourth Joint Elements*

The assembly process is explained in the following video: [https://youtu.be/r_z3i4-z8CU](https://youtu.be/r_z3i4-z8CU)

### 10.1.4 FIFTH AND SIXTH JOINT ASSEMBLY

This pair of joints solves the orientation problem of the robots. They work like a human wrist.

| Component | Name | Count |
|---|---|---|
|  | Forearm Top | 1 |
|  | Wrist | 1 |
|  | Hand Tool Support | 1 |
|  | Belt tensioner | 1 |
|  | Dynamixel XL 320 | 1 |
|  | Bearing 6x19x6 | 1 |
|  | M5 Washer | 1 |
|  | M5 Brake Nut | 2 |
|  | M6 Brake Nut | 2 |
|  | Screw M2 x 5 mm | 4 |

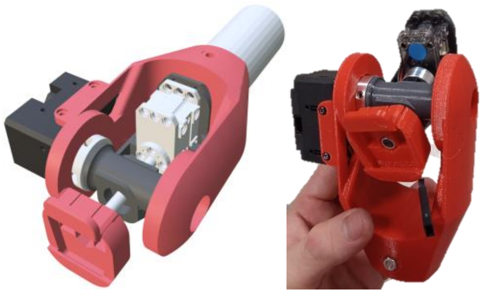
| Component | Name | Count |
|---|---|---|
|  *Illustration 35: Joints 5 & 6 Assembly* | | |
|  | Coupling Shaft | 2 |
|  | Bearing 5x14x5 | 1 |
|  | Dynamixel XL 430 | 1 |
|  | Aluminium tube | 1 |
|  | Bearing 3x10x4 | 2 |
|  | Screw M3 x 5 mm | 2 |
|  | Screw M5 x 40 mm | 1 |
|  | Screw M6 x 50 mm | 1 |

*Table 15: Fifth & Sixth Joint Elements*

The assembly process is explained in the following video: [https://youtu.be/w52qcxzaf_0](https://youtu.be/w52qcxzaf_0)

### 10.1.5 ASSEMBLING ELECTRONICS

Under the Robot there are ubicated the electronics. These electronics are formed by a Raspberry Pi 3, the Robot connectors panel and a specific HAT[1] over the Raspberry.

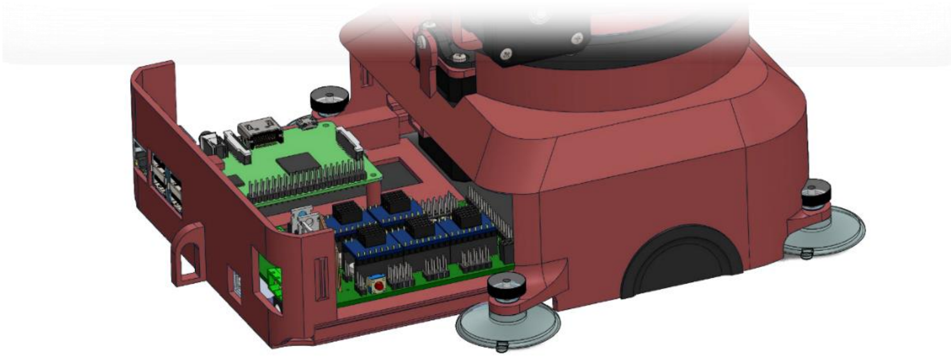
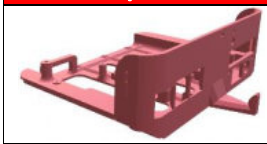
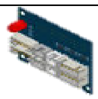


*Illustration 36: Elecronics Assembly*

| Component | Name | Count |
|---|---|---|
|  | Drawer | 1 |
|  | Ribbon cable | 2 |
|  | Switch | 1 |
|  | Connectors Panel | 1 |
|  | HAT | 1 |
|  | Screw M2 x 5 mm | 4 |
|  | Screw M3 x 10 mm | 5 |
|  | Screw M4 x 40 mm | 1 |

| Component | Name | Count |
|---|---|---|
|  | Raspberry Pi 3 | 1 |
|  | Label | 1 |
|  | DC Connector | 3 |
|  | M3 Brake Nut | 1 |
|  | M4 Brake Nut | 1 |
|  | Screw M3 x 5 mm | 2 |
|  | Screw M3 x 16 mm | 2 |

*Table 16: Electronics elements*

The assembly process is explained in the following video: [https://youtu.be/ym8SnfpQdpg](https://youtu.be/ym8SnfpQdpg)

[1] A "HAT" (Hardware Attached on Top), introduced along with the Raspberry Pi B+ in 2014, are expansion boards that connect to the Raspberry Pi's set of 40 GPIO pins and easily add functionality such as lights, motors, sensors and fans without a mess of wires. HATs are the nearest equivalent of Arduino Shields.

## 10.2 GRIPPER ONE

The first gripper uses two crank mechanisms to transform the rotary motion from the tool dynamixel into the open-and-close movement of the clamp. This is de CAD design and the equivalent mechanism figure drawing:

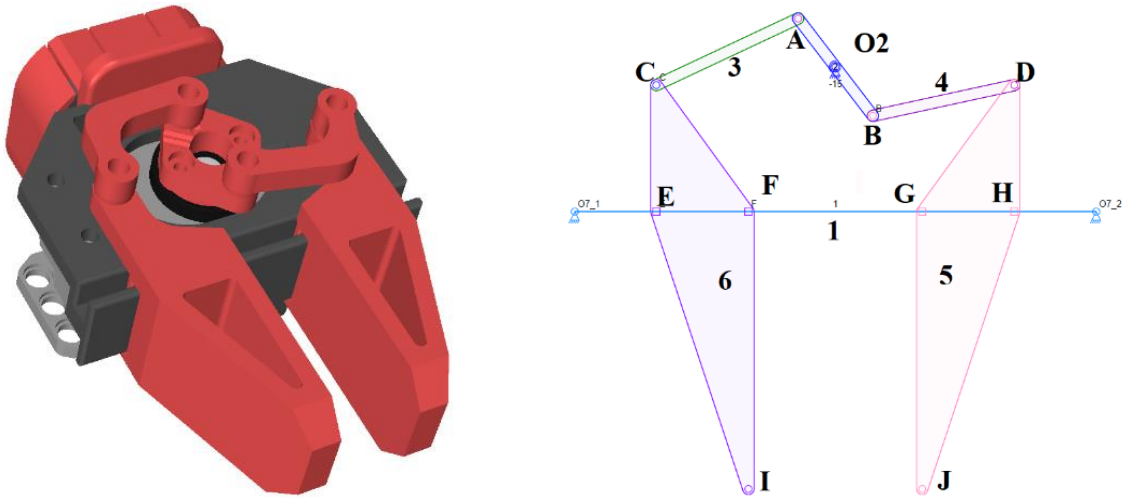



*Illustration 37: Gripper 1*

It is a Crank-Crank type mechanism. It is double; one for each side of the clamp. One (the left) is made up of links 1 (frame), 2, 3 and 6. The other (the right) is made up of links 1, 2, 4 and 5.

- Joints:
  - O2: System input. it is a ring attached to the engine. Rotate bar 2.
  - A, B, C, D: Revolving or rotating pair.
  - E, F, G, H: Prismatic pair; links 5 and 6 slide in the direction of bar 1 (frame).

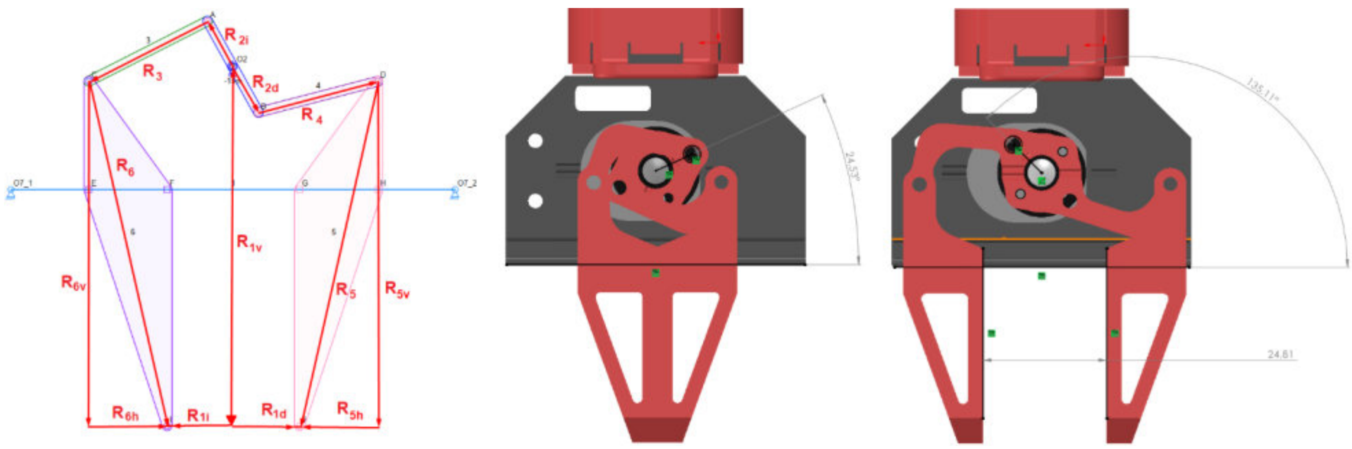



*Illustration 38: Gripper one - Matrix algebra*

Now we analyse the minimum and maximum position limits and the matrix algebra equations. All the calculations have been developed using Matlab® [11]. The code can be checked in the Appendix documents.

Data:

***$R_{1v}$ = 54,25; $R_{2d}$ = 8; $R_{2i}$ = $R_{2d}$; $R_{5h}$ = 13,1225; $R_{5v}$ = 52; $R_{6h}$ = $R_{5h}$; $R_{6v}$ = $R_{5v}$; $R_3$ = 20.43; $R_4$ = $R_3$***

Equations:

$$R_3 \sin(\theta_3) - R_{6v} + R_{2i} \sin(\theta_2) = -R_{1v} \qquad \text{[G1.1]}$$

$$R_{6h} + R_3 \cos(\theta_3) + R_{2i} \cos(\theta_2) = -R_{1i} \qquad \text{[G1.2]}$$

Clear $\theta_3$ as a function of $\theta_2$.

$$\theta_3 = \begin{pmatrix} -\mathrm{asin}\left(\frac{R_{1v} - R_{6v} + R_{2i}\sin(\theta_2)}{R_3}\right) \\ \pi + \mathrm{asin}\left(\frac{R_{1v} - R_{6v} + R_{2i}\sin(\theta_2)}{R_3}\right) \end{pmatrix} \qquad \text{[G1.3]}$$

Substitute the clear $\theta_3$ in the equation G1.2.

$$R_{6h} + R_{2i}\cos(\theta_2) \pm R_3 \sqrt{1 - \frac{(R_{1v} - R_{6v} + R_{2i}\sin(\theta_2))^2}{{R_3}^2}} = -R_{1i} \qquad \text{[G1.4]}$$

Finally solve the positive values of $R_{1i}$ in the equation G1.4. Must be considered that the opening distance of the clamp "***D***" is the twice measure of $R_{1i}$.

$$\mathrm{D} = R_3 \sqrt{1 - \frac{(R_{1v} - R_{6v} + R_{2i}\sin(\theta_2))^2}{{R_3}^2}} - R_{2i}\cos(\theta_2) - R_{6h} \qquad \text{[G1.5]}$$

Let's substitute the numerical values and simplify.

$$\mathrm{D} = \frac{2043\sqrt{1 - \left(\frac{800\sin(\theta_2)}{2043} + \frac{25}{227}\right)^2}}{100} - 8\cos(\theta_2) - \frac{5249}{400} \qquad \text{[G1.6]}$$

For better understanding of the calculus shown it is convenient to graph the relation between the opening distance of the clamp and the rotation angle of the motor.

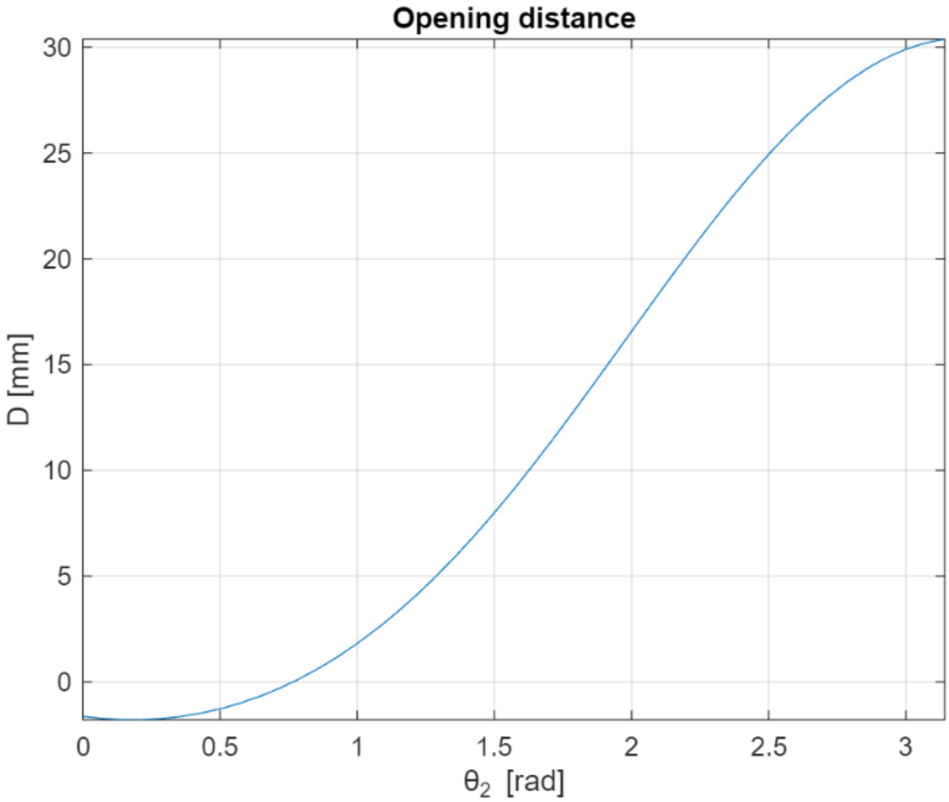



*Illustration 39: Relation betwen rotation and displacement*

### 10.2.1 ASSEMBLY GRIPPER 1

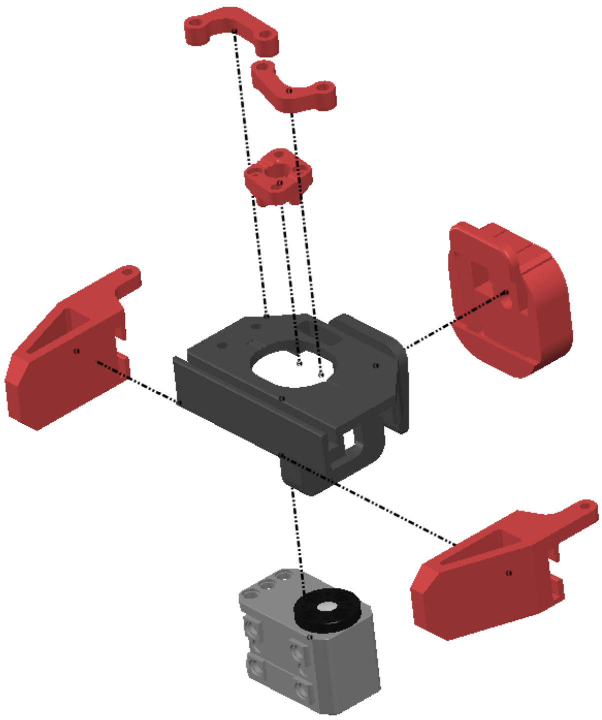
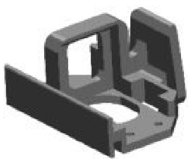


*Illustration 40: Gripper 1 Assembly*

| Component | Name | Count |
|---|---|---|
|  | Base | 1 |
|  | Clamp | 2 |
|  | Hand<br>Tool Support | 1 |
|  | Dynamixel<br>XL 320 | 1 |

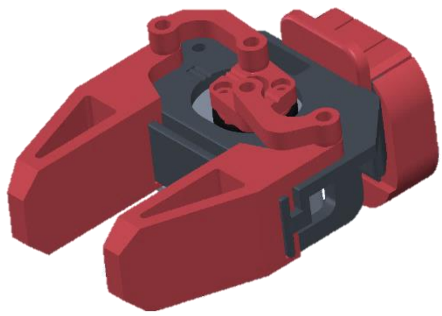
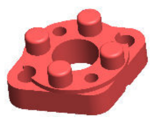
| Component | Name | Count |
|---|---|---|
| <br>*Illustration 41: Gripper 1* | | |
|  | rod | 2 |
|  | Rod | 2 |

*Table 17: Gripper 1 elements*

## 10.3 GRIPPER TWO

The second gripper is the installed gripper, and it is a double four bar mechanism.

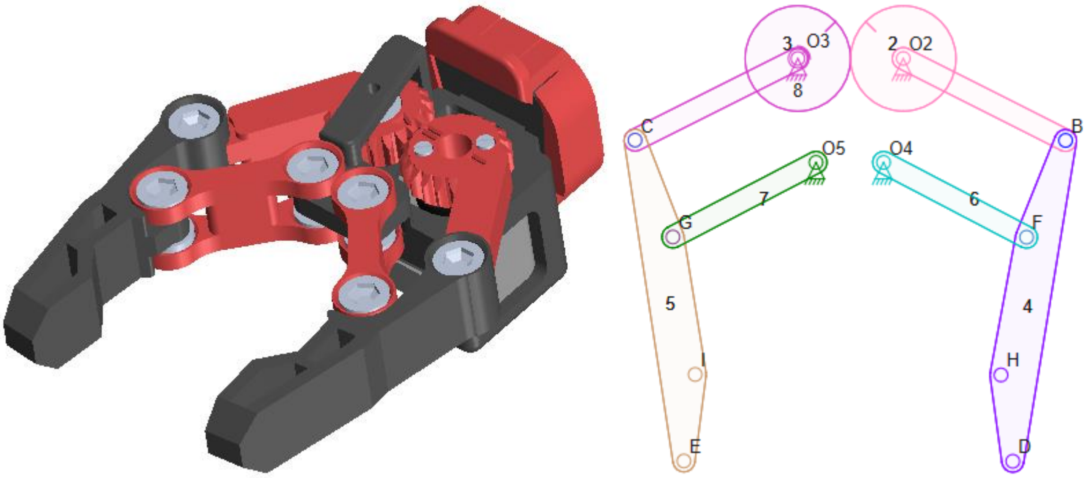



*Illustration 42: Gripper 2*

Basically, the servo rotates de second link (according to the following illustration) and the symmetrical link because they are both linked gears. When the link 2 spins clockwise the other does counterclockwise. This ensures the opposite movement at the other side of the clamp.

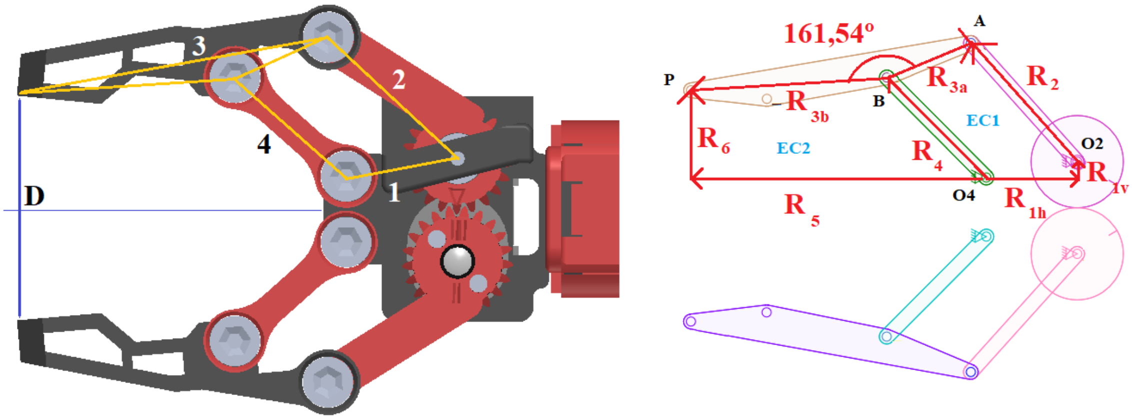



*Illustration 43: Gripper 2 - four bar mechanism*

If we analyse the upper four bar mechanism [12] and formulate the first closed-form equation, we get these expressions:

$$R_{1\mathrm{h}} + R_2 \cos(\theta_2) = R_4 \cos(\theta_4) + R_{3\mathrm{a}} \cos(\theta_3) \qquad [G2.1]$$

$$R_{1\mathrm{v}} + R_2 \sin(\theta_2) = R_4 \sin(\theta_4) + R_{3\mathrm{a}} \sin(\theta_3) \qquad [G2.2]$$

If we analyse the second closed-form equation, we get these expressions:

$$R_5 = R_2 \sin(\theta_2) + R_{3a} \sin(\theta_3) + R_{3b} \sin\left(\theta_3 - \frac{8077\,\pi}{9000}\right) \quad \text{[G2.3]}$$

$$R_6 = R_2 \cos(\theta_2) + R_{3a} \cos(\theta_3) + R_{3b} \cos\left(\theta_3 - \frac{8077\,\pi}{9000}\right) \quad \text{[G2.4]}$$

Operating we express $\theta_4$ in function of $\theta_2$ and $\theta_3$.

$$\theta_4 = R_{1v} + R_2 \sin(\theta_2) = R_{3a} \sin(\theta_3) + R_4 \sqrt{1 - \frac{(R_{1h} + R_2 \cos(\theta_2) - R_{3a}\cos(\theta_3))^2}{{R_4}^2}} \quad \text{[G2.5]}$$

The relation between the aperture movement (the variation of $R_6$) and $\theta_2$ will be:

$$\mathrm{R}_6 = R_4 - R_{1v} + R_{3b} \cos\left(\frac{923\,\pi}{9000} + \operatorname{acos}\left(\frac{R_{1v} - R_4 + R_2\cos(\theta_2)}{R_{3a}}\right)\right) \quad \text{[G2.6]}$$

The aperture of the clamp will be the twice of this number.

The relation between the advance movement of the clamp (the variation of $R_5$) and $\theta_2$ will be:

$$\mathrm{R}_5 = R_{3b} \sin\left(\frac{923\,\pi}{9000} + \operatorname{acos}\left(\frac{R_{1v} - R_4 + R_2\cos(\theta_2)}{R_{3a}}\right)\right) + R_2 \sin(\theta_2) - R_{3a}\sqrt{1 - \frac{(R_{1v} - R_4 + R_2\cos(\theta_2))^2}{{R_{3a}}^2}} \quad \text{[G2.7]}$$

This advance movement must be considered in the already studied complete cinematic of the robot.

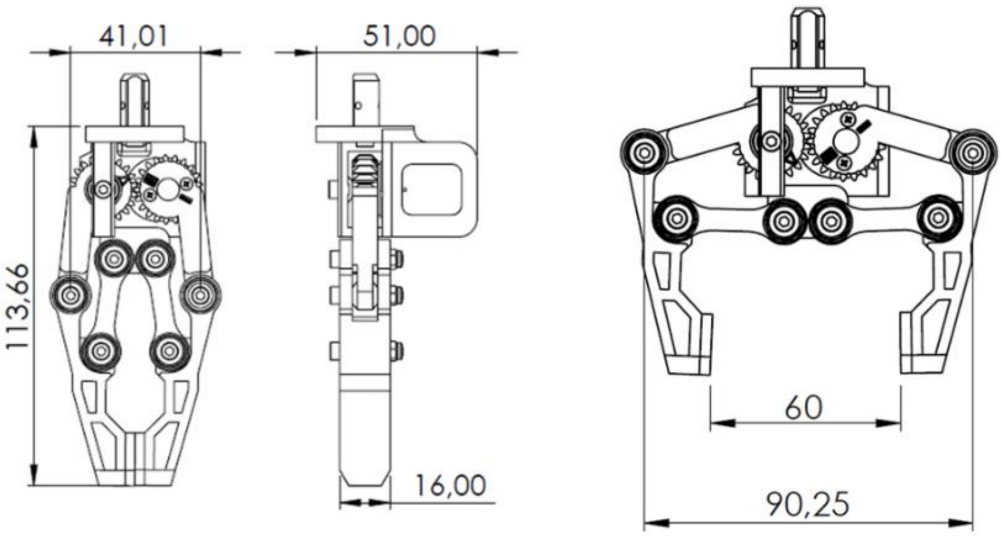



*Illustration 44: Mechanical Specification - Gripper 2*

Now we substitute the values with the real gripper specifications. From equation G2.8 to G2.11 are the equivalent equation from G2.1 to 4 respectively.

$$0 = 35\sin(\theta_2) + \frac{201\sin(\theta_3)}{10} - \frac{147\sin(\theta_4)}{5} + 22 \qquad \text{[G2.8]}$$

$$0 = 35\cos(\theta_2) + \frac{201\cos(\theta_3)}{10} - \frac{147\cos(\theta_4)}{5} + \frac{7}{2} \qquad \text{[G2.9]}$$

$$R_5 = 35\sin(\theta_2) + \frac{201\sin(\theta_3)}{10} + \frac{4321\sin\left(\theta_3 - \frac{8077\pi}{9000}\right)}{100} \qquad \text{[G2.10]}$$

$$R_6 = 35\cos(\theta_2) + \frac{201\cos(\theta_3)}{10} + \frac{4321\cos(\left(\theta_3 - \frac{8077\pi}{9000}\right))}{100} \qquad \text{[G2.11]}$$

The final relation between clamp aperture and $\theta_2$ is:

$$D = 2 \cdot \frac{4321\cos\left(\frac{8077\pi}{9000} - \operatorname{acos}\left(\frac{259}{201} - \frac{350\cos(\theta_2)}{201}\right)\right)}{100} + \frac{259}{5} \qquad \text{[G2.12]}$$

Here is a better representation of this relation:

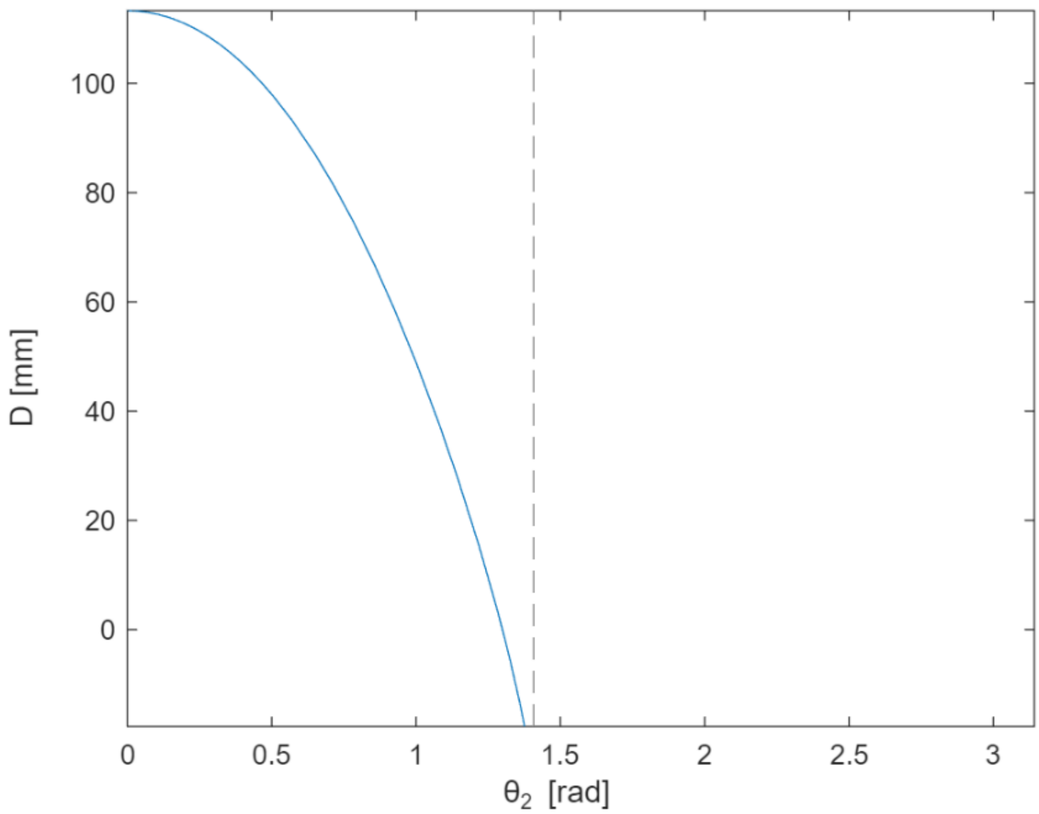



*Illustration 45: Gripper 2 aperture against dynamixel gyration*

The final relation between clamp advance movement when closing and $\theta_2$ is:

$$a = R_5 = 35\sin(\theta_2) - \frac{4321\sin\left(\frac{8077\,\pi}{9000} - \operatorname{acos}\left(\frac{259}{201} - \frac{350\cos(\theta_2)}{201}\right)\right)}{100} + \frac{201\sqrt{1-\left(\frac{350\cos(\theta_2)}{201} - \frac{259}{201}\right)^2}}{10} \quad \text{[G2.13]}$$

Here is a better representation of this relation:

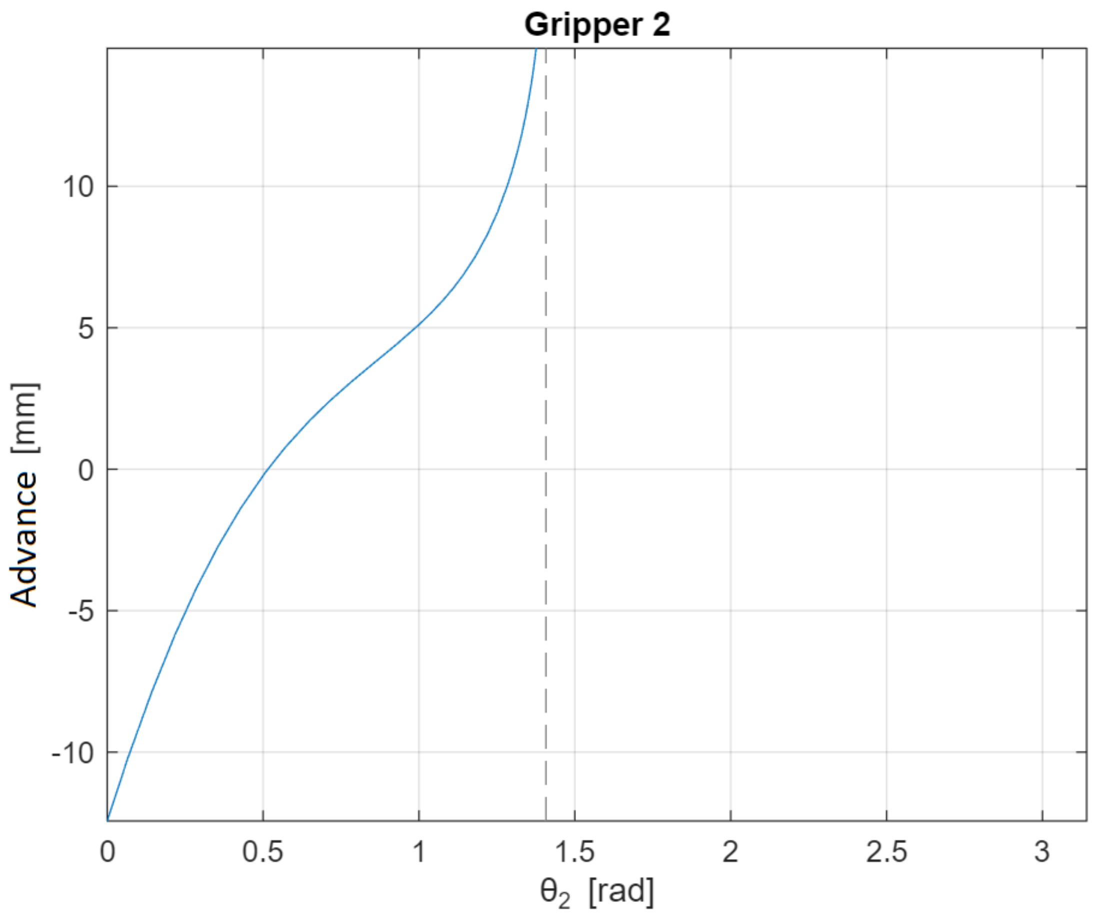



*Illustration 46: Gripper 2 advance against dynamixel gyration*

Both grips detailed drawings can be seen in the blueprints appendix

### 10.3.1 ASSEMBLY GRIPPER 2

This is the assembly scheme:

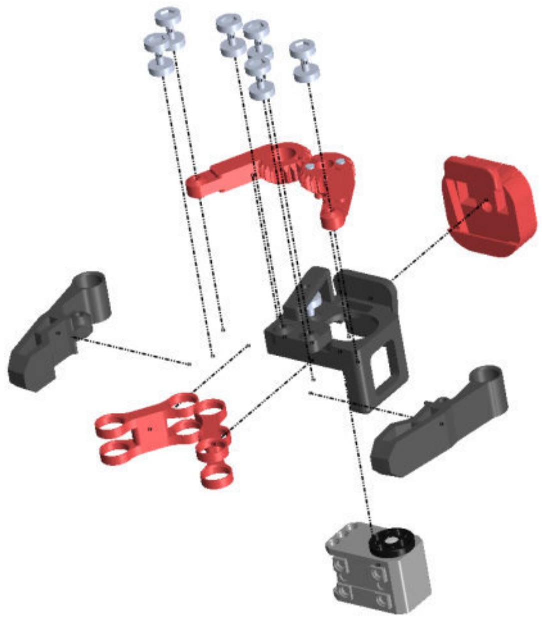


*Illustration 47: Gripper 2 assembly*

| Component | Name | Count |
|---|---|---|
|  | Base | 1 |
|  | Clamp | 2 |
|  | Hand Tool Support | 1 |
|  | Gear Right | 1 |
|  | Dynamixel XL 320 | 1 |

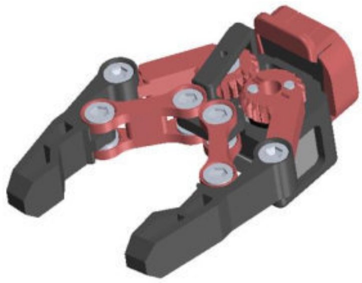
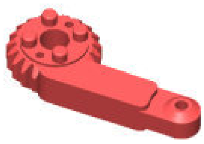
| Component | Name | Count |
|---|---|---|
|  *Illustration 48: Gripper 2* | | |
|  | Gear Left | 2 |
|  | Bearing 3x10x4 | 14 |
|  | Rod | 2 |

*Table 18: Gripper 2 elements*

In the assembly process it is essential to firmly fit and tighten every screw in order to avoid mechanical clearances. All the joints have one or multiple bearings. Excluding the fourth joint, the bearing and screw supports in he joints can be perfectly tighten because of the bearings. The Fourth Koint has a thrust bearing that is open and if the screws around it are installed too tight, the bearing will not spin freely. This is the joint consequently with the most lateral displacement. The rest of joints must be firmly mounted and have no play (excepting que rotation movement).

## 10.4 CONTROLLER - RASPBERRI PI 3B

The Raspberry Pi 3 Model B is the third generation of Raspberry Pi designs. This powerful small sized single board computer is often used for several applications. Raspberry Pi 3 Model B brings a powerful processer, 10x faster than the first-generation Raspberry Pi. Additionally, it includes ethernet, USB, wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs like this project.

This is the prime brain of the robot. Is the master in TTL communication with dynamixels and is the one which no stopping communicates with the stepper boards and closes the position control loop of the three first joints. It also reads the inputs panel of the robot. It is also in charge of the communication with de PC, managing several different communications at the same time.

## 10.5 CABLING

It is used wired connectivity between the controller and the actuators and sensors. There are two separated networks; one CAN bus that communicates the steppers with the Raspberry in a star topology network. Simultaneously, there is a TTL serial bus[13] topology network which connects the Raspberry (master) with the Dynamixel servomotors (slaves). All the Wires are inside the robot. In the last three joints the three wires of TTL bus connect the different Dynamixel connectors. All the cables connect with a single connector to the Niryo HAT over the raspberry. During the assembly process is essential to wire the robot carefully in order to avoid mechanical jams and wires rubbing. In the following illustration are shown all the wires inside the robot, from the connector to the HAT, to the last Joint.

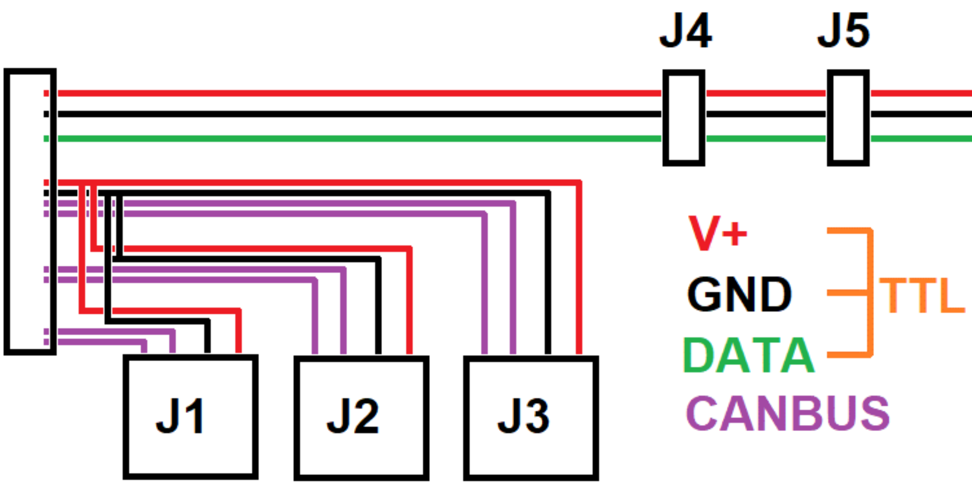



*Illustration 49: wiring diagrama*

# 11 ROS (*ROBOTICS OPERATING SYSTEM*)

The Robot Operating System (ROS) is framework that helps researchers and developers build and control different robotics applications. ROS is an open-source, meta-operating system that provides the services you would expect from an ordinary user operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. But the principal advantage is the reactivity and low latency in robot control. ROS is NOT a real-time operating system (RTOS). However, it is possible to integrate ROS with real-time code; what is very useful to make a robot do accurate movements in an enclosed period of time.

### 11.1.1 ROS – PC COMMUNICATION

There are several options to stablish the communication between ubuntu and the robot using Wiffi connection, Ethernet connection, Can Bus connection, etc. If we procede to use Microsoft Windows ® SO will be necessary the use a SCADA application called Niryo One Studio. In order avoid the excessive number of contents this multiple types of communication will not be further discussed.

# 12 ROS – ACTUATORS COMMUNICATION

The control architecture of the full robots consists of the continuous communication between the ROS and the Actuators which obey the ROS instructions and give to him the position of each joint.

## 12.1 DYNAMIXEL COMMUNICATION

It is used a distributed network between the ROS (master) and the Dynamixels (slaves) via TTL communication. TTL stands for Transistor-Transistor Logic, a serial communication commonly found in UART (universally asynchronous receiver/transmitter) transmission method, a method seen in most microcontrollers these days. Communication specification can me red in the second addendum.

### 12.1.1 TTL COMMUNICATION

To control the DYNAMIXEL actuators, the main controller needs to convert its UART signals to the half duplex type. The recommended circuit diagram for this is shown below.

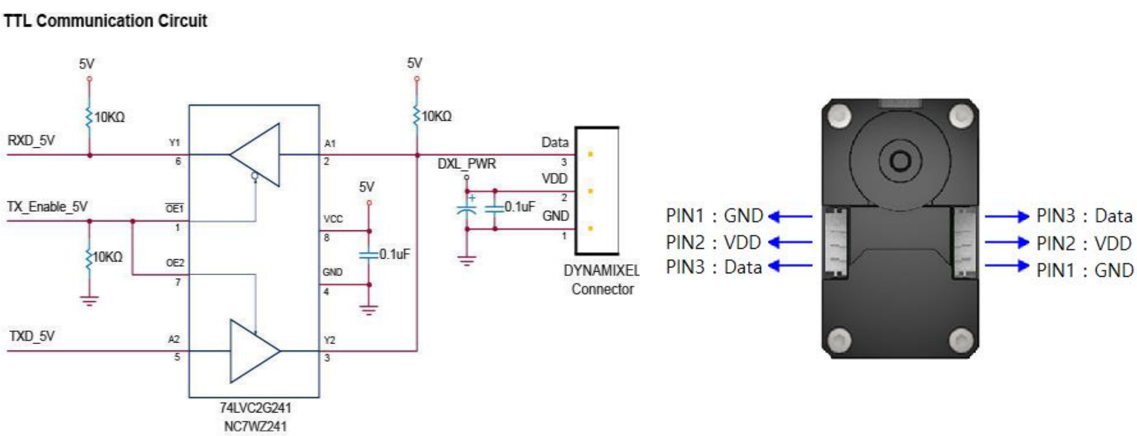



*Illustration 50: TTL Communication*

### 12.1.1 ID

The ID is a unique number in the TTL network to identify each DYNAMIXEL with an Instruction Packet. There can be used values around 0~253 (0xFD) as an ID, taking into account that 254(0xFE) is occupied as a broadcast ID. The Broadcast ID allows us to send an Instruction Packet to all connected DYNAMIXEL at the same time. “0” ID is no allowed. NEVER an identical ID can be used for multiple DYNAMIXEL in this kind of robots. This can only be done in systems where a same movement is carried by to motors (like vertical displacement in 3D printers). ROS will not allow you to use same ID in two network points and the user may face communication failure or may not be able to detect DYNAMIXEL at all. It is important to clarify that if the Instruction Packet ID is set to the Broadcast ID (0xFE), Status Packets will NOT be returned for READ or WRITE Instructions regardless of the set value of Status Return Level (68). ROS will not be able to know immediately if the broadcasted instruction arrived at all nodes nor if they executed the command correctly.

#### 12.1.2 BAUD RATE

The Baud Rate (instruction 8) determines serial communication speed between the ROS and DYNAMIXEL. Less than 3% of the baud rate error margin will affect to UART communication. The values to assign according to the different baud rate values are shown in appendix 2.3.1.

#### 12.1.3 RETURN DELAY TIME

When the DYNAMIXEL receives an Instruction Packet, it always returns the Status Packet after the time of the set Return Delay Time (instruction 9). Note that the range of admitted values is 0 to 254 (0XFE) and its unit is twice microseconds as expressed. For instance, if the Return Delay Time is set to '0X10', the Status Packet will be returned after 32 µsec. (16·2) when the Instruction Packet is received.

### 12.2 STEPPER COMMUNICATION

Stepper position controller boards use CAN bus (using MCP2515 to make an SPI-CAN interface). It is used the MCP2515 that allows you to make an SPI-CAN interface. The MCP2515 is implemented in the Niryo HAT. The board is programmed using Arduino IDE. The correspondent firmware can be downloaded in this link: https://github.com/NiryoRobotics/niryo_stepper/releases?_ga=2.242435328.326773273.1658014305-439348228.1655107917. There is need to install the Arduino IDE software, and install the "Arduino SAMD boards (32-bits ARM Cortex-M0+)" library from the boards manager (Tools -> Board -> Boards manager). In terms of communication, like in Dynamixel actuators is essential to make sure that the correct ID for each motor (1-2-3-4) on the Niryo Stepper code is written, before the uploading it to each board.

#### 12.2.1 CAN INTERFACE

CAN protocol is described by the international standard ISO 11898. The CAN protocol can be framed as a tool to solve a need within a specific period and market. It was developed by the Bosch brand in the 1980s for use in automotive industry, in need of a highly reliable bus network. This was widely accepted and only 10 years after its creation, the publication of its

ISO standard, establishing and standardizing said protocol. Its inclusion in commercial vehicles is common. The CAN protocol occupies the first and second level of the OSI model (physical layer and data link). The CAN bus is made up exclusively of a pair of twisted cables (with the possibility of shielding), calling one of them “CAN high” (H) and the other “CAN low” (L). In the transmission of data, they use 2 logic levels “0” and “1”. When sending a zero this is represented by a minimum difference of 1.5V between both wires. In our case we apply a difference of 2V to avoid wrong readings. For the representation of a logical “1”, both cables must present the same voltage, with a maximum difference of 1V. Being the signal transmitted through a differential pair, within a twisted cable, this makes the communication hardware media resilient against electromagnetic noise.

As in is common in other network protocols, due to the capacitances of the cables the transfer rate is limited by the distance. In this case, we are working in very short distances, and it is not a problem. However, and in the interest of full disclosure, the maximum speed is 1Ms for lengths less than 40 m and the minimum of only 50 kbps over distances of up to 1000 m.

### 12.2.2 DATA TRANSFER

The data transfer within the bus is structured in the form of packages, each with its own ID (identifier, just like with Dynamixels) for its distinction and the determination of priorities. Data transfer frames can be structured with respect to to two types already predefined, type "base" or "A" and type "extended" or type "B". Both formats are very similar to each other, with the main difference is that this second (B) presents an ID with a higher number of bits, 29 (0x1FFF FFFF); against the 11 (0x7FF) offered by the "base". In our case we have chosen the use of the “extended” model to have greater flexibility when it comes to determine or change package IDs. On the other hand, both frames can pack the same amount of data per packet from 0 to 64 bits. Likewise, the bit frame includes elements common to other protocols such as the start and end bits of frame, the inclusion of a CRC code for verification of correct reception.

# 13 NYRIO SOFTWARE

Niryo One Studio is a software (Windows - MacOS - Linux) aimed to control Niryo One. Some of its functionalities are the setup of the robot, calibration, change some settings, move the robot, program a sequence in different languages, or debugging.

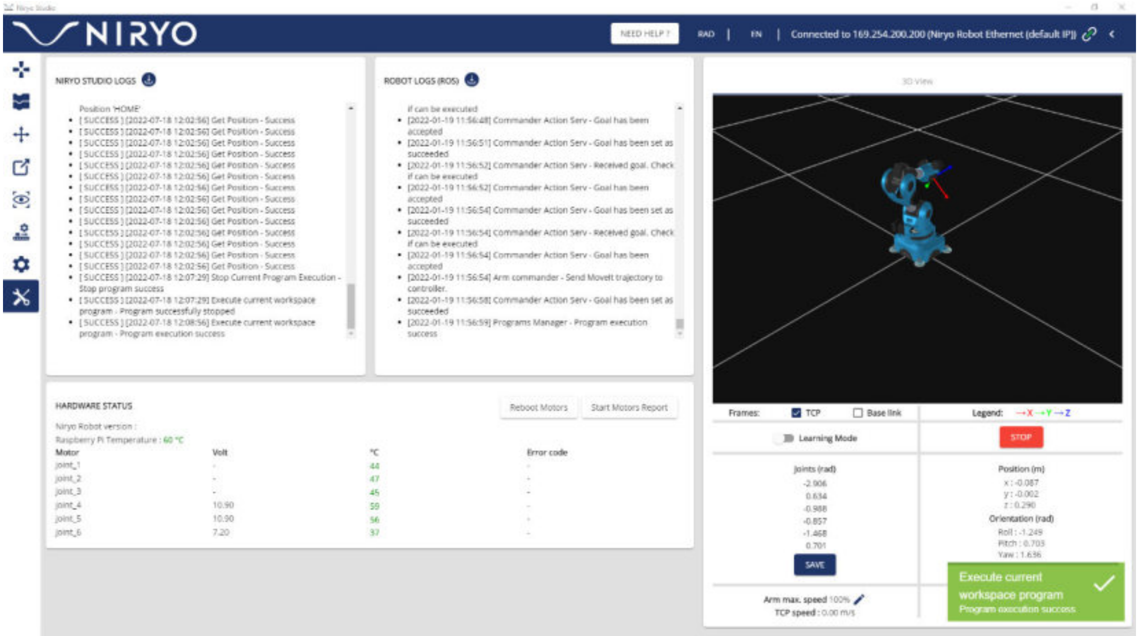


*Illustration 51: Niryo One Studyo interface*

It is basically a SCADA system which monitors the robot state in “real time” (not real – real time; as it’s been said, ROS is NOT a real time operating system) and allows the user to control the whole system with some graphic interfaces.

# 14 PROGRAM MOVEMENT SECUENCE USING PYTHON

With the robot ready to work, we can proceed to program a sequence of movements. In this paragraph will be shown a little example of complete and effective movement with the robot. The sequence will be programmed using Python, a high-level interpreted language which is rising in popularity again. To do this Niryo Studio has habilitated a space to write and execute Python scripts (The software itself was created using this language) [14].

## 14.1. SEQUENCE DESCRIPTION

- Set speed to the 30% of the maximum possible value
- Calibrate the robot autonomously
- Set new arm max speed
- Open gripper with torque at 100% and old torque at 20%
- Wait one second so the system stabilizes
- Change led colour to green
- Repeat three times:
  - Go to point
  - Wait one second
  - Close gripper at half speed gripper with torque at 100% and old torque at 20%
  - Go to point
  - Open gripper at full speed with torque at 100% and old torque at 20%
  - Wait two seconds
- Set TCP (Tool Centre Point) to a determine coordinates (position and rotation)
- Change led colour to purple
- Interpolate a SPLINE trajectory using three points
- Change colour to Red
- End

## 14.2 PYTHON CODIFICATION

```
# !/usr/bin/env python

from niryo_robot_python_ros_wrapper.ros_wrapper import *
import sys
import rospy

rospy.init_node('niryo_blockly_interpreted_code')
n = NiryoRosWrapper()

n.calibrate_auto()

try:
   n.set_arm_max_velocity(30)
   n.request_new_calibration()
   n.calibrate_auto()
   n.set_arm_max_velocity(30)
   n.open_gripper(500, 100, 20)
   n.wait(1)
   n.led_ring.solid([51,255,51,0], wait=True)
   for count in range(3):
     n.move_joints(*[(-2.842), 0.634, (-0.989), (-0.891), (-1.477),
0.701])
     n.wait(1)
     n.close_gripper(500, 100, 50)
     n.move_joints(*[(-1.257), 0.309, 0.793, 0.005, (-0.006), (-0.005)])
     n.open_gripper(500, 100, 20)
     n.wait(2)
   n.set_tcp(*[-0.007, 0.246, 0.02, -1.46, 1.421, 1.632])
   n.led_ring.solid([204,51,204,0], wait=True)
   n.execute_trajectory_from_poses_and_joints([[1.584, (-0.7), 0.027, (-
0.145), 0.038, (-1.375)], [1.495, (-0.143), 0.692, (-0.145), 0.051, (-
1.396)], [(-0.999), (-0.841), 0.011, 1.554, 1.578, 1.58]],
["joint","joint","joint"], 0)
   n.led_ring.solid([255,0,0,0], wait=True)

except NiryoRosWrapperException as e:
   sys.stderr.write(str(e))
```

### 14.3 TRAJECTORY CONTROL PYTHON EXAMPLE

```
# !/usr/bin/env python
from niryo_robot_python_ros_wrapper.ros_wrapper import *
import sys
import rospy

rospy.init_node('niryo_blockly_interpreted_code')
n = NiryoRosWrapper()
n.calibrate_auto()

try:
   n.move_joints(*[0, 0.01, 0.01, 0.02, (-0.006), (-0.005)])
   n.set_arm_max_velocity(40)
   n.move_joints(*[(-0.496), 0.493, (-0.124), (-0.865), 1.091, (-1.089)])
   n.move_joints(*[0.001, 0.5, (-1.25), (-0.012), 0.006, (-0.036)])
   n.execute_trajectory_from_poses_and_joints([[0.253, 0.219, 0.173, 2.312,
-0.223, -0.302], [-0.144, 0.278, 0.398, -0.485, 0.008, 2.051], [-0.177,
0.048, 0.533, 0.531, 0.195, -2.238]], ["pose","pose","pose"], 0.2)
   n.set_arm_max_velocity(100)
   for count in range(10):

     try:
         n.move_linear_pose(*[0.16, -0.292, 0.176, 2.326, -0.017, -2.562])
     except NiryoRosWrapperException as e:
         if 'cannot be reached with a linear trajectory' in str(e):
             n.move_pose(*[0.16, -0.292, 0.176, 2.326, -0.017, -2.562])

     try:
         n.move_linear_pose(*[0.234, 0.004, 0.516, 0.59, -0.072, -1.343])
     except NiryoRosWrapperException as e:
         if 'cannot be reached with a linear trajectory' in str(e):
             n.move_pose(*[0.234, 0.004, 0.516, 0.59, -0.072, -1.343])

     try:
         n.move_linear_pose(*[0.018, 0.225, 0.541, -1.556, -0.598, 1.504])
     except NiryoRosWrapperException as e:
         if 'cannot be reached with a linear trajectory' in str(e):
             n.move_pose(*[0.018, 0.225, 0.541, -1.556, -0.598, 1.504])

     try:
         n.move_linear_pose(*[-0.004, 0.343, 0.211, -1.517, 0.635, 1.591])
     except NiryoRosWrapperException as e:
         if 'cannot be reached with a linear trajectory' in str(e):
             n.move_pose(*[-0.004, 0.343, 0.211, -1.517, 0.635, 1.591])
     n.move_joints(*[(-2.842), 0.634, (-0.989), (-0.891), (-1.477), 0.701])

except NiryoRosWrapperException as e:
   sys.stderr.write(str(e))
```

# 15 PEDAGOGIC SCOPE

This dissertation is intended to have a didactic extent. This project has a Pedagogy of Learning (POL) component in which encompasses research and innovations pertaining to the education of students in various areas of education planification, mechanics design, industrial communications, control theory and many other aspects related to the betterment of the method and scientific techniques.

## 15.1 SECONDARY SCHOOL

This robot can be a powerful to seed scientific curiosity in the youngsters' minds. Without diving in the complex functionality of the robot, students can witness a real robotics application and interact with a real device. Furthermore, important concepts about kinematics and programming basics can be taught using this learning tool.

## 15.2 UNIVERSITY

Being a freeware project, students who want to understand the ins and outs of manipulators (like me) can dive as much as they want and even make modifications and suggest new applications. This project opens the door to advanced programming, mechanical design, communication networks, artificial vision, Industrial automation, additive manufacturing technologies, modern control engineering, PID design, digital systems and engineering itself.

# 16 CONCLUSIONS

In conclusion, the advancement in this project, understanding the whole aspects of a real manipulator robot, from mechanical design to complex communications networks and advanced control theory, while making decisions and modifications to improve a real commercial device. In addition, if we see the budget, we can conclude that we have reached an important prize lowering from the more tan two thousand euros of the original kit provided by Niryo (which is not longer available); to a materials cost under seven hundred euros. This has been achieved thanks to the 3D printing. This price makes the device more accessible.

After several months developing this project, the prototype device has been tested during numerous experiments and has suffered significant modifications. The results of the testing showed that de device is not prepared to real industrial applications. Despite that is a powerful learning tool.

Overall, the objectives of the project have been met and the robot is capable to do accurate fluid movements, and the project report clarifies all the procedures followed and facilitates the understanding of all the robot functionalities.

For future recommendations, the robot should be upgraded to industrial applications and artificial vision in the future. Apart from that, the robot movement speed should be increased. The system should also be linked or connected to complex automation process with strokes of decision making and neural networks.

# 17 BIBLIOGRAPHY


[1] A. Grozin, *Introduction to Mathematica® for Physicists*. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-00894-3.

[2] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, vol. 124. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-01851-5.

[3] N. Ida, *Engineering Electromagnetics*. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-07806-9.

[4] S. Krenk and J. Høgsberg, *Statics and Mechanics of Structures*. Dordrecht: Springer Netherlands, 2013. doi: 10.1007/978-94-007-6113-1.

[5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*, vol. 16, no. 4. London: Springer London, 2009. doi: 10.1007/978-1-84628-642-1.

[6] B. Kuo *et al.*, *PID avanzado*, vol. 23, no. 4. 2007. [Online]. Available: https://www.academia.edu/33825646/Control_PID_avanzado_-_1ed%0Awebdelprofesor.ula.ve/ingenieria/djean/index.../Teoria_Control.pdf%0Ahttp://www.slideshare.net/accelerate786/process-systems-analysis-and-control-third-edition

[7] H. Marsh, "Modern compressible flow," *International Journal of Heat and Fluid Flow*, vol. 4, no. 1, pp. 59–60, 1983, doi: 10.1016/0142-727x(83)90029-2.

[8] B. M. Rodríguez, "Estudio Sobre La Optimización De Los Parámetros De Fabricación En Una Impresora 3D Con Tecnología Fdm," 2017.

[9] 3Dsystems, "3D printer buyer's guide for professional and production applications [Brochure]," pp. 1–19, 2014, [Online]. Available: www.3dsytems.com

[10] M. Dumas, M. la Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*, vol. 64, no. 1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: 10.1007/978-3-662-56509-4.

[11] L. Keviczky, R. Bars, J. Hetthéssy, and C. Bányász, *Control engineering: MATLAB exercises*. 2019. doi: 10.1007/978-981-10-8321-1.

[12] D. Gross, W. Hauger, J. Schröder, W. Wall, and J. Bonet, *Engineering Mechanics 2*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-12886-8.

[13] "PLC Communications | PLC Manual." http://www.plcmanual.com/plc-communications (accessed Jan. 18, 2021).

[14] K. D. Lee, *Python Programming Fundamentals*, vol. 48, no. 09. London: Springer London, 2014. doi: 10.1007/978-1-4471-6642-9.

UNIVERSIDAD DE LA RIOJA

*Escuela Técnica Superior de Ingeniería Industrial*

*Grado en Ingeniería Electrónica Industrial y Automática*

# LOW-COST DIDACTIC ROBOT

## ANNEXES

Autor: Miguel Ángel Ezquerro Ezquerro

Tutor: Carlos Elvira Izurrategui

Date: 18 July 2022

# INDEX

# LIST OF FIGURES



*All illustrations have been created exclusively for this paper*.

# List of Abbreviations

| | |
|---|---|
| 6-DOF | Six-degree-of-freedom |
| EMC | Electromagnetic Compatibility |
| FDM | Fused Deposition Material |
| HAT | Hardware Attached on Top |
| HTM | Homogeneous Transformation Matrices |
| IS | International Standard |
| MCU | Micro Controller Unit |
| PWM | Pulse Width Modulation |
| ROS | Robotics Operating System |
| RS | Reference System |
| RTOS | Real Time Operating System |
| spp. | Steps per second |
| TTL | Transistor-Transistor Logic |

# 1 ANNEX I – STEPPER NEMA 17

*Illustration 1: Nema 17 Stepper*

## 1.1 COMPONENTS

For the correct understanding of the operation of the device, it will be necessary to have a very clear idea of how it is built and what parts and components make it up. A general breakdown of all the elements that intervene in its assembly and therefore in its operation is therefore presented.

### 1.1.1 STATOR

This is where the winding is to be excited sequentially to achieve the desired rotation and positioning. It consists of 2 phases (red and green) wound on a protective housing made up of two symmetrical halves, which not only precisely locate the winding to achieve the desired flux and field directions, but also provide support and prevent short circuits.

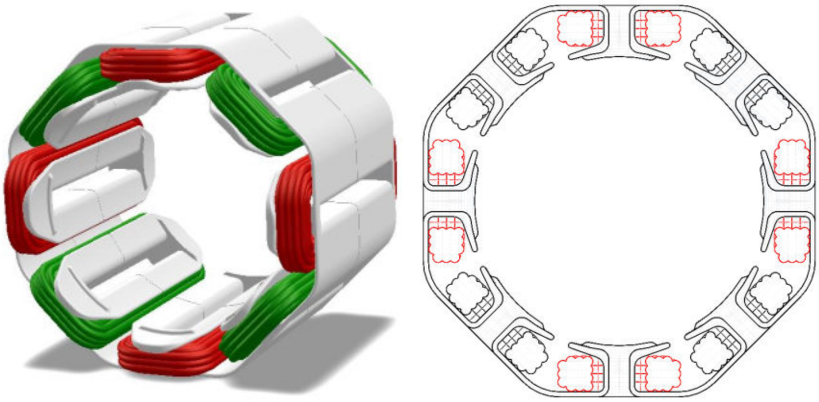


*Illustration 2: Nema 17 Winding*

Accompanying the winding, we have a protective plastic protector and, on the inside, arcs with slits.

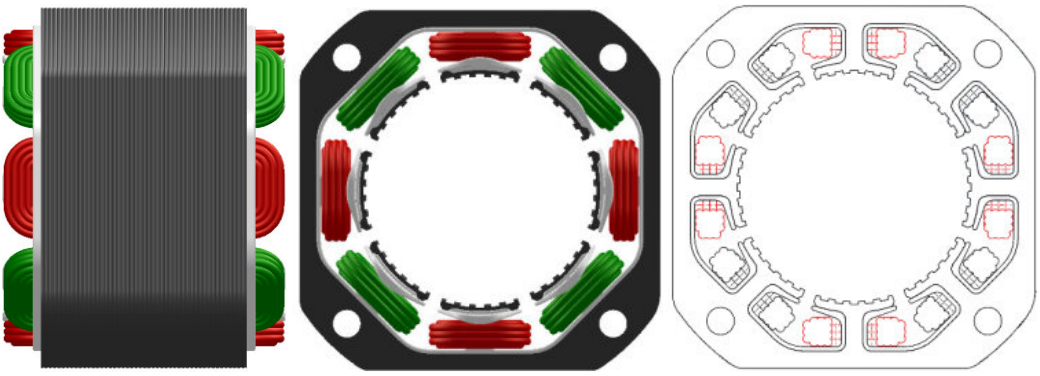


*Illustration 3: Nema 17 frame*

### 1.1.2 ROTOR

The rotor is made up of a magnet with several serrated steel plugs. Two of the steel plugs have north polarity and the other two, south. These discs consist of 50 teeth.

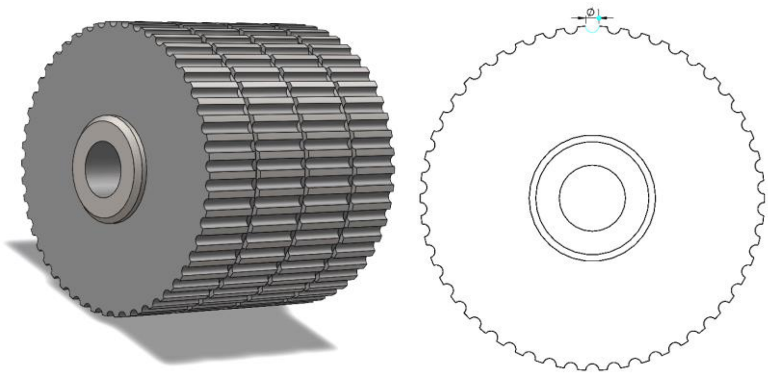


*Illustration 4: Nema 17 Rotor*

Attached to the rotor, there is a shaft with a key which will allow us to couple to the rotation of the rotor the GT2 Gear to move the belt.

## 1.2 ASSEMBLY

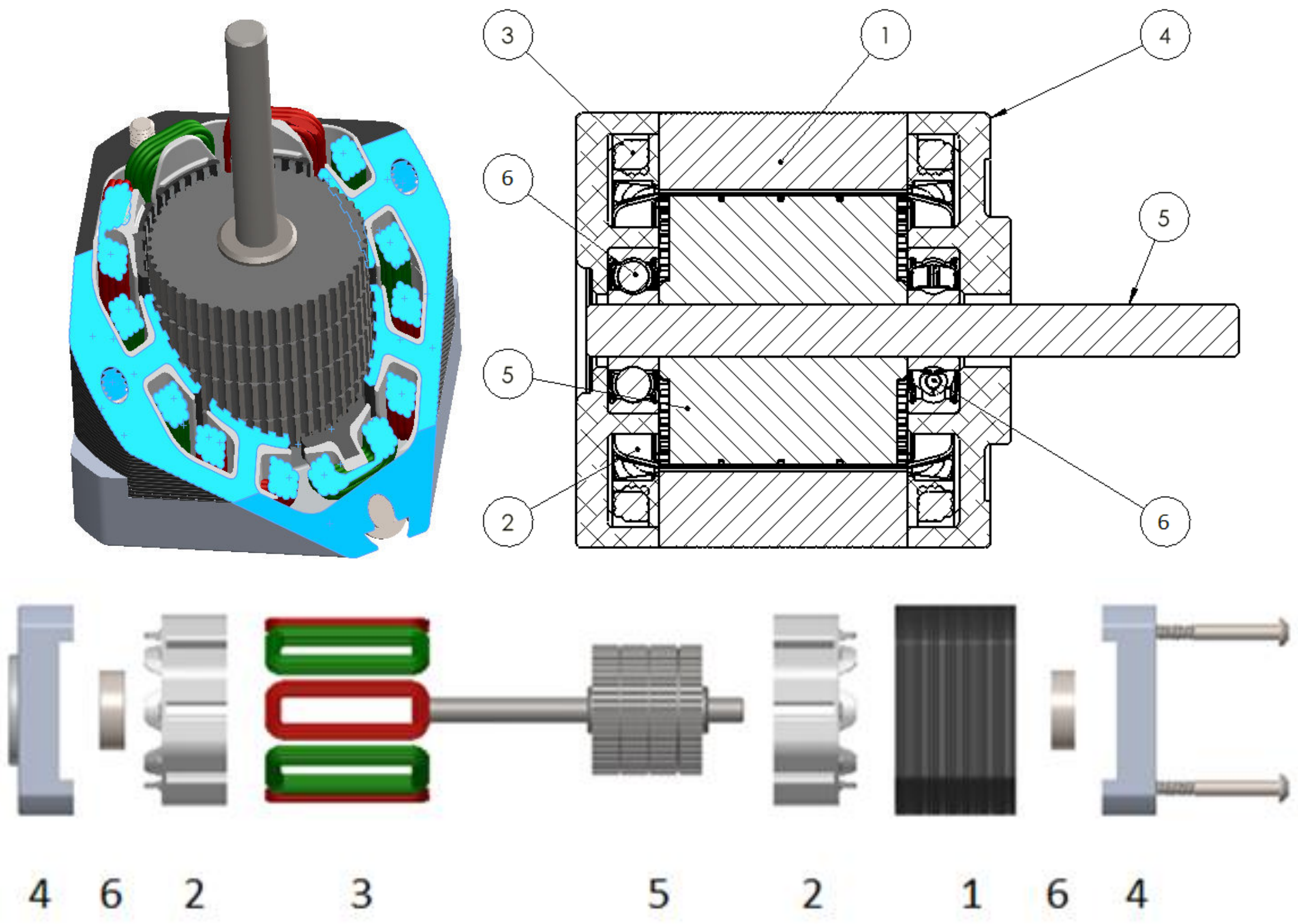



*Illustration 6: Nema 17 Assembly I*

1- Stator

2- Plastic protector

3- Winding

4- cover

5- Rotor

6- Bearing

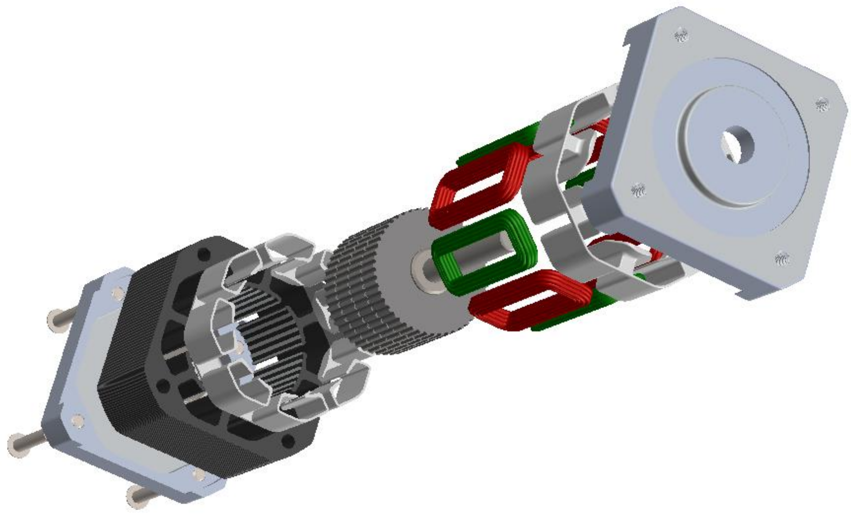


*Illustration 5: Nema 17 Assembly II*

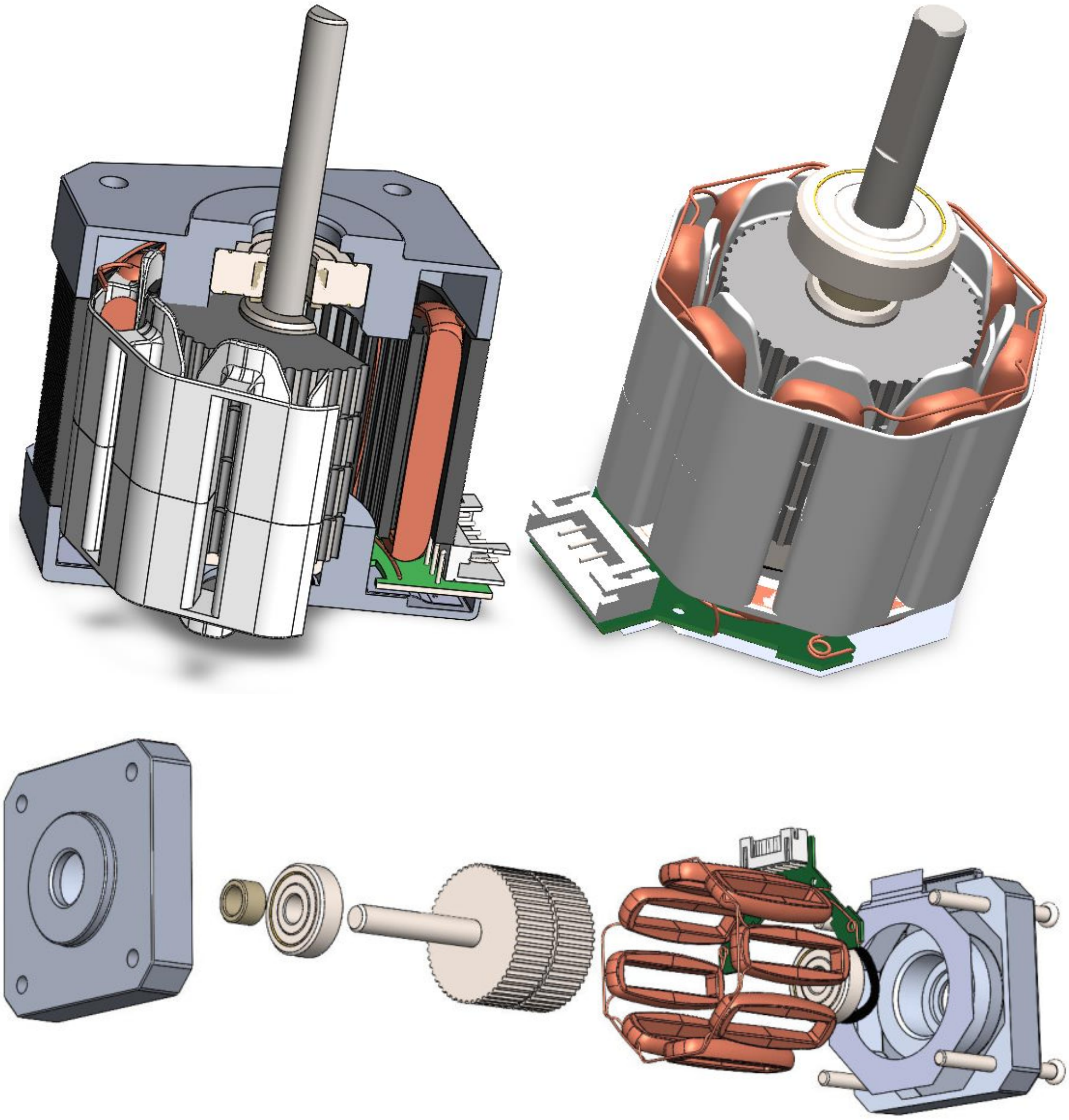


*Illustration 7: Nema Assembly III*

## 1.3 TORQUE-SPEED CURVE

Like all electric motors, Steppers also have a Torque-Speed characteristic curve.

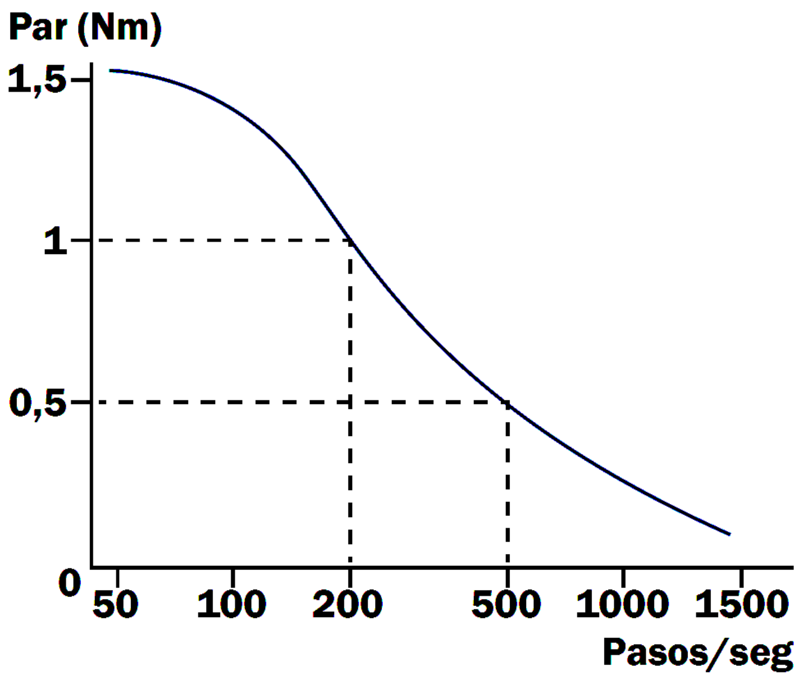



*Illustration 8: Stepper Torque-Speed Characteristic curve*

When the step rate is high, the excitation pulses of the winding are of shorter duration. The typical concurrent effect translates into a deformation of the current pulse, making its slope get reduced, reaching a significant portion of the excitation interval. This implies that when the motor works at high speeds, the current per phase never has time to reach the nominal value, as the excitation interval ends before the current reaches the nominal value and begins to decay.

At this moment, despite the fact that the current per phase begins to decay, it continues to flow through the reverse diodes, causing the excitation interval to be greater than desired, extending beyond the cut-off or blocking time. As a result, the working torque decay when increasing frequency or speed at which the winding is energized.

It is complicated to calculate the working torque (pull-out) at high speeds due to the variations in current during the excitation time of each phase mentioned. Under these conditions, the relationship between the static torque and the dynamic torque of the motor is directly proportional. For pitch ratios where the phase is driven for a time similar to the winding time constant, the waveform will be considerably distorted with an exponentially rising and falling slope. In turn, and also operating at a very high step frequency, the voltage induced in the windings of each phase due to the movement of the rotor is also not negligible. The waveform consequently cannot be defined in terms of a simple exponential function and, in order to carry out a complete analysis of the dynamic characteristics or pull-out curve the effects of the mentioned induced voltage have to be included.

One of the most influential parameters in the selection process of any electric motor and especially the stepper type is, according to the above, the torque-speed ratio, which in our case is usually measured in steps per second. This torque decreases as speed increases. However, this effect can be easily demonstrated since, assuming a constant voltage supply, it is verified that the growth of the current through the windings depends on the characteristic time constant of an RL circuit; In this way, as the frequency of the pulses increases, the effective current decreases and therefore, when the magnetic field is reduced, the torque decreases.

The pull-in and pull-out characteristic curves are defined by easily distinguishable working regions, where the curves represent the limits of said regions, which represent the zone of correct operation of the motor for certain excitation and control conditions.

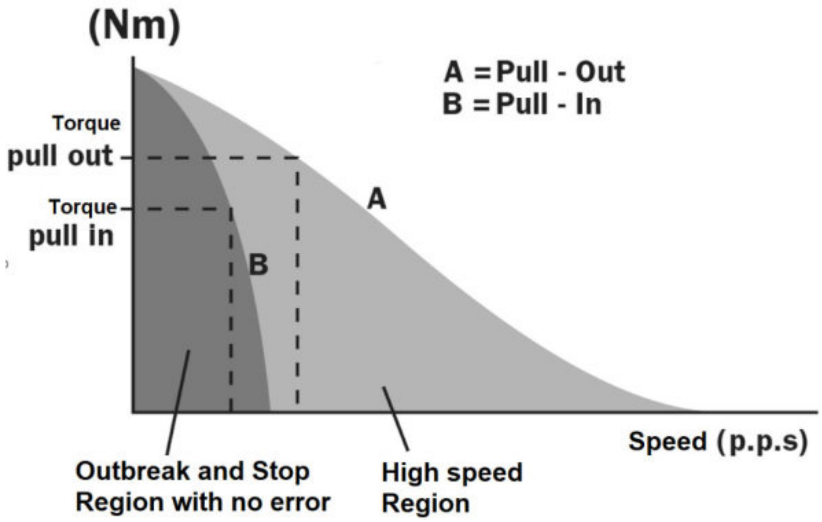



*Illustration 9: Stepper working curves*

With this study, we can extract the following conclusions:

- The pull-out torque or synchronism output torque corresponds to the maximum torque applicable to a stepper motor that allows it to continue turning without leaving the synchronism state.

- The input torque in synchronism or pull-in is the maximum torque that allows the stepper motor to start (begin to rotate) and can also remain in synchronism at a certain speed later.

Between curve B and the Cartesian axes is the error-free start-stop region (dark gray); Within this zone, the motor will be able to start from standstill (with resistive torque, applying a certain current) at a given speed without losing steps and remain in synchronism, being able to overcome any load torque that is below the limit curve. .

On the other hand, between curves A and B is the high speed region (light gray), where to reach this work zone the motor must be accelerated. In this zone the device can remain in synchronism only if the limit or synchronism output curve is not exceeded. You will not be able in any case to start or stop without losing steps in this region.

## 1.4 SLEW RATE

When a stepper motor carries a torque load (which it always is, even when working no load due to rubbing against bearings and bearings), it cannot suddenly go from zero to a certain forward speed, for instance 5000 spp. Similarly, a motor that is running at a uniform speed of 5000 spp. cannot be stopped in one step. Therefore, the engine can only be revved up gradually. Likewise, to stop a motor that is running at high speed, it must be decelerated gradually due to avoid inferring high accelerations and vibrations in the system that could damage the device. The process by which a motor speeds up or slows down is known as ramping. During the acceleration phase, the slew effect produces a progressive increase in the frequency of the electrical pulses that excite the coils. Typically, the ramping phase is completed in a fraction of a second and is generated by the power source driving the motor. This slew effect does not affect possible position or speed errors once the permanent regime has been reached.

## 1.5 MAGNETIC RESONANCE

Stepper motors respond with angular displacements of the rotor between the different equilibrium positions also called steps, according to the changes in the excitation of the stator winding. The system response to a change in input, or drive signal, which is generally highly oscillating, in applications where accurate positioning is required; will be subject to a certain degree of uncertainty. With the appearance of these effects, it can be said that the stepper motor has natural frequencies of oscillation, whose value will depend on various factors such as the inductance of the windings involved, the moment of inertia of the rotor, the friction of the bearings and bearings or the device's own damping. The windings can be arbitrary excited with a current that oscillates at a frequency such that the magnetic field-rotor set enters in a particular state called “resonance” where the rotor irretrievably loses synchronism with the supply pulses and, consequently, the torque that the motor can contribute tends to zero.

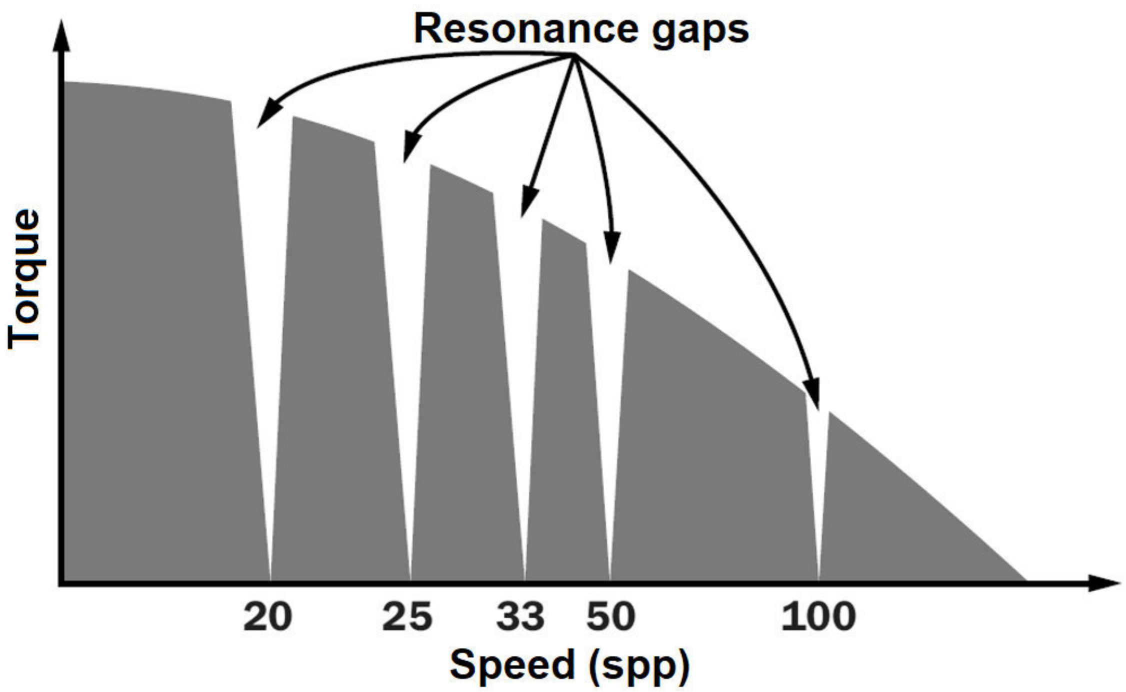



*Illustration 10: Torque looses due to resonance*

This resonance occurs at the natural oscillation frequency for the single impulse response. It can also occur if the excitation gap occurs at the instant the rotor advances towards the equilibrium position with positive speed. In the case where the natural oscillation frequency of a motor is around 100 Hz, we would have the following regions in the torque-speed characteristics curve corresponding to the harmonics of the natural frequency mentioned (examine harmonic theory and Fourier series).

$$Resonant\ Step\ Sequence = \frac{natural\ freqency}{k} \qquad [A1]$$

$$fot\ k \in \{\mathbb{Z} \mid k \geq 1\}$$

$$RPS = \frac{100}{k} = (100 \quad 50 \quad 33{,}3 \quad 25 \quad 20 \quad \ldots) \qquad [A2]$$

Possible solutions to this problem are based on adding viscous damping, friction and eddy currents to the device. All of these, however, add elements to the motor that increase its cost and reduce its performance, so a thorough study of the characteristics of the necessary motor must be made for robotics applications.

When this type of motor is already operating at high speeds, it can become unstable. The rotor may turn irregularly or simply begin to vibrate without turning any more. This instability, also known as resonance, is due to reaching the natural vibration frequency of the stepper motor (set by the material), which manifests itself in one or more speed ranges (we can move through the different harmonics of said resonant frequency). For example, a motor may enter the instability range between 2,000 and 8,000 steps per second. However, it is possible to stay within this range without losing step, achieving stable smooth speeds of up to 15,000 spp.

## 1.6 EXCITATION SEQUENCE

The control of a stepper motor is achieved by circulating electric current through the different phases of the stator winding. The different responses of this type of system can be obtained by varying the level of intensity and polarity of the excitation currents, as well as the sequential order of excitation. They are called supply or excitation sequences.

The key lies in the study of the relationship between the number of rotor and stator teeth (as well as their number). The rotor has 50 teeth (see Annex I). The stator however has two fewer teeth than the rotor (48 teeth). These 48 teeth are organized into four pairs of groups, maintaining a uniform distribution of the teeth within the set. In this way, we will be left with:

- Two groups completely aligned with the rotor (green)
- Four semi-aligned groups with the rotor (blue and purple)
- Two groups completely misaligned with the rotor (red)

Below is a diagram of the winding and how it should be excited in order to achieve the rotation of the rotor.

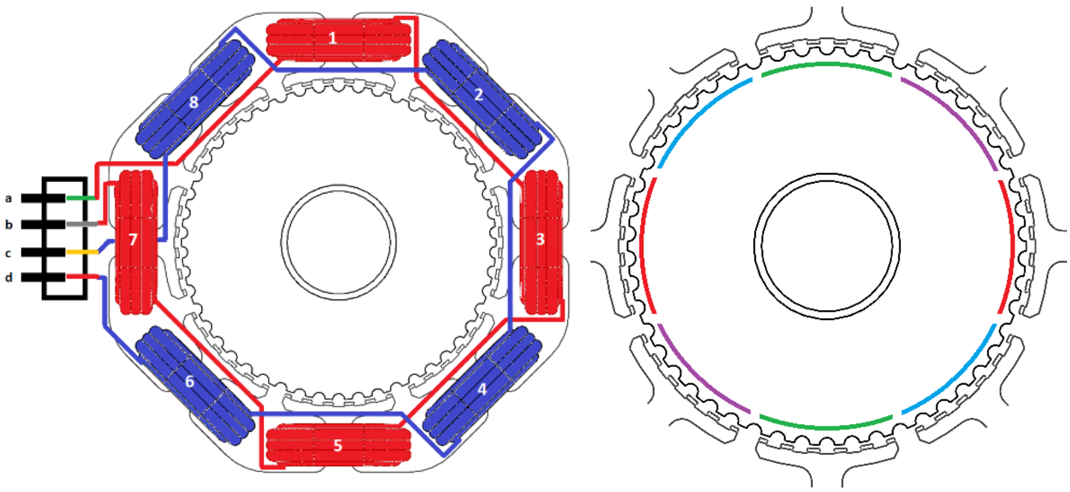



*Illustration 11: Stator winding*

The stator coils are connected as shown in the figure. If the red set is excited, the stator forms a magnetized pattern such that windings 1 and 5 act as the north pole and windings 3 and 7 as the south pole. As the rotor seen from this side acts as a south pole, the teeth in the green areas will attract each other (because they have different rotor and winding polarities) and will tend to align. The teeth of the red areas, having equal poles, will tend to repel each other, thus maintaining an initial state of equilibrium.

When we energize the blue winding, the blue and purple arcs of teeth will align (in the case of coils 4 and 8) or misalign (in the case of coils 2 and 6) completely with the new induced poles; thus producing a small turn to the next magnetic equilibrium position. This angle of rotation is the fourth part of the angle between two teeth of the rotor and is given by the expression:

$$\Phi = \frac{360}{n} \cdot \frac{1}{4} = \frac{360}{50 \cdot 4} = 1{,}8^{\text{º}} \qquad [S1.4]$$

To continue the rotation, a current is induced through the red winding again, but this time in the opposite direction to the previous one (otherwise we would return to the starting point) causing the toothed arcs corresponding to coils 3 and 7 to attract and repelling those of coils 1 and 5; this causes another turn of 1.8º until reaching the new equilibrium position.

This would be the excitation sequence of pins ***a***, ***b***, ***c*** and ***d***; to achieve a semi-continuous rotation of the rotor.

| | 0 | 1,8 | 3,6 | 5,4 | 7,2 | 9 | 10,8 | 12,6 | 14,4 | 16,2 |
|---|---|---|---|---|---|---|---|---|---|---|
| a | + | | - | | + | | - | | + | |
| b | - | | + | | - | | + | | - | |
| c | | + | | - | | + | | - | | + |
| d | | - | | + | | - | | + | | - |

Furthermore, the resulting angle can be further improved by using half steps. This method is explained in the first Addendum. The half step excitation method allows us to achieve 0,9º steps, in other words an increment of resolution of the 100% (double).

## 1.7 HALF STEP

This method is only applicable to permanent magnet motors. Note that the north and south slotted casings of the rotor should be offset by half the pitch angle.

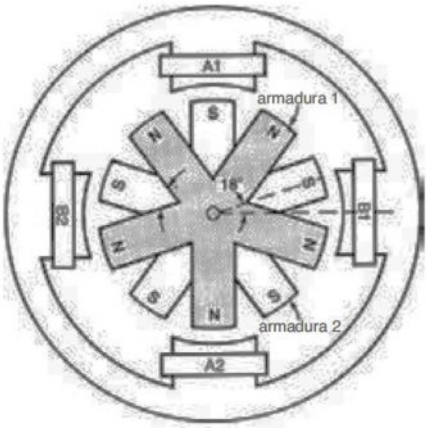



*Illustration 12: Simplified Stepper*

This example Stepper has a 5-tooth rotor. The angle of rotation is the fourth part of the angle between two teeth of the rotor and is given by the expression:

$$\Phi = \frac{360}{n} \cdot \frac{1}{4} = \frac{360}{5 \cdot 4} = 18º \qquad [A3]$$

There are three possible sequences possible:

**Simple step**. Coils A1, A2, B1 and B2 would be energized sequentially to achieve a locked turn every 18 degrees.

**Reenforced Step**. This sequence amplifies the torque achieved by maintaining the same step. Will consume almost twice the current as before.

- The coils A1 and A2 with reverse currents so that while the south shell is attracted to A1, the north shell is attracted to A2.
- Then the same thing would be done, but with B1 and B2.
- We excite again A1 and A2 but with the respective inverse currents to step 1.
- We excite B1 and B2 again but with the respective inverse currents of step 2.
- We repeat in a loop.

Half Step. With the previous configuration we can also achieve half step angles. If we drive coils A1 and B2 in such a way that they attract the north casing, and we drive A2 and B1 in such a way that they attract the south casing, we will get the rotor to balance at an intermediate angle where no tooth is completely aligned with any other coil. To continue, the previously explained double excitation scheme is followed, thus achieving a continuous rotation passing through the equilibrium states every 18º but being able to stop and lock the rotor in multiples of 9º as well.

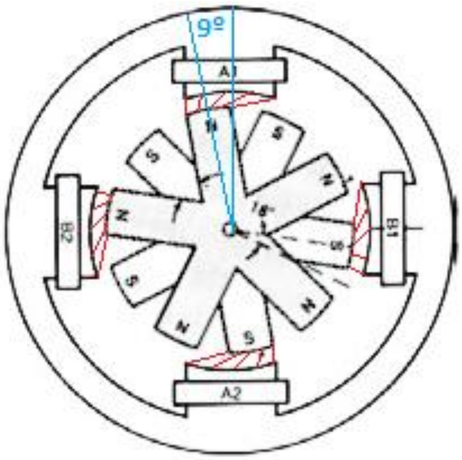



*Illustration 13: Half Step*

The issue of this method is that the **effective torque decreases** (despite the current consumption increment) because the average distance between the rotor permanent magnet and winding magnetic poles grows; and at the same time the confronted effective surface also decreases

If we use the half step excitation method in our robot we will be able to achieve 0,9º steps.

# 2 ANNEX II – DYNAMIXEL SERVOMOTORS

## 2.1 CONTROL TABLE OF EEPROM AREA

| Address | Size(Byte) | Data Name | Access | Initial Value | Range | Unit |
|---|---|---|---|---|---|---|
| 0 | 2 | Model Number | R | 1,06 | - | - |
| 2 | 4 | Model Information | R | - | - | - |
| 6 | 1 | Firmware Version | R | - | - | - |
| 7 | 1 | ID | RW | 1 | 0 ~ 252 | - |
| 8 | 1 | Baud Rate | RW | 1 | 0 ~ 7 | - |
| 9 | 1 | Return Delay Time | RW | 250 | 0 ~ 254 | 2 [μsec] |
| 10 | 1 | Drive Mode | RW | 0 | 0 ~ 5 | - |
| 11 | 1 | Operating Mode | RW | 3 | 0 ~ 16 | - |
| 12 | 1 | Secondary(Shadow) ID | RW | 255 | 0 ~ 252 | - |
| 13 | 1 | Protocol Type | RW | 2 | 1 ~ 2 | - |
| 20 | 4 | Homing Offset | RW | 0 | -1,044,479 ~ 1,044,479 | 1 [pulse] |
| 24 | 4 | Moving Threshold | RW | 10 | 0 ~ 1,023 | 0.229 [rev/min] |
| 31 | 1 | Temperature Limit | RW | 72 | 0 ~ 100 | 1 [°C] |
| 32 | 2 | Max Voltage Limit | RW | 140 | 60 ~ 140 | 0.1 [V] |
| 34 | 2 | Min Voltage Limit | RW | 60 | 60 ~ 140 | 0.1 [V] |
| 36 | 2 | PWM Limit | RW | 885 | 0 ~ 885 | 0.113 [%] |
| 44 | 4 | Velocity Limit | RW | 265 | 0 ~ 1,023 | 0.229 [rev/min] |
| 48 | 4 | Max Position Limit | RW | 4,095 | 0 ~ 4,095 | 1 [pulse] |
| 52 | 4 | Min Position Limit | RW | 0 | 0 ~ 4,095 | 1 [pulse] |
| 60 | 1 | Startup Configuration | RW | 0 | 3 | - |
| 63 | 1 | Shutdown | RW | 52 | - | - |

## 2.2 CONTROL TABLE OF RAM AREA

| Address | Size(Byte) | Data Name | Access | Initial Value | Range | Unit |
|---|---|---|---|---|---|---|
| 64 | 1 | Torque Enable | RW | 0 | 0 ~ 1 | - |
| 65 | 1 | LED | RW | 0 | 0 ~ 1 | - |
| 68 | 1 | Status Return Level | RW | 2 | 0 ~ 2 | - |
| 69 | 1 | Registered Instruction | R | 0 | 0 ~ 1 | - |
| 70 | 1 | Hardware Error Status | R | 0 | - | - |
| 76 | 2 | Velocity I Gain | RW | 1 | 0 ~ 16,383 | - |
| 78 | 2 | Velocity P Gain | RW | 100 | 0 ~ 16,383 | - |
| 80 | 2 | Position D Gain | RW | 4 | 0 ~ 16,383 | - |
| 82 | 2 | Position I Gain | RW | 0 | 0 ~ 16,383 | - |
| 84 | 2 | Position P Gain | RW | 640 | 0 ~ 16,383 | - |
| 88 | 2 | Feedforward 2nd Gain | RW | 0 | 0 ~ 16,383 | - |

| Address | Size(Byte) | Data Name | Access | Initial Value | Range | Unit |
|---|---|---|---|---|---|---|
| 90 | 2 | Feedforward 1st Gain | RW | 0 | 0 ~ 16,383 | - |
| 98 | 1 | Bus Watchdog | RW | 0 | 1 ~ 127 | 20 [msec] |
| 100 | 2 | Goal PWM | RW | - | -PWM Limit(36) ~ PWM Limit(36) | 0.113 [%] |
| 104 | 4 | Goal Velocity | RW | - | -Velocity Limit(44) ~ Velocity Limit(44) | 0.229 [rev/min] |
| | | | | | 0 ~ 32,767 | 214.577 [rev/min$^2$] |
| 108 | 4 | Profile Acceleration | RW | 0 | 0 ~ 32,737 | 1 [ms] |
| 112 | 4 | Profile Velocity | RW | 0 | 0 ~ 32,767 | 0.229 [rev/min] |
| 116 | 4 | Goal Position | RW | - | Min Position Limit(52) ~ Max Position Limit(48) | 1 [pulse] |
| 120 | 2 | Realtime Tick | R | - | 0 ~ 32,767 | 1 [msec] |
| 122 | 1 | Moving | R | 0 | 0 ~ 1 | - |
| 123 | 1 | Moving Status | R | 0 | - | - |
| 124 | 2 | Present PWM | R | - | - | - |
| 126 | 2 | Present Load | R | - | -1,000 ~ 1,000 | 0.1 [%] |
| 128 | 4 | Present Velocity | R | - | - | 0.229 [rev/min] |
| 132 | 4 | Present Position | R | - | - | 1 [pulse] |
| 136 | 4 | Velocity Trajectory | R | - | - | 0.229 [rev/min] |
| 140 | 4 | Position Trajectory | R | - | - | 1 [pulse] |
| 144 | 2 | Present Input Voltage | R | - | - | 0.1 [V] |
| 146 | 1 | Present Temperature | R | - | - | 1 [°C] |
| 147 | 1 | Backup Ready | R | - | 0 ~ 1 | - |
| 168 | 2 | Indirect Address 1 | RW | 224 | 64 ~ 661 | - |
| 170 | 2 | Indirect Address 2 | RW | 225 | 64 ~ 661 | - |
| 172 | 2 | Indirect Address 3 | RW | 226 | 64 ~ 661 | - |
| … | … | … | … | … | - | - |
| 218 | 2 | Indirect Address 26 | RW | 249 | 64 ~ 661 | - |
| 220 | 2 | Indirect Address 27 | RW | 250 | 64 ~ 661 | - |
| 222 | 2 | Indirect Address 28 | RW | 251 | 64 ~ 661 | - |
| 224 | 1 | Indirect Data 1 | RW | 0 | 0 ~ 255 | - |
| 225 | 1 | Indirect Data 2 | RW | 0 | 0 ~ 255 | - |
| 226 | 1 | Indirect Data 3 | RW | 0 | 0 ~ 255 | - |
| … | … | … | … | … | - | - |

| Address | Size(Byte) | Data Name | Access | Initial Value | Range | Unit |
|---|---|---|---|---|---|---|
| 249 | 1 | Indirect Data 26 | RW | 0 | 0 ~ 255 | - |
| 250 | 1 | Indirect Data 27 | RW | 0 | 0 ~ 255 | - |
| 251 | 1 | Indirect Data 28 | RW | 0 | 0 ~ 255 | - |
| 578 | 2 | Indirect Address 29 | RW | 634 | 64 ~ 661 | - |
| 580 | 2 | Indirect Address 30 | RW | 635 | 64 ~ 661 | - |
| 582 | 2 | Indirect Address 31 | RW | 636 | 64 ~ 661 | - |
| … | … | … | … | … | - | - |
| 628 | 2 | Indirect Address 54 | RW | 659 | 64 ~ 661 | - |
| 630 | 2 | Indirect Address 55 | RW | 660 | 64 ~ 661 | - |
| 632 | 2 | Indirect Address 56 | RW | 661 | 64 ~ 661 | - |
| 634 | 1 | Indirect Data 29 | RW | 0 | 0 ~ 255 | - |
| 635 | 1 | Indirect Data 30 | RW | 0 | 0 ~ 255 | - |
| 636 | 1 | Indirect Data 31 | RW | 0 | 0 ~ 255 | - |
| … | … | … | … | … | - | - |
| 659 | 1 | Indirect Data 54 | RW | 0 | 0 ~ 255 | - |
| 660 | 1 | Indirect Data 55 | RW | 0 | 0 ~ 255 | - |
| 661 | 1 | Indirect Data 56 | RW | 0 | 0 ~ 255 | - |

## 2.3 SERIAL COMMUNICATION

These are the serial communication specifications, essential to stablish connection.

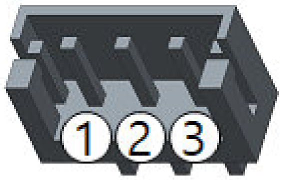
### 2.3.1 BAUDRATE TABLE

Here is shown the value to assign at the instruction 8 to set communication baudrate

| Value | Baud Rate |
|---|---|
| 7 | 4.5M [bps] |
| 6 | 4M [bps] |
| 5 | 3M [bps] |
| 4 | 2M [bps] |
| 3 | 1M [bps] |
| 2 | 115,200 [bps] |
| 1(Default) | 57,600 [bps] |
| 0 | 9,600 [bps] |

## 2.4 TTL CONNECTOR

| Item | TTL |
|---|---|
| Pinout | 1 GND |
| | 2 VDD |
| | 3 DATA |
| Housing | JST EHR-03 |
| PCB Header | |
| | JST B3B-EH-A |
| Crimp Terminal | JST SEH-001T-P0.6 |
| Wire Gauge for DYNAMIXEL | 21 AWG |




## 2.5 DYNAMIXEL SDK

DYNAMIXEL SDK is a software that provides DYNAMIXEL control functions using packet communication. The API of DYNAMIXEL SDK is designed for DYNAMIXEL actuators and DYNAMIXEL-based platforms. Supported Programming Languages and Features:

*C, C++, C#, Python, Java, MATLAB, LabVIEW*

*Windows, Mac, Linux.*

*ROS*

*Arduino*

## 2.6 FCC

The DYNAMIXEL equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.

- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

## 2.7 DRIVE MODE

The Drive Mode (address 10) configures Drive Mode of DYNAMIXEL.

| Bit | Item | Description |
|---|---|---|
| Bit 7(0x80) | - | Unused, always ‘0’ |
| Bit 6(0x40) | - | Unused, always ‘0’ |
| Bit 5(0x20) | - | Unused, always ‘0’ |
| Bit 4(0x10) | - | Unused, always ‘0’ |
| Bit 3(0x08) | Torque On by Goal Update | [0] Performing a given command only if the value of Torque Enable(64) is ‘1’<br>[1] Performing a given command regardless of the set value of Torque Enable(64). If the value of Torque Enable(64) is ‘0’ and the command is given, the Torque Enable(64) switches to ‘1’ and perform the command. |
| Bit 2(0x04) | Profile Configuration | [0] Velocity-based Profile: Create a Profile based on Velocity<br>[1] Time-based Profile: Create Profile based on time<br>※ See What is the Profile |
| Bit 1(0x02) | - | Unused, always ‘0’ |
| Bit 0(0x01) | Normal/Reverse Mode | [0] Normal Mode: CCW(Positive), CW(Negative)<br>[1] Reverse Mode: CCW(Negative), CW(Positive) |

## 2.8 TEMPERATURE LIMIT

The Temperature Limit instruction (address 31) limits operating temperature of the DYNAMIXEL. When the Present Temperature (address 146) is higher than the settled temperature limit, the Overheating Error Bit(0x04) and Alert Bit(0x80) in the Hardware Error Status (address 70) will be set. If Overheating Error Bit(0x04) is configured in the Shutdown (address 63), Torque Enable (address 64) will be set to ‘0’ (Torque OFF).

## 2.9 VOLTAGE LIMIT

The Min Voltage Limit (address 32) and Max Voltage Limit (address 34) determine the maximum and minimum operating voltages. When the Present Input Voltage (address 144) indicating the present input voltage to the device exceeds the range of Max Voltage Limit (address 32) and Min Voltage Limit (address 34), the Input Voltage error Bit(0x10) in the Hardware Error Status (address 70) will be set, and the Status Packet will send Alert Bit(0x80) via the Error field.

## 2.10 PWM LIMIT

The PWM Limit (address 36) indicates maximum PWM output. Goal PWM (address 100) cannot be configured with any values exceeding PWM Limit (address 36). PWM Limit (address 36) is used in operating mode as an output limit, therefore a variation in PWM output will result in the change of torque and velocity.

## 2.11 SPEED LIMIT

Velocity Limit (address 44) indicates the maximum value of the target speed that can be written in Goal Velocity (address 104).

## 2.12 POSITION LIMIT

These values limit maximum and minimum target positions for Position Control Mode. Therefore, Goal Position (address 116) should be configured within the position limit range. These values are not used in Extended Position Control Mode.

For more information about Dynamixels go to https://emanual.robotis.com/docs/en/dxl/.

# 3 BUY LINKS FOR DIFFERENT COMPONENTS

| | |
|---|---|
| **Nema 17** | https://www.amazon.es/Stepper-Longruner-Connector-Mounting-LQD03/dp/B071P6JPF9/ref=sr_1_2_sspa?keywords=Nema%2B17%2BStepper%2BMotor&qid=1658091042&sr=8-2-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzNjBNUkdJQ0ZVWlBZJmVuY3J5cHRlZElkPUEwNjYwNTA1VVZVNkhSNklDQk1NJmVuY3J5cHRlZEFkSWQ9QTA2MDM3MzcxWllCTFlKMFpKTzBOJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ&th=1 |
| **Dynamixel XL 430** | https://www.robot-advance.com/EN/art-servomotor-dynamixel-xl430-w250-t-2363.htm |
| **Dynamixel XL 320** | https://ro-botica.com/Producto/ROBOTIS-DYNAMIXEL-XL-320/ |
| **Raspberry Pi 3B** | https://www.amazon.es/Raspberry-Pi-3-Model-3-Modelo/dp/B07BDR5PDW/ref=sr_1_7?keywords=raspberry%2Bpi%2B3b&qid=1658092233&sprefix=rasp%2Caps%2C363&sr=8-7&th=1 |
| **SOLDIWORKS License** | https://www.solidworks.com/es/product/students |

UNIVERSIDAD DE LA RIOJA

*Escuela Técnica Superior de Ingeniería Industrial*

*Grado en Ingeniería Electrónica Industrial y Automática*

# LOW-COST DIDACTIC ROBOT

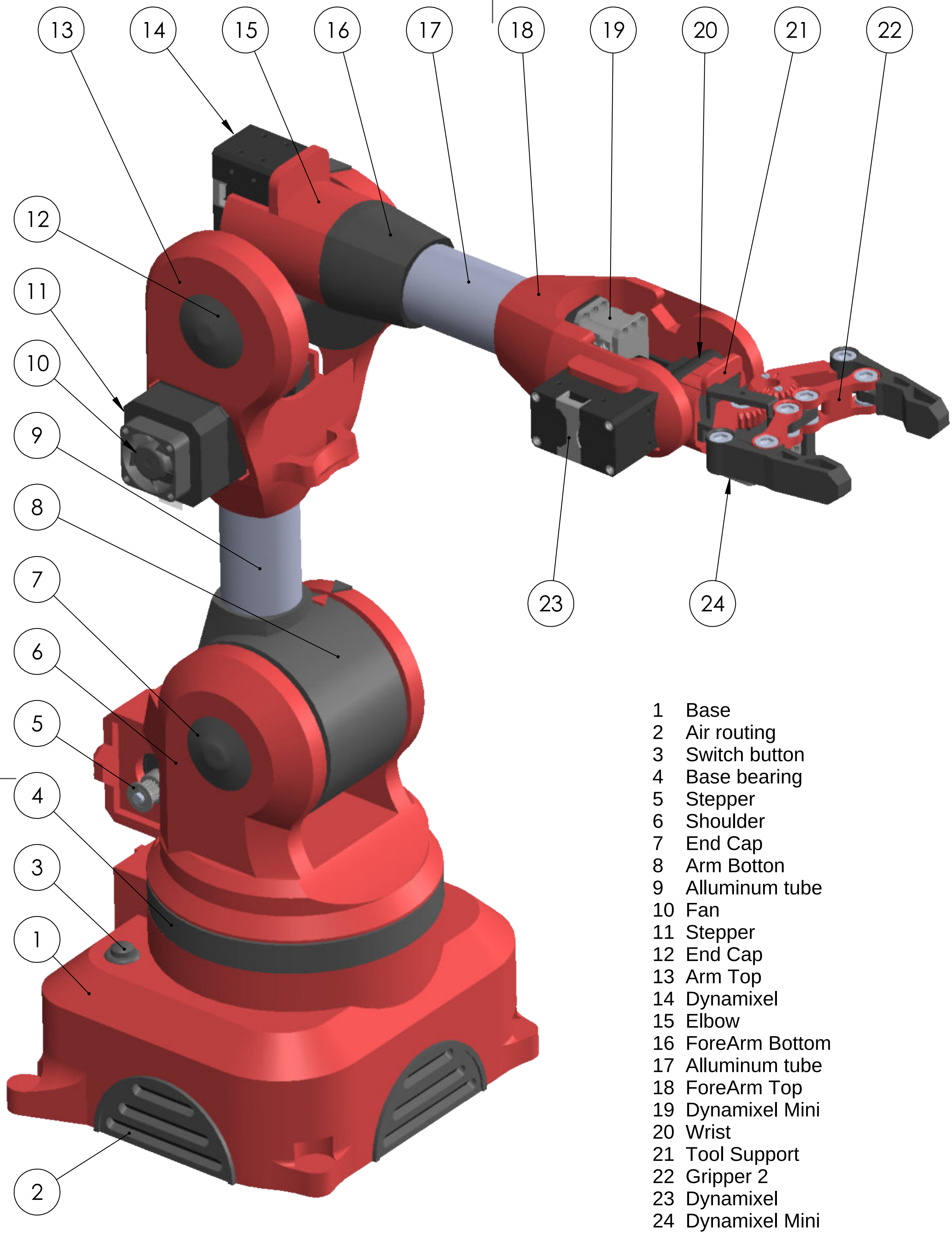
## **BluePrints**

Autor: Miguel Ángel Ezquerro Ezquerro

Tutor: Carlos Elvira Izurrategui

Date: 18 July 2022

# Index




1 Base
2 Air routing
3 Switch button
4 Base bearing
5 Stepper
6 Shoulder
7 End Cap
8 Arm Botton
9 Alluminum tube
10 Fan
11 Stepper
12 End Cap
13 Arm Top
14 Dynamixel
15 Elbow
16 ForeArm Bottom
17 Alluminum tube
18 ForeArm Top
19 Dynamixel Mini
20 Wrist
21 Tool Support
22 Gripper 2
23 Dynamixel
24 Dynamixel Mini

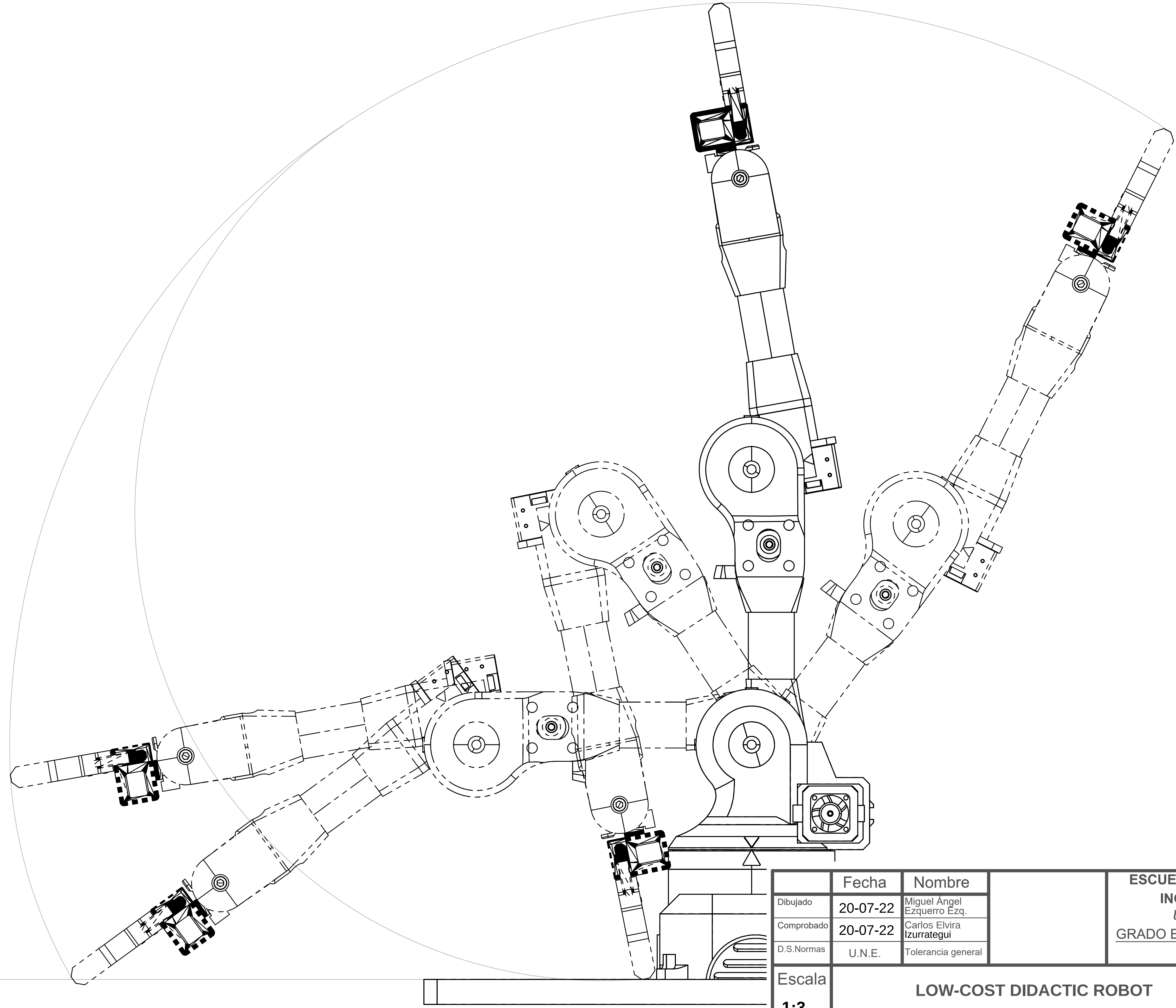
| | Fecha | Nombre |
|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. |
| Comprobado | 20-07-22 | J. Santamaría Peña |
| D.S.Normas | U.N.E. | Tolerancia general |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

| | | | |
|---|---|---|---|
| Escala **none** | **LOW-COST DIDACTIC ROBOT** | Número: | 1 |
| Proyección | **Full Assembly** | Referencia: | 001 |
| | | Sustituye a | |
| | | Sustituido por | |



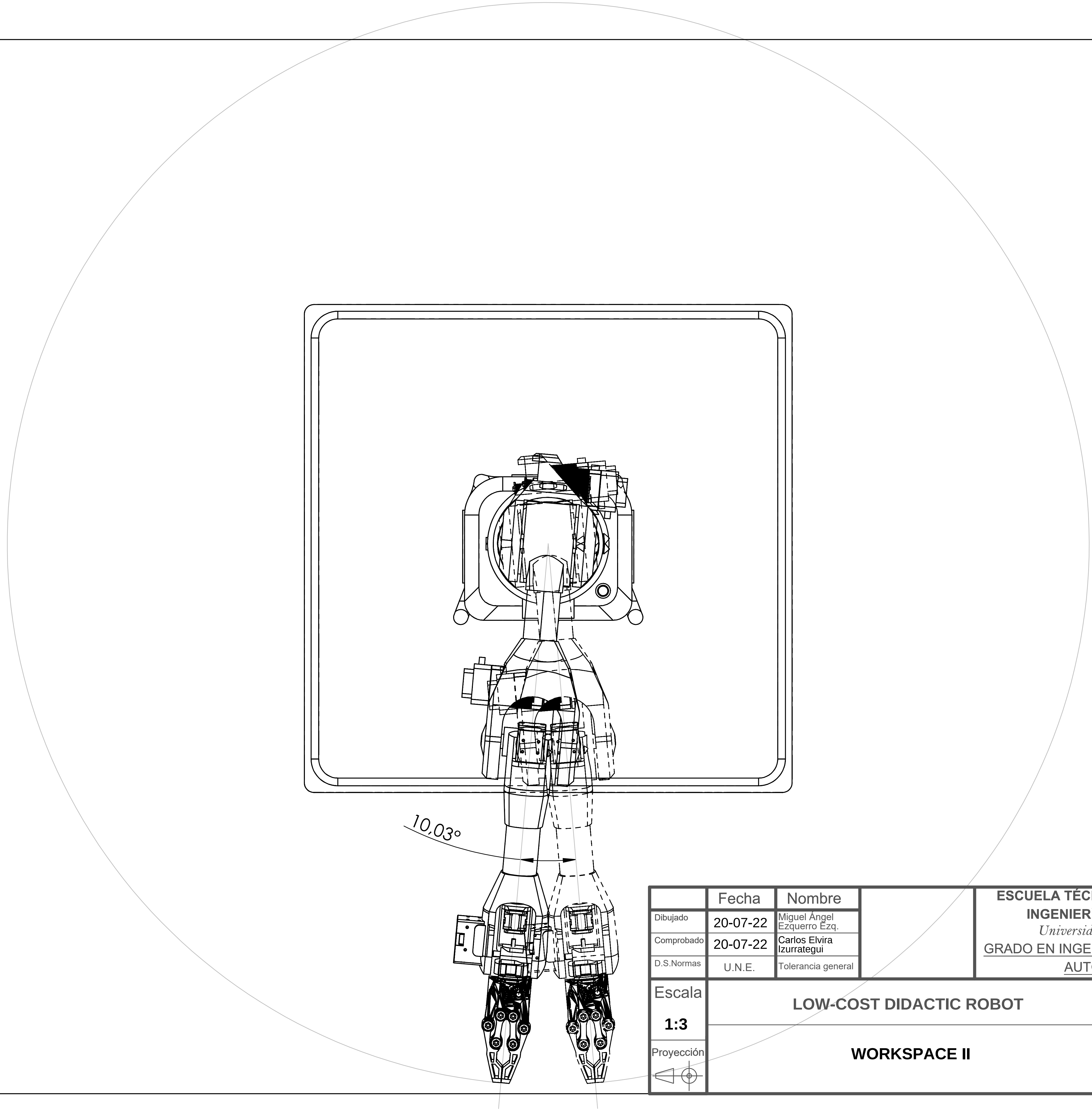
| | Fecha | Nombre | |
|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | |
| Comprobado | 20-07-22 | Carlos Elvira Izurrategui | |
| D.S.Normas | U.N.E. | Tolerancia general | |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

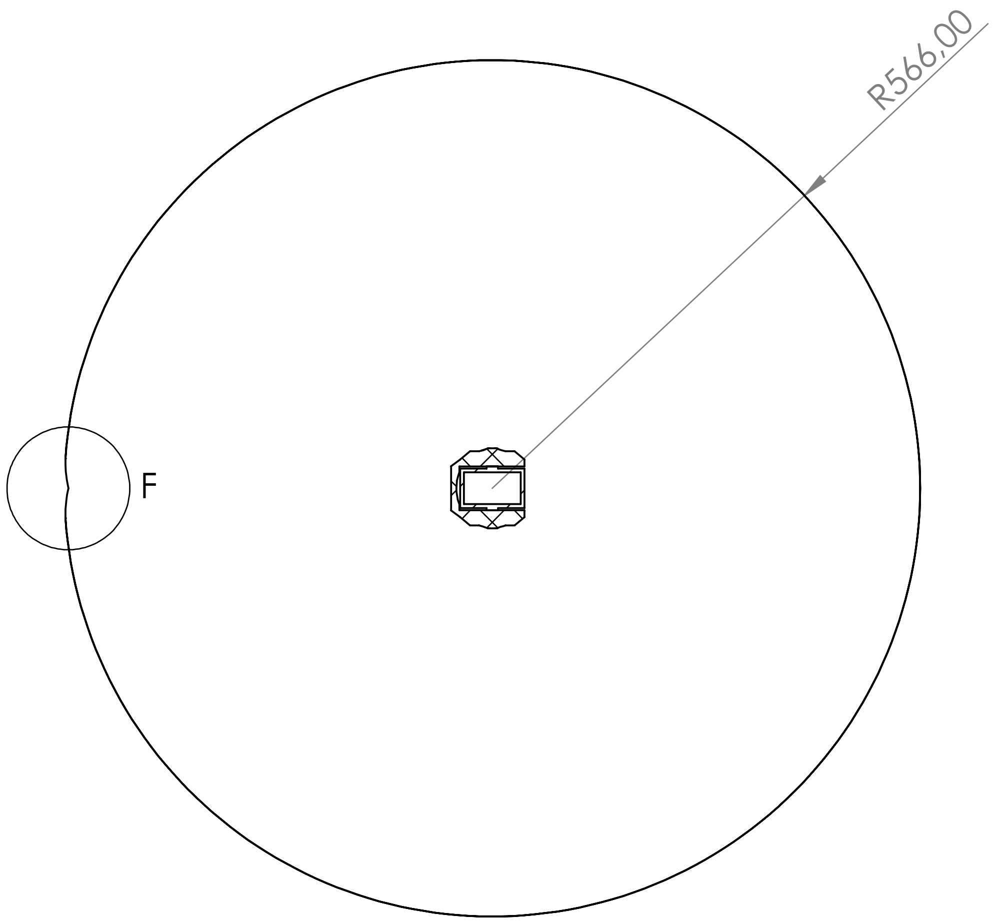
| | | | |
|---|---|---|---|
| Escala 1:3 | **LOW-COST DIDACTIC ROBOT** | Número: | 2 |
| | | Referencia: | 002 |
| Proyección | **WORKSPACE I** | Sustituye a | |
| | | Sustituido por | |



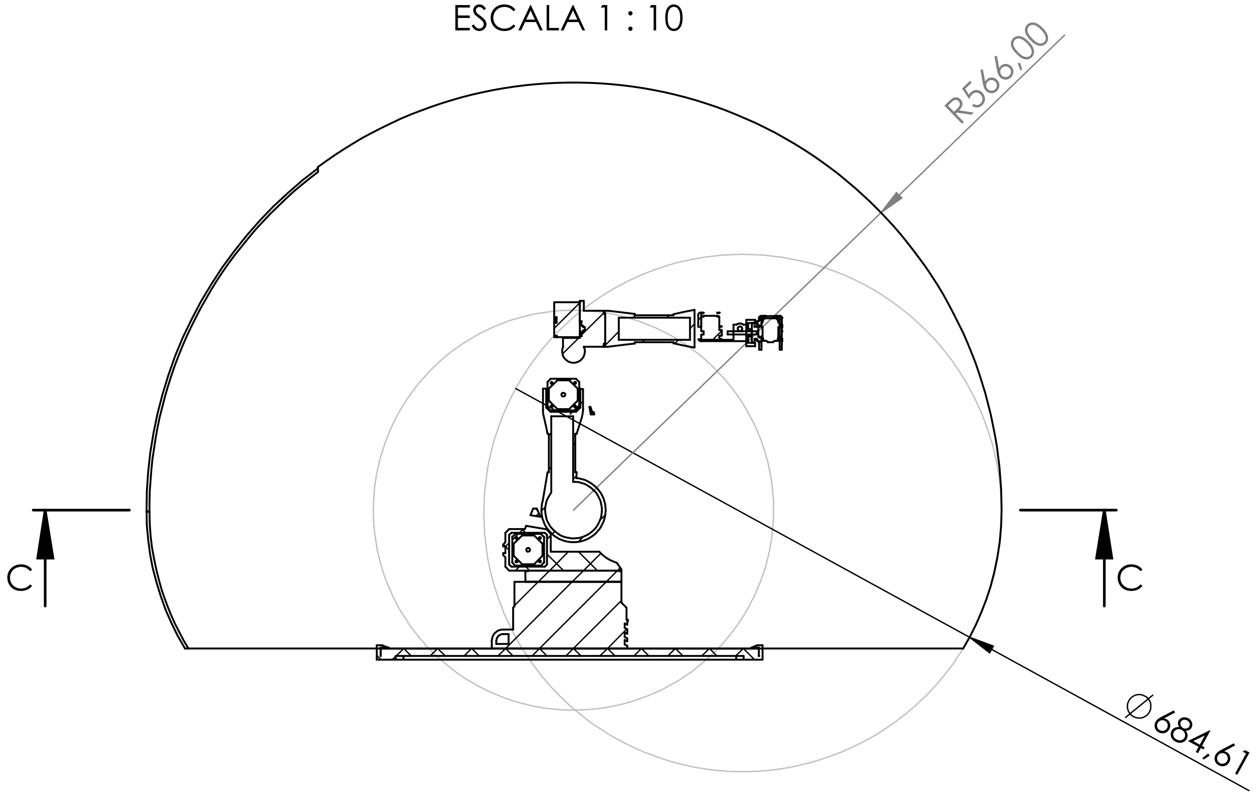

| | Fecha | Nombre | |
|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | |
| Comprobado | 20-07-22 | Carlos Elvira Izurrategui | |
| D.S.Normas | U.N.E. | Tolerancia general | |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**
*Universidad de La Rioja*
GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

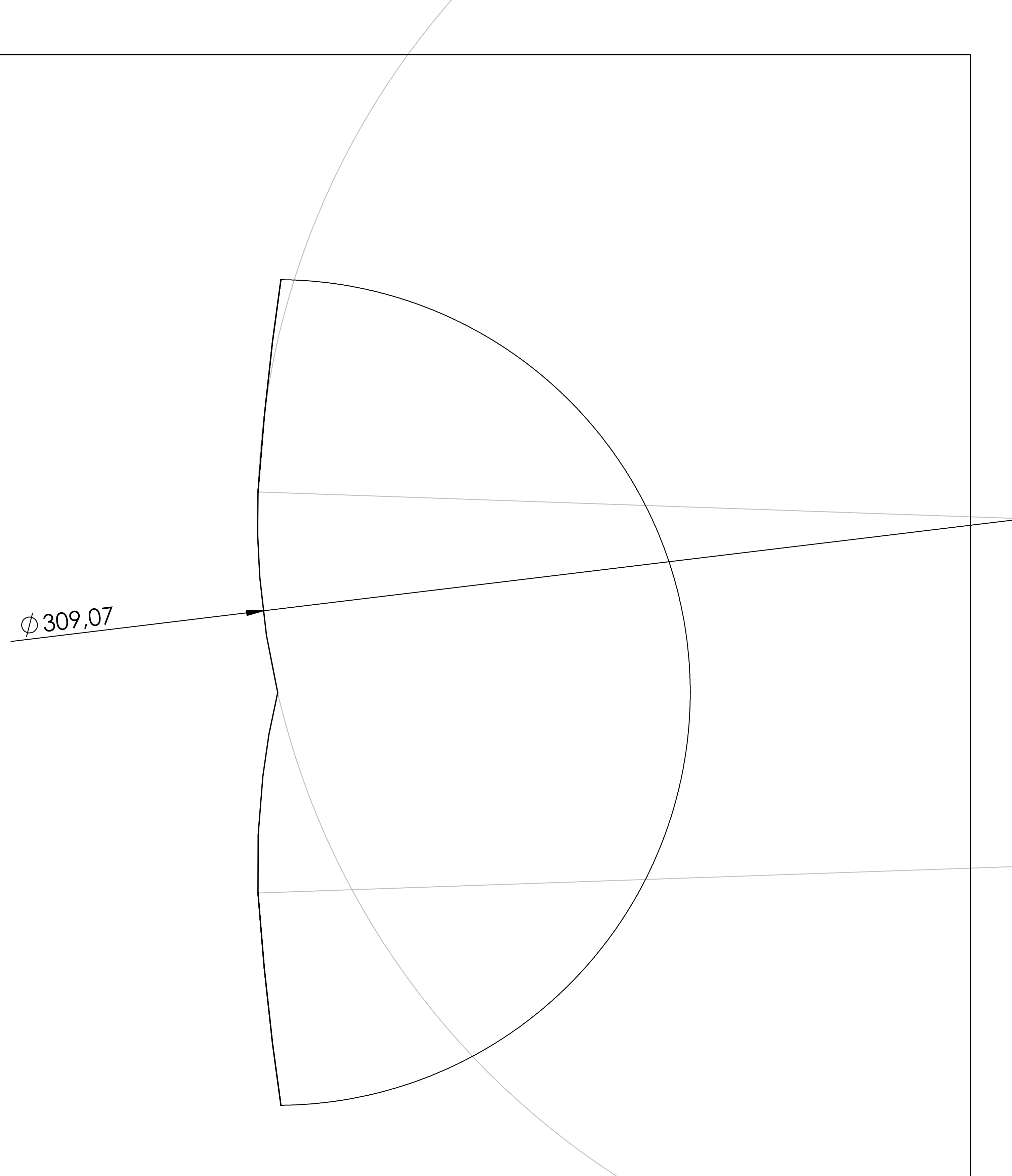
| Escala | | Número: 3 |
|---|---|---|
| 1:3 | **LOW-COST DIDACTIC ROBOT** | Referencia: 003 |
| Proyección | **WORKSPACE II** | Sustituye a |
| | | Sustituido por |




SECCIÓN C-C
ESCALA 1 : 10

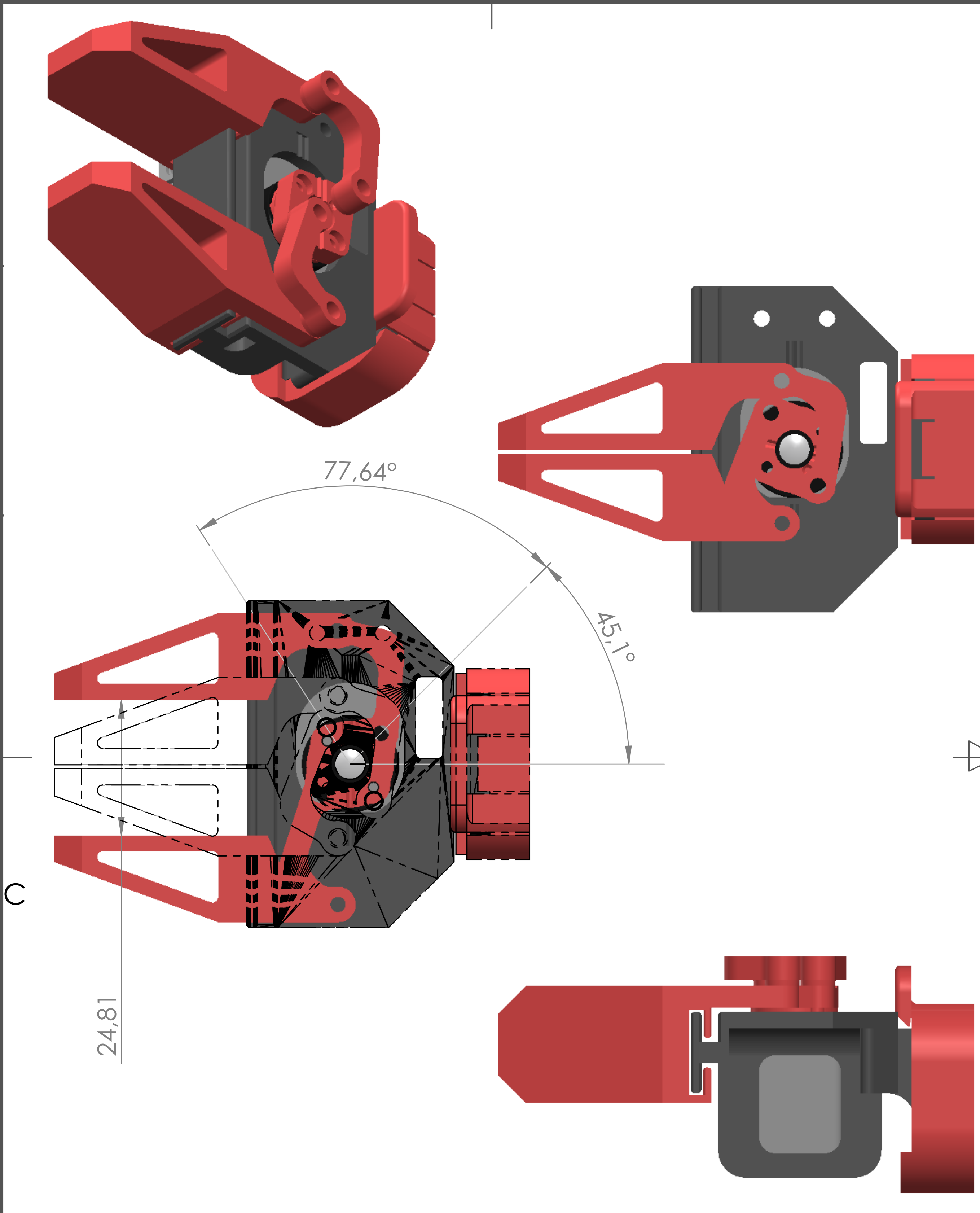



SECCIÓN A-A
ESCALA 1 : 10




| | Fecha | Nombre | | |
|---|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL<br>*Universidad de La Rioja*<br>GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA |
| Comprobado | 20-07-22 | Carlos Elvira Izurrategui | | |
| D.S.Normas | U.N.E. | Tolerancia general | | |

| | | | |
|---|---|---|---|
| Escala<br>**1:3** | **LOW-COST DIDACTIC ROBOT** | Número: | 4 |
| | | Referencia: | 004 |
| Proyección | **WORKSPACE III** | Sustituye a | |
| | | Sustituido por | |



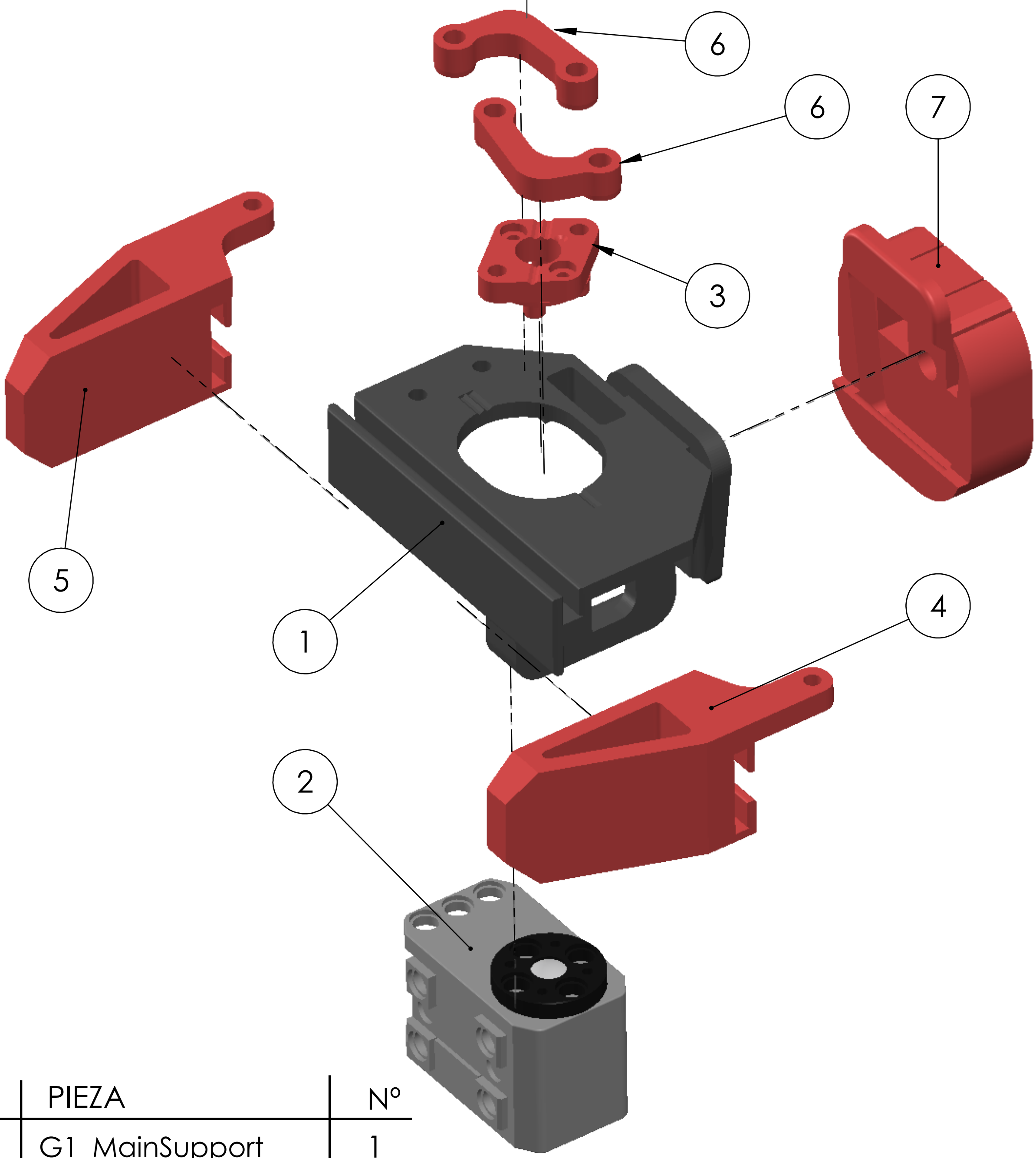

| | Fecha | Nombre | |
|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | |
| Comprobado | 20-07-22 | J. Santamaría Peña | |
| D.S.Normas | U.N.E. | Tolerancia general | |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

| Escala 1:1 | **LOW-COST DIDACTIC ROBOT** | Número: 5 |
|---|---|---|
| | | Referencia: 005 |
| Proyección | **Gripper 1** | Sustituye a |
| | | Sustituido por |




| | PIEZA | Nº |
|---|---|---|
| 1 | G1_MainSupport | 1 |
| 2 | G1_dynamixel_mini | 1 |
| 3 | G1_ServoHead | 1 |
| 4 | clamp left | 1 |
| 5 | clamp right | 1 |
| 6 | rod | 2 |
| 7 | 7-Hand_support_2 | 1 |

| | Fecha | Nombre |
|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. |
| Comprobado | 20-07-22 | J. Santamaría Peña |
| D.S.Normas | U.N.E. | Tolerancia general |

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL
*Universidad de La Rioja*
GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

Escala **1:2**

Proyección

**LOW-COST DIDACTIC ROBOT**

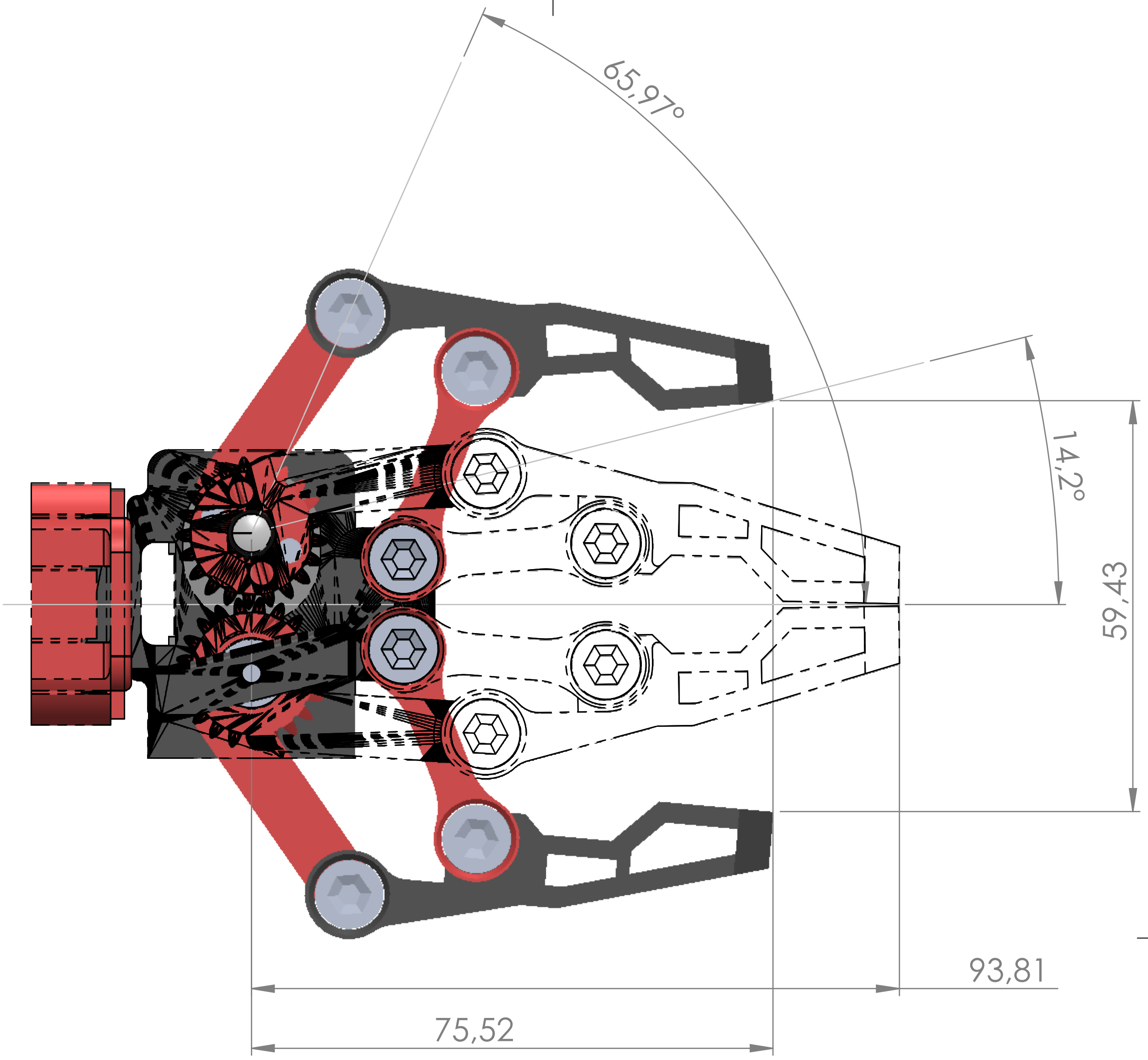
**Gripper 1 Assembly**

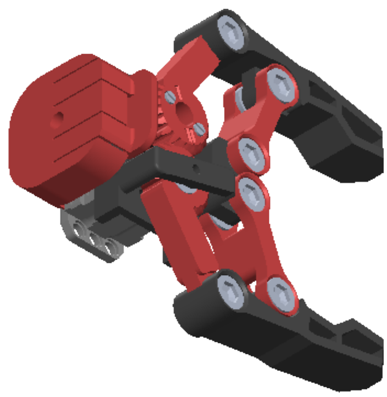
Número: 6

Referencia: 006

Sustituye a

Sustituido por

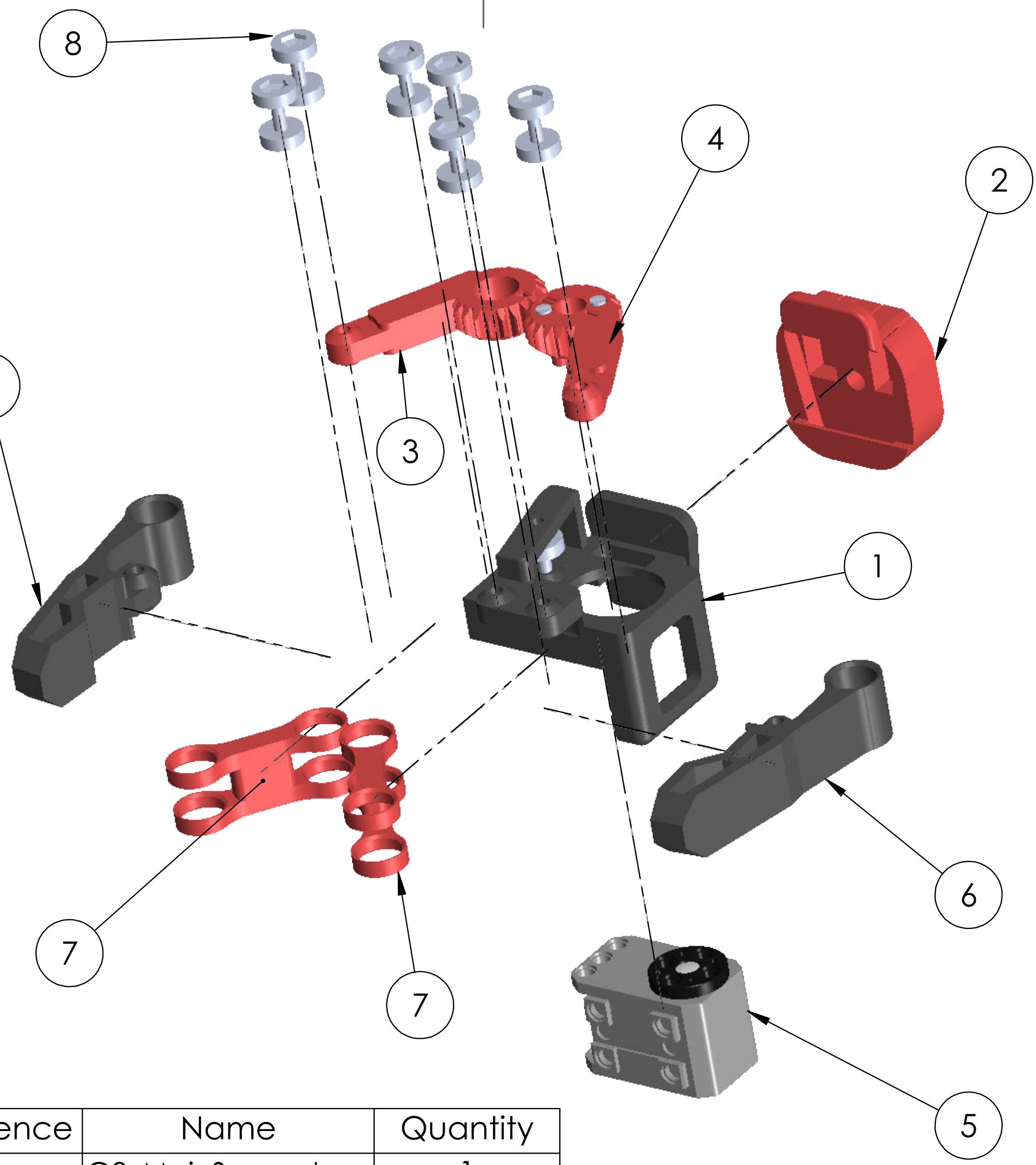


| | Fecha | Nombre |
|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. |
| Comprobado | 20-07-22 | J. Santamaría Peña |
| D.S.Normas | U.N.E. | Tolerancia general |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**
*Universidad de La Rioja*
GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

| | | | |
|---|---|---|---|
| Escala **1:1** | **LOW-COST DIDACTIC ROBOT** | Número: | 7 |
| | | Referencia: | 007 |
| Proyección | **Gripper 2** | Sustituye a | |
| | | Sustituido por | |




| Reference | Name | Quantity |
|---|---|---|
| 1 | G2_MainSupport | 1 |
| 2 | 7-Hand_support_2 | 1 |
| 3 | G2_GearRight | 1 |
| 4 | G2_GearLeft | 1 |
| 5 | G2_dynamixel_mini | 1 |
| 6 | CLAMP | 2 |
| 7 | G2_Rod | 2 |
| 8 | nexo | 7 |

| | Fecha | Nombre |
|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. |
| Comprobado | 20-07-22 | J. Santamaría Peña |
| D.S.Normas | U.N.E. | Tolerancia general |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

Escala 1:1

Proyección

**LOW-COST DIDACTIC ROBOT**

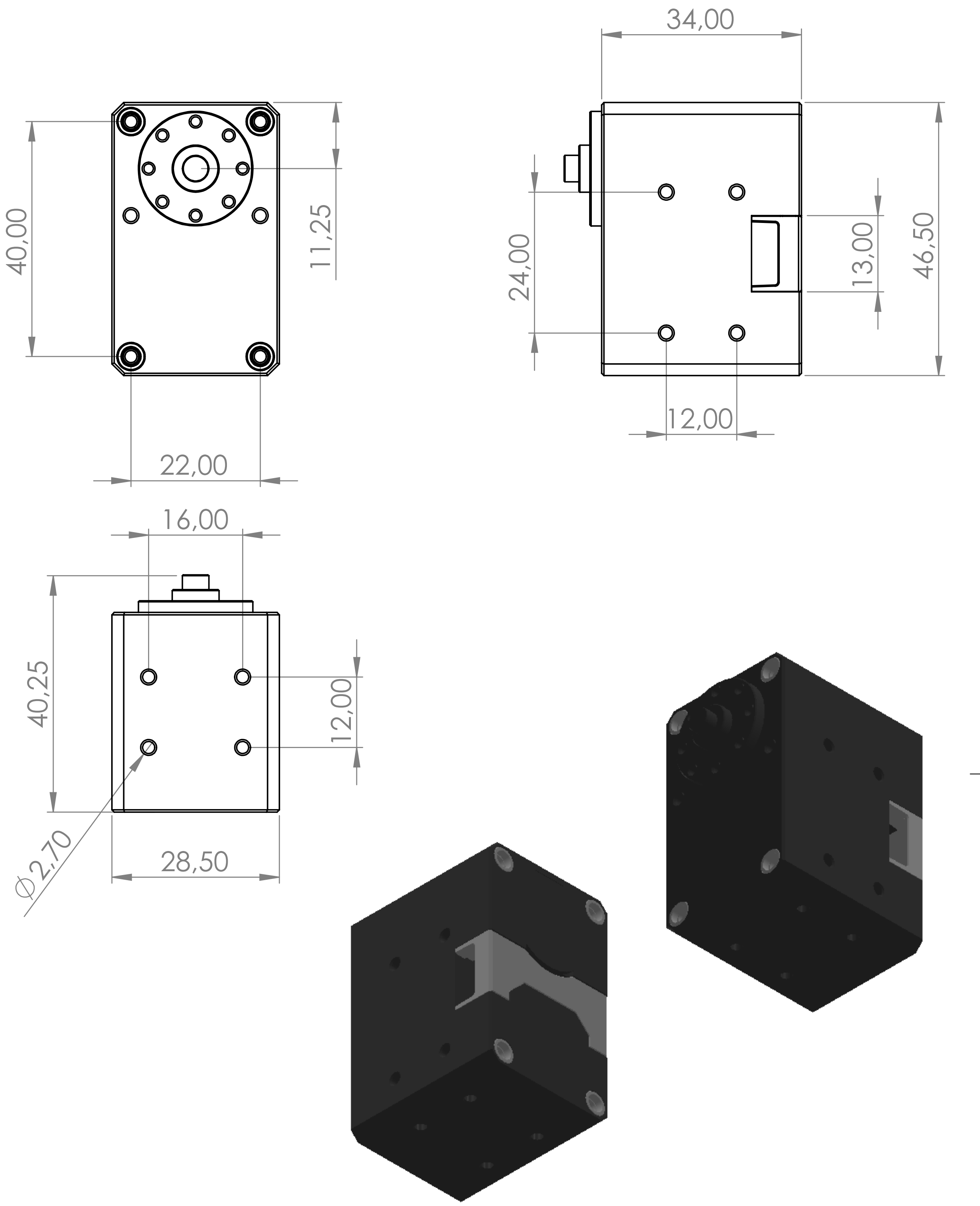
**Gripper 2 Assembly**

Número: 8

Referencia: 008

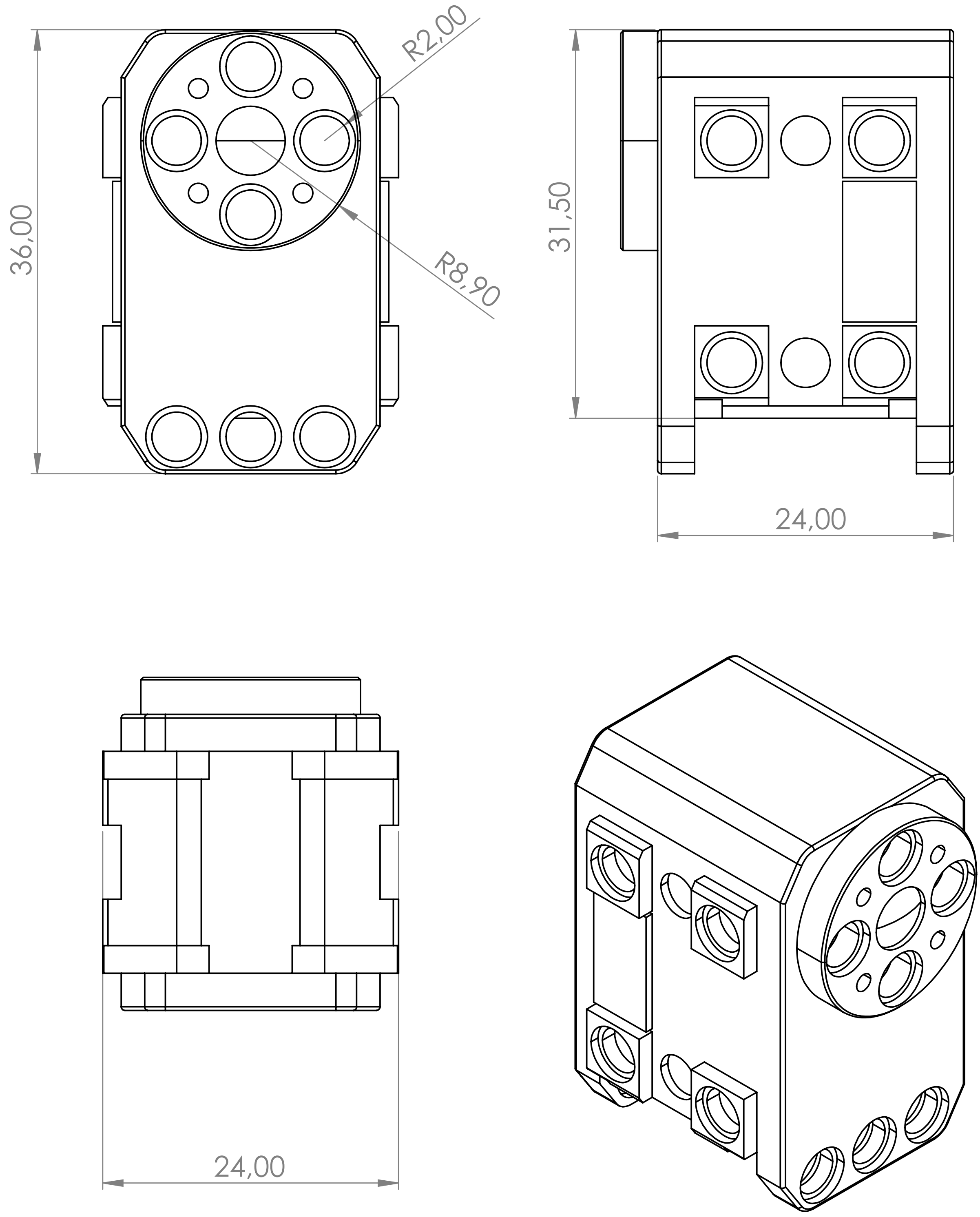
Sustituye a

Sustituido por




| | Fecha | Nombre | | |
|---|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | | **ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL** *Universidad de La Rioja* GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA |
| Comprobado | 20-07-22 | J. Santamaría Peña | | |
| D.S.Normas | U.N.E. | Tolerancia general | | |

| Escala **1:1** | **LOW-COST DIDACTIC ROBOT** | Número: 9 |
|---|---|---|
| | | Referencia: 009 |
| Proyección | **dynamixel XL 430** | Sustituye a |
| | | Sustituido por |




| | Fecha | Nombre |
|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. |
| Comprobado | 20-07-22 | J. Santamaría Peña |
| D.S.Normas | U.N.E. | Tolerancia general |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

Escala **2:1**

Proyección

**LOW-COST DIDACTIC ROBOT**

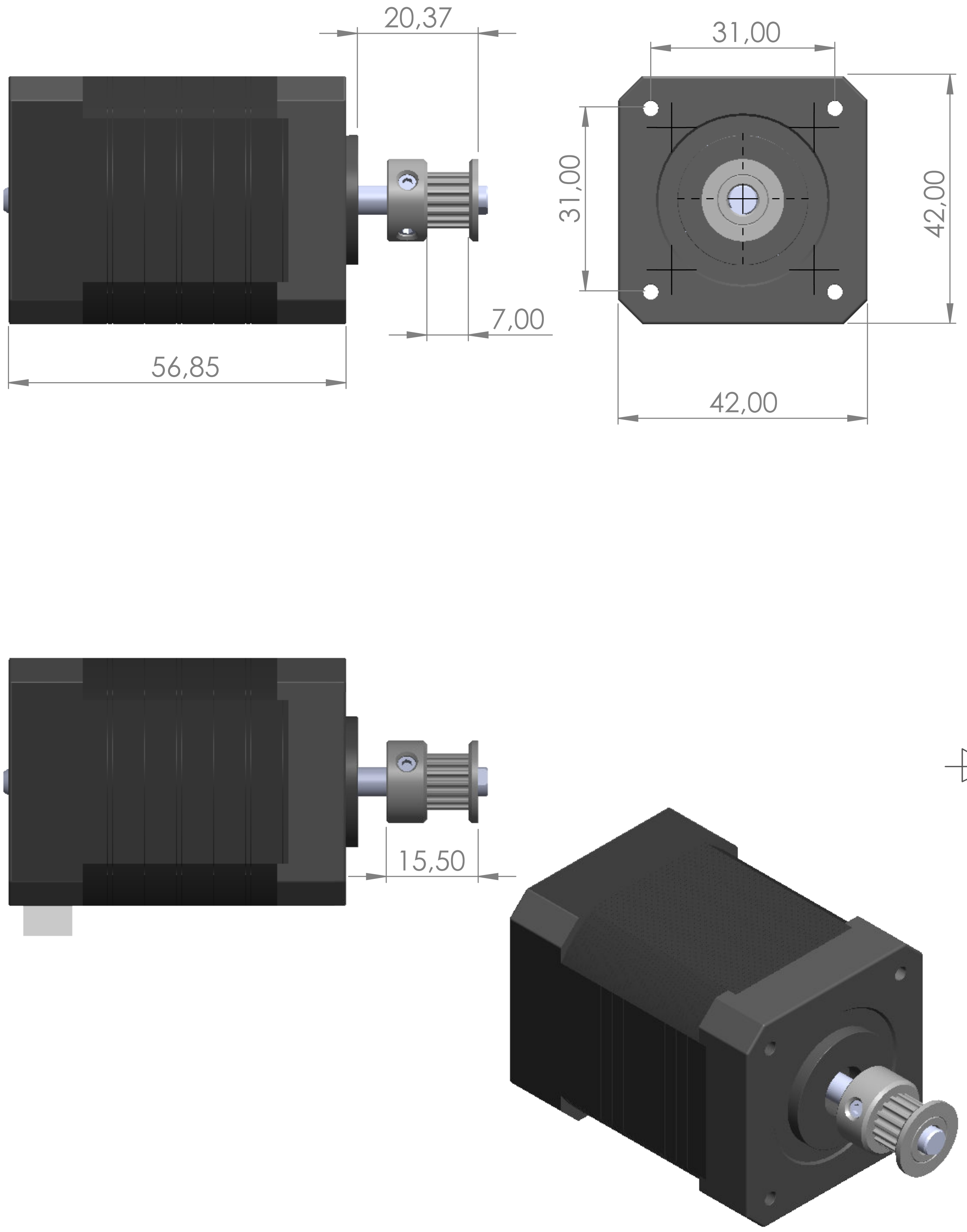
**Dynamixel XL 320**

Número: 10

Referencia: 010

Sustituye a

Sustituido por




| | Fecha | Nombre | |
|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | |
| Comprobado | 20-07-22 | J. Santamaría Peña | |
| D.S.Normas | U.N.E. | Tolerancia general | |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

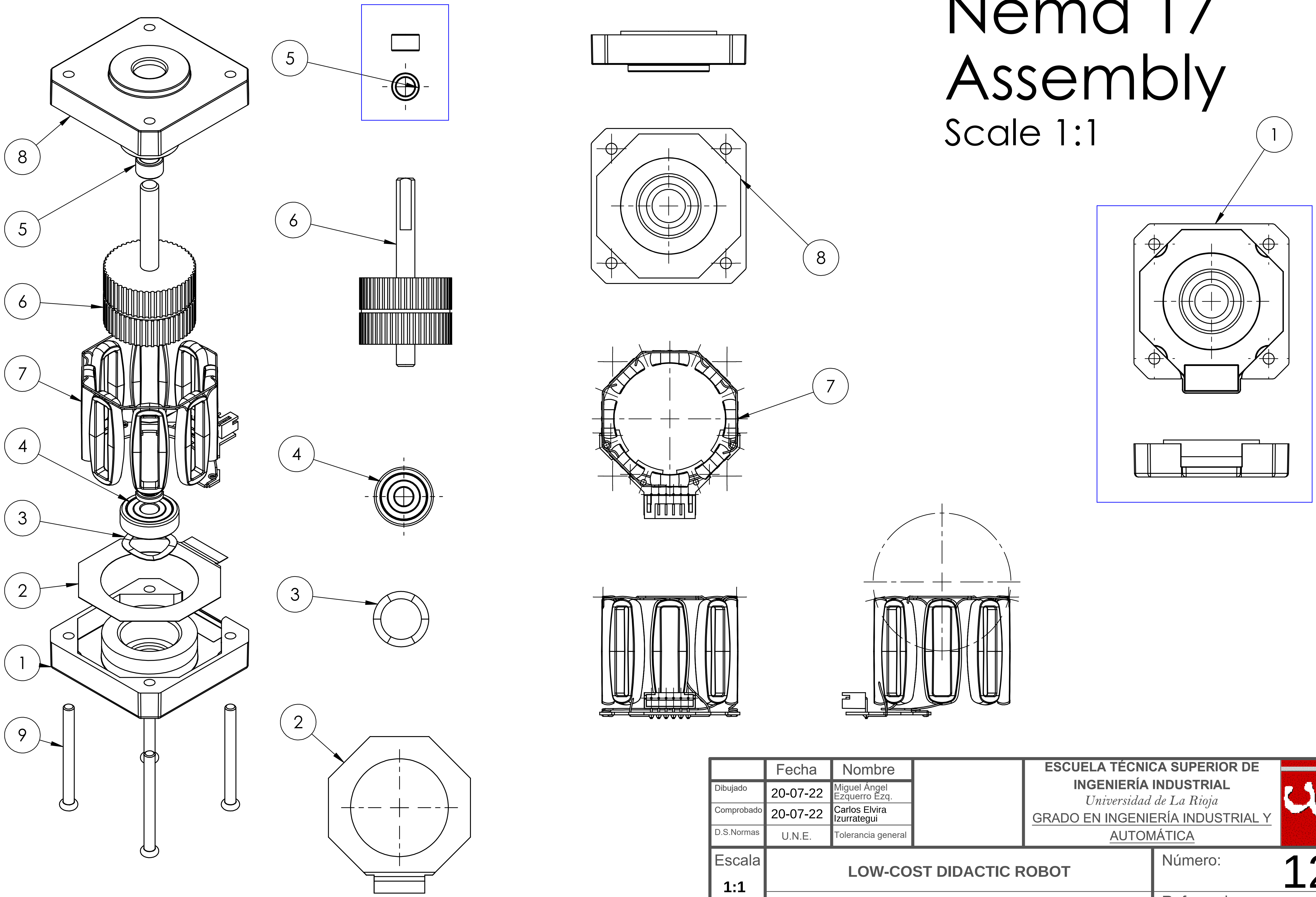
*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

| | | | |
|---|---|---|---|
| Escala **1:2** | **LOW-COST DIDACTIC ROBOT** | Número: | 11 |
| Proyección | **Nema 17 I** | Referencia: | 011 |
| | | Sustituye a | |
| | | Sustituido por | |

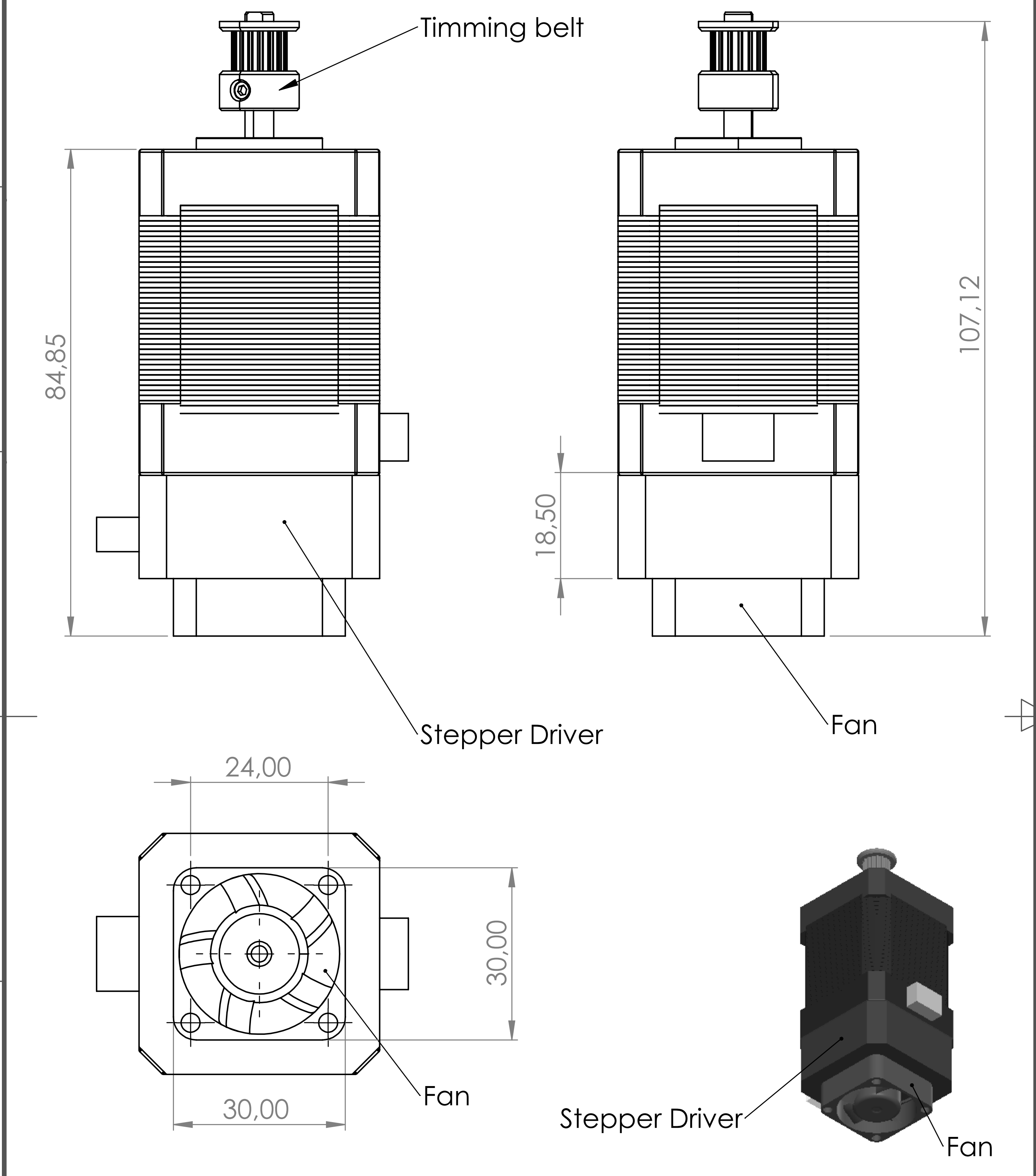
# Nema 17 Assembly

## Scale 1:1



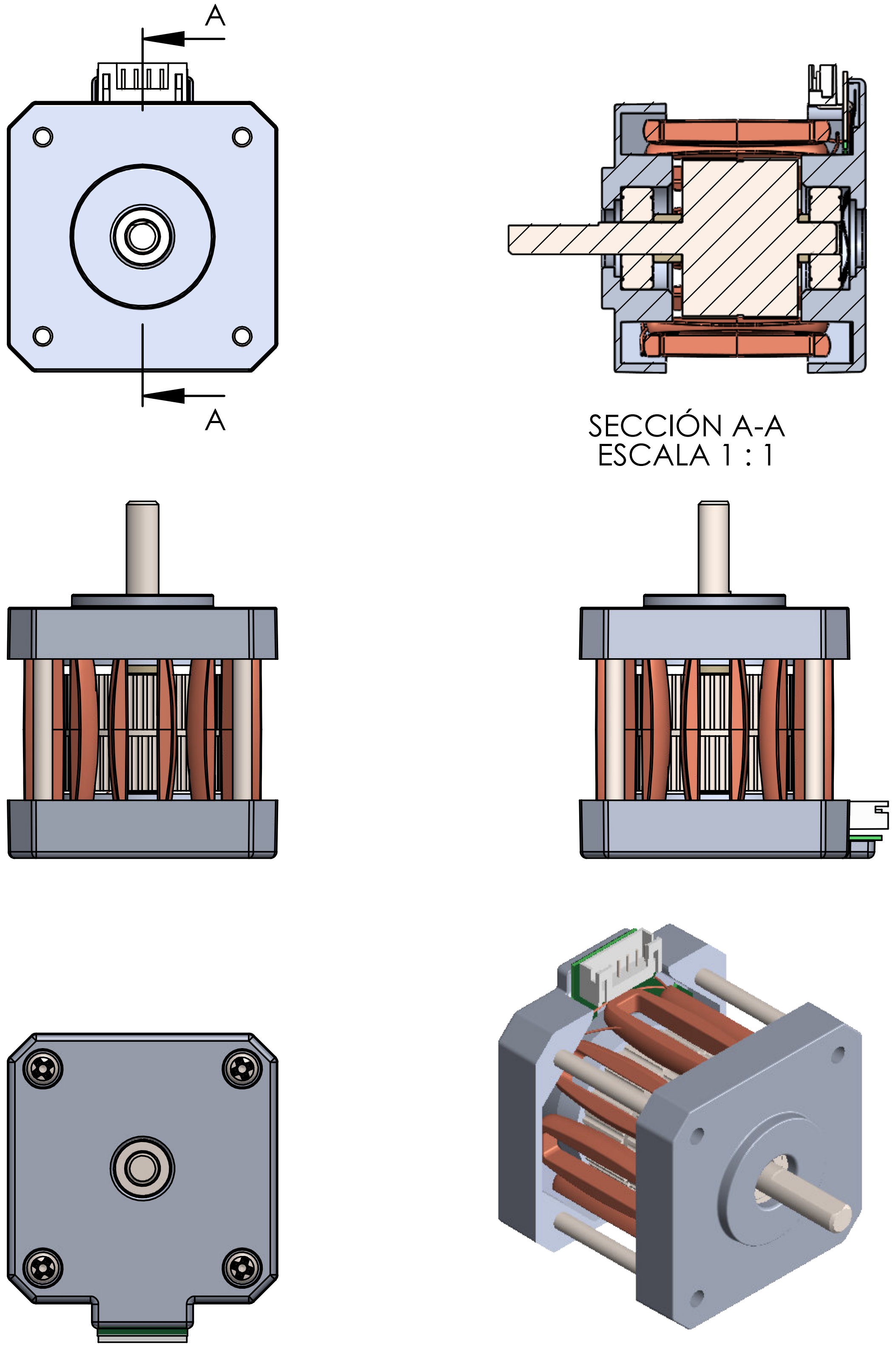

| | Fecha | Nombre | | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL<br>*Universidad de La Rioja*<br>GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA |
|---|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | | |
| Comprobado | 20-07-22 | Carlos Elvira Izurrategui | | |
| D.S.Normas | U.N.E. | Tolerancia general | | |

| Escala 1:1 | LOW-COST DIDACTIC ROBOT | Número: 12 |
|---|---|---|
| | | Referencia: 012 |
| Proyección | Nema 17 II | Sustituye a |
| | | Sustituido por |



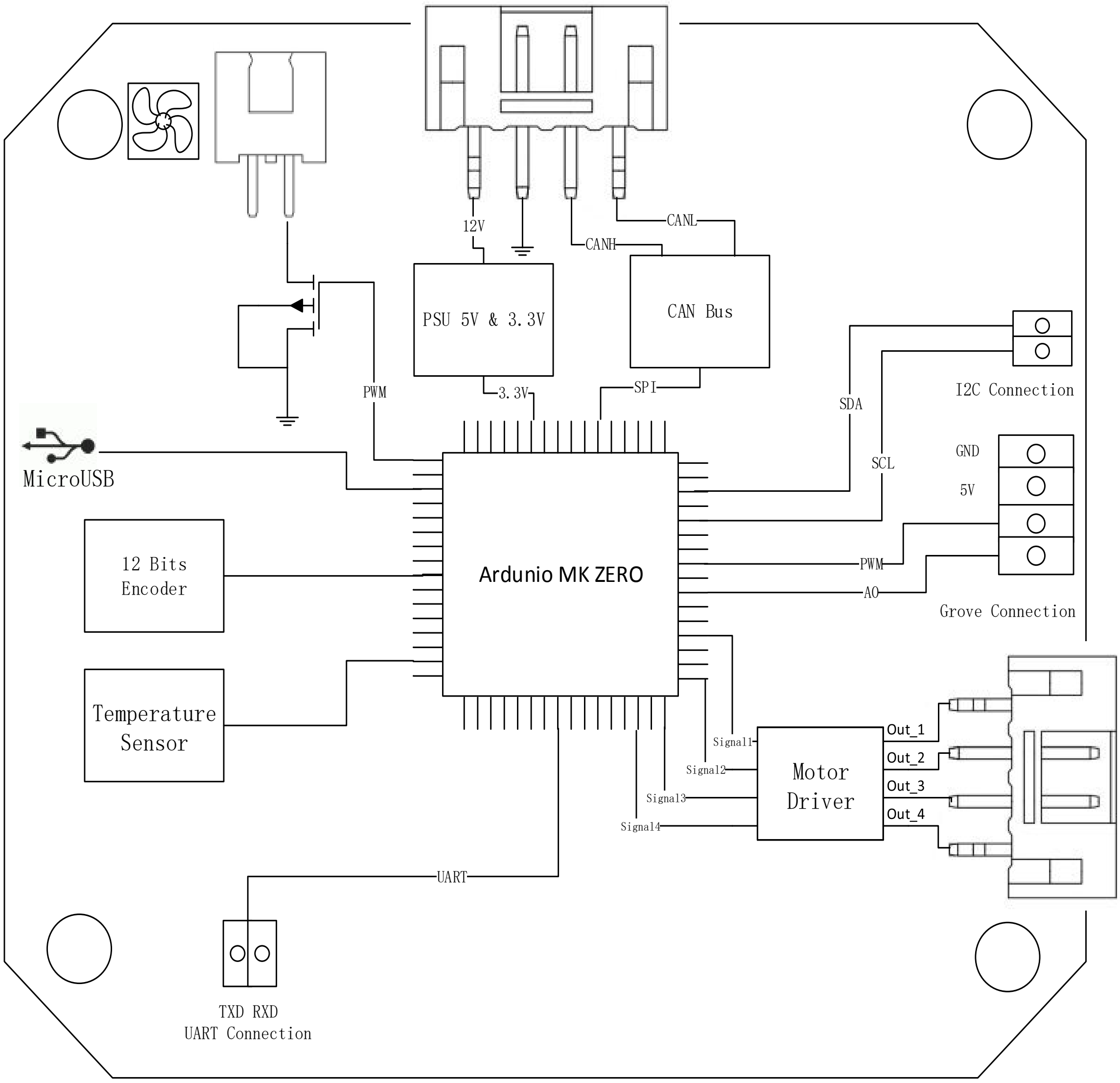

| | Fecha | Nombre | | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL<br>*Universidad de La Rioja*<br>GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA |
|---|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | | |
| Comprobado | 20-07-22 | J. Santamaría Peña | | |
| D.S.Normas | U.N.E. | Tolerancia general | | |

| Escala | **LOW-COST DIDACTIC ROBOT** | Número: | 13 |
|---|---|---|---|
| **1:2** | | Referencia: | 013 |
| Proyección | **Nema 17 III** | Sustituye a | |
| | | Sustituido por | |




| | Fecha | Nombre | | |
|---|---|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. | | **ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL** |
| Comprobado | 20-07-22 | J. Santamaría Peña | | *Universidad de La Rioja* |
| D.S.Normas | U.N.E. | Tolerancia general | | GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA |

| Escala | | |
|---|---|---|
| **1:2** | **LOW-COST DIDACTIC ROBOT** | Número: 14 |
| Proyección | **Nema 17 IV** | Referencia: 014 |
| | | Sustituye a |
| | | Sustituido por |




| | Fecha | Nombre |
|---|---|---|
| Dibujado | 20-07-22 | Miguel Ángel Ezquerro Ezq. |
| Comprobado | 20-07-22 | J. Santamaría Peña |
| D.S.Normas | U.N.E. | Tolerancia general |

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

*Universidad de La Rioja*

GRADO EN INGENIERÍA INDUSTRIAL Y AUTOMÁTICA

Escala **4:1**

Proyección

**LOW-COST DIDACTIC ROBOT**

**Nema 17 Stepper Controller**

Número: 15

Referencia: 015

Sustituye a

Sustituido por

UNIVERSIDAD DE LA RIOJA

*Escuela Técnica Superior de Ingeniería Industrial*

*Grado en Ingeniería Electrónica Industrial y Automática*

# LOW-COST DIDACTIC ROBOT

## **Specifications**

Autor: Miguel Ángel Ezquerro Ezquerro

Tutor: Carlos Elvira Izurrategui

Date: 18 July 2022

# INDEX

# LIST OF ABBREVIATIONS

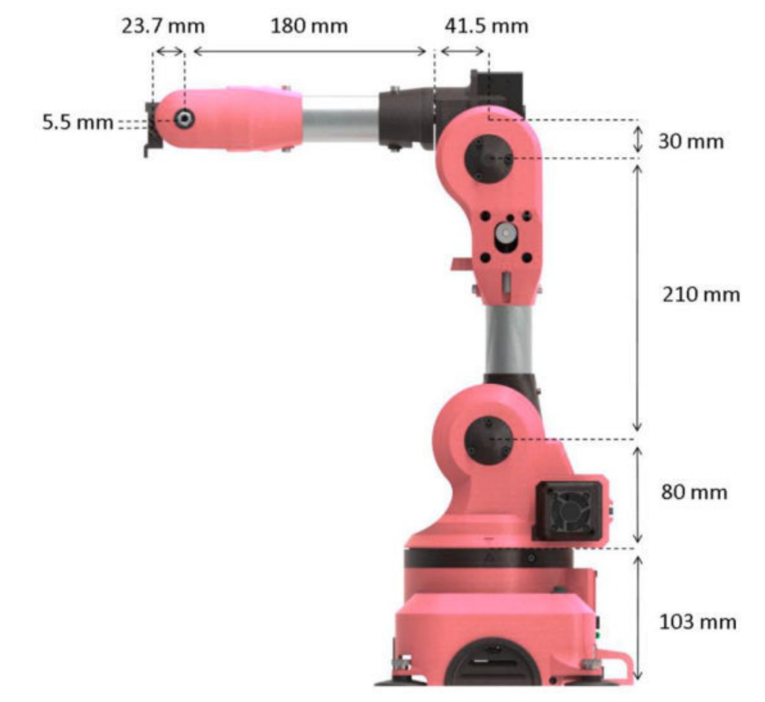
| | |
|---|---|
| 6-DOF | Six-degree-of-freedom |
| EMC | Electromagnetic Compatibility |
| FDM | Fused Deposition Material |
| HAT | Hardware Attached on Top |
| HTM | Homogeneous Transformation Matrices |
| IS | International Standard |
| MCU | Micro Controller Unit |
| PWM | Pulse Width Modulation |
| ROS | Robotics Operating System |
| RS | Reference System |
| RTOS | Real Time Operating System |
| spp. | Steps per second |
| TTL | Transistor-Transistor Logic |

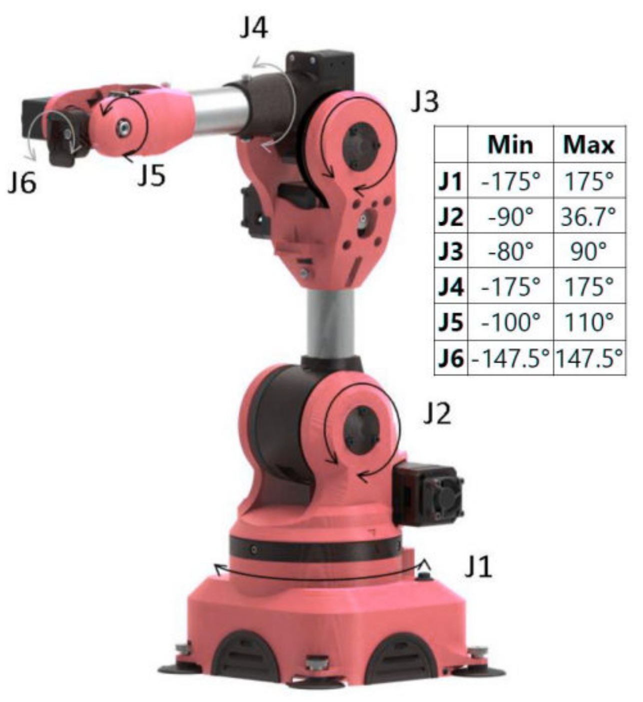
# 1 GENERAL SPECIFICATIONS

This project is only responsible for providing the plans and diagrams necessary for the replicability of the previously discussed Robot. In case of any type of contradictory information throughout this document, what is described in this chapter will prevail.

## 1.1 DIMENSIONS

The robot has a weight of 6 kilograms and a maximum reach of 566 mm when using the Gripper 2. The base joint range is ±175° and has a repeatability of ±1 mm.

<table>
<tr><th>Dimensions</th><th>Max Rotation</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

## 1.2 POWER SUPPLY

The robot needs a power supply capable to provide a current of 6 A. This power supply referring to the voltage must be within the limits described by the manufacturer. For the 11,1 V electronics power supply, a source that can provide the total power demanded; to wit 60 W.

## 1.3 COMPONENTS

For a correct performance of the final assembly, it will be used the materials listed in the Measurements document. The steppers in case of replacement must withstand a minimum of 1,2 A with a maximum tolerance of 1,8 step degrees (200 steps per revolution). The motor dimensions cannot differ by more than 10% with those shown in the blueprints of the Nema 17 stepper.

All measured dimensions of every component can be found in blueprints document.

### 1.3.1 DYNAMIXEL XL 430 SPECIFICATIONS

| Item | Specifications |
|---|---|
| MCU | ARM CORTEX-M3 (72 [MHz], 32Bit) |
| Position Sensor | Contactless absolute encoder (12Bit, 360 [°])<br>Maker : ams(www.ams.com), Part No : AS5601 |
| Motor | Cored |
| Baud Rate | 9,600 [bps] ~ 4.5 [Mbps] |
| Control Algorithm | PID control |
| Resolution | 4096 [pulse/rev] |
| Operating Modes | Velocity Control Mode<br>Position Control Mode (0 ~ 360 [°])<br>Extended Position Control Mode (Multi-turn)<br>PWM Control Mode (Voltage Control Mode) |
| Weight | 57.2 [g] |
| Dimensions (W x H x D) | 28.5 x 46.5 x 34 [mm] |
| Gear Ratio | 258.5 : 1 |
| Stall Torque | 1.0 [N.m] (at 9.0 [V], 1.0 [A])<br>**1.4 [N.m] (at 11.1 [V], 1.3 [A])**<br>1.5 [N.m] (at 12.0 [V], 1.4 [A]) |
| No Load Speed | 47 [rev/min] (at 9.0 [V])<br>**57 [rev/min] (at 11.1 [V])**<br>61 [rev/min] (at 12.0 [V]) |
| Operating Temperature | -5 ~ +72 [°C] |
| Input Voltage | 6.5 ~ 12.0 [V] (**Recommended : 11.1 [V]**) |
| Command Signal | Digital Packet |

| | |
|---|---|
| Physical Connection | TTL Multidrop Bus (5V Logic)<br>TTL Half Duplex Asynchronous Serial Communication<br>(8bit, 1stop, No Parity) |
| ID | 253 ID (0 ~ 252) |
| Feedback | Position, Velocity, Load, Realtime tick, Trajectory, Temperature, Input Voltage, etc |
| Case Material | Engineering Plastic |
| Gear Material | Engineering Plastic |
| Standby Current | 52 [mA] |

### 1.3.1 DYNAMIXEL XL 320 SPECIFICATIONS

| Item | Specifications |
|---|---|
| Motor | Cored |
| Baud Rate | 7343 [bps] ~ 1 [Mbps] |
| Resolution | 0.29 [°] |
| Operating Modes | Joint Mode (0 ~ 300 [°])<br>Wheel Mode (Endless Turn) |
| Weight | 16.7 [g] |
| Dimensions (W x H x D) | 24 x 36 x 27 [mm] |
| Gear Ratio | 9,91736111 |
| Stall Torque | 0.39 [N.m] (at 7.4 [V], 1.1 [A]) |
| No Load Speed | 114 [rev/min] (at 7.4 [V], 0.18 [A]) |
| Operating Temperature | -5 ~ +65 [°C] |
| Input Voltage | 6 ~ 8.4 [V] (**Recommended : 7.4 [V]**) |
| Command Signal | Digital Packet |
| Protocol Type | TTL Half Duplex Asynchronous Serial Communication with 8bit, 1stop, No Parity |
| Physical Connection | TTL Multidrop Bus |
| ID | 253 ID (0 ~ 252) |
| Feedback | Position, Temperature, Load, Input Voltage, etc |
| Case Material | Engineering Plastic |
| Gear Material | Engineering Plastic |

## 1.4 REFRIGERATION

The cooling system (with the improvements) must be able to maintain a stable temperature within the range supported by the components. This is the table of temperature rates in different components:

| Device | Minimum | Maximum |
|---|---|---|
| Stepper | -10º C | 40º C |
| Dynamixel | -5 ºC | 75 ºC |
| Raspberry Pi | 0 ºC | 85 ºC |

## 1.5 ELECTROMAGNETIC NOISE

The presence of electromagnetic noise must be avoided as much as possible. Avoiding placing the robot and their electronics near potential sources of strong electromagnetic fields. Requiring the addition of electromagnetic isolation methods if These measures were not enough. Worth noting that the inner Dynamixel TTL bus and the CAN bus for the steppers are quite resilient to electromagnetic noise according to the electromagnetic compatibility (EMC) Directive 2014/30/EU.

# 2 PRICE

The prices indicated in the budget document are only indicative and are subject to change. Buy links are provided to see the real price at each moment of consulting.

# 3 ELECTRONICS

- Raspberry Pi 3B
- Niryo Pannel
- Niryo Shield
- Stepper Drivers

The correct operation of the PCB is guaranteed under the conditions facilitated by the manufacturer.

# 4 SOFTWARE

All included software will be functional as long as it is used under the conditions described in the Memory document.

| NAME | DESCRIPTION | PURVEYOR | LICENSE |
|---|---|---|---|
| **MATLAB** | Programming platform | MathWorks | Educational License |
| **ROS** | Robotics Operating System | Free Software Foundation, Inc. | GNU General Public License v3.0 |
| **NIRYO ONE STUDIO** | Graphical HMI | Niryo | Freeware |
| **STEPPER FIRMWARE** | Stepper control frimware | Niryo | Freeware |
| **ARDUINO IDE** | Programming platform | Arduino | Freeware |

# 5 3D PRINTS

These are the printing specifications of the seven prints.

### 5.1.1 PRINT 1

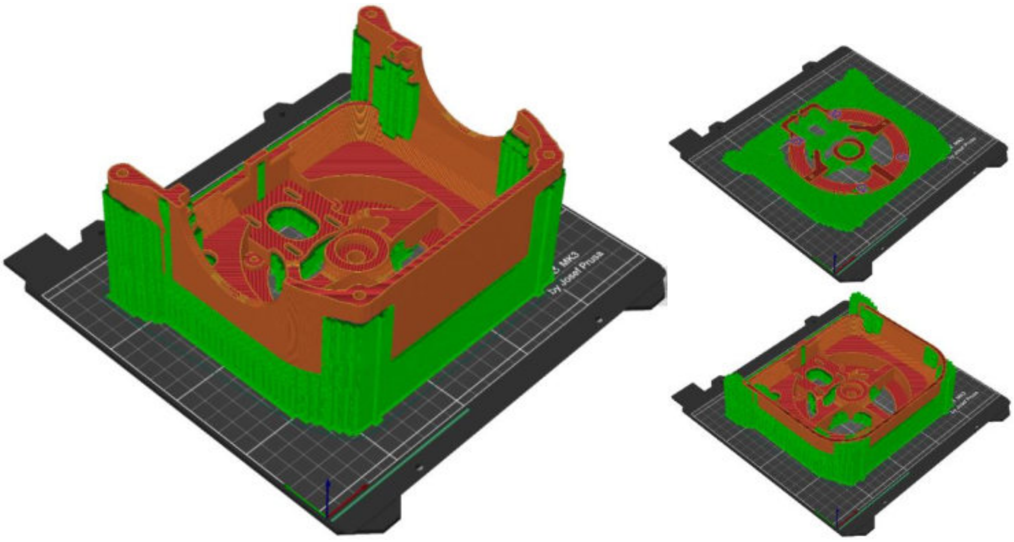


*Illustration 1: Print 1*

| | | | |
|---|---|---|---|
| Pieces: Base | Color: Red | Weight: 380 g | Time: 25 h |

### 5.1.2 PRINT 2

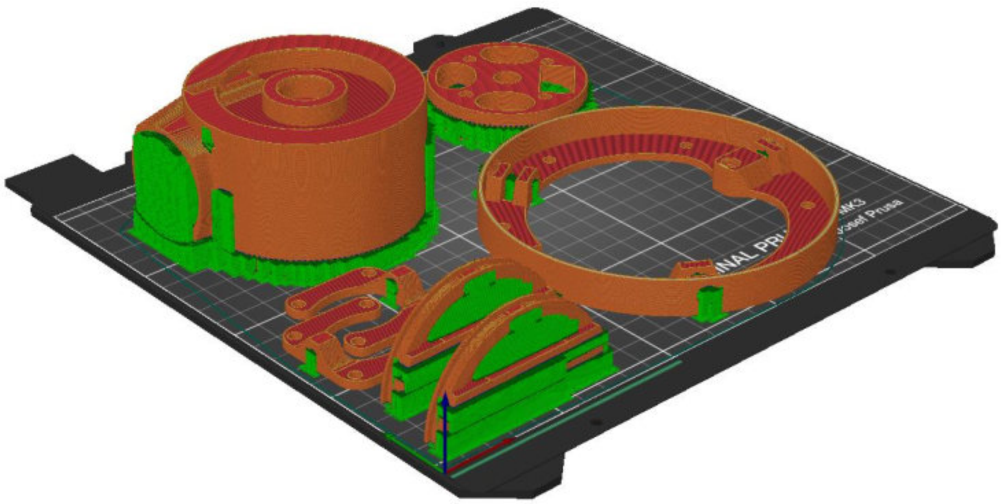


*Illustration 2: Print 2*

| | | | |
|---|---|---|---|
| Shoulder gearing, Belt tensioner, Base gearing support, Arm Bottom | Color: Grey | Weight: 424 g | Time: 28 h |

### 5.1.3 PRINT 3

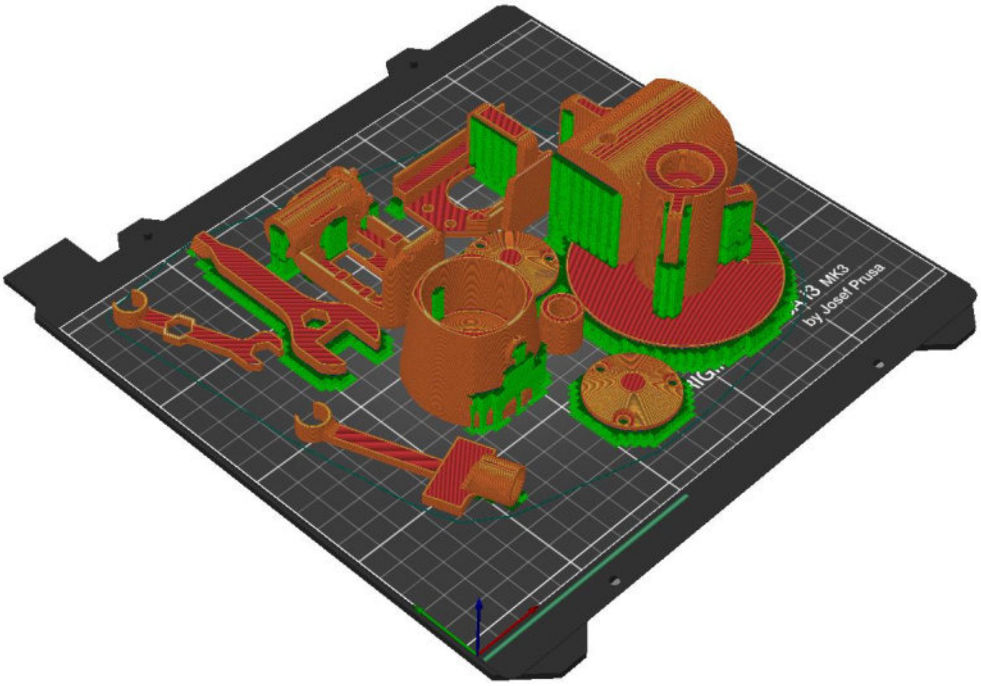



*Illustration 3: Print 3*

| | | | |
|---|---|---|---|
| Shoulder, Forearm Bottom, End Cap x2, Assembly tools, Grip 1 base, Wrist | Color: Grey | Weight: 200 g | Time: 19 h |

### 5.1.4 PRINT 4

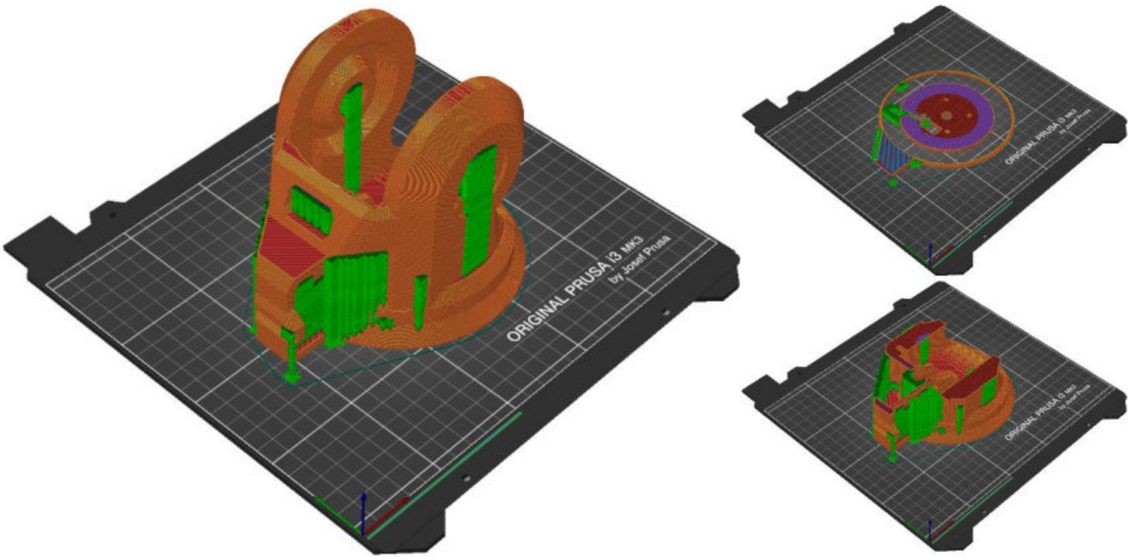



*Illustration 4: Print 4*

| | | | |
|---|---|---|---|
| Shoulder | Color: Red | Weight: 436 g | Time: 24 h |

### 5.1.5 PRINT 5

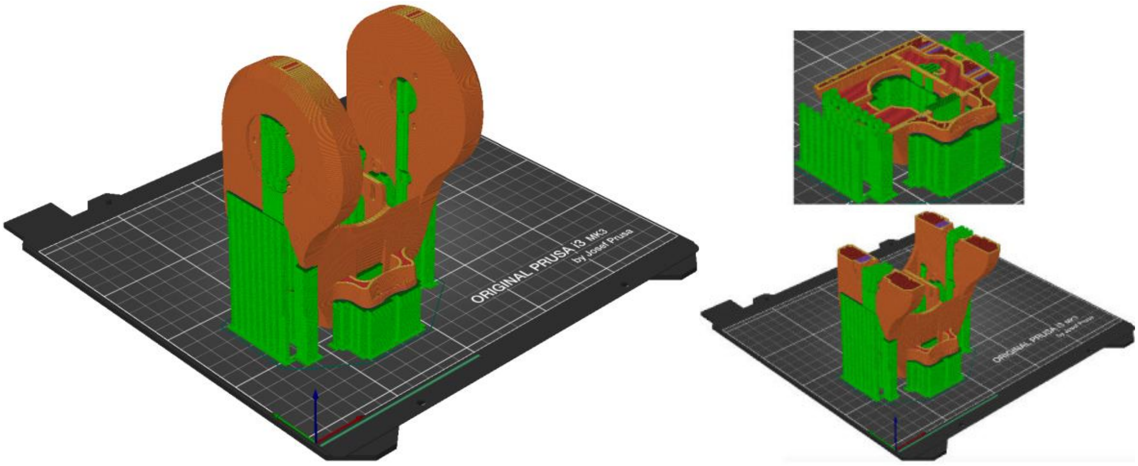



*Illustration 5: Print 5*

| | | | |
|---|---|---|---|
| Arm Top | Color: Red | Weight: 200 g | Time: 19 h |

### 5.1.6 PRINT 6

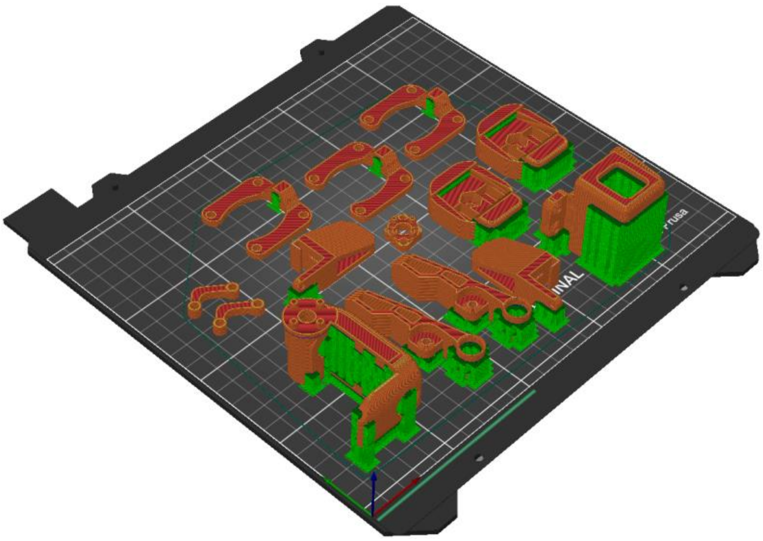


*Illustration 6: Print 6*

| | | | |
|---|---|---|---|
| Belt tensioner, Grip 2 clams, Grip 1 clamps, Grip 2 base, Grip 1 rods, Tool Support x2 | Color: Red | Weight: 129 g | Time: 14 h |

### 5.1.7 PRINT 7

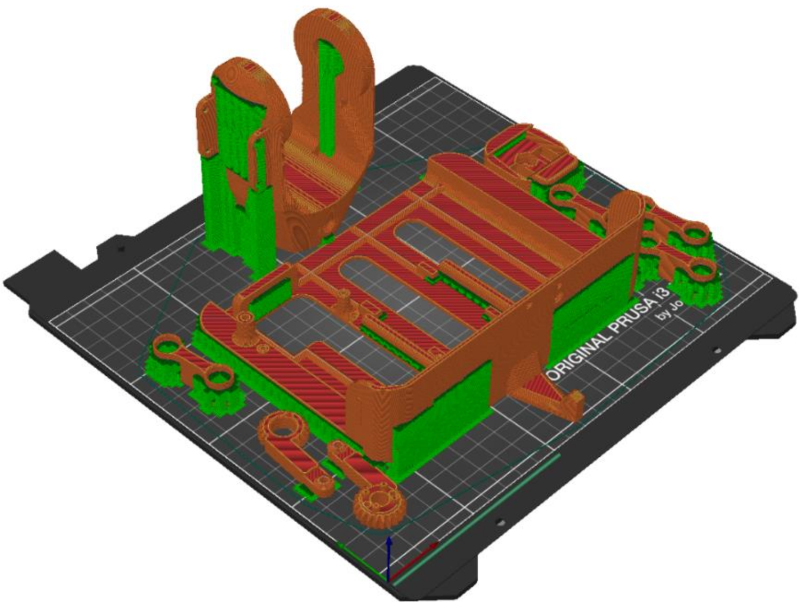



*Illustration 7: Print 7*

| | | | |
|---|---|---|---|
| Forearm Top, Drawner, Grip 2 gearings and rods, Tool Support | Color: Red | Weight: 203 g | Time: 21 h |

UNIVERSIDAD DE LA RIOJA

*Escuela Técnica Superior de Ingeniería Industrial*

*Grado en Ingeniería Electrónica Industrial y Automática*

# LOW-COST DIDACTIC ROBOT

## **Measuring**

Autor: Miguel Ángel Ezquerro Ezquerro

Tutor: Carlos Elvira Izurrategui

Date: 18 July 202

| Ref | | Description | Provider | Marca/model | ud. |
|---|---|---|---|---|---|
| | **3D Printing** | | | | |
| 001 | Print 1 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 380 g |
| 002 | Print 2 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-DGR-1KG | 424 g |
| 003 | Print 3 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-DGR-1KG | 200 g |
| 004 | Print 4 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 436 g |
| 005 | Print 5 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 200 g |
| 006 | Print 6 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 129 g |
| 007 | Print 7 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 203 g |
| | **Actuators** | | | | |
| 008 | Nema 17 | Stepper | Longruner | LQD03-UK | 3 |
| 009 | Dynamixel XL 430 | Servomotor | ROBOTIS | XL 430-W250-T | 2 |
| 010 | Dynamixel XL 320 | Servomotor | ROBOTIS | XL 320 DARWIN-mini | 1 |
| | **Electronics** | | | | |
| 011 | Raspberry Pi 3B | ROS controller | Raspberry Pi Spain | Raspberry Pi 3 Model B+ | 1 |
| 012 | Niryo Pannel | I/O Pannel | Niryo | Specific PCB | 1 |
| 013 | Niryo Shield | Raspberry HAT | Niryo | Specific PCB | 1 |
| 014 | Stepper Drivers | Psition reader stepper | Niryo | Specific PCB | 1 |
| 015 | Switch | SPST contact | RS | 419-794 | 1 |
| | **Software Liscenses** | | | | |
| 016 | SOLDIWORKS | CAD Design software | Dassault Systémes | Educational License | 1 |
| 017 | Prusa Slicer | Slicing software | Prusa | Freeware | 1 |
| 018 | Matlab | Programming platform | MathWorks | Educational License | 1 |
| 019 | ROS | Robotics Operating System | Free Software Foundation, Inc. | GNU General Public License v3.0 | 1 |
| 020 | Niryo One Studio | Graphical HMI | Niryo | Freeware | 1 |
| 021 | Niryo Stepper (motors firmware) | Stepper control frimware | Niryo | Freeware | 1 |
| 022 | Arduino IDE | Programming platform | Arduino | Freeware | 1 |
| | **Miscellany** | | | | |
| 023 | Screws | Diverse Metric Screws | Mayoral | Various | 1 |
| 024 | Wires | Cables for CANBus and TTL | Electónica de Luis | Various | 1 |
| 025 | Connectors | Pligs and connectors | Electónica de Luis | Various | 7 |

UNIVERSIDAD DE LA RIOJA

*Escuela Técnica Superior de Ingeniería Industrial*

*Grado en Ingeniería Electrónica Industrial y Automática*

# LOW-COST DIDACTIC ROBOT

## <u>**Budget**</u>

Autor: Miguel Ángel Ezquerro Ezquerro

Tutor: Carlos Elvira Izurrategui

Date: 20 July 2022

| *Ref* | *Item* | *Description* | *Provider* | *Marca/model* | *€/und* | *ud.* | *€* |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | **3D Printing** | | | | | | |
| *001* | Print 1 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 0,02 € | 380 g | 8,93 € |
| *002* | Print 2 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-DGR-1KG | 0,02 € | 424 g | 9,96 € |
| *003* | Print 3 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-DGR-1KG | 0,02 € | 200 g | 4,70 € |
| *004* | Print 4 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 0,02 € | 436 g | 10,25 € |
| *005* | Print 5 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 0,02 € | 200 g | 4,70 € |
| *006* | Print 6 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 0,02 € | 129 g | 3,03 € |
| *007* | Print 7 | Plastic components | Eolas Prints | SKU: PLA-PRM-1.75M-RED-1KG | 0,02 € | 203 g | 4,77 € |
| *008* | Amortization | 3D Printer amortization | Prusa | Prusa i3 MK3S+ | 8,50 € | 7 | 59,50 € |
| *009* | Electricity | Power supply | UR | ~ | 0,34 € | 12 | 4,04 € |
| | **Actuators** | | | | | | |
| *010* | Nema 17 | Stepper | Longruner | LQD03-UK | 15,00 € | 3 | 45,00 € |
| *011* | Dynamixel XL 430 | Servomotor | ROBOTIS | XL 430-W250-T | 50,00 € | 2 | 100,00 € |
| *012* | Dynamixel XL 320 | Servomotor | ROBOTIS | XL 320 DARWIN-mini | 35,00 € | 1 | 35,00 € |
| | **Electronics** | | | | | | |
| *013* | Raspberry Pi 3B | ROS controller | Raspberry Pi Spain | Raspberry Pi 3 Model B+ | 206,00 € | 1 | 206,00 € |
| *014* | Niryo Pannel | I/O Pannel | Niryo | Specific PCB | 20,00 €* | 1 | 20,00 € |
| *015* | Niryo Shield | Raspberry HAT | Niryo | Specific PCB | 30,00 €* | 1 | 30,00 € |
| *016* | Stepper Drivers | Psition reader stepper | Niryo | Specific PCB | 15,00 €* | 3 | 45,00 € |
| *017* | Switch | SPST contact | RS | 419-794 | 1,85 € | 1 | 1,85 € |
| | **Software Liscenses** | | | | | | |
| *018* | SOLDIWORKS | CAD Design software | Dassault Systémes | Educational License | 60,00 € | 1 | 60,00 € |
| *019* | Prusa Slicer | Slicing software | Prusa | Freeware | 0,00 € | 1 | 0,00 € |
| *020* | Matlab | Programming platform | MathWorks | Educational License | 0,00 € | 1 | 0,00 € |
| *021* | ROS | Robotics Operating System | Free Software Foundation, Inc. | GNU General Public License v3.0 | 0,00 € | 1 | 0,00 € |
| *022* | Niryo One Studio | Graphical HMI | Niryo | Freeware | 0,00 € | 1 | 0,00 € |

| Ref | Item | Description | Provider | Marca/model | €/und | ud. | € |
|---|---|---|---|---|---|---|---|
| 023 | Stepper Firmware | Stepper control frimware | Niryo | Freeware | 0,00 € | 1 | 0,00 € |
| 024 | Arduino IDE | Programming platform | Arduino | Freeware | 0,00 € | 1 | 0,00 € |
| | **Miscellany** | | | | | | |
| 025 | Screws | Diverse Metric Screws | Mayoral | Various | 20,00 € | 1 | 20,00 € |
| 026 | Wires | Cables for CANBus and TTL | Electónica de Luis | Various | 8,00 € | 1 | 8,00 € |
| 027 | Connectors | Pligs and connectors | Electónica de Luis | Various | 0,50 € | 7 | 3,50 € |
| | **Manpower** | | | | | | |
| 028 | Design | Original Design Modifications | | | 20,00 € | 20 h | 400,00 € |
| 029 | Programming | Movement squences and communication | | | 20,00 € | 15 h | 300,00 € |
| 030 | Assembly | Build the robot | | | 12,00 € | 40 h | 480,00 € |
| 031 | Verifying | Set communication and configurations | | | 10,00 € | 10 h | 100,00 € |
| 032 | Testing | Test the robot | | | 10,00 € | 150 h | 1500,00 € |
| | | | | | | **Material:** | **684,23 €** |
| | | | | | | **TOTAL:** | **1.864,23 €** |

THE BUDGET AMOUNTS TO ONE THOUSAND EIGHT HUNDRED FOURTY-FOUR EUROS AND TWENTY-THREE CENTS.

** - The assigned value is illustrative. These components are included in the Niryo One pack (which is no longer available) and its price is not specified.*