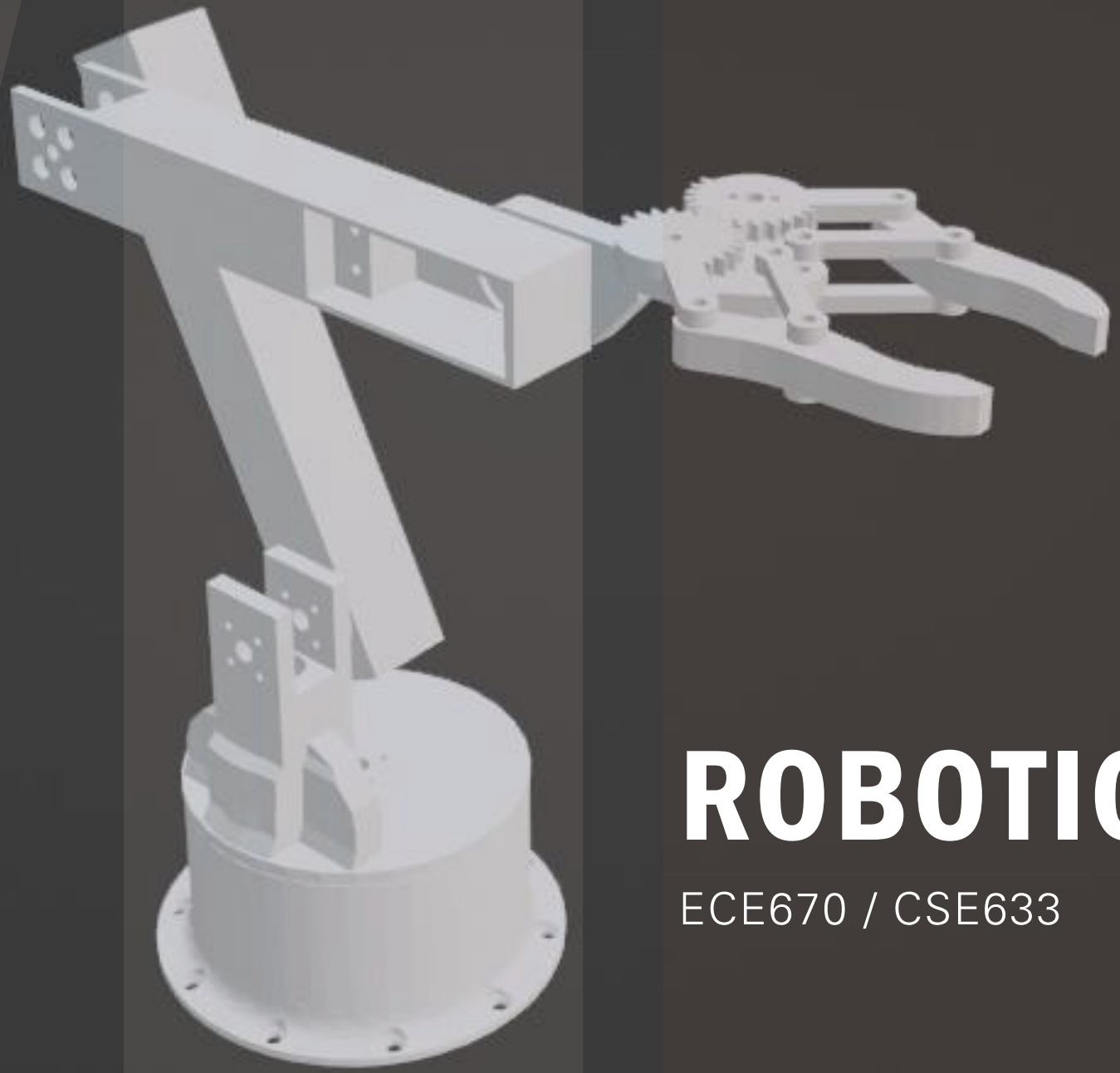
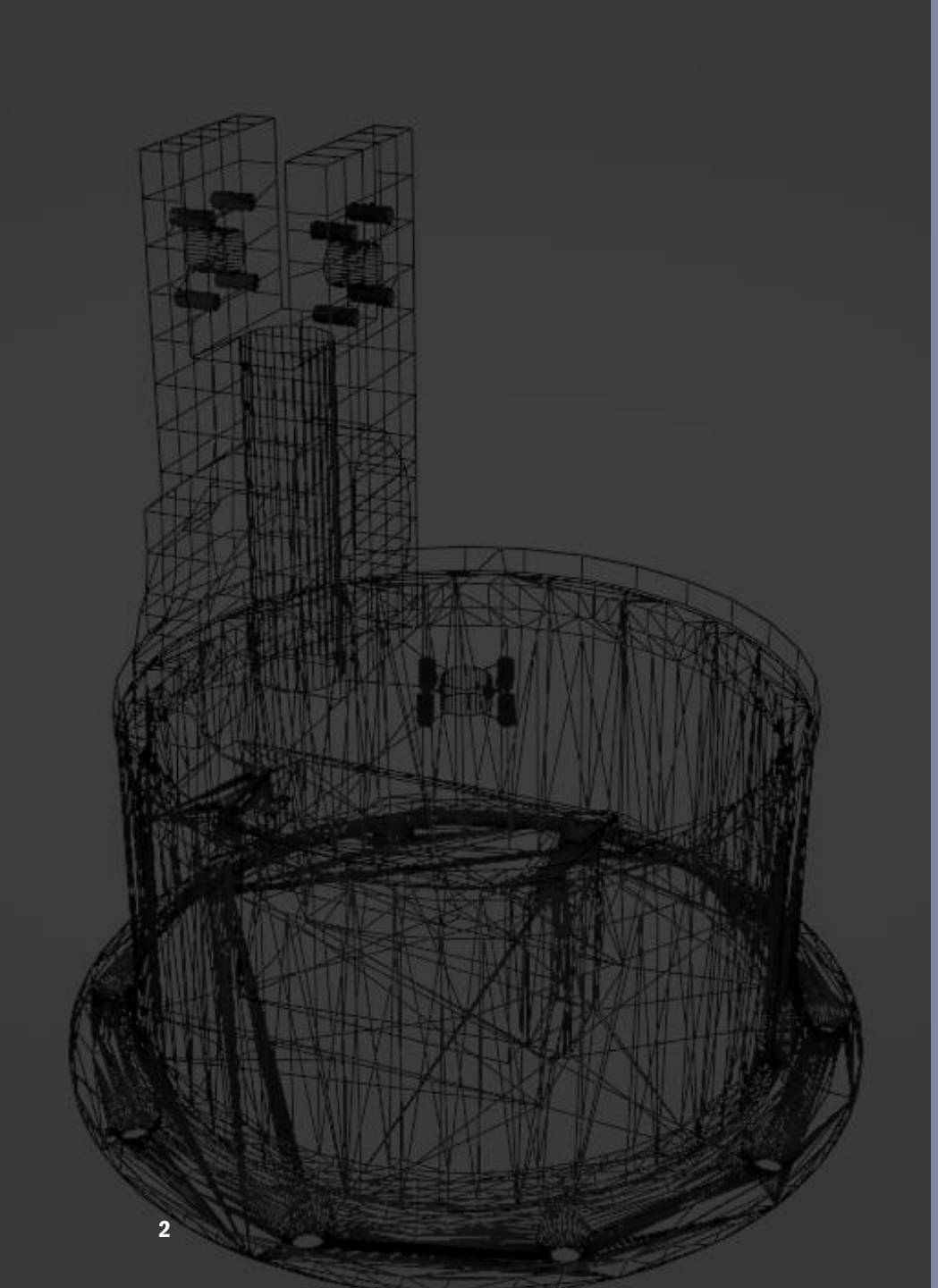


ROBOTICS

ECE670 / CSE633



AGENDA

Introduction

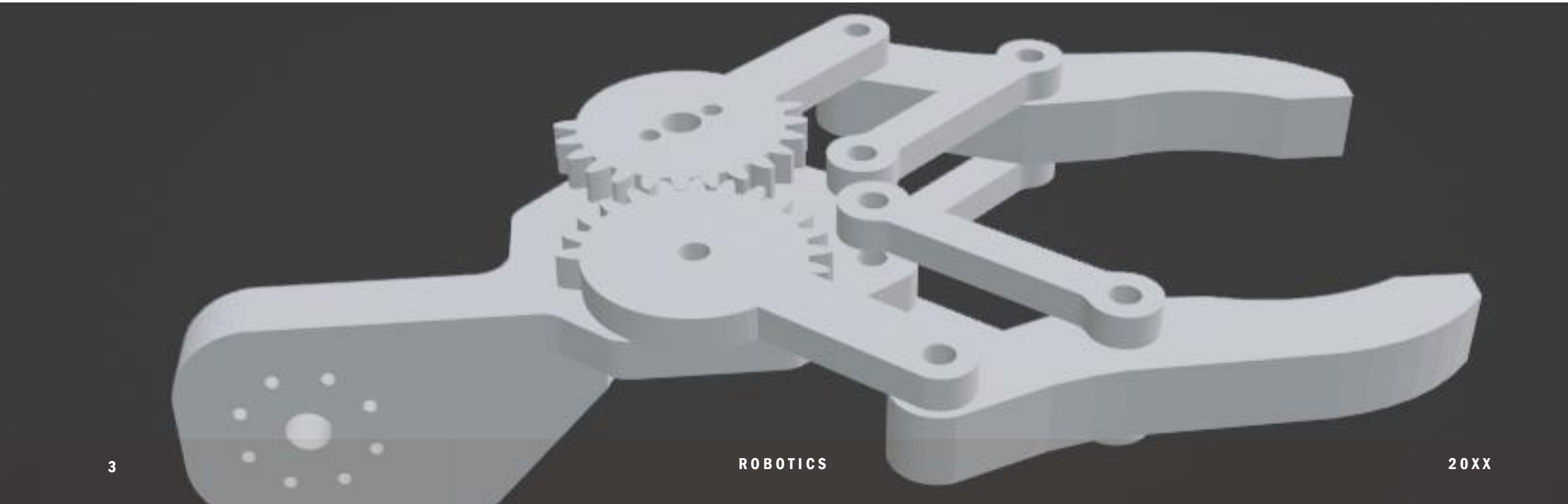
Primary goals

Plan of implementation

Summary

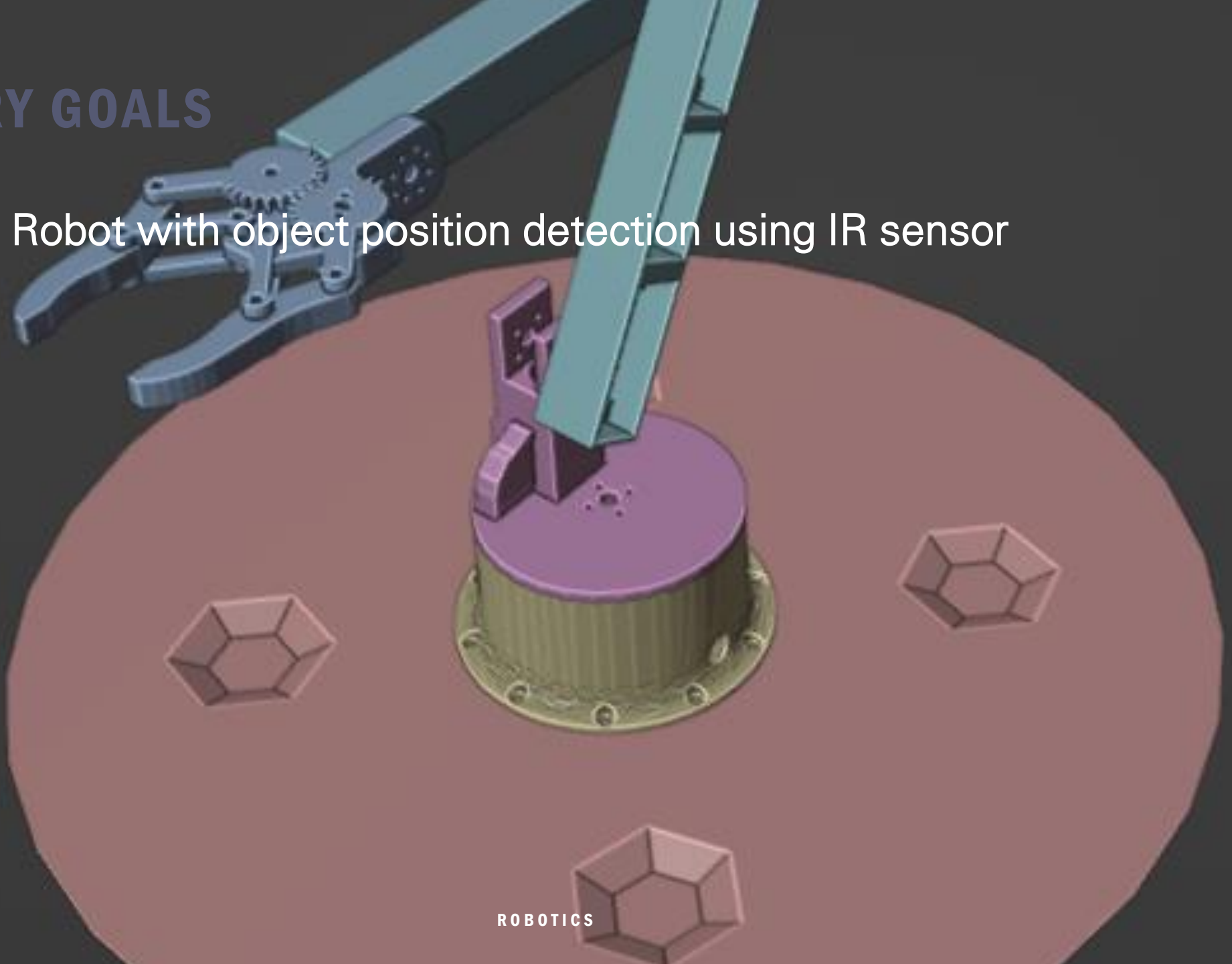
INTRODUCTION

A 3 degree arm robot with a mechanized arm designed to hold an object from a detected position to a new position



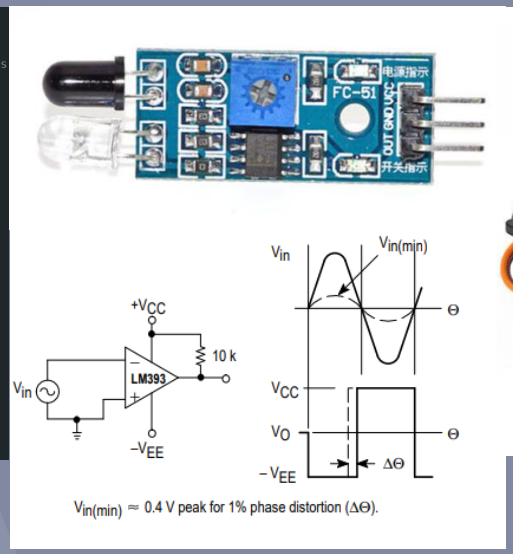
PRIMARY GOALS

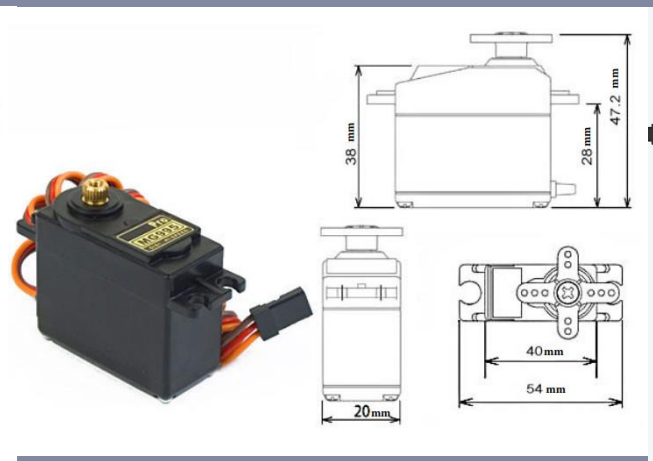
A working Robot with object position detection using IR sensor

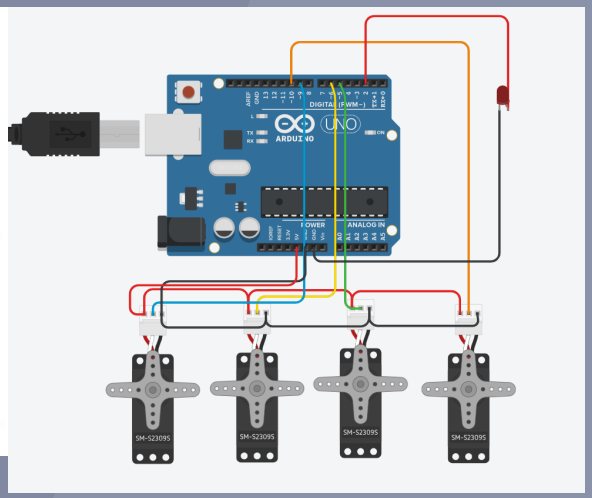


SENSOR READING MECHANISM, CONTROL ALGORITHM/LOGIC AND ACTUATOR (MOTOR) INPUT DETAILS.

```cpp
#include <Servo.h>
// Define the pin for the IR sensor

const int irSensorPin = 2;  // Change this to the pin your sensor is
Servo servo_3;

Servo servo_5;

Servo servo_base;

Servo servo_hook;
void setup() {
  Serial.begin(9600);  // Initialize serial communication
  pinMode(irSensorPin, INPUT);  // Set the IR sensor pin as input
  servo_3.attach(10, 500, 2500);
  servo_5.attach(5, 500, 2500);
  servo_base.attach(6, 500, 2500);
  servo_hook.attach(9, 500, 2500);

  servo_3.write(90);
  servo_5.write(90);
  servo_base.write(0);
  servo_hook.write(90);
}

void loop() {
```







CONTROL ALGORITHM/LOGIC

OBJECT_DETECTION.ino

```
#include <Servo.h>
// Define the pin for the IR sensor

const int irSensorPin = 2;  // Change this to the pin your sensor is
Servo servo_3;

Servo servo_5;

Servo servo_base;

Servo servo_hook;
void setup() {
  Serial.begin(9600);  // Initialize serial communication
  pinMode(irSensorPin, INPUT);  // Set the IR sensor pin as input
  servo_3.attach(10, 500, 2500);
  servo_5.attach(5, 500, 2500);
  servo_base.attach(6, 500, 2500);
  servo_hook.attach(9, 500, 2500);

  servo_3.write(90);
  servo_5.write(90);
  servo_base.write(0);
  servo_hook.write(90);
}

void loop() {
  // Read the digital signal from the IR sensor
  int objectDetected = digitalRead(irSensorPin);
```

The provided Arduino code sets up a system for an object sorting robot using an IR sensor and four servo motors. In the `setup` function, it initializes serial communication, configures the IR sensor pin, and attaches and positions the servo motors. The main logic is implemented in the `loop` function, where the code reads the digital signal from the IR sensor. If an object is detected, the system executes a predefined sorting sequence, involving movements of the base, two arm servos servo_3 and servo_5, and a servo hook. The sequence is followed by status messages printed to the serial monitor. The loop then introduces a delay of 1000 milliseconds to prevent rapid serial prints. If no object is detected, it prints a corresponding message. The code serves as a foundation for a basic object sorting robot, responding to the presence or absence of objects sensed by the IR sensor.

OBSTACLE AVOIDANCE SENSOR REFLECTION PHOTOELECTRIC MODULE INFRARED SENSOR

The module features a 3 wire interface with Vcc, GND and an OUTPUT pin on its tail. It works fine with 3.3 to 5V levels. Upon hindrance/reflectance, the output pin gives out a digital signal (a low-level signal)

Detection distance: 2 ~ 30cm

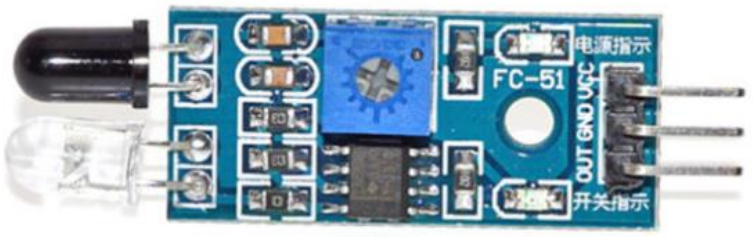
Detection angle: 35 °

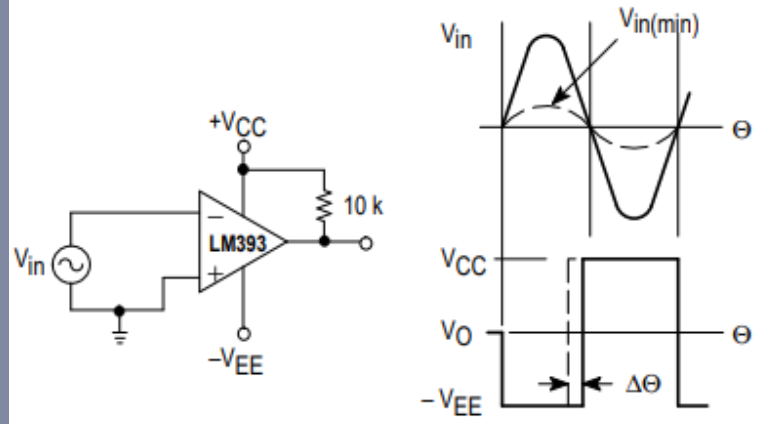
Comparator chip: LM393

3mm screw holes for easy mounting

effective distance range of 2cm to 80cm

A preset knob to fine-tune distance range

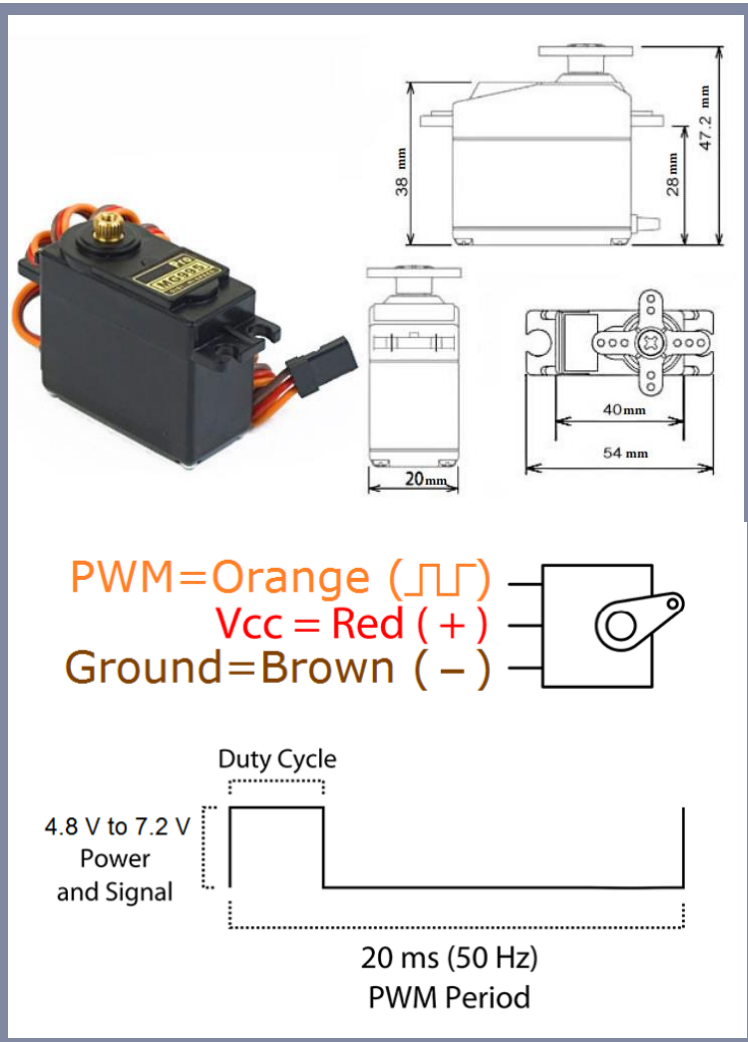




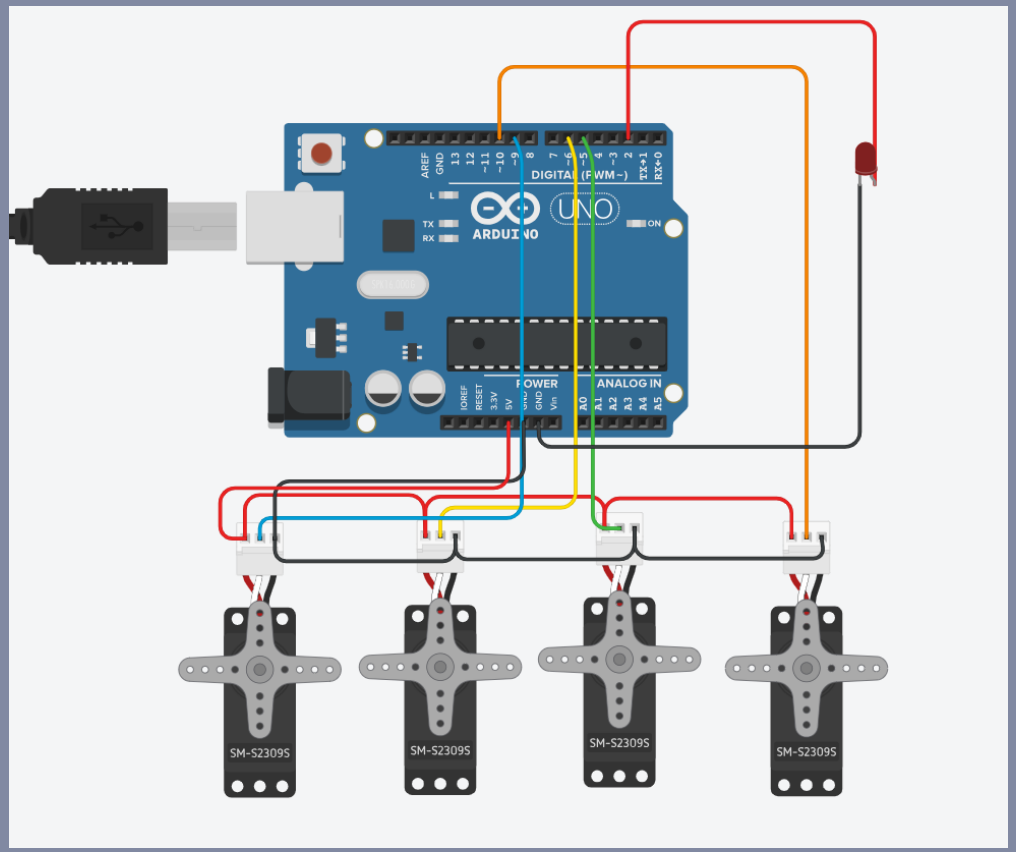
$V_{in(min)} \approx 0.4$ V peak for 1% phase distortion ($\Delta\Theta$).

MG995 HIGH SPEED SERVO ACTUATOR

This high-speed servo actuator is not code dependent
Weight: 55 g
Dimension: 40.7 x 19.7 x 42.9 mm approx.
Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
Operating voltage: 4.8 V to 7.2 V
Dead band width: 5 µs
Stable and shock proof double ball bearing design
Temperature range: 0 °C – 55 °C



CIRCUIT DIAGRAM



PLAN OF IMPLEMENTATION

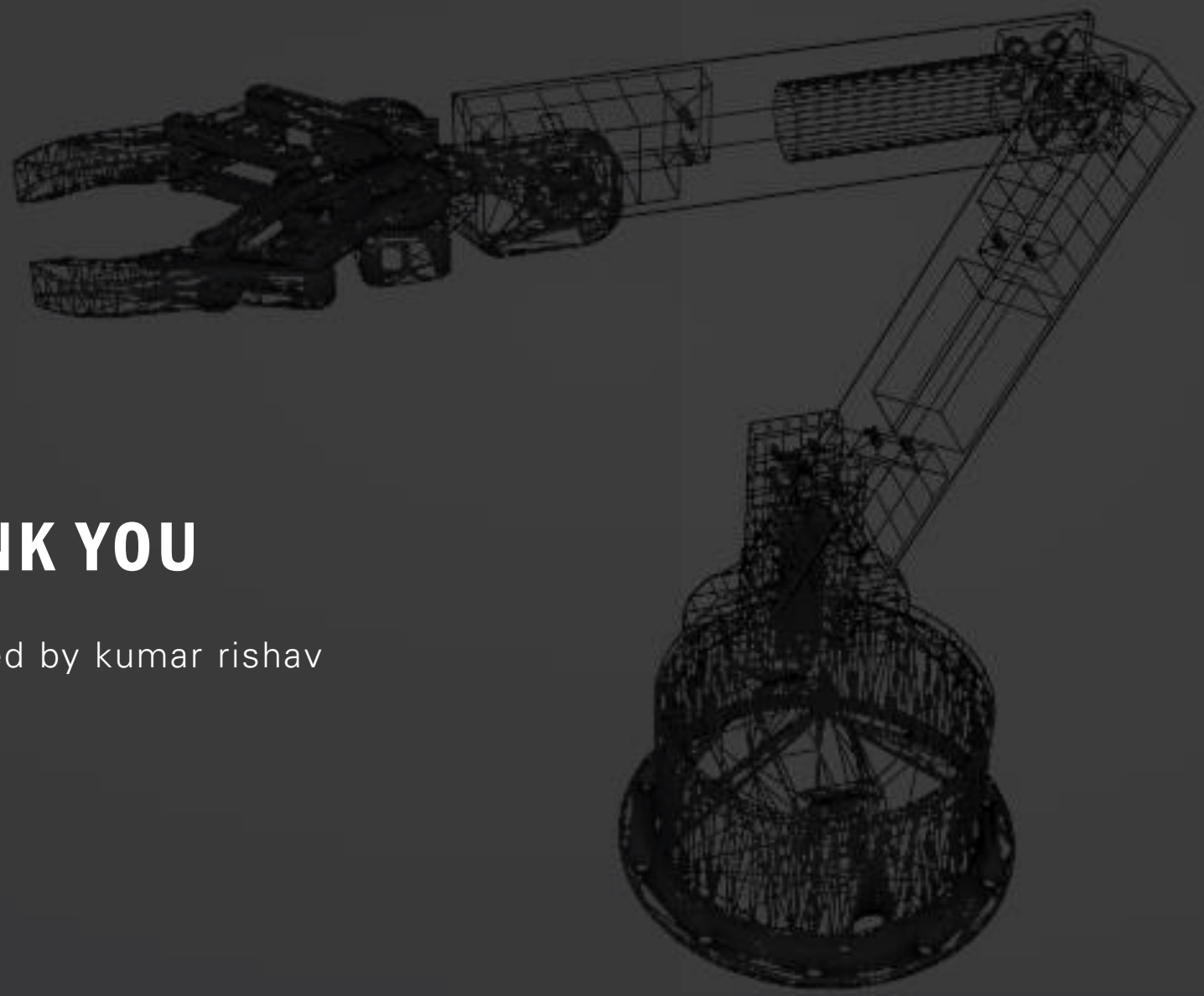
1 PLANNING

2 DESIGNING

3 PRINTING

4 ASSEMBLY

5 PROGRAMMING



THANK YOU

Presented by kumar rishav