DEMETER PROJECT

13/03/2021

ROBOTIC ARM UNIT

"The way that the robotics market is going to grow, at least in the home, is that we'll have a number of different special purpose robots". **Colin Angle**.

Co-funded by the Erasmus+ Programme of the European Union

INDEX

UNIT: CREATION OF A ROBOTIC ARM

“The way that the robotics market is going to grow, at least in the home, is that we'll have a number of different special purpose robots”. **Colin Angle**.

From the teachers of this unit, we thank Florin Tobler because this documentation has been prepared using his robotic arm.

Product: Creation of a robotic arm using a 3D printer and arduino

- Country: Spain
- Class: 1º SMR
- IT Teacher: (Assembly and maintenance of equipment).
- IT Teacher: (Single user operating systems).

In this unit, the main goal is to combine the following concepts:

- **TO KNOW**. Know the theoretical concepts of the activity to be carried out. To be a good professional, the student must learn all possible theoretical knowledge to be able to carry out their work in the world of work.
- **KNOW TO DO/KNOW HOW**. Know how to apply theoretical knowledge and deal effectively and efficiently with the devices and technology used. When the student combines knowledge and know-how, the result of their work will be of a high level.
- **KNOW TO BE**. Attitude towards the project. Work collaboratively with colleagues and support each other. Collaborate with students from other countries. In many jobs, motivation, availability, the desire to do a good job are highly appreciated even more than knowledge.

In any profession, a professional makes use of these three ingredients in their daily work. Depending on the job, more of one will be needed than the other, but what is undeniable is that when it comes to training we must keep it in mind.

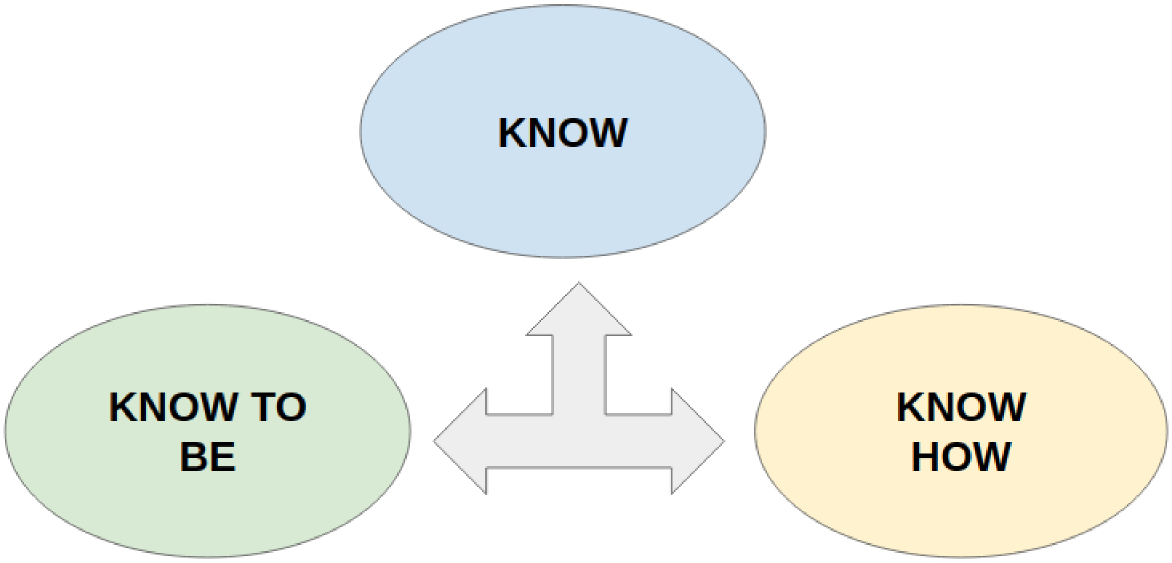

Figure. A good professional is a combination of know, know how and know to be.

In each of the lessons the different knowledge will be labeled so that the teacher takes this into account when teaching it.

Methodology followed in this unit

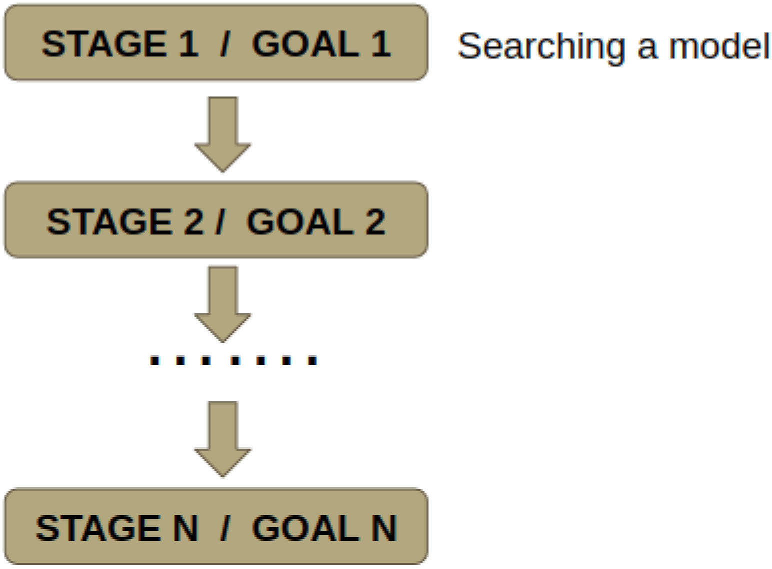

Figure 2. Define the objectives.

The teacher will be responsible for defining the objectives and preparing the corresponding lessons to achieve these objectives. Each stage or lesson will have specific objectives and some of them could be evaluated in a formal way, others in an observational way.

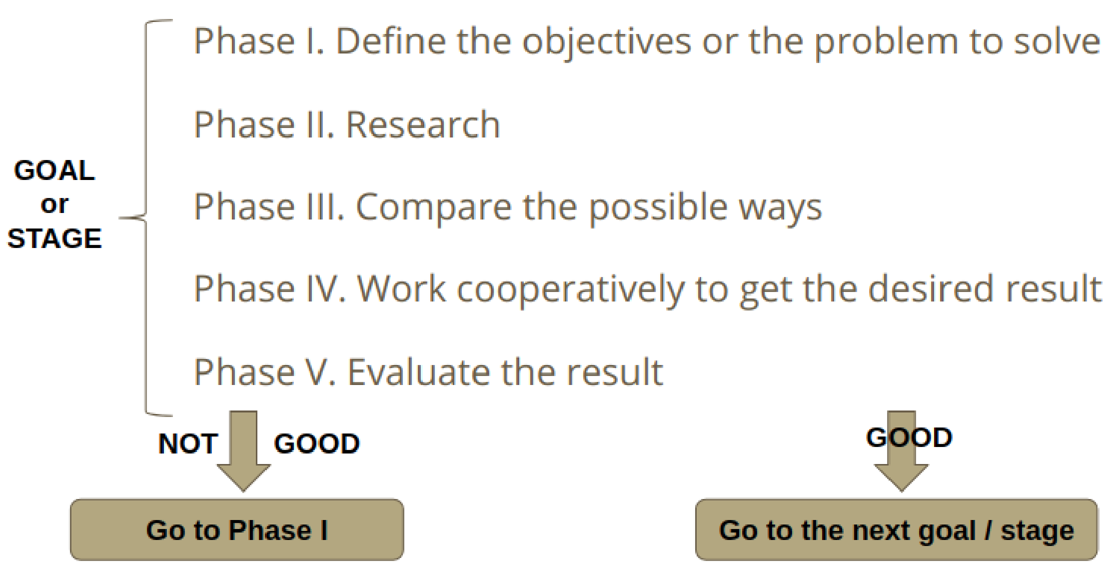

Figure . Way of working the stages.

Each of these stages will be worked in an evolutionary way and divided into phases in such a way that if it is observed that the result obtained is not satisfactory, it can be repeated but this time with more knowledge on the subject in such a way that the result be as expected.

In our case, the first phase of each stage begins with a brainstorming in which the objectives or the problem to be solved are going to be defined. In some cases, an investigation or search for information must be carried out because generally in a new or complex field there must always be a theoretical or theoretical-practical support.

In a collaborative way, different solutions, programs, ways of doing things will be proposed and they will be compared to establish which is the best approach. Once the plan to follow is clear, we will work collaboratively to obtain the result we want.

The last phase of each stage will always be to evaluate the result. It must always be kept in mind that the process is always the most important because it is the place and time in which knowledge is acquired and fixed.

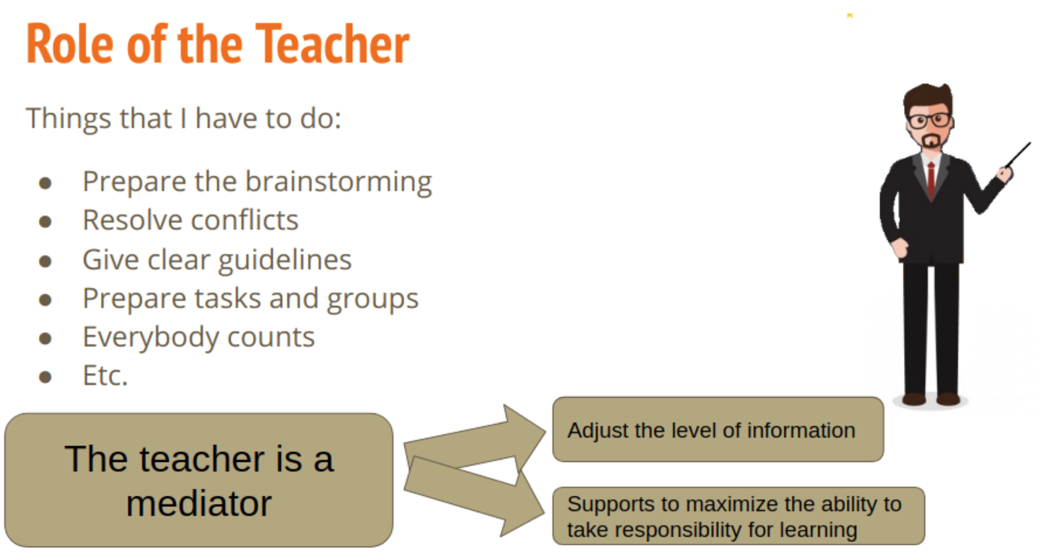

Figure. Rol of the teacher

In our opinion, the teacher when working on projects of this type should adopt a mediating role. Mediator or organizer of the work, leaving the student clear work guidelines. He will be responsible for creating the working groups and assigning responsibilities to each of the members.

Conflicts always arise in this type of project but the teacher will be in charge of solving them taking into account that all students are the main characters "everybody counts". As in a company, all students are essential and have their responsibility and leadership.

It is important to focus the investigation and objectives. The teacher will adjust the level of information to the student but he won't solve the problems. In this way, the student can be made responsible for his/her own learning.

Based on experience, we can say that the teacher should not let the students organize themselves or make their own teams. Collaboration yes, but with well-organized work and clear responsibilities.

Tools and things needed

Filament. Remember that you are going to use a little less than half a kilo of filament. Later we will explain the different filaments that you can use. Use PLA or PETG (better PETG than PLA).

Figure. PLA filament.

A 3D model. We have used the model of Florin Tobler (https://www.thingiverse.com/thing:1718984) that is licensed under CC-Attribution-Non-Commercial in the thingiverse repository.

Figure 7. 3D model by Florin Tobler

3D Printer. There are thousands of printers on the market but we recommend a REP-RAP printer (https://reprap.org/wiki/RepRap). ¿Why?

- Because they are low-cost printers.
- If a plastic component spoils you, you can print it and replace it.
- The other non-plastic components can be easily found on the Internet if they are damaged.
- You contribute to more open-source printers.
- The price-quality ratio is excellent.
- There are many people who use these printers. If you have a problem they can help you.
- You can have a fun and instructive time with your students by setting up the printer.

Arduino Mega 2560. Here you have a description of this controller extracted from (arduino.cc) "The Arduino MEGA 2560 is designed for projects that require more I/O lines, more sketch memory and more RAM. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities maintaining the simplicity and effectiveness of the Arduino platform. This document explains how to connect your Mega2560 board to the computer and upload your first sketch."

RAMPS 1.4. RAMPS is RepRap Arduino Mega Pololu Shield, a shield (a board that is connected on your arduino). RepRap printers use this shield because it is a low cost device to control all the NEMA step motors of a 3D printer. The RAMPS not only controls the stepper motors but also controls the hot bed and hot-end thermistors as well as the x, y and z axis limit switches. You will have to fit in the RAMPS the Pololu A4988 drivers. A driver will be needed for each axis of the printer or in this case the mechanical arm.

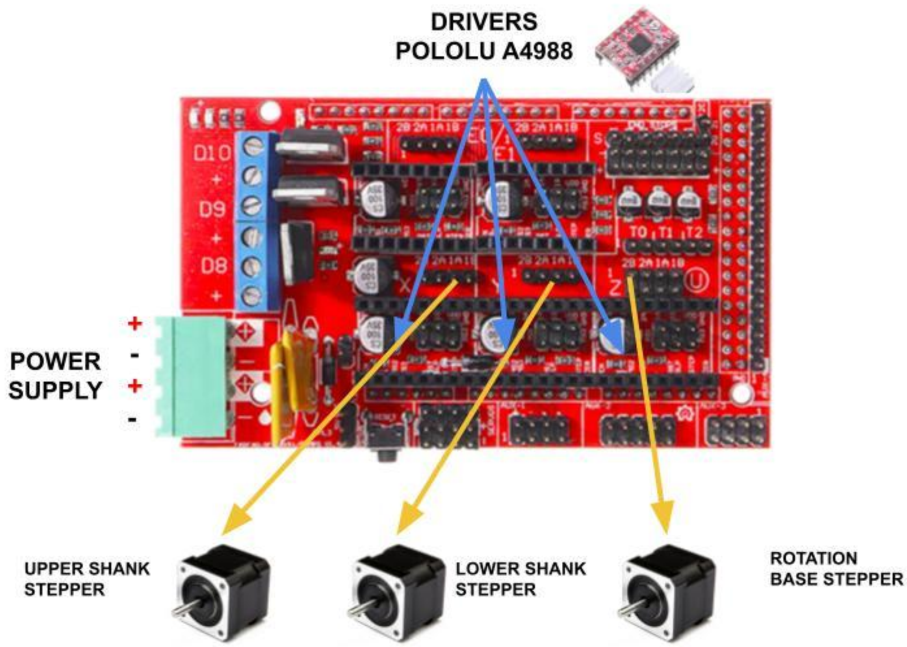

Figure. RAMPS 1.4 and wiring diagram.

Nema 17 Stepper motor. The NEMA 17 motor is a stepper motor so named because it is 1.7x1.7 inches wide and tall. These types of motors are very precise. In our case, we use this type of motor for its reliability, precision and because they are very powerful. The weak point of these motors is the speed but that is what worries us the least since we need our arm not to be fast, we prefer it slow.

Bearings (flanged, normal and thrust bearings). We will need these types of parts for all moving parts.

M3 and M4 screws and nuts, M5 threaded rod & nuts, wires, cable ties, etc.

Computer. It doesn't matter what operating system you are using.

Arduino IDE. You will need this program to write the code in the Arduino Mega. You can also modify the code if you think it is necessary. If you don't have much idea of how this program is installed you can follow the guide that you will find in(https://www.arduino.cc/en/Guide/ArduinoMega2560).

Lesson 1. 3D printers and printing materials (TO KNOW).

Lesson objective: To know the different components of a 3D printer and the different printing materials (PLA, ABS, PETG, etc.) as well as their origin (organic, oil, etc.)

This is a theoretical lesson that explains how a 3D printer and thermoplastic printing materials work.

Begin the lesson by explaining the electronic elements of a printer. It is essential to carry out the explanation with a printer in front of you to be able to see each and every one of the components and explain how they work.

Students must previously have knowledge of basic electronics and Arduino.

The different types of existing thermoplastics, their characteristics and properties and their application will also be explained. In our case we will rely more on PLA since it is an easier material to work with and also, as it comes from organic materials, it is more recyclable than, for example, ABS that comes from oil for example.

It is true that PLA is less recommended than ABS for mechanical parts, but by giving it the necessary thickness and infill, parts will be solid enough to support the robotic arm that is going to be generated.

PLA filaments are biodegradable and have excellent mechanical properties as well as high quality in 3D printing. It is the ideal filament for those who start printing due to its great ease of printing. It does not have warping.

ABS is the acronym for Acrylonitrile Butadiene Styrene. This filament is one of the most used thermoplastics in 3D printing. It is a very hard and rigid material, and has great chemical resistance to abrasion. It is soluble in acetone. It is not biodegradable, and it degrades a lot if it is exposed to the sun. It requires a hot bed between 80ºC and 100ºC. LEGO pieces are made with ABS.

PETG filament has low shrinkage, which makes it easy to print. People like this filament because you can get very shiny, translucent pieces. It has a great chemical resistance and compatible for food use. It has a great resistance to sunlight, making it perfect for outdoors.

ASA It is a filament with exceptional mechanical resistance to the elements, both to UVA rays and to water. If you are going to use plastic for a piece that is going to be outdoors or is going to be hit, ASA should be the choice since it maintains its color and resistance to impact, even after long-term outdoor use.

How can the student's work be evaluated?

By taking a test or orally through questions

Lesson 2. STL files (TO KNOW and KNOW HOW)

Lesson objective: Know what are STL files and create a simple 3D design.

In this part of the lesson it will be explained what is an STL file that is neither more nor less than a CAD file which only contains the geometry of a 3D object. It is a very simple file which does not contain information about the texture, color or other types of properties that other CAD files do have.

In order for the student to experiment with 3D design in a simple way, a practical exercise has been proposed which consists of reproducing a 3D design in the tinkercad.com program and generating an STL file of a well-known place in the town such as the temple of the “Playa del Angel”.

Figure 10. Photo collected from Diariosur.es

The objective is for the student, using the tinkercad 3D design tools, to generate a replica of the template with the Tinkercad tools (tinkercad.com). This online program is very easy to use and very intuitive, which allows the teacher that with a minimum of explanation the students work on a simple design.

The result of one of our students is the following one:

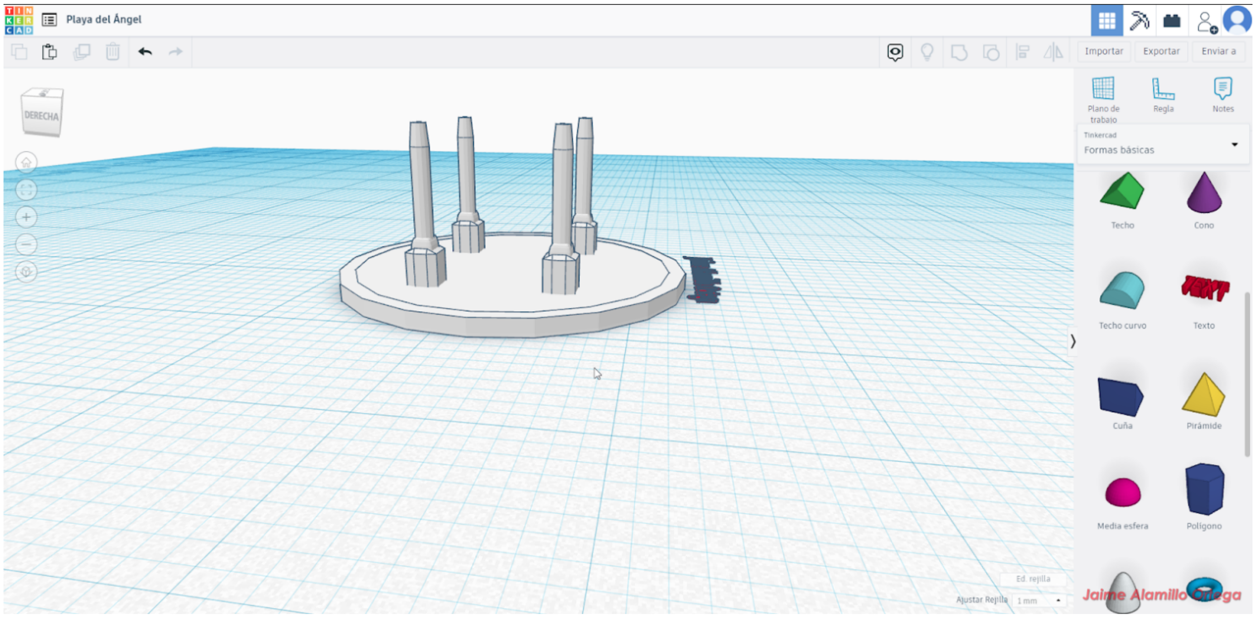

Figure. 3D Model result.

The student is not asked to make a perfect replica, but his design is required to keep the proportions and shapes of the original design.

How can the student's work be evaluated?

By taking a test or orally through questions and by evaluating the work done

Lesson 3. The STL resources. Object repositories (TO KNOW)

Lesson objective: Know where to download and use pre-existing designs. Know how to evaluate the different designs.

In this lesson, the most used object banks will be explained to the students. By far the most widely used STL object bank is Thingiverse. In it you can find objects classified by categories:

- Art

- fashion
- Gadgets
- Hobby
- Hosehold
- Learning
- Models
- Tools
- Toys & Games
- Other

This is a repository where you can find a large number of objects. In addition, you can find objects that fulfill part of the desired functionality and we can modify them to meet our needs. In other years we have created in this subject a working group to print and create objects related to biology and we have printed mitochondrias, chloroplasts, etc.

Licenses of use. Before carrying out the search, an enumeration of the licenses of the different STL objects that can be found in thingiverse will be made. For this, we will base ourselves on the information on the following website: https://creativecommons.org/licenses/. The search will be done taking into account the license and reminding the students that it is important to attribute the work to the creator.

The activity that is proposed with the students is to do an exhaustive search of the different objects related to robotic arms and after a complete search it has been decided among all the class members to use the robotic arm of Florin Tobler. The user license has been analyzed to avoid problems and it has a **CC_NC** license

CC BY-NC. This license lets others remix, adapt, and build upon your work non-commercially, and although their new works must also acknowledge you and be non-commercial, they don't have to license their derivative works on the same terms.

More information about this license can be found at: https://creativecommons.org/licenses/

The object in question is shown in the following figure:

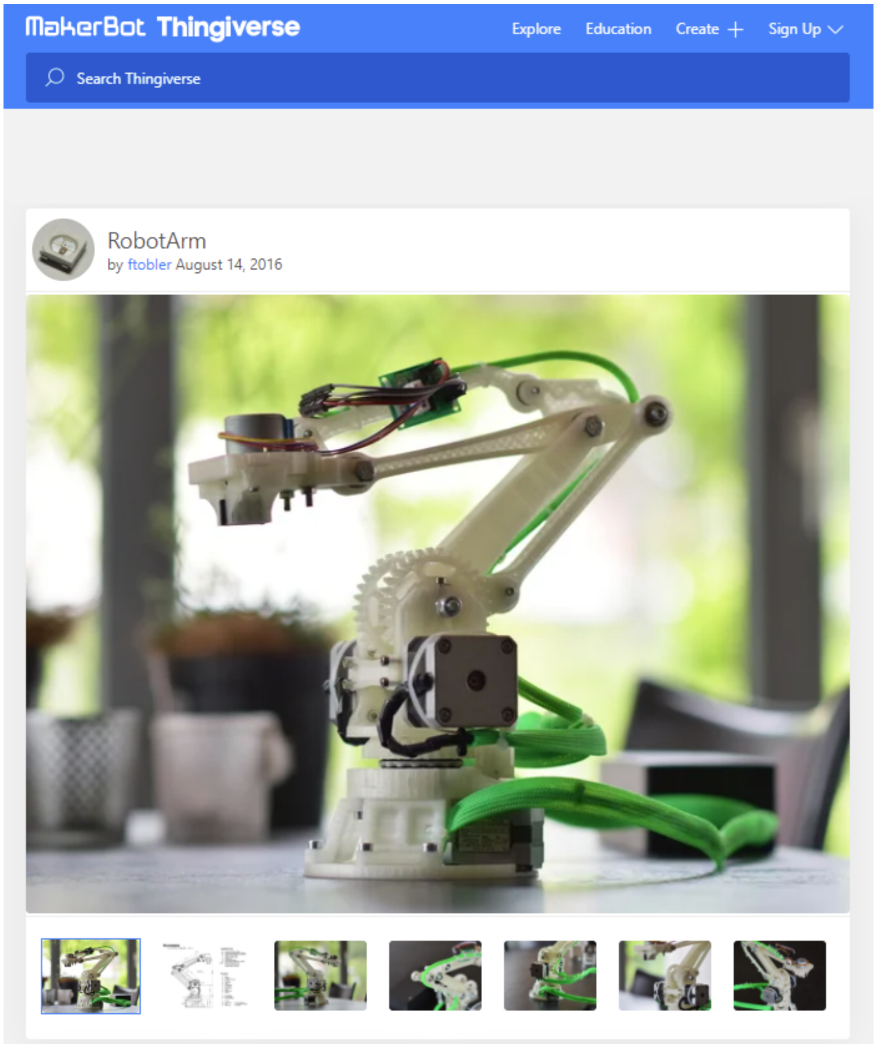

Figure. Robotic arm by Florin Tobler.

In addition to the STL of the different parts, you can find STP files which will show the assembly process to follow.

List of STL repositories

Thingiverse (http://www.thingiverse.com/). In this site you can find many objects. No registration is required to download. The site also allow uploading your 3D objects and then creation of apps with their API in order to interact with the site.

GrabCAD (http://grabcad.com/library/category/3d-printing/software/stl?per_page=20). GrabCAD is a CAD user community. This community has over one million registered users and over 300,000 CAD files available including STL files. Registering is compulsory to download models.

TurboSquid (http://www.turbosquid.com/Search/3D-Models/free/stl). TurboSquid was founded in 2000 and aims to produce 3D models for industry. There are more than 300,000 models to download. Just search by keyword or browse the categories to find the model wanted to download. As in GrabCAD, you must be registered to download models.

SketchUp (https://3dwarehouse.sketchup.com/). SketchUp is designing 3D objects software. With this software you can design and share 3D models with the SketchUP community. There are two kinds of software, the SketchUp and the SketchUp Pro. The SketchUp Pro has extra functionality and is recommended for advanced use.

3D Via (http://www.3dvia.com/search/?search[format]=stl&search[file_types]=1). is a web where more than 350,000 users share STL files collaboratively. There are many models grouped by categories.

The 3D Studio is one of the largest 3D model database. The only problem is that you have to pay for the models that you want to download.

Yeggi (http://www.yeggi.com/) is a search engine for STL models. There are thousands of downloadable models and the database is updated daily with more and more models.

STL File (http://stlfile.com/) is a site where you can find objects to download. We recommend to go to STLdownload.com. STLdownload.com is a search engine of STL files.

STL Download (http://stldownload.com/). STL is a web where you can download and upload STL models. Currently there are lots but the database is growing.

Lesson 4. How to calibrate a 3D printer. (KNOW HOW)

Lesson objective: To know how to calibrate a printer.

To calibrate a printer you have to check that when you start printing, only one sheet of paper can fit between the bed and the nozzle.

If the nozzle is too close to the bed the filament will not come out and if it is too far from the bed the filament will not be deposited properly.

In the printer settings you can adjust the z axis to move the nozzle closer to or away from the print bed.

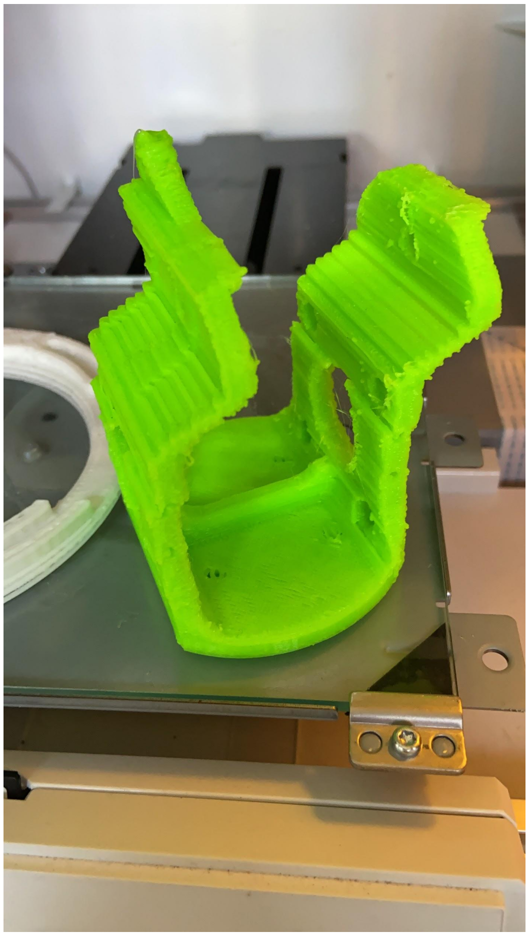

Figure. Example of a printer with a loose belt.

Keep in mind that calibrating a printer is not only adjusting the nozzle to the bed. You also have to verify that the belts have the recommended tension. A loose belt causes the parts to print like the image above, while if the belt is too tight it is in danger of breaking.

The image above shows a part that has been printed with a printer with a loose belt.

How can the student's work be evaluated?

By evaluating the work done

Lesson 5. 3D printing. (TO KNOW, KNOW TO DO and KNOW TO BE)

Lesson objective: Know how to generate a 3D part from an STL file. Monitor the process and verify its achievement.

The first thing you have to do is download all the STL files that you are going to print and evaluate the files that are going to be printed before or after. You have to keep in mind that a large file like the base will take a long time to print so it is a good idea to leave it printing for after school or at night. Smaller pieces can be printed during classes.

Advice: it is recommended to verify the first minutes of printing as they are the most critical since the piece may lose adherence or generate warping.

As many pieces have to be printed, a round-robin system has been chosen among the students so that each and every one will print a piece of the arm.

Figure 14. Student with the part that he has printed verifying the result with the 3D model.

The goal is for each student to print a piece from scratch choosing the STL file, the printing parameters, etc. And be in charge from the time the piece is sent to printer until the print job is finished.

The student must take care that the base has a sufficient proportion of fixing lacquer and, above all, will be aware of the most critical phases of printing, such as the start and end of it when the height is considerable.

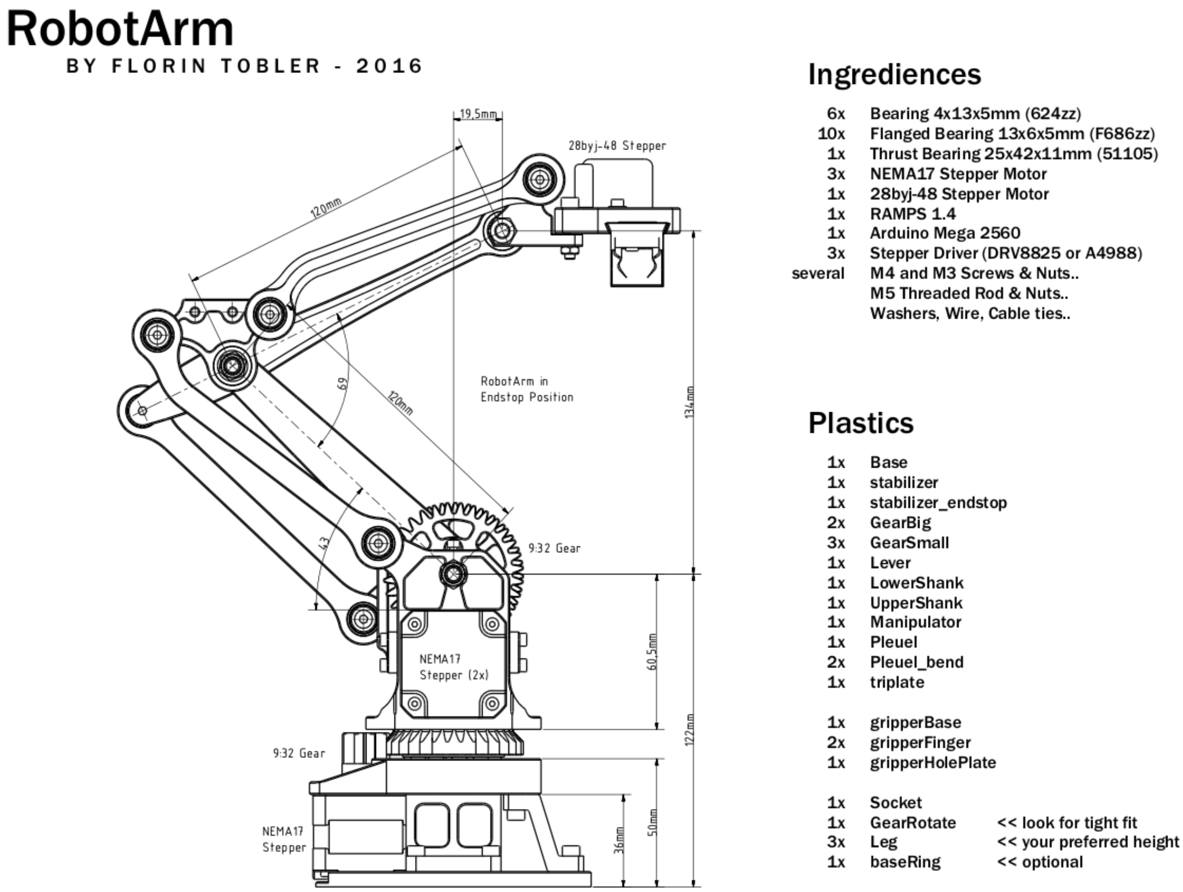

Figure. Model design

As you can see in the figure above, the arm will have many plastic parts in addition to other electronic and mechanical parts like motors, nuts, arduino board, RAMPS, etc.

The pieces to be printed are listed in the following figure:

Plastics

1x	Base	
1x	stabilizer	
1x	stabilizer_endstop	
2x	GearBig	
3x	GearSmall	
1x	Lever	
1x	LowerShank	
1x	UpperShank	
1x	Manipulator	
1x	Pleuel	
2x	Pleuel_bend	
1x	triplate	
1x	gripperBase	
2x	gripperFinger	
1x	gripperHolePlate	
1x	Socket	
1x	GearRotate	<< look for tight fit
3x	Leg	<< your preferred height
1x	baseRing	<< optional

Before printing, the part has to be visualized in a specific 3D visualization program to determine the best position of the part to be printed.

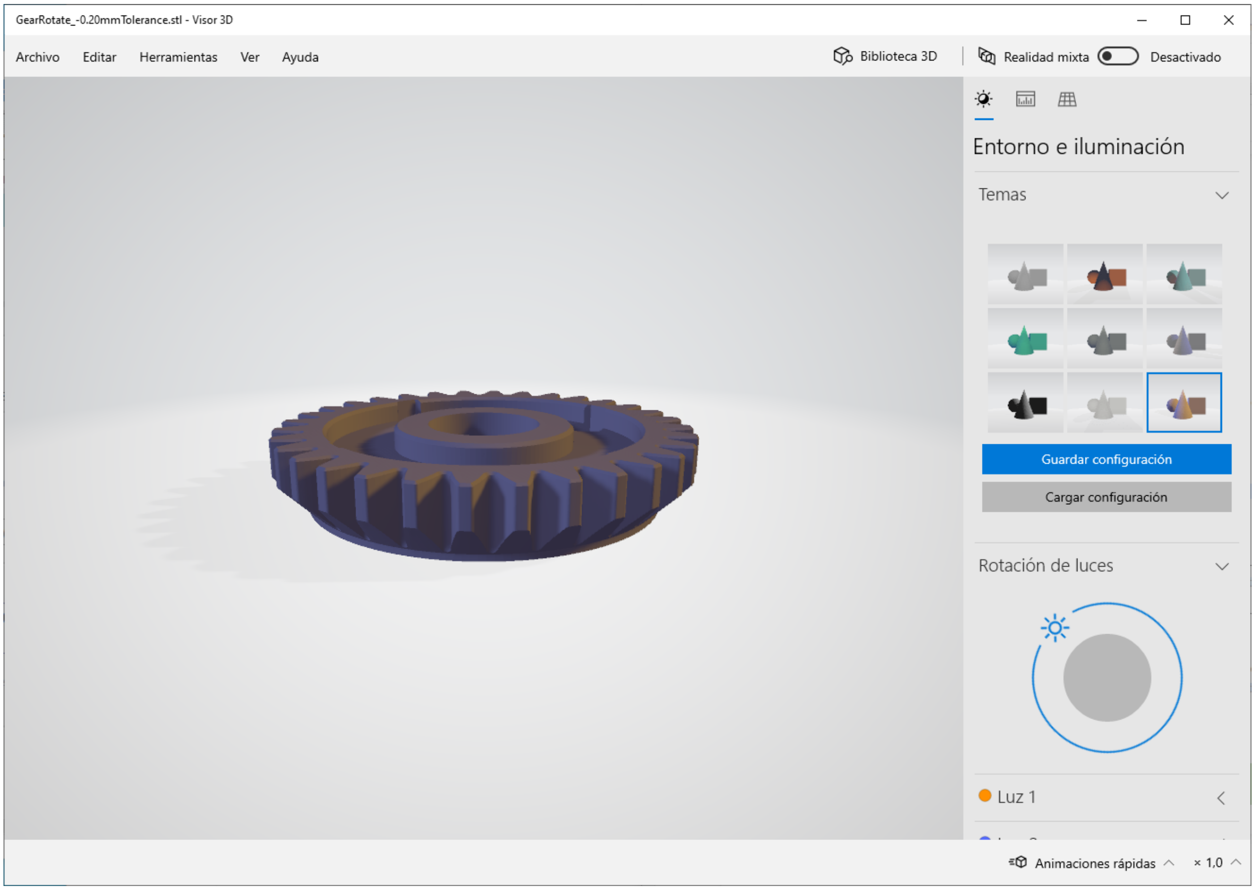

Figure. 3D object viewer.

With a 3D viewing program, the piece can be viewed from different positions and focused with lights from different angles. In this way the geometry can be visualized in a complete way.

Generally, the position of the pieces is usually adequate for printing, but nevertheless the student has to assess this fact since when creating or designing a piece, the position it takes on the printing bed is important.

Activity for learning about design 3D parts

IMPORTANT

Activity created for students who have no knowledge of graphic design.

In order for the student to know the whole process from the moment a design is created until it is finally printed and is involved in the retouching process, the following activity has been created:

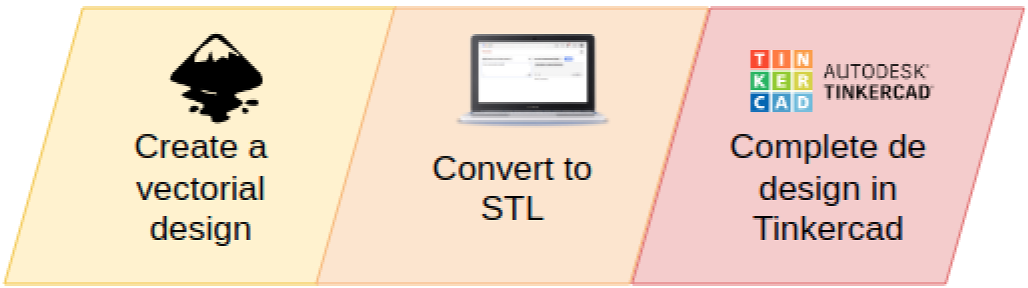

Figure. From a vectorial image to a 3D model.

In the first step of the activity, a vector image will be created using a tool such as Inkscape. Inkscape is a free and open source vector graphics editor. Inkscape (https://inkscape.org/) can create and edit complex diagrams, lines, graphics, logos, and illustrations. The main format used by the program is Scalable Vector Graphics (SVG).

We have created the logo for our project with this tool:

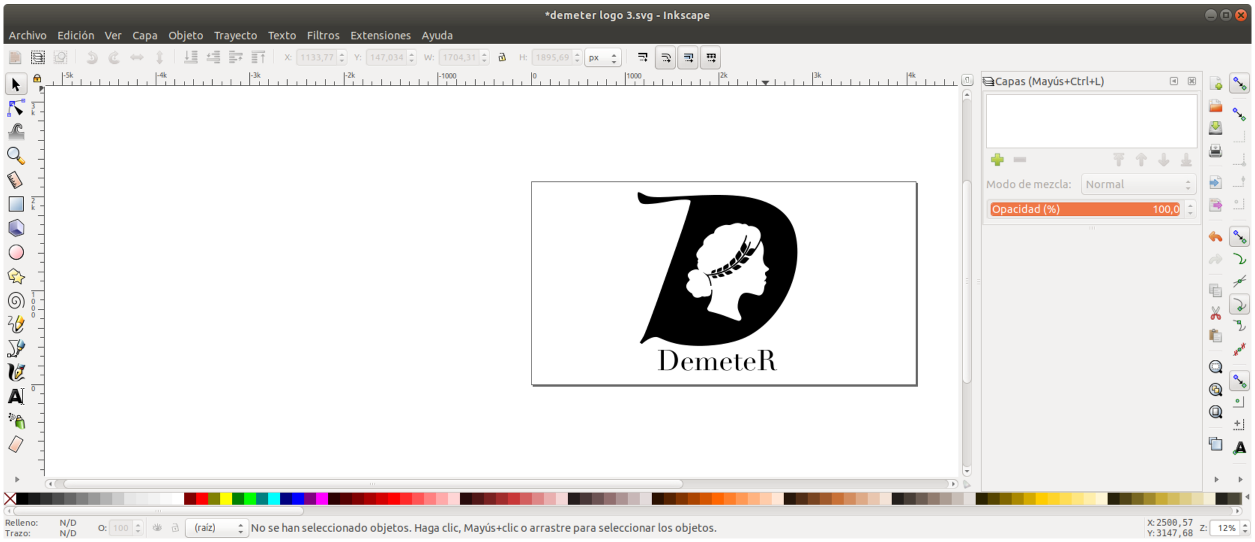

Figure. Inkscape.

The SVG format, which is a vector format, will allow us to transform it into an STL format. For this you can use online converters such as svg2stl.com, anyconv.com, etc. It is not necessary to install any extra software on the computer to carry out the transformation.

With the tinkercad application, many transformations can be done in a simple way and without having knowledge of designing.

For example, we have decided to add a base to the design so that when printing there are no warping problems.

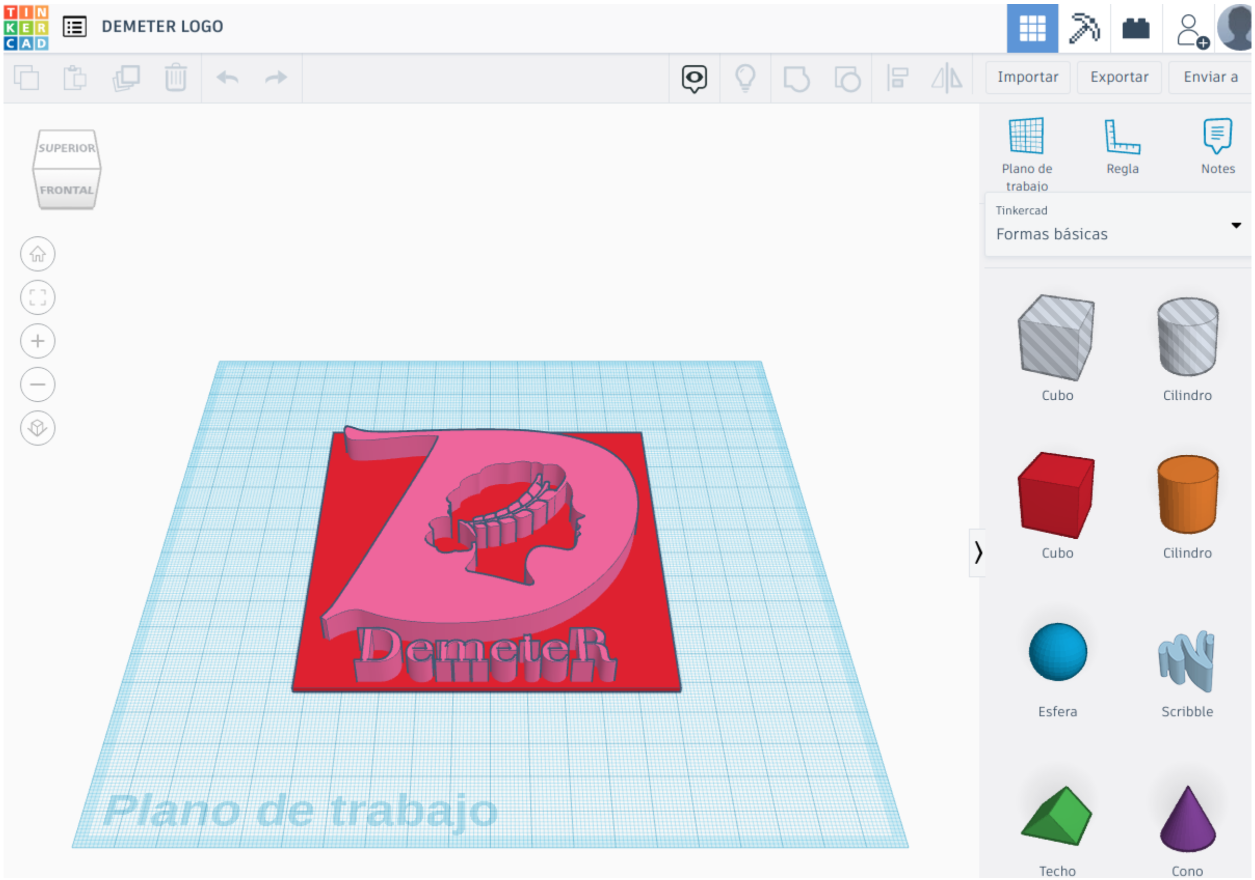

Figure. Tinkercad.

Once the STL design has been created, the generated file will be transformed into gcode or the program that uses the 3D printer. This last step will depend on the printer being used.

How can the student's work be evaluated?

By evaluating the work done

Lesson 6. Problem resolution. Printer Settings (TO KNOW and KNOW HOW)

Lesson objective: Know how the printer is adjusted, what parameters must be selected in the setup, know how the filament is changed, monitor its key elements such as belts, motors, etc.

Adjusting the printer mechanics and adjusting the printer settings

One of the skills students have to learn is problem solving by adjusting the printer. In REP-RAP type printers (prusa and derivatives) and low cost printers, the adjustment has to be continuous. In other more sophisticated printers, they themselves do an auto-adjustment which allows the user greater productivity.

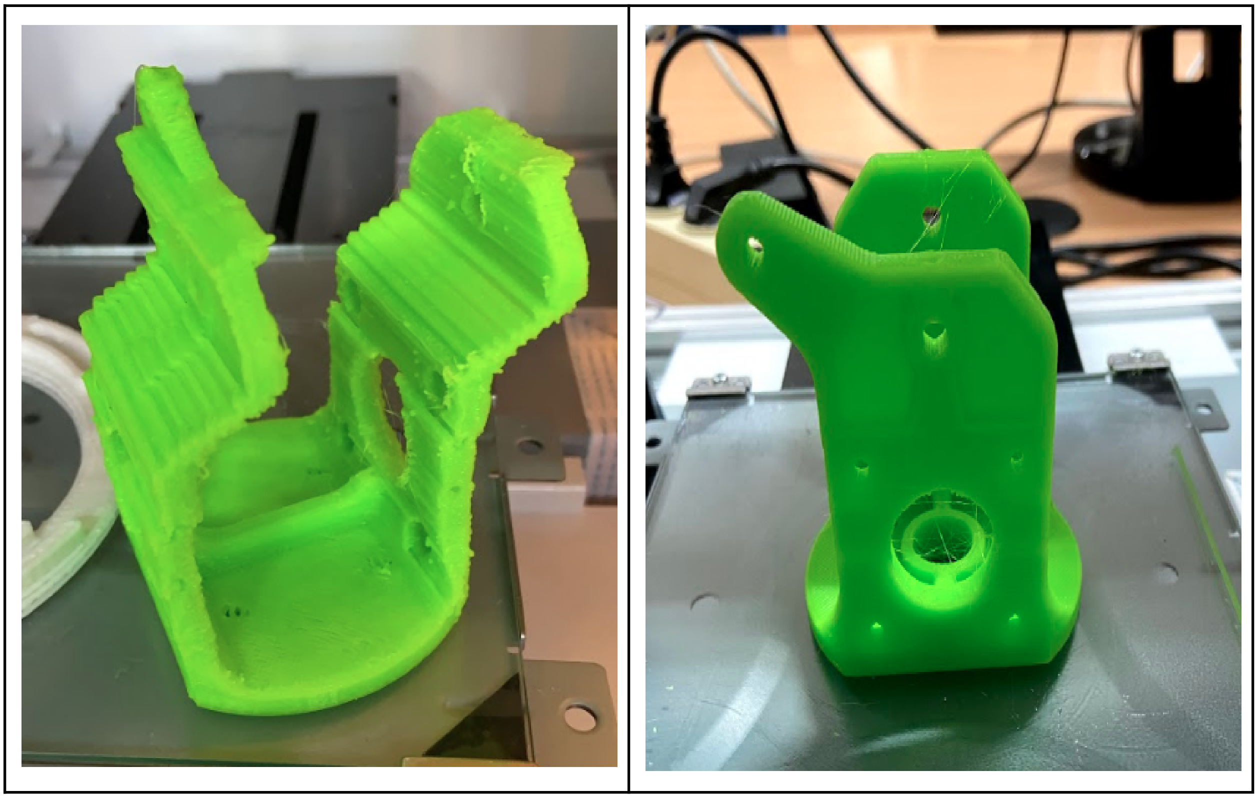

Figure 20. 3D parts.

In our case we can see the two images that we obtained, the first one is the impression of a piece with a mismatch in the Y axis. And the second, the same piece once the belt of the Y axis has been adjusted.

Troubleshooting helps the technician better understand how to calibrate the printer and how it works. There are times when you learn more from failures than when everything works properly.

It is also important to know how to do the filament change, how to safely remove the filament and insert it for optimal printing.

What to do when the object is bigger than the bed

When the object is very large there are several options. The first and easiest is to decrease its size. In this simple way the object can be adjusted to the bed.

Another option is rotation in the Z axis. Rotating the object in the Z axis is usually the solution for some objects that do not fit correctly on the bed in the home position but rotating it 45 degrees (more or less) can fit in the printing space.

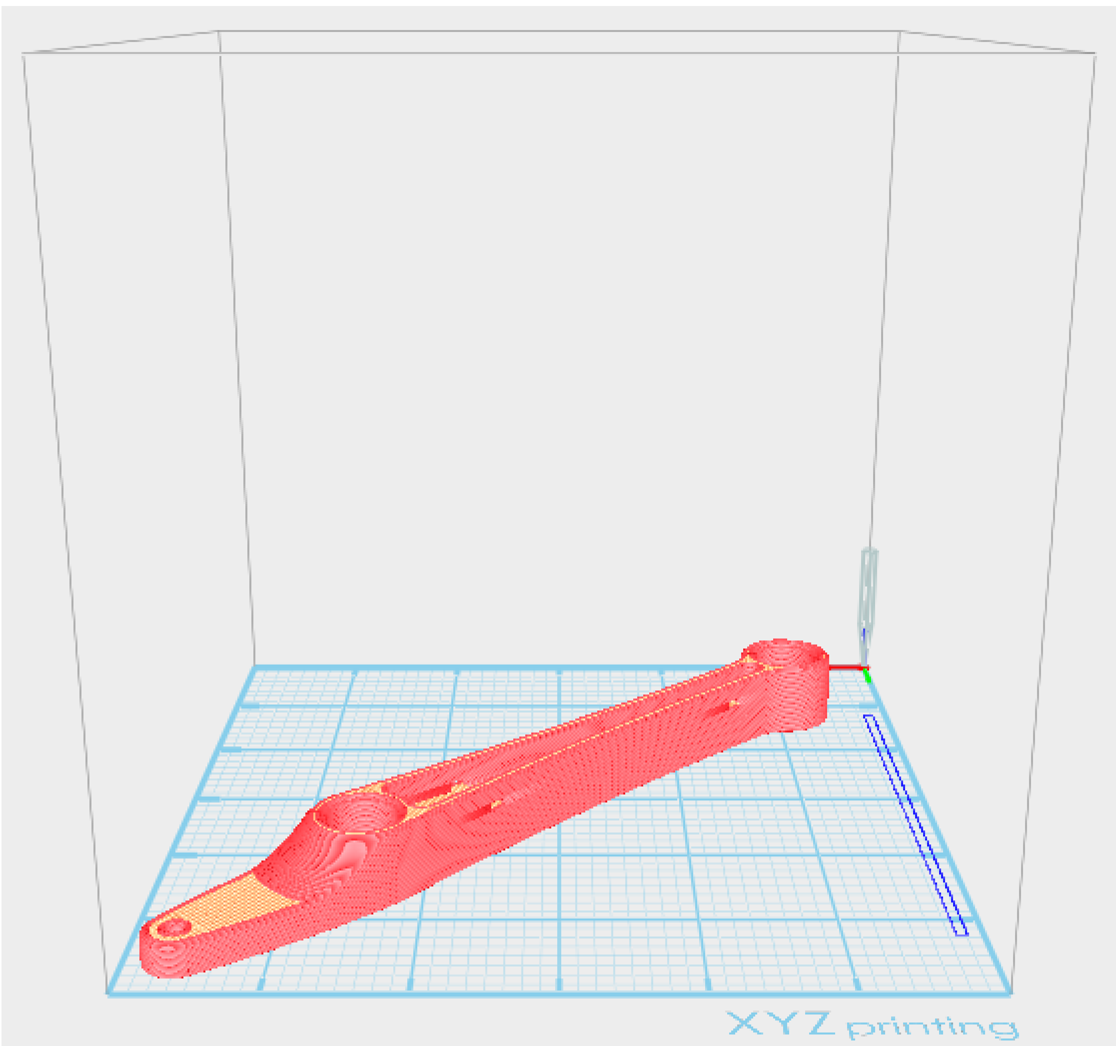

Figure. Rotating a piece to fit in the printing bed.

How can the student's work be evaluated?

By evaluating the work done

Lesson 7. Arm assembly (TO KNOW, KNOW TO DO and KNOW TO BE)

Lesson objective: Assembling the arm. Use CAD programs to visualize complex 3D objects. Know how to interpret 3D designs and assemble the arm based on them.

One of the reasons why we have chosen this arm is because the author has taken great care of the material that he distributes including several STP files which can be opened by a CAD program and you can clearly see how the arm has to be assembled.

For this, an investigation has been made of which will be the simplest and most suitable program for the group. Finally we have chosen Freecad and gcad3d for Ubuntu. Two free open-source tools that will meet the needs we have.

Freecad and gcad3d

For the assembly of the arm, the project has an STP file which allows it to be opened with a 3D modeling program. Given that in class we use the Ubuntu Linux system, we have chosen Freecad and gcad3d which are a more than adequate tools to visualize 3D projects through different perspectives and to be able to get an idea of the assembly sequence and the relative position of each of the pieces.

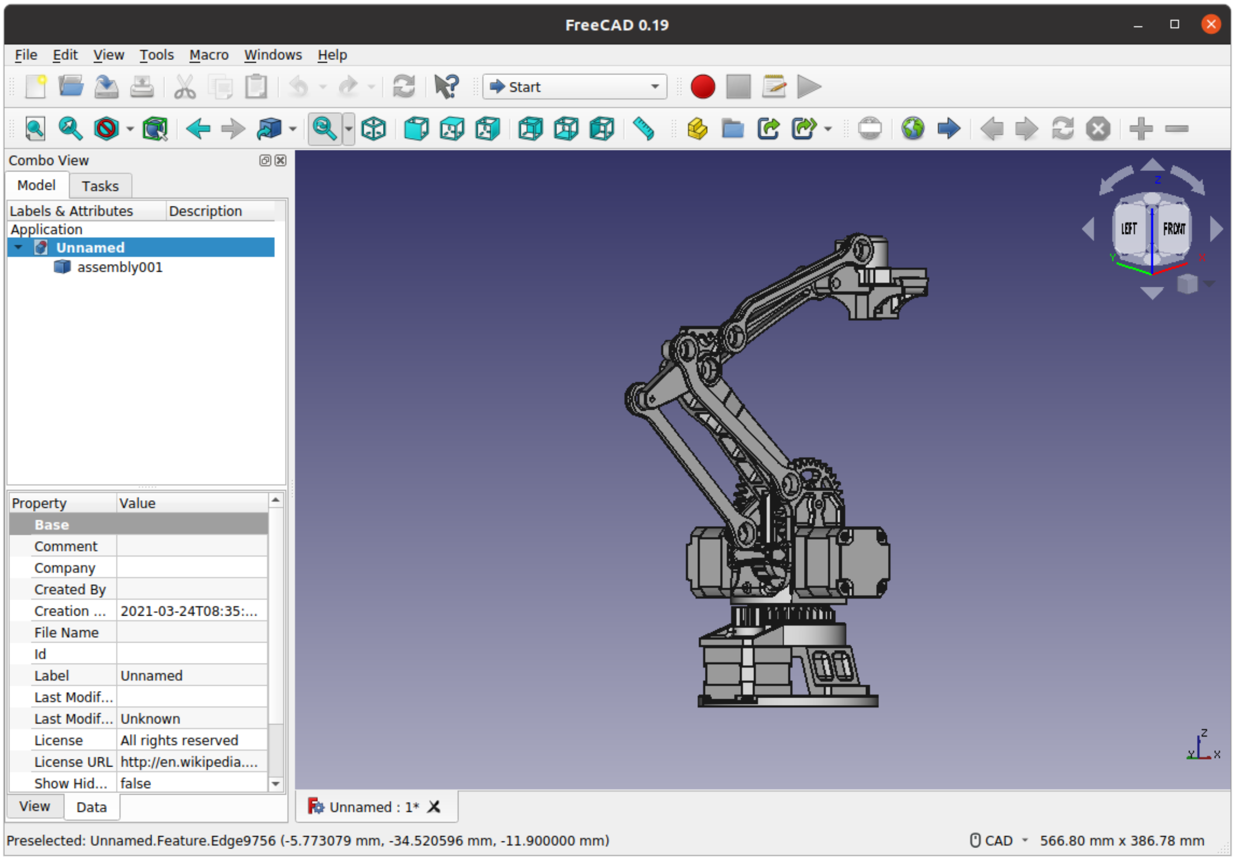

Figure 22. Viewing the robotic arm design in FreeCAD

View of the same STP file with gcad3d

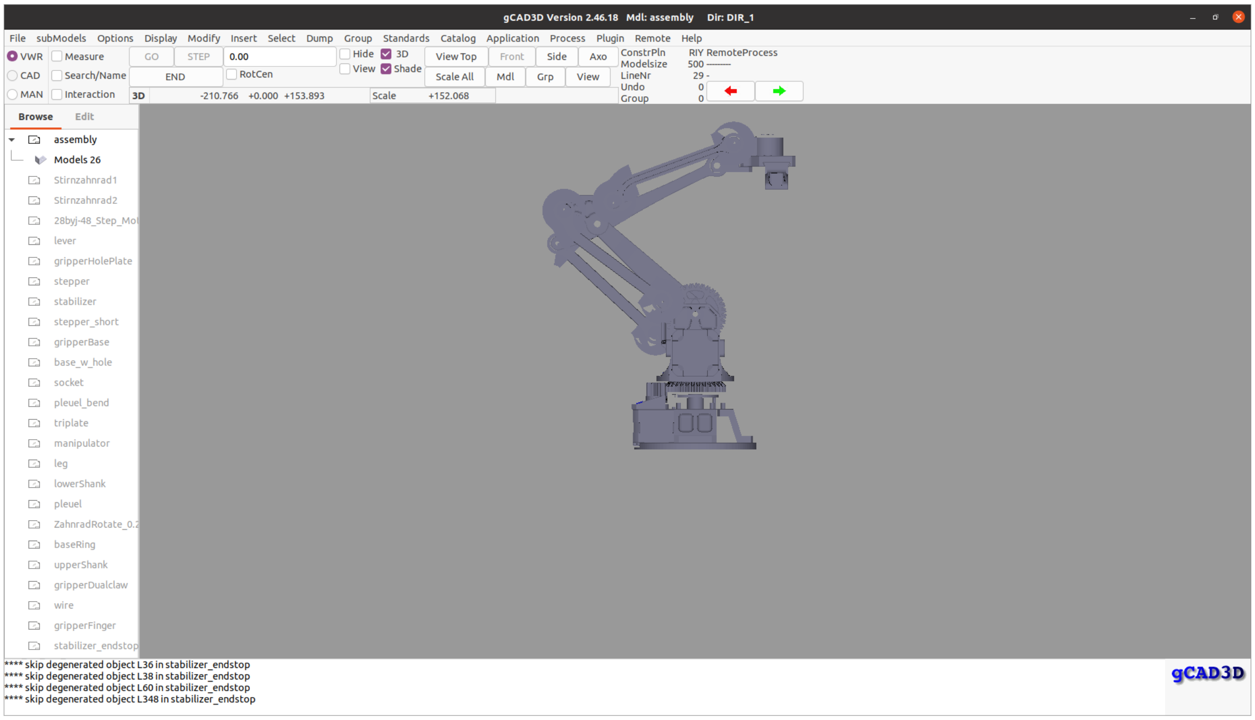

Figure. Viewing the robotic arm design in gcad3d

Figure 24. Assembling the arm

The advantage of gcad3d is the ability to display all the pieces together and highlight an individual piece as well as its name.

After analyzing how to assemble the robotic arm with the different programs, it is time to proceed to assemble it.

It is important to have the right tools, enough light, the right screws, nuts and washers, etc.

Assembly must be done safely and with enough time to finish properly.

Figure. Assembling the arm

One of the most delicate parts of the assembly is the base. It is important that the base is correctly assembled because it will receive most of the weight and because they will be the parts that will have the most friction and tension.

Below is an image from the creator of the Florin Tobler arm showing the correct way to assemble the base.

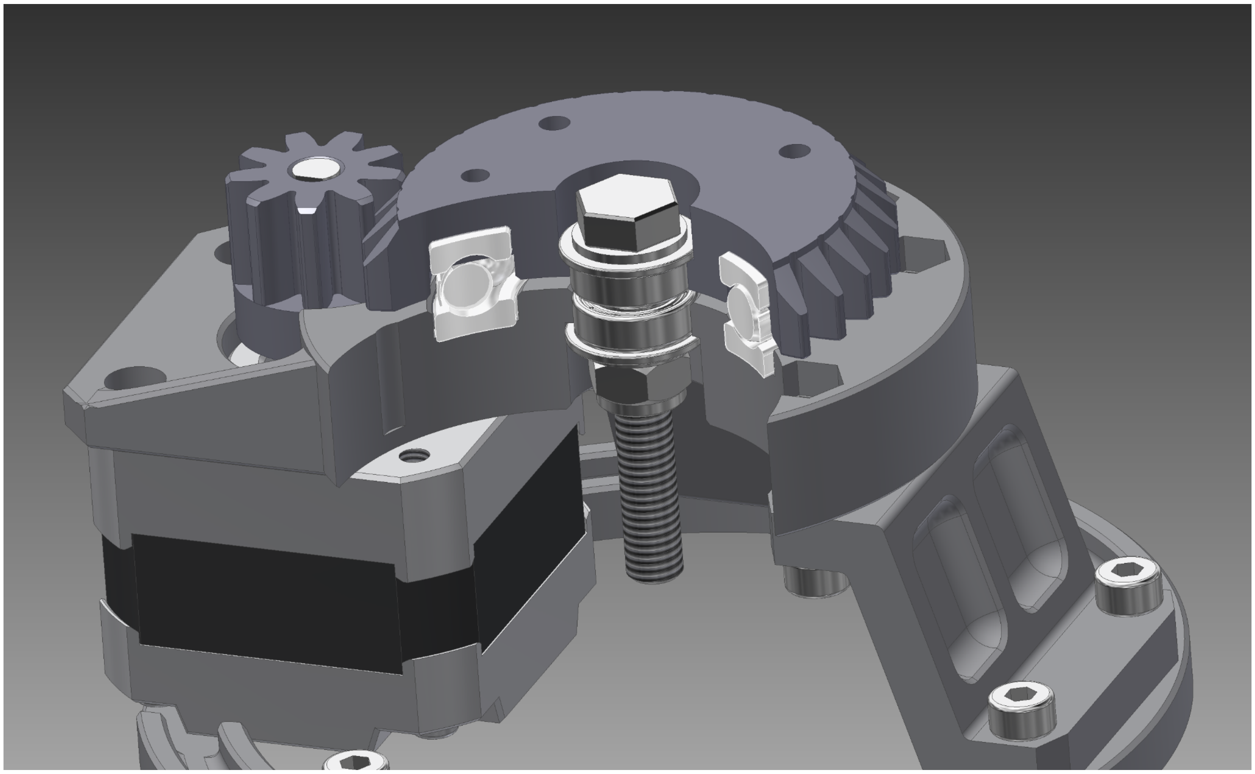

Figure. 3D model of the arm

Below is a list of nuts and bolts recommended for mounting. This listing has been provided by the creator of the robotic arm Florin Tobler:

- 16x M3x6
- 21x M3x8 (3x used as set screws)
- 4x M3x10
- 2x M3 Washer (large for gripper)
- 4x M3 Nut
- 6x M4x10
- 11x M4x16
- 2x M4x25
- 14x M4 Washer (small enough for Bearing)
- 8x M4 Nut
- 12x M6 Washer (small enough for Bearing)
- 3x M6 Nut
- 3x M6 Self Locking Nut
- 3x M6x45
- 1x M6x80mm Threaded Rod

How can the student's work be evaluated?

By evaluating the work done

Lesson 8. The electronic

Things that you are going to need for this lesson:

1 Arduino Mega

- 1 RAMPS 1.4
- 1 Power supply. You can use one from an old computer. If you do that, remember, yellow cable is 12 volts, red cable is 5 volts and black cable is GND. To make the power supply running you will have to connect the green cable from the ATX connector (the big one) with a black cable (GND) before plugging the power supply to the socket.
- 3 POLOLU drivers A4988
- 3 Step motors
- Cables to connect the power supply to the RAMPS.
- Tools

To connect all the electronics you will have to follow these steps:

1. Connect the A4988 drivers to the RAMPS in the proper position. See the image and the scheme below.
2. Connect the RAMPS to the Arduino Mega.
3. Connect the step motors to the RAMPS (see the scheme below).
4. Connect the power supply to the RAMPS.

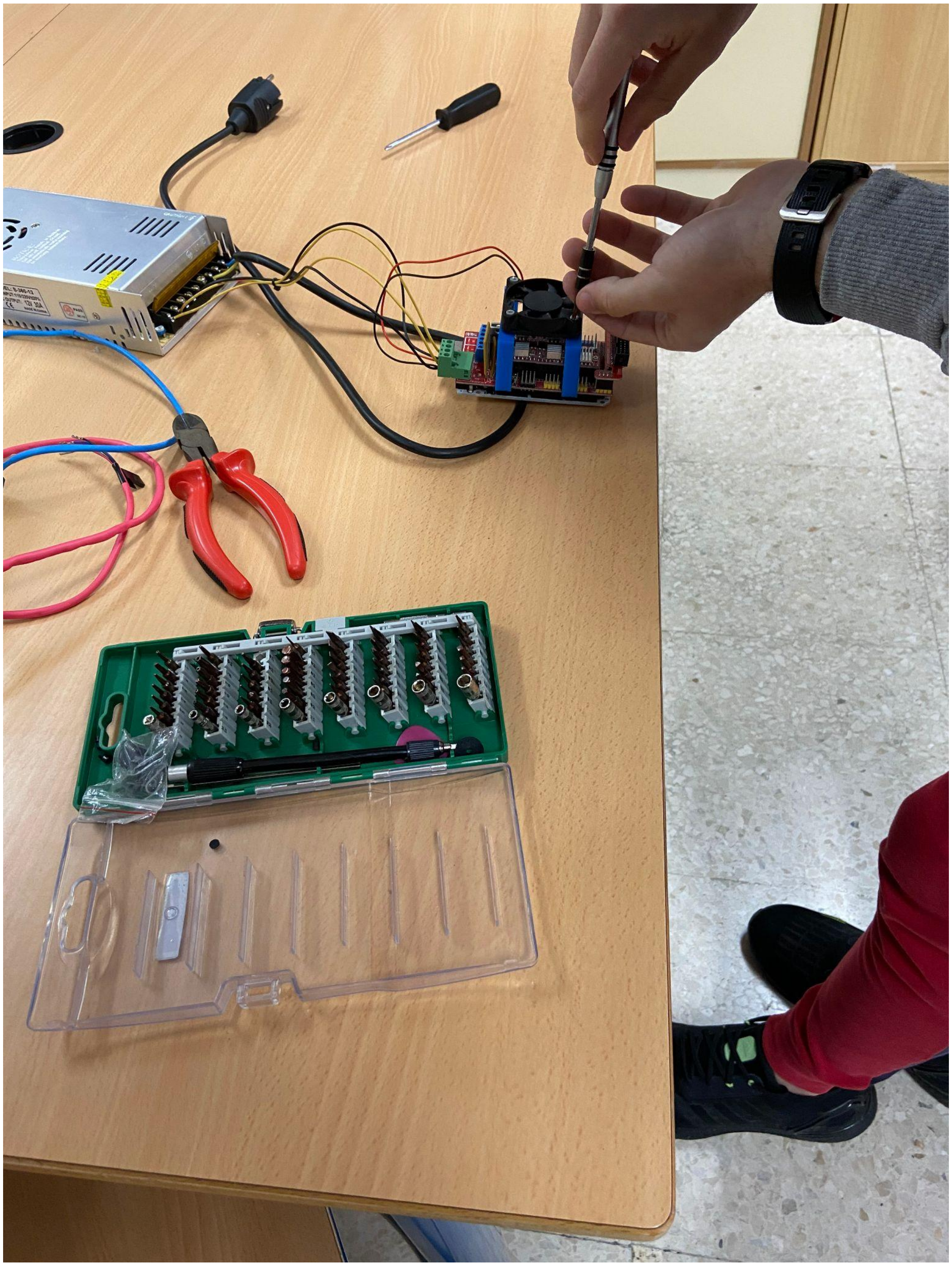

Figure. Assembling the electronics

Component connection is important in this section. A unpropper configuration can make the robot not work or even ruin the components that we are using.

It is also important to install the firmware and modify those parts of it to adapt it to our needs.

8.1 Connecting the components

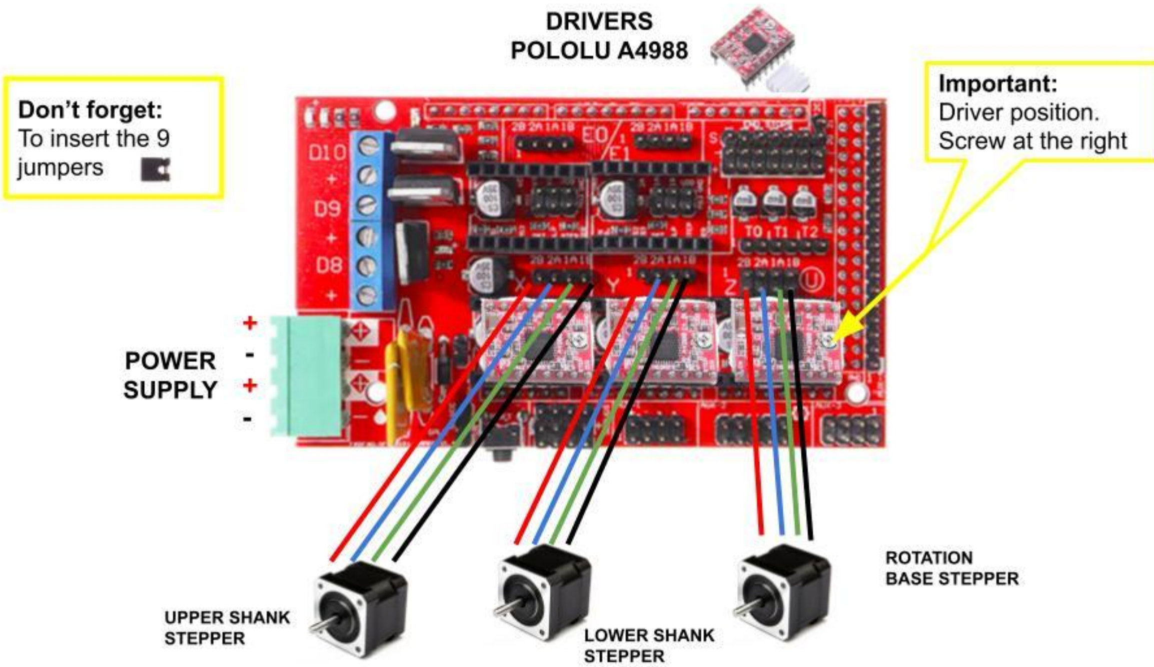

Figure 28. Electronic schema

It is important to connect a power supply or battery that provides a voltage of 12 volts and a current of 4 amps.

The fan can be connected directly to the power cables and that way it will always be on. It is important to have the drivers well ventilated as they get very hot and can be damaged if they get too hot.

To control the stepper motors we will need to use A4988 POLOLU drivers which have to be correctly positioned in the RAMPS because otherwise the motor will overheat and surely stop for this reason. Generally the drivers have a small screw to modify the power and that screw should be on the right taking into account the previous diagram.

The RAMPS package comes with 9 jumpers that must be put in the right position if you want to work with microsteps instead of steps. Depending on the configuration used, the microsteps can be a half step, a quarter step, an eighth step or a sixteenth step. If you use a 16th step you can have up to 3200 steps per revolution.

Each step in our motors is 1.8 degrees, so each turn (360 degrees) we will have 200 steps. Using a sixteenth of a step we will have 200 * 16 = 3200 microsteps. That way we will increase the precision of our robotic arm.

8.2 The firmware.

Figure. Arduino IDE

To install the firmware to the electronic components that we are using we will need an IDE (Integrated Development Environment). Arduino has its own. You can download it from https://www.arduino.cc/en/software. Download the latest version for your operating system from this web page and install it on your operating system.

Once downloaded and installed, run the program (Arduino IDE) and an empty script like the following one will appear:

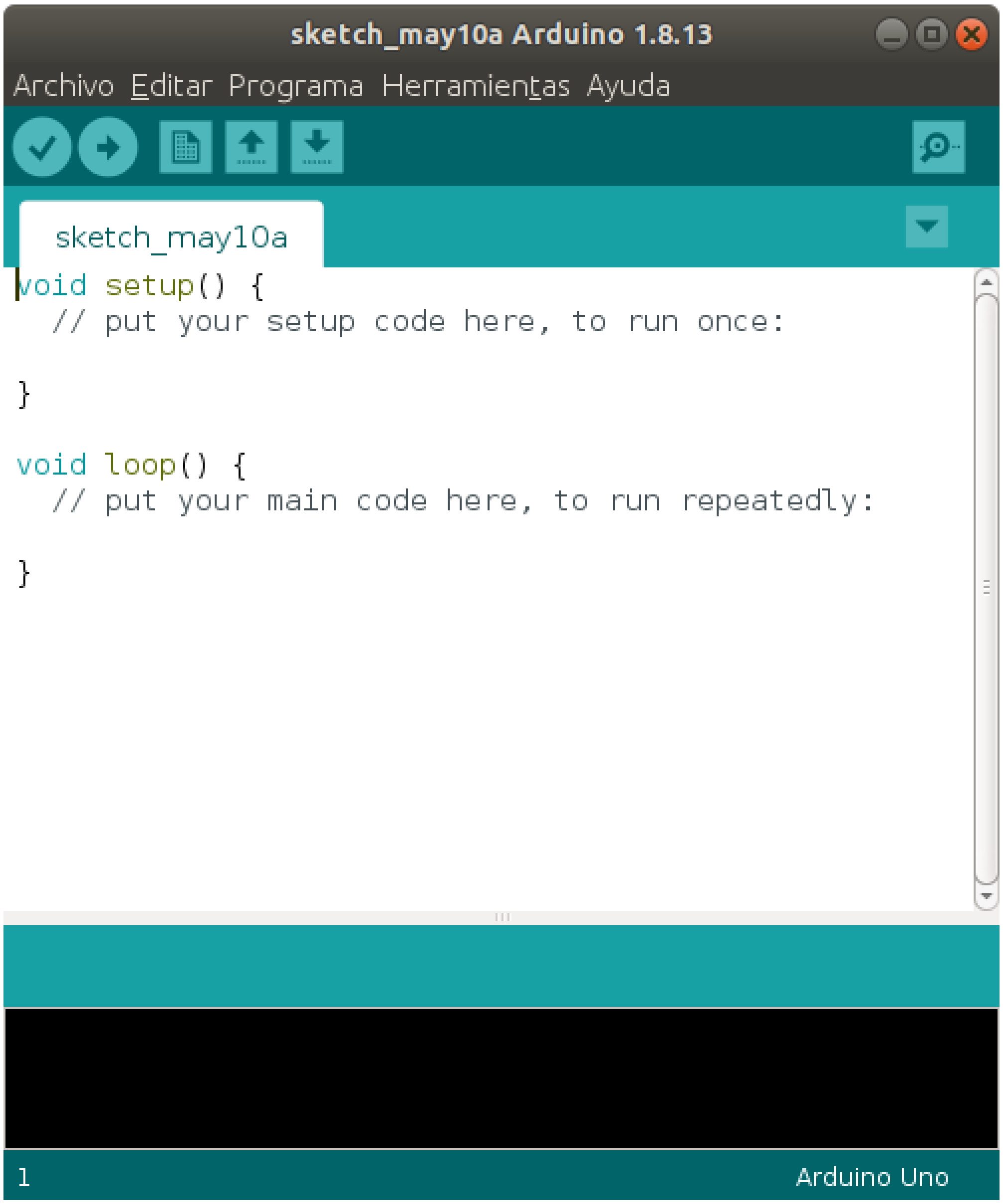

Figure 30. Our first program.

The next step is to download the firmware for the robotic arm that you can find on this web page (Github): https://github.com/ftobler/robotArm/

Remember that the robot software and the STL files of the robotic arm have a Creative Commons CC BY-NC license, so you must always indicate the author if you are going to disseminate your work and,of course, the license does not allow you to obtain economic benefit from the objects or software.

In the Arduino/robotArm folder of the software you will find the main project file called **RobotArm.ino**. Run it and you can load the project into the Arduino IDE.

In principle, if you are not going to use the robot for anything other than what it is designed for, you should not make any modifications to the firmware, but if you need it to be part of a larger project, it will be necessary to modify the firmware. For this we are going to make a brief description of all the files and classes used in C ++.

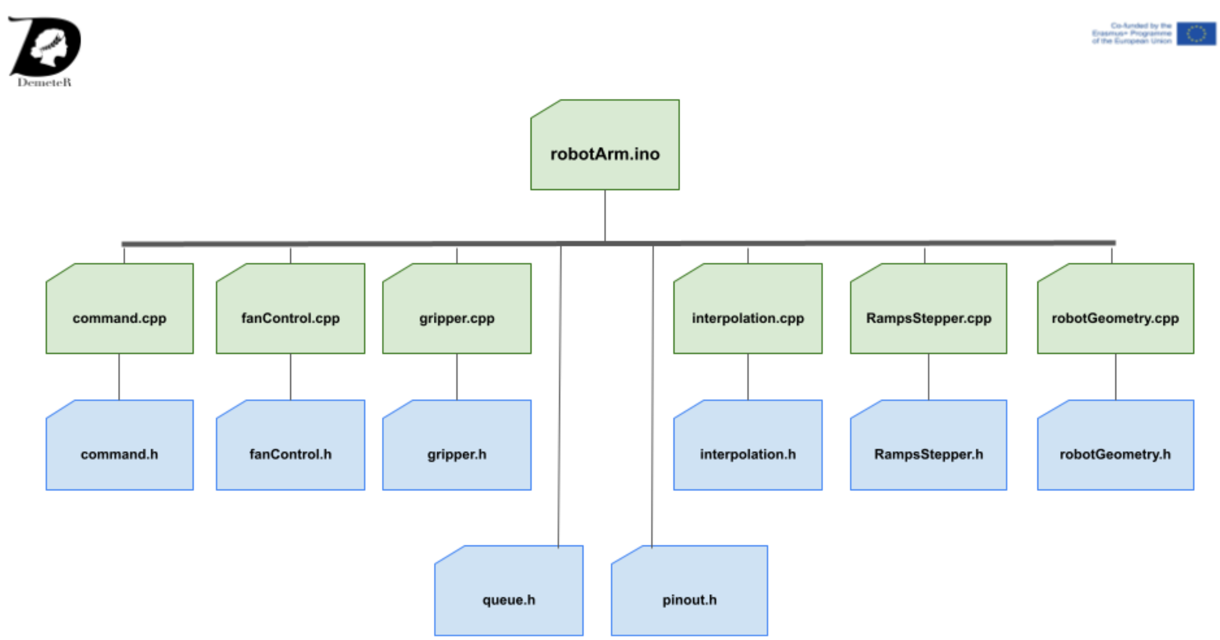

Figure. Robot arm class hierarchy

In the previous figure you can see a class scheme of the software with which we are going to work. The system is simple. The robotic arm will receive commands via serial and to process the commands it receives, it uses the classes located in the command.cpp file. All commands are being pushed into a typical C ++ queue that supports **push()** and **pop()** commands. If you have notions of programming and have studied queues, the code for this is not something difficult to understand. All the information and classes of the queue are in queue.h.

fanControl.cpp and **.h** will be used to control the fan as the name suggests. It allows you to indicate the pin where the fan is controlled and enable and disable it. We have placed it directly into the stream in our design, so we will make little use of this code.

gripper.h and **cpp** are used to control the arm gripper (open and close). In our project they will not be used but if you need to grab objects you will use this code to give instructions to the gripper servo.

interpolation.cpp and **.h** are necessary for the robot to interpolate and place the arm at the correct coordinates.

robotGeometry.cpp and **.h** are required to calculate the geometry of the robot. The most complex function is **calculateGrad ()** which uses functions such as square roots, sine and cosine functions to calculate the degrees and direction of the arm in rotation. It is certainly the most difficult code to understand.

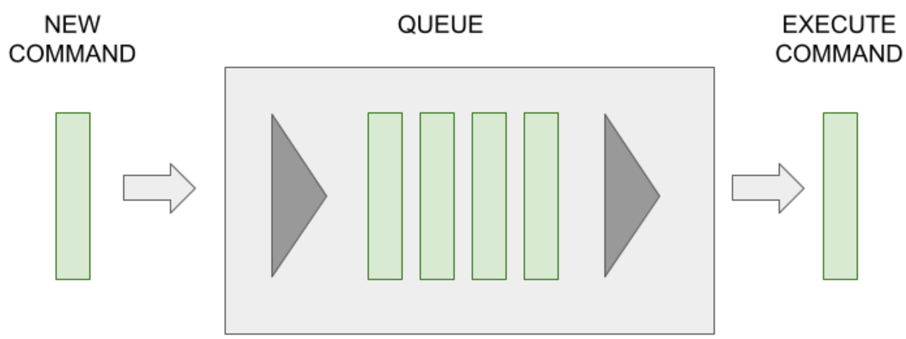

Figure. Arm. Queue system.

As previously explained, the robot will receive G-Code commands via serial and will add them to a queue. When the robot needs to execute an instruction it will check if there are instructions in the queue and if it is positive it will take out and execute the oldest one.

Tip

If you want to add new commands to the robot, the right place for it is the function `void executeCommand(Cmd)`

Lesson 9. Programming the movements

The robot uses gcode commands. G-Code is a programming language for CNC machines. These commands will tell our robot to move the X, Y or Z coordinates, which are the motors that we have installed in our arm and connected to our RAMPS.

Although the G-Code is used in CNC machines and there are different implementations such as DIN66025, ISO 6983, Siemens, Haas or FANUC. Many manufacturers expand the instruction set to work with the machines they have designed. We will also expand the instructions to be able to control not only the arm but also the engines of a rover that is where we are going to place our robot.

Here are tips on the GCODE that Florin Tobler has programmed for the robot:

A valid gcode command looks like G1 X0 Y120 Z120
There have to be spaces in between.

- G0=G1: Move XYZ in mm (cartesian), F in mm/s. Always uses Absolute coordinates. Unlike 3D Printers, there is no E-Axis. On every move there is an acceleration and deacceleration.
- G4: Dwell / Sleep T in milliseconds
- M3: Close Gripper / Aux Motor T in Steps. Cannot move simultaneously to G
- M5: Open Gripper / Aux Motor T in Steps. Cannot move simultaneously to G
- M17: Enable Stepper Motors, also Enables Fan
- M18: Disables Stepper Motors, also Disables Fan with delay
- M106: Enables Fan
- M107: Disables Fan

We are not going to use the M106 or M107 commands since we want to always have the fan running and not depend on the programming. That way we make sure that bad programming does not overheat the drivers and damage them.

In conclusion:

- G commands. They are used to move the robot.
- M commands. They are used to control the gripper, motors and fans.

Lesson 10. Set up the arm

To put the robot into operation you must open the Arduino serial port and configure two things:

- Baud Rate to 115200. It is important that the firmware on the arm and the serial monitor operate at the same speed. Otherwise, the arm will not receive the commands correctly.
- The line ending to NL & CR. If this parameter is not set to NL & CR, the arm will not detect that a command has been entered and it will not work correctly.

In the following image you can see how the two parameters are configured correctly and the command g28 to home all the motors if homing enabled is set:

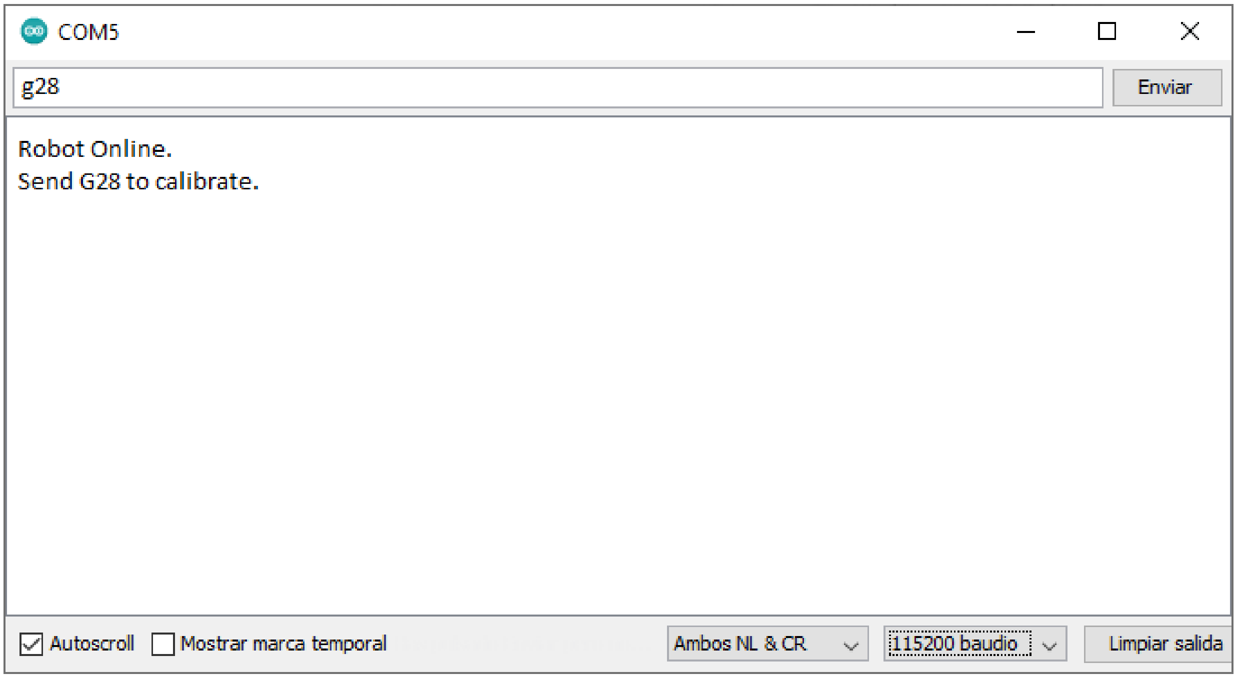

Figure. Arduino IDE. COM window.

Try entering all the GCode commands shown in the previous section and verify that they work correctly.

Bibliography

- Tobler, f (August 14, 2016). RobotArm. Makerbot Thingiverse. Retrieved from: https://www.thingiverse.com/thing:1718984
- RAMPS 1.4. RepRap.org. Retrieved from: https://www.reprap.org/wiki/RAMPS_1.4
- Tobler, F. (31 May 2020). Robot Arm. Github. Retrieved from: https://github.com/ftobler/robotArm/
- Inkscape. Retrieved from: https://inkscape.org/
- Tinkercad. Retrieved from https://www.tinkercad.com/