



## Development and 3D printing of a System with a Displacement of Increments of Less than 3mm Using Arduino UNO Programming

Dina Rochman 

Universidad Autónoma Metropolitana, Cuajimalpa, [drochman@cua.uam.mx](mailto:drochman@cua.uam.mx)

Corresponding author: Dina Rochman, [drochman@cua.uam.mx](mailto:drochman@cua.uam.mx)

**Abstract.** Our research aims to design and 3D print a robot that can precisely move a threaded prism along a threaded rod using Arduino Uno programming and maintain its orientation perpendicular to the axis of the threaded rod. We use M8x1 and M8x1.25 threaded rods and a 28BYJ-48 stepper motor with an L298N driver. We carried out several tests to find the interval increments, with different combinations between the number of steps (10, 4, and 2), different speeds (10, 20, and 30) in the two directions (- and +), and a delay of 2000 microseconds on the stepper motor. Our research shows that when using different programming alternatives in Arduino Uno to control the displacement of a threaded prism, the distance that the threaded prism travels fluctuate between 0.4 and 3 mm, remaining perpendicular to the axis of the threaded rod. The research we present will contribute to the development of 3D printing of customized robots with specific functions and more precise controls programmed with the Arduino Uno, which could have applications in various fields of research, such as biology, anthropology, and medicine.

**Keywords:** Robot Spur gear, Threaded rod, Threaded prism, Arduino UNO.

**DOI:** <https://doi.org/10.14733/cadaps.2024.536-550>

### 1 INTRODUCTION

The field of robotics is booming thanks to additive manufacturing. Among the 3D-printed robots, we find humanoid, four-legged, and small-scale robots. Robotic arms, and automata robots [1].

In the literature, we find some articles related to the 3D printing of robots and the use of stepper motors and servomotors. In these articles, the authors mention that they designed and manufactured the robots for various purposes.

Hunt et al. (2014) [5] present the design and characterization of an aerial 3D printer; a flying robot capable of depositing expanding polyurethane foam in mid-flight. They evaluated various print materials and described the design and integration of a lightweight print module into a quadcopter.

Telegenov et al. (2015) [10] present an open-source-low-cost three-finger robotic gripper platform for research and educational purposes. The 3-D design model of the gripper is presented and manufactured with a minimal number of 3-D-printed components and an off-the-shelf servo actuator.

Shekhawat et al. (2020) [9] discuss the design, construction, and control of a two-wheel study-level, small self-balancing robot, using a CAD-designed, laser-cut, and 3D-printed structure.

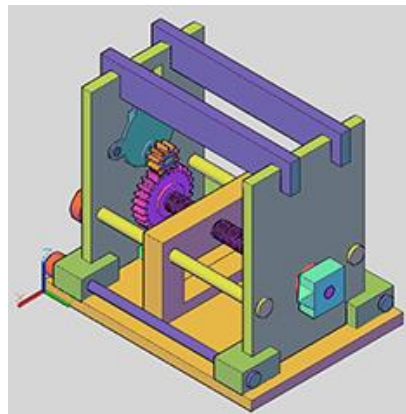
Qi et al. (2015) [11] The focus of the project was to design and develop the mechanism for a robotic arm for lifting. The 3D printing method is used in this project to fabricate the components of the robot arm. The robot arm is equipped with 4 servo motors to link the parts and bring arm movements. Arduino, open-source computer hardware, and software is applied to control the robotic arm by driving servo motors to be capable to modify the position.

Folgheraiter et al. (2018) [3] present a robot that measures 1.1 meters tall, using a combination of 3D printing techniques, and lightweight, using a servo motor.

Our research aims to design, and 3D print a robot that can precisely move a threaded prism along a threaded rod using Arduino Uno programming and maintain its orientation perpendicular to the axis of the threaded rod.

The primary goal of the project we are carrying out at the Metropolitan Autonomous University-Cuajimalpa is the "Development of a 3D printing robot to move a threaded prism in less than 3mm increments on the x-axis left/right and the y-axis up/down using a stepper motor and Arduino Uno."

In this paper, we present the development of one of the robot's parts formed by a structure that supports a threaded rod, a system of spur gears, a threaded prism, and a prism with an arrow (Figure 1).



**Figure 1:** Prototype modeling.

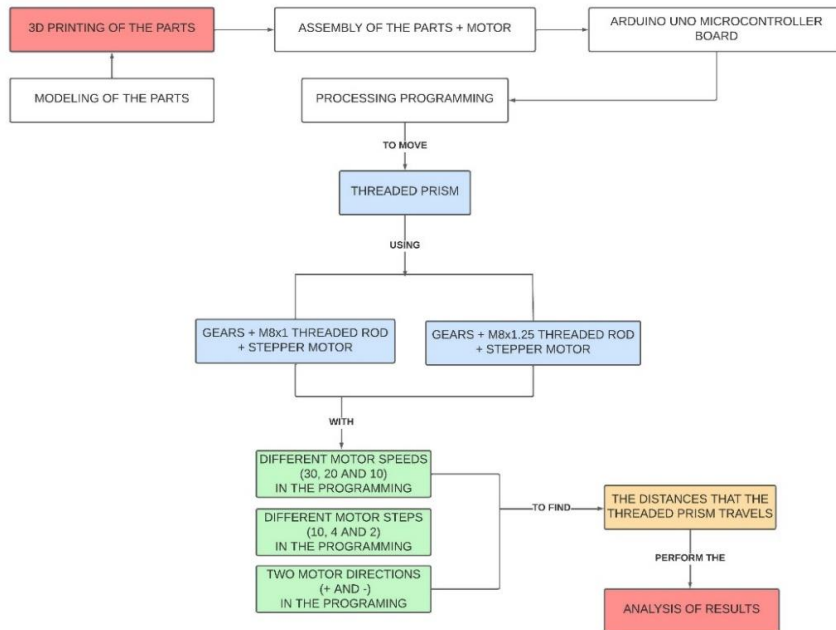
In the robot, the threaded rod is connected to the stepper motor through a gear system. Through programming on the Arduino Uno, we will control the number of steps of the stepper motor so that the threaded prism moves on the x-axis right/left in less than 3mm increments. We will verify that when the threaded prism moves, it maintains its orientation perpendicular to the axis of the threaded rod by drawing lines on a piece of paper. We designed and programmed a graphical user interface (GUI) for the user to input data to turn the stepper motor on and off so that the threaded prism moves in the desired direction.

This article has the following sections: Section 2, describes the methodology we followed, Section 3 describes the 3D modeling and printing of the prototype. Section 4 describes the development of the graphical interface and programming in Arduino UNO. Section 5 lists the

results, and Section 6 presents the conclusions. All the figures in this document are original and created by the author that works at the Metropolitan Autonomous University-Cuajimalpa.

## 2 METHODOLOGY

The methodology used to reach the results of the proposed objective is the following (Figure 2):



**Figure 2:** Methodology.

## 3 3D MODELING AND PRINTING OF THE PROTOTYPE

The prototype consists of 26 parts: a base, two walls, a threaded rod attached to the driven gear, a driving gear, six washers, four clamping cylinders, four lower supports, two upper supports, a prism with an arrow, a prism with a thread and two cylinders to hold the stepper motor (Figure 3).

The spur gear system consists of a 10-tooth drive gear with a 10mm diameter attached to the stepper motor. And a 25-tooth driven gear with a 25 mm diameter to reduce speed.

The transmission ratio ( $i$ ) of the gear is equal to 0.4, defined by the formula [2] (3.1). Were  $Z_e = 10$  and  $Z_s = 25$ .

$$i = \frac{Z_e}{Z_s} \quad (3.1)$$

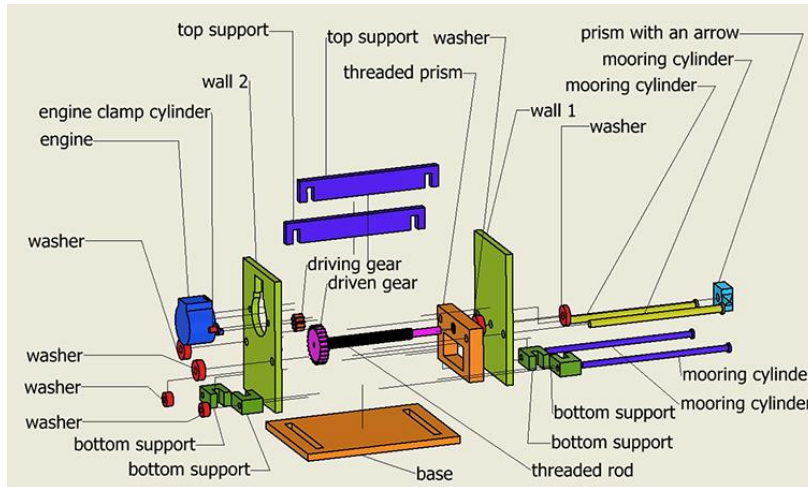
The driven gear speed ( $W_s$ ) is equal a 400 rpm, defined by the formula [2] (3.2). Were  $W_e$ , is the velocity of the driver gear, 1000 rpm.

$$W_s = \frac{Z_e \cdot W_e}{Z_s} \quad (3.2)$$

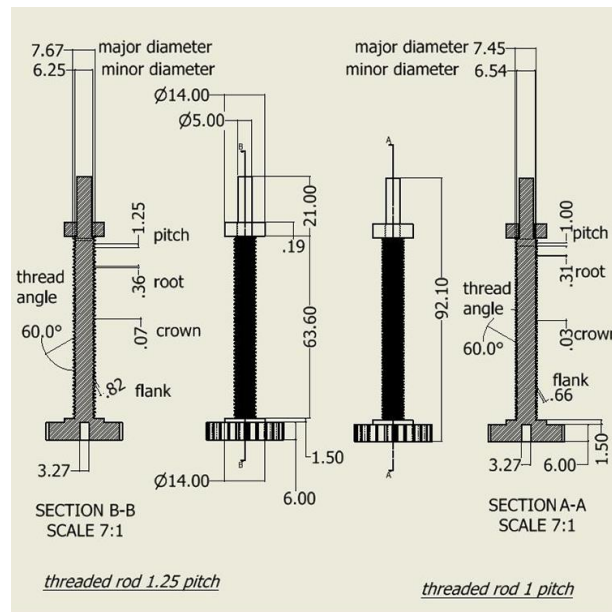
In the prototype design, the threaded prism will move according to the threaded rod's thread pitch at the stepper motor speed. The prism with an arrow is used to observe in which direction the threaded rod rotates.

We use a 28BYJ-48 stepper motor with an L298N driver, and program on the Arduino UNO, to perform various tests to verify that the interval increments when the threaded prism moves in the x-axis to the right and left are less than 3mm.

The first 3D-printed threaded rod is M8x1 (M = ISO metric; 8 = the basic major diameter of the thread, and 1 = is the pitch), and the second 3D-printed threaded rod is M8x1.25 (M = ISO metric; 8 = the basic major diameter of the thread; 1 = is the pitch, and 0.25 = is the length of the shift) (Figure 4).

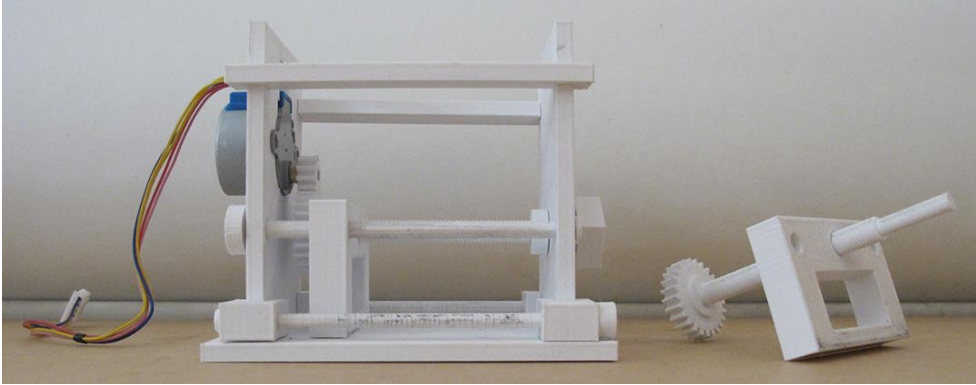


**Figure 3:** Exploded view of the prototype.



**Figure 4:** Section views threaded rod.

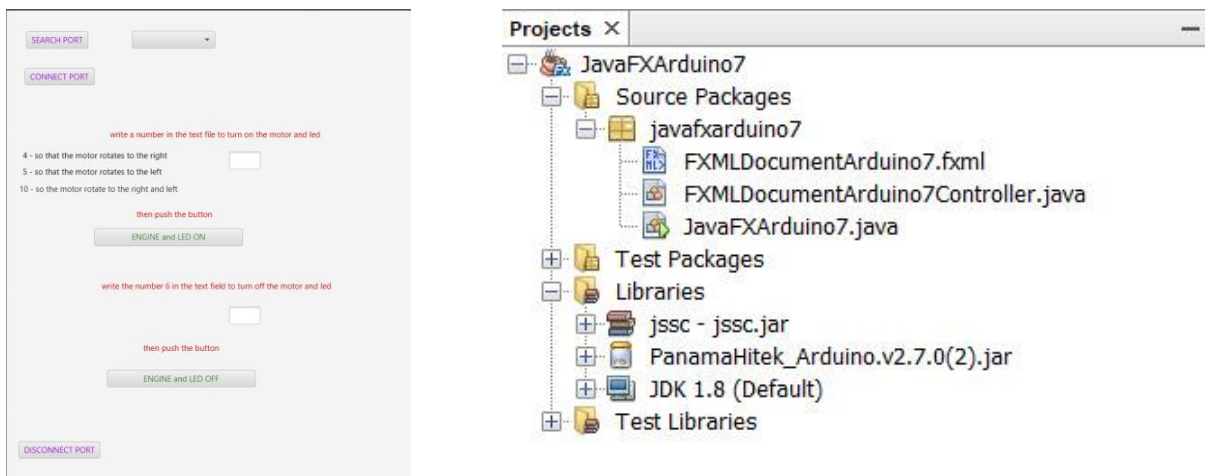
All the prototype was modeled in the Fusion 360™ program and printed on the MakerBot Replicator 2 printer using PLA (polylactic acid) filament (Figure 5). PLA filament is an organic and renewable material used to create complex parts and prototypes in 3D printing since it has tensile strength, its hardness is high, its flexibility is low, and its shrinkage is minimal.



**Figure 5:** 3D printing of the prototype.

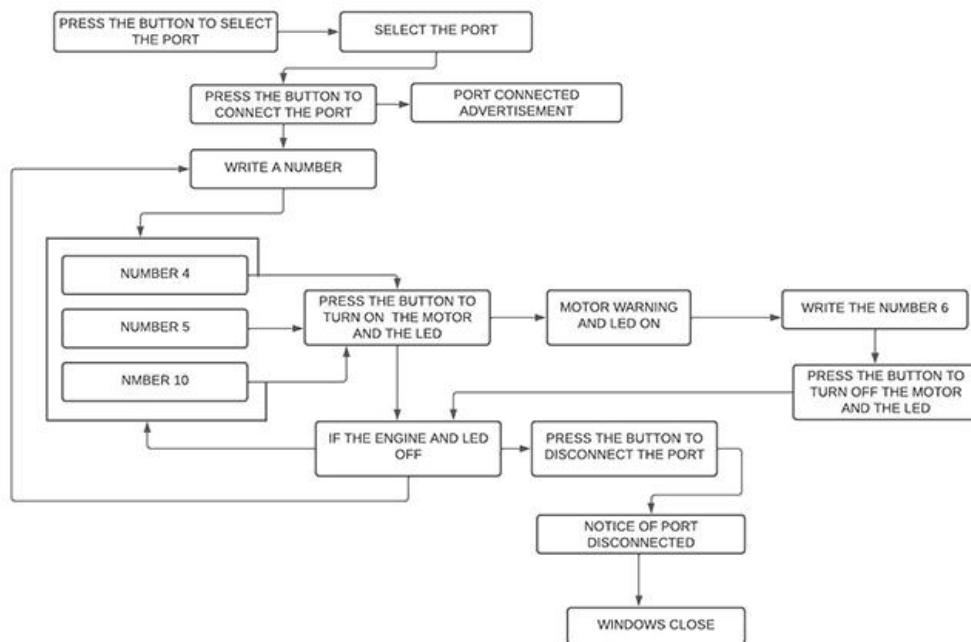
#### 4 DEVELOPMENT OF THE GRAPHICAL INTERFACE AND PROGRAMMING IN ARDUINO UNO

For the user to interact between the prototype and the Arduino Uno programming, the graphical interface (Figure 6 (a)) in NetBeans IDE 8.2 Scene Builder [8] and the interface schematic were designed and programmed (Figure 7).



**Figure 6:** (a) Graphical interface, and (b) JavaFX Arduino 7 project tree.

In the source package folder of the project (Figure 6 (b)) is the javafxarduino7 folder (which is the name of the project). Inside this folder are three FXML folders. Ayoma Joseph (2021) [4] explains in her article that FXML as the name suggests is an XML-based scripting language. It is designed to help build application user interfaces for JavaFX-based applications. FXML helps build a JavaFX graphical representation of the UI called scene. The resultant file is an FXML file. An XML file format representing the JavaFX UI when using the scene builder.



**Figure 7:** Interface schematic.

In the first folder of the project tree are the properties of all the elements, the labels, buttons, text fields, and the texts placed in the Scene Builder. In the second folder of the project tree are the stage, the main, and the serial event properties. In this folder is where the serial port of the Arduino UNO was programmed to connect. The third folder of the project tree is where all the buttons are programmed.

In the library folder of the project tree, we add two libraries. To work with the serial port was used the `jssc-jssc.jar` library [6]. For communication with Arduino, the `PanamaHitek_Arduino` [7] library is used. This library includes the classes belonging to the RXTX library.

The Arduino Uno was programmed with different speeds (30, 20, and 10), with three stepper motor steps (10, 4, and 2), and the stepper motor rotation in both directions (- and +). Added a LED that turns on and off to check when the motor turns on and off.

In the programming, we define the Stepper library, the variables (`pasosMotor` (1024), `distancia`, `dato`, `input`), and the number of the pin where we connected the LED to the Arduino UNO.

In the void `setup()`, we defined that the LED is the output and that the serial communication is 9600 Bauds per second.

In the void `loop()` six variables we define: `loop0`, `loop1`, `loop4`, `loop5`, `loop6` and `loop10`. In void `loop0` turn on the LED, in void `loop1` turn off the LED, in void `loop4` moves the stepper motor to the right (+) according to the number of steps, and the LED turns on (`loop0`). In void `loop5` moves the stepper motor to the left (-) and turns on the LED (`loop0`). In void `loop6` turns off the stepper motor and turn off the LED (`loop1`). And in void `loop10` moves the motor to the right (`loop4`) and the left (`loop5`).

In `loop4` and `loop5`, we define the loop with the following programming. In this case, the threaded prism will advance every 2.5mm (M8X1 threaded rod) until reaching the total distance of 47.5mm, executing the instruction 19 times.

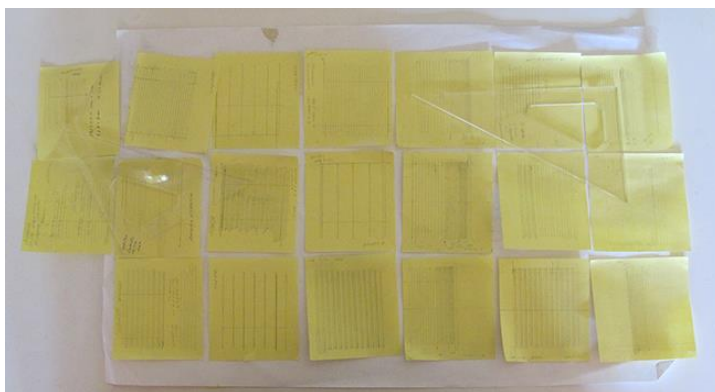
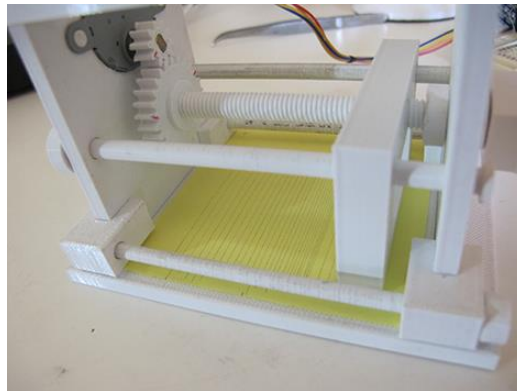
```

for(int h = 0; h<=18; h++){
  motor1.setSpeed(30);
  motor1.step(distancia*10);

  for(int i = 0; i<=18; i++){
    motor1.setSpeed(30);
    motor1.step(-distancia*10);
  }
}

```

In the programming, we use a delay of 2000 microseconds (two seconds) so that when the processor waits, we mark the line on the paper where the threaded prism stops (Figure 8 (a), and (b)).

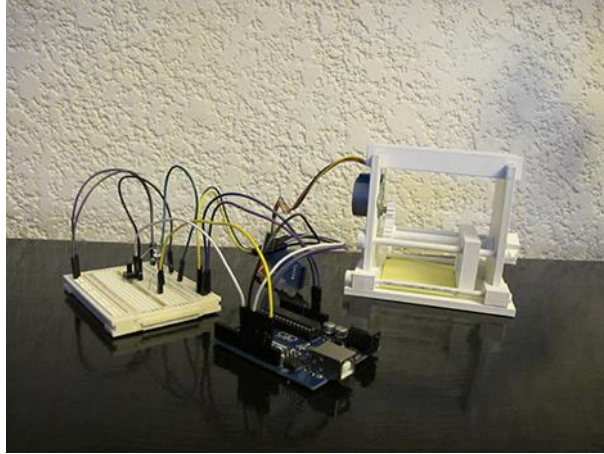


**Figure 8:** Lines marked. (a) On the prototype paper, and (b) On the paper.

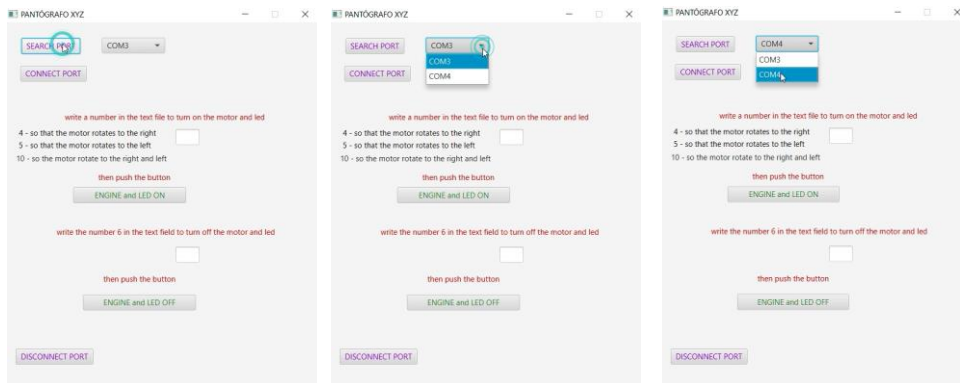
## 5 RESULTS

We performed several tests with the graphical interface to verify that the 3D-printed threaded prism moves correctly in less than 3mm increments along the threaded rod and maintains its orientation perpendicular to the axis of the threaded rod using Arduino Uno programming. (Figure 9).

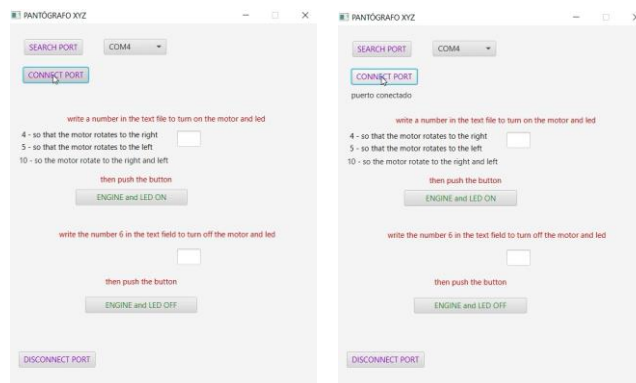
Through the video frames we took, we show how the prototype works using the graphic interface designed and programmed in Scene Builder with FXML code and with Arduino UNO programming. Figures 10, 11, 12, 13, and 14 describe the steps of the interface schematic.



**Figure 9:** Prototype in 3D printing and Arduino UNO.



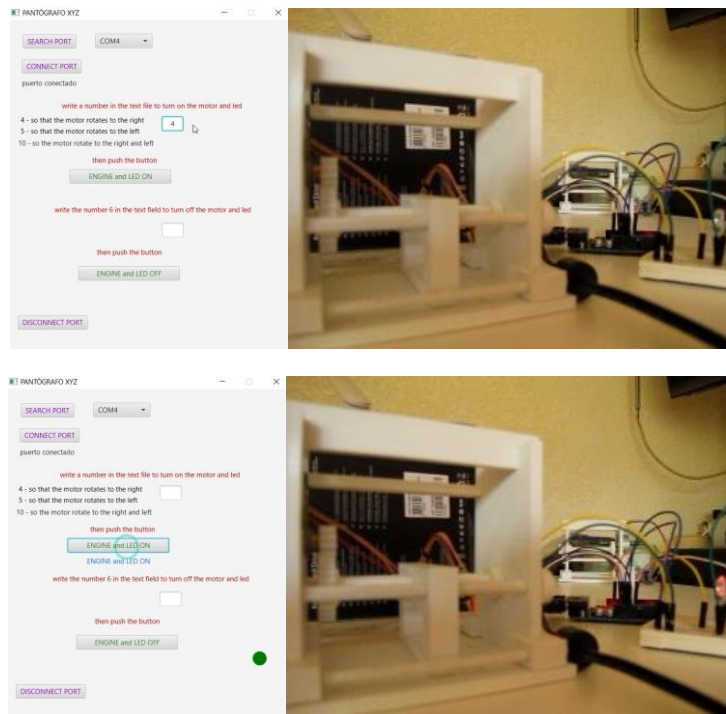
**Figure 10:** (a) Press the button to search the serial port, (b) Select the serial port, and (c) The selected serial port is COM4.



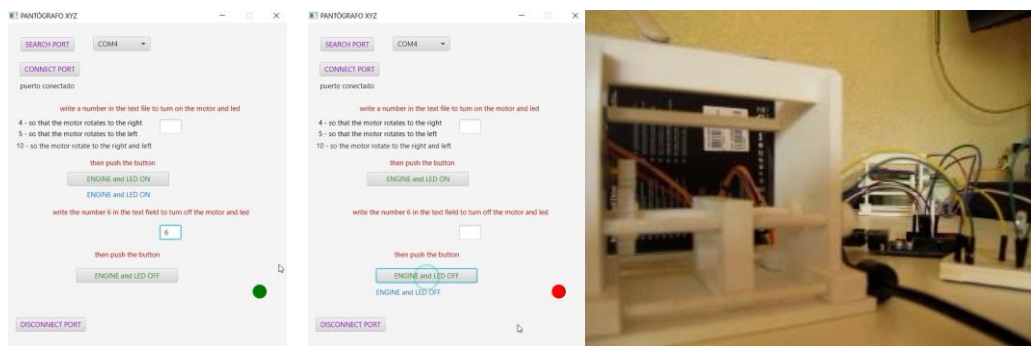
**Figure 11:** (a) Press the button to connect the serial port, and (b) Port-connected advertisement.



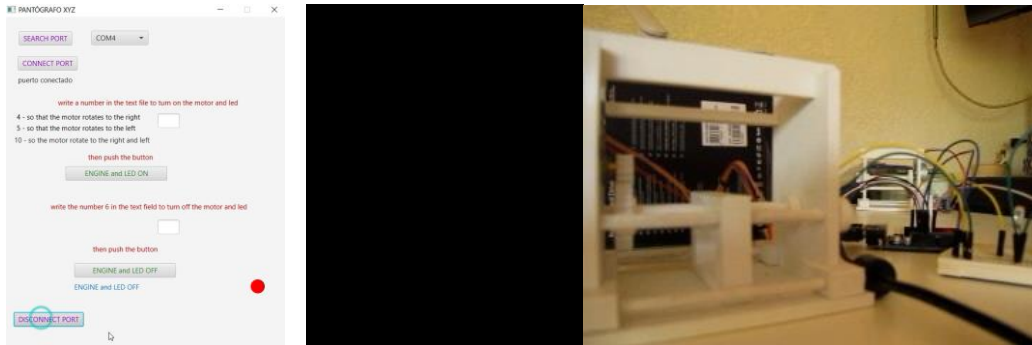
Once the serial port is connected, we can write the numbers 4, 5, or 10 in the text field. With number 4, the stepper motor will rotate CW; with number 5, the stepper motor will rotate ACW. And, with the number 10, the stepper motor will rotate CW and ACW.



**Figure 12:** (a) The number 4 was written in the text field, and (b) Press the button, an advertisement that the stepper motor and the LED turned on. The prototype, the gears, the threaded rod, and the threaded prism are moving, and on the Arduino UNO, the LED is on.



**Figure 13:** (a) The number 6 was written in the text field and (b) Press the button, an advertisement that the stepper motor and the LED were turned off. The prototype, the gears, the threaded rod, and the threaded prism stopped, and on the Arduino UNO, the LED is off.



**Figure 14:** (a) Press the button to disconnect the serial port, and (b) System out.

After completing several tests and corrections with the Scene Builder both the stepper motor and the LED turn on when the numbers 5, and 10 are written in the text field, as well the stepper motor and the LED turn off when the number 6 is written in the text field.

According to the proposed methodology, with the data in Table 1, the analysis of the 3D printing system will be carried out in the displacement of increments of less than 3mm through the Arduino UNO programming.

	Case 1 Gear + M8x1 road	Case 2 Gear + M8x1.25 road
motor speed	30, 20 and 10	30, 20 and 10
motor steps	10, 4 and 2	10, 4 and 2
motor direction	+ and -	+ and -
delay	2000	2000

**Table 1:** Cases 1 and 2.

*Case 1*

In this case, we use the 3D-printed threaded rod M8x1.

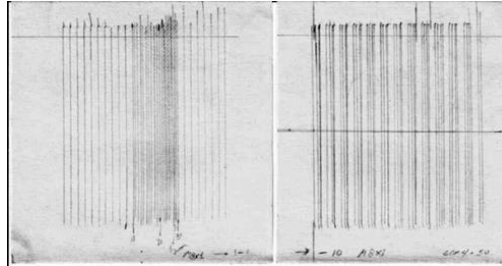
Due to residues left by 3D printing, the threaded prism was not initially coupled to the threaded rod. After several passes, the 3D printing residues were removed due to friction.

With stepper motor speeds of 30 and 20, the thread of the prism skipped the thread pitch of the threaded rod. Finding that the distance that the threaded prism moved was variable and the line that was marked on the paper was not perpendicular to the axis of the threaded rod.

With stepper motor speeds of 10, motor step 10, and in the two directions (+ and -), the thread of the prism did not skip the pitch of the threaded rod. In this case, the driving gear makes five turns, and the driven gear makes two turns. By marking the lines on the paper, we verify that the distance between the lines is, 2mm.

We continue the analysis by marking the lines of the steps on the paper, first of 10, 4, and 2 steps at every ten intervals, and second, we alternate steps 10, 4, and 2 at a single interval.

In all cases, the lines that were marked on the paper are perpendicular to the axis of the threaded rod (Figure 15). The distance that the threaded prism will move according to the number of steps of the stepper motor is indicated in Table 2.

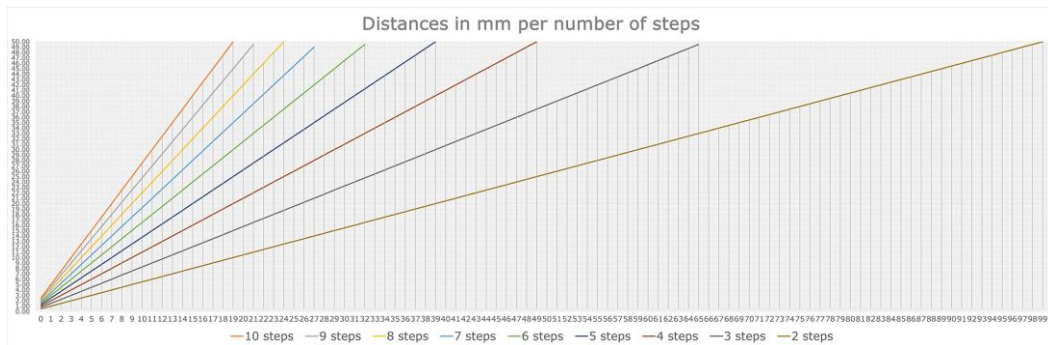


**Figure 15:** Lines marked on the paper, case 1.

Steps	10	9	8	7	6	5	4	3	2
Distance (mm)	2	1.8	1.6	1.4	1.2	1	0.8	0.6	0.4

**Table 2:** Steps and distances case 1.

The graph of Figure 16 shows the analysis of the steps and distances of case 1. This graph will allow us to program in Arduino UNO different combinations between the number of steps for the stepper motor and different increments less than 3mm (Table 4) or total distances for the threaded prism (Table 3).



**Figure 16:** Analysis steps and distances, case 1.

<code>for(int h = 0; h&lt;=18; h++){</code>		<code>for(int h = 0; h&lt;=26; h++){</code>		<code>for(int h = 0; h&lt;=48; h++){</code>	
Steps	Total distance in mm	Steps	Total distance in mm	Steps	Total distance in mm
10	50.00	7	49.00	4	50.00
9	45.00	6	42.00	3	37.50
8	40.00	5	35.00	2	25.00
7	35.00	4	28.00		
6	30.00	3	21.00		
5	25.00	2	14.00		
4	20.00				
3	15.00				

2	10.00				
---	-------	--	--	--	--

**Table 3:** Examples of steps and total distances according to Arduino UNO programming, case 1.

	if	0	1	2	3	4	5	6
10 steps	2	4	6	8	10	12	14	
9 steps	1.8	3.6	5.4	7.2	9	10.8	12.6	
8 steps	1.6	3.2	4.8	6.4	8	9.6	11.2	
7 steps	1.4	2.8	4.2	5.6	7	8.4	9.8	
6 steps	1.2	2.4	3.6	4.8	6	7.2	8.4	
5 steps	1	2	3	4	5	6	7	
4 steps	0.8	1.6	2.4	3.2	4	4.8	5.6	
3 steps	0.6	1.2	1.8	2.4	3	3.6	4.2	
2 steps	0.4	0.8	1.2	1.6	2	2.4	2.8	
CASE1	1	2	3	4	5	6	7	

**Table 4:** The number of steps and different increments is less than 3mm, case 1.

### Case 2

In this case, we use the 3d-printed threaded rod M8x1.25.

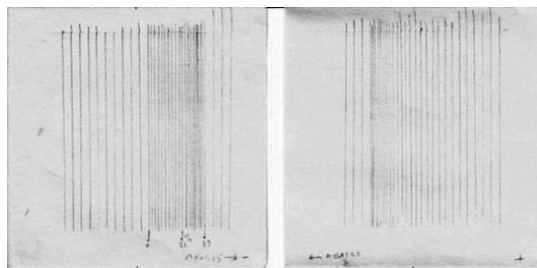
As in case 1, initially, the threaded prism did not couple to the threaded rod due to the residues left by the 3D printing. After several passes, the 3D printing residues were removed due to friction.

With stepper motor speeds of 30, 20, and 10, the threaded prism did not skip the thread pitch of the threaded rod.

With stepper motor speeds of 10, motor step 10, and in the two directions (+ and -), the thread of the prism did not skip the pitch of the threaded rod. In this case, the driving gear makes five turns, and the driven gear makes two turns. By marking the lines on the paper, we verify that the distance between the lines is, 2.5 mm.

We continue the analysis by marking the lines of the steps on the paper, first of 10, 4, and 2 steps in intervals of ten, and second, we alternate steps 10, 4, and 2 at a single interval.

In all cases, the lines marked on the paper were perpendicular to the axis of the threaded rod (Figure 17). The distance that the threaded prism will move according to the number of steps of the stepper motor is indicated in Table 5.

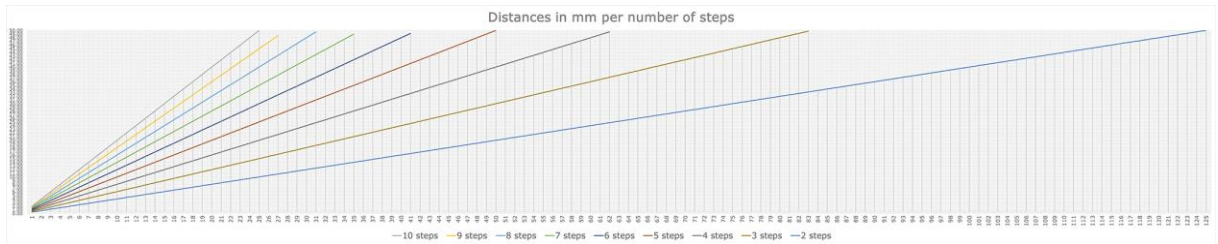


**Figure 17:** Lines marked on the paper, case 2.

Steps	10	9	8	7	6	5	4	3	2
Distance (mm)	2.5	2.25	2	1.75	1.5	1.25	1	0.75	0.5

**Table 5:** Steps and distances case 2.

The graph of Figure 18 shows the analysis of the steps and distances of case 2. This graph will allow us to program in Arduino UNO different combinations between the number of steps for the stepper motor and different increments less than 3mm (Table 7) or total distances for the threaded prism (Table 6).



**Figure 18:** Analysis steps and distances, case 2.

<code>for(int i = 0; i&lt;=18; i++){</code>		<code>for(int i = 0; i&lt;=26; i++){</code>		<code>for(int i = 0; i&lt;=48; i++){</code>	
Steps	Total distance in mm	Steps	Total distance in mm	Steps	Total distance in mm
10	40.00	6	44.80	5	50.00
9	36.00	7	39.20	4	40.00
8	32.00	6	33.60	3	30.00
7	28.00	5	28.00	2	20.00
6	24.00	4	22.40		
5	20.00	3	16.80		
4	16.00	2	11.20		
3	12.00				
2	8.00				

**Table 6:** Examples of steps and total distances according to Arduino UNO programming, case 2.

if	0	1	2	3	4	5
10 steps	2.5	5	7.5	10	12.5	15
9 steps	2.25	4.5	6.75	9	11.3	13.5
8 steps	2	4	6	8	10	12
7 steps	1.75	3.5	5.25	7	8.75	10.5
6 steps	1.5	3	4.5	6	7.5	9
5 steps	1.25	2.5	3.75	5	6.25	7.5
4 steps	1	2	3	4	5	6
3 steps	0.75	1.5	2.25	3	3.75	4.5
2 steps	0.5	1	1.5	2	2.5	3

CASE 2	1	2	3	4	5	6
--------	---	---	---	---	---	---

**Table 7:** The number of steps and different increments is less than 3mm, case 2.

#### *Comparison between the two cases*

The exposed analysis explains that with the two 3D-printed threaded rods, M8x1 or M8x1.25, the threaded prism moves less than 3mm. The comparison between the two cases from the point of view of the number of steps, number of loops, and the different increments of 1, 2, and 3mm analyzed are exposed in Table 8. The other distances are not repeated in the two cases.

	Case 1	Case 2
Minimum increment (mm) two steps one loop	0.4	0.5
Maximum increment (mm) ten steps one loop	2.0	2.5
Steps to reach 3mm	5 and 3	6, 4, 3 and 2
Steps to reach 2mm	10, 5 and 2	8, 4 and 2
Steps to reach 1mm	5	4 and 2
Loops to reach 3mm	3 and 5	2, 3, 4 and 6
Loops to reach 2mm	1, 2 and 5	1, 2 and 4
Loops to reach 1mm	1	1 and 2
Loops to reach 3, 2, and 1mm at the same step	5	4 and 2

**Table 8:** Comparison of the number of steps, number of loops, and the different increments between the two cases.

With this last analysis, we can define in the final project which threaded rod is the most convenient for the minimum and maximum increments of the threaded prism since we can be made several combinations between the number of steps and the number of loops.

## 6 CONCLUSIONS

Throughout this writing, we have presented the design, development, and programming of a robot prototype manufactured by 3D printing, consisting of a stepper motor, a gear system, a threaded rod, a threaded prism, and Arduino UNO. Our research objective is to verify that the threaded prism moves in increments less than 3mm and that it remains perpendicular to the axis of the threaded rod by using a stepper motor programmed in Arduino UNO. The most relevant aspect of our project is that increments smaller than 3mm are achieved through different steps and turns, and we can program a matrix in Arduino UNO with the values found in Figures 16 and 18. In case our distance objective is 24.5mm, we will program a matrix with the value of the distances found in Figures 16 and 18, to find the closest value to the objective distance, determine in which row and column of the matrix it is located, and which is the first number of the row in which that value is found. If we use an M8x1.25 threaded rod, the result of the programming would be the following: the lowest value found in the matrix is 24.0, it is in row 2, column 11 and the first number in row 2 is 2.0; the lowest value found in the array is 24.0, it is in row 4, column 15, and the first number in row 4 is 1.5. As can be seen, the value closest to the target distance is

repeated twice in the matrix, which gives us two alternatives, the values 2.0 and 1.5, which would be the increments in which the threaded prism could be moved. By presenting these alternatives, our research will contribute to the development of 3D printing of customized robots with specific functions and more precise controls programmed with the Arduino Uno, which could find applications in various research fields, such as biology, anthropology, and medicine.

## ACKNOWLEDGMENTS

I want to express my gratitude to Miriam Hernández Ramírez for her support in the analysis of the movement of the gears and the programming review.

*Dina Rochman*, <http://orcid.org/0000-0001-8902-3513>

## REFERENCES

- [1] All3dp, <https://all3dp.com>, 3D printed robots.
- [2] Bailey, O.; Pickup, R.; Lewis, R.; Patient, P.: Modular course in technology, School Council, Oliver & Boyd, Great Britain, Edinburgh, Internet archive, ISBN 0050033867, 1988 Ninth Impression.
- [3] Folgheraiter, M.; Aubakir, B.: Design and modeling of a lightweight and low power consumption full-scale biped robot, *International Journal of Humanoid Robotics*, 15(5), 2018.
- [4] FXML, <https://www.section.io/engineering-education/mysql-connection-graphical-user-interface-design-using-javafx-and-scene-builder/#fxml>, Languages.
- [5] Hunt, G.; Mitzalis, F.; Alhinai, T.; Hooper, P.; Koval, M.: 3D orienting with flying robots, *IEEE International Conference on Robots & Automation (ICRA)*, 2014.
- [6] jssc-jssc.jar, [https://github.com/java-native/jssc\\_Library](https://github.com/java-native/jssc_Library).
- [7] PanamaHitek\_Arduino, [https://sourceforge.net/projects/arduinojava/files/v2.7.0/.\\_Library](https://sourceforge.net/projects/arduinojava/files/v2.7.0/._Library).
- [8] Scene Builder, [https://gluonhq.com/products/scene-builder/,\\_Graphical interface\\_](https://gluonhq.com/products/scene-builder/,_Graphical%20interface_)
- [9] Shekhawat, A.; Rohilla, Y.: Design and control of two-wheeled self-balancing robot using Arduino UNO, *Proceeding of the International Conference on Smart Electronics and Communication (ICOSE)*, 2020.
- [10] Telegenov, K.; Tlegezion, Y.; Shintemirov, A.: A low-cost open-source 3D printed three-finger gripper platform for reach and educational purposes, *IEEE Access*, 2015.
- [11] QI, A.; Nurfathin, M.; Meera, I.; Mohamad, H.: Design and development of a mechanism of robotic arm for lifting Part 1, 2<sup>nd</sup> Integrate Design Project Conference (IDPC) 2015.