Jawatha Engineers

FUTURE ENGINEERS

Introduction

This project is a collaborative effort by a team of aspiring engineers to design and build a state-of-the-art robot.
The team consists of members with diverse skills and expertise in areas such as programming, mechanical design, and electronics.
We will examine various renewable energy sources, including solar, wind, hydro, and This document details the journey of the project, from the initial concept to the final product, highlighting the challenges and successes encountered along the way.

Our Team

The team comprises of an expert programmer, a skilled 3D modeler, and a seasoned coach. Their combined expertise ensures high-quality results.

Hasan Abo Fares brings expertise in couch construction and team building skills.

Khalid excels in programming the robot's functionalities and soldered many electrical parts.

Abdullah Alnasser designed the robot's 3D model and created the wiring of the circuit board .

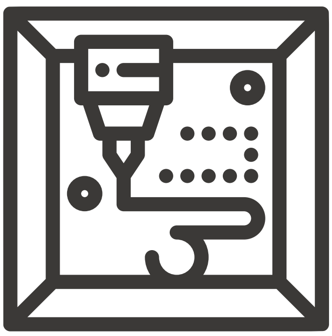
Robot Design

Sensoria's Form

Our robot, Sensoria, is a testament to innovative design. is fully 3d printed that sets it apart from the crowd. The 3D modeling process allowed us to make a fully costume robot for certain tasks.

benifits of fully 3D modeled design

Fully 3D design enhances visualization, improves accuracy, enables rapid prototyping, and fosters collaboration, leading to more innovative solutions and streamlined production processes.

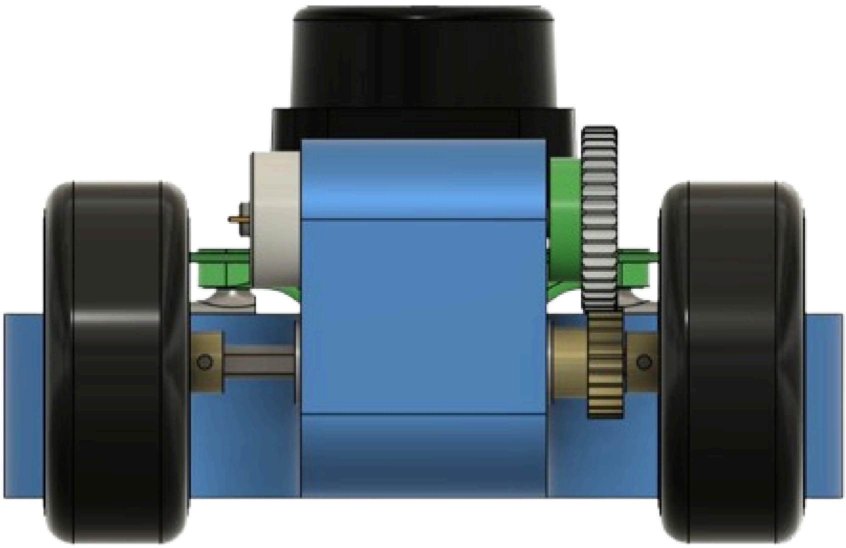


Fully 3D design allows for easier modifications, saving time and reducing costs during the development process.

A fully modeled design enhances accuracy, reduces errors, and improves communication during project execution.

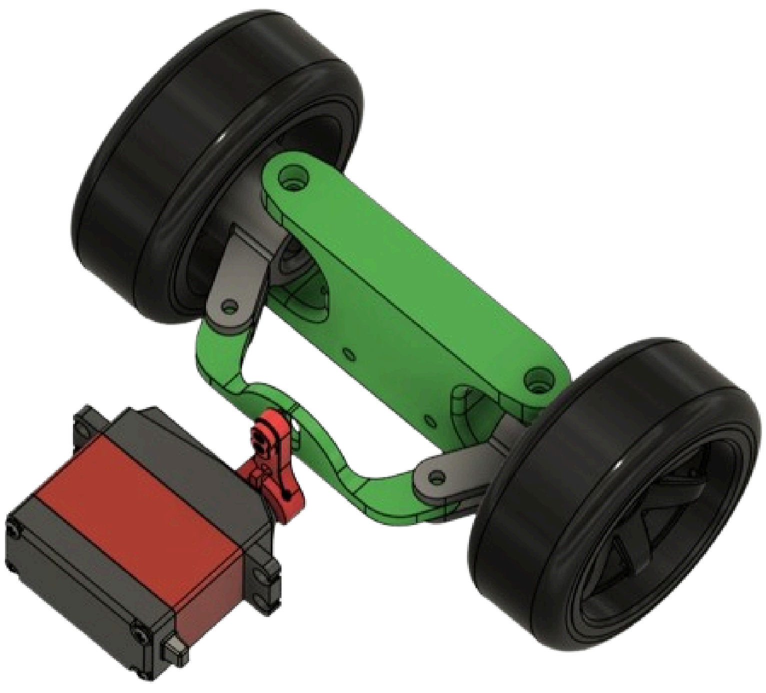
Robot Design Prototyping

Our robot went through multiple prototype iterations before reaching perfection. Each prototype allowed us to identify areas of improvement and make necessary adjustments. It was crucial for us to print every prototype in order to physically test and validate our design ideas. By doing so, we were able to ensure that our final product was not only functional but also met all of our desired specifications. The iterative process of prototyping ultimately led us to a successful and refined end result that exceeded our initial expectations.



Robot Steering System

Our steering system is meticulously designed and fully 3D modeled to ensure precision and reliability. By leveraging the capabilities of 3D printers, we are able to create intricate components that are not only durable but also highly accurate. With a torque of 20 kg, our steering system is able to provide stable and responsive control, making it ideal for a wide range of applications. Our advanced design ensures that our steering system is not only efficient but also built to last.



3D Modeling

3D Modeling journey

Software

Designing our robot using Fusion 360 was an incredible journey that combined creativity, engineering, and a lot of trial and error. Fusion 360 allowed us to create detailed 3D models, simulate movements, and visualize our design before building anything physical. This software was essential for prototyping and refining our ideas.

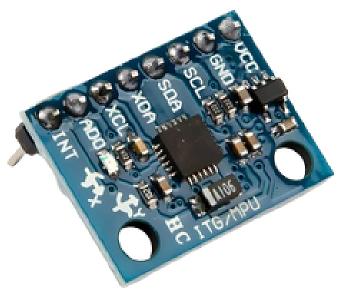
Technology

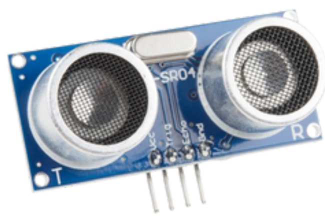
To bring our robot design to life, we relied heavily on advanced 3D printing technology and high-quality printers. The complexity of our design required precision and durability that standard printers simply couldn't provide. We needed a printer capable of handling various materials, such as strong plastics and flexible components, to accurately replicate our Fusion 360 models. The advanced 3D printer allowed us to create intricate geometries and tight tolerances that were crucial for our robot's performance.

Robot Capabilities

Sensoria is equipped with an array of sensors for comprehensive environmental understanding.

Sensors we used:





Ultrasonic Sensing

Sensoria uses five ultrasonic sensors positioned precisely in a designated area to detect objects in its path.

IMU Integration

The MPU6050 is important since it offers information on orientation and acceleration.

LiDAR Vision

Sensoria utilizes LiDAR for accurate 3D mapping of its surroundings.

The PIXY Camera 2

what are the reasons the camera is our main sensor

1. Light Sensitivity : Cameras can capture a wide range of light intensities, allowing them to function in various lighting conditions, from bright daylight to dimly lit environments.
2. Dynamic Range : They can record details in both highlights and shadows, providing a more complete representation of a scene compared to the human eye.
3. Color Accuracy : Cameras can reproduce a wide spectrum of colors, helping to capture the richness and vibrancy of a scene.

How we coded the robot

We utilized Arduino IDE and Pixymon, with their cutting-edge technology, to program the robot efficiently and effectively. The user-friendly interface of the IDE and Pixymon allowed us to easily write and upload code to the robot's microcontroller. The advanced features of the IDE, such as the serial monitor and debugging tools, enabled us to troubleshoot and fine-tune the robot's behavior. Overall, the Arduino IDE played a crucial role in the successful programming of the robot, showcasing the power and versatility of modern technology in robotics.

Arduino IDE

OUR CODE

Using functions in the Arduino IDE significantly improves both the efficiency and reliability of your code. By encapsulating specific tasks into functions, you make the code more modular, which reduces repetition and makes debugging easier. Functions allow you to organize your code logically, breaking it down into manageable, reusable blocks. This not only shortens the code but also makes it easier to maintain and scale for more complex projects.

The uniqueness of the code comes from how we design custom functions tailored to our specific project requirements. Every system or project has its unique parameters, inputs, and outputs, so structuring functions around these specifics makes the code fully individualized. Instead of relying on generic approaches, we create functions that directly address our goals, making the code more efficient for our use case.

Reliability is enhanced by simplifying the main loop and isolating potential sources of error within specific functions. When each function handles only one aspect of the project, it's easier to test and ensure its correct performance, thus reducing bugs or unexpected behavior. This method of coding increases the overall robustness of the system.

CODE EXPLAINED

The camera code

Using a camera to extract color signatures for robot navigation is an effective technique often employed in robotics for tasks like line following or object tracking. The basic idea is to use the camera as a sensor to detect specific colors or patterns in the environment, which then guide the robot’s movements.

```
  pixy.init();                // Initialize the Pixy camera
}

// Main Loop
void loop() {
  for (y = 0; y < 1; y++) { // Loop to execute the following block once
    servo1.write(175);      // Set the servo position
    delay(500); // Wait for 0.5 seconds
    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH); // Turn on the motors
    delay(1000); // Run motors for 1 second
    servo1.write(85); // Reset servo position
    delay(500); // Wait for 0.5 seconds
    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH); // Turn on the motors again

    // Check for detected color blocks
    if (pixy.ccc.numBlocks > 0) {
      // Control based on the signature of the detected block
      if (pixy.ccc.blocks[0].m_signature == 1) {
        RedControl();      // Call function to handle red block
      } else if (pixy.ccc.blocks[0].m_signature == 2) {
        BlueControl();     // Call function to handle blue block
      }
    } else {
      Serial.println("No blocks detected"); // Print debug message if no blocks are found
    }
```

CODE EXPLAINED

The steering system code

In the steering system code, we used the `HIGH` and `LOW` commands to control the direction of the motors responsible for steering. These commands allow us to directly control the output signals to the servo, which in turn controls the movement of the steering mechanism.

```
  pixy.init();                  // Initialize the Pixy camera
}

// Main Loop
void loop() {
  for (y = 0; y < 1; y++) { // Loop to execute the following block once
    servo1.write(175);      // Set the servo position
    delay(500); // Wait for 0.5 seconds
    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH); // Turn on the motors
    delay(1000); // Run motors for 1 second
    servo1.write(85); // Reset servo position
    delay(500); // Wait for 0.5 seconds
    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH); // Turn on the motors again

    // Check for detected color blocks
    if (pixy.ccc.numBlocks > 0) {
      // Control based on the signature of the detected block
      if (pixy.ccc.blocks[0].m_signature == 1) {
        RedControl();       // Call function to handle red block
      } else if (pixy.ccc.blocks[0].m_signature == 2) {
        BlueControl();      // Call function to handle blue block
      }
    } else {
      Serial.println("No blocks detected"); // Print debug message if no blocks are found
    }
```