Kaunas University of Technology

Faculty of Electrical and Electronics Engineering

Technical University of Cartagena

School of Industrial Engineering

Use of a 6-axis Robotic Arm for Covid-19 Vaccination

Bachelor's Final Degree Project

Fulgencio Blázquez Conesa

Project author

Assoc. Prof. Dr. Miguel Almonacid Kroeger
Asso. Prof. Dr. Virginijus Baranauskas

Supervisors

Cartagena, Kaunas 2021

Kaunas University of Technology

Faculty of Electrical and Electronics Engineering

Technical University of Cartagena

School of Industrial Engineering

Use of a 6-axis Robotic Arm for Covid-19 Vaccination

Bachelor's Final Double Project

Intelligent Robotics Systems (6121EX013)

Industrial Electronics and Automation Engineering (5071)

Fulgencio Blázquez Conesa

Project author

Assoc. Prof. Dr. Miguel Almonacid Kroeger
Assoc. Prof. Dr. Virginijus Baranauskas

Supervisors

Lect Dr. Kestas Rimkus

Reviewer

Cartagena, Kaunas 2021

Kaunas University of Technology

Faculty of Electrical and Electronics Engineering

Technical University of Cartagena

School of Industrial Engineering

Fulgencio Blazquez Conesa

Use of a 6-axis Robotic Arm for Covid-19 Vaccination

Declaration of Academic Integrity

I confirm the following:

1. I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;

2. All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;

3. I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;

4. I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Fulgencio Blázquez Conesa

Confirmed electronically

APPROVED BY:

KTU Faculty of Electrical and Electronics Engineering

Head of the Department of Automation

Assoc. prof. dr. Gintaras Dervinis

2021 05 04

TASK OF FINAL PROJECT OF UNDERGRADUATE (BACHELOR) STUDIES

Issued to the Student: *Fulgencio Blázquez Conesa* Group ERB7/2

1. Project Subject:

Lithuanian Language:	6 ašių robotinės rankos panaudojimas Covid-19 vakcinacijai
English Language:	Use of a 6-axis Robotic Arm for Covid-19 Vaccination

Approved 2021 May . 6d. Decree of Dean Nr. *V25-03-8*

2. Goal of the Project: To develop the 6-axis robotic arm for Covid-19 vaccination

3. Specification of Final Project: The work must meet the methodological requirements for the preparation of final projects of KTU Faculty of Electrical and Electronics Engineering.

4. Project's Structure. *The content is concretized together with supervisor, considering the format of the final project, which is listed in 14 and 15 points of Combined Description of Preparation , Defence and Keeping of Final Projects Methodical Requirements*

4.1 Analyse structure of the robot and its usage in medical applications
4.2 Create control algorithm of the 6-axis robotic arm for Covid-19 vaccination and make its simulation
4.3 Assemble 6-axis robotic arm prototype and make accuracy testing

5. Economical Part. *If economical substantiation is needed; content and scope is concretized together with supervisor during preparation of final projects*

none

6. Graphic Part. *If necessary, the following schemes, algorithms and assembly drawings; content and scope is concretized together with supervisor during preparation of final projects*

Show component assembling schemes and explain them

5. This Task is Integral Part of Final Project of Undergraduate (Bachelor) Studies

6. The Term of Final Project Submission to Defense Work at a Public Session of Qualification Commission. *until 2021-05-11*
(date)

I received this task: ***Fulgencio Blázquez Conesa*** *2021-03-02*
(student's name, surname, signature) *(date)*

Supervisors: **Assoc. Prof. Dr. Miguel Almonacid Kroeger**
Assoc. Prof. Dr. Virginijus Baranauskas *2021-03-02*
(position, name, surname, signature) *(date)*

Blázquez Conesa, Fulgencio. Use of a 6-axis Robotic Arm for Covid-19 Vaccination. Bachelor's Final Degree Project / supervisor: Assoc. Prof. Dr. Virginijus Baranauskas. Assoc. Prof. Dr. Miguel Almonacid Kroeger; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering; Technical University of Cartagena, School of Industrial Engineering

Study field and area (study field group): electronics and electrical engineering, technological science, engineering.

Keywords: robotic arm, vaccination process, 3D print.

Kaunas, Cartagena 2021. p. 69

Summary

The present project aims to automate the covid-19 vaccination process using a 6-degree-of-freedom robotic arm. For the development of this, an electric clamp has been designed to handle the syringes. The automation of the vaccination process will be carried out through simulation. And the second part of the project is the construction of a robotic arm with 6 degrees of freedom printed on a 3D printer and assembled at the Kaunas University of Technology.

Fulgencio Blázquez Conesa. Uso de un brazo robótico de 6 ejes para la vacunación del Covid-19. Trabajo de Fin de Grado / supervisores: Assoc. Prof. Dr. Miguel Almonacid Kroeger, Assoc. Prof. Dr. Virginijus Baranauskas; Universidad Tecnológica de Kaunas, Facultad de Ingeniería Eléctrica y Electrónica; Universidad Politécnica de Cartagena, Facultad de Ingeniería Industrial.

Campo de estudio y área : Ingeniería electrónica y eléctrica, ciencias de tecnología, ingeniería.

Palabras clave: brazo robótico, proceso de vacunación, impresión 3D.

Kaunas, Cartagena 2021. p. 69

Sumario

El presente Proyecto tiene como objetivo automatizar el proceso de vacunación del covid-19 utilizando para ello un brazo robótico de 6 grados de libertad. Para el desarrollo de este se ha diseñado una pinza eléctrica para la manipulación de las jeringuillas. La automatización del proceso de vacunación se llevará acabo a través de simulación. Y la segunda parte del proyecto es la construcción de un brazo robótico de 6 grados de libertad impreso en impresora 3D y montado en la Universidad Tecnológica de Kaunas.

Blázquez Conesa, Fulgencio. 6 ašių robotinės rankos panaudojimas Covid-19 vakcinacijai. Bakalauro baigiamasis projektas / vadovai / doc. dr. VIrginijus Baranauskas, doc. dr. Miguel Almonacid Kroeger; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Universidad Politécnica de Cartagena, Pramonės inžinerijos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): elektronikos ir elektros inžinerija, technologijos mokslai, inžinerija.

Reikšminiai žodžiai: robotinė ranka, vakcinacija, 3D spausdinimas..

Kaunas, Cartagena 2021. p. 69

Santrauka

Šiuo projektu siekiama automatizuoti Covid-19 vakcinacijos procesą naudojant 6 laisvės laipsnių robotinę ranką. Projekte sukurtas elektrinis spaustukas/griebtuvas švirkštams laikyti. Vakcinacijos proceso automatizavimas bus atliekamas imituojant šį procesą. Antroje projekto dalyje aprašoma - roboto rankos su 6 laisvės laipsniais, atspausdinta 3D spausdintuve ir sumontuota Kauno technologijos universitete, konstrukcija.

Table of contents

List of figures

List of tables

Introduction

The society are currently experiencing an unprecedented situation which has called upon the dedication of an excessive number of health personnel, researchers, and other jobs directly related to covid-19. Due to the rapid development of vaccines and the high number of infections, the demand for health personnel is significantly greater than the current supply, since many nurses are needed to vaccinate the entire population. And in many cases hospitals are saturated. If the vaccination process could be automated, this would free up health personnel to take care of all the patients who need it. Furthermore, most vaccines require more than one dose to be effective, with each person having to be vaccinated at least twice. As can be seen in Figure 1, herd immunity, or 70% of the population fully vaccinated, is expected to be achieved between September and October 2021 [1]. To meet this objective and be able to return to normality, the vaccination rate must be increased. One effective way of improving the efficiency of the process would be to automate the vaccination process.

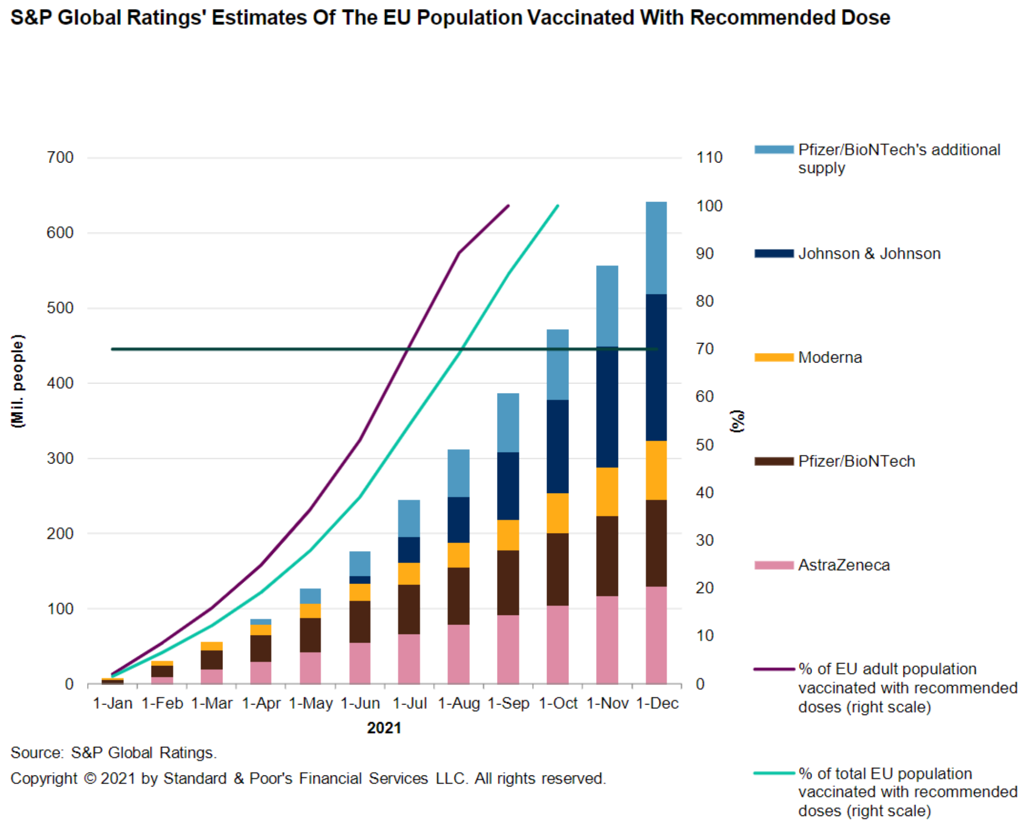


Fig. 1. S&P Ratings‘ Estimates Of The EU Population Vaccinated With Recommended Dose

This whole idea came up after assembling a 6-degree-of-freedom robotic arm at Kaunas University of Technology. Once the arm was finished, it was time to decide what it could be used for. At that time, it was thought that the best idea was to use it to give vaccines, since it is a repetitive process, and it is not very complicated to carry out. Due to the end of the stay at the Kaunas University of Technology, the automation of the vaccination process had to be done through simulations. It has not been an easy road because the process must satisfy a series of very rigorous requirements that exist when it comes doing an injection. Finally, a viable solution was reached that would reduce the number of nurses currently dedicated to vaccination.

The objectives that have been pursued to carry out the project were:

- Analyse structure of the robot and its usage in medical applications
- Design of an electronic gripper capable of handling syringes.
- Design of the vaccination station, with its respective elements.
- Programming the robotic arm
- Assemble 6-axis robotic arm prototype and make accurancy testing

1. State of art

In this part there is a brief summary about robotics used in the world of medicine. The first thing that is mentioned is the history of robots in this sector. And later more specific sectors are discussed. This provides us with some context and illustrates that the whole world of robotics is relatively new and still has an incredible capacity for growth.

1.1. Medical robotics

Robotics is the set of sciences aimed at designing, producing, and using robots for tasks typically intended for humans. The advancement of new technologies has made possible the application of robotics to different sectors, including medicine.
In this sense, robotics in medicine consists of employing automated machines for the development of medical functions such as surgery, diagnostic, or rehabilitation therapies [2]. The objective of robotics in medicine is to facilitate healthcare professionals, surgeons for example, in maximizing the precision of operations or surgeries, and to minimize the limitations and deficiencies related to human error.

Origins and evolution of medical robotics
From 1980 the first surgical experiments with robots in neurosurgery and orthopaedics began. In 1985, for the first time, a robot was used in surgery, this robot was the PUMA 560 robot, and it was used to orient a needle for a brain biopsy. That same year, due to this advance, research began on robotics oriented to neurosurgery [3].

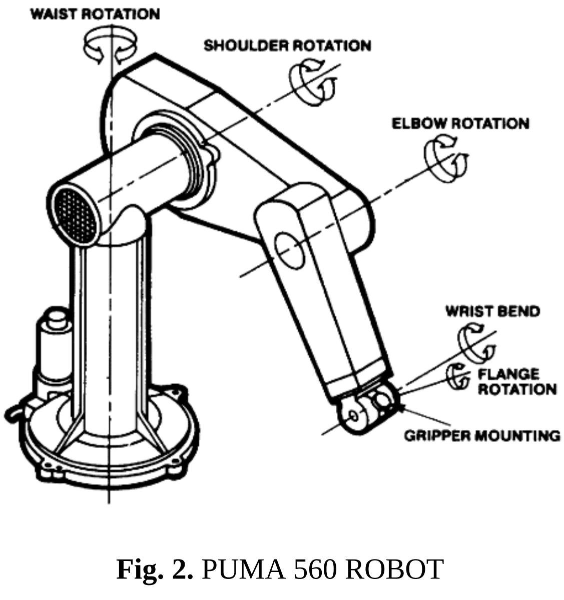


Fig. 2. PUMA 560 ROBOT

In 1988, the robot “World First” was the first in the world to extract fragments of human tissue in an operating room, specifically in a prostatectomy (to treat prostate cancer), so this milestone in the history of robotics corresponds to urology.
Computer Motion Inc., founded by Y Wang in 1989 Y Wang founded in 1989 the company Computer Motion Inc. specialized in the construction of surgical robots.

The AESOP (Automated Endoscopic System for Optical Positioning) project dates to 1994, with the Model 1000, the world's first FDA-approved robot. In 1996 Computer Motion Inc. continued its improvements to the AESOP 4000, available today. An intelligent robotic arm controlled using a previously configured digitized card recognizes the voice of each surgeon.. The robot can obey verbal commands due to a software interface called Hermes.
In 1997, Intuitive Surgical Inc. completed a Mona prototype, which is the precursor robot of the current Da Vinci, which used a master-slave system with a control console and independent arms so that a remote surgeon could control them.
In 1999, the most advanced robot known to date, the Da Vinci Surgical System, was created, with more than 200,000 operations being carried out to date.
As of 2007, different types of robots appear in medicine, such as servo-assistant robots (AESOP), coordinating assistants (Hermes).

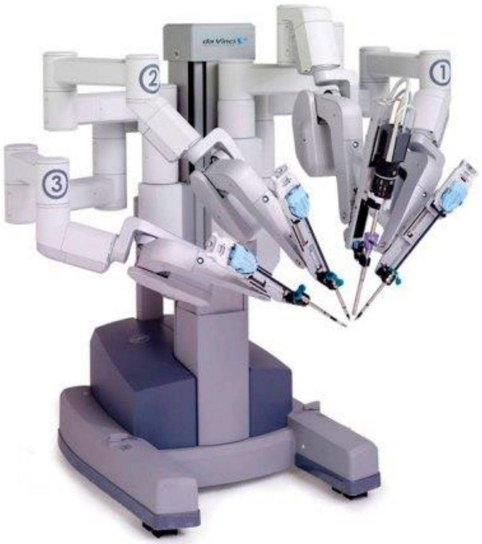


Fig. 3. Da Vinci Surgical System

Currently, the most widely used robots in medicine are surgical ones, which allow the implementation of less invasive surgical techniques, as they have greater precision than a human being.
In figure 4, we can see in which areas of surgical robotics are used the most.[4]

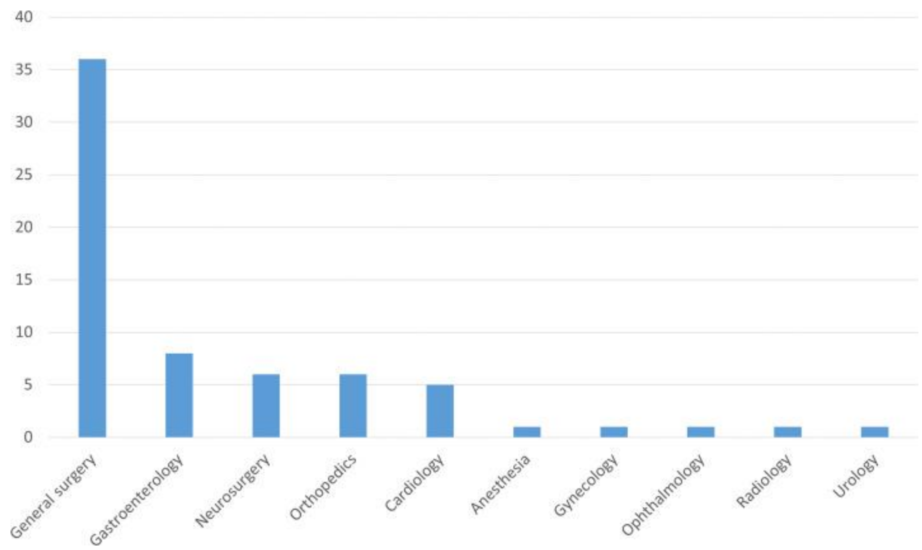


Fig. 4. Areas of surgical with more robotics systems

1.2. Applications of medical robotics

Currently, robotics is not only applied to medicine in the field of surgery, it is increasingly being integrated into different fields. In this section the most important applications in other branches of medicine will be detailed.

1.2.1. Robotics in rehabilitation

Rehabilitation is the set of procedures aimed at helping a person to reach the fullest physical, psychological, social, and educational potential compatible with their physiological or anatomical deficiency and environmental limits [5].
The first robot to be mentioned is the Monark 871E, a robot that allows rehabilitating patients who have lost mobility in their legs/arms. It is small and therefore perfect for all types of patients.

Fig. 5. Monark 871E

Another widely used robot is the Lokomat, it is a kind of exoskeleton designed in Germany, for the training and rehabilitation of people with cerebral palsy, Down syndrome, people with amputations, or physical limitations. This robot helps to re-educate the walking process in people with disabilities. Thanks to this, the patient's walking activity can be supervised, determined, and directed according to her needs. It consists of a system to carry the patient's weight and an automatic walker.

Fig. 6. Lokomat System

The last robot to be mentioned is the Rutgers Ankle. This idea was born with the initiative to turn rehabilitation into a more enjoyable process and for this purpose, this robot was designed that allows rehabilitation as if they were playing a video game. The system consists of a Stewart platform that provides the regeneration of force in 6 degrees of freedom, it has a pedal similar to that of a vehicle where the foot becomes a remote control that controls the graphic interface.

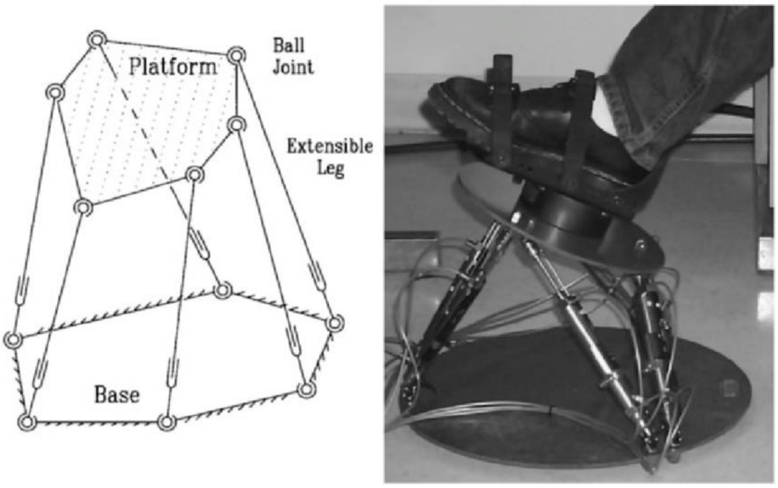


Fig. 7. Stewart Platform

1.2.2. Assisted robotics

The objective of assisted robots is to improve the quality of life of people with disabilities, providing them with greater autonomy and safety. These robots are generally designed for specific environments.

The first assistance robot on record was the MANUS, which was invented in 1980. This consists of an electric chair with a built-in robotic arm, controlled by the user [6].

Fig. 8. MANUS ROBOT

Currently, a robot has been designed to exercise disabled children, helping them to carry out mobility exercises and correcting them in case of error. This robot is called URSUS and from time to time it sends the training data to the doctor over the internet [7].

Fig. 9. URSUS Robot

Advances and robots in this area of medicine do not stop growing, other important developments and more used by society are prostheses.

2. Automation of the vaccination process

To automate the vaccination process, it was necessary to find all the information regarding the vaccination process. This information is mentioned in section 2.1. "Vaccination process". Subsequently, it was necessary to propose a scenario in which this process could be carried out using a robotic arm. It was assumed that the vaccine vial is already prepared and ready to extract the doses. It is also necessary for healthcare personnel to place the vial and syringes in their respective holders and for the person to be vaccinated to adjust their chair to the indicated height. Once these initial conditions were established, the main problem was the design of an electric gripper that could handle a syringe to extract the vaccine liquid and later inject it. Various protection systems have also been considered in the event of a system failure, such as an emergency stop button accessible to the person and plastic panels to prevent someone or something from crossing the robot's path.

2.1. Vaccination process

The vaccination process must meet a series of requirements depending on the type of vaccine to be given. Thus, it is necessary to follow a series of indications given by the health service and by the vaccine manufacturers themselves.

Before administering the vaccine, it is necessary to obtain the consent of the person to be immunized.

The most important factors to consider in the vaccination process are the injection technique, the proper selection of the needle (length and gauge) and the injection site. These factors can be decisive to avoid risks [8].

2.1.1. Injection technique

There are 3 types of injection technique:

- **Intradermal Injections**

In an intradermal injection, the needle penetrates only the skin (dermis). For this type, needles of length between 9.5 and 16 mm are used, and they should be injected with an angle between 10 and 15 degrees to the skin. Recommended sites for this type of injection are the forearm, upper chest, and back. This method is possibly the most difficult to execute correctly.

- **Subcutaneous Injections**

In subcutaneous injection, the needle penetrates very little space under the skin. The injection angle to the skin should be between 45 and 90 degrees. Short needles between 16 to 22 mm are used. The most common areas for subcutaneous injection are: the arms, the thighs, the periumbilical region.

- **Intramuscular Injections**

In an intramuscular (IM) injection, the needle penetrates muscle tissue at a 90 ° angle. The needles used in this method have a length of 25 to 75mm. They are usually injected into the arms (deltoids), buttocks, or thighs in the human body. IM is the most widely used method for injecting vaccines.

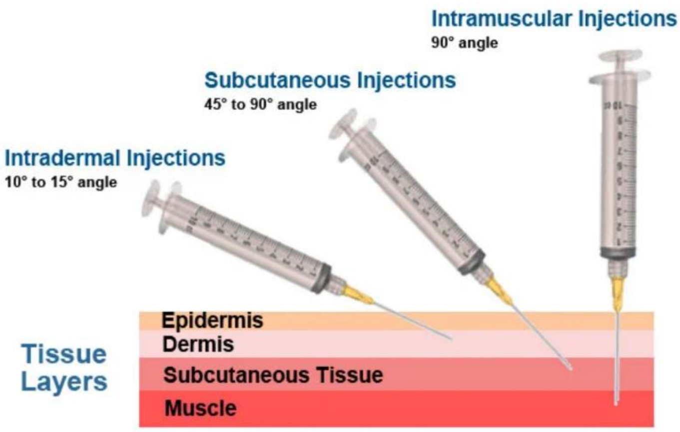


Fig. 10. Types of injection technique

2.1.2. Injection site

As important as the choice of method and needle is the correct site choice, the main thing is to avoid blood vessels and nerves. Figure 11 shows the leading site according to the technique used.

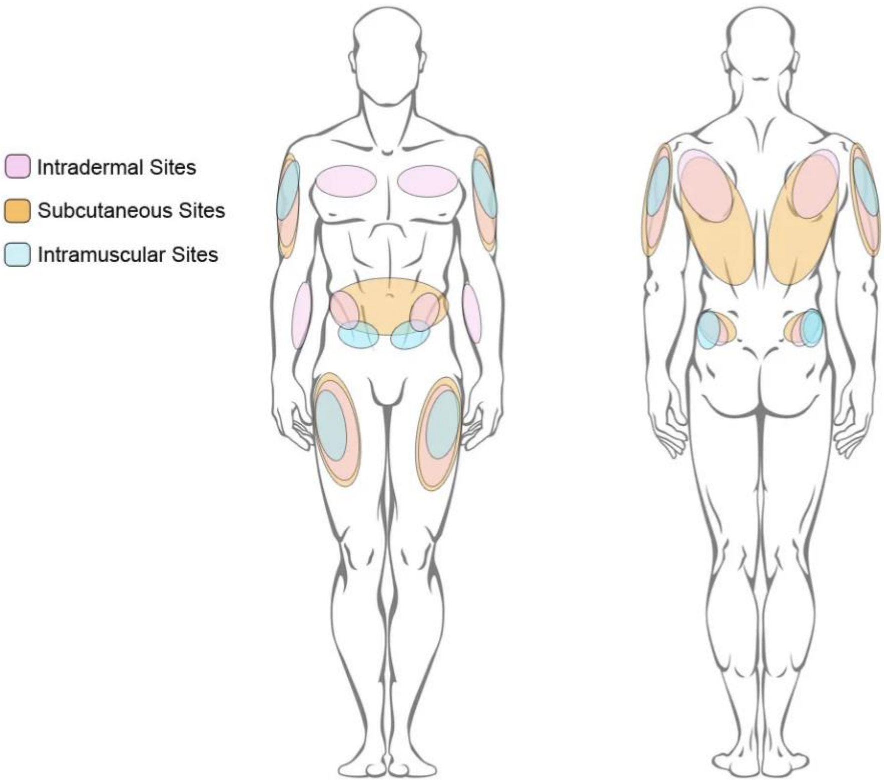


Fig. 11. Injection site

2.1.3. Needle length and gauge

Selecting the correct needle requires choosing the correct length and gauge. The length is from the end of the shaft to the tip of the needle. Moreover, the gauge can indicate weight or diameter. For choosing the correct needle, it is necessary to know how deep it needs to reach.

- **Needle Gauge**

The selection of needles by gauge is made considering the thickness of the skin and the injection depth. The smaller the gauge number, the larger the diameter. A larger diameter also means thicker and stronger walls. Low gauges are best for high viscosity fluids. Tiny diameter needles cause less pain and are recommended for low viscosity medications. Select a lower gauge number when using a high viscosity drug. Fine gauge (small diameter) needles offer less pain to the patient and are suitable for low viscosity medications.

- **Needle Length**

The length chosen depends on the area where the injection will be made. Typically, the deeper the depth, the longer lengths will be needed. For example, intramuscular injections require longer needles.
Figure 12 shows the different types that exist depending on the gauge and length

Fig. 12. Types of needles

2.1.4. Application process

The following image provided by the World Health Organization shows the procedure to be followed to vaccinate a person [9].

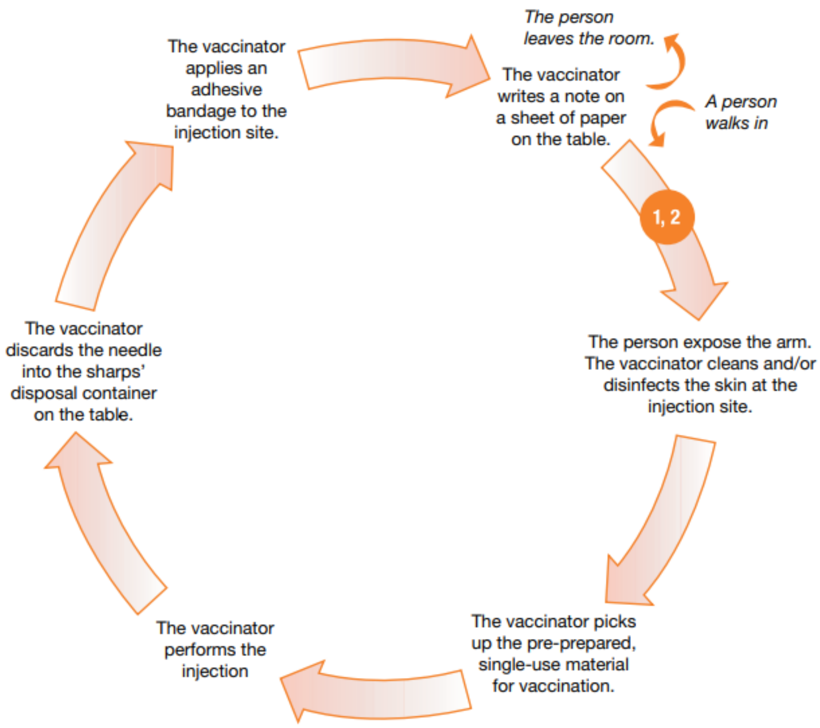


Fig. 13. Injection process

All these steps can be easily performed by a robotic arm. Including the procedure of disinfecting the skin and disposing of the syringe once the operation is finished. The only point that the person must make is to put on a cotton ball once the puncture is finished.

2.1.5. Security

The Centers for Disease Control and Prevention (CDC) give a series of recommendations that must be followed for the total safety of the person who is going to receive the vaccine, as well as health personnel [10].

- The sharps container should not be placed on the ground. These objects must be placed in a safe area, on the wall, or the table, and they need to be always in an upright position.
- Immediately after using the syringe, it must be placed in the respective specific container.
- The container should not be filled more than the indicated fill line. If this line is not indicated, 3/4 of the total capacity should not be exceeded.
- In sharps containers, only sharps should be placed. For any other item, we can use standard waste containers.
- If there is non-sharp material contaminated with blood or other material that may be infectious, it must be deposited in a red bag for biohazard waste.

2.1.6. Current vaccines against Covid-19

Due to the growing demand for covid-19 vaccines, all are supplied in multi-dose vials, allowing more than one person to be vaccinated with a single vial.
The image below shows a summary of the data for vaccines currently administered in Europe.

	Pfizer/ BioNTech BNT162b2	**Moderna** mRNA-1273	**AstraZeneca/ Oxford** ChAdOx1-S/ AZD1222	**Janssen (Johnson & Johnson)** Ad26COVS1
Type of vaccine	mRNA in lipid nanoparticles 	mRNA in lipid nanoparticles 	Non-replicating adenovirus vector 	Non-replicating adenovirus vector 
Dosage	2 doses 21 days apart	2 doses 28 days apart	2 doses 28 days apart	1 dose or 2 doses 56 days apart
Antibody detection	7 days after booster	14 days after booster	14 days after booster	14 days after booster
Efficacy	95%	95%	70%	N.A.
Planned production volume	50M (2020) 1.3B (2021)	20M (2020) 0.5-1B (2021)	3B (2021)	1B (2021)
Storage requirement	-70°C±10°C	-20°C	2-8 °C	2-8 °C
Shelf life once thawed	5 days	30 days	180 days	180 days
Phase III trial enrollment	43,000 (age 16-85)	30,000 (age 18+)	11,500 (age 18+)	Single dose 60,000 Two dose 30,000 (age 18 +)
Percentage high-risk population in phase III trial	40.90%	42%	N.A.	N.A.

Fig. 14. Current Covid-19 Vaccines Information

2.2. Components

2.2.1. CAD software

The program chosen for the design of these components has been AutoCAD, an AutoDesk program designed in 1982. This program allows optimizing the design work of many professionals [11].
With the arrival of AutoCAD 3D, the design of parts and structures in three dimensions was no longer a difficult task to carry out since, thanks to this program, it was optimized. This software allows us to create all kinds of shapes. Computer-aided design (CAD) has become part of the tools used by design, architecture, and engineering professionals. It was found that it is possible to optimize all the development of project parts previously done freehand.
Knowing the features of AutoCAD 3D means being able to create professional-grade projects. With views from all angles, which allows us to verify the piece's structure in detail, detecting possible flaws present in the design.
Replacing manual work with digitized designs allows us to present projects with more order and confidence that they are error free.

2.2.2. Gripper design

The objective of this component is to manipulate the syringe. For this, it has at least two degrees of freedom necessary to extract and inject the vaccine. As seen in the Figure 15, the set of the gripper is made up of 3 individual pieces.

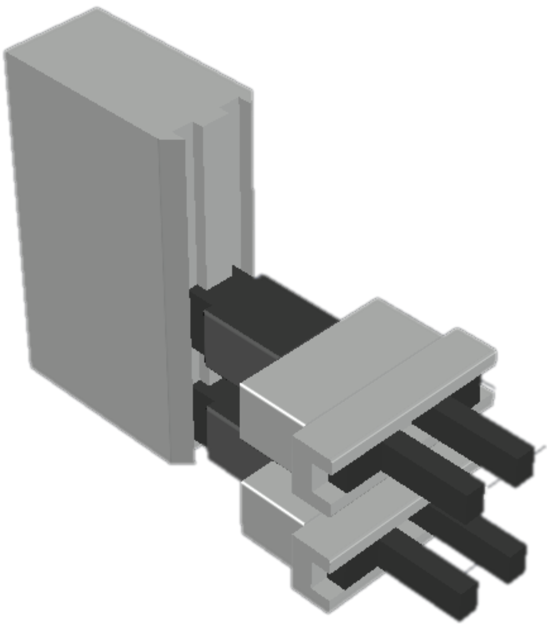


Fig. 15. Full Gripper

In Figure 16, we can see the base of the mechanism connected to the robotic arm on one side and where the independent grippers are located on the other. The disinfectant liquid spray is connected to the upper part. To move the grippers, there are two mechanisms that move independently, which allows the syringe to be manipulated for the operations mentioned above. The maximum aperture is 25 mm.

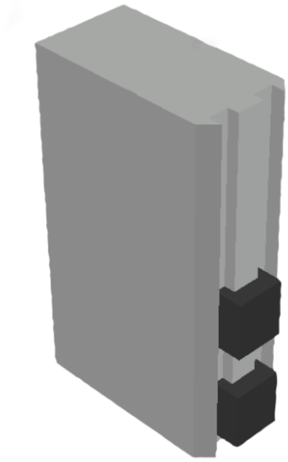


Fig. 16. Gripper Base

In Figure 17, an independent gripper is observed, which can quickly grasp the syringe through its callipers. Its temples are adaptable and have a maximum opening of 80 mm. In the final piece, there are two grippers like this, precisely the same.

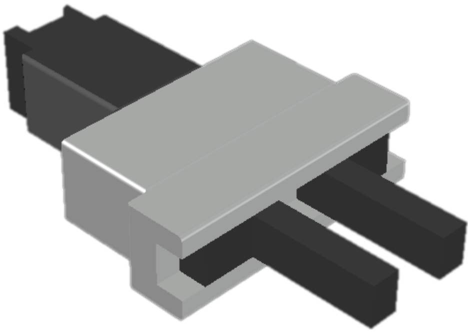


Fig. 17. Individual Gripper

In Figure 18 you can see how the disinfectant spray is, which is connected to the upper part of the base, so that the syringe does not interfere with the path of the spray.

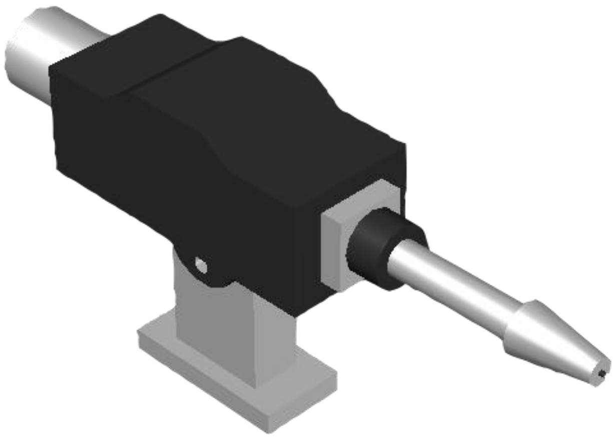


Fig. 18. Disinfectant spray

2.2.3. Arm holder design

The main objective is to securely fix the arm of the person who is going to receive the vaccine in the same position, regardless of their height or arm length. For this, it is proposed that the person adjust the height of the chair so that his shoulder touches the maximum established by the piece, resulting in the point where the vaccine should be consistently injected in the same position regardless of the anatomical differences in vaccine recipients. As can be seen in Figure 19, the piece is hooked to the table through a grip system on the legs, thus making the piece protrude from the table, allowing the person to have the arm fully stretched and at no time feel uncomfortable.

Fig. 19. Arm holder

2.2.4. Syringe design

As seen in section 2.1, there are many different types of syringes as well as needles. For this project, a standard syringe for the injection of 1ml insulin has been taken, which is the one used by many countries for the injection of the covid-19 vaccine. The following image shows the actual syringe from which the 3D model was designed for the simulation program.

Fig. 20. Real Syringe

The use of this syringe is just an example, the robot and its programming allow the necessary parameters to be freely modified to fit all types of syringes. Then, the 3D model of the syringe is observed, which has been used for the development of the project.

Fig. 21. Virtual Syringe

2.2.5. Syringe holder design

This support has been designed to allow the robotic arm to reach the syringes easily. In this case, the number of syringes has been set to 6. Because the most widely distributed vaccine in Europe today is the Pfizer-BioNTech vaccine. There are six doses per vial [DOC PFIZER]. Therefore, when changing the vial, the syringes must also be recharged, optimizing the time as much as possible. The syringes must be placed with their protective hood. The support is prepared to hold the hood and allow the robotic arm to extract the syringe without any problem.

As with the syringe, the support can also be modified with some ease. It is only necessary to respect the distance between the syringes (necessary so that the gripper does not stumble), so that syringes can be added or syringe types changed.

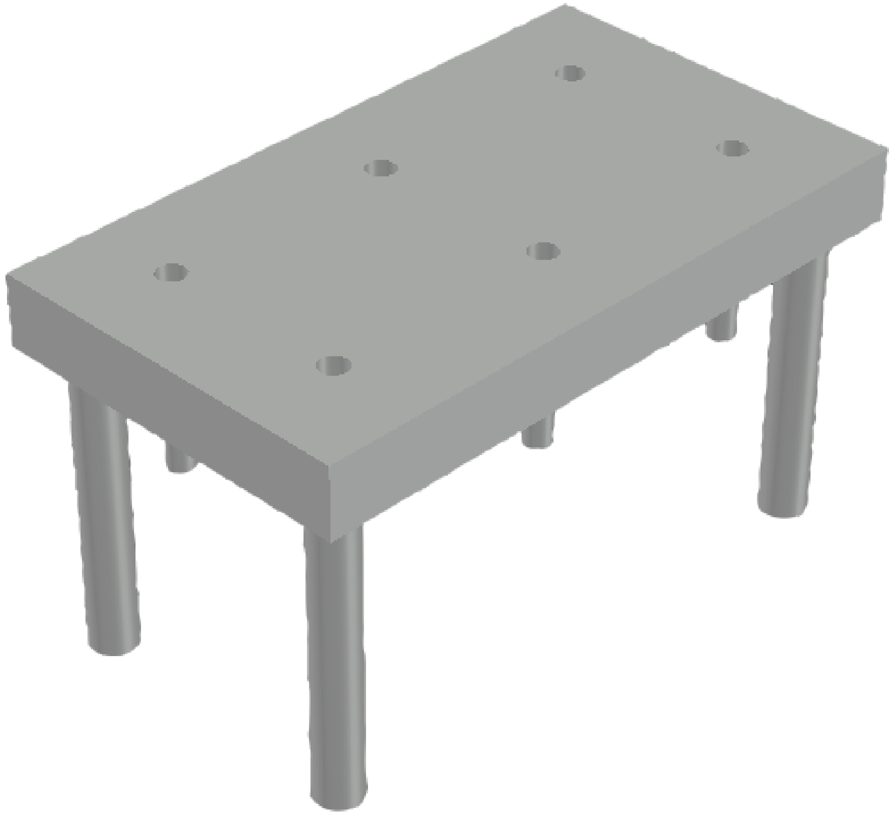


Fig. 22. Syringe holder

2.2.6. Vial and vial holder design

This support is responsible for holding the vaccine vial turned entirely downwards to allow all the liquid to settle near the needle entrance. There is no risk of liquid coming out when injecting or removing the needle. The vials have a safety system made up of a synthetic bromobutyl rubber, which prevents the liquid from escaping.
We can see the comparison between an actual vial and the 3D design made in AUTOCAD in Figure 23.

Fig. 23. Real vs simulation vial

The support would have a system to adjust to all vial sizes since each vaccine comes in different size vials. However, their shape is the same, allowing excellent versatility when putting vaccines from different manufacturers.
Below in Figure 24 is the 3D design image of the vial holder.

Fig. 24. Vial holder

2.2.7. Security elements design

To avoid accidents, safety systems have been created. The best way to show them is in the design program, to be able to explain their operation more simply and functionally. The first thing we find is the plastic protector, this plastic serves to protect the people who pass in front of the robot during its trajectory with the syringe.

Another element is the emergency button that is on the chair of the person who is going to receive the vaccine. This button is used to be able to stop the robot in case of a malfunction that has been observed by the person.

A screen has also been placed, in which the procedure to be followed is briefly explained. And behind the robot is its control console, in which you can control the robot manually, change the parameters necessary for the process, and activate it in the event of an emergency stop.

Finally, the yellow sharps container is observed. This container is mandatory in the vaccination process as syringes with the needle are considered sharp objects. This container must meet a series of requirements which are specified in section 2.1.5. "Security".

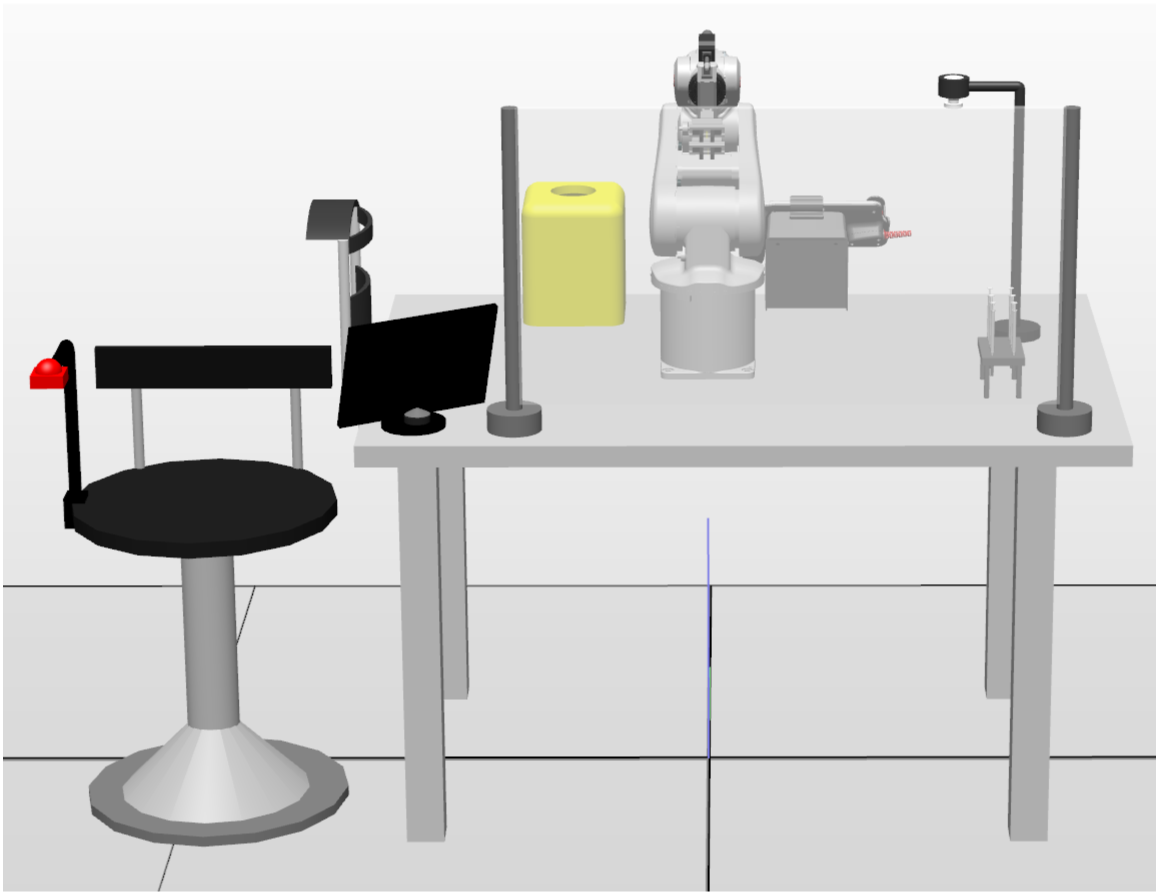


Fig. 25. Security element in Station

2.3. Simulation

2.3.1. Software RobotStudio

RobotStudio is a computer program designed by the ABB company. It is an offline programming and simulation software that allows us to program a robot from our computer without having to intervene physically and stop production [12].
RobotStudio is based on ABB VirtualController, which is an exact copy of the existing software that all ABB robots have.
It is the most used offline robot programming tool in the robotics world. Because we can import geometries designed in other more powerful design software, program the robots with the conditions that can be found in reality and all this without having the robot physically at the time of the simulation.

The steps to create a simulation in RobotStudio are:

1. Design the geometries, either with RobotStudio or with other software.
2. Create the tool to use.
3. Create a Smart Component from the previously created tool
4. Create the station, that is, add the geometries and graphically model them.
5. Create inputs and outputs
6. Create the work objects. These are the reference systems.
7. Create station logic
8. Create trajectories
9. RAPID programming

2.3.2. ABB IRB120 robot information

For the simulation, a 6-degree-of-freedom robot was used, approximately the size of the one I built at the Kaunas University of Technology.
The robot chosen is the ABB IRB120. It is a small industrial robot, which supports loads of up to 3 kg, has an open structure and can be adapted to countless tasks.
The robot is equipped with the IRC5 Compact controller, and its control software is RobotWare, which supports the control of movement, development, program execution, communication, etc [12]

Payload	3 kg
Weight	25 kg
Reach	0.58 m
Noise level	<78 dB

Table 1. ABB IRB120 information

Fig. 26. ABB IRB120

In Figure 27, we can see the range of motion that the IRB120 robot has.

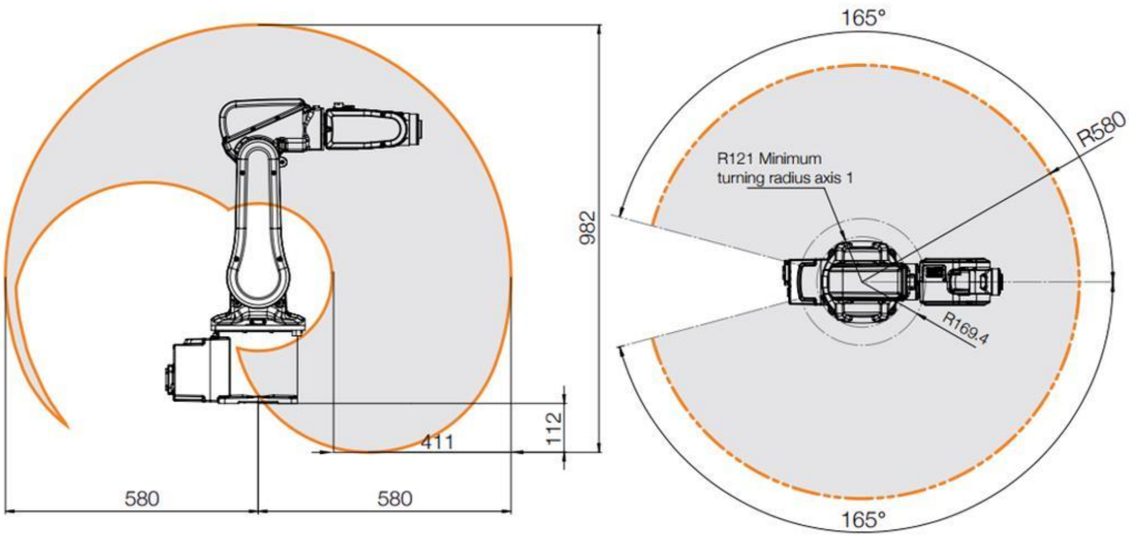


Fig. 27. ABB IRB120 range of motion

2.3.3. Tool creation

Once we have designed all the parts that we are going to need, the first step is to create the tool in RobotStudio. For this, RobotStudio allows us to choose between several types of mechanisms.

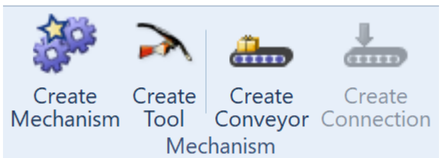


Fig. 28. RobotStudio menu

For the development of this simulation, it is necessary to create a mechanism, that is, the first option of the Figure 29, since it is the option that allows us to give our tool movement. If the second option were used, we could not move any part of the tool. It only serves for the software to detect geometry as a tool.
RobotStudio asks us for specific information and values that we give to our tool.

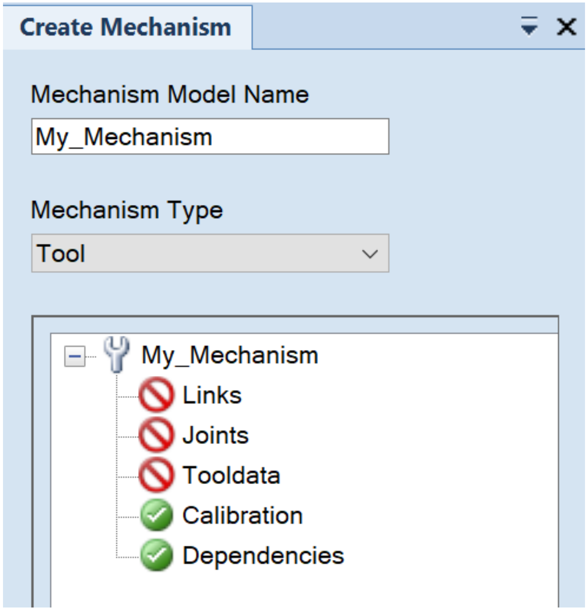


Fig. 29. Create Mechanism Menu

The first thing that is requested is the Links, which are the pieces that will form the tool. In this tab, we must establish the base link, and we must add piece by piece.

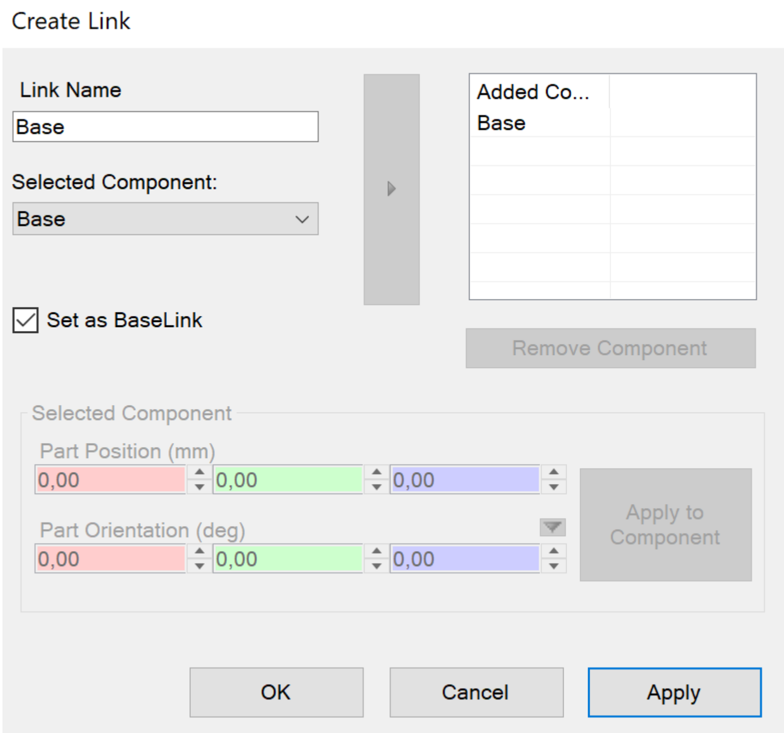


Fig. 30. Create Link Menu

The second thing would be the Joints, which are the movement axes that the parts are going to carry, an axis must be established per part (it is only necessary if it moves), and it must also be specified if a part is connected to another, for example, the callipers of gripper 1, we must specify that they are part of it so that when the gripper is moved, the callipers are also moved with it.
As shown in the following image, we must specify which object is moving and where it is connected. In this case, it is calliper one, and it is connected to gripper 1. The next thing would be to select the type of movement, in our case, prismatic. Once the type is established, we must establish the axis. For this, it can be done by giving it 2 points. Furthermore, finally, it is necessary to set the maximum and minimum that can be moved in mm. This step must be repeated with all the pieces that have mobility.

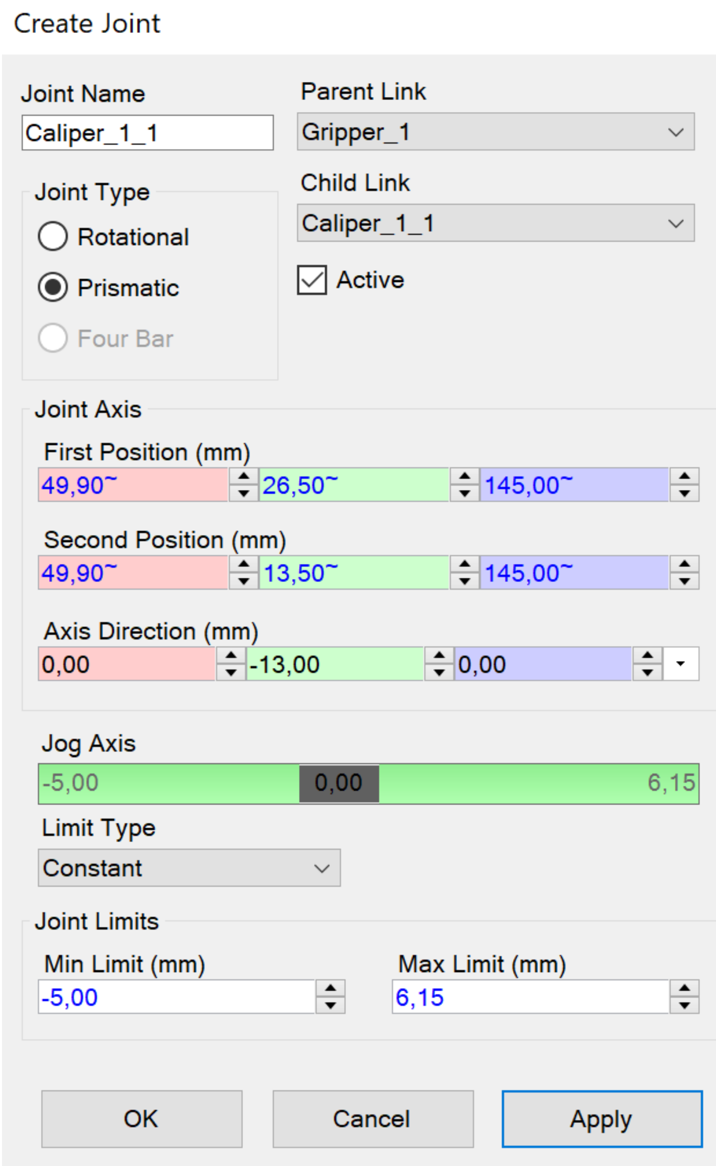


Fig. 31. Create Joint Menu

Third, we must enter the tool's data, that is, its working point, which we will use later to place the gripper in the position we want. For this, the midpoint between the two callipers has been selected. It is also necessary to establish the centre of gravity. In this case, the value is approximate since, as it is not a very heavy piece, it is not necessary to calculate it precisely. This step must be done with both grippers and with the base.

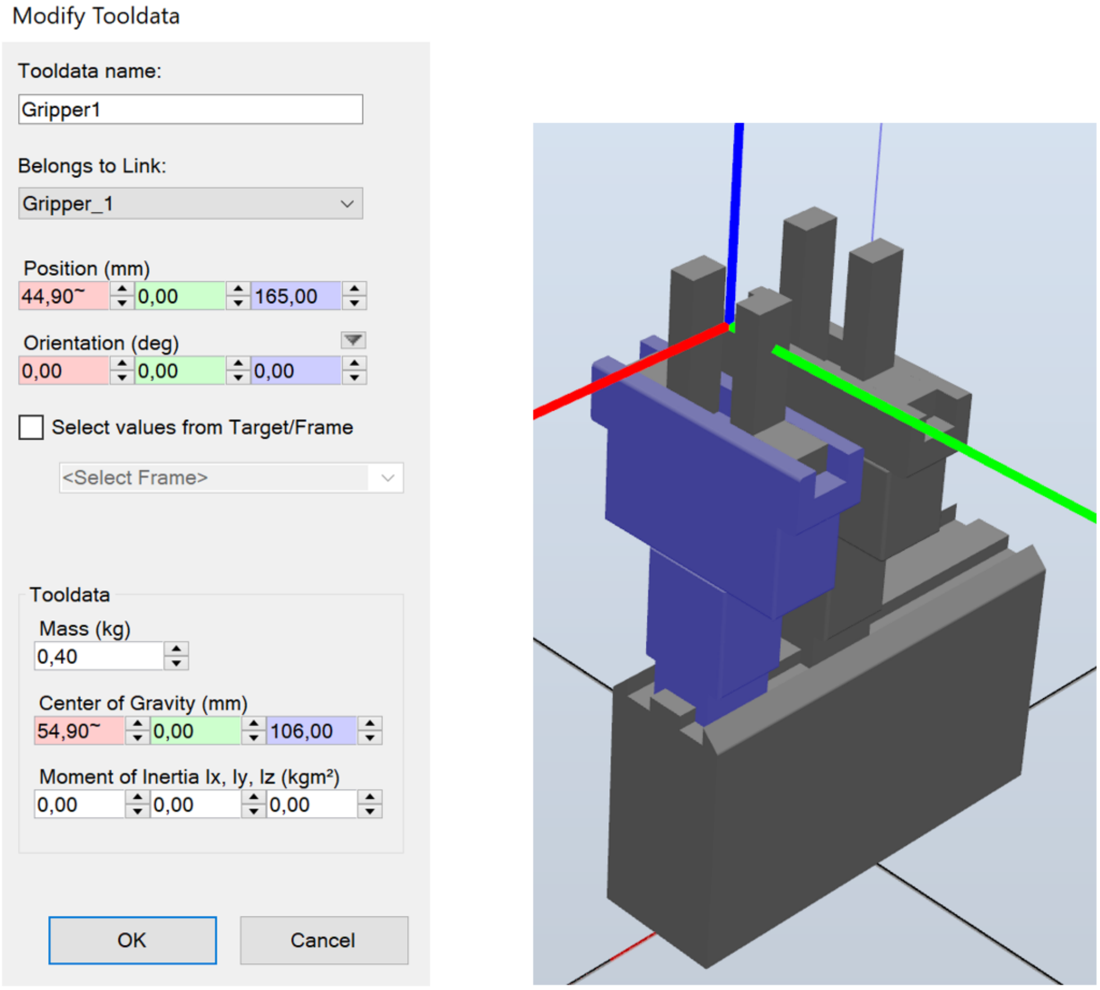


Fig. 32. Tooldata Menu and WorkingPoint of the gripper

The calibration part does not need to be set, since in this case, it is not necessary.

Finally, the dependencies section is already predetermined as valid because the components do not need to depend on each other, in this case, if the dependency of the gripper callipers has been established to prevent one from opening or closing more than the other. As can be seen in the image, the dependency is a factor of 1, which means that when calliper 1 moves a certain distance, calliper two will move precisely the same distance.

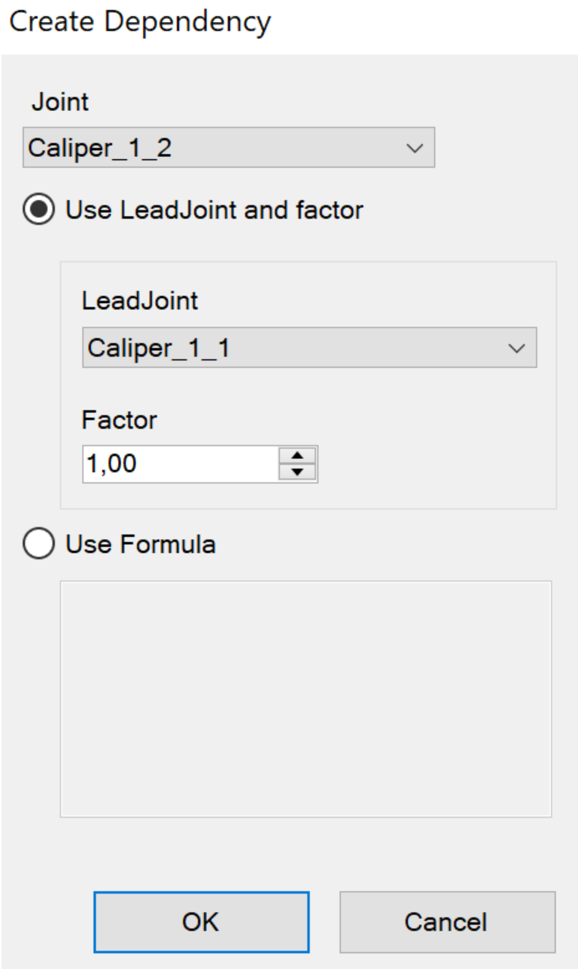
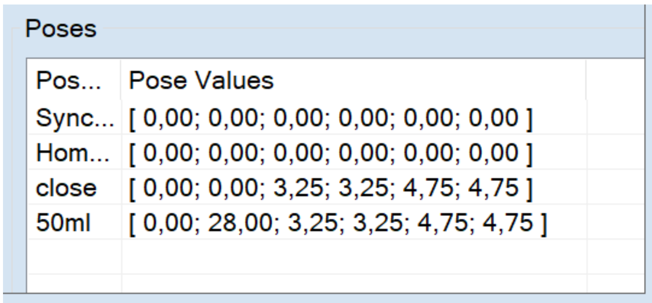


Fig. 33. Dependency Menu

Once all the steps are finished, the positions of the gripper must be defined. These are the ones that will be used later. The following positions have been defined:


Poses

Pos...	Pose Values
Sync...	[0,00; 0,00; 0,00; 0,00; 0,00; 0,00]
Hom...	[0,00; 0,00; 0,00; 0,00; 0,00; 0,00]
close	[0,00; 0,00; 3,25; 3,25; 4,75; 4,75]
50ml	[0,00; 28,00; 3,25; 3,25; 4,75; 4,75]

Fig. 34. Gripper Poses

"Close" is the position with the gripper closed to hold the syringe. In this position, we can see how each gripper opens the necessary size since each part of the syringe has a different thickness and "50ml" is the exact position in which the syringe draws 50 ml.

2.3.4. Smart component creation

A Smart Component is a RobotStudio object that exhibits behaviour that can be implemented programmatically.
Before starting with the programming of the intelligent component, it is necessary to establish which components we are going to need for its programming.
The “PlaneSensor” are sensors that have been placed between the callipers of the gripper to detect when the syringe is present and to be able to pick it up.
The “Attacher” and “Detacher” functions are used to connect and disconnect an object, and this is used to simulate that the gripper is holding the syringe.
Moreover, finally, the “PoseMover” functions move the axes of our tool to a specific position. As we can see, there is one for each position that had previously been established.
Once we have the functions, it is necessary to add our previously created tool.

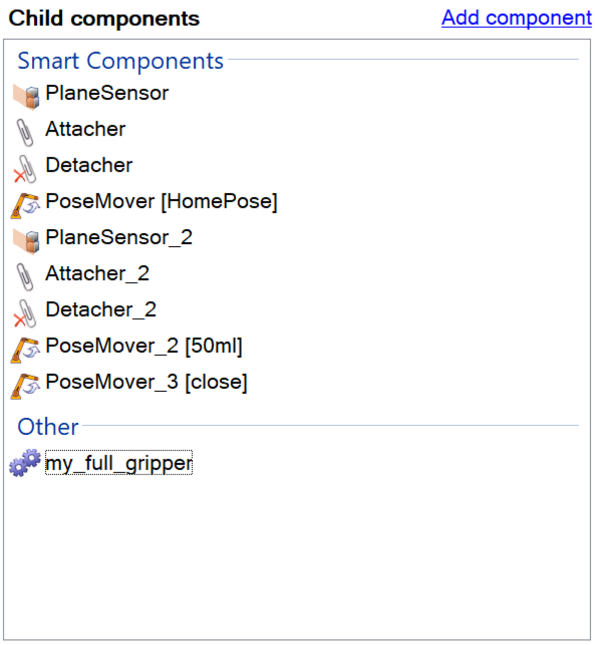


Fig. 35. Components of the SmartComponents

The next step is to program the Smart Component, for which input signals have been created, one signal for each position.

The “open” signal activates the “detachers” and moves the grippers to their initial position. The “close” signal activates the sensors, and these, in turn, activate the “attachers” when they detect something, and then the callipers of the gripper are closed. Furthermore, finally, the “extract” signal retracts the syringe’s plunger, moving the tool to the “50ml” position.

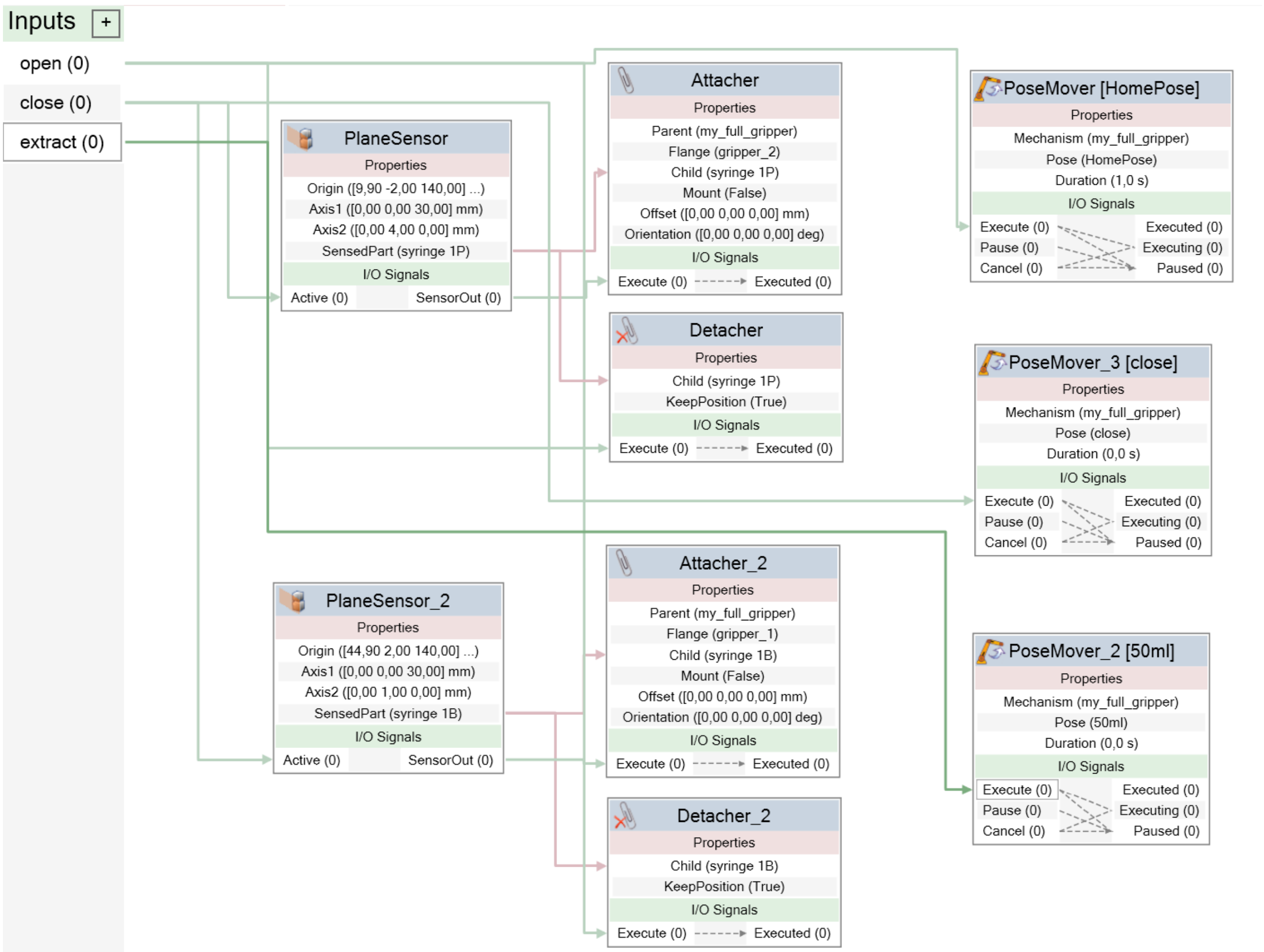


Fig. 36. SmartComponent programming blocks

2.3.5. Station creation

To create the station, we must first place all the pieces in their positions. Once established, we have chosen to place different WorkObjects, the reference systems, to make the program versatile.. A reference system has been placed at the base of the robot, syringe holder, vial and arm holder. It gives us the possibility of moving the pieces concerning their initial position, and all the paths associated with that WorkObject will automatically adjust to the new position.

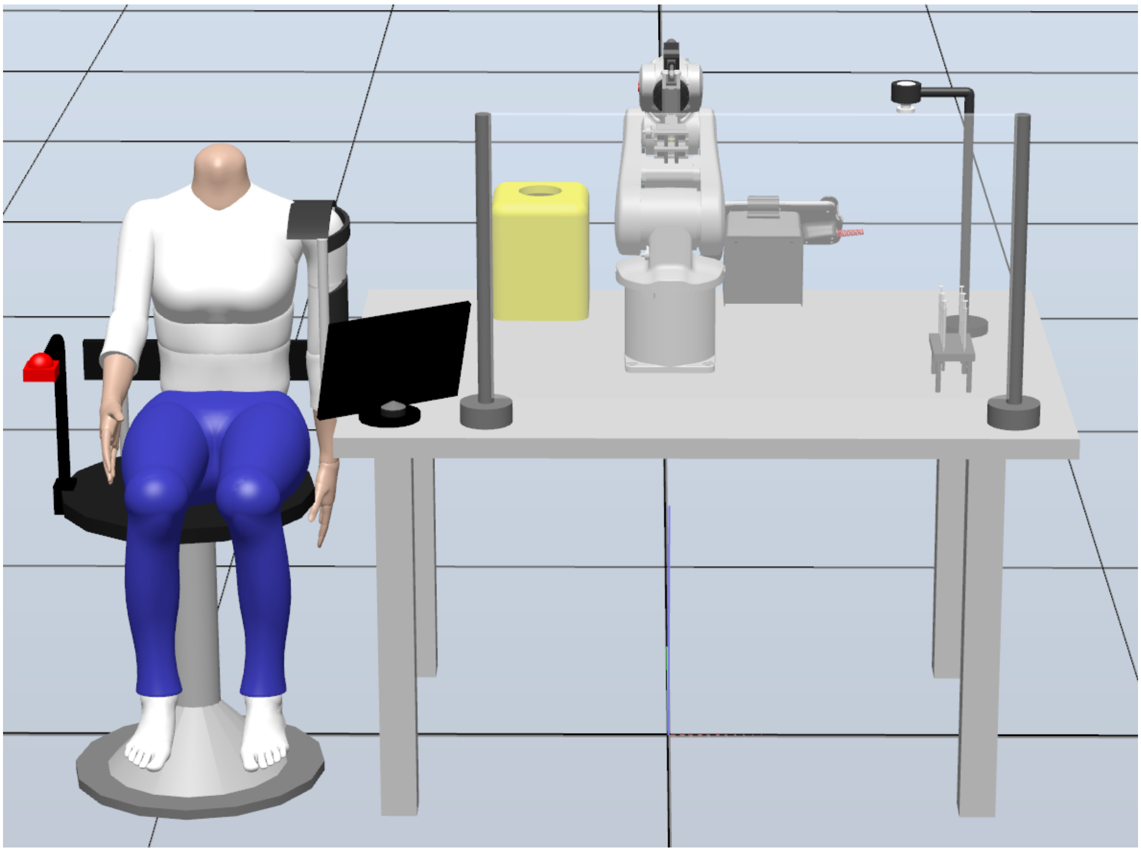


Fig. 37. Complete Station

To control the gripper is necessary to establish a series of outputs in the controller responsible for activating and deactivating the gripper inputs. Here is the station's logic:

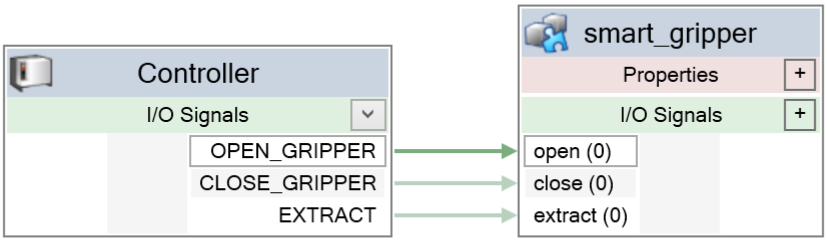


Fig. 38. Station's logic

2.3.6. Path creation

The creation of the trajectories is done by moving the robot manually and saving the coordinates of the positions we want to move the robot.

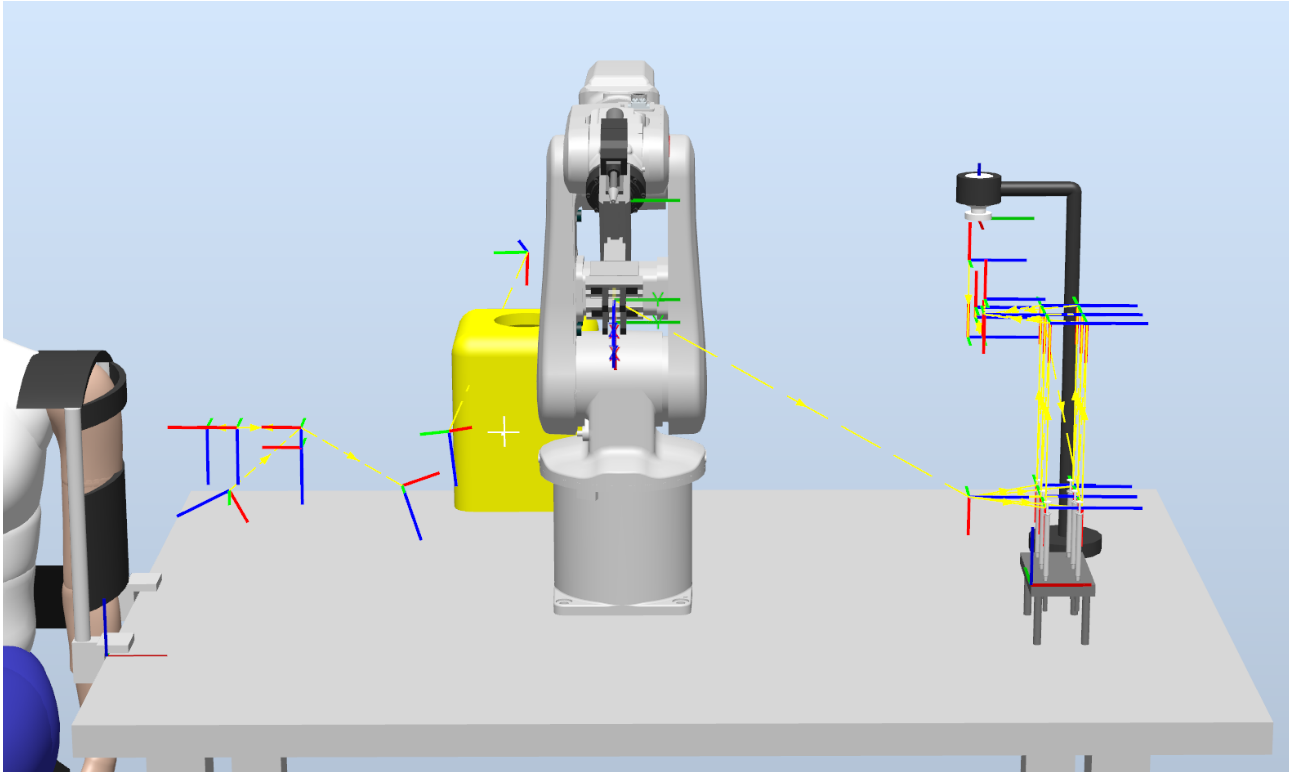


Fig. 39. Paths

2.3.7. RAPID Programming

RAPID is a high-level programming language. When we speak of a high-level language, we refer to the type of programming language that does not express algorithms taking into account the ability of machines to execute orders, but rather the one used taking into account the cognitive capacities of human beings.
Before analyzing the code, it is necessary to define specific RAPID commands for easy code understanding.

- "SetDo" sets the value of digital output (0 or 1).
- "TEST" acts as a switch.
- "WaitDi" is an instruction to wait until the digital signal is activated.
- "WaitTime" is used to wait the specified time in seconds.
- "MoveL" is an instruction that moves the robot linearly (in a straight line) from its current position to the specified position.
- "MoveJ" is used to move the robot quickly from one point to another when the movement does not have to follow a straight line

Next, the operation of certain functions will be detailed. The complete code is available in the appendix.
In the main function, all the gripper inputs are deactivated first if activated in a previous run. Then, a loop is entered that will repeat the vaccination process six times, which is the number of available syringes. Depending on the repetition of the program, the TEST function will execute the position of

the respective syringe. The program waits for the confirmation signal that the person is ready to receive the vaccine. Once the confirmation is obtained, the “takingVaccine” function extracts the vaccine from the vial. Then, with the “puttingVaccine” function, the vaccine is injected, and finally, with the “throwSyringe” function, the syringe is thrown into the particular container.

```
PROC main()
    VAR num i:=0;
    SetDO OPEN_GRIPPER,0;
    SetDO CLOSE_GRIPPER,0;
    SetDO EXTRACT,0;
    SetDO DISINFECTANT,0;
     FOR i FROM 0 TO 5 DO
        TEST i
        CASE 0:
            moveToSyringe1;
        CASE 1:
            moveToSyringe2;
        CASE 2:
            moveToSyringe3;
        CASE 3:
            moveToSyringe4;
        CASE 4:
            moveToSyringe5;
        CASE 5:
            moveToSyringe6;
        ENDTEST

        WaitDI go, 1;
        takingVaccine;
        puttingVaccine;
        throwSyringe;
     ENDFOR

ENDPROC
```

Fig. 40. Main function

The function of picking up the first syringe consists of moving the robot to the syringe position. Once there, pick it up and separate it from the syringe holder. The Targets are the constants in which the positions we want to access are defined. In the "Move" functions, it is necessary to configure the speed at which the robot will move, in mm / s, and the precision, where "z0" is the precision of 0 mm, but there is a more remarkable precision than that, and it is fixed as "fine", where the error that can occur is less than that of "z0".
This function is the same for all syringes. Only the Targets change since each syringe has its specific position.

```
PROC moveToSyringe1()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_20,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 2;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_30,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;
ENDPROC
```

Fig. 41. moveToSyringe1() function

As shown in Figure 41 in the extraction and injection moments, “fine” precision and a lower speed are used to avoid the minimum error. And the speed in the moments of injection and extraction is slower than in the moments of displacement.

```
PROC takingVaccine()
    MoveL Target_160,v200,z0,gripper2\WObj:=vial;
    MoveL Target_170,v200,z0,gripper2\WObj:=vial;
    MoveL Target_180,v200,z0,gripper2\WObj:=vial;
    MoveL Target_190,v100,z0,gripper2\WObj:=vial;
    MoveL Target_200,v20,fine,gripper2\WObj:=vial;
    WaitTime 1;
    SetDO EXTRACT,1;
    WaitTime 2;
    SetDO EXTRACT,0;
    MoveL Target_190,v50,fine,gripper2\WObj:=vial;
    MoveL Target_180,v200,z0,gripper2\WObj:=vial;
    MoveL Target_170,v200,z0,gripper2\WObj:=vial;
ENDPROC
```

```
PROC puttingVaccine()
    MoveJ Target_290,v200,z0,gripper2\WObj:=arm_holder;
    WaitTime 1;
    SetDO DISINFECTANT,1;
    WaitTime 1;
    SetDO DISINFECTANT,0;
    MoveJ Target_270,v50,fine,gripper2\WObj:=arm_holder;
    MoveL Target_280,v50,fine,gripper2\WObj:=arm_holder;
    MoveL Target_320,v50,fine,gripper2\WObj:=arm_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1.5;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_270,v50,fine,gripper2\WObj:=arm_holder;
    MoveJ Target_300,v200,z0,gripper2\WObj:=arm_holder;
ENDPROC
```

Fig. 42. tackingVaccine and puttingVaccine function

2.4. Results

In this section it is possible to observe images obtained during the simulation.

Figure 43 shows the clamp ready to take syringe 1, and Figure 44 shows how the clamp grasps syringe 1.

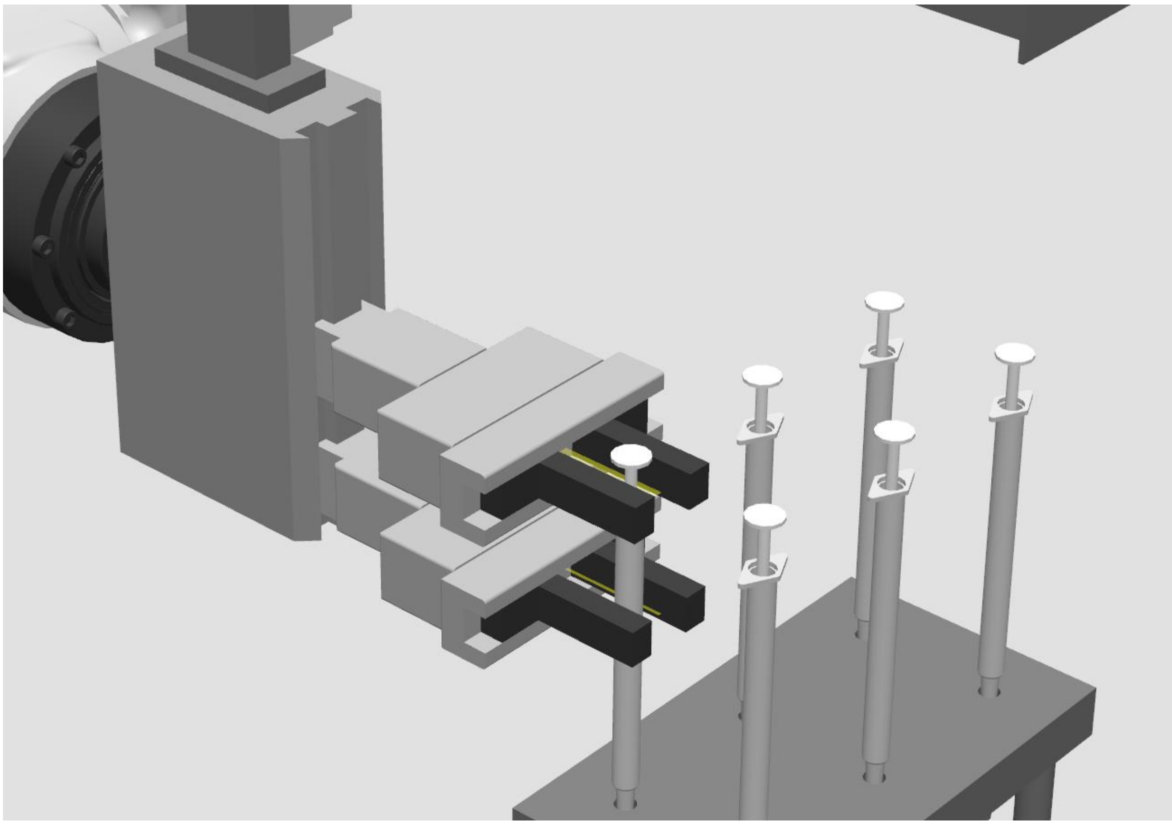


Fig. 43. Syringe position 1

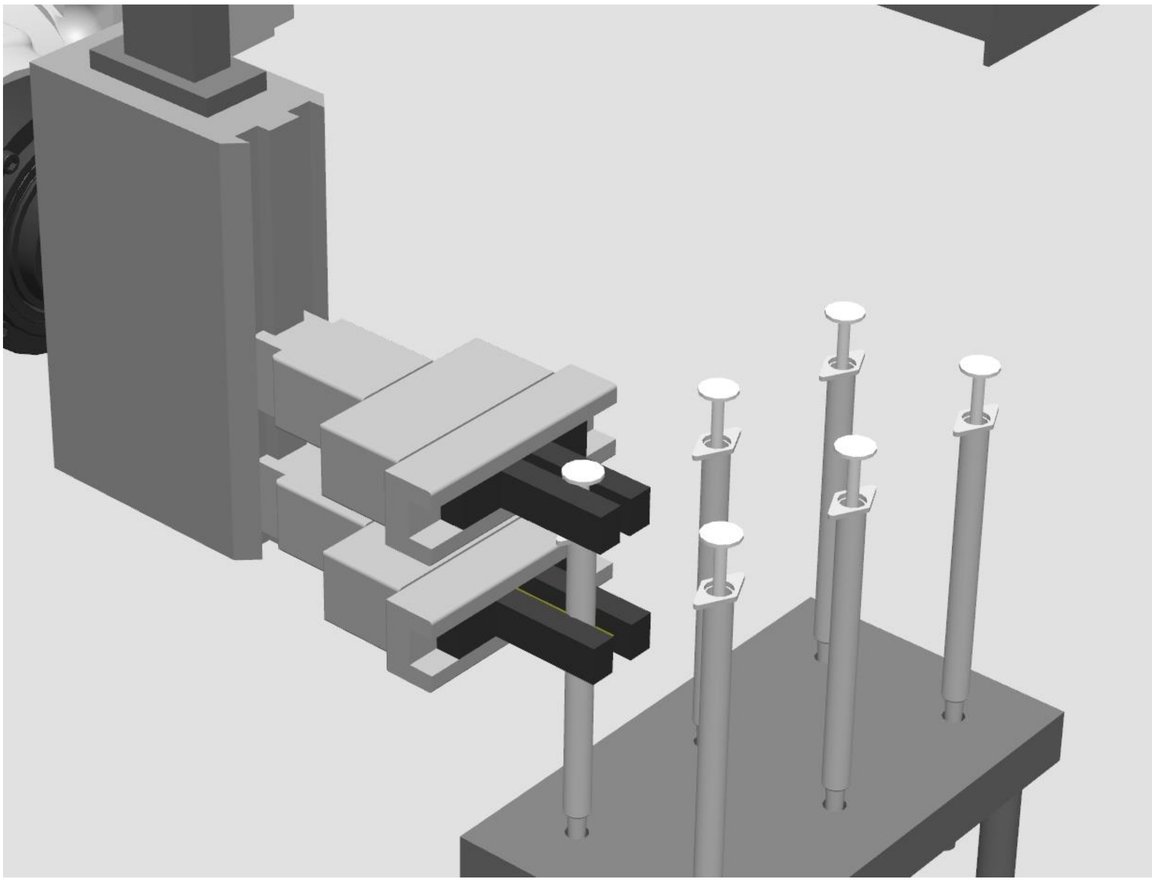


Fig. 44. Taking Syringe 1

In Figure 45 it is observed how the needle penetrates the vial, as it is observed the precision is maximum, since the needle enters with the necessary depth, to optimize the number of vaccines per vial to the maximum. In Figure 46 the same operation is observed, this time from a global perspective of the station, and it can be seen how the robotic arm manages to rotate the syringe 180º to easily reach the vial. And Figure 47 is observed at the moment when the vaccine is withdrawn from the vial.

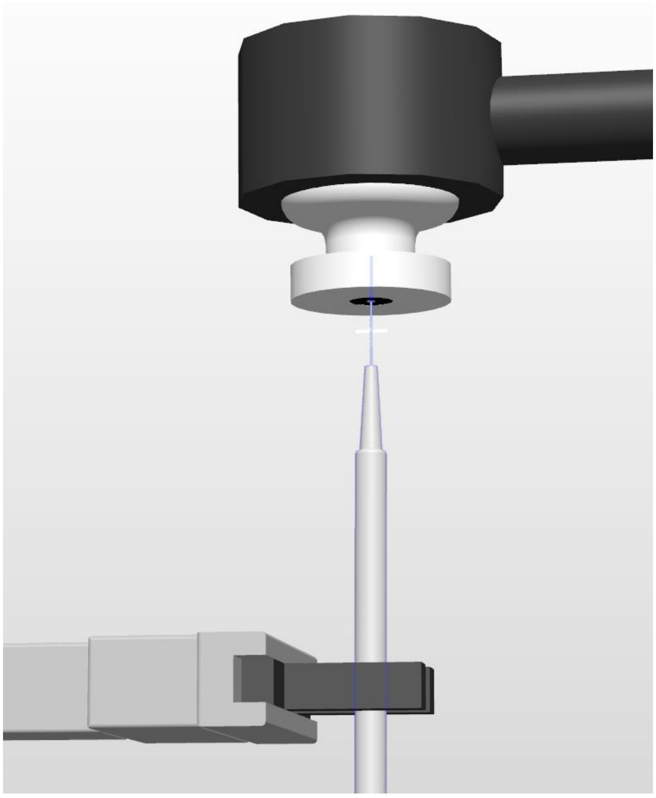


Fig. 45. Needle in the vial

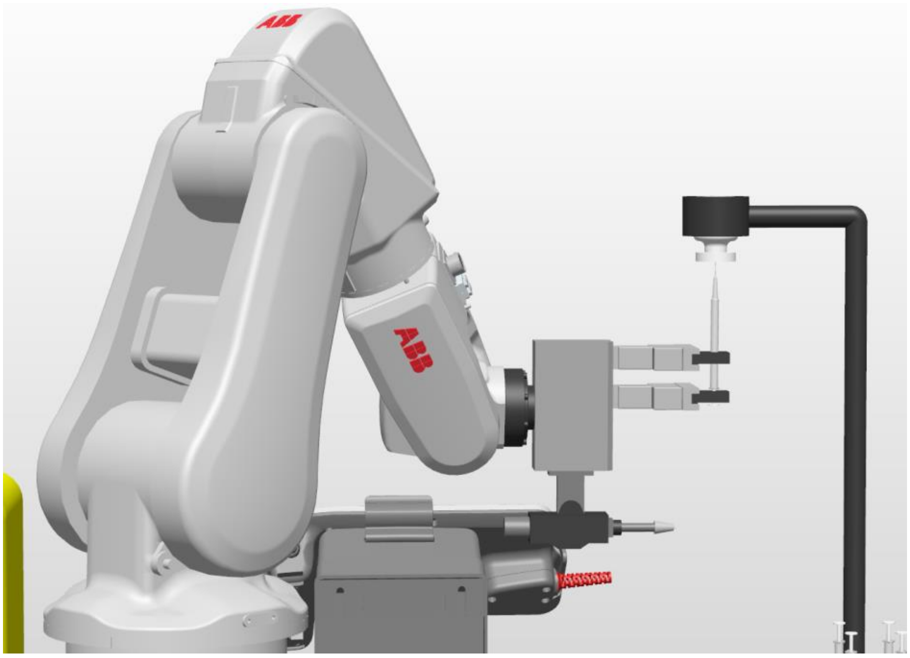


Fig. 46. Taking Vaccine

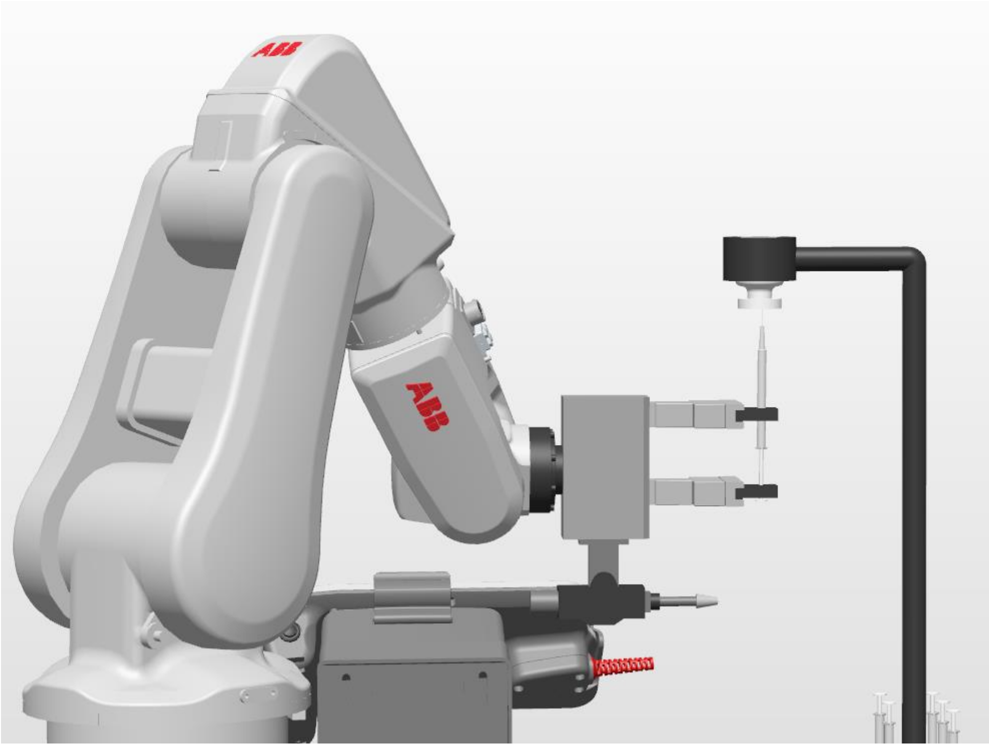


Fig. 47. Extracting Vaccine

Figure 47 shows the robotic arm in the position to apply disinfectant just before injection.

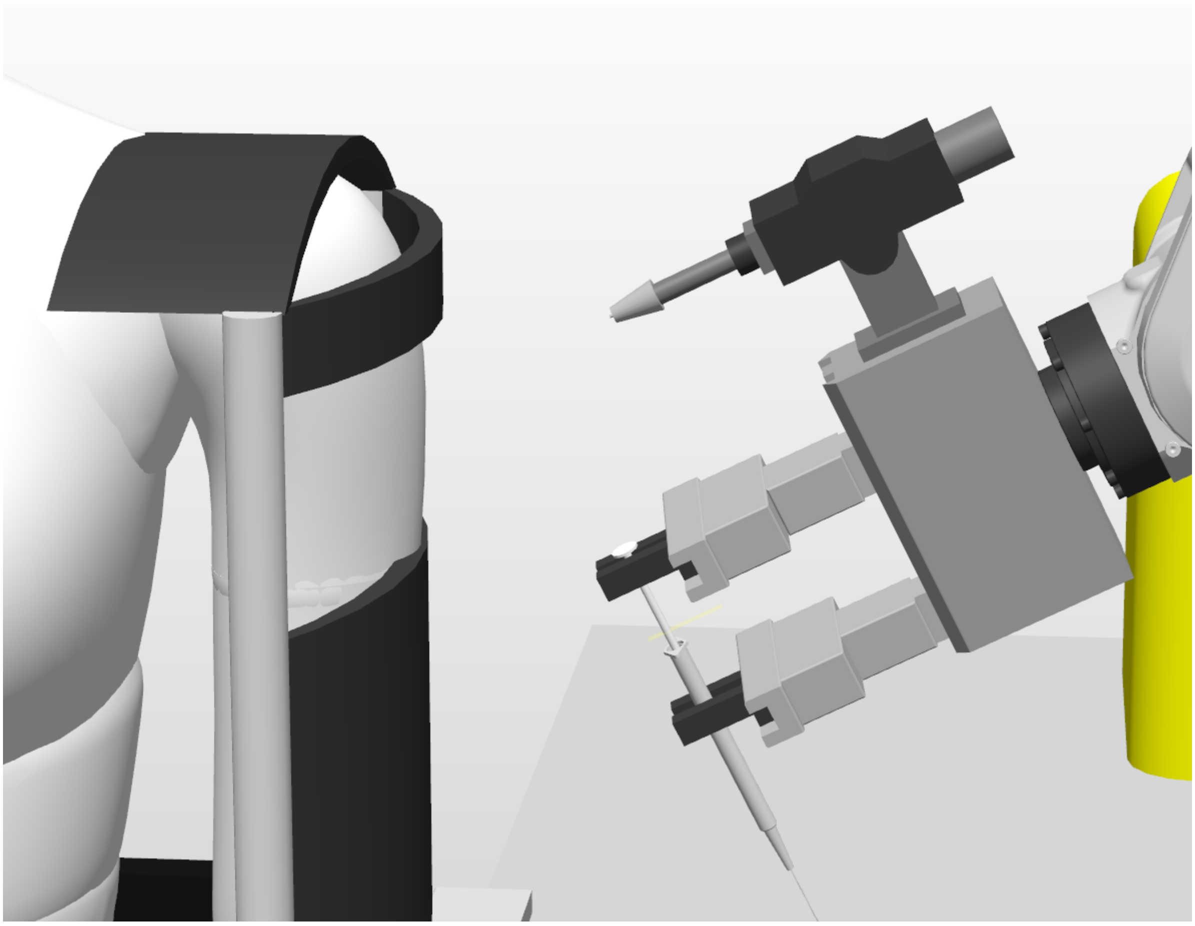


Fig. 48. Applying disinfectan

Figure 48 shows how the needle penetrates the person's deltoid. It can be seen how the needle penetrates 90 degrees to the person's arm. In Figure 49 the same operation is observed but in a general plane, the robotic arm has to be oriented that way to be able to inject the needle with a 90-degree inclination, because it is an intramuscular vaccine. This orientation also prevents the disinfectant spray from colliding with the robotic arm.

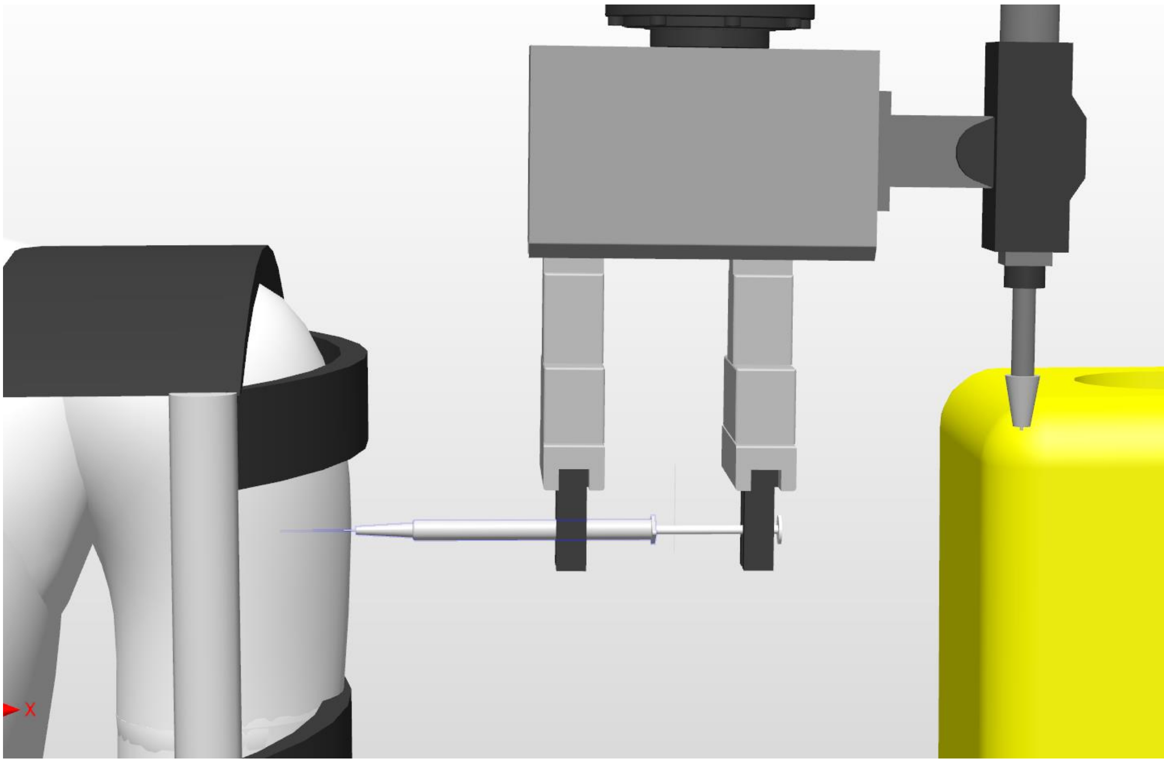


Fig. 49. Needle into deltoid

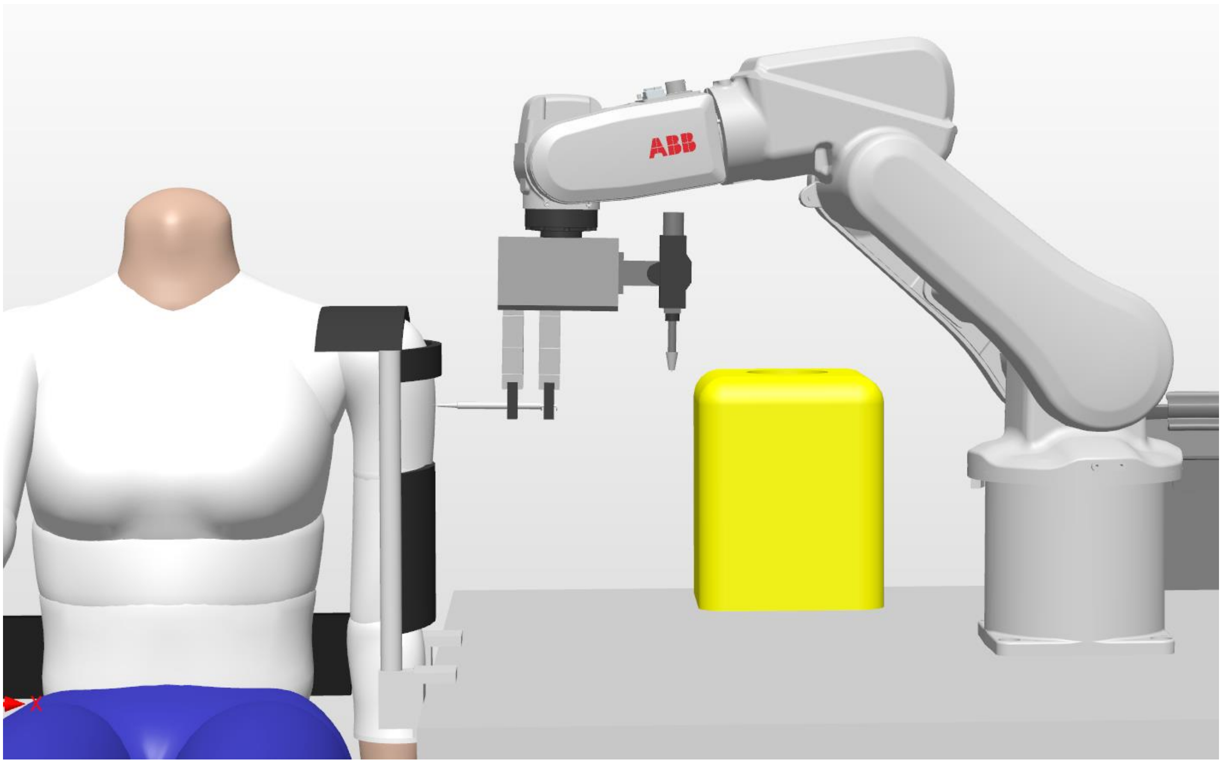


Fig. 50. Putting Vaccine

Finally, Figure 51 shows the moment at which the syringe is released into the sharps container.

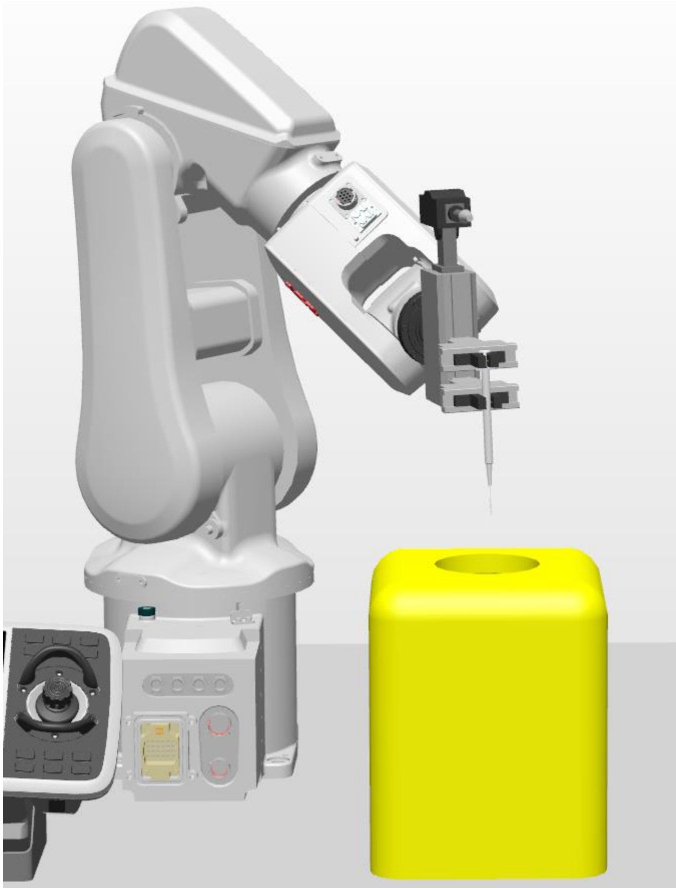


Fig. 51. Sharps container position

3. Robotic arm manufacturing six degrees of freedom

As mentioned in the introduction, the idea of automating the vaccination process appeared after the construction of a 6-degree-of-freedom robotic arm. This section will explain this robotic arm's manufacturing, assembly, and programming process since this type of robotic arm, manufactured with 3D printers, could be a cheaper alternative to the IRB120 arm used in the simulation.

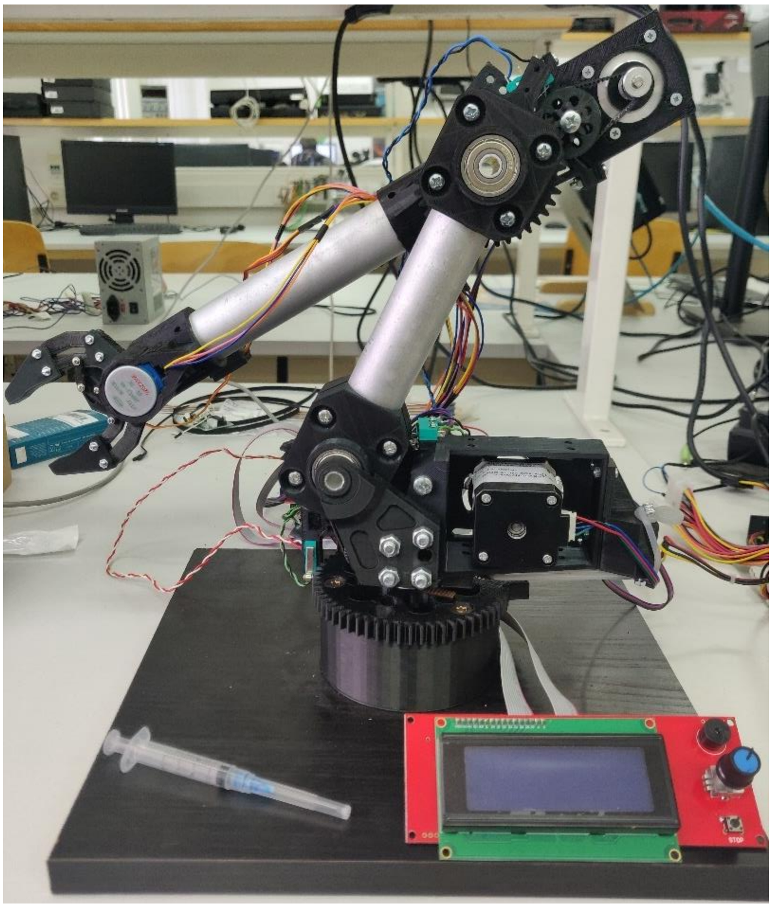


Fig. 52. Robotic arm

3.1. 3D print

3D printing technology was born from the idea of creating physical objects from a CAD program on a computer. The technology used by 3D printers is called "Additive manufacturing" (AM) and consists of creating an object by adding material in layers. The use of 3D printing has grown exponentially in recent years. As its cost decreased, its use increased. The fields where it is most used are engineering, industrial design, architecture, automotive jewellery, medicine, and even food [14].

3.1.1. Process

First, the part is designed using 3D design CAD software. Once the part is designed, it is exported to the STL (STereo Lithography) file format [15]. The STL format is the type of file that printers, or printing software, can understand. This type of file allows the part's geometry to be obtained through triangular faceting, thus eliminating unnecessary information and reducing computational calculation.

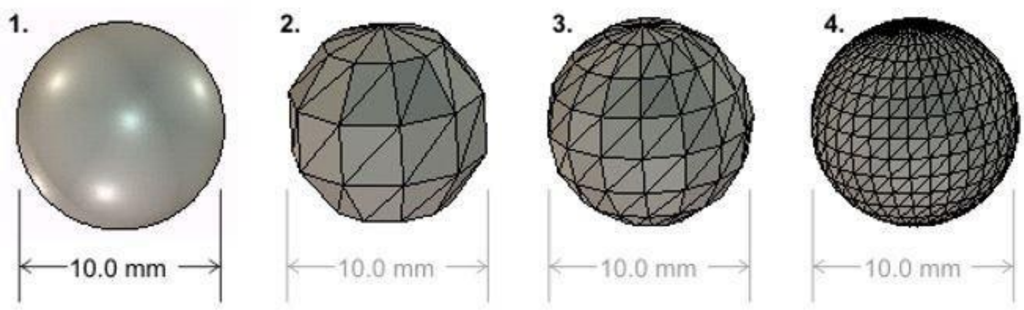


Fig. 53. Faceting process

Once we have the STL file, the printing software decomposes the part into parallel horizontal layers. The printing characteristics are also determined, such as filling, smoothing, type of adhesion, width. Once all the parameters have been established, the printing software creates the printing instructions in G-code, which is the language that 3D printers understand.

3.1.2. Robotic arm parts printing

3.1.2.1. Printer used

The Creality brand 3D printer was used to manufacture the parts, specifically the CR-20 Pro model. This printer uses fused deposition modeling (FDM) technology [16].

Fig. 54. Creality 3D CR-20

The printer has the following technical specifications:

- Mechanical arrangement: Cartesian
- Axis Position Accuracy: X = 0.012 mm, Y = 0.012 mm, Z = 0,04 mm
- Max. Extruder Temperature: 250°C
- Max. Bed Temperature: 110°C
- Print Speed: 100 mm /sec
- Filament Diameter: 1.75 mm
- Nozzle Diameter: 0.4 mm
- Filament Compatibility: ABS, PLA, TPU, etc
- Connectivity: Online, USB, SD Card

3.1.2.2. Material used

The material used is PLA, which are biopolymers derived from lactic acid. The use of PLA within 3D printing is ideal as it is dimensionally stable, readily available, and inexpensive. Some of its main characteristics are [17]:

- It is odorless, permanent, shiny
- It is biodegradable
- Extendable and elastic
- Not flammable
- It can adopt a great variety of mechanical characteristics depending on its manufacturing process.

3.1.2.3. Printing characteristics

Creality Slicer is the software used to create the paths that the printer's extruder must follow from the 3D model in STL format. Once the model is loaded, the software automatically generates the GCODE code for the established parameters. Figure 46 shows the Creality Slicer interface [18].

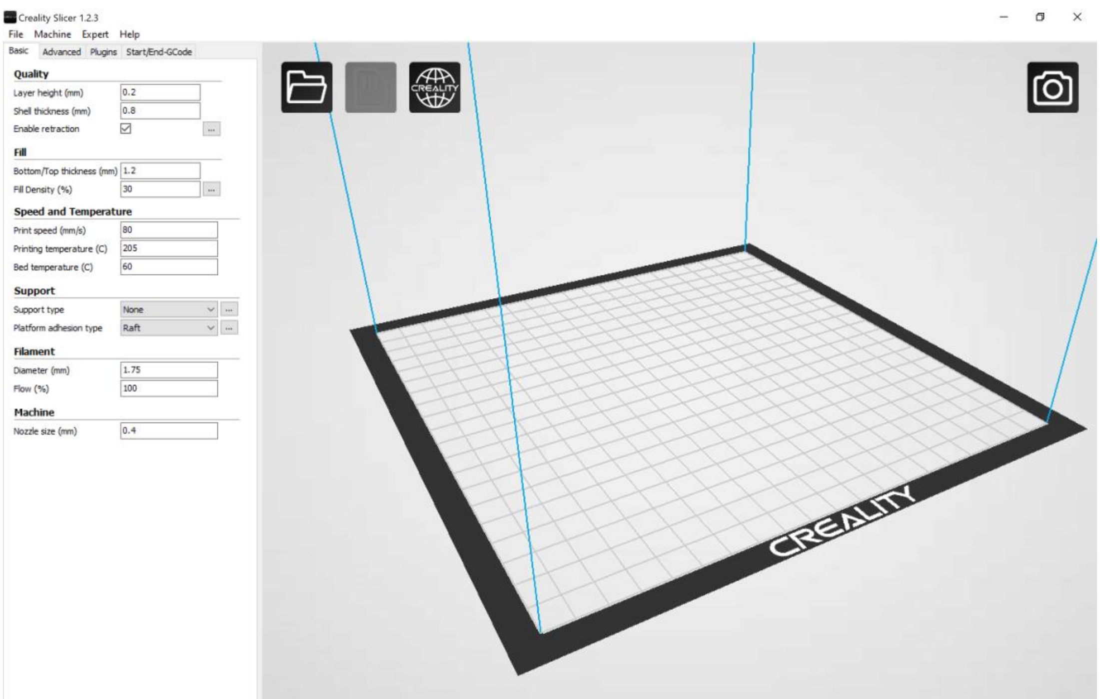


Fig. 55. Creality Slicer interface

Creality Slicer has specific parameters that can be manipulated, the most important and used are the following [19]:

Layer height: defines the advance of the z-axis, each layer that the printer generates deposits this thickness. In general, this value is limited between 0.05 mm and the diameter of the extruder nozzle. This value is one of the most important for finishing.
A shallow layer height generates solids with excellent finishes, while time is inversely proportional. That is, it increases as the layer height decreases.
A height of 0.2 mm has been used for all the project pieces, which leaves a good finish, adhesion between layers and does not require excessive printing time.

Layer, bottom, and top thickness: The existing thickness in the perimeter of the solid. In other words, it determines the thickness of the shell and the first and last layer.
In general, the values of 1.2 are used for “bottom / top thickness” and 0.8 for “layer thckness”. However, these values can vary to modify the resistance of the pieces.

Fill Density: it will be the amount of material inside the impression. This internal design can be changed depending on the resistance that is needed in the piece. Keep in mind that prints with more fill will take much longer and use more material. The density can vary from 0% (void volume) to 100% (solid part). A value of 30% has been used for this project. This value allows good times, and the pieces have sufficient hardness.

Print speed: It is the speed at which the printer head moves when it is printing. The lower this speed, the better the finish. Speeds can also be set for when you are not printing and when you are printing the perimeter. In this case, a speed of 50 mm / s has been used

Temperature: We can modify the printing temperature and the temperature of the bed. The printing temperature is the extruder temperature, which depends on the type of filament and the printing speed. The bed’s temperature serves to avoid a significant contraction of the filament when it cools quickly and that the piece does not separate from the bed.
In this case, a printing temperature of 205º and 60º has been used for the bed temperature.

Support: They are supports that serve as support for parts that need it, that is, parts that, if the supports were not carried, would crumble. Always try not to use supports because the material is wasted, and printing time is longer. To avoid using the brackets, try to place the part in the correct position. They can also be avoided in the design process.

3.1.2.4. Results

The robotic arm consists of 38 pieces, of different sizes, below is a table with the printing time and the amount of PLA used for each piece. The pieces are grouped depending on the axis to which they belong, since it is very difficult to name so many pieces and many of them so small.

Piece	Time (hh:mm:ss)	PLA used (mm)
Gripper part 1	00:02:01	128
Gripper part 2	00:36:26	3255
Gripper part 3	00:03:24	253
Gripper part 4	00:05:44	476
Gripper part 5	00:05:29	447
Gripper part 6	00:05:44	476
Gripper part 7	00:05:29	447
Gripper part 8	00:13:02	1213
Gripper part 9	00:13:02	1213
Axis 1 part 1	00:06:58	462
Axis 1 part 2	00:57:06	4570
Axis 1 part 3	02:09:52	12202
Axis 1 part 4	02:22:33	13007
Axis 2 part 1	00:48:12	4319
Axis 2 part 2	00:06:56	538
Axis 2 part 3	00:09:16	657
Axis 2 part 4	03:52:25	21600
Axis 2 part 5	00:15:25	1298
Axis 2 part 6	01:12:26	6589
Axis 2 part 7	01:58:22	11270
Axis 2 part 8	00:39:38	3817
Axis 2 part 9	00:19:10	1541
Axis 2 part 10	00:26:21	2074
Axis 2 part 11	02:55:14	14853
Axis 2 part 12	00:55:10	5158
Axis 3 part 1	01:01:37	4850
Axis 3 part 2	00:01:48	123
Axis 3 part 3	02:05:07	10805
Axis 3 part 4	00:31:44	1055
Axis 3 part 5	01:05:41	5899
Axis 3 part 6	02:04:54	11285
Axis 3 part 7	00:33:46	2633
Axis 3 part 8	00:03:46	66
Axis 3 part 9	01:05:23	5396
Axis 4 part 1	00:14:55	1232
Axis 4 part 2	02:01:29	10388
Axis 5 part 1	00:14:55	1232
Axis 5 part 2	01:34:12	8255
TOTAL	**33:24:42**	**175,082 m**

Table 2. 3D printing results

3.2. Mounting

Once all the parts are printed, the assembly of the robot begins, for this purpose, it has been built axis by axis, and subsequently, they have been joined. For the mechanical assembly, distribution belts have been required, and the tubes that have been used for the robot arms and to join the shafts are made of metal since they needed to be strong enough. Tubes printed on the 3D printer were not strong enough to support the total weight of the components. This is mainly due to the heavyweight of the main engines.

AXIS 1. This axis allows the robot arm to rotate on its base (X-AXIS).

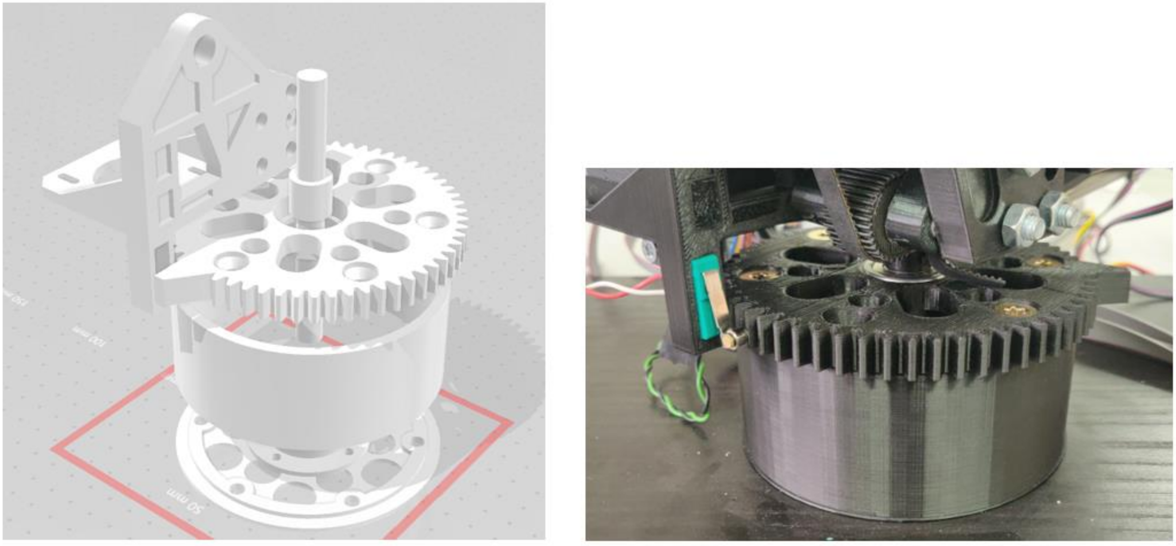


Fig. 56. Simulation vs real axis 1

AXIS 2. This axis generates the corresponding forward movement (Y-AXIS).

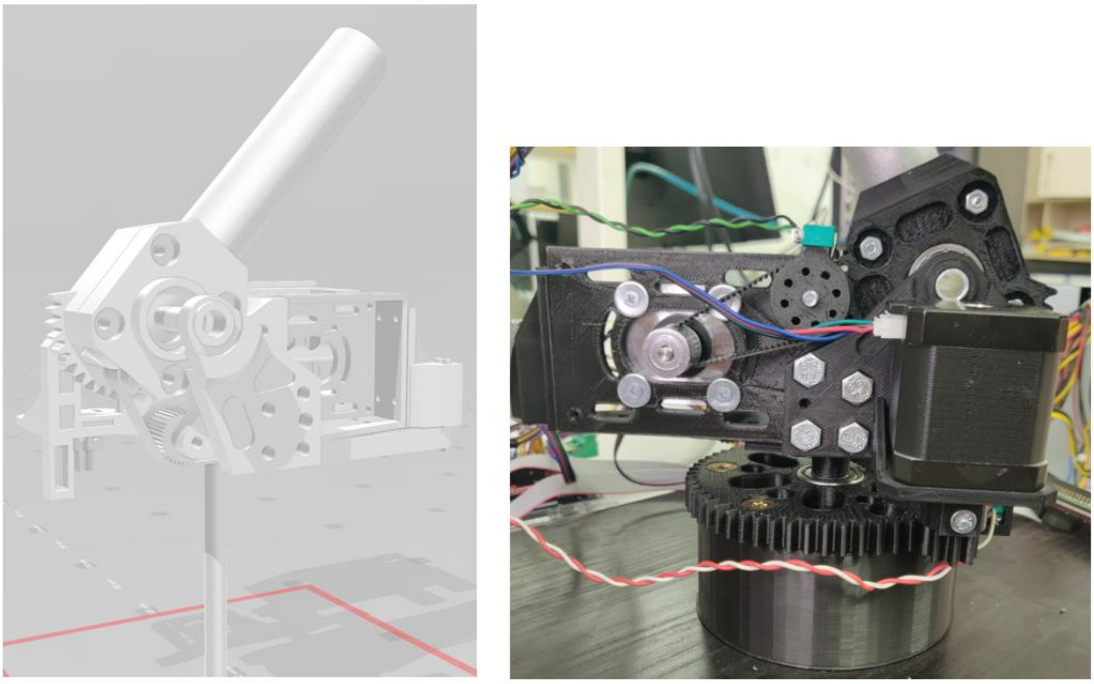


Fig. 57. Simulation vs real axis 2

AXIS 3. This axis allows the arm's height to be modified, thus reaching higher points (Z-AXIS).

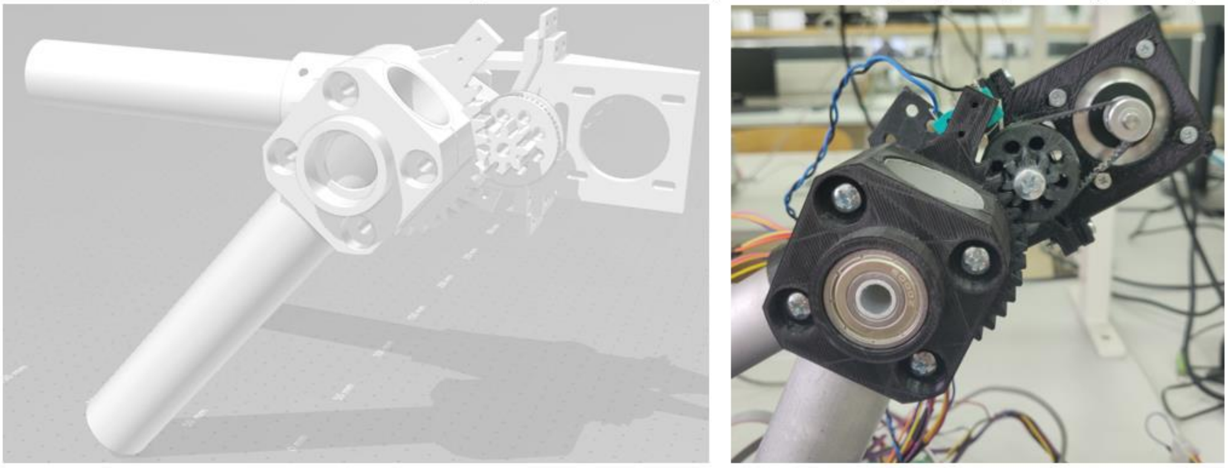


Fig. 58. Simulation vs real axis 3

AXIS 4 - 5. With these axes, the angle and inclination of the gripper can be modified.

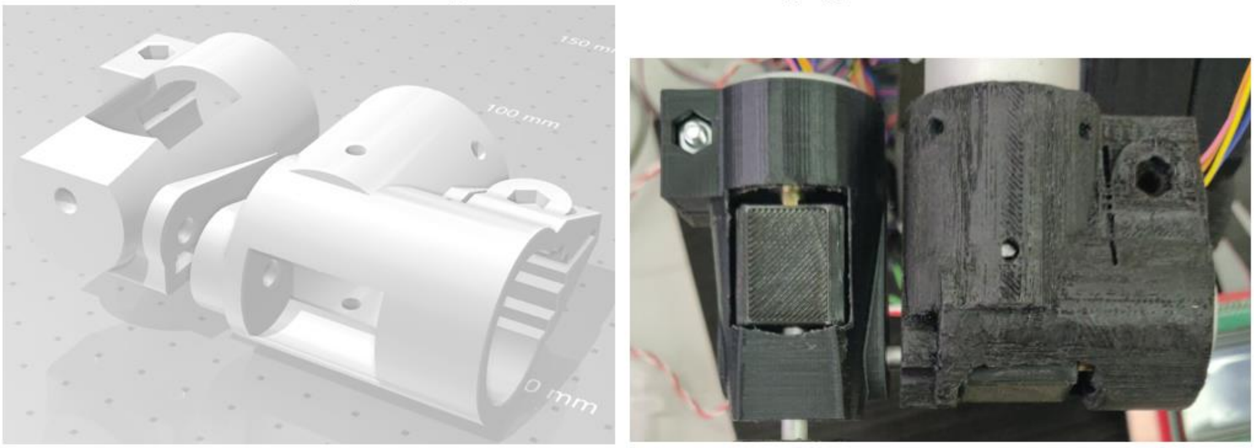


Fig. 59. Simulation vs real axis 4 and 5

AXIS 6 - GRIPPER. The creator chooses the gripper for this arm, but this item can be changed freely. The clip designed in the section "Automation of the vaccination process" could be placed.

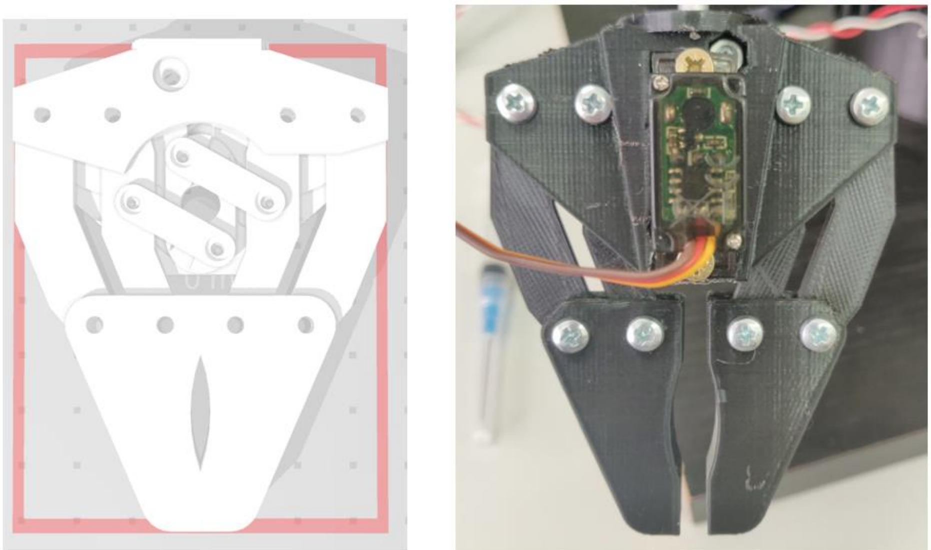


Fig. 60. Simulation vs real axis 6

3.3. Electronic part

3.3.1. Component used

3.3.1.1. Power supply

For any system to work, it is necessary to use an element that provides us with the necessary power. In this case, an ATX type power supply has been used. These types of sources can be found in desktop computers.

Fig. 61. ATX Power Supply

These types of power supplies have numerous outputs at different voltage levels. For the robotic arm, the 12V outputs have been used for the motors and the 5V one for Arduino.

The main advantage that these types of feeders offer is their low cost depending on the power they offer.

3.3.1.2. Stepper motors

A stepper motor is an electromechanical device that converts a series of electrical impulses into discrete angular displacements, which means that it can rotate many degrees depending on its control inputs. The stepper motor behaves in the same way as a digital-to-analog converter and can be driven by pulses from digital systems. This motor has the advantages of having precision and repeatability in terms of positioning [20].

Two types of stepper motors have been used, the Nema 17 motor for the movement of the main shafts and the 28BYJ-48 motor for the movement of the gripper shafts.

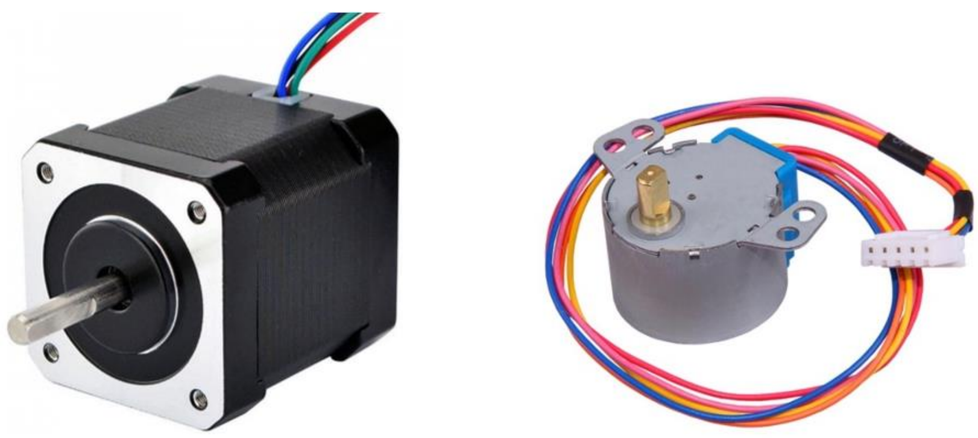


Fig. 62. Nema 17 and 28BYJ-48 Motors

The NEMA 17 motor is of the bipolar type, with a step angle of 1.8º. That is, it can divide each one of the revolutions or turns into 200 steps. Each winding that it has inside supports 1.2A of intensity at 4v of tension, with which it can develop a considerable force of 3.2 kg/cm [21].
The 28BYJ-48 motor is a unipolar motor capable of dividing the revolutions into 512 steps, which is 0.7º per step. It supports a current of 55 mA and can generate a force of 0.34 kg/cm [22]. Working with this motor in the Ramps is necessary to modify it and make it bipolar. For this, it is simply necessary to cut the red cable to access both windings separately, thus obtaining a bipolar motor.

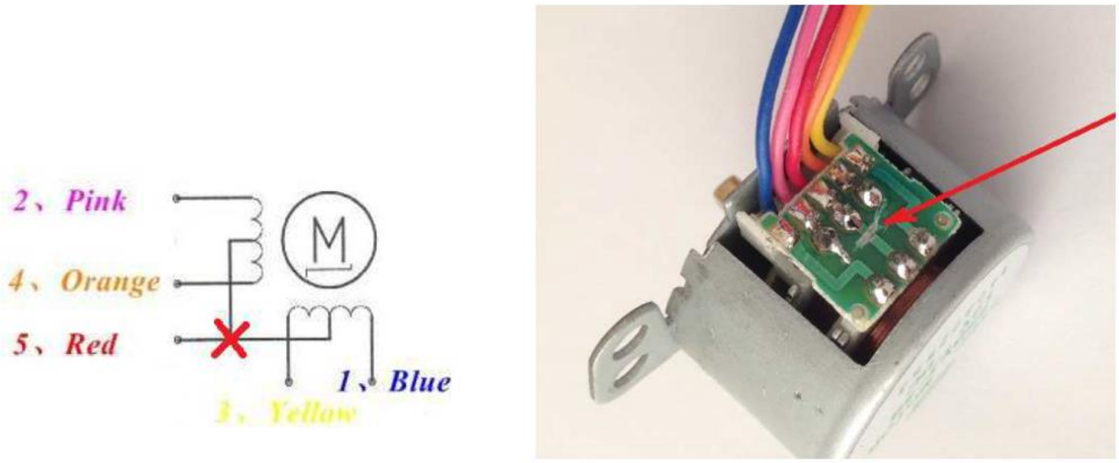


Fig. 63. 28BYJ-48 Modified

3.3.1.3. Servomotor

A servomotor is an electromechanical system that allows controlling the angular position of the axis of rotation. It is designed to rotate a certain number of degrees and then stay in that position [23]. They are characterized by having a high torque.
A servomotor is made up of:

- DC motor: Generates the rotational movement.
- Gears: It is made up of gears by which the speed and torque of the DC motor can be varied.
- Microcontroller: It is the system that allows the motor to be controlled through electrical impulses.
- Position potentiometer: It is connected to the central shaft and allows one to always know the exact position of the motor shaft.

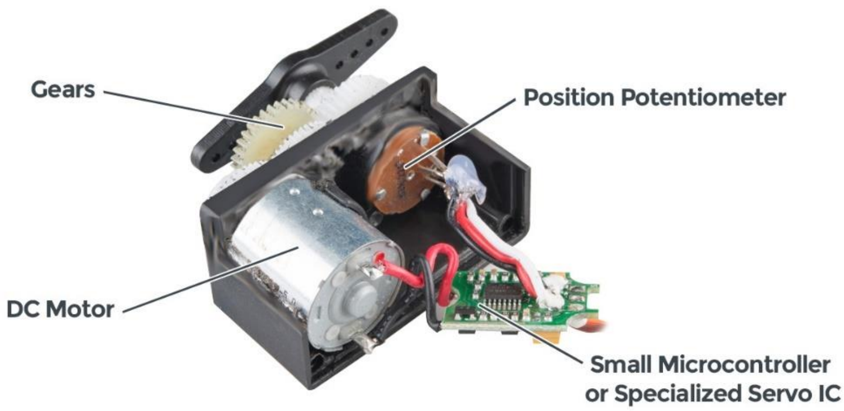


Fig. 64. Servomotor parts
In this project, a servomotor is used to control the opening of the gripper. This motor is ideal for this since it allows us to know and control the gripper's position with great precision.

3.3.1.4. Limit Switch

Within the electronic components, there is the limit switch or endstop. They can be electronic, pneumatic, or mechanical devices. In this case, electrical sensors have been used. These are located at the end of the path or of a mobile element. For example, in the robotic arm, they have been used to limit the maximum of the X, Y, and Z axes to send signals that can modify the state of a circuit. Internally they can contain normally open (NO), closed (NC) switches, or switches. NA has been used for this project. Its means that when the switch is closed, it sends a high-level signal to the microcontroller [24].
The robotic arm has 3 limit switches, one for each X, Y and Z axis.

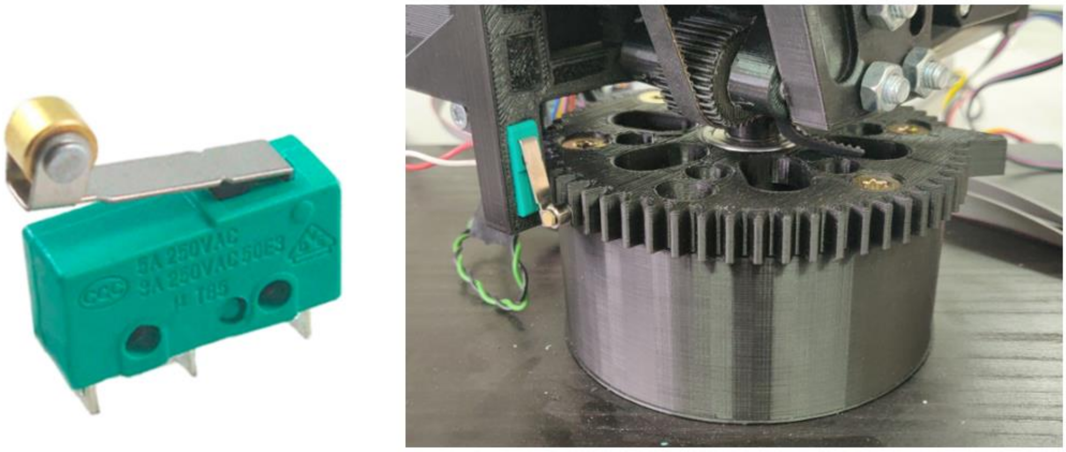


Fig. 65. Limit Switch

3.3.1.5. Arduino

Arduino is an open-source electronics creation platform based on free hardware and software, flexible and easy to use for creators and developers [25]. This platform allows us to create different types of single-board microcomputers. In order to understand what Arduino is, we must first explain the concepts of free hardware and software. Free hardware is devices whose specifications and diagrams are publicly accessible. Anyone can replicate it, and free software are computer programs whose code is accessible by anyone so that whoever wants to use it and modify it.
Arduino is a board based on an ATMEL microcontroller. Microcontrollers are integrated circuits in which instructions can be recorded, which creators write with the programming language you can use in the Arduino IDE environment. The Arduino microcontroller has an input interface, a connection in which we can connect different types of peripherals on the board. It also has an output interface, which is responsible for carrying the information that has been processed in the Arduino to other peripherals. There are many different types of Arduino boards, which can be seen in the Figure 56.

Fig. 66. Arduino types

The Arduino Mega 2560 board has been used to carry out this project, which is the most powerful of all. The microcontroller that uses this board is the ATmega2560, and the board has 54 digital inputs/outputs (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs, a 16Mhz crystal, USB connection, DC power jack, ICSP connector, and a reset button [26].

3.3.1.6. Ramps

The name RAMPS is given by its acronym "Raprap Arduino Mega Pololu Shield". They are shields for Arduino MEGA designed to control stepper motors, generally NEMA, using POLOLU A4988 or DVR8825 drivers. This board is placed on top of the Arduino Mega using all its pins, and which is prepared to control different types of components, such as stepper motors, LCD screens, limit switches, etc [27].

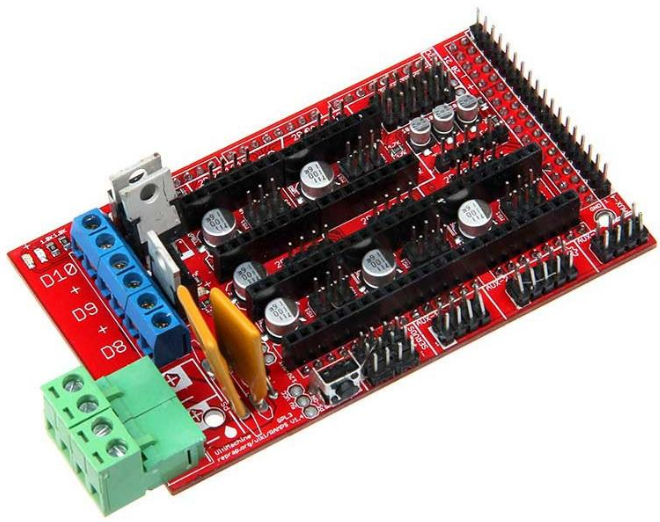


Fig. 67. Ramps 1.4

3.3.1.7. Driver DRV8825

The DRV8825 is a driver that simplifies the use of stepper motors from a microprocessor like Arduino. These drivers allow to handle the high voltages and currents that these motors need, limit the current that circulates through the motor, and provide the necessary protections to prevent the electronics from being damaged [28].

Table 3 shows the characteristics of this driver.

Colour	Purple
Maximum current	2.5A
Maximum voltage	45V
Microsteps	32
Normal Rs	0.1
Formulas	I_max = Vref / (5 * Rs)
	Vref = I_max * 5 * Rs

Table 3. DRV8825 characteristics

Figure 68 shows the image of the DRV8825 Driver

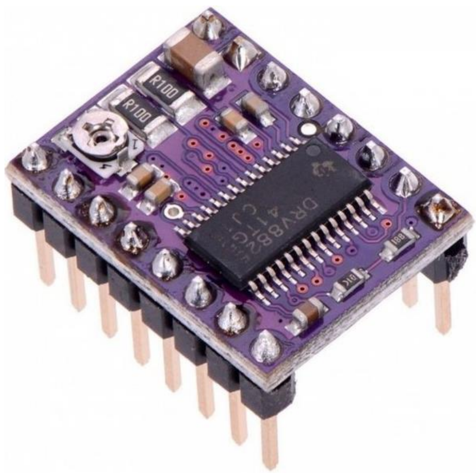


Fig. 68. DRV8825 Driver

3.3.1.8. LCD Display

LCD is the acronym for Liquid Crystal Display. These are devices used to display different types of content or information graphically through different characters, symbols, or drawings.
The screen used in this project to indicate the axes position of the robotic arm is a 20x4 LCD, which means that the liquid crystal screen has four rows of twenty characters each [29]. The main advantage of this type of display is that they consume very little energy, and its programming is relatively simple. For this project, a module has been used that incorporates an LCD, an SD card reader, and a rotary encoder. The card reader is to enter the G-CODE directly through an SD without the need for a computer. Moreover, the rotary encoder is used to scroll through the marlin menu, to be able to control the motors manually or execute a program that is on the SD card.
The module used is the LCD 2004 SMART CONTROLLER, this is designed to connect directly to a RAMPS, so it is perfect for this project.

Fig. 69. LCD 2004 SMART CONTROLLER

3.3.2. Electrical schematic

Figure 70 shows the electrical schematic of the robotic arm, as it is possible to observe the Drivers are in purple in their corresponding position, each stepper motor needs one of these to work. It can also be seen that 3 endstops have been used, one for each main axis. The power supply generates two different voltage levels, 5V to power the Arduino Mega and 12V to power the rest of the components. And finally, the 2004 LCD Display comes directly with a connection system that uses all the marked pins, thus eliminating the possibility of connecting them incorrectly.

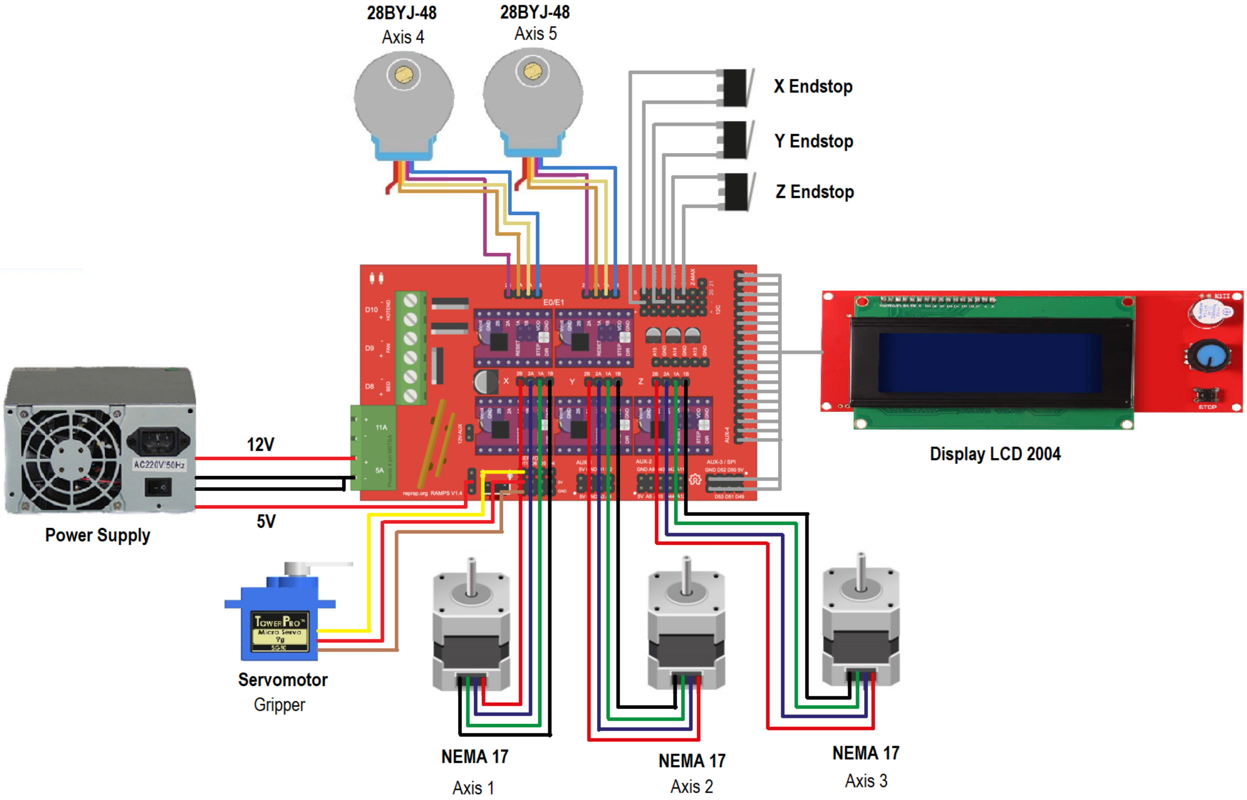


Fig. 70. Electrical schematic

3.4. Programming

The firmware that the robotic arm uses is Marlin. This firmware is the same as that used in 3D printers, which allows us to move all the axes of the robotic arm through a G-CODE.
Marlin is an open-source firmware based on the Arduino platform. The firmware runs on the robot arm's control board and manages all the arm's real-time activities.
Marlin is preconfigured to be used in 3d printers, but with a few changes, it can be adapted to any robotic arm configuration.

3.4.1. G-CODE

G-CODE, also known as RS-274, is the most widely used programming language in computer numerical control (CNC) machines. These machines can be from lathes, milling machines to 3D printers, robotic arms, etc [30].
A G-CODE file is made up of simple instructions that tell the machine what operations to do. For example, move some part, activate, or deactivate outputs.
The instructions that can be received are standardized, and for this, the following rules are followed. Each function in the code is represented by a specific letter. Below is a table with the main letters and their function.

COMMAND LETTER	FUNCTION
G	Motion
X	Horizontal distance
Y	Vertical distance
Z	Depth
F	Feed rate
S	Spindle speed
T	Tool selection
M	Miscellaneous functions
I and J	Incremental center of an arc
R	Radius of an arc

Table 4. G-CODE characters – function

Below, in Table 4 is a list of the most common standardized instructions.

COMMAND	FUNCTION
M0	Stop
M1	Sleep
M2	Program's end
M70	Show message on screen
M112	Bed temperature
M280	Set or get the position of a servo.
G0	Rapid linear movement
G1	Controlled movement
G4	Pause
G21	set units to millimeters
G28	Move home
G90	Absolute positioning
G91	Relative positioning
G92	Set position

Table 5. G-CODE main commands

Using G-CODE, it is possible to write almost any program for a robotic arm, since it can move all its axes, linearly, non-linearly, waiting a certain time, activating digital signals and many more functions, which make the use of G-CODE for a robotic arm is one of the best possible options.

3.5. Accuracy testing

This section shows the results obtained in the accuracy test (repeatability). This test consists of repeatedly doing the same program and calculating the deviation of the end point in each of the experiments from the real point. To be more specific, the experiment was carried out by placing a marker on the gripper, to draw a point on a paper where the end point was placed.

Figure 71 shows the graph in which the data of the experiment are represented. As can be seen, the first experiment had no error, and as the number of experiments increased, the error increased. It can also be observed that the deviation between the distances in the final experiments is greater, causing greater uncertainty and making it impossible to predict the error, because if we increase the number of experiments the deviation between the points will also increase.

These results are possibly due to a lack of components to determine the exact position, such as an encoder, it can also be a failure of axes 5 and 6 since they do not have endstop to determine the starting position. Another possible problem is the error that the endstop can generate because once pressed it has a certain margin of movement. And for all these reasons it is possible that the error increases with the number of experiments.

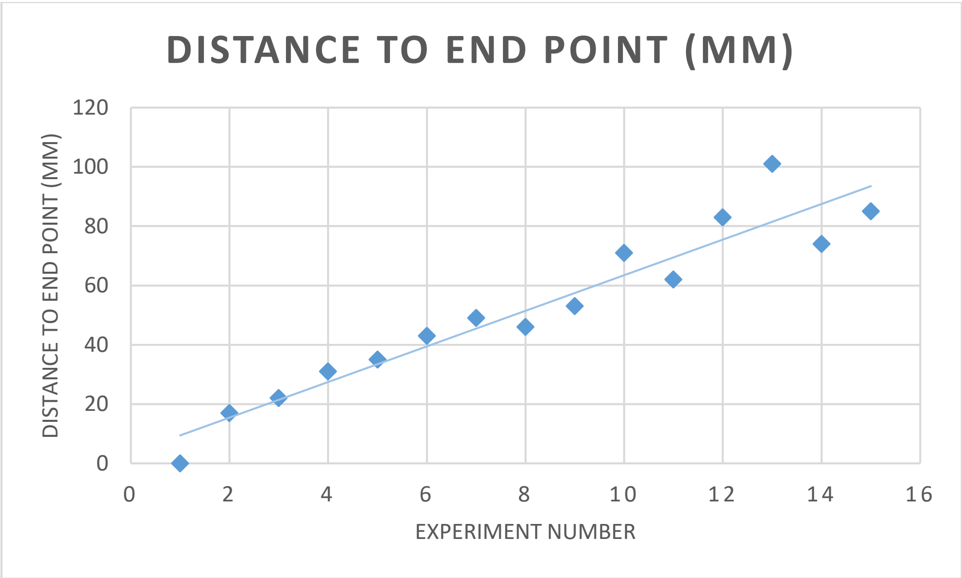


Fig. 71. Accurancy testing graph

Conclusions

1. The successful simulation demonstrates that the robotic arm is capable of carrying out all the programmed trajectories and with high precision. The clamp design fulfills its function perfectly; it is simple, modular and can be used for any type of syringe. This opens a wide range of possibilities and future improvements. There is also the possibility of utilising these robotic arms within the hospital and thus the profitability of using these robotic arms for vaccination would multiply, because once the vaccination process is finished, they could be used for different purposes within a hospital or health center.
2. The robotic arm built at Kaunas University of Technology could be a cheaper alternative to ABB's IRB120 arm. Drawback, 3D printed arms have is their precision, which is not good enough, as can be seen in section 3.5. "Accuracy testing", to be able to use them directly to give vaccines. In my opinion, it is possible to find a middle point between the industrial arm and the 3D arm, that is, to try to manufacture an arm with less demands (payload, size, materials) than those of an industrial arm but that maintains its high precision. It would be a great advance for the health sector to have a type of autonomous robotic arm that performs the most repetitive and tedious tasks for health personnel.
3. The proposed vaccination process still has certain parts that are not automated and that can be automated. One of them is the distribution of syringes and vials. It is possible to create a distribution device for these. Another possible improvement would be to reduce the degrees of freedom of the robotic arm to make it more economical. This can be carried out by utilising a machine that directly prepares the syringes already loaded with the vaccine, and these are distributed to the tables, so the robotic arm would only have to pick it up and inject it. Another possible improvement would be to apply Artificial Intelligence (AI) to patient detection and the location of the exact vaccination point, thus allowing vaccines to be given in different situations and not only in the chair with the arm support. And what is possibly the greatest improvement within the world of vaccination, would be to create a machine that is available in hospitals, and that is capable of injecting any vaccine, taking into account the medical record of the patient and being able to inject different types of vaccines in the same day. In short, the possibilities for innovation in this field are endless and, as they say, "the sky's the limit."

List of references

[1] "EU Could Meet 70% Vaccination Target By Late July If Production Steps Up", Spglobal.com, 2021. [Online]. Available: https://www.spglobal.com/ratings/en/research/articles/210211-eu-could-meet-70-vaccination-target-by-late-july-if-production-steps-up-11823334.

[2] "Medical robot - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Medical_robot.

[3] "Robotic surgery - Robotic procedures", Encyclopedia Britannica, 2021. [Online]. Available: https://www.britannica.com/science/robotic-surgery/Robotic-procedures.

[4] M. Abedin-Nasab, Handbook of Robotic and Image-Guided Surgery, 1st ed. 2019.

[5] "Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective", Biomedcentral. [Online]. Available: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-018-0383-x

[6] Eecs.ucf.edu, 2021. [Online]. Available: http://eecs.ucf.edu/~abehal/AssistiveRobotics/presentation/UCF_MANUS_ARM_082009.pdf

[7] "Ursus | Robolab", Robolab.unex.es, 2021. [Online]. Available: https://robolab.unex.es/index.php/2011/01/13/ursus/.

[8] "Syringes and Needles For Sale | Medicine Syringes | Injection Needles | Vitality Medical", Vitalitymedical.com, 2021. [Online]. Available: https://www.vitalitymedical.com/syringes-needles.html.

[9] Apps.who.int, 2021. [Online]. Available: https://apps.who.int/iris/rest/bitstreams/1327517/retrieve.

[10] "Safe and Proper Sharps Disposal During the COVID-19 Mass Vaccination Campaign | CDC", Cdc.gov, 2021. [Online]. Available: https://www.cdc.gov/vaccines/covid-19/training-education/safe-proper-sharps-disposal.html.

[11] "AutoCAD - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/AutoCAD.

[12] "RobotStudio downloads", Robotics, 2021. [Online]. Available: https://new.abb.com/products/robotics/robotstudio/downloads#:~:text=RobotStudio%20is%20ABB's%20simulation%20and,in%20your%20production%20line%20remotely.

[13] "IRB 120 data - Industrial Robots (Robotics)", Robotics, 2021. [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/irb-120/irb-120-data.

[14] "3D printing - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/3D_printing.

[15] W. https://www.zarr.com, "Preparing SOLIDWORKS Models for 3D Printing", Solidsolutions.co.uk, 2021. [Online]. Available: https://www.solidsolutions.co.uk/blog/2015/04/preparing-SOLIDWORKS-models-for-3d-printing/#.YKV2NLUzbD4.

[16] "Creality CR-20 Pro: Review the Specs", All3DP, 2021. [Online]. Available: https://all3dp.com/1/creality-cr-20-pro-review-3d-printer-specs/.

[17] "Plastic Material for 3D Printing: PLA", Sculpteo, 2021. [Online]. Available: https://www.sculpteo.com/en/materials/fdm-material/pla-material/#:~:text=PLA%20plastic%20or%20polylactic%20acid,cornstarch%20as%20a%20raw%20material.&text=PLA%20is%20a%20fully%20biodegradable,materials%20used%20for%20additive%20manufacturing.

[18] "Creality Slicer: The Best Slicers for Creality 3D Printers", All3DP, 2021. [Online]. Available: https://all3dp.com/2/creality-slicer-what-s-the-best-slicer-for-creality-3d-printers/.

[19] C. Service, "What do the different settings in Cura do?", Monoprice, 2021. [Online]. Available: https://monopricesupport.kayako.com/article/6-what-do-the-different-settings-in-cura-do.

[20]"Stepper motor - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Stepper_motor.

[21] "NEMA17 motor Datasheet pdf - stepper motor. Equivalent, Catalog", Datasheetspdf.com, 2021. [Online]. Available: https://datasheetspdf.com/pdf/1260602/Schneider/NEMA17/1.

[22] "28BYJ-48 Motor Datasheet pdf - Stepper Motor. Equivalent, Catalog", Datasheetspdf.com, 2021. [Online]. Available: https://datasheetspdf.com/pdf/1006817/Kiatronics/28BYJ-48/1.

[23] "Arduino - Servo Motor - Tutorialspoint", Tutorialspoint.com, 2021. [Online]. Available: https://www.tutorialspoint.com/arduino/arduino_servo_motor.htm#:~:text=Advertisements,angular%20position%20of%20the%20shaft.

[24] E. Fahad, "Arduino Uno limit switch Interfacing and Programming", Electronic Clinic, 2021. [Online]. Available: https://www.electroniclinic.com/arduino-uno-limit-switch-tutorial/#:~:text=A%20Limit%20Switch%20is%20connected,is%20a%20Pull%2DUp%20resistor.&text=When%20the%20switch%20is%20closed,give%200%20as%20the%20signal.

[25] "Arduino - Introduction", Arduino.cc, 2021. [Online]. Available: https://www.arduino.cc/en/guide/introduction.

[26] "Arduino Mega 2560 Rev3 | Arduino Official Store", Store.arduino.cc, 2021. [Online]. Available: https://store.arduino.cc/arduino-mega-2560-rev3.

[27] "RAMPS 1.4/es - RepRap", Reprap.org, 2021. [Online]. Available: https://reprap.org/wiki/RAMPS_1.4/es.

[28] "DRV8825 Stepper Motor Driver Module Pinout, Datasheet, Features & Equivalents", Components101, 2021. [Online]. Available: https://components101.com/modules/drv8825-stepper-motor-driver-module#:~:text=The%20DRV8825%20is%20a%20Motor,Robotics%2C%20ATMs%20and%20Gaming%20Machines.

[29] Beta-estore.com, 2021. [Online]. Available: https://www.beta-estore.com/download/rk/RK-10290_410.pdf.

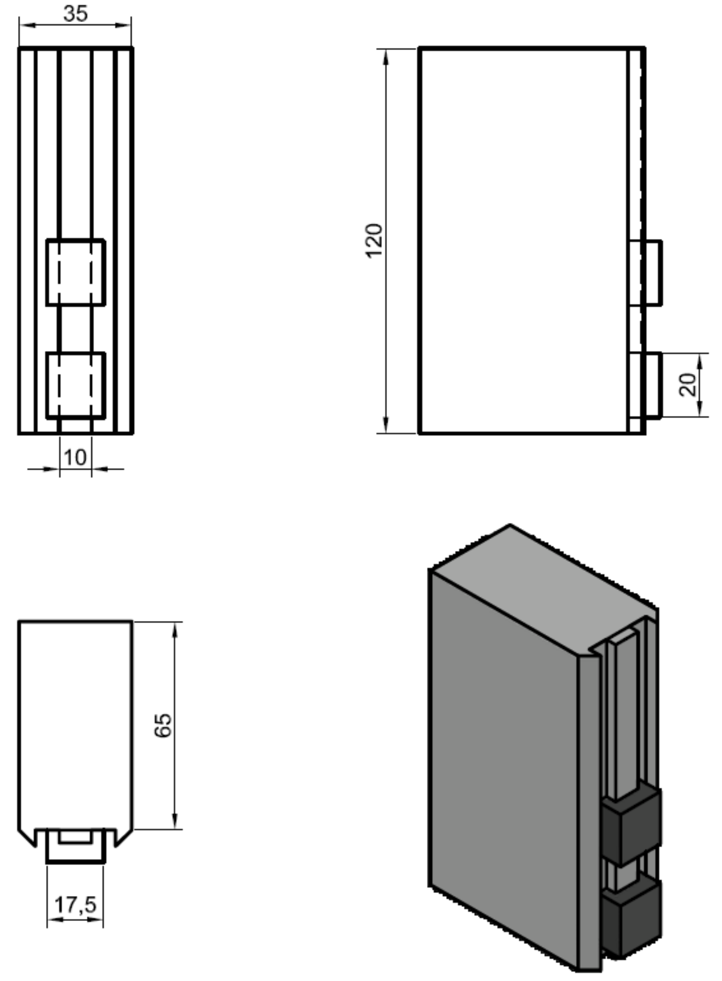
[30] "G-code - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/G-code.

List of information sources

1. Operating manual RobotStudio. https://library.e.abb.com/public/244a8a5c10ef8875c1257b4b0052193c/3HAC032104-001_revD_en.pdf
2. Technical reference manual RAPID Instructions, Functions and Data types. https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf
3. Tutorials for RobotStudio. https://new.abb.com/products/robotics/robotstudio/tutorials
4. AutoCAD 3D Tutorials. https://www.andrew.cmu.edu/course/48-568/PDFs/3D_AutoCAD.pdf

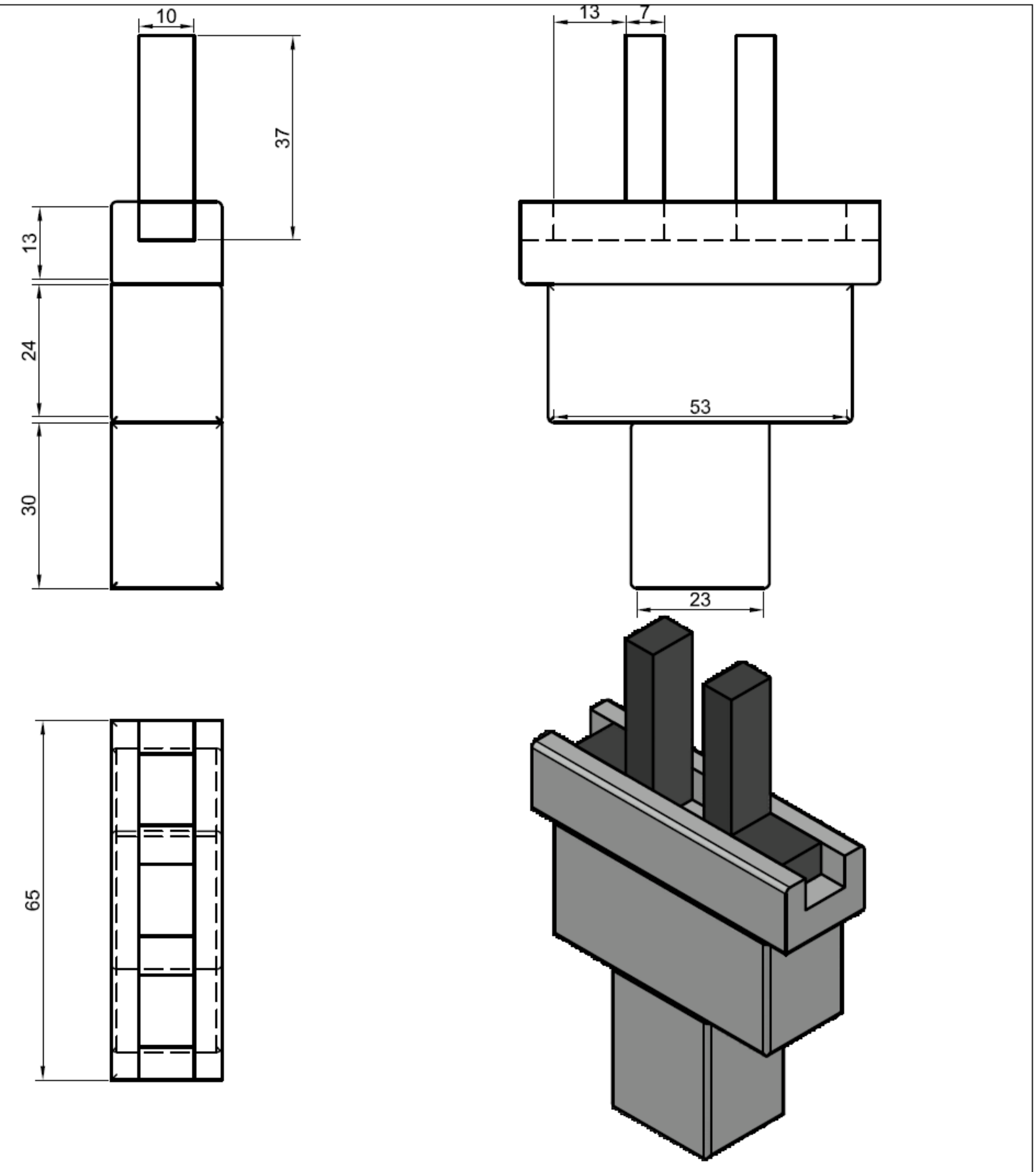
Appendices

Appendix 1. Design drawings



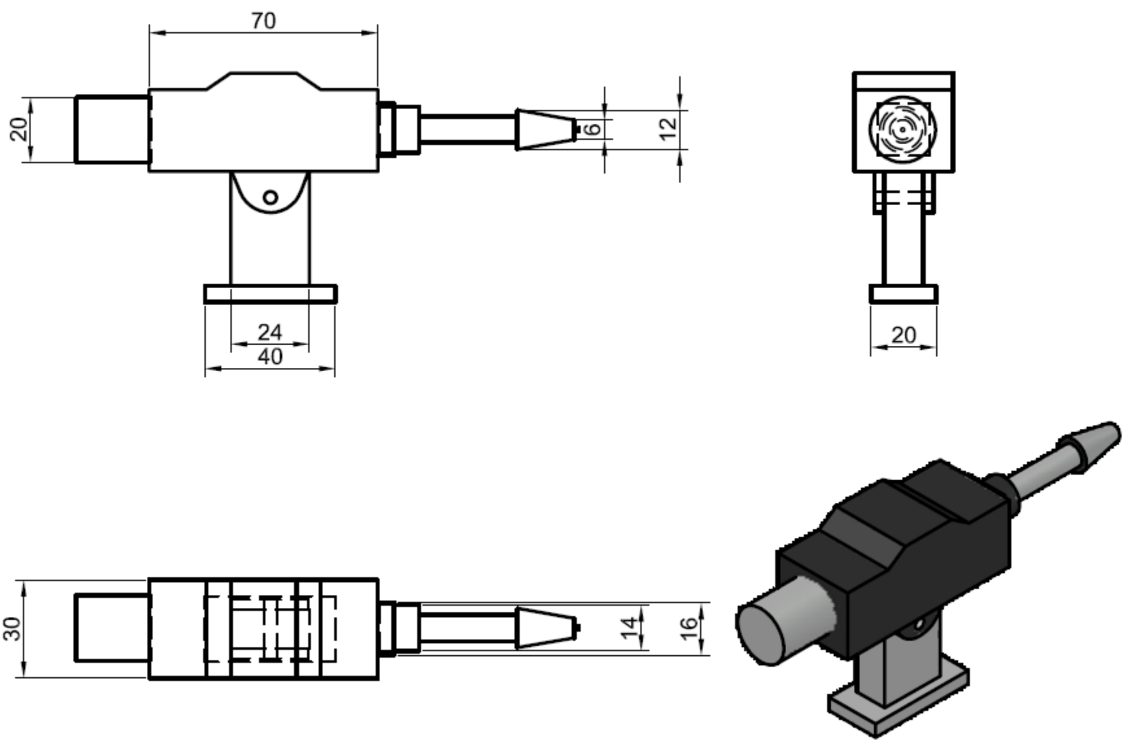
	Date	Name	TItle
Drawn	19-05-20	Fulgenclo Blázquez	GRIPPER BASE
Checked			

	DWG NO	Designed	Fulgencio Blázquez
		Standard	
		Approved	



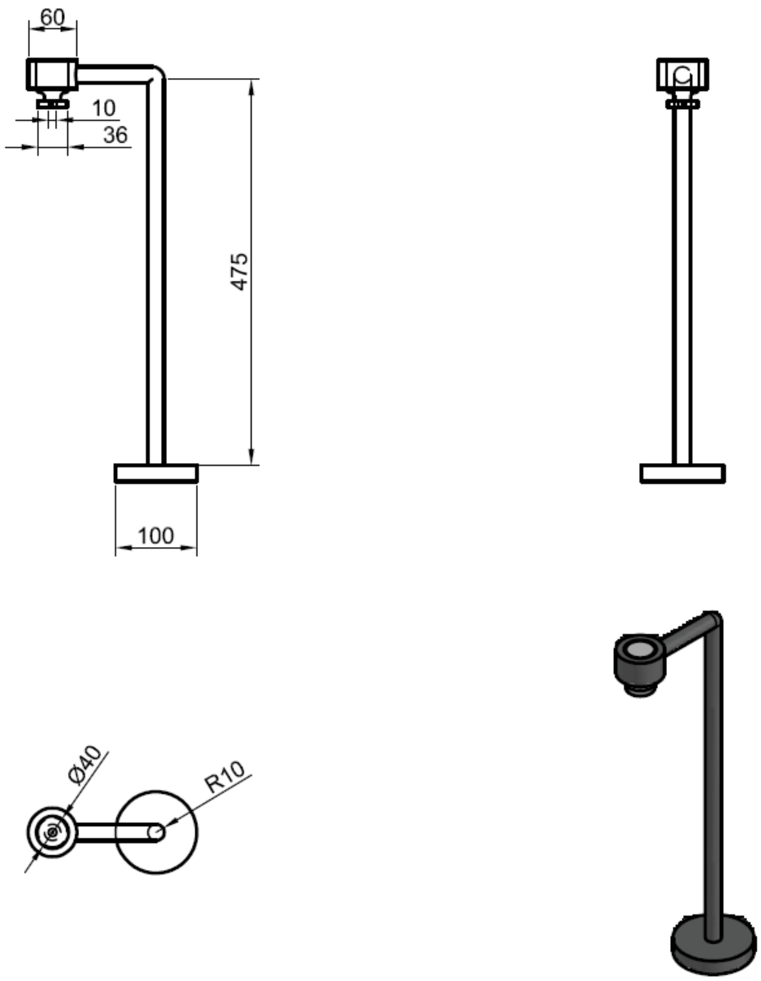
	Date	Name	TItle
Drawn	19-05-20	Fulgenclo Blázquez	GRIPPER
Checked			

	DWG NO		
		Designed	Fulgencio Blázquez
		Standard	
		Approved	

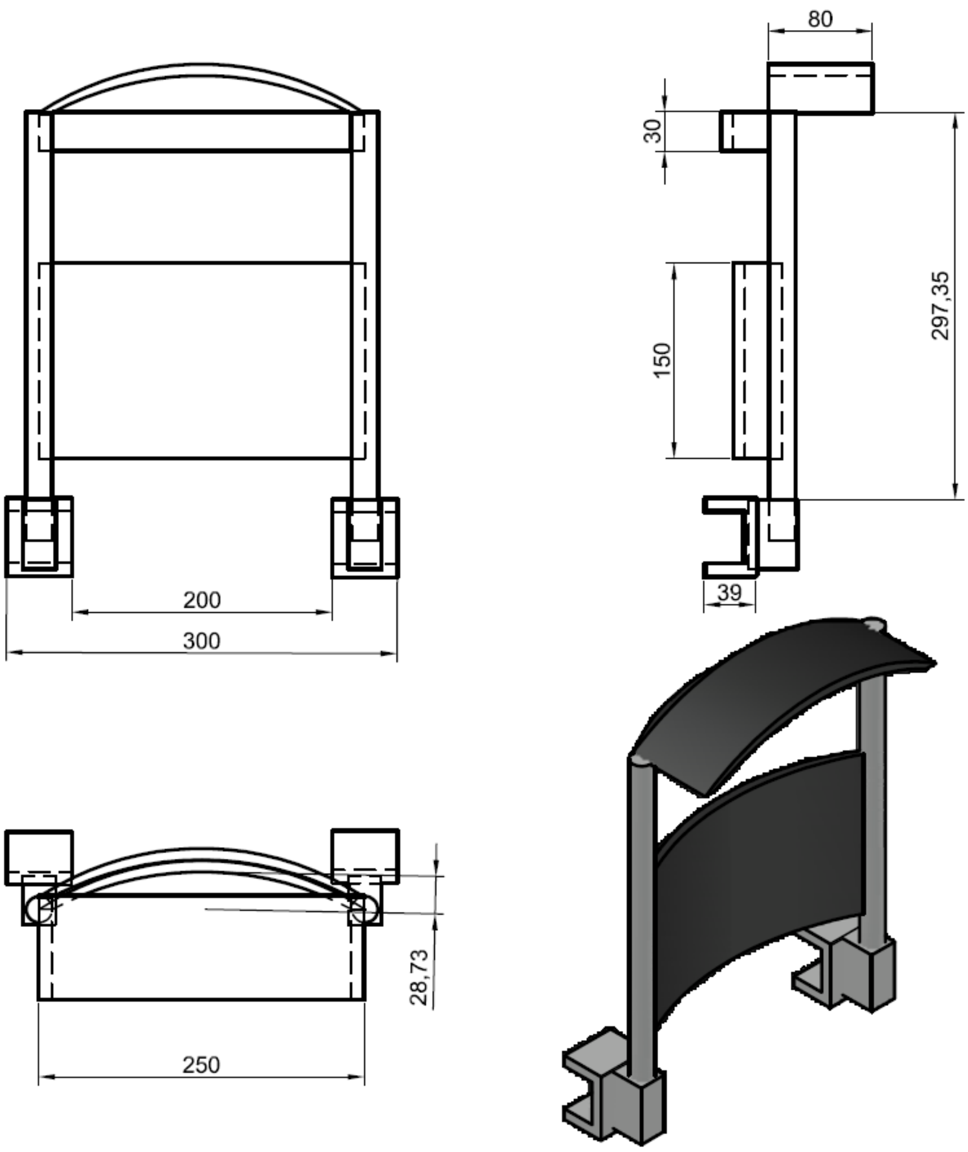


	Date	Name	TItle
Drawn	19-05-20	Fulgenclo Blázquez	DISINFECTANT SPRAY
Checked			

DWG NO		Designed	Fulgencio Blázquez
		Standard	
		Approved	

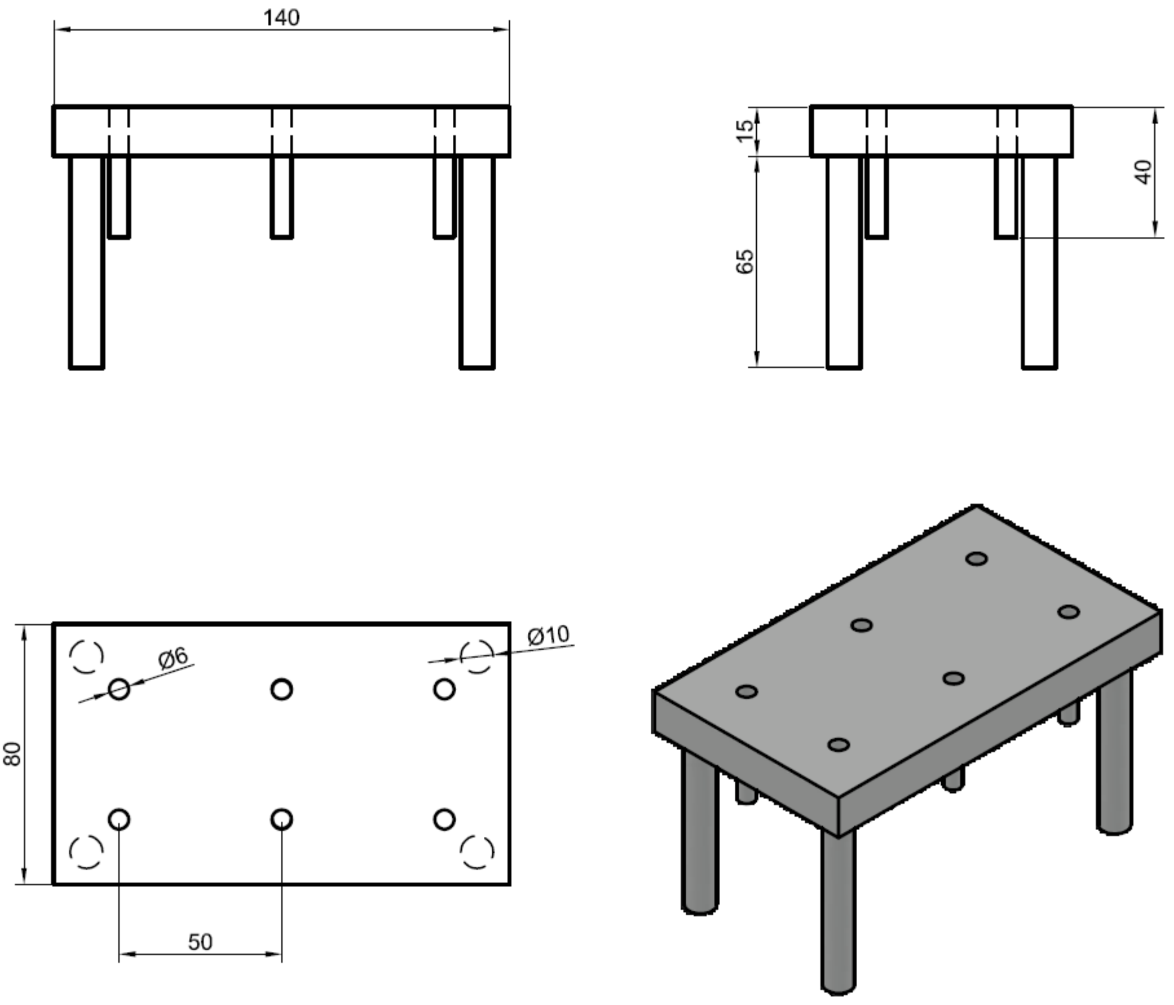


	Date	Name	TItle VIAL HOLDER	
Drawn	19-05-20	Fulgenclo Blázquez		
Checked				
	DWG NO		Designed	Fulgencio Blázquez
			Standard	
			Approved	



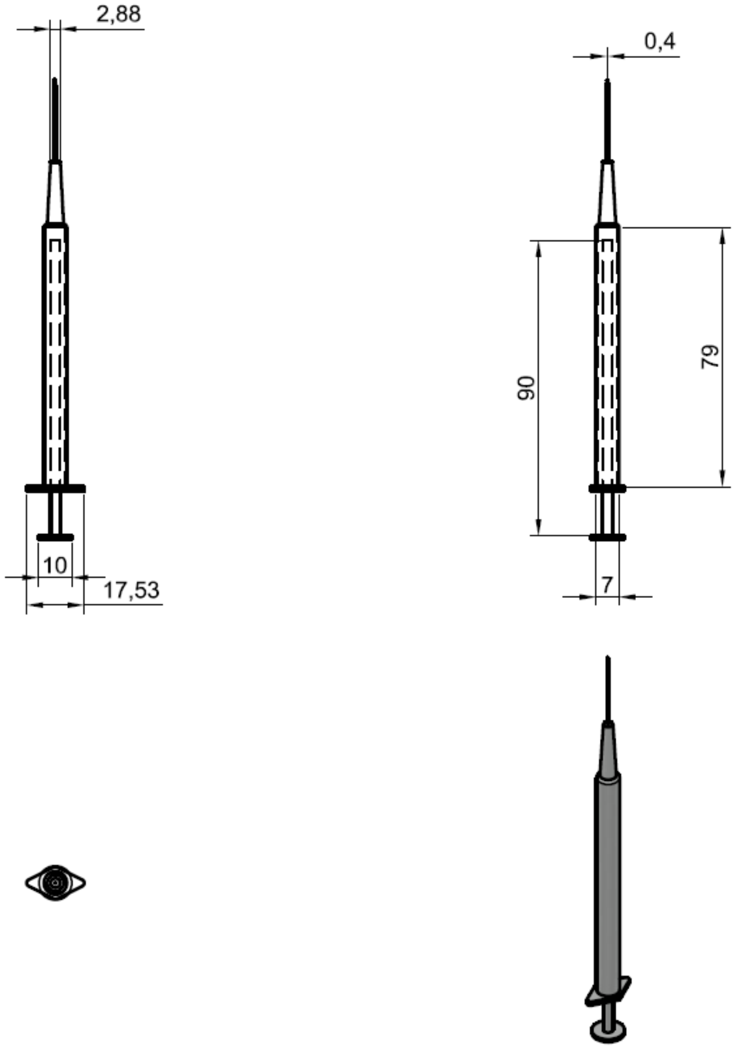
	Date	Name	Title
Drawn	19-05-20	Fulgencio Blázquez	ARM HOLDER
Checked			

	DWG NO	Designed	Fulgencio Blázquez
		Standard	
		Approved	



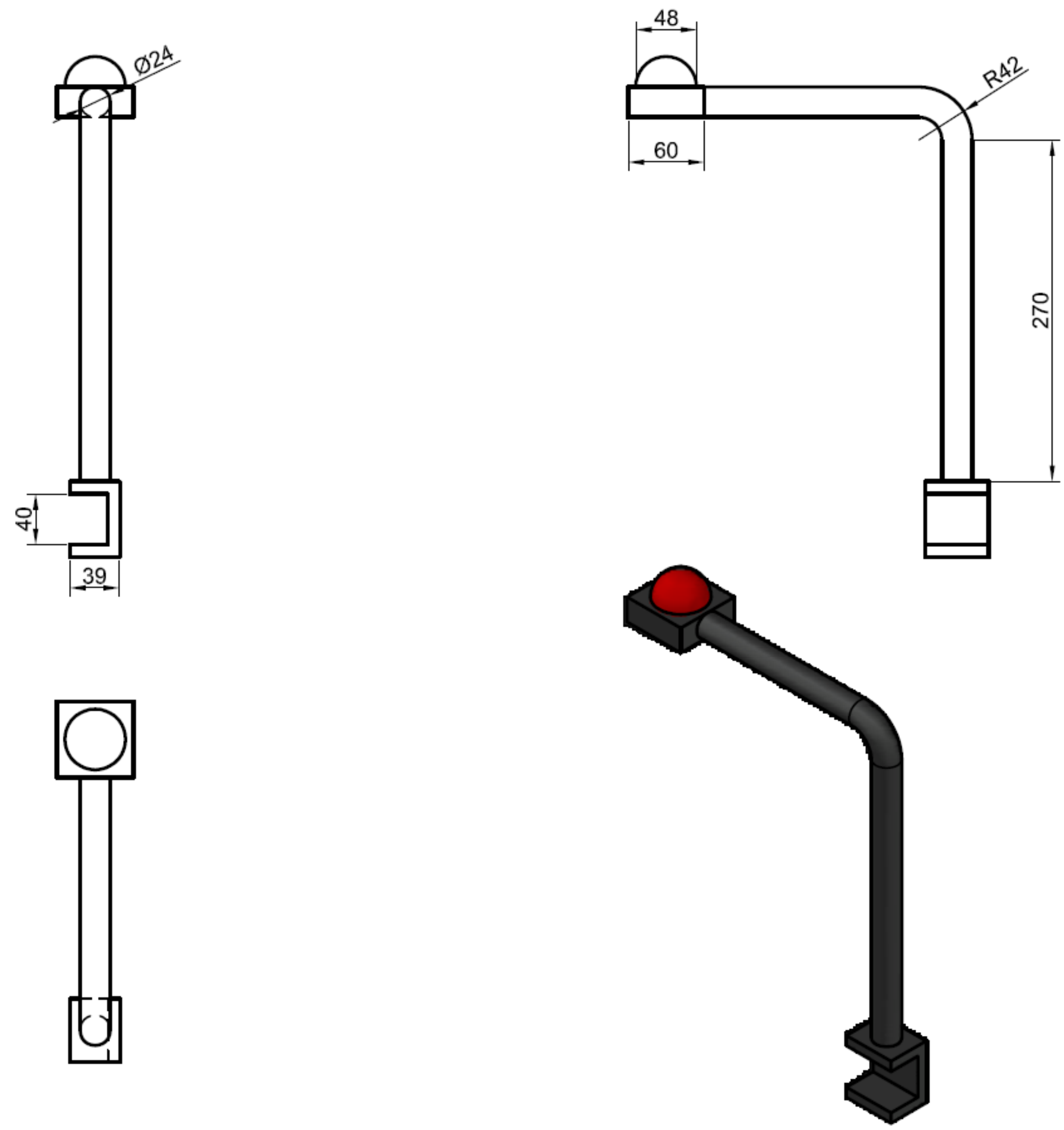
	Date	Name	TItle
Drawn	19-05-20	Fulgenclo Blázquez	SYRINGE HOLDER
Checked			

DWG NO		Designed	Fulgencio Blázquez
		Standard	
		Approved	



	Date	Name	TItle
Drawn	19-05-20	Fulgenclo Blázquez	SYRINGE
Checked			

	DWG NO	Designed	Fulgencio Blázquez
		Standard	
		Approved	



	Date	Name	Title
Drawn	19-05-20	Fulgencio Blázquez	STOP BUTTON
Checked			

DWG NO		
	Designed	Fulgencio Blázquez
	Standard	
	Approved	

Appendix 2. Full Code

```
!***********************************************************
    !
    ! Module:  Module1
    !
    ! Description:
    !   RAPID programming code for an ABB IRB120 robot to inject vaccines.
    !
    ! Author: FULGENCIO BLAZQUEZ CONESA
    !
    ! Version: 1.0
    !
!***********************************************************

    PROC main()
        !Add your code here
        VAR num i:=0;
        SetDO OPEN_GRIPPER,0;
        SetDO CLOSE_GRIPPER,0;
        SetDO EXTRACT,0;
        SetDO DISINFECTANT,0;
         FOR i FROM 0 TO 5 DO
            TEST i
            CASE 0:
                moveToSyringe1;
            CASE 1:
                moveToSyringe2;
            CASE 2:
                moveToSyringe3;
            CASE 3:
                moveToSyringe4;
            CASE 4:
                moveToSyringe5;
            CASE 5:
                moveToSyringe6;
            ENDTEST

            WaitDI go, 1;
            takingVaccine;
            puttingVaccine;
            throwSyringe;
         ENDFOR

    ENDPROC
```

```
PROC moveToSyringe1()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_20,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 2;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_30,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;
ENDPROC
PROC moveToSyringe2()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_40,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_50,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;
ENDPROC
PROC moveToSyringe3()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_60,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_70,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
```

```
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;

ENDPROC
PROC moveToSyringe4()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_80,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_90,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;
ENDPROC
PROC moveToSyringe5()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_100,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_110,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;
ENDPROC
PROC moveToSyringe6()
    MoveJ home,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_10,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveL Target_120,v200,z0,gripper2\WObj:=syringe_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1;
```

```
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_130,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_140,v200,z0,gripper2\WObj:=syringe_holder;
    MoveJ Target_150,v50,z0,gripper2\WObj:=syringe_holder;
  ENDPROC
  PROC takingVaccine()
    MoveL Target_160,v200,z0,gripper2\WObj:=vial;
    MoveL Target_170,v200,z0,gripper2\WObj:=vial;
    MoveL Target_180,v200,z0,gripper2\WObj:=vial;
    MoveL Target_190,v100,z0,gripper2\WObj:=vial;
    MoveL Target_200,v20,fine,gripper2\WObj:=vial;
    WaitTime 1;
    SetDO EXTRACT,1;
    WaitTime 2;
    SetDO EXTRACT,0;
    MoveL Target_190,v50,fine,gripper2\WObj:=vial;
    MoveL Target_180,v200,z0,gripper2\WObj:=vial;
    MoveL Target_170,v200,z0,gripper2\WObj:=vial;
  ENDPROC
  PROC puttingVaccine()
    MoveJ Target_290,v200,z0,gripper2\WObj:=arm_holder;
    WaitTime 1;
    SetDO DISINFECTANT,1;
    WaitTime 1;
    SetDO DISINFECTANT,0;
    MoveJ Target_270,v50,fine,gripper2\WObj:=arm_holder;
    MoveL Target_280,v50,fine,gripper2\WObj:=arm_holder;
    MoveL Target_320,v50,fine,gripper2\WObj:=arm_holder;
    WaitTime 1;
    SetDO CLOSE_GRIPPER,1;
    WaitTime 1.5;
    SetDO CLOSE_GRIPPER,0;
    MoveL Target_270,v50,fine,gripper2\WObj:=arm_holder;
    MoveJ Target_300,v200,z0,gripper2\WObj:=arm_holder;
  ENDPROC
  PROC throwSyringe()
    MoveJ Target_310,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_250,v200,z0,gripper2\WObj:=Table;
    WaitTime 1;
    SetDO OPEN_GRIPPER,1;
    WaitTime 1;
    SetDO OPEN_GRIPPER,0;
    MoveJ Target_250,v200,z0,gripper2\WObj:=Table;
    MoveJ Target_310,v200,z0,gripper2\WObj:=Table;
  ENDPROC
ENDMODULE
```