

MARS PROJECT

Automated Fire Extinguishing Bot

OUR PROJECT: What You Need to Know

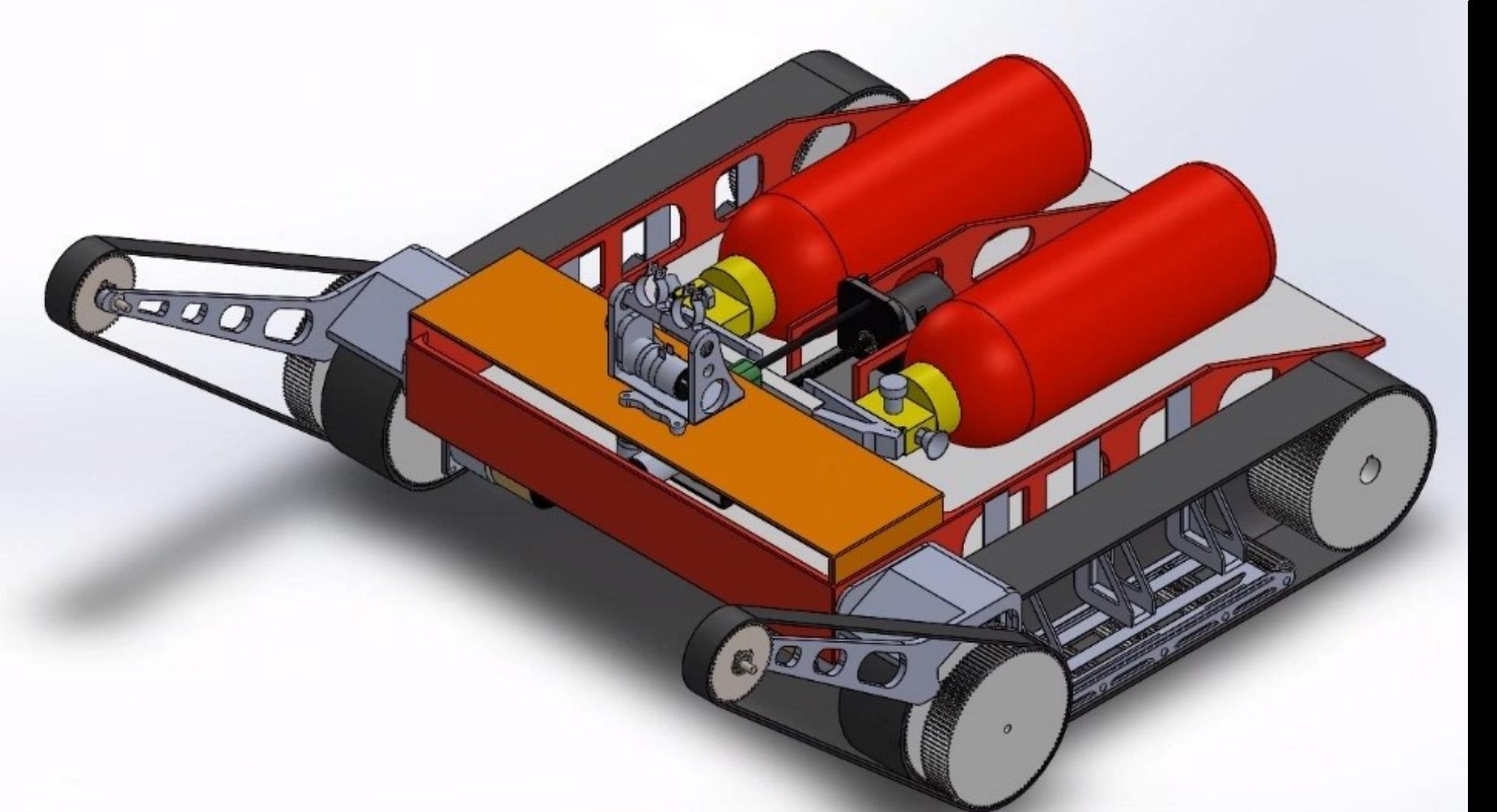
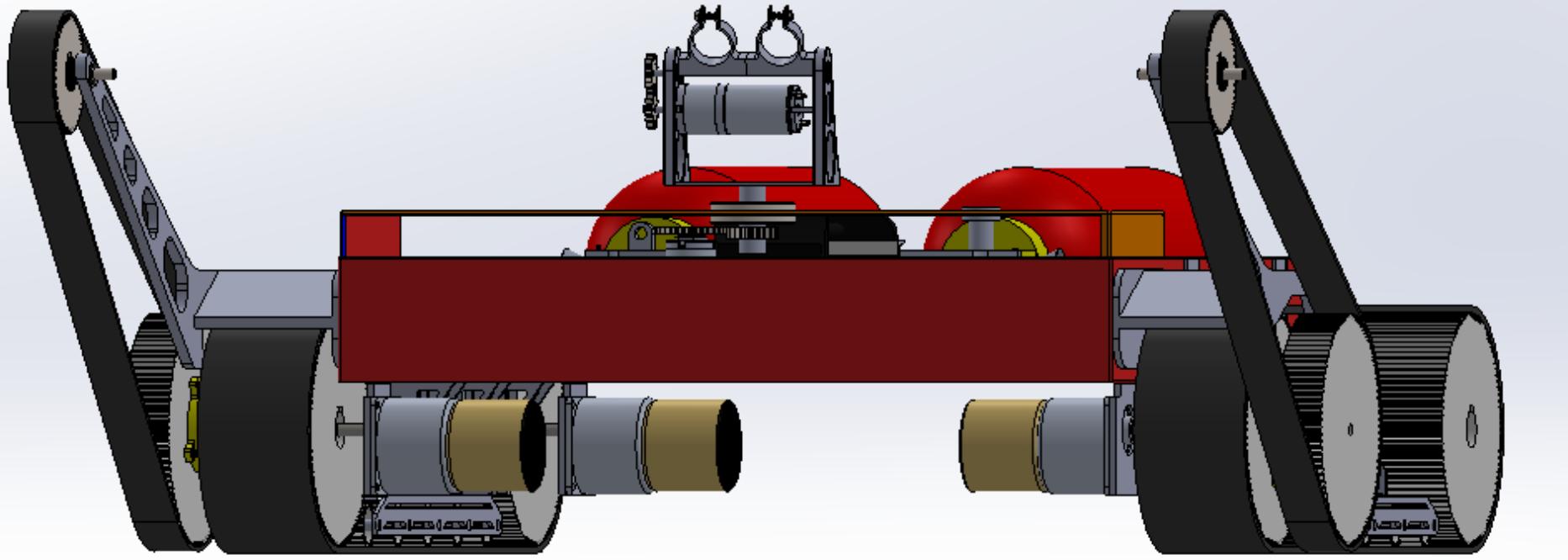
UNDERSTAND HOW THE PROJECT WORKS

MECHANICAL DESIGN

IMAGE PROCESSING

AUTONOMOUS WALL FOLLOWING

AUTONOMOUS FIRE DETECTION

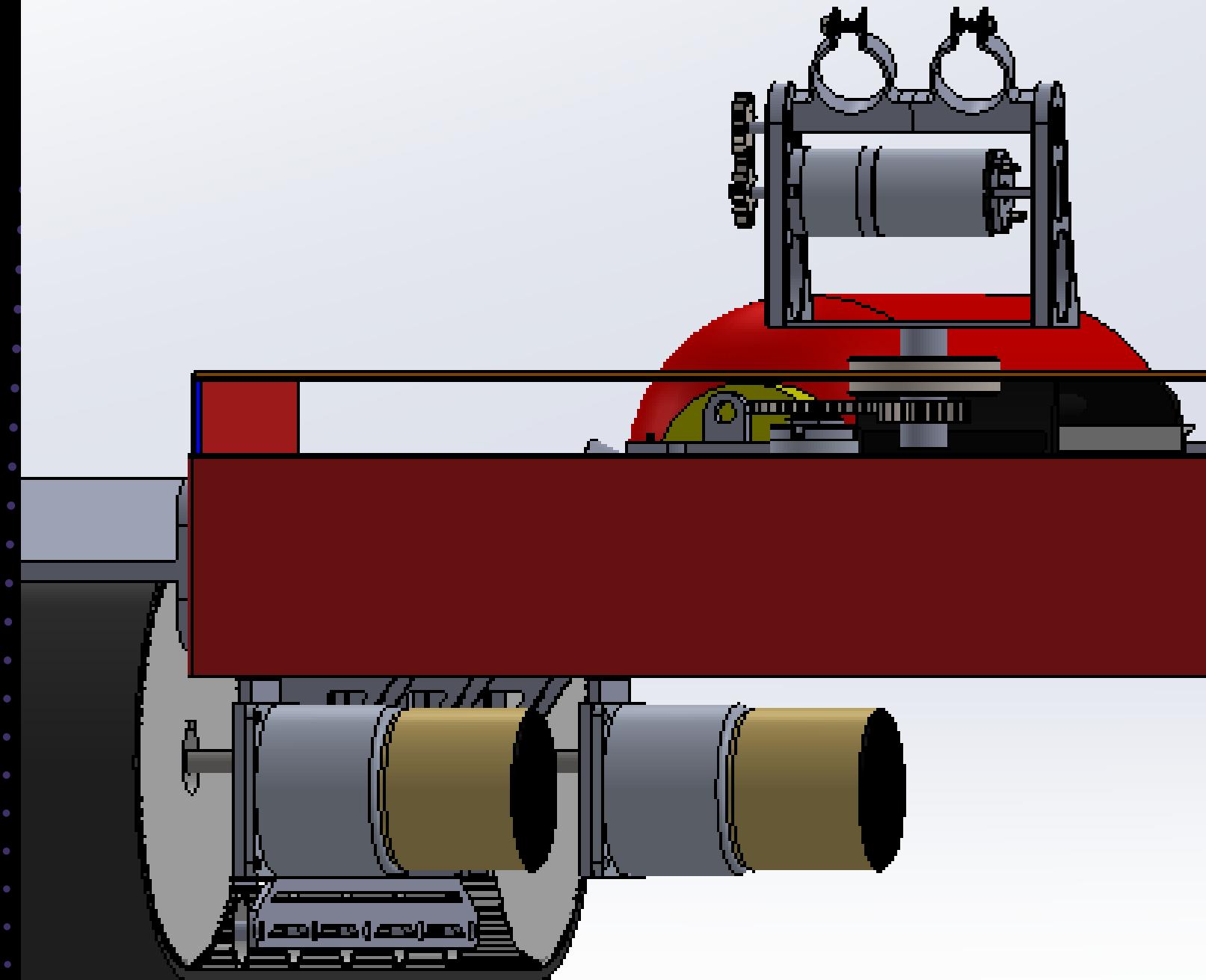


**FINAL
BOT**

MECHANICAL DESIGN

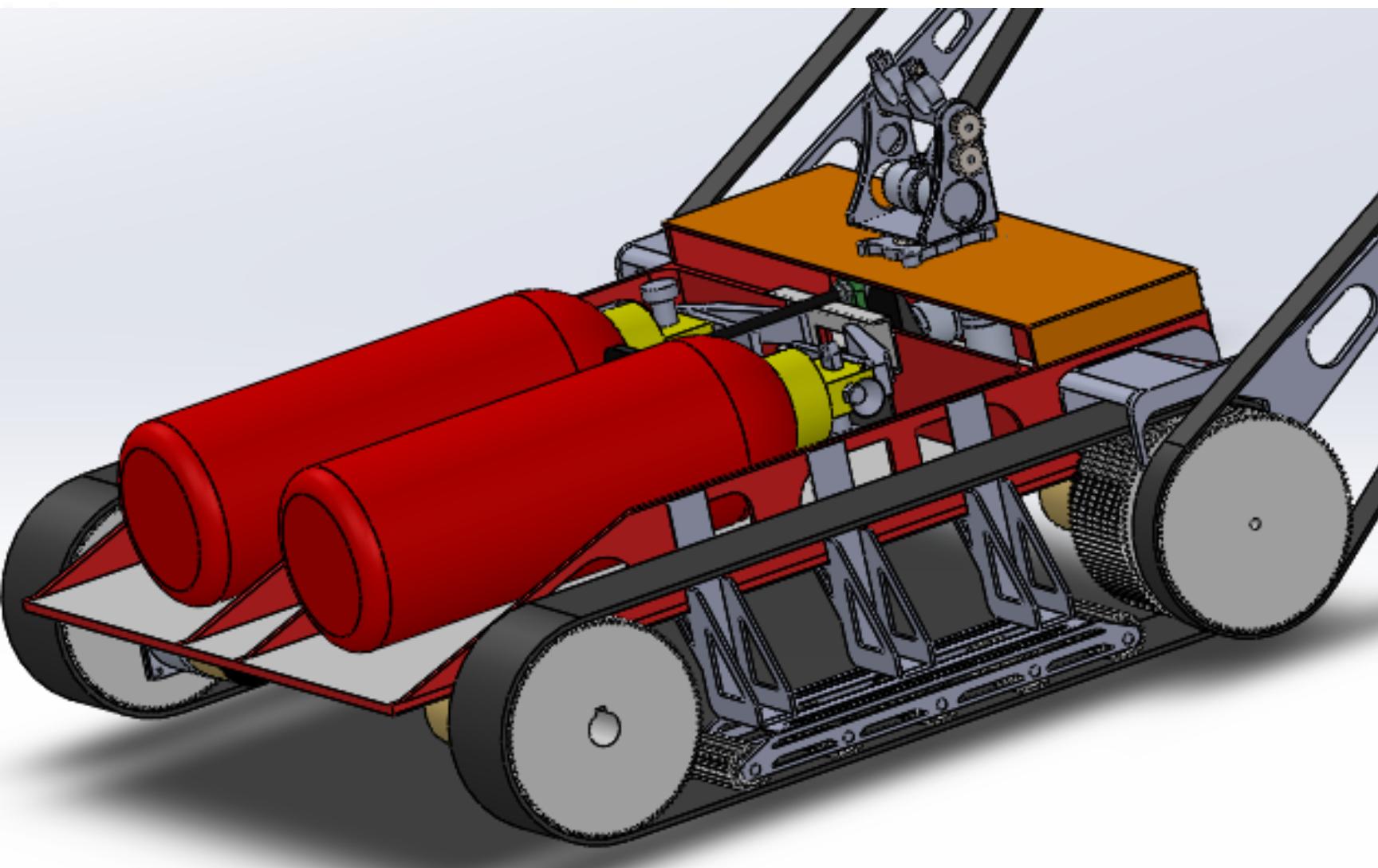
Mechanical Design encompasses 4 major Components -

- Wheel Assembly
- Nozzle Mechanism
- Lead Screw Mechanism
- Stair Climbing Mechanism



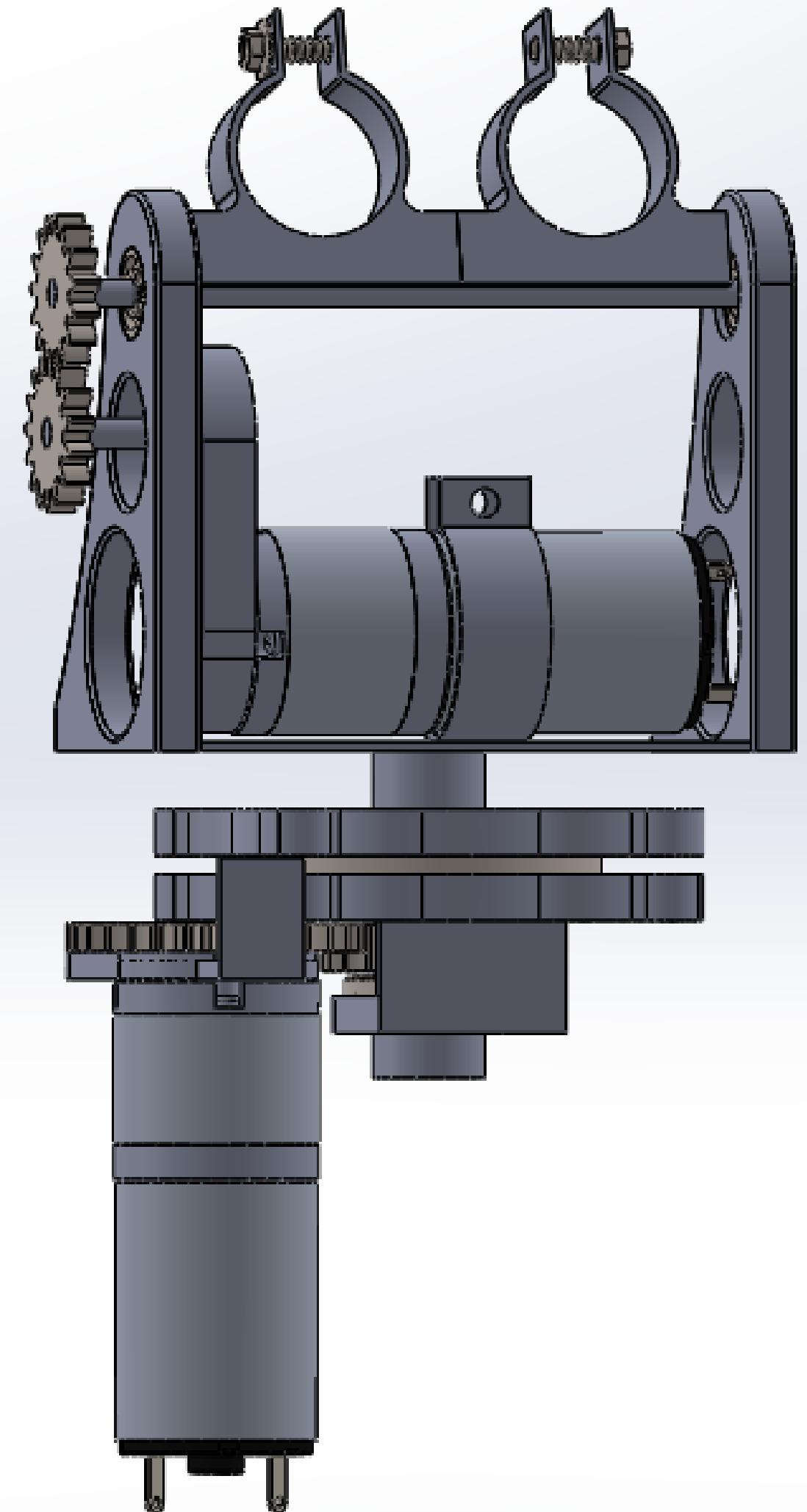
WHEEL ASSEMBLY

The Wheel assembly is a **Timing Belt Mechanism**. This provides an advantage of **easier mobility** and **larger weight carrying capacity** over other mechanisms such as the Rocker Boogie Mechanism.



NOZZLE MECHANISM

Nozzle Mechanism is a 2-axis gimbal. This Mechanism can rotate 30 degrees in both horizontal directions and 30 degrees in both vertical directions.

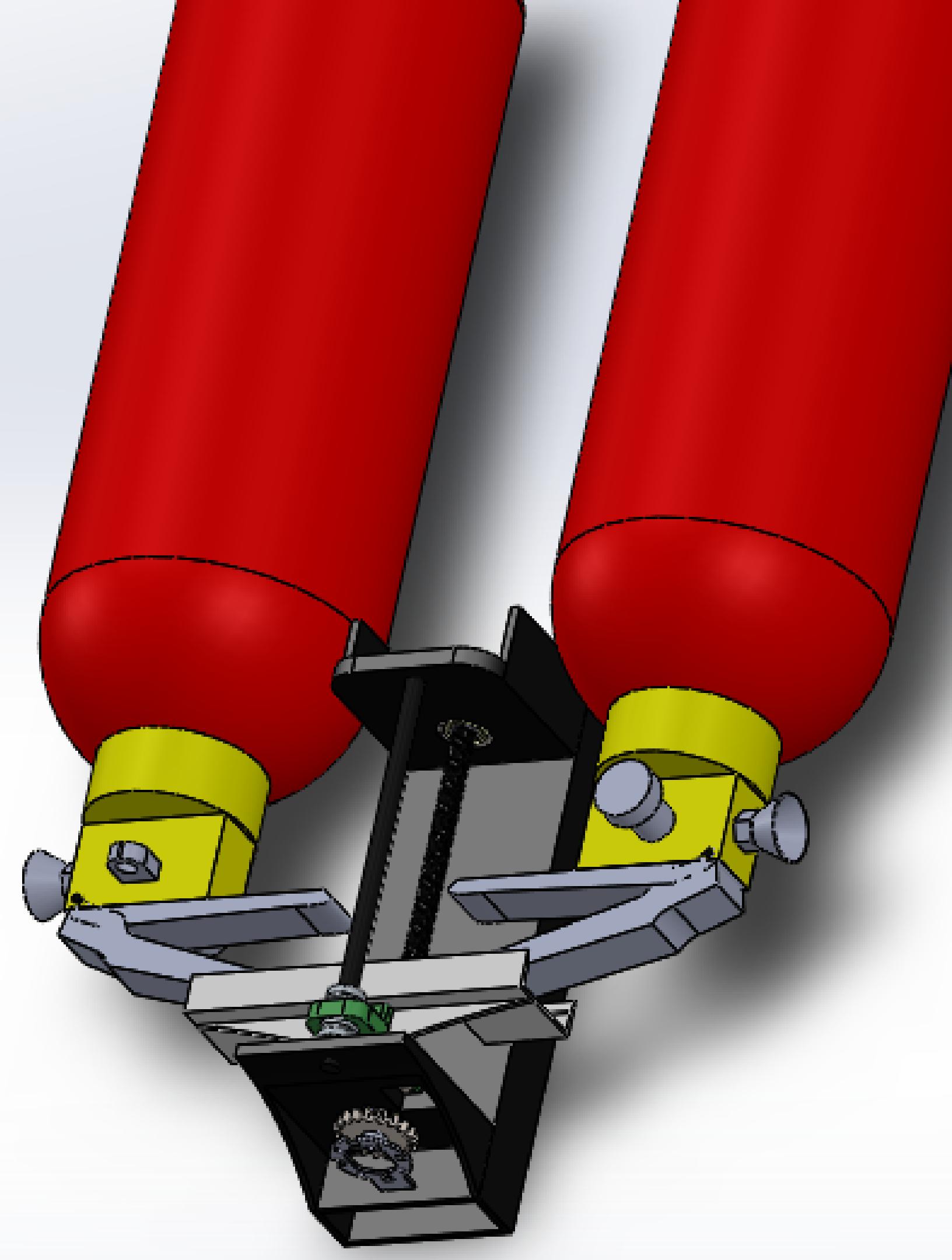


LEAD SCREW MECHANISM

for Handle Pushing

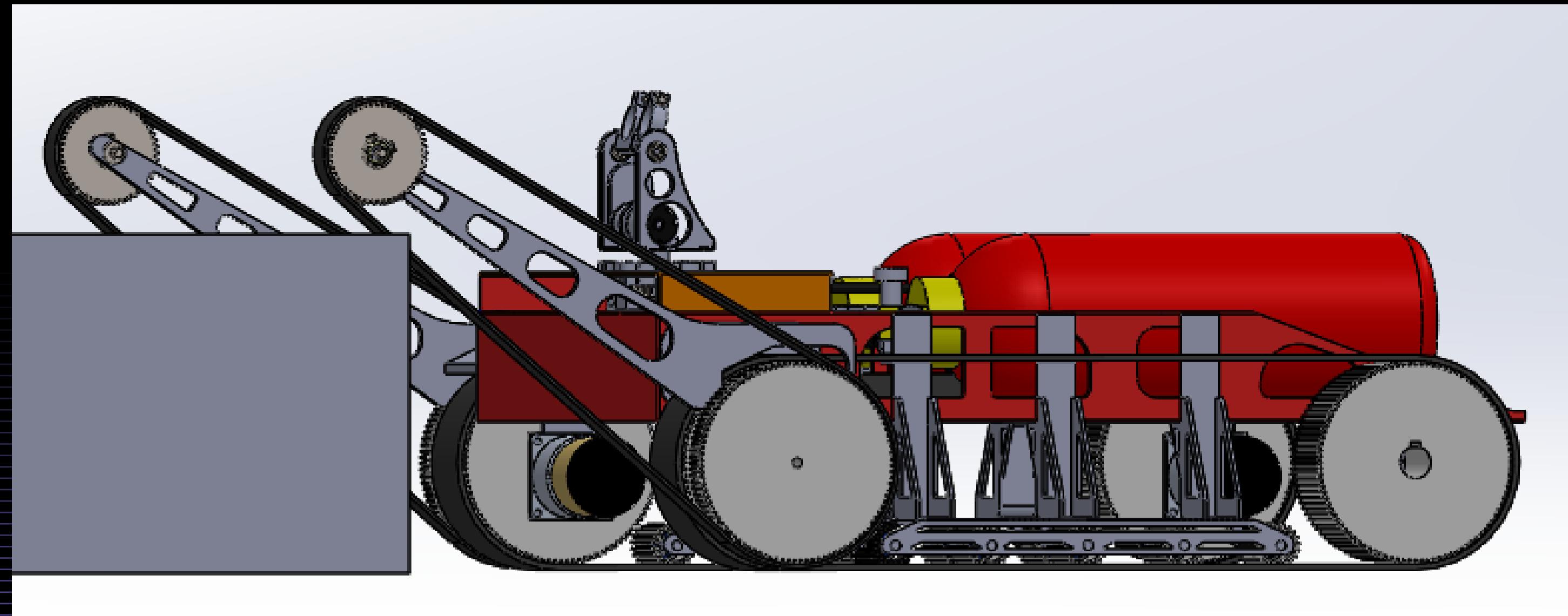
OBJECTIVE BEHIND LEAD SCREW MECHANISM

Leadscrew Mechanism is used because it can provide a torque of large magnitude, and the motion cannot be reversed by the back force applied by the handle.

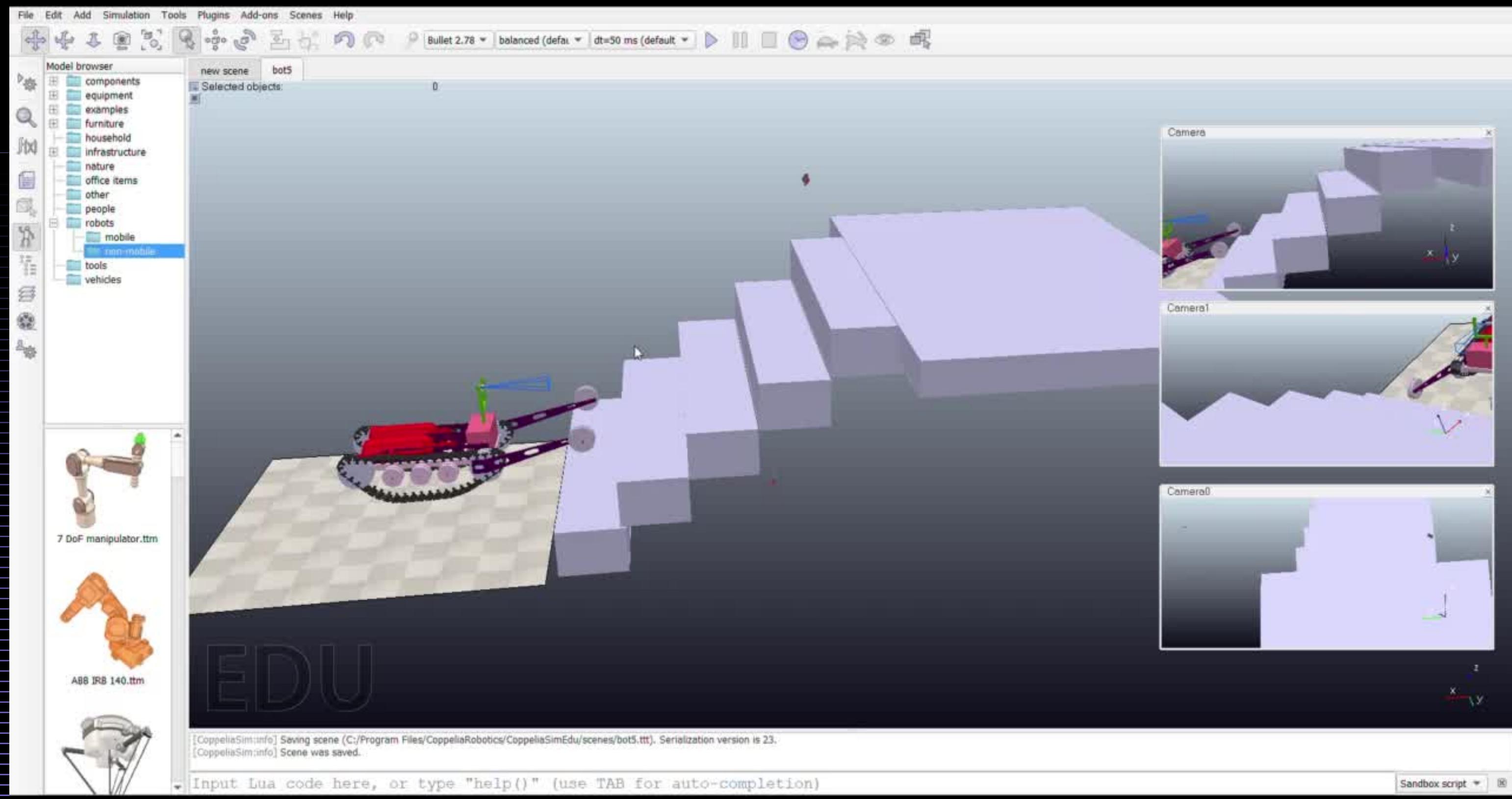


STAIR CLIMBING MECHANISM

The stair climbing mechanism works on the basis of two-timing belts, making an angle of 23 Degrees. This angle was Decided after a brief round of simulation on coppeliasim.

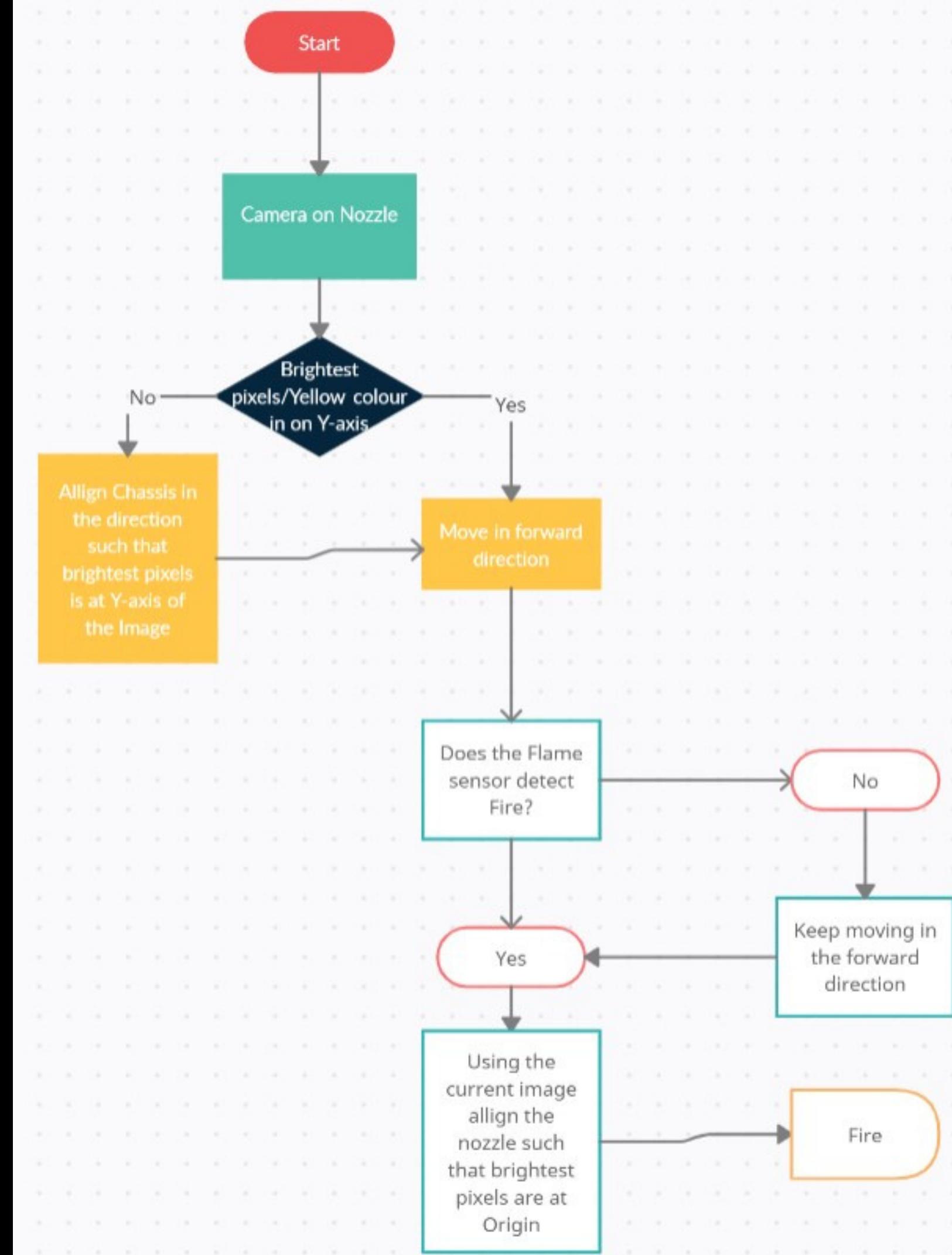


STAIR CLIMBING IN COPPELIASIM

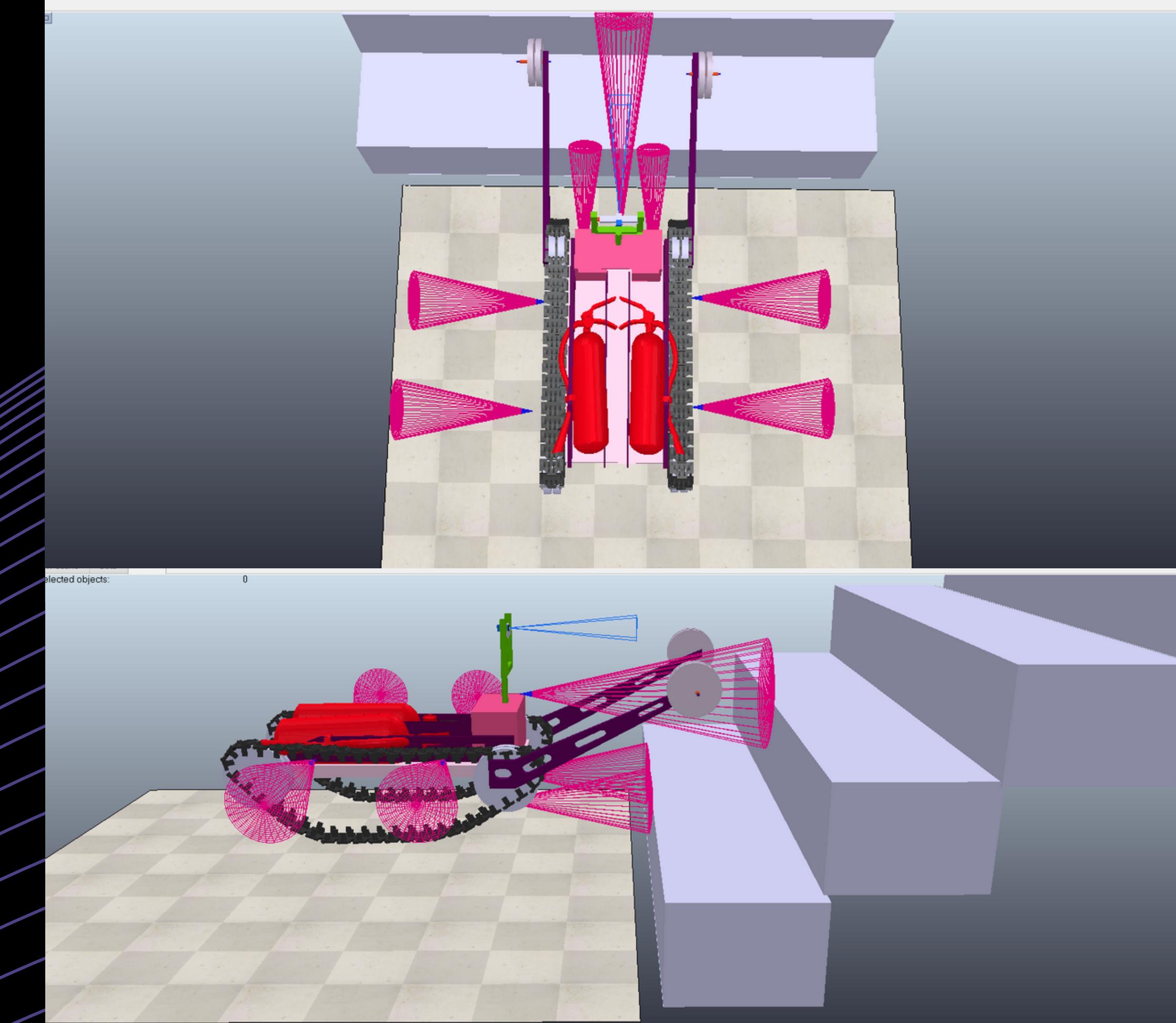


SENSORS

- CAMERA
- FLAME SENSOR
- ULTRASONIC SENSOR
- TEMPERATURE SENSOR



ULTRASONIC DISTANCE SENSOR



TEMPERATURE SENSOR

tinkercad.com/things/fNUXLO1MyTo-temp-sensor-buzzer-and-lcd/editel

Apps InterviewBit Moodle@IITR Solve C++ | Hacker... Instagram Tinkercad INDIAN INSTITUTE... Mechanical | IISc Arduino - Home API ref Webots Reading list

TIN KER CAD TEMP SENSOR-BUZZER AND LCD All changes saved

Code Start Simulation Export Share

Text

```
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
4 float a;
5 void setup() {
6   lcd.begin(16, 2);
7   Serial.begin(9600);
8   pinMode(6,OUTPUT);
9 }
10
11 void loop() {
12   a=analogRead(1);
13   a=a*0.0048828125;
14   a=(a-0.5)*100;
15   lcd.clear();
16   lcd.setCursor(0, 0);
17
18   Serial.println(a);
19   lcd.print(a);
20   lcd.print("C");
21   if(a>75)
22   {
23     digitalWrite(6,HIGH);
24   }
25   else
26     digitalWrite(6,LOW);
}
```

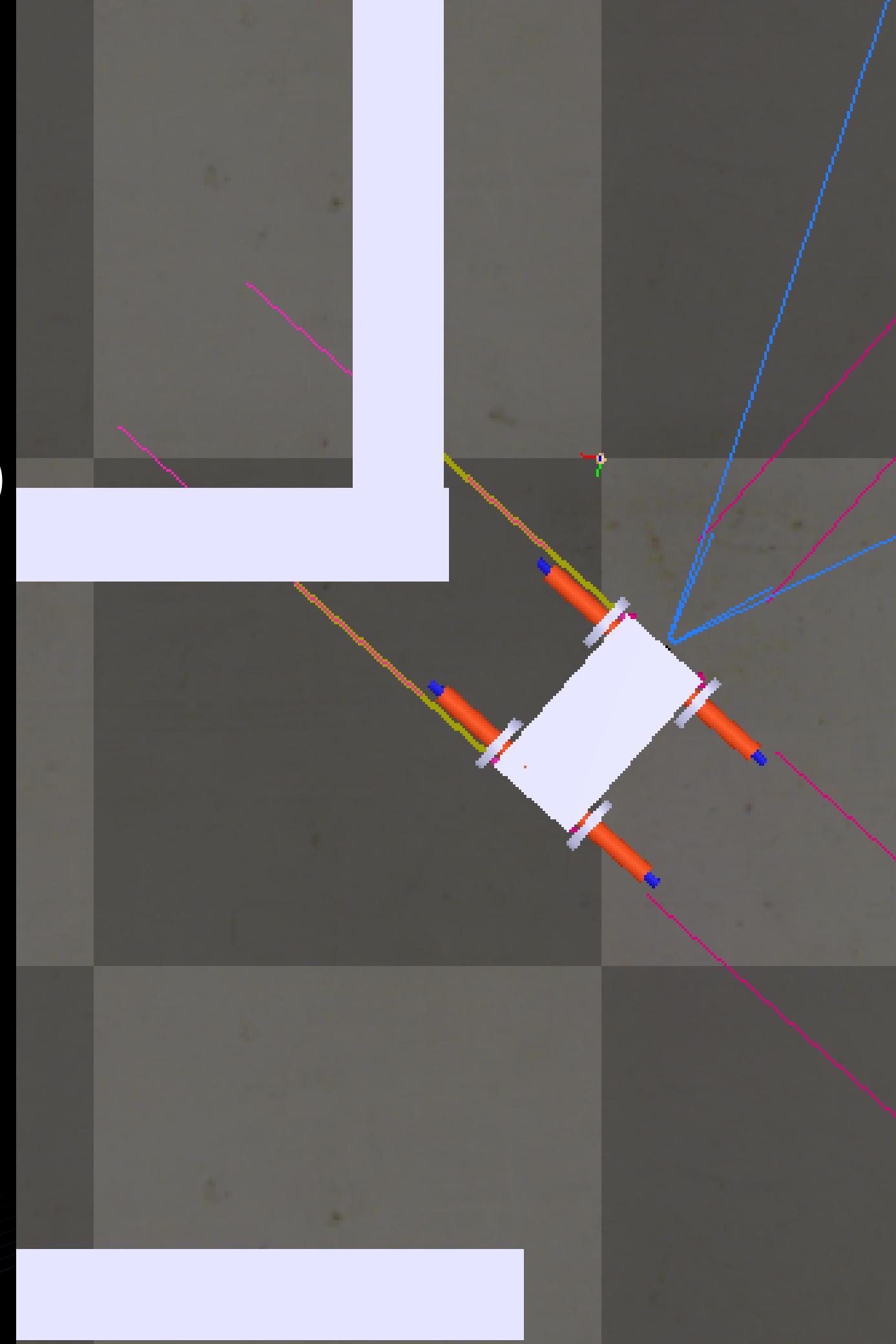
Serial Monitor

26
25
24

Send Clear

The circuit diagram shows an Arduino Uno connected to a breadboard. On the breadboard, there is a green LCD module, a blue temperature sensor (analog input), and a black buzzer. The Arduino's digital pins 11, 12, 13, 14, 15, and 16 are connected to the breadboard. Pin 11 is connected to the LCD's VSS pin. Pins 12 and 13 are connected to the LCD's RS and R/W pins respectively. Pin 14 is connected to the LCD's E pin. Pin 15 is connected to the LCD's D4-D7 pins. Pin 16 is connected to the LCD's VDD pin. The temperature sensor is connected to pins A0 and A1. Pin A0 is connected to the Arduino's A0 pin, and pin A1 is connected to the Arduino's A1 pin. The Arduino's ground pin is connected to the breadboard's common ground. The breadboard's common ground is also connected to the temperature sensor's GND pin and the buzzer's ground pin. The breadboard's common ground is also connected to the Arduino's 5V pin. The Arduino's 5V pin is connected to the breadboard's common 5V rail. The breadboard's common 5V rail is connected to the temperature sensor's VCC pin and the buzzer's VCC pin. The breadboard's common 5V rail is also connected to the Arduino's GND pin.

AUTONOMOUS INDOOR NAVIGATION



BOT MOTION

- Left wall following
- Motion using PID
- Move towards the centre of the fire.
- Stop at a distance followed by motion of nozzle

THE WALL FOLLOWING

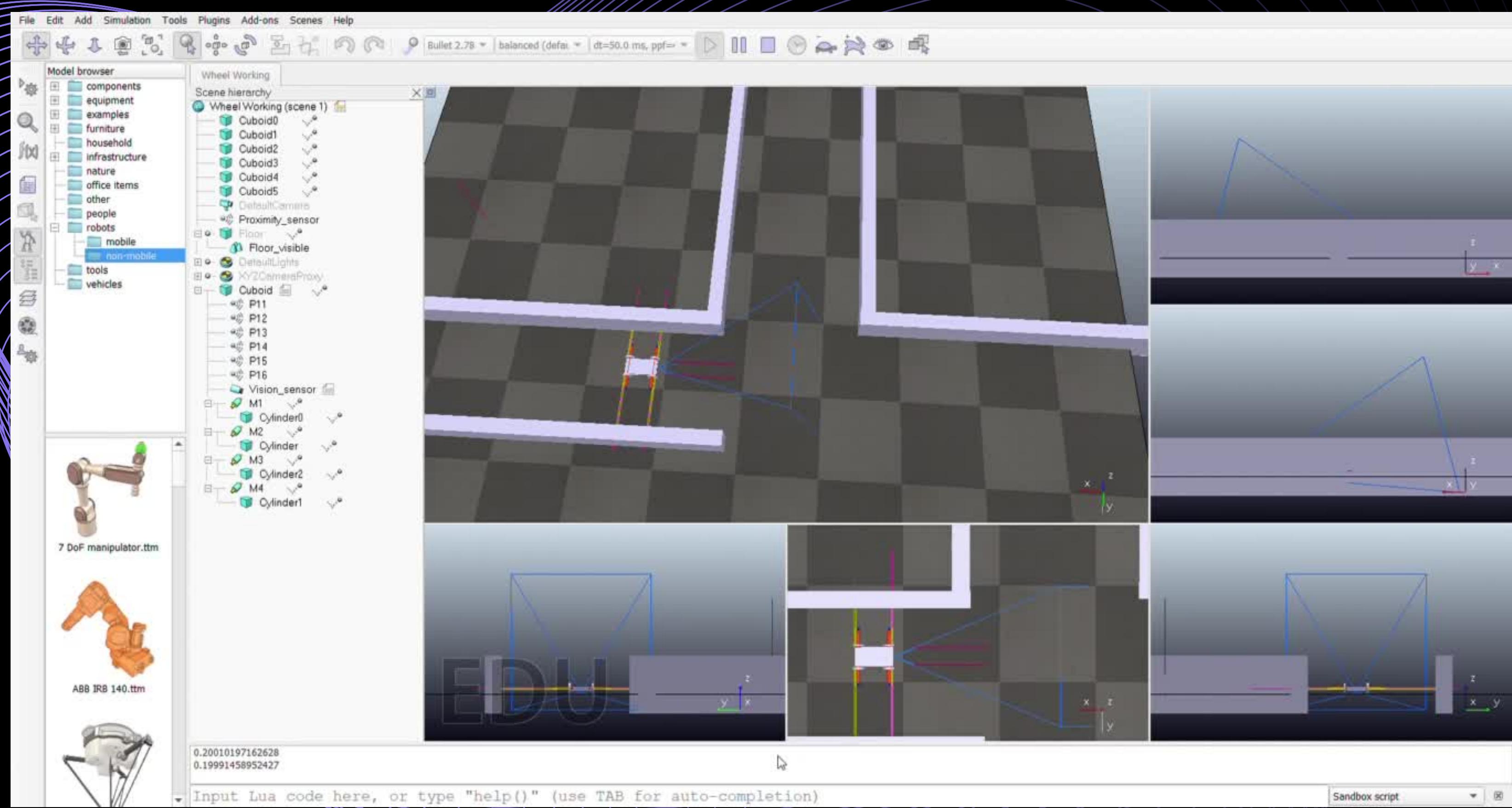
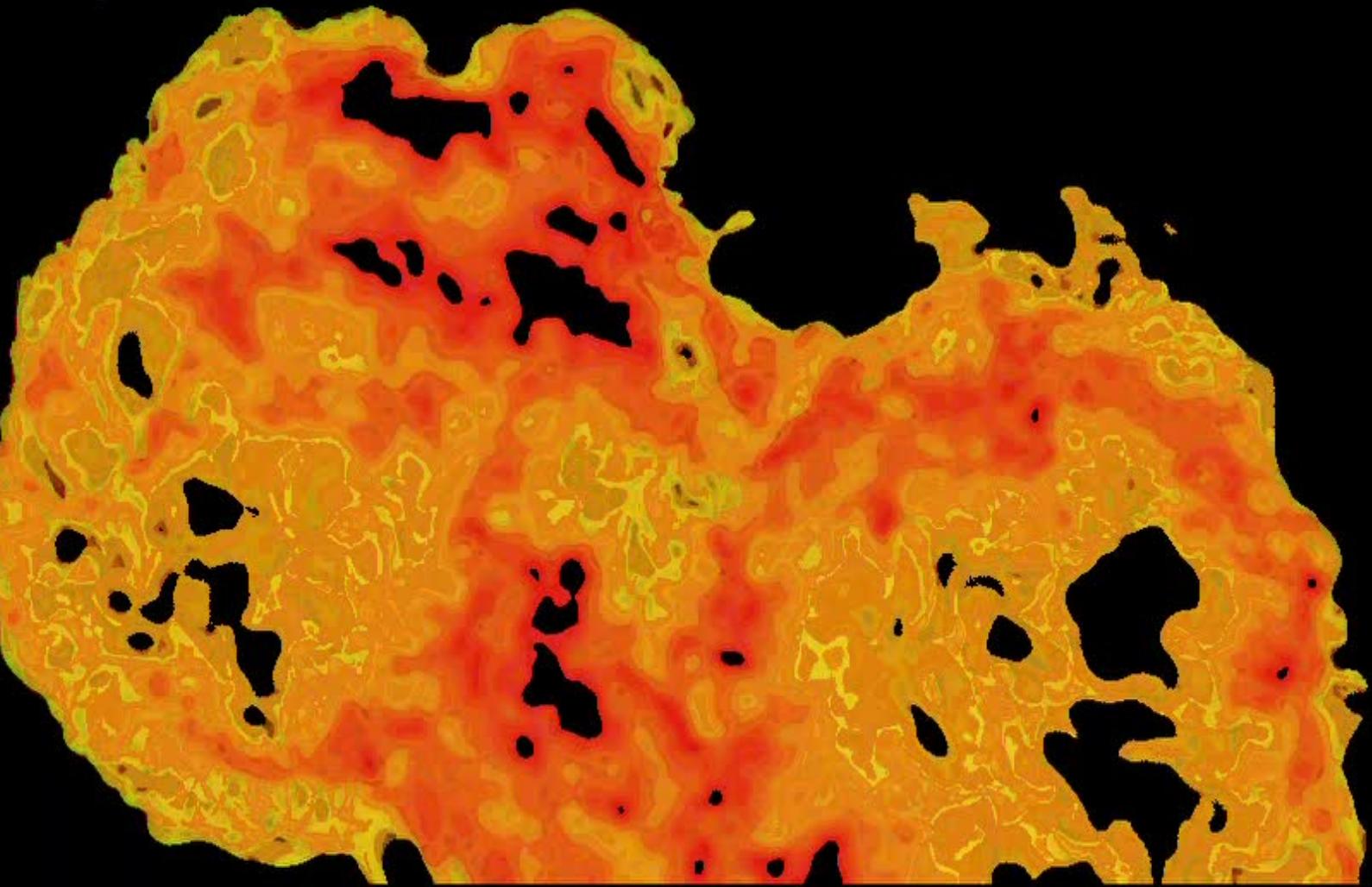


Image Processing And Fire Detection



FIRE DETECTION

For Fire Detection, we use [Image Processing](#) to find the approximate coordinates of the center of the fire.

This is done in [OpenCV](#)

HSV VALUES

In CoppeliaSim, the HSV values of fire lie between [18,50,50] and [60,255,255]

GRAY SCALE IMAGE

This calculates the light intensity in order to detect Fire. The Gray Scale Image values lie in between [190 to 220]

COMPLETE PROCESS OF FIRE DETECTION

Getting Data from Coppeliasim

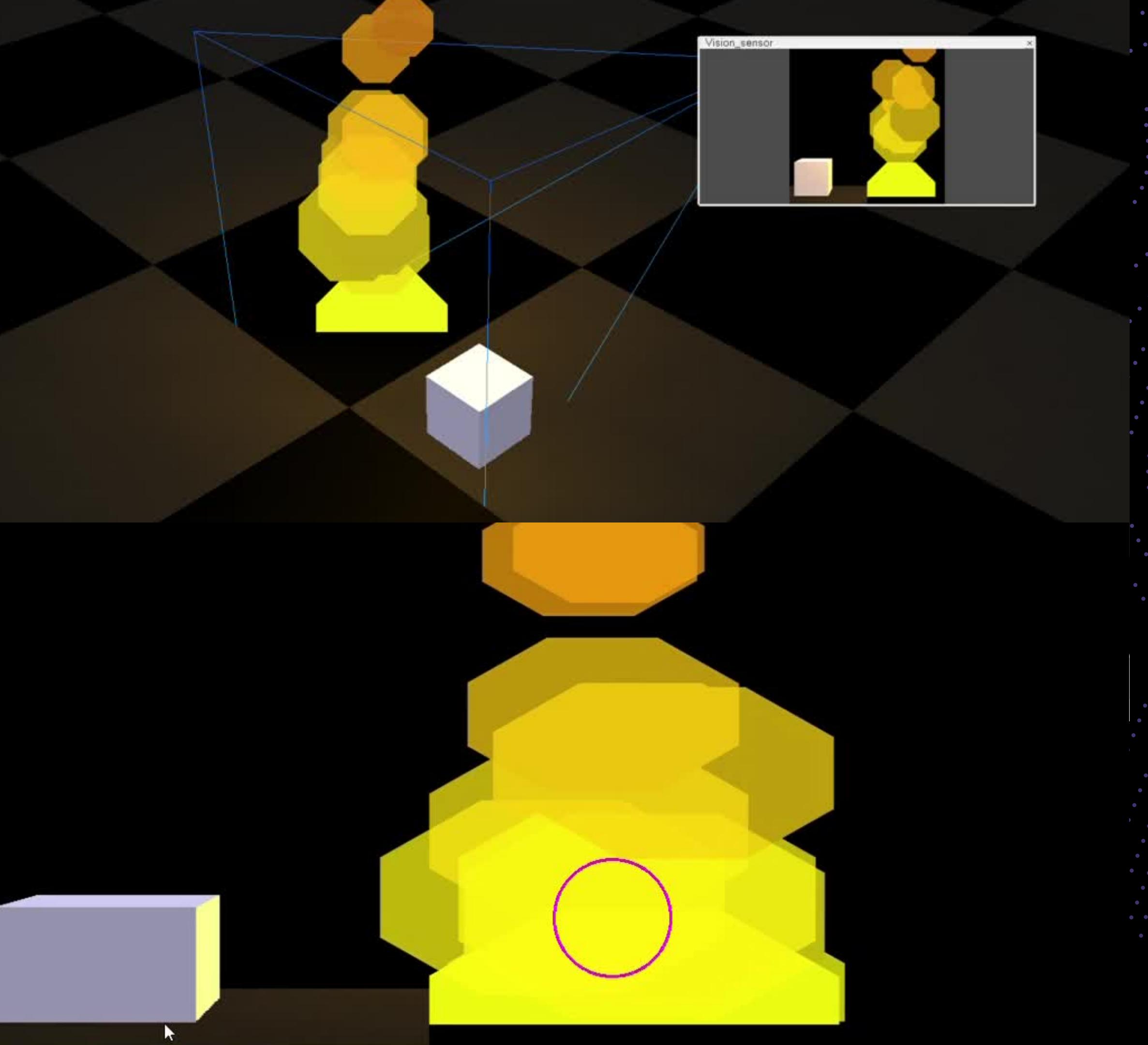
The input for OpenCV is taken from Vision Sensor in Coppeliasim.

Processing Information in openCV

The constantly updated Image input is taken in OpenCV and fire is identified on the basis of HSV values. Then the image is converted into GrayScale to identify the highest intensity spots. We identify a circle around the fire and the coordinates of the centre of this circle are stored in a text file.

Exporting Information to coppeliasim

In Coppeliasim, the coordinates of the centre of fire in the text file help bot to move to the exact location of fire.



Simulation

SHORT COMMINGS

- Takes time to detect fire
- Needs manual assistance while changing the fire extinguishers
- Usable for household fires only

IDEAS FOR FUTURE EXPANSION

- Install cameras in every room
- Link with smoke sensors in the area
- Use custom made cylinders to avoid manual assistance

M
E
M
B
E
R
S

- Akanksha Vijayvergiya
- Eeshan Punde
- Ganesh Kalanidhi
- Mritunjay Kumar Choubey
- Poreddy Sahith Reddy
- Srushti Parbat

M E N T O R S

- Naman Agarwalla
- Parul Chaudhary
- Shubham Goyal