İZMİR KATİP ÇELEBİ UNIVERSITY
FACULTY OF ENGINEERING AND ARCHITECTURE
DEPARTMENT OF MECHATRONICS ENGINEERING

ROBOTIC PROJECT:
ROLLING CUBE

Ayana Tleulenova 160412059

Furkan Soylu 160412043

Abdussamet Şirin 180412065

Hasan Avcı 150412035

Ayşe Tomurcuk 170412001

SUPERVISOR: Doç. Dr. Erkin Gezgin

JANUARY,2022

Introduction:

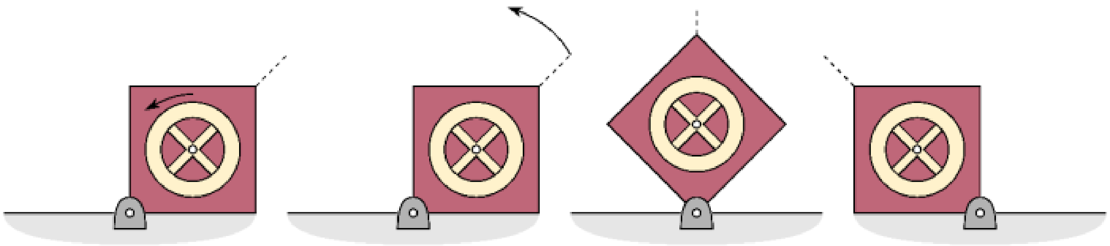
The purpose of this project is to create a cube that jumps due to the sudden transfer in angular momentum during braking causes. There are multiple ways to keep a cube in its balance, but jumping up requires a sudden release of energy. So, it means we need strong enough motors. We placed two motors in the cube to provide enough power. And the placement of these engines was done with the center of gravity in mind. We will do the braking with the microservo we use extra. 2400 rpm DC motors seemed like a good idea to store enough energy while still keeping the cube compact and self-contained. We apply a voltage supplier to dc motors with a value of 12 volts. This is the part used to distribute the required energy evenly. The braking system, on the other hand, will be due to the sudden stopping effect on the rotating wheels, which allows the robot to advance one step.

The main task wasto develop a cube which is able to jump up from being on its surface to one of its edges and balance there, like it`s shown on the figure below:

Modelling:

Working Principle:

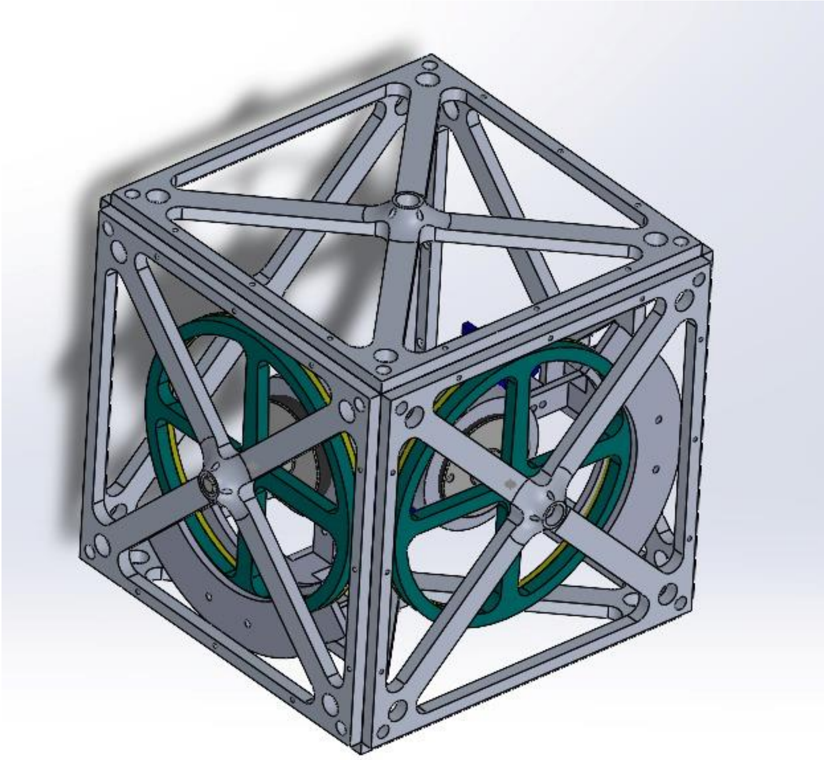
In this project we planned to use two motors and two braking systems in X and Y direction, for jump in this two directions. Moreover, we will attach both motors to the flywheels so that they can spin inside. In addition, we tried to make the flywheels heavier in order to increase the torque and the cube had enough energy to roll over to the other side.

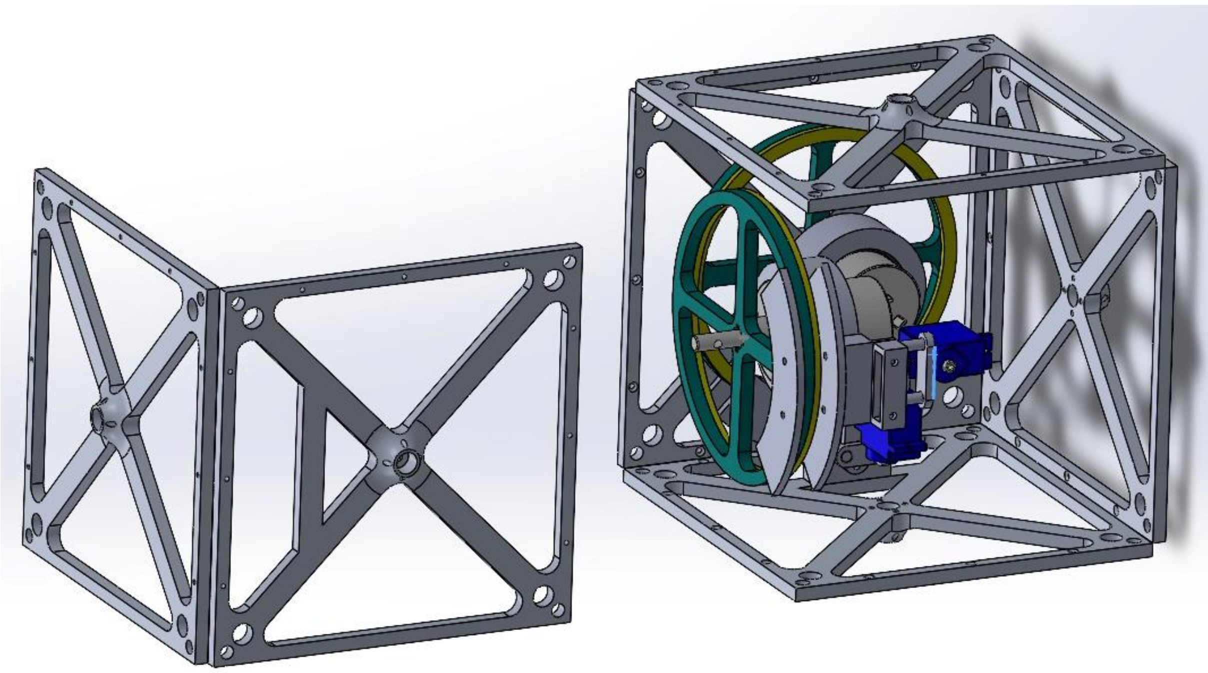


We decided to produce our cube from Flexi glass, on the ground that its` mass is not heavy. So, the motors` energy would be enough for jumping. It is also important that the cube is small, so we came to the conclusion that it should be 14 * 14 * 14 cm in size. For this procedure, we cut out the parts we need with the laser after we created the simulation using the SolidWorks program.

Comparing the early results given in simulations with the final results of the real system, it could be seen that all the simulation results are optimistic.

Here is our simulation:





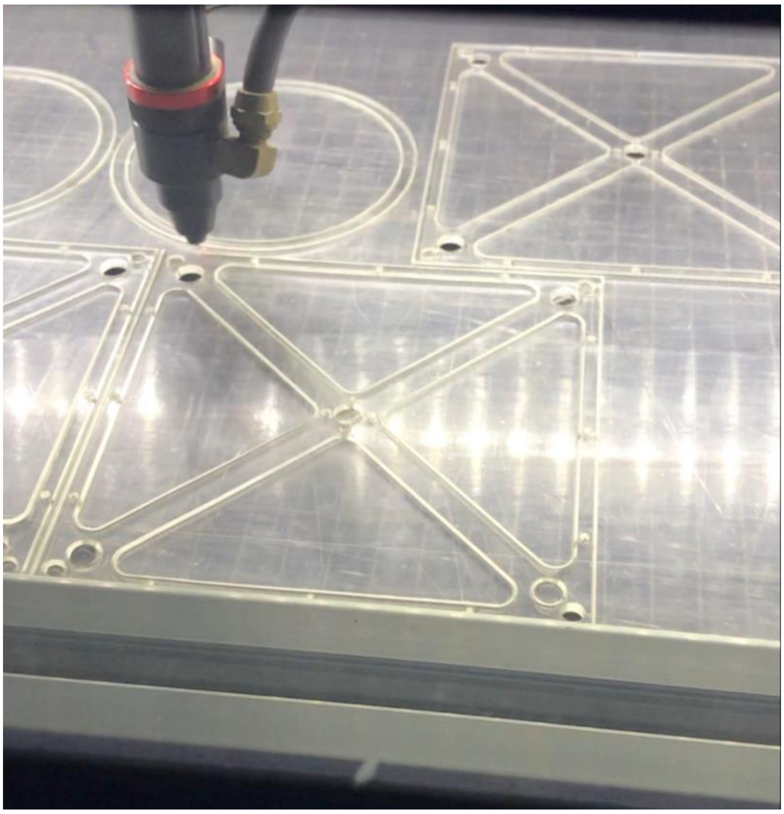
Manufacturing Process:

This section contains the system design.

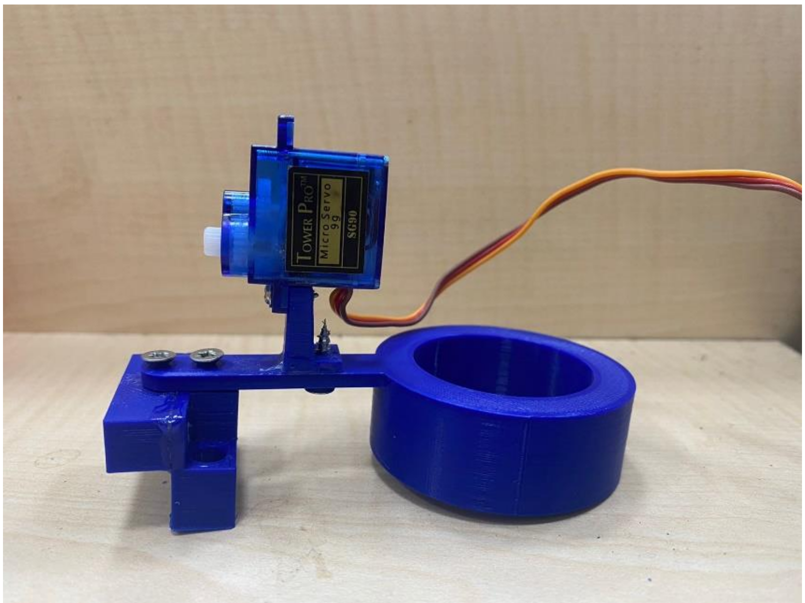
Components that we used:

- Flywheel
- Flywheel Holder
- Frame
- Motor driver
- Brake system actuator
- Arduino
- Dc Motor
- Servo Motor
- DC Power Supply

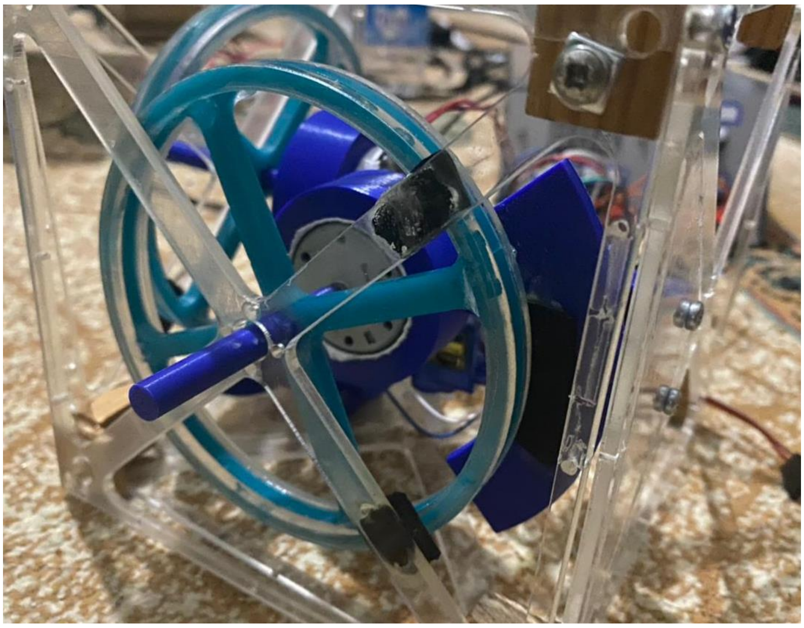
Laser Cut Process:



Also, it was important to create a good brake system. The braking system must work flawlessly and respond instantly to convert torque into inertia. We chose Tower Pro Micro Servo motor for this system. Here is our design that was printed by 3D printer:



We also glued pieces of rubber to the sides of the cube and the front of the brake system to increase the friction force and for better brake performance, the holders are pushed away from the reaction wheel with help of two rubber bands and are pressed together with the servo motor.it can be observed on the photo below:

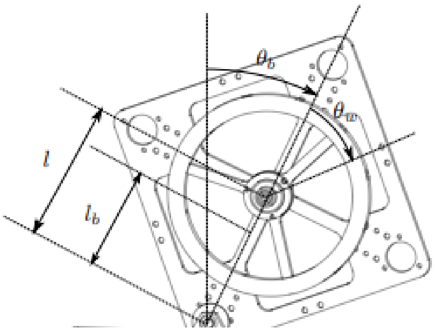


This

Calculations:

<u>System Dynamics :</u>

To do some calculations, we calculated the weight and angle values by considering the robot in one dimension.



Let θ_b represent the angle of inclination of the pendulum body, and θ_w the rotational displacement of the momentum wheel relative to the pendulum body. The nonlinear dynamics of the setup shown above.

$$\ddot{\theta}_b = \frac{(m_b l_b + m_w l) g \sin\theta_b - T_m - C_b\dot{\theta} + C_w\dot{\theta}_w}{I_b + m_w l^2}$$

$$\ddot{\theta}_w = \frac{(I_b + I_w + m_w l^2)(T_m - C_w\dot{\theta}_w)}{I_w(I_b + m_w l^2)} - \frac{(m_b l_b + m_w l) g \sin\theta_b - T_m - C_b\dot{\theta}}{(I_b + m_w l^2)}$$

where m_b, m_w are the pendulum body and wheel masses, I_b is moment of inertia of the pendulum body around the pivot point, I_w is the moment of inertia of the wheel and the motor rotor around the rotational axis of the motor, l is the distance between the motor axis and the pivot point, l_b is the distance between the center of mass of the pendulum body and the pivot point, $g = 9.81\ m.s^{-2}$ the gravitational acceleration, T_m is the torque produced by the motor, and C_w, C_b are the dynamic friction coefficients of the pendulum body and wheel.

<u>Jumping Up:</u>

Assuming a perfectly inelastic collision i.e., a zero coefficient of restitution, the angular velocity w_w of the momentum wheel required for jump-up is calculated using: The conservation of angular momentum during the impact

$$I_w w_w = (I_w + I_b + m_w l^2) w_b$$

where w_b is the angular velocity of the pendulum body after the impact. The conservation of energy after the impact (θ_b = π/4) and until the pendulum body reaches the top (θ_b = 0).

$$\frac{1}{2}(I_w + I_b + m_w l^2) {w_b}^2 = (m_b l_b + m_w l) g (1 - \frac{1}{\sqrt{2}}$$

Eliminating w_b from both equations gives

$$w_w{}^2 = (2-\sqrt{2})\frac{(I_w + I_b + m_w l^2)}{I_w{}^2}(m_b l_b + m_w l)g$$

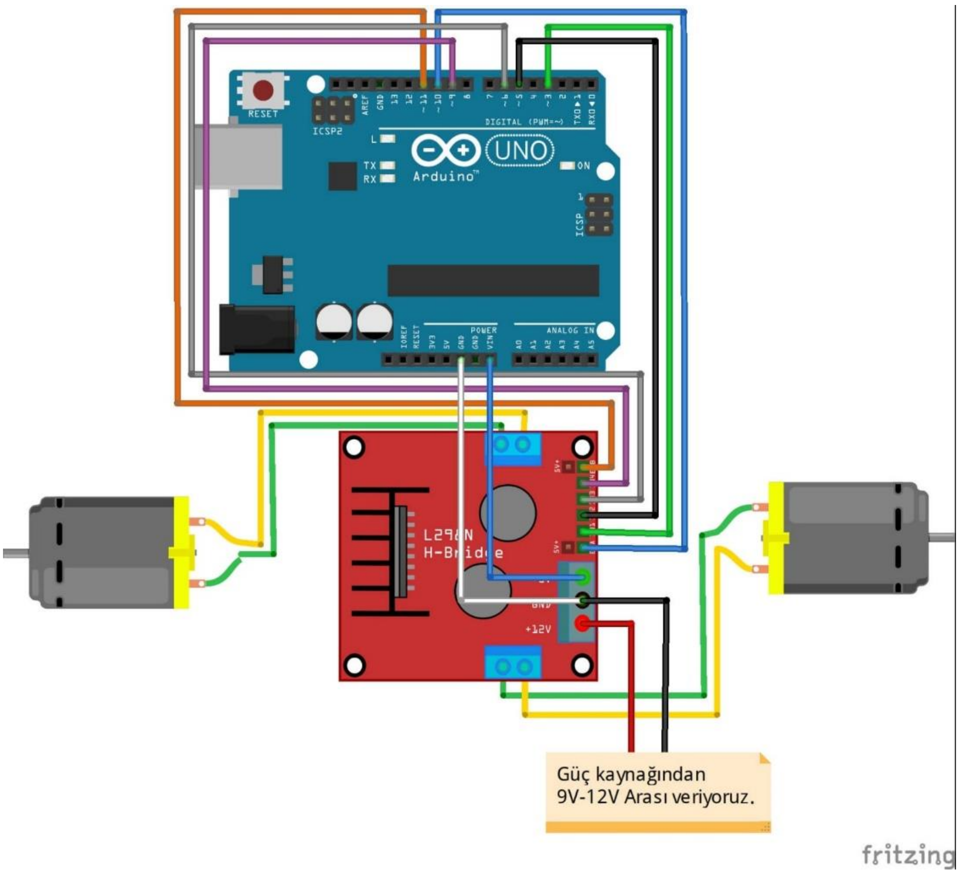
The uncertainties in the parameters or a non zero coefficient of restitution can make w_w deviate from the above calculated value. In this case a simple bisection-based trial and error learning procedure can be applied to learn the required ωw, exploiting the monotic relationship between θ_b achieved after the impact and w_w. The w_w given by above can be used as the initial condition in this case.

Electrical Circuit:

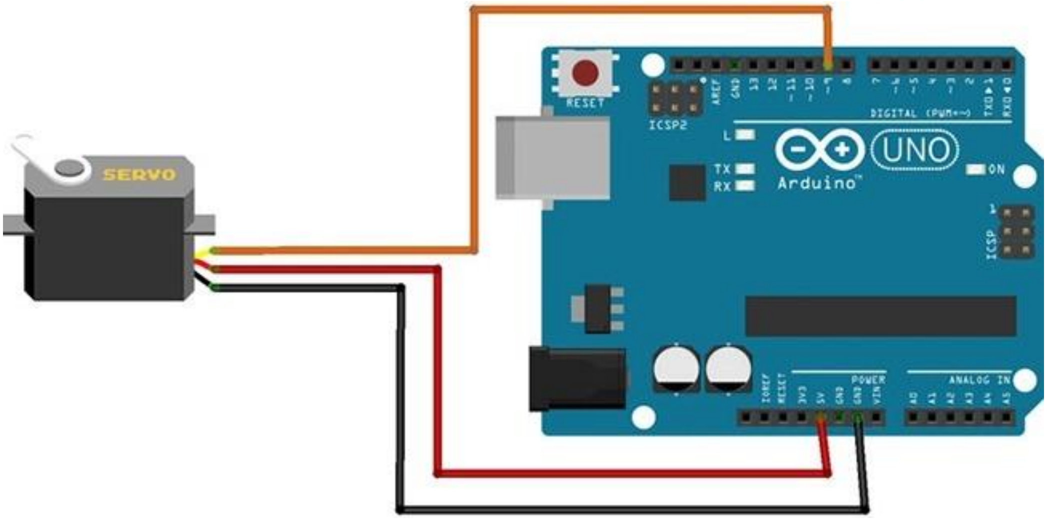
It was also important to correctly assemble the Arduino circuit for both motors.

First of all, for the ds motor, we used the Motor Driver in order to provide enough electrical energy to the motor to gain proper speed, torque as per requirement.

Here is the circuit:



Circuit for Servo motor:



Software:

For the Software we used Arduino.

Here is our code for ***DC motors***:

```
const int Enable_A = 10;
const int Enable_B = 11;
const int inputA1 = 3;
const int inputA2 = 5;
const int inputB1 = 6;
const int inputB2 = 9;
void setup()
{
pinMode(Enable_A, OUTPUT);
pinMode(Enable_B, OUTPUT);
pinMode(inputA1, OUTPUT);
pinMode(inputA2, OUTPUT);
pinMode(inputB1, OUTPUT);
pinMode(inputB2, OUTPUT);
}
void loop()
{
```

```
digitalWrite(Enable_A, HIGH);
digitalWrite(Enable_B, HIGH);
digitalWrite(inputA1, HIGH);
digitalWrite(inputA2, LOW);
digitalWrite(inputB1 , HIGH);
digitalWrite(inputB2, LOW);
delay(10000);
digitalWrite(Enable_A, LOW);
digitalWrite(Enable_B, LOW);
delay(10000);
digitalWrite(Enable_A, HIGH);
digitalWrite(Enable_B, HIGH);
digitalWrite(inputA1, LOW);
digitalWrite(inputA2, HIGH);
digitalWrite(inputB1 , LOW);
digitalWrite(inputB2, HIGH);
delay(10000);
digitalWrite(Enable_A, LOW);
digitalWrite(Enable_B, LOW);
delay(10000);
}
```

Conclusion:

In this project we learnt a lot. Firstly, it was a long way from creating the idea until the manufacturing process. A balancing cube has been developed during this project, the development includes modeling, simulations, software design, hardware design and construction. The main goal is to build a mechatronic system, i.e. cube, which is able to jump up without any external force, up to one of its edges. The production was difficult because the parts had to be produced without any errors, all measurements were accurate, since any mistake could lead to the brake system not working and there ц would be not enough inertia to turn the cube. Therefore, one part of the parts was cut bt laser and the other part was printed on a 3D printer.