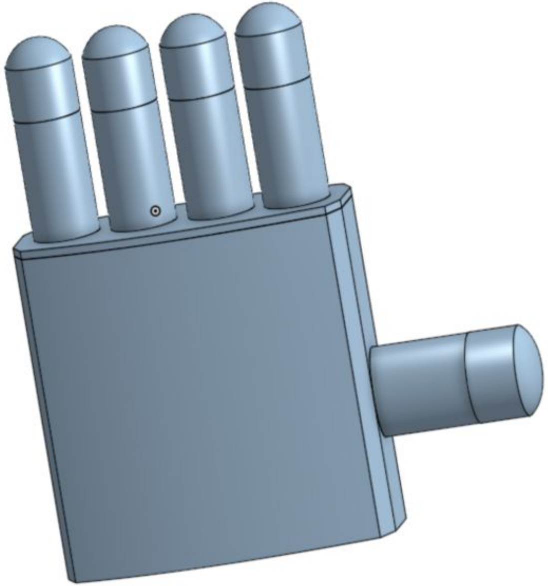
AstroGears

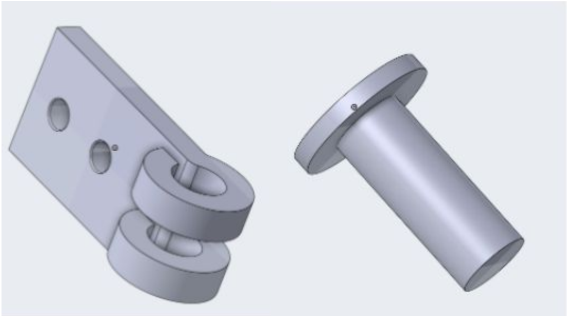
Empowering Heroes to Save Lives

Mohamed Elshal
Xingbo Wu
Naimeh Dughbaj

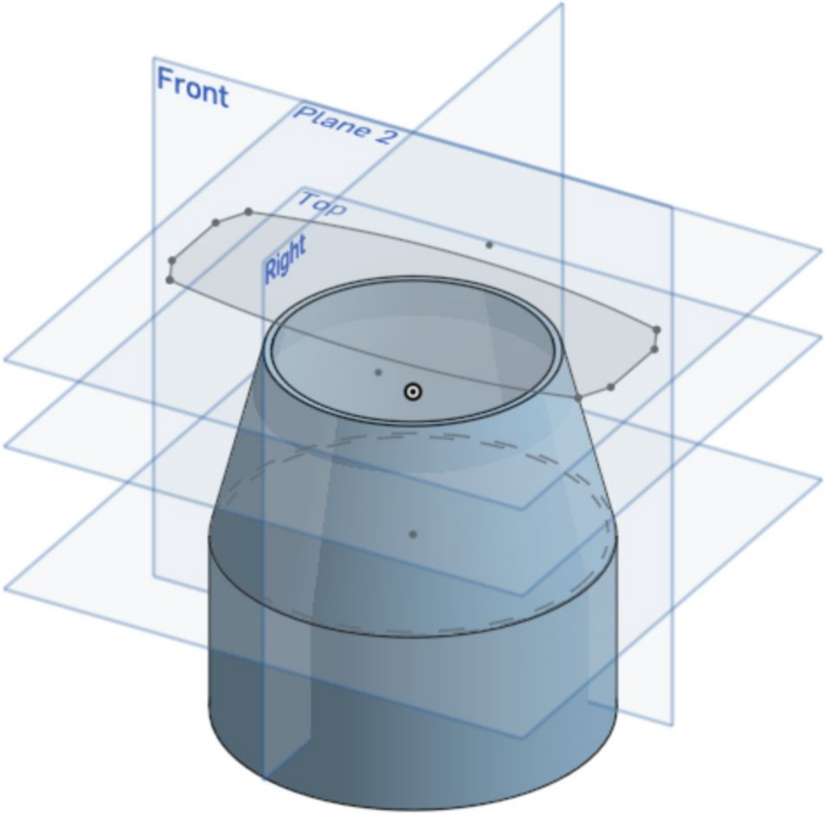
Evolution of the Vision

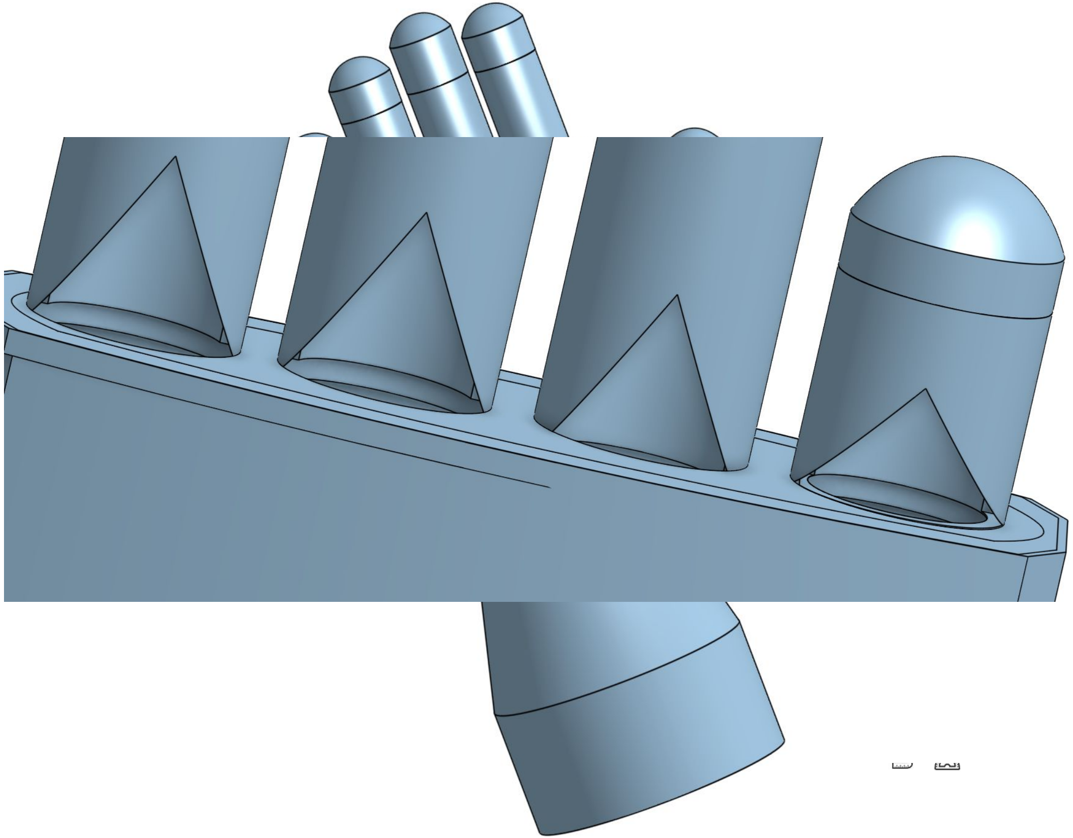
In the initial design, we attempted to incorporate hinges into the fingers to make them bendable. However, after numerous trial and error attempts, we faced challenges in designing a suitable hinge mechanism that would allow the fingers to bend as intended. Despite our efforts, we were unable to achieve the desired flexibility in the fingers using hinges.

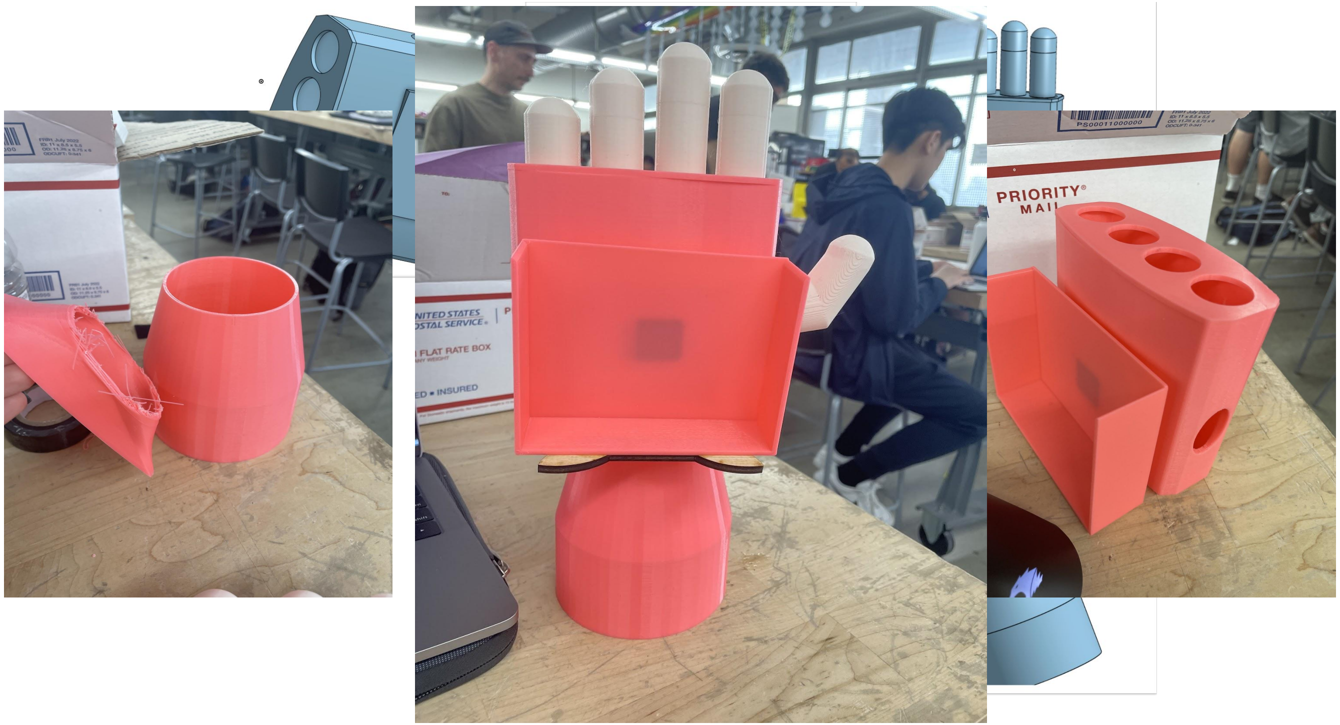




CAD Design Iteration






FLAT RATE BOX
INSURED
PRIORITY®

Circuit Network

Wiring of ESP32:
1. USB <- -> red rail;
2. GND <- -> blue rail;

Gas Sensor and Yellow LED Synergy

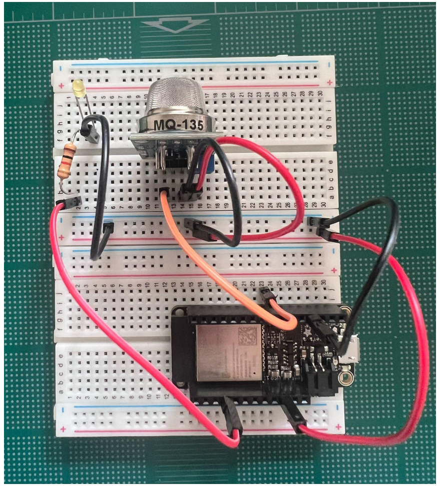
Wiring of MQ-135 Gas Sensor:
1. VCC <- -> red rail;
2. GND <- -> blue rail;
3. A0 <- -> ESP32 A3;

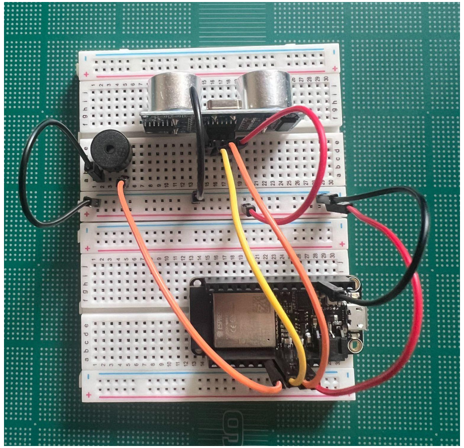
Wiring of Yellow LED:
1. Anode <-(R_1K) -> ESP32 pin 32;
2. Cathode <- -> blue rail;

Ultrasonic Sensor and Buzzer Synergy

Wiring of Ultrasonic Sensor:
1. VCC <- -> red rail;
2. GND <- -> blue rail;
3. Trig <- -> ESP32 pin 13
3. Echo <- -> ESP32 pin 12

Wiring of Buzzer:
1. Anode <- -> ESP32 pin 27;
2. Cathode <- -> blue rail;





Circuit Network

Temp. & Humi. Sensor Synergy

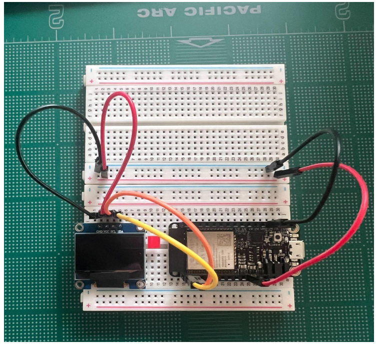
Wiring of Temperature and Humidity Sensor:
1. VDD <- -> red rail;
2. GND <- -> blue rail;
3. SDA <- -> ESP32 SDA pin;
4. SCL <- -> ESP32 SCL pin;
5. SCL and SDA require a pullup of 2-10K to VDD

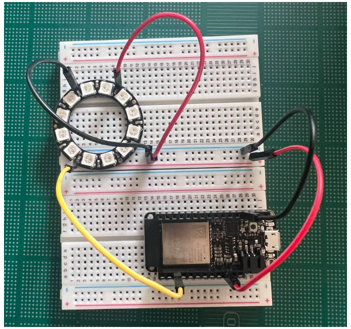
Wiring of Red LED:
1. Anode <- R_1k -> ESP32 pin 33;
2. Cathode <- -> Blue rail

Wiring of Green LED:
1. Anode <- R_1k -> ESP32 pin 15;
2. Cathode <- -> blue rail

OLED Screen Connection

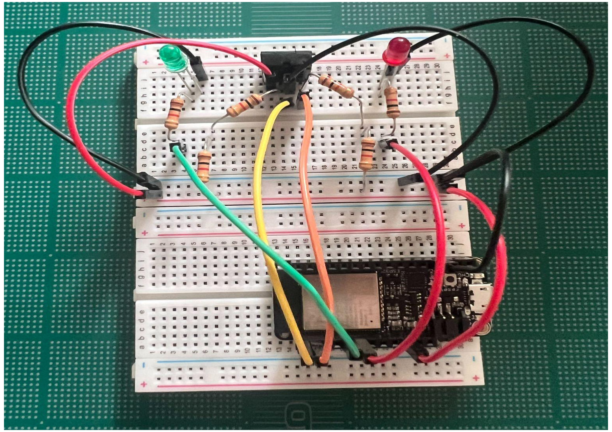
Wiring of OLED:
1. VCC <- -> red rail;
2. GND <- -> blue rail;
3. SDA <- -> ESP32 SDA pin;
4. SCL <- -> ESP32 SCL pin;





NeoPixel Ring Connection

Wiring of NeoPixel Ring:
1. PWR <- -> red rail;
2. GND <- -> blue rail;
3. Data Input <- -> ESP32 pin 14;



Gauntlet Code Iterations

Beside changing the power from 3V pin to the VBUS (USB), we refined the code of the ultrasonic sensor and measuring the distance using speed of sound instead of using a function.

```
// Measure the pulse duration on the Echo pin
unsigned long pulseDuration = pulseIn(echoPin, HIGH);
// Calculate the distance based on the speed of sound in ai
distance = pulseDuration * 0.0343 / 2.0; //calculating dist
```

Using <AM232X.h> lib instead of <Adafruit AM2320> to use getHumidity() function and solve humidity reading issue

```
// including Temperature and Humidity Sensor Libray
#include <AM232X.h>

// Read temperature and humidity data from the AM2320 sensor
float temperature = am2320.getTemperature();
float humidity = am2320.getHumidity();
```

Modified the distance threshold at which the buzzer provides audio feedback to better-fit rescue missions

```
// Turn on the buzzer if the distance is between 45 cm and 100 cm
if (45 <= distance && distance <= 100) {
  tone(buzzerPin, 255); // Generate a tone with frequency 255 Hz
  publisher.store("DistanceInfo", "Obstacle Detected");  //another cool f
} else {
  noTone(buzzerPin); // Turn off the buzzer
  publisher.store("DistanceInfo", ""); //id for our website to share dist
}
```

Additional more intuitive and user-friendly text display is used on the website when the data exceeds the threshold

```
if (temperature >= highTempThreshold) publisher.store("TemperatureInfo", "High Temperature"); //
else publisher.store("TemperatureInfo", "");
if (humidity <= lowHumidityThreshold) publisher.store("HumidityInfo", "Low Humidity");
else publisher.store("HumidityInfo", "");
```

Gauntlet Code Iterations

```
for(int i=0;i<NUMPIXELS;i++)
{
  // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
  pixels.setPixelColor(i, pixels.Color(255,0,0)); // Moderately bright green color.
  pixels.show(); // This sends the updated pixel color to the hardware.
  //delay(250); // Delay for a period of time (in milliseconds).
}
for(int i=0;i<NUMPIXELS;i++)
{
  // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
  pixels.setPixelColor(i, pixels.Color(0,255,0)); // Moderately bright green color.
  pixels.show(); // This sends the updated pixel color to the hardware.
  //delay(250); // Delay for a period of time (in milliseconds).
}
for(int i=0;i<NUMPIXELS;i++)
{
  // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
  pixels.setPixelColor(i, pixels.Color(0,0,255)); // Moderately bright green color.
  pixels.show(); // This sends the updated pixel color to the hardware.
  //delay(250); // Delay for a period of time (in milliseconds).
}
```

```
// RGB ring
pixels.setBrightness(100);
if (digitalRead(redLED) == HIGH) {
   for(int i=0;i<NUMPIXELS;i++)
   {
      pixels.setPixelColor(i, pixels.Color(255,0,0));
      pixels.show();
   }
}
else if (digitalRead(greenLED)  == HIGH && digitalRead(yellowLED)== HIGH) {
  for(int i=0;i<6;i++)
   {
      pixels.setPixelColor(i, pixels.Color(0,255,0));
      pixels.show();
   }
   for(int i=6;i<NUMPIXELS;i++)
   {
      pixels.setPixelColor(i, pixels.Color(255,255,0));
      pixels.show();
   }
}
else if (digitalRead(greenLED)  == HIGH) {
  for(int i=0;i<NUMPIXELS;i++)
   {
      pixels.setPixelColor(i, pixels.Color(0,255,0));
      pixels.show();
   }
}
else if (digitalRead(yellowLED)== HIGH) {
  for(int i=0;i<NUMPIXELS;i++)
   {
      pixels.setPixelColor(i, pixels.Color(255,255,0));
      pixels.show();
   }
}
else {
```

Instead of having the Neopixel Ring just show some gorgeous lights for no reason, we utilized its radiant characteristics to enhance the color of the current LED and reflect the current environment more intuitively

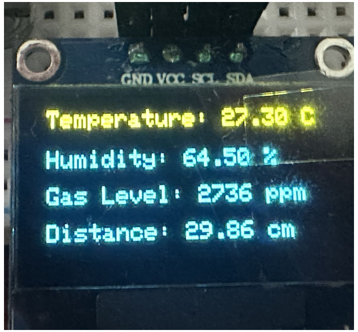
Iterative Testing and Improvements

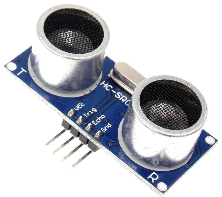
Iteration 1: Sensor Calibration

Variable: Gas Sensor Calibration

Measurable Tweaks: Adjusting the gas sensor sensitivity threshold.

Result: Improved accuracy in detecting gas leaks and providing timely warnings.





Iteration 2: Distance Sensor Enhancement

Variable: Ultrasonic Sensor Distance Calculation

Measurable Tweaks: Refining the speed of sound calculation for distance measurement.

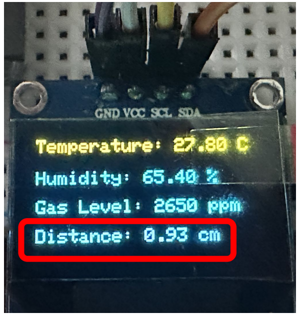
Result: Increased precision in measuring distances and obstacle detection.

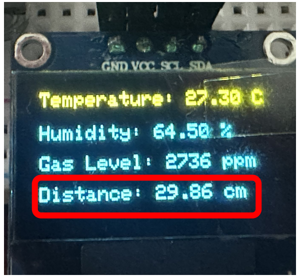
Gauntlet Code Challenges

Challenge 1: Incorrect Sensor Readings

Issue: We encountered inaccurate sensor readings from the gas sensor and ultrasonic sensor.

Solution: By adopting an engineering mindset, we meticulously checked sensor connections and ensured proper wiring, resolving the inaccuracies.





Challenge 2: Unexpected Behavior

Issue: We faced unexpected behavior in the code, leading to irregular outputs from the sensors.

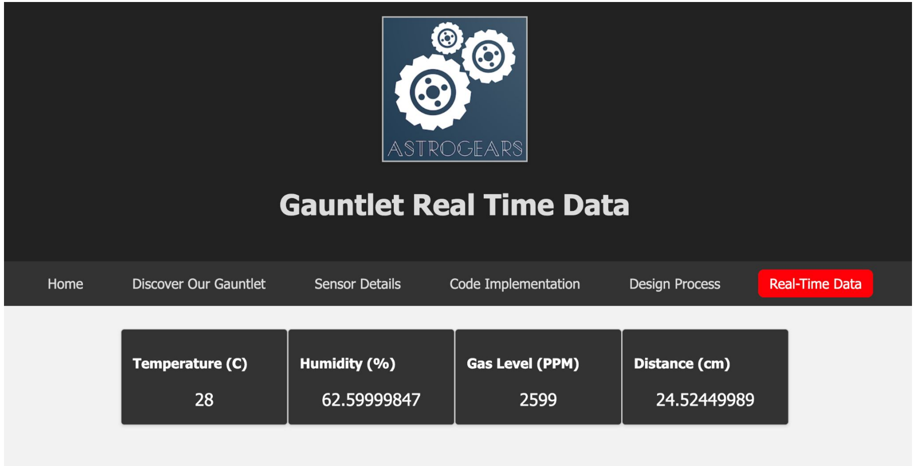
Solution: Utilizing the Serial Monitor, we printed sensor readings and critically analyzed the code logic to identify and fix the issues.

AstroGears Website – Connecting Technology and Rescuers

Our commitment to innovation extended beyond the physical gauntlet prototype. We also developed a cutting-edge website that serves as a crucial interface between our rescue technology and rescuers.

Key Features:

- **Modern Design**: The AstroGears website boasts a sleek and intuitive design, making it easy for users to navigate and access vital information.
- **Aesthetically Pleasing**: Our website features a dynamic layout with a blend of metallic dark blue, silver, and gray colors, reflecting the high-tech nature of our gauntlet.
- **Real-Time Data Monitoring**: Rescuers can access real-time data from the gauntlet sensors, including temperature, humidity, gas levels, and distance.
- **Seamless Data Transfer**: The website smoothly receives sensor data from the gauntlet and presents it in a visually appealing and easy-to-understand format.



AstroGears.W3spaces.com

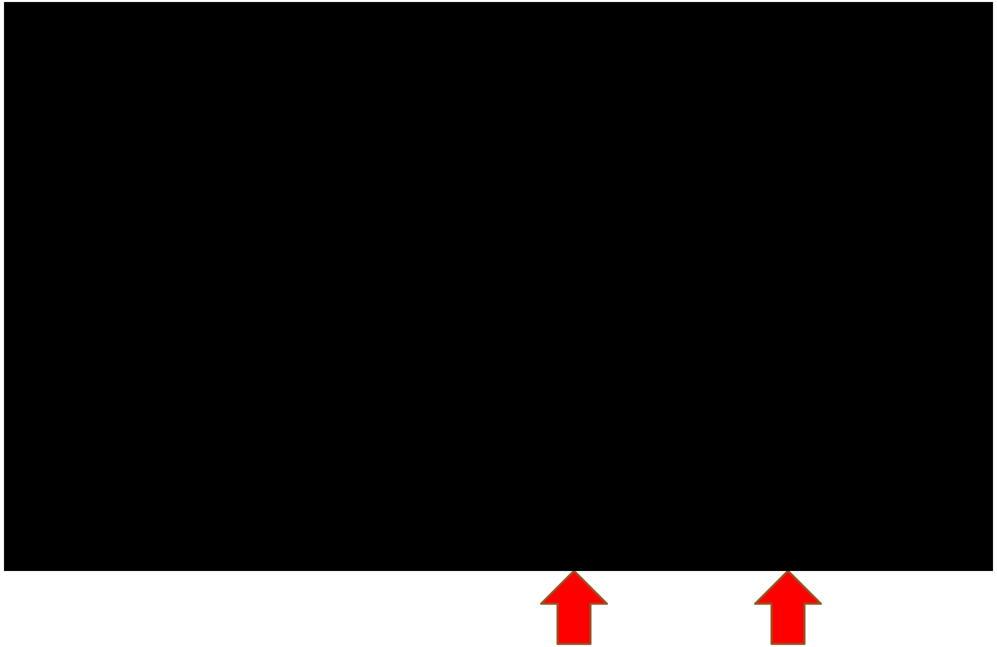
Seamless Integration for Rescuers

Our goal was to create a seamless integration of our gauntlet technology and website interface, enhancing the efficiency and safety of rescuers in disaster scenarios.

Immediate Alerts: Gas leaks and unsafe environmental conditions trigger instant alerts on the website, ensuring immediate responses.

User-Friendly Interface: Rescuers of all technical backgrounds can effortlessly navigate the AstroGears website and access essential features.
Intuitive Controls: The website's user-friendly controls allow rescuers to toggle various gauntlet functionalities with ease.

Collaborative Efforts: The website fosters collaboration among rescue teams, enabling multiple rescuers to access the same gauntlet data simultaneously.



Example of our cool feature “Immediate Alerts”

Real-World Testing and Optimization

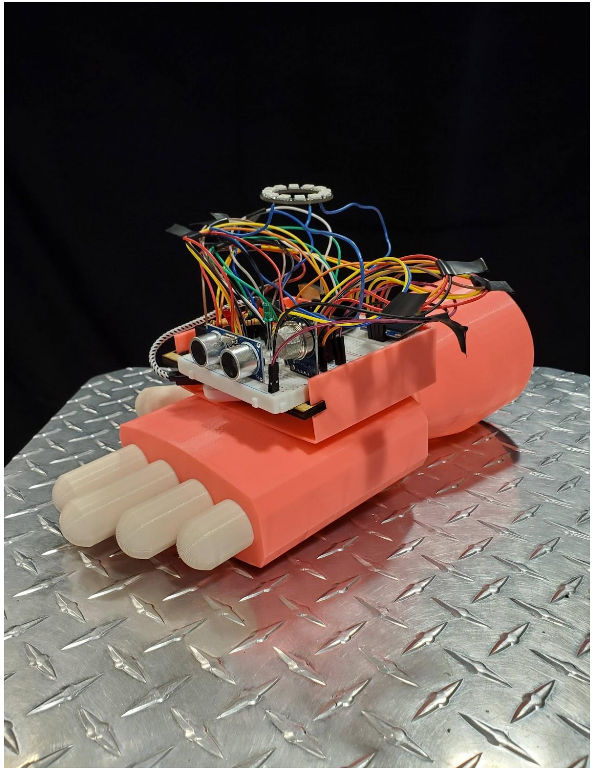
After completing the initial prototypes, we conducted rigorous real-world testing to evaluate the gauntlet's performance under various conditions.

Environmental Simulations such as:

Gas Leak Detection: Simulated gas leak scenarios to assess the accuracy of the MQ-135 gas sensor readings.

Temperature and Humidity Variation: Exposed the gauntlet to different temperature and humidity levels to evaluate sensor response.

Obstacle Navigation: Tested the gauntlet's ability to detect and avoid obstacles during movement.



Real-World Testing and Optimization

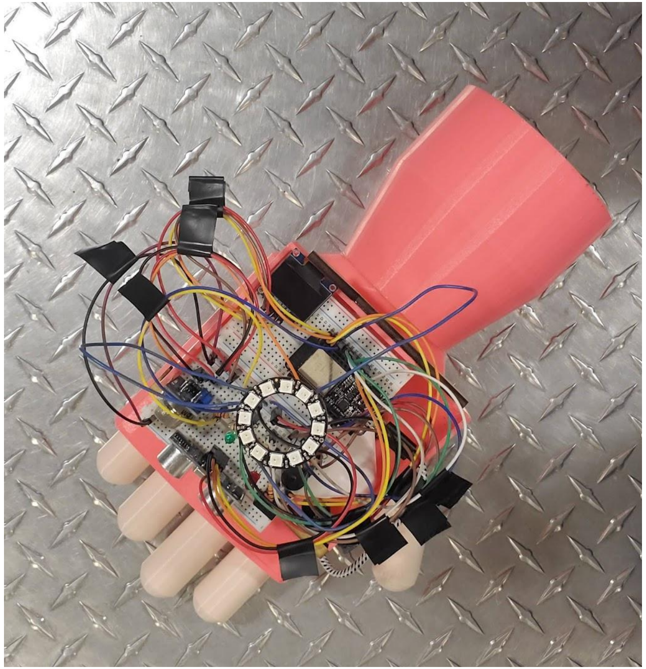
We also optimized the user experience as follows:

Audio Feedback: Incorporated buzzer feedback for proximity warnings.

Ergonomics: Gathered feedback from users to optimize the design for comfort during prolonged use.

Aesthetics and Credibility: Evaluated the visual appeal of the gauntlet to inspire confidence in the rescuer.

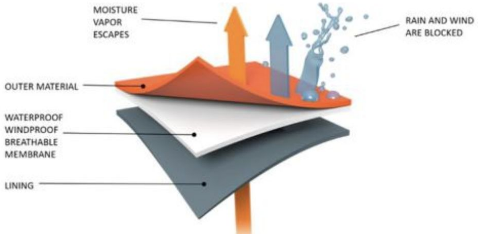
Code Refinement: Made code optimizations to reduce response times and improve overall performance.

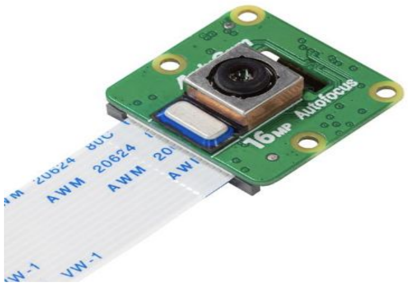


Future Design and Continuous Improvement

Next Steps:

- **Advanced Materials**: Investigate lightweight and durable materials to enhance comfort and mobility without compromising safety.
- **Biometric Sensors**: Explore biometric sensors to monitor vital signs and the rescuer's health during missions.
- **Built-in Camera**: Incorporate a camera for live video streaming and situational awareness.
- **Emergency Response Teams**: Collaborate with rescue organizations for real-world testing and user feedback.





Reflection and Closing Remarks

The journey of designing and developing the gauntlet has been an incredible learning experience for Team AstroGears. As we conclude this presentation, we would like to share our final reflections and gratitude.

Engineering Mindset:

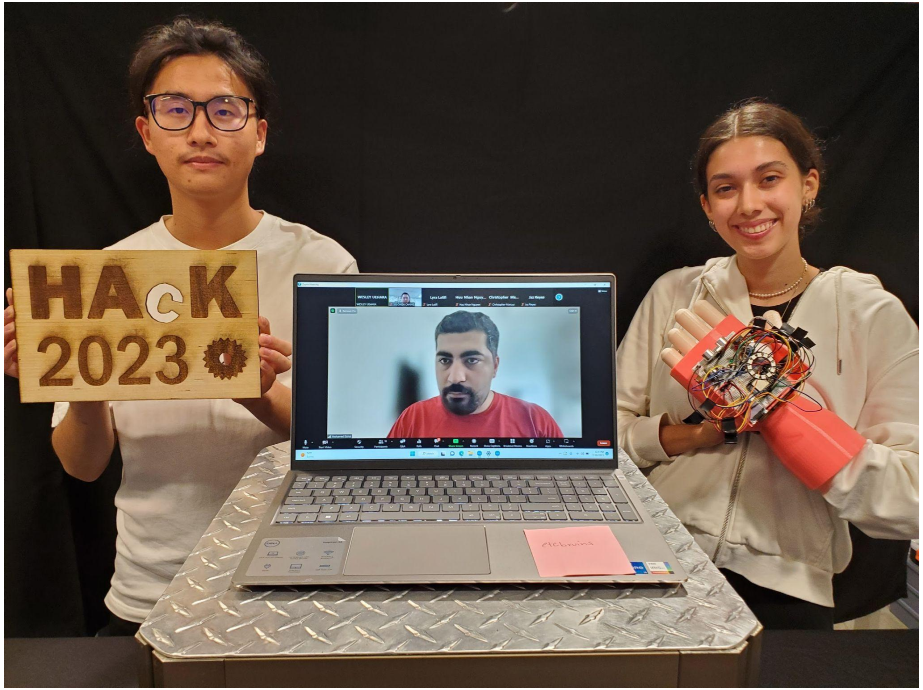
- **Embracing Challenges**: We learned to view challenges as opportunities for growth and problem-solving.
- **Iterative Approach**: Our iterative design process allowed us to continuously improve and innovate.

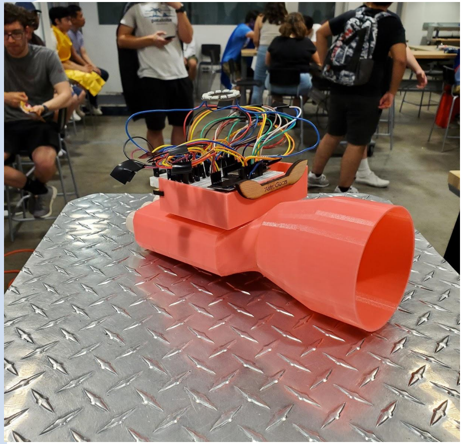
Team Collaboration:

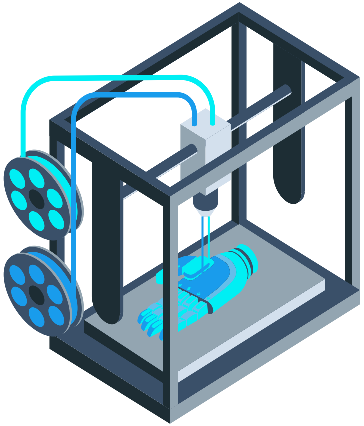
- **Synergy of Skills**: Our diverse team with different backgrounds and expertise complemented each other's strengths.
- **Effective Communication**: Open communication fostered creativity and a positive team environment.

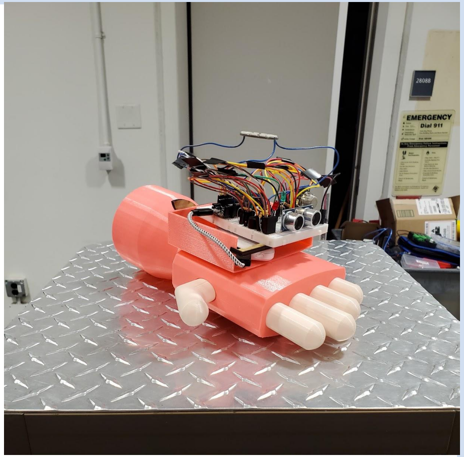
Gratitude and Acknowledgments:

We extend our sincere gratitude to Wes and our mentors for their guidance and support throughout the HAcK project.









Thank you!

Team 3: AstroGears