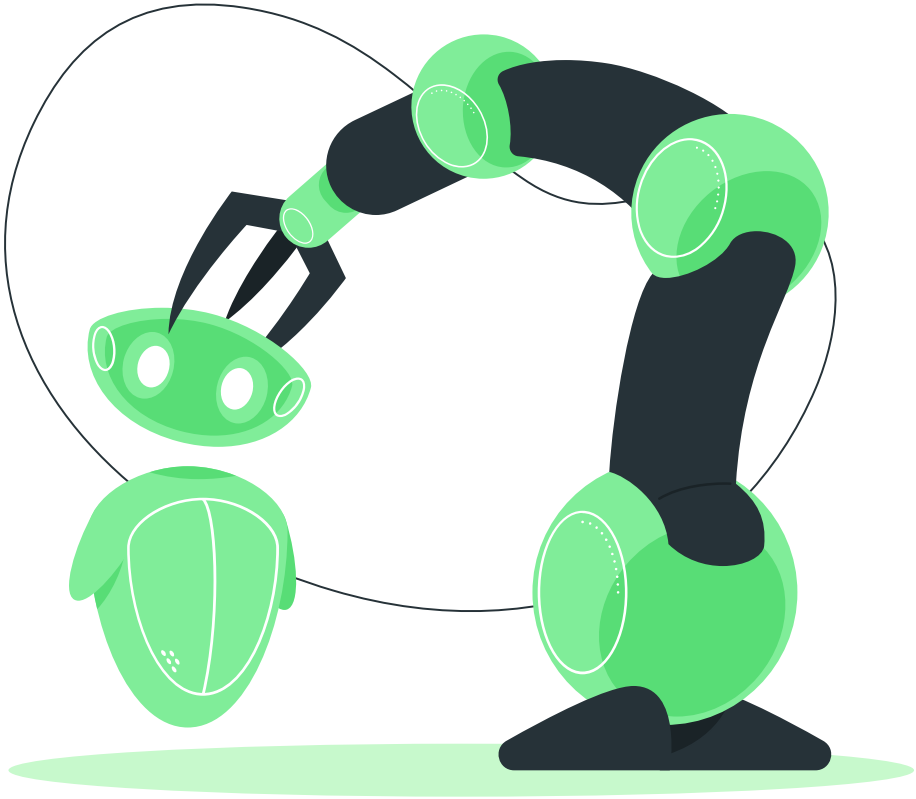
IEEE RAS

Robotic presentation



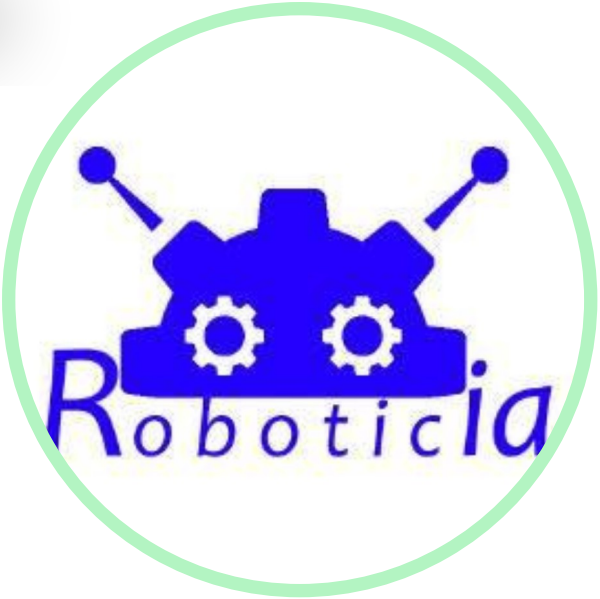
Introduction

Team Name:

Roboticia

Team members:

- Adham Amr
- Mohamed Essam
- Abdurahman Diaa
- Mohamed Kamel



Contents of Presentation

1

Design

- Function of each part.
- Assembly picture.
- Problems & Solutions.

2

Code

- Algorithm.
- Problems & Solutions.
- Get out of maze.
- Used Controller.

3

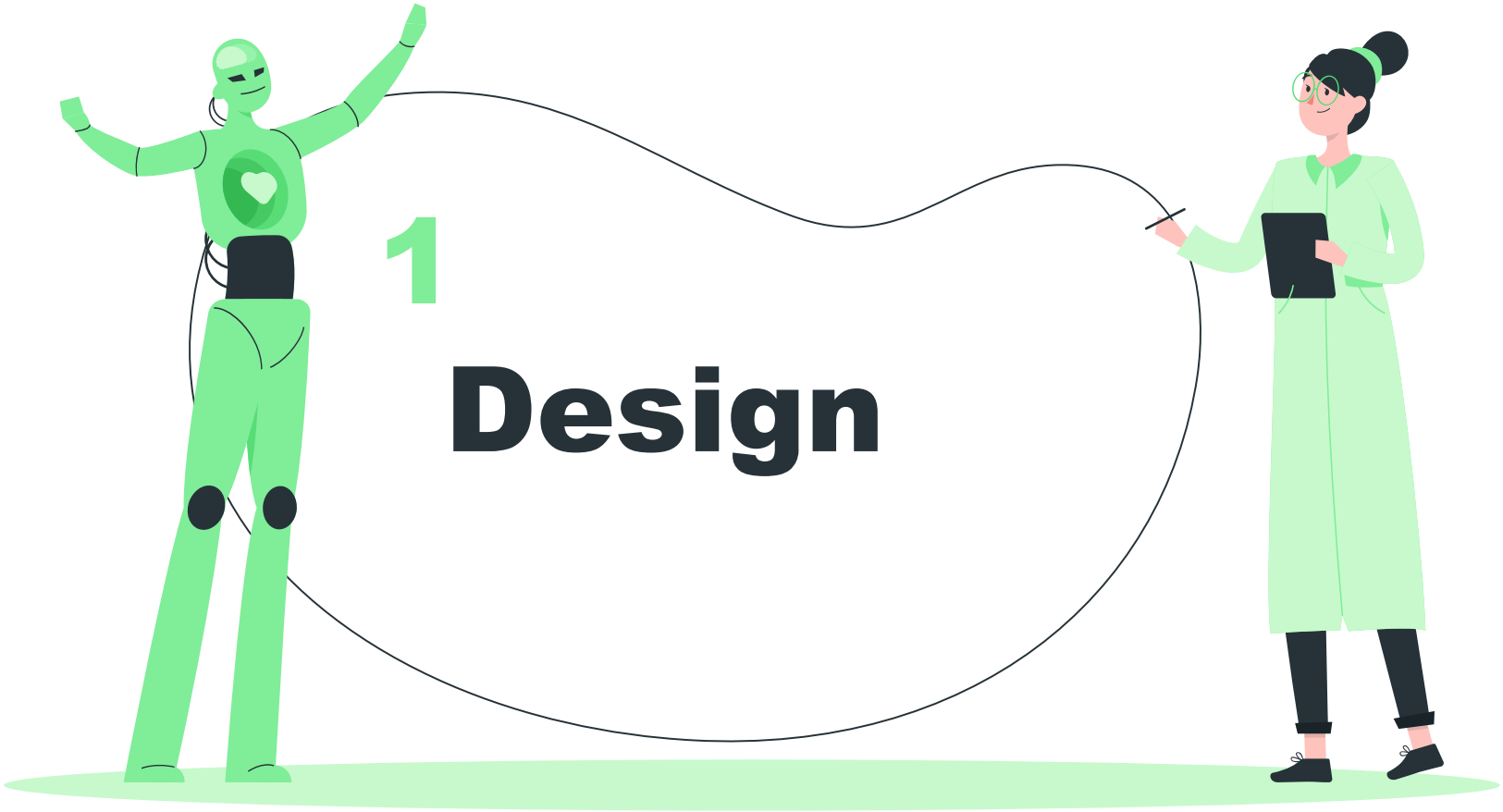
Cost

Total cost of our robot and how we manage it

4

Robot’s Operation

We just want to talk about how the robot works and how it moves.


1
Design

Some components used

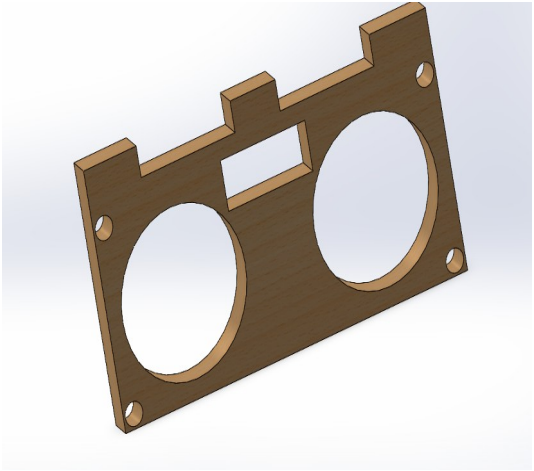
- **Ultra Sonic Sensors**
- **Sensor holders**
- **Hock**
- **Servo Motor**
- **DC Motor**
- **Shape “T”**
- **Spacers**
- **Arduino**
- **DC Driver**
- **Breadboard**
- **Emergency Button**
- **Bluetooth Module**

Other parts:

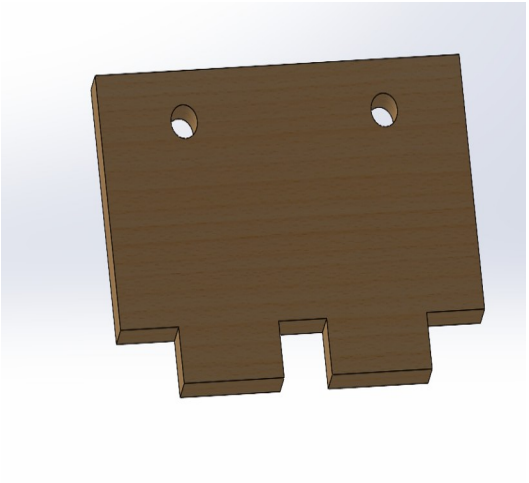
Wheels, Jumpers, Nuts, Base, Roof

Sensor Holder

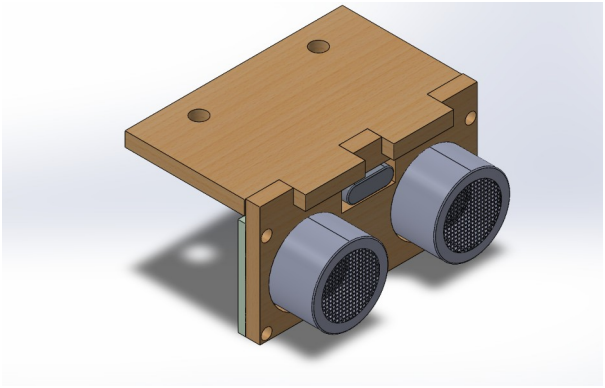
Part 1



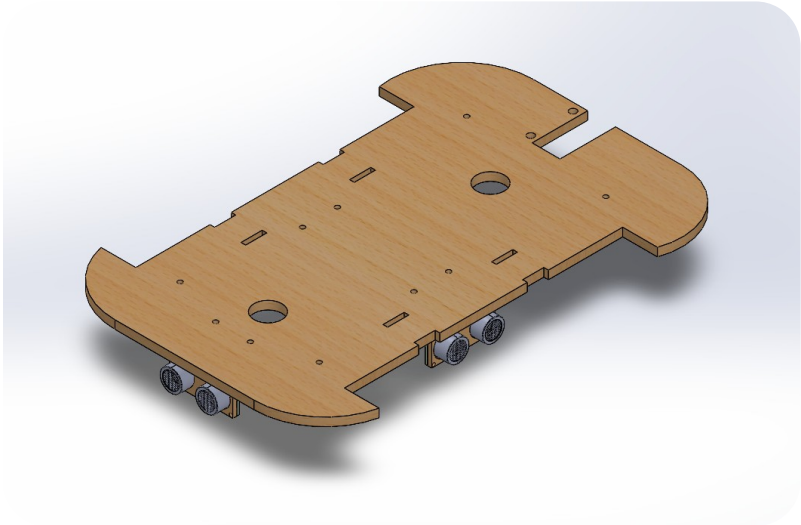
Part 2



Assembly

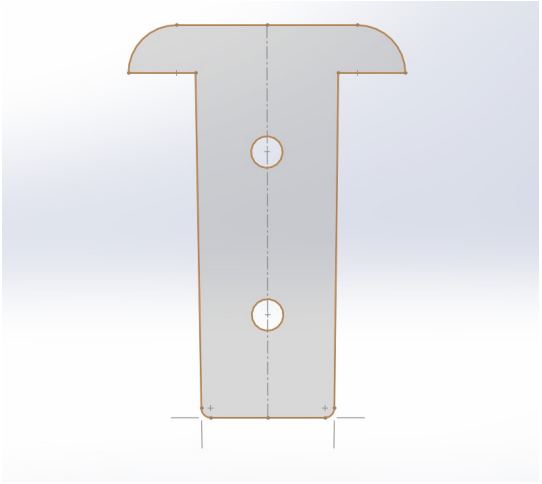


Base and sensor holder

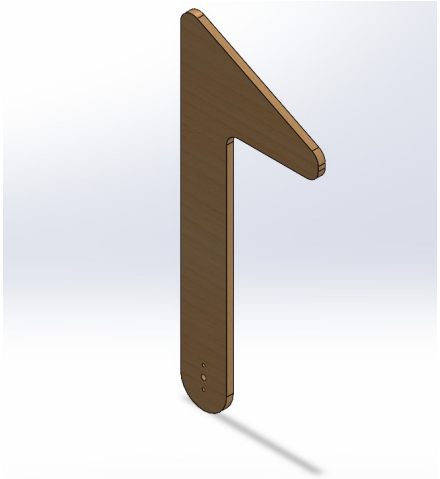


Hock and “T” section

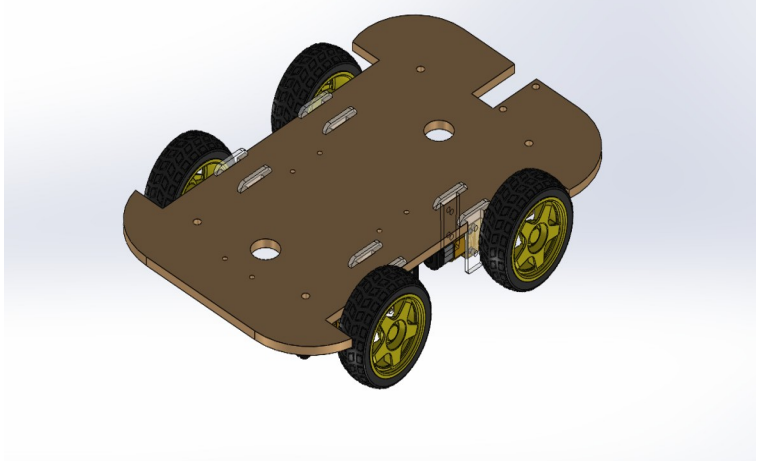
T section



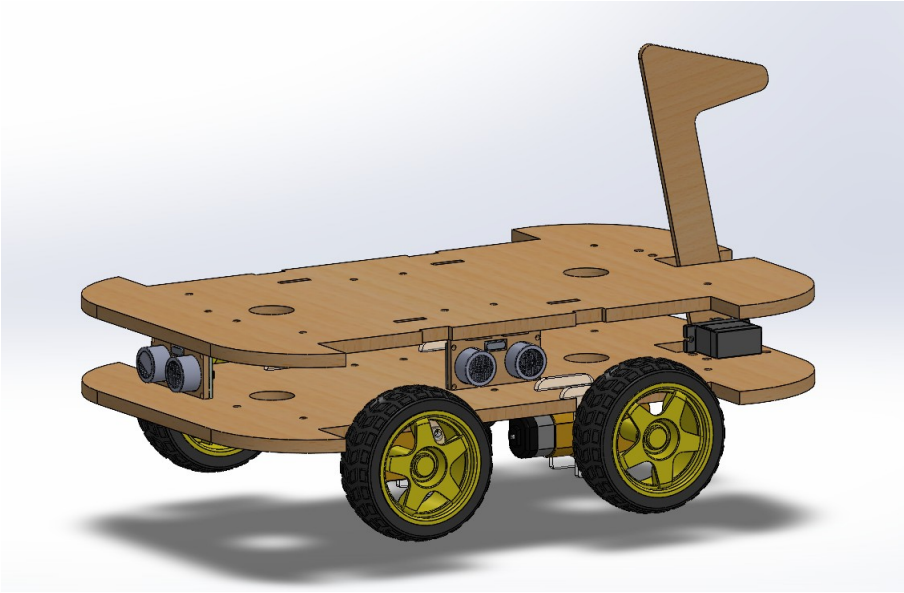
Hock

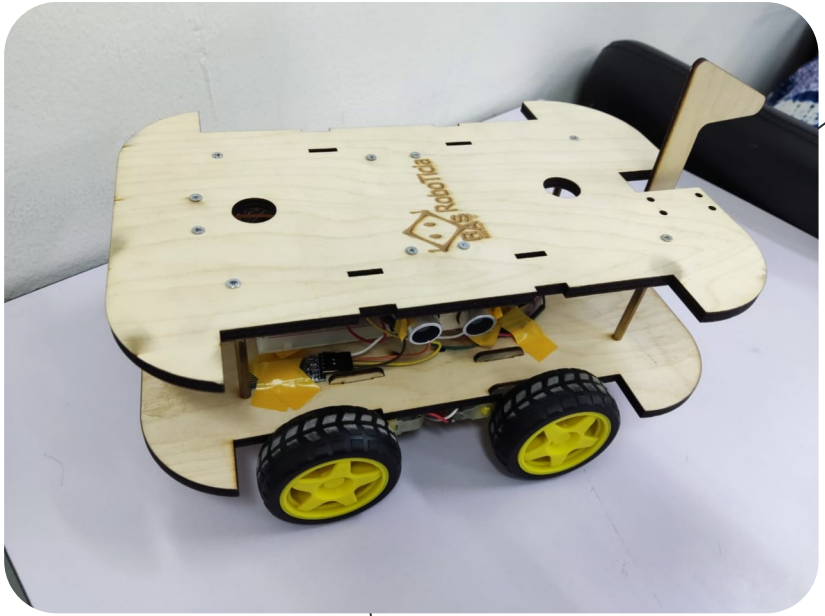


Assembly



Final Assembly





Some Problems & Solutions

We can not buy a sensor holder

Design 2 parts with respect to ultrasonic sensor dimensions

Fixing ultrasonic in the holder

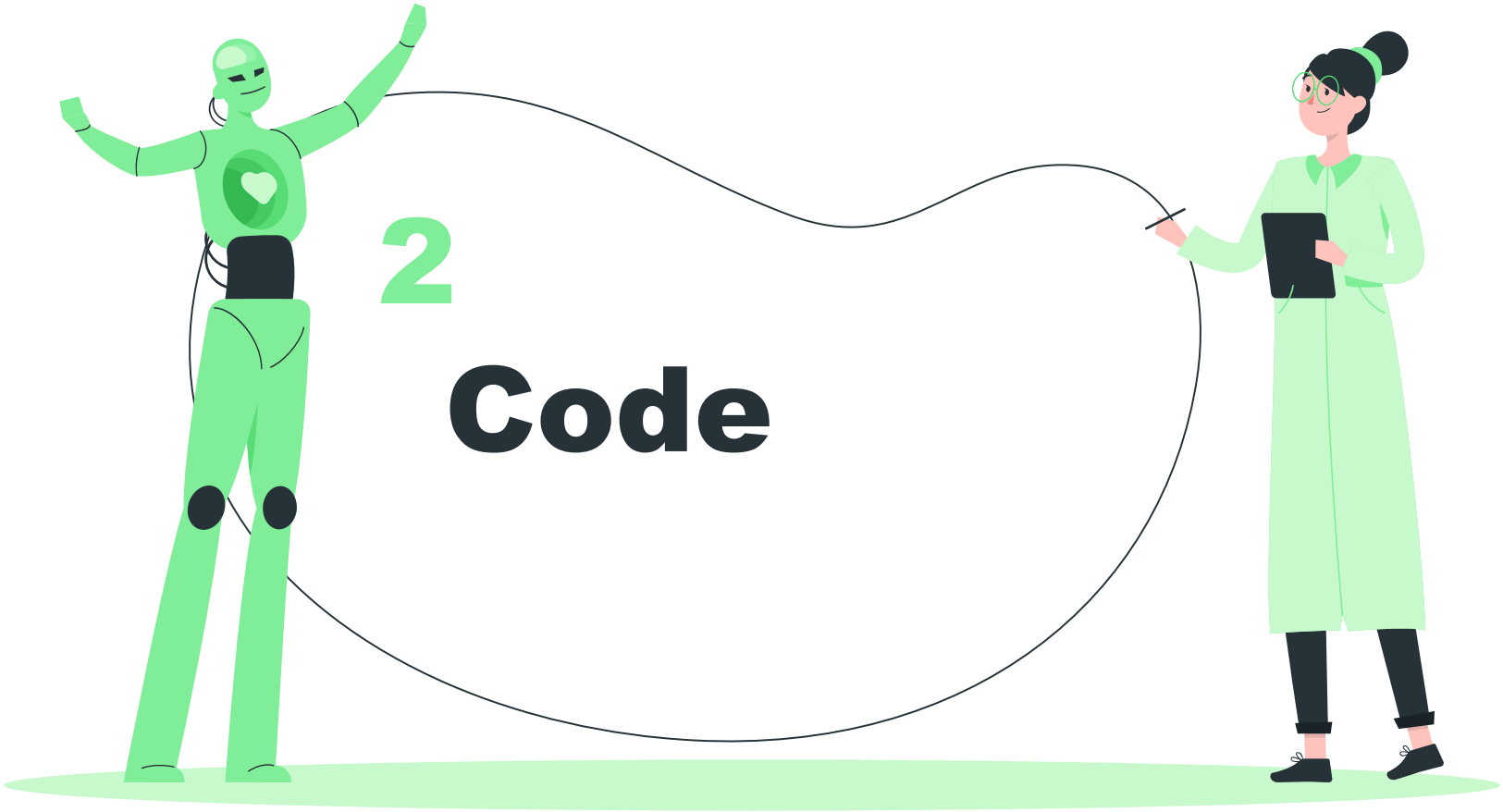
4 holes of ultrasonic, for fixing, are 2 mm and there aren't 2mm

locations of holes to install Dc motor .was incorrect

This mistake because we used wrong motor and we **fixed** it by replacing the motor with correct one.

Dimensions are not accurate in laser cutting

Increasing 0.3 mm for each hole


2
Code

Declarations

```
#include <Servo.h>
Servo myservo;
int pos = 90;
#define IN1 2
#define IN2 3
#define IN3 4
#define IN4 8
/*#define en1 A7
#define en2 A6*/
#define TRIG_FRONT A2
#define ECHO_FRONT A3
#define TRIG_LEFT A5
#define ECHO_LEFT A4
#define TRIG_RIGHT 10
#define ECHO_RIGHT 11
#define servoo 9
```

Explain:

some declarations of functions that will be used in next code.

Movement &Directions

```
char Direction;
void forward();
   void Back();
   void Left();
  void Right();
  void stope();
 void servoD();
int readSensor(int, int);
    int time_delay = 100;
         void setup() {
  // put your setup code
     here, to run once:
    Serial.begin(9600);
```

Explain:

forward- it's a function that make the car move forward.
Back- it's a function that make the car move Backward.
Stope- Stop the car when i release the button.
right- it's a function that make the car rotate 360 degree Right.
Why ?To make the motion of car easy in maze as the car rotate around itself to get away from the maze without crack with obstacles.
left- it's a function that make the car rotate 360 degree Left.
Why ?To make the motion of car easy in maze as the car rotate around itself to get away from the maze without crack with obstacles.
ServoD-Move the hang Down to catch the carriage
ServoB-Move the hang Up to leave the carriage
readSensor-take the trig and echo pins to work and return the distance in cm.

Motor Driver &Ultrasonic Sensors

Motor Driver

```
// Initialize motor driver
          pins as outputs
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
/*pinMode(en1, OUTPUT);
pinMode(en2, OUTPUT);*/
              stope();
```

•Ultrasonic Sensors

```
        // UltraSonic Sensors

pinMode(TRIG_FRONT, OUTPUT);
 pinMode(ECHO_FRONT, INPUT);
 pinMode(TRIG_LEFT, OUTPUT);
  pinMode(ECHO_LEFT, INPUT);
pinMode(TRIG_RIGHT, OUTPUT);
 pinMode(ECHO_RIGHT, INPUT);
     myservo.attach(servoo);
         myservo.write(pos);
                          }
```

Directions code

```
          void forward(){
 digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
 /*analogWrite(en1,150);
 analogWrite(en2,150);*/
                        }
             void Back(){
digitalWrite(IN1,HIGH);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,HIGH);
 /*analogWrite(en1,150);
 analogWrite(en2,150);*/
                        }
            void Right(){
 digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,HIGH);
 /*analogWrite(en1,150);
analogWrite(en2,150);  */

                        }
             void Left(){
digitalWrite(IN1,HIGH);
 digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
 /*analogWrite(en1,150);
analogWrite(en2,150);*/

                        }
            void stope(){
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);
 /*analogWrite(en1,150);
 analogWrite(en2,150);*/
                        }
```

Distance

```
int readSensor(int trig_pin, int echo_pin) {
                    // Send ultrasonic pulse
                digitalWrite(trig_pin, LOW);
                       delayMicroseconds(2);
               digitalWrite(trig_pin, HIGH);
                       delayMicroseconds(10);
                digitalWrite(trig_pin, LOW);

    long duration = pulseIn(echo_pin, HIGH);

   // Convert duration to distance in cm
    int distance = duration * 0.034 / 2;

                             return distance;
                                          }
                           void servoD(){
                       myservo.write(0);
                                          }
                           void servoB(){
                       myservo.write(90);
                                          }
```

Bluetooth Module & Serial Display

Bluetooth Module Directions

```
void loop() {
if(Serial.available()){
Direction =
Serial.read();
switch(Direction){
case 'F':{
forward();
break;
}
case 'B':{
Back();
break;
}
case 'L':{
Left();
break;
}
case 'R':{
Right();
break;
}
case 'S':{
stope();
break;
}
case 'X':{
```

Serial Display

```
int front_dist = readSensor(TRIG_FRONT, ECHO_FRONT);
int left_dist = readSensor(TRIG_LEFT, ECHO_LEFT);
int right_dist = readSensor(TRIG_RIGHT, ECHO_RIGHT);
Serial.print("Front: ");
Serial.print(front_dist);
Serial.print(", Left: ");
Serial.print(left_dist);
Serial.print(", Right: ");
Serial.println(right_dist);
break;
}
case 'U':{
servoD();
break;
}
case 'D':{
servoB();
break;
}
}
}
}
```


3
Cost

Total Cost : 1,400 EGP

- Ultra Sonic X3 — 45 EGP
- Dc Motor + wheels — 60 EGP
- Servo — 90 EGP
- Bluetooth Module — 200 EGP
- Arduino nano + cable — 200 EGP
- H-Bridge — 85 EGP
- TestBoard + Wires — 65 EGP
- Holder + Batteries X6 — 440 EGP

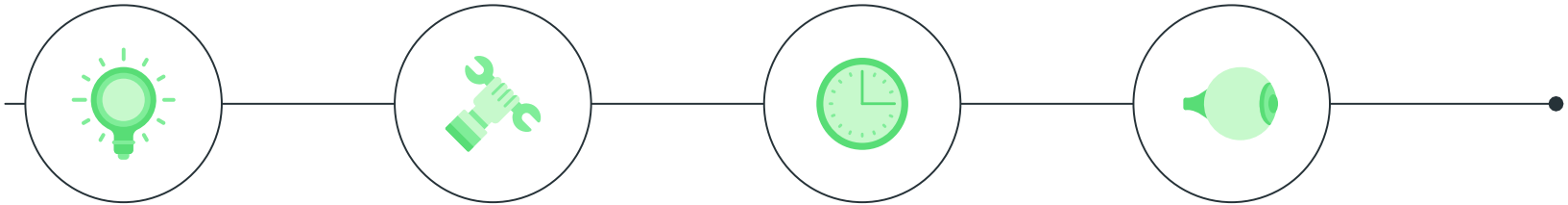
Project Timeline

Stage 1

Brainstorming stage

Stage 3

Set up the time plan



Stage 2

Determine the capabilities and materials that will be used

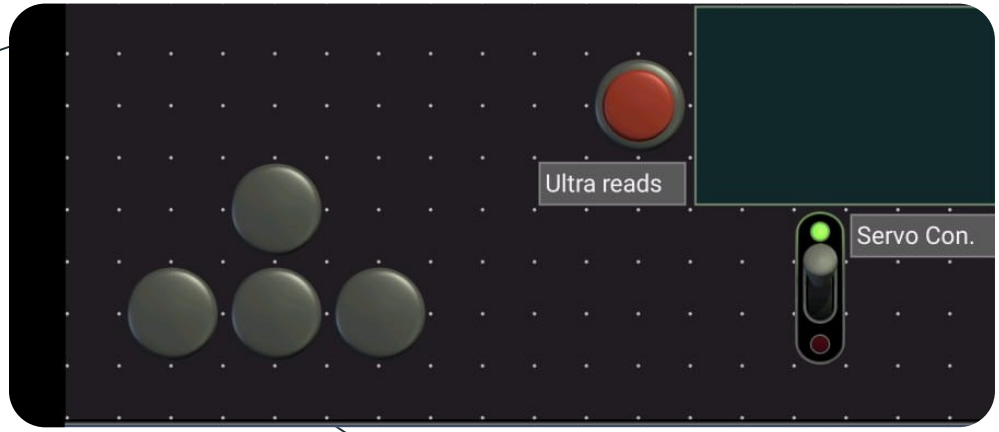
Stage 4

Finishing and preparing to the competition

Desktop Software

We used two programs:

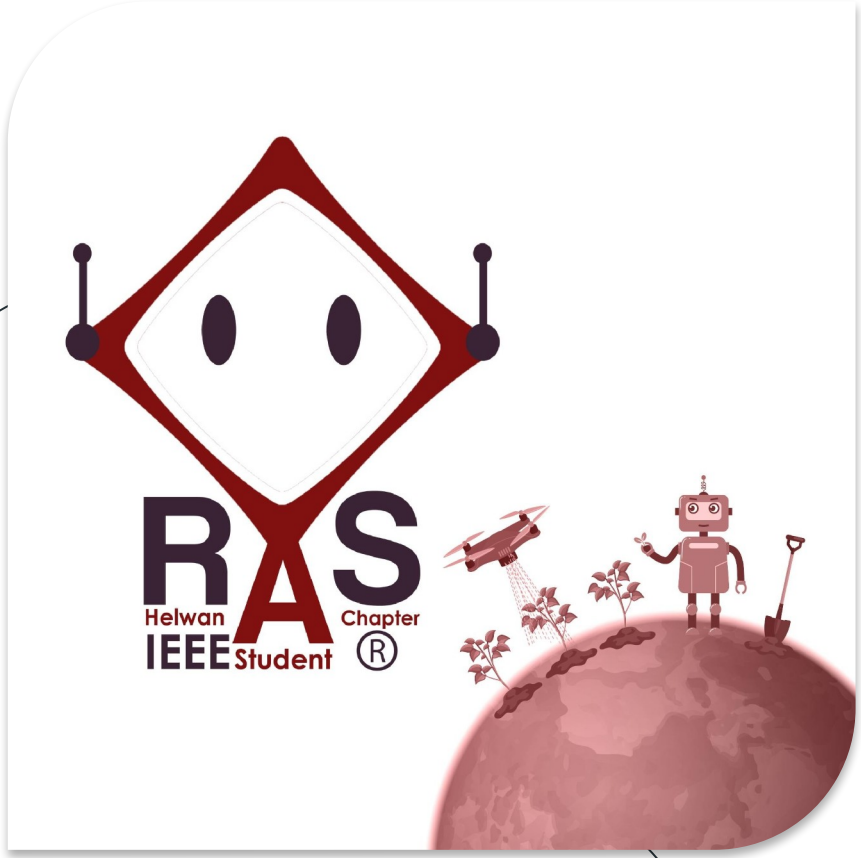
- SOLIDWORKS
- Arduino IDE



Mobile App

We used Blutooth Electronics app.


It was amazing
experience



Thanks

RAS

Thank you for this opportunity to participate in the course and the competition.