Rack and pinion mechanism

Stratec Internship Challenge - Hardware

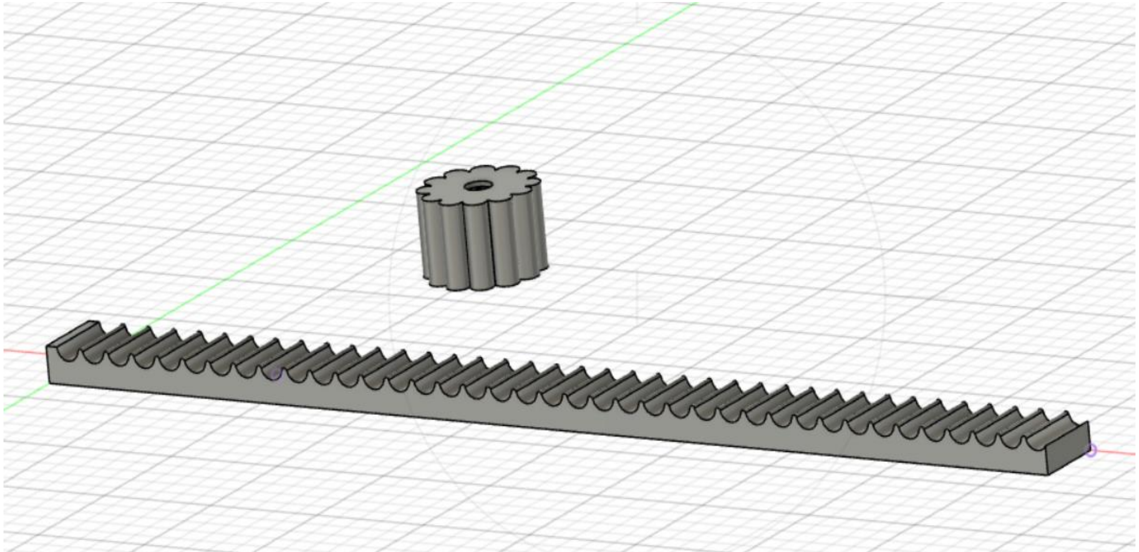
Aldea Ovidiu Alexandru

Technical University of Cluj-Napoca, Computer Science

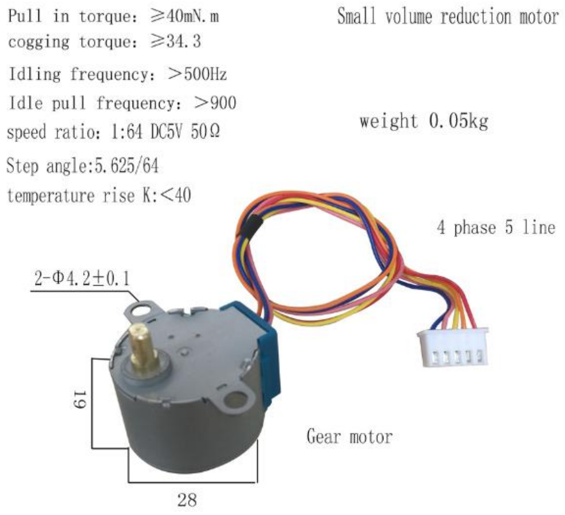
Contents

1. Components and circuit diagram

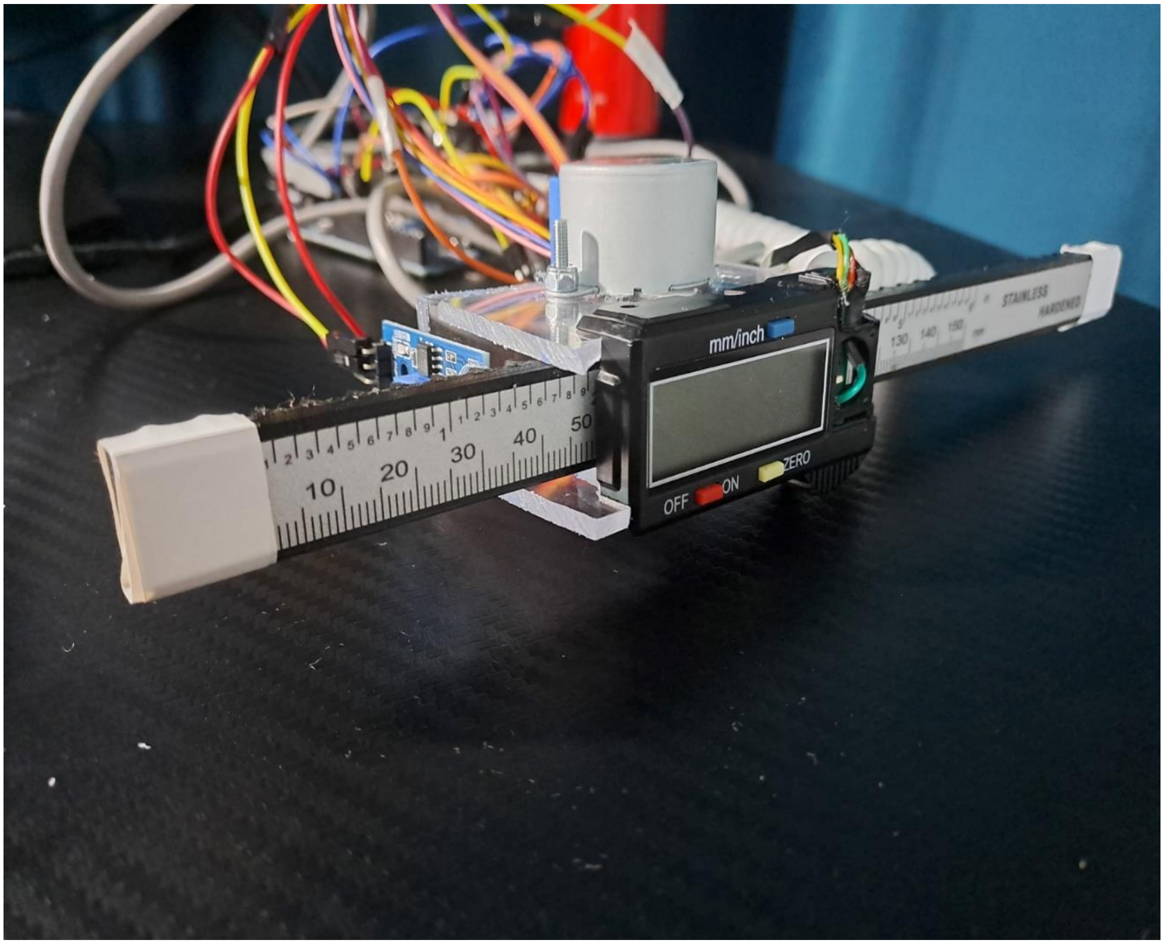
The device is built around a rack and pinion mechanism that I designed in CAD. The components were drawn using Fusion360 and they were 3D printed by a friend.



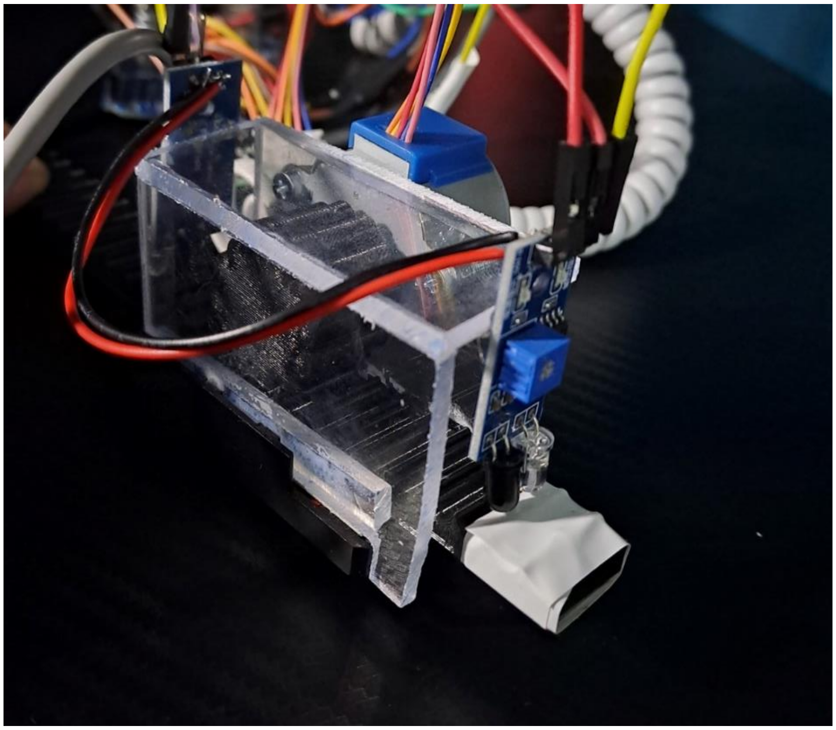
In order to rotate the pinion, I chose a 4 phase stepper motor, 28BYJ-48. The reasons why I chose this motor are: the speed of the motor can be controlled, it has a gear reduction that makes it really powerful with only 5V and it is also very precise. It has a few disadvantages, the main one being that it overheats really fast, even without a load. The motor is controlled using the ULN2003APG stepper motor driver.



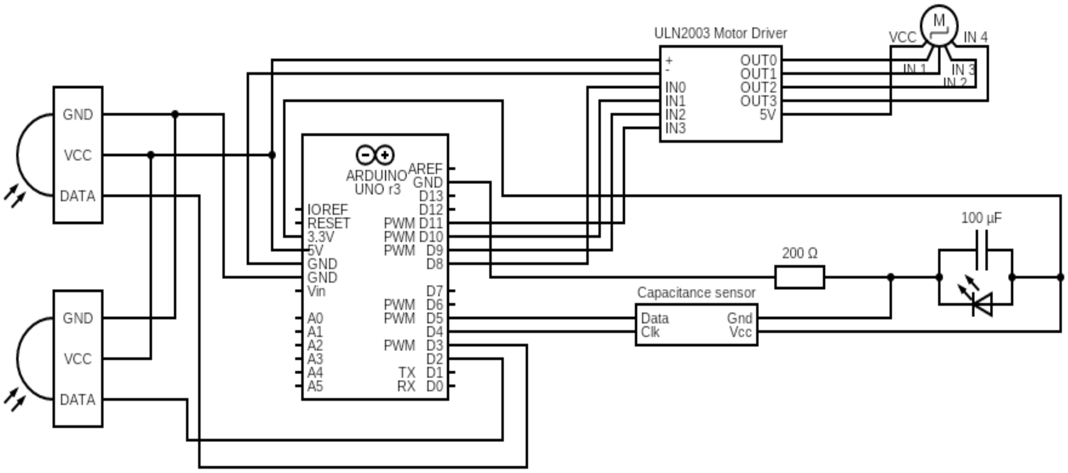
The position of the rack is measured using the capacitance encoder from digital calipers. The capacitance band is glued on the back of the rack and the pinion is set so it keeps a good connection between the encoder and the band.



One of the requirements of the challenge was to implement a way that restricts the mechanism to go above the rack limits. Firstly, I tried to use two D2FC-F-7N micro switches, but they were hard to position so I used IR sensors and some white electrical tape. It works really good and I found it more reliable than the switches.

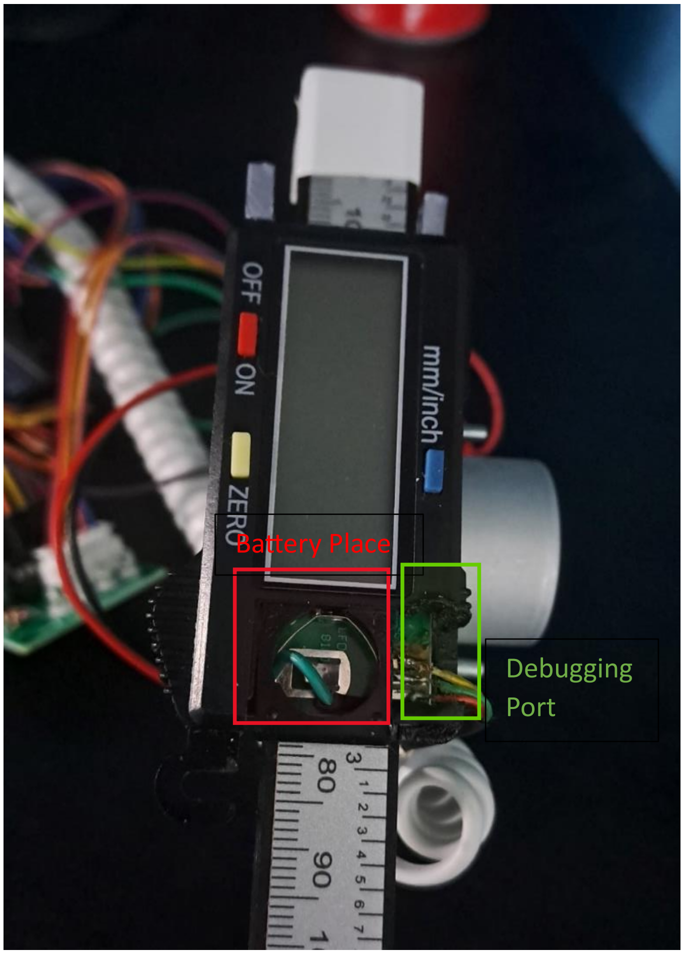


The whole mechanism is controlled by an Arduino Uno board. The user can communicate with the board using UART, in order to control the motor speed and rack position and see the current position of the rack in the user interface. The motor driver is connected to the pins D8, D9, D10, D11, 5v and GND; the encoder is connected to the pins D4, D5, 3.3v, GND; the IR sensors are connected to D2, D3, 5v, GND. The circuit diagram can be seen below.



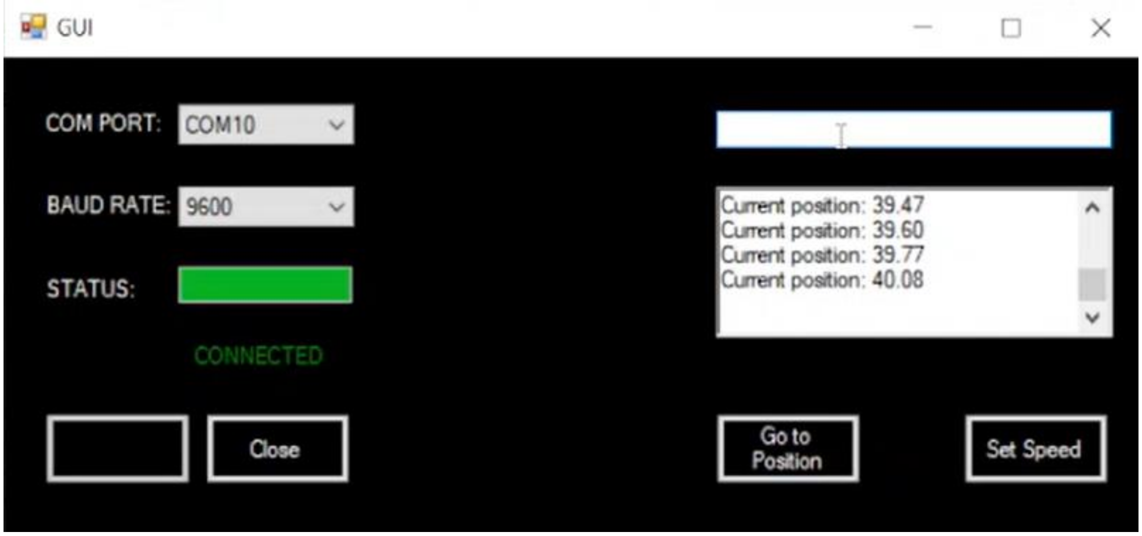
The device is powered via USB. The reason I chose to power the Arduino from USB is that I need the serial connection in order to control the device and the mechanism doesn't require a bigger power supply to spin the motor.

A bonus requirement was to remember the position after a power cycle. For this I wanted to use the ability of the encoder to keep the position after it is turned off. Unfortunately, I broke a pin from the debugging port, and I had to use the negative connector from the battery holder. Without a battery, the encoder cannot remember its last position.



2. User interface

The user interface was built using C# in Microsoft Visual Studio. The user can select the device port and baud rate, then connect to the device using the serial port. After the device is connected, the user can set the speed of the motor or set a position for the rack. The user can see in real time when the position of the rack changes.



COM PORT: COM10

BAUD RATE: 9600

STATUS:

CONNECTED

```
Current position: 39.47
Current position: 39.60
Current position: 39.77
Current position: 40.08
```

3. Arduino Code

The code begins with the declaration of the Stepper.h library, variables, pins and the initial setup. In the loop() function, firstly, the data from the encoder is decoded to find the current position. After that, the serial communication with the user is handled. If the user changes the speed, the program calls the setSpeed(speed) function from Stepper.h. If the user sets a new position, the moveIt(nextPosition) is called. Also, if the position changes, the loop function sends the position to be displayed.

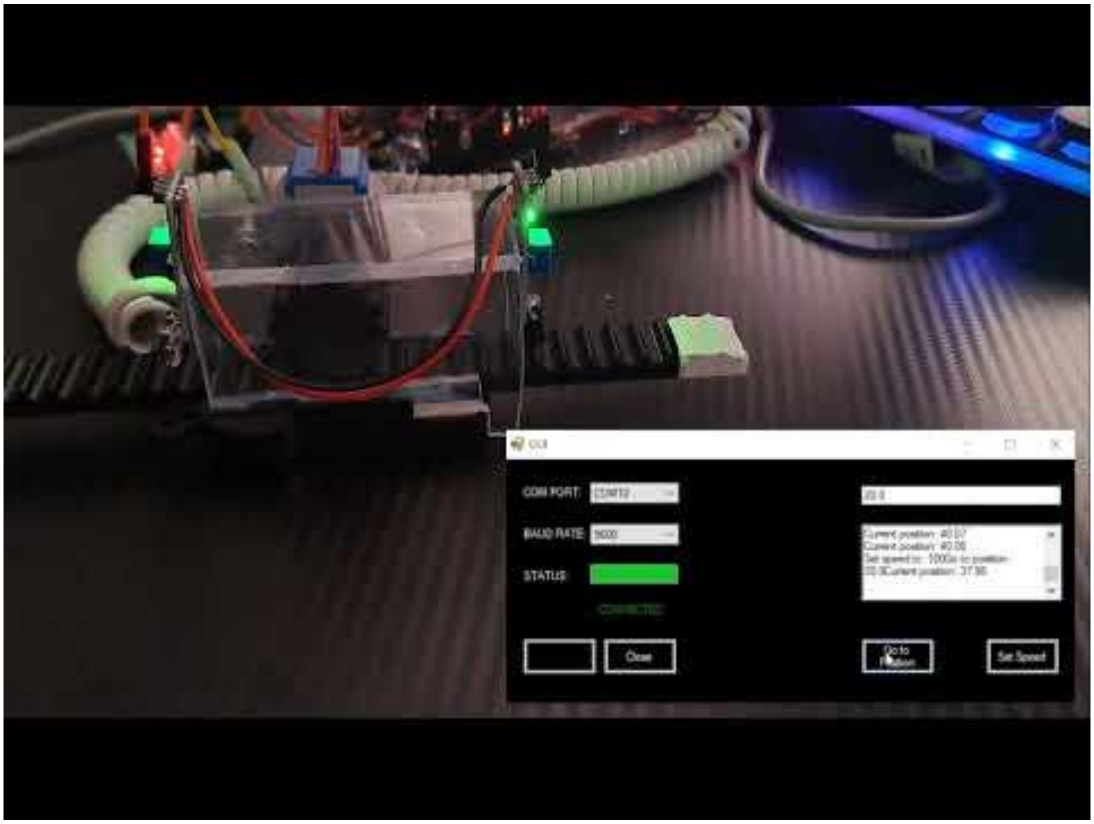
The decode() function returns the current position. The code for this function was written by Marthalprojects and more details about it can be found here.

The moveIt(nextPosition) function checks if the desired position is reached, otherwise it checks if the rack can move and if it didn't reach any limit. The motor steps are adjusted every time the rack gets closer to the desired position in order to be as accurate as possible.

4. Video

The video of the working mechanism can be seen below or on Youtube.

https://youtu.be/_ETW9B_RU-Y



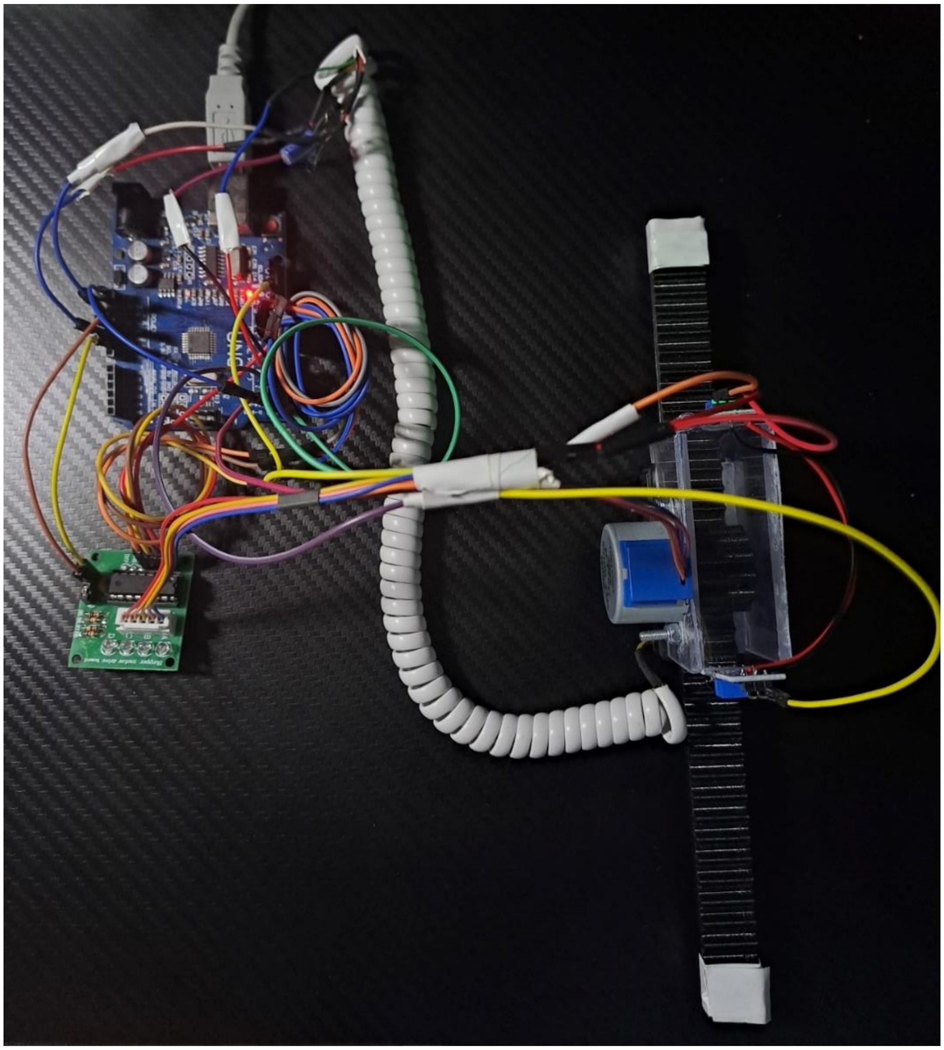
COM PORT: COM12

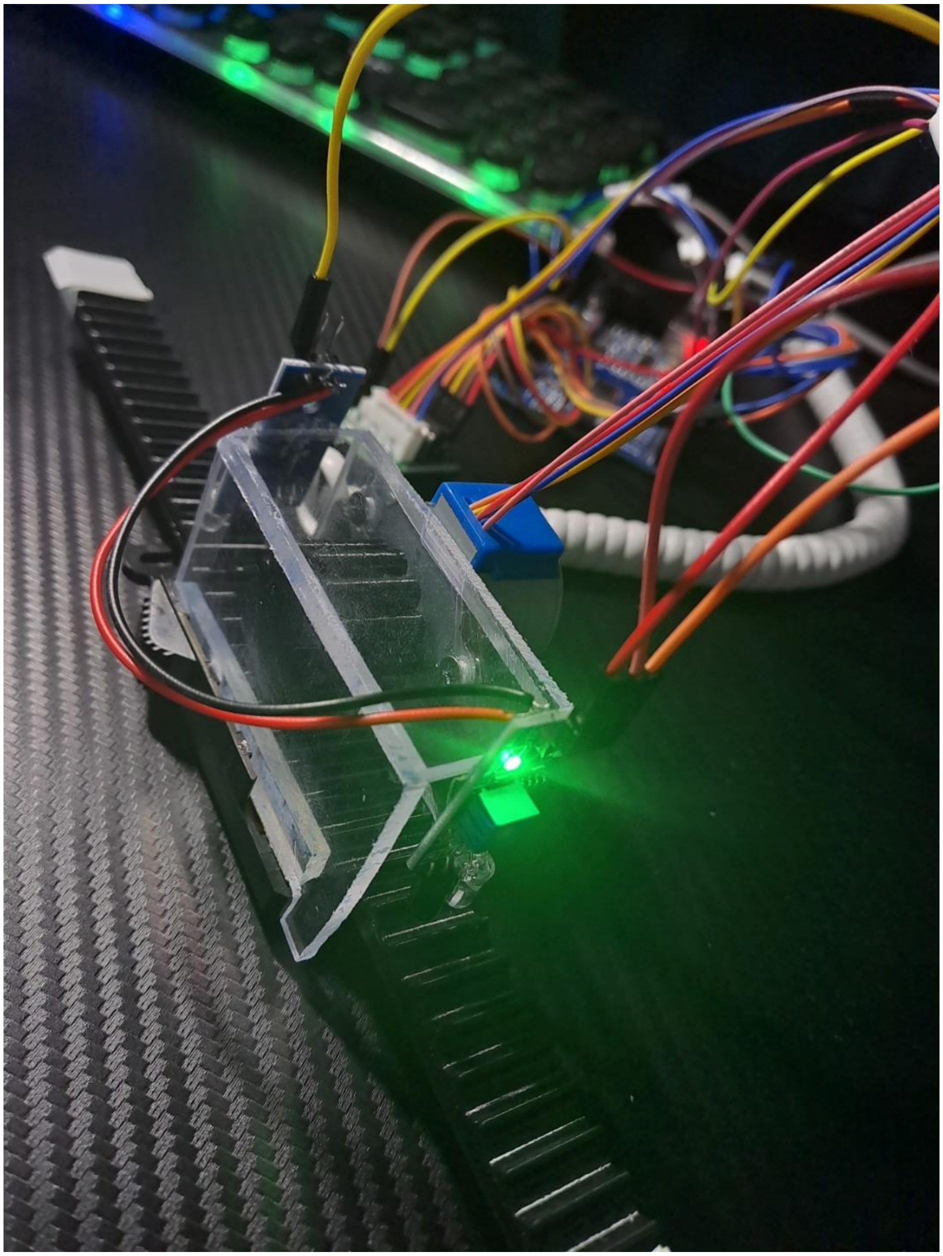
BAUD RATE: 9600

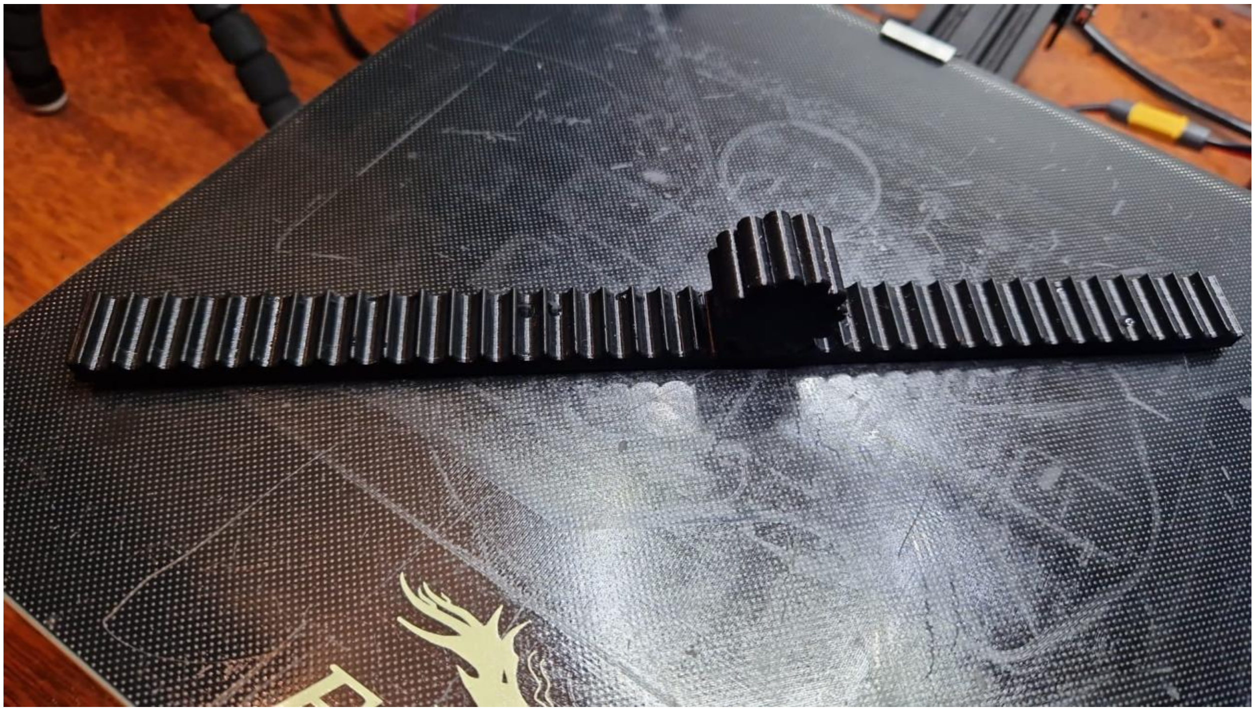
20.0

```
Current position: 40.07
Current position: 40.08
Set speed to: 1000Go to position:
20.0Current position: 37.98
```

5. Pictures







6. Bibliography

https://www.instructables.com/Reading-Digital-Callipers-with-an-Arduino-USB/

https://sites.google.com/site/marthalprojects/home/arduino/arduino-reads-digital-caliper

https://www.youtube.com/watch?v=o0AKEjtY4-4&ab_channel=CaturPebriandani