

S.H.I.N.E. (Saving Humans in Need Everywhere) 2.0

We Make Your Future S.H.I.N.E.!

**By: Victor Acuna, Evelyn Miguel, and
Wendi Aguilar**

Marco Antonio Firebaugh High School



Abstract

The purpose of this project is to create a cost-effective, yet efficient search and rescue robot. My hypothesis was that using the platform of Arduino and a variety of sensors and modules, a small car could be assembled which would be able to communicate remotely on the environmental status and the location of humans that are trapped, providing vital data to their potential rescue team. We created S.H.I.N.E. (Saving Humans In Need) 1.0 to meet the criteria for the robot, assuring the safe rescue of people in a natural disaster. This robot is a good solution for locating humans, as it not only saves the lives of the people trapped, but it also protects the lives of the first responders who go to help those people in need during those tough times.

Objective

To create a rescue robot that was cheaper than the market average, as the price is the main issue. It has to be sturdy enough to last a long time and be versatile, so that a single robot can be used to accomplish a variety of tasks.

Problem Statement

When disaster strikes, people panic, causing them to get hurt or worse. For instance, in the event of a natural disaster, such as an earthquake, there may be people trapped in fallen structures. These hazards make rescuing people dangerous for rescuers, so they would prefer to rely on robots. The problem is that high tech robots are expensive and lack certain needed tools. That is why I built a cost-effective, yet efficient, search and rescue robot to resolve the issue.

Design Requirements

- It needs to be much cheaper than the search and rescue robots that are already available.
- It needs to be able to detect humans and notify the rescuers if a human has been found.
- It needs to be able to detect dangerous gases and be able to notify people nearby, so that they can stay away from the area if they have the opportunity to do so.
- They need to be able to control the robot wirelessly.

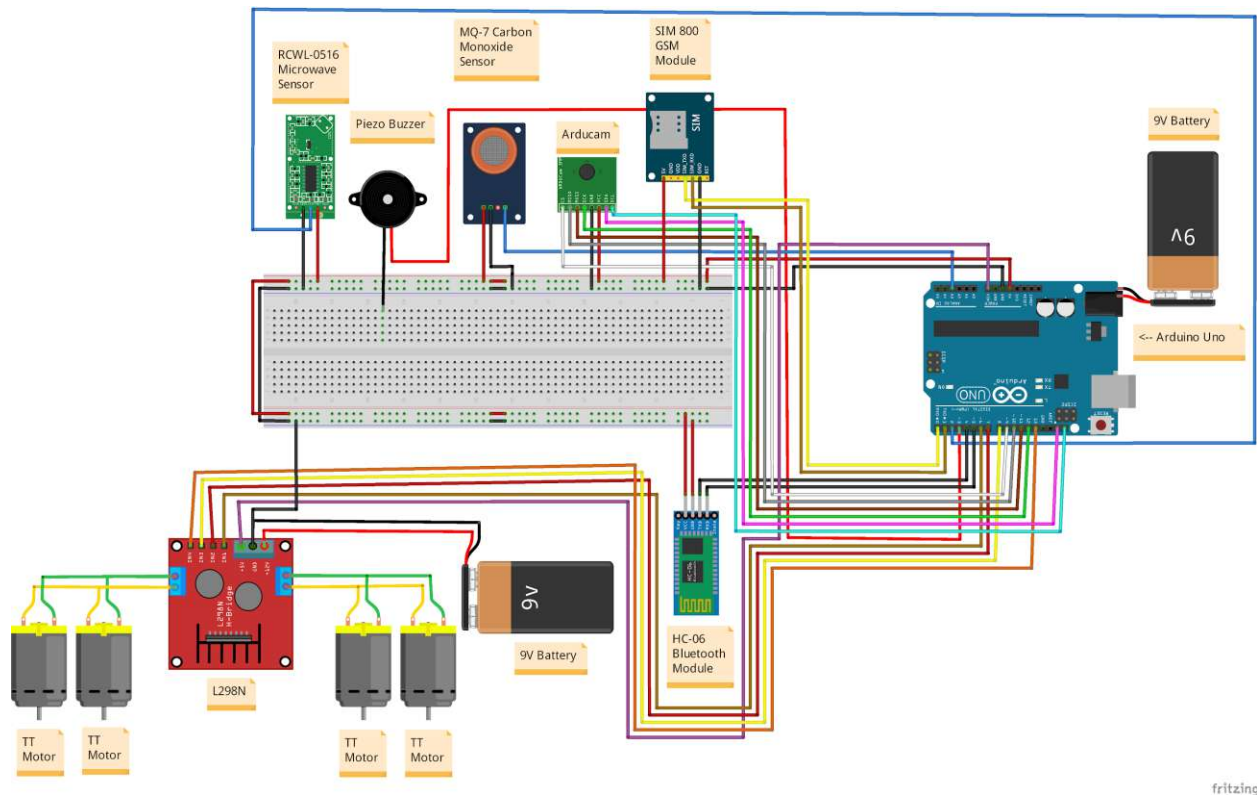
- The robot should have a camera of some sort, so that we can see remotely the place the car is surveying.
- It needs to be able to fit inside holes and gaps, that are too small for people to go into.

Solutions

We had a long design process where I came up with two plans and then compared the pros and cons of each plan to the criteria listed above. Below are two original proposals

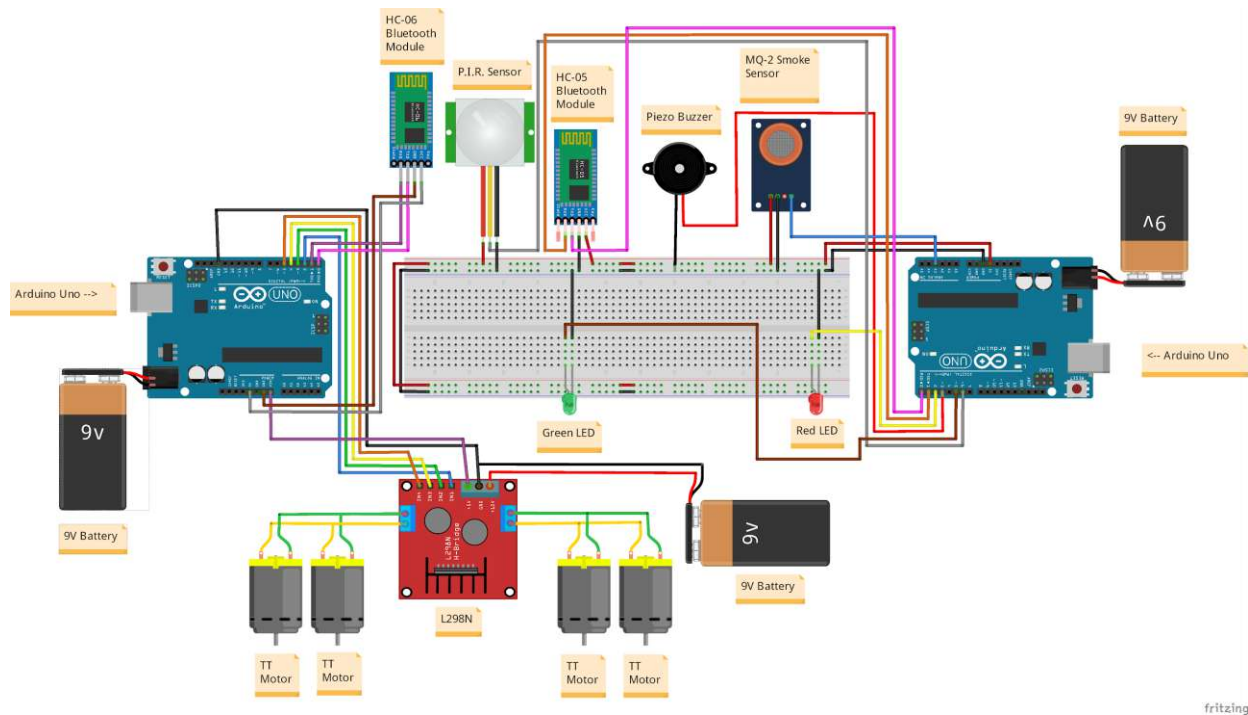
** Note that the GPS sensor and ESP-32 Cam I was going to include on my robot (as mentioned in my application) did not make the cut for any of these ideas due to the fact that people who created libraries for these sensors stopped updating their libraries for a while. This meant that there was no way for me to include those sensors to the project. As an alternative to the ESP-32 Cam, I used the camera of a phone, which will be attached to the idea that I end up choosing, to video call the first responders. The GPS sensor was excluded. **

Solution 1: Our first idea was a robot made with cardboard as its base and four TT with lightweight rubber wheels. It would have an RCWL-0516 microwave sensor to detect humans that are lost and/or trapped, and an MQ-7 carbon monoxide sensor to detect carbon monoxide that is in the area. A piezo buzzer would be included as well, to make a sound to alert people nearby of the carbon monoxide that is in the air. That way, the people, if they have the opportunity to do so, can stay away from the area. It would have an HC-06 Bluetooth module to be able to control the robot and an Arducam-2MP-Plus, to know where the robot is traveling in real-time. You would use a mobile phone and/or a laptop to stream what the robot is seeing using the camera's IP address (which you would type into Google). Also, a GSM module would be used to send data in case the PIR sensors picked up any human movement. All of this would be powered with two 9V batteries and one Arduino Uno.



Solution 2: The alternative was a simpler and rudimentary design. Instead of cardboard, it would have a pre-made acrylic base. It would still have four TT motors and lightweight rubber wheels, which would be controlled through Bluetooth with an HC-06 Bluetooth module. The motors, as well as the Bluetooth module which controls the wheels, would be powered with its own Arduino

Uno (AU1), and two 9V batteries. The first responders can use their phone and install Arduino Bluetooth Control to control the robot wirelessly. It would have a PIR sensor, a green LED, and an HC-05 Bluetooth module to detect if a human has been located. If a human has been located, the green LED will light up, and the reading will be sent onto the global organization's laptop or PC (if they have Arduino IDE installed) through Bluetooth. Instead of a camera, it would use the camera on a smartphone which would be mounted onto the robot. First responders can use their phone to receive a video call from the robot. The main difference with this idea is that this prototype would contain a red LED and a piezo buzzer to alert the lost or trapped person, if the person was still conscious, about possible gases and/or smoke that are dangerous. All of the other sensors, that were not connected to Arduino Uno (AU2) which controlled the car motors, would be connected to another Arduino Uno (AU3) with one 9V battery.



Decision

We decided to choose idea 2 for the initial prototype. The reasons why we went with that idea are as follows:

- It covers the criteria that the client gave us
- The PIR sensor is cheaper than microwave sensor
- The method of video streaming is cheaper than the one from idea 1.
- It includes a more durable base since acrylic is stronger than cardboard.
- The MQ-2 can detect more dangerous gases, as well as smoke, than the MQ-7 which only detects carbon monoxide.
- It includes both a buzzer AND an LED to notify people about any harmful gases nearby, so that people can stay away from the area if they have a chance.
- It includes an HC-05, which is cheaper than the GSM module, which notifies the first responders of the readings from the PIR sensor so they can know if a human has been found in those locations.

Itemized Budget Sheet (Prototype)

Part	Unit Dimension s	Retail Price \$	Price Per Unit \$	Quantity Used	Total Cost \$	Retail Source
Elegoo UNO Board	3" x 2.1"	\$10.90	\$10.90/ Board	1 Board	\$10.90	Amazon.com
SparkFun Redboard	2.7" x 2.1"	\$19.95	\$19.95/Board	1 Board	\$19.95	SparkFun.com
Breadboard 830 Point	2.2" x 7"	\$1.65	\$1.65/Breadboard	1 Breadboard	\$1.65	AliExpress.com
4 WD Robot Chassis Kit	70 mm x 22 mm	\$16.88	\$16.88/Kit	1 Kit	\$16.88	AliExpress.com
HC-06 Bluetooth Module	26.7 mm x 13 mm	\$2.23	\$2.23/Bluetooth Module	1 Bluetooth Module	\$2.33	AliExpress.com
L298N Motor Driver	43 mm x 43 mm	\$1.34	\$1.34/ Motor Driver	1 L298N Motor Driver	\$1.34	AliExpress.com
PIR Sensor with Holder	1.27" x 0.96"	\$1.10	\$1.10/ PIR Sensor With Holder	1 PIR Sensor With Holder	\$1.10	AliExpress.com
100 Piece 5mm LED Assorted Kit	5mm x 31.7 mm/ LED	\$1.55	About \$0.02/ LED	2 LED's (1 Green and 1 Red)	\$0.04	AliExpress.com
Piezo Buzzer	12 mm x 9.5 mm	\$0.88 for 10 Buzzers	\$0.09/ Buzzer	1 Buzzer	\$0.09	AliExpress.com
9V Battery Button Power Cable	6" x 0.5"	\$1.25 for 10 Battery Buckle Snaps	\$0.13/Battery Buckle Snaps	2 Battery Buckle Snaps Power	\$0.26	AliExpress.com

Battery Buckle Snaps Power Cable Connector		Power Cable Connector	Power Cable Connector	Cable Connector		
9V Battery Snap Connector Clip with Wire Holder Cable Leads Cord	15.5 mm × 1.4 mm	\$0.41 for 10 9V Battery Snap Connectors	About \$0.04/9V Battery Snap Connector	1 9V Battery Snap Connector	\$0.04	AliExpress.com
Male to Male Jumper Wires	Assorted Sizes	\$1.75 For 120 Male to Male Wires	\$0.01/ Wire	25 Male to Male Wires	\$0.25	AliExpress.com
Male to Female Jumper Wires	Assorted Sizes	\$3.36 for 120 Wires	\$0.03/Wire	11 Male to Female Wires	\$0.33	AliExpress.com
9V Battery	1.9 in × 1 in × 0.6 in	\$1.00	\$1.00/Battery	3 9V Battery	\$3.00	City Fashion Center in Lynwood California
Obama Free Phone (Safelink)	5.91" x 3" x 0.39"	\$0.00	\$0.00/Phone	1 Obama Free Phone	\$0.00	Plaza Mexico in Lynwood California
Electric Tape	¾" x 20 yds	\$2.29	About \$0.04/Foot of Tape	3 Inches of Tape	About \$0.01	Ekono 99 Cents Store
Total Cost					\$58.17	

Prototype Code

Code 1- Wheels with Bluetooth Module:

```
char t; // The HC-06 will store a character value
```

```
void setup() {
```

```
  pinMode(2, OUTPUT); // Left motors forward (Pin 2) is output
```

```
  pinMode(3, OUTPUT); // Left motors reverse (Pin 3) is output
```

```
  pinMode(4, OUTPUT); // Right motors forward (Pin 4) is output
```

```
  pinMode(5, OUTPUT); // Right motors reverse (Pin 5) is output
```

```
  Serial.begin(9600); // Get the Arduino ready to communicate with the serial monitor
```

```
  // At a rate of 9600 bits per seconds
```

```
}
```

```
void loop() {
```

```
  if (Serial.available()) { // If serial monitor readings are available regarding the motors
```

```
    t = Serial.read(); // The HC-06 will read the readings
```

```
    Serial.println(t); // The Hc-06 will print the readings on the Arduino Bluetooth Control
```

```
  }
```

```
  if (t == '1') { // Move forward (all motors rotate in forward direction)
```

```
    digitalWrite(2, HIGH); // Left motor will be going forward
```

```
    digitalWrite(3, LOW); // Left motors will not go reverse
```

```
    digitalWrite(4, HIGH); // Right motors will be going forward
```

```
    digitalWrite(5, LOW); // Right motors will not go reverse
```

```
  }
```

```
  else if (t == '2') { // Move reverse (all motors rotate in reverse direction)
```



```
digitalWrite(2, LOW); // Left motors will not be going forward
digitalWrite(3, HIGH); // Left motors will be going reverse
digitalWrite(4, LOW); // Right motors will not be going forward
digitalWrite(5, HIGH); // Right motors will be going reverse
}
```

```
else if (t == '3') { // Turn right (left side motors rotate in forward direction, right side motors
doesn't rotate)
```

```
digitalWrite(2, LOW); // Left motors will not be going forward
digitalWrite(3, LOW); // Left motors will not be going reverse
digitalWrite(4, HIGH); // Right motors will be going forward
digitalWrite(5, LOW); // Right motors will not be going reverse
}
```

```
else if (t == '4') { // Turn left (right side motors rotate in forward direction, left side motors
doesn't rotate)
```

```
digitalWrite(2, HIGH); // Left motors will be going forward
digitalWrite(3, LOW); // Left motors will not be going reverse
digitalWrite(4, LOW); // Right motors will not be going forward
digitalWrite(5, LOW); // Right motors will not be going reverse
}
```

```
else if (t == '5') { // Stop (all motors stop)
```

```
digitalWrite(2, LOW); // Left motors will not be going forward
digitalWrite(3, LOW); // Left motors will not be going reverse
digitalWrite(4, LOW); // Right motors will not be going forward
digitalWrite(5, LOW); // Right motors will not be going reverse
}
```

```
delay(100); // Wait 0.1 seconds before repeating the code from void loop again
```

```
}
```

Code 2- Sensors:

```
#include <SoftwareSerial.h> // Library used for serial communication with a digital pin other  
than that of the serial port
```

```
SoftwareSerial mySerial(0, 1); // RX, TX pins for HC-05 module
```

```
//PIR Sensor
```

```
long unsigned int lowIn1; //the time when the sensor outputs a low impulse
```

```
long unsigned int pause = 5000; //the amount of milliseconds the sensor has to be low before we
```

```
    // Assume all motion has stopped
```

```
int pirPin1 = 6; // The digital pin connected to the PIR sensor 1's output
```

```
boolean lockLow1 = true;
```

```
boolean takeLowTime1;
```

```
int greenLed = 5; // The digital pin connected to the green LED
```

```
//Smoke Sensor
```

```
int redLed = 2; // The digital pin connected to the red LED
```

```
int buzzer = 3; // The digital pin connected to the buzzer
```

```
int smokeA0 = A3; // The analog pin connected to the MQ-2 sensor
```

```
int sensorThres = 200; // The minimum value that the MQ-2 sensor needs to detect smoke and/or  
harmful gases
```

```
void setup() {
```

```

pinMode(redLed, OUTPUT); // Set the red LED as output
pinMode(greenLed, OUTPUT); // Set the green LED as output
pinMode(buzzer, OUTPUT); // Set the buzzer as output
pinMode(smokeA0, INPUT); // Set the MQ-2 sensor as input

// put your setup code here, to run once:
mySerial.begin(9600); //Send the MQ-2 readings to the HC-05 serial monitor found on the
Arduino IDE

pinMode(pirPin1, INPUT); // Set the PIR sensor as input
}

void loop() {
  if (digitalRead(pirPin1) == HIGH) { // If the readings of the PIR sensor is high
    if (lockLow1) {
      lockLow1 = false; // Human motion was detected
      mySerial.println("Living Thing Found! (PIR 1)"); // The HC-05 serial monitor will print
        // Out that a human was found by the PIR sensor
      digitalWrite(greenLed, HIGH); // The green LED will turn on

      delay(1000); // Wait 0.2 seconds before moving on to the next line of code
    }
    takeLowTime1 = true; // Make sure this is only done at the start of a HIGH phase
  }

  if (digitalRead(pirPin1) == LOW) {

    if (takeLowTime1) {
      lowIn1 = millis(); // Save the time of the transition from high to LOW
      takeLowTime1 = false; // Make sure this is only done at the start of a LOW phase
    }
  }
}

```

```
if (!lockLow1 && millis() - lowIn1 > pause) { // If the sensor is low for more than the given
pause
```

```
lockLow1 = true; // Assume that no more motion is going to happen
```

```
mySerial.println("Living Thing Lost (PIR 1)"); // The HC-05 serial monitor will print
```

```
// Out that a human was lost according to the PIR sensor
```

```
digitalWrite(greenLed, LOW);
```

```
delay(1000); // Wait 0.2 seconds before moving on to the next line of code
```

```
}
```

```
void smoke();
```

```
int analogSensor = analogRead(smokeA0);
```

```
Serial.println(analogSensor); // Print the MQ-2 Sensor's readings onto the serial monitor
```

```
if (analogSensor > sensorThres) // If the MQ-2's readings of the air is greater than the
sensorThres
```

```
// (200), then there is smoke and/or dangerous gas in the air.
```

```
{
```

```
digitalWrite(redLed, HIGH); // The red LED will turn on
```

```
tone(buzzer, 800, 800); // The buzzer will make a sound from tone 800 and 800
```

```
delay(200); // Wait 0.2 seconds before moving on to the next line of code
```

```
tone(buzzer, 600, 800); // The buzzer will make a sound from tone 600 and 800
```

```
delay(200); // Wait 0.2 seconds before moving on to the next line of code
```

```
}
```

```
else
```

```
{
```

```
digitalWrite(redLed, LOW); // The red LED will be off; it will not turn on
```

```
noTone(buzzer); // The buzzer will give off no tone
```

```

}

delay(100); // Wait 0.1 seconds before moving on to the next line of code. In this case,

//it will repeat the void loop again

}

}

```

Prototype Description

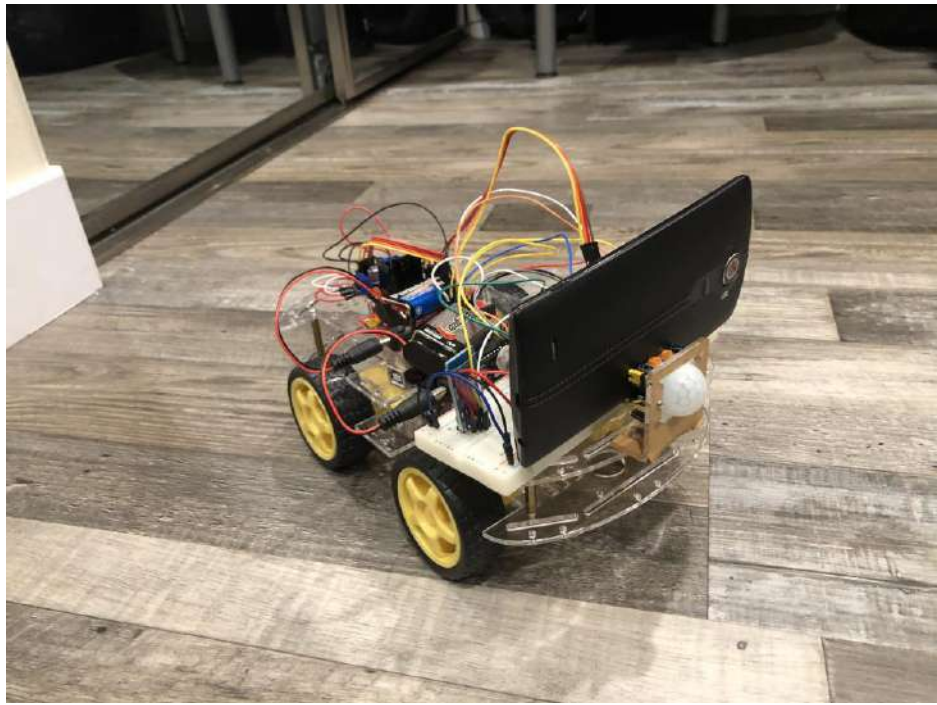
The main goal of S.H.I.N.E. 1.0 is to detect people that are trapped and/or hurt in places that are too dangerous or too small for the first responders to go into. These include places like collapsed buildings, small tunnels, etc. It also needs to have a sensor to detect any natural gasses, such as methane, propane, hydrogen, and carbon monoxide that are found in the air. Most importantly, there must be a way for first responders to be able to see what the robot is seeing/experiencing in real-time.

The prototype we ended up creating is manually controlled. It uses four TT motors, each with a lightweight plastic rubber tire. It is controlled through Bluetooth, with the help of an HC-06 Bluetooth module, and an app called Arduino Bluetooth Control, which is an app with a remote to control the car. It also includes an MQ-2 sensor that detects smoke and/or harmful gasses in the atmosphere. If harmful gas or smoke is detected, a red LED lights up and a piezo buzzer makes a sound, indicating that people nearby should stay away from that area if they have the opportunity to do so. The robot also includes a PIR sensor that detects human movement. If the sensor detects any human activity in the area that they are investigating, a green LED will light up, indicating that they have been found and that they should not move, and the readings of the sensor will be sent wirelessly, through Bluetooth, to the Arduino serial monitor. This is so that first responders can know that a human is in that area and that they should rescue them. Lastly, we included a free government phone to video call. This is the method that we used to be able to see what the robot is seeing in real-time.

The Arduino is essential to the prototype functionality. It allows us to be able to program how we want the car to move. It gives us a variety of ways to make the motors move. We chose to make it be controlled through Bluetooth with an Android phone that has Android Remote Controller installed. With the Arduino, we were also able to adjust the sensitivity of the PIR sensor to make it be triggered by certain movements. We adjusted it to make it sensitive to human movements. The Arduino allows us to simultaneously turn on an LED when a human has been found. We were also able to make the piezo buzzer have a variety of sounds. There is an infinite amount of sounds to choose from in the Arduino library. For the project, we wanted to make it sound like a fire notifier, so we used tones 800 and 600. Not to mention, it also allowed us to simultaneously turn on the LED and make a sound with the buzzer when the smoke and dangerous gases have been found. The smoke sensor, when used with Arduino, is triggered by certain types of smoke. Our research showed that a sensitivity of 300 would detect methane and propane. Lastly, the Arduino allowed us to use an HC-05 Bluetooth module to send the PIR

sensor readings, of whether a human had been found or not, wirelessly through Bluetooth. The laptop or PC that the first responders will use, which has Arduino IDE installed, will receive those readings which the first responders can see.

One of the main failures of our robot is that the Arduino, that was in charge of all the sensors, did not operate successfully. It only focused on the detection of humans, rather than detecting humans and detecting harmful gases and smoke. Whenever the Arduino tried to focus on both sensors simultaneously, the readings would get mixed up. I did further research and according to a post on Stack Exchange, “To use multiple loops at the same time you need an Arduino Due Board.” That is when I figured out that I cannot include multiple void loops in our code since it could only focus on one void loop. To fix this issue, I had to use three Arduino Uno boards. One Arduino board focused on the motors (AU1) , the second Arduino board focused on the PIR sensor (AU2), and the last Arduino Uno focused on the MQ-2 sensor (AU3). This was needed in order to be able to do the tests for the data. The modifications for the final design will include the three Arduino Uno’s. Another failure was that the PIR sensor did not successfully detect humans both in the back and front of the car. It was only able to detect humans, at a very good range, from the front of the car.



Tests Made with Prototype:

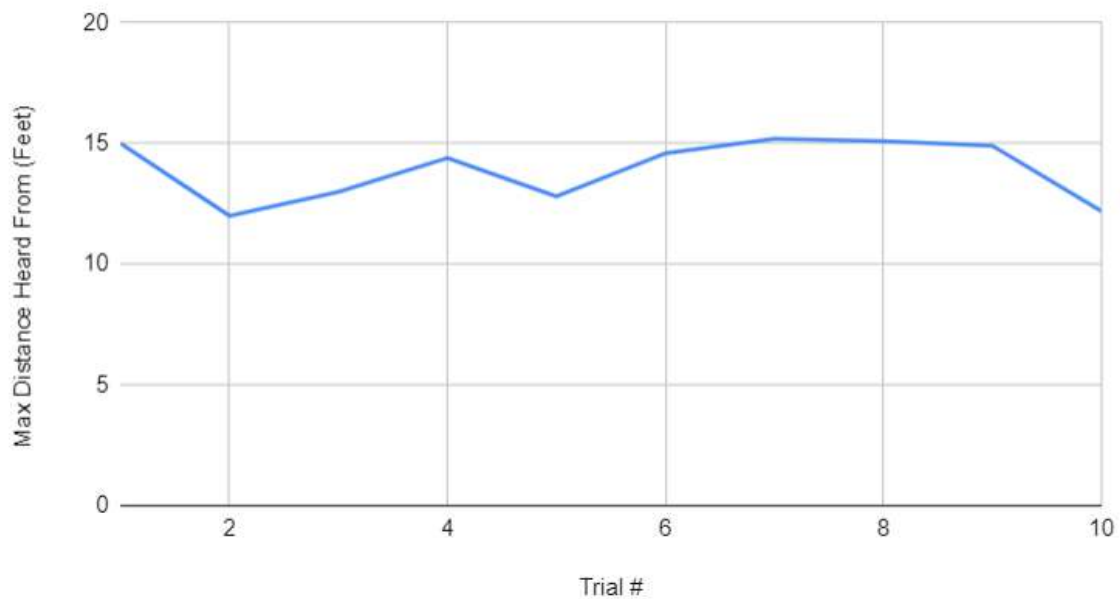
Gas Sensor Test

I lowered the sensitivity to 100 to detect smoke from an incense stick. I did our tests based on that

Distance from sensor (Inches)	Did it Activate?
12	No
10	No
8	Yes
6	Yes

Buzzer Test

Max Sound Distance for Buzzer



I tested the buzzer to see what is the max distance in which the buzzer can be heard from in feet. The distance was not very far, so I added a louder buzzer to our final design.

9V Battery Test (TT Motors)

Time (Minutes)	Battery Status
1	Good (Powering Motors)
2	Good (Powering Motors)
3	Good (Powering Motors)
4	Good (Powering Motors)
5	Good (Powering Motors)
6	Good (Powering Motors)
7	Good (Powering Motors)
8	Good (Powering Motors)
9	Good (Powering Motors)
10	Good (Powering Motors)
11	Bad (Not Powering Motors)

The test here tells us that the 9V battery (which powered the motors) only lasted 10 full minutes before dying. It was not a good battery choice because the first responders would have to spend more on buying replacement batteries.

9V Battery Test (Sensors)

Time (Hours)	Battery Status
1	Good (Powering Motors)
6	Good (Powering Motors)
12	Good (Powering Motors)
18	Good (Powering Motors)
24	Good (Powering Motors)
30	Good (Powering Motors)
36	Good (Powering Motors)
42	Good (Powering Motors)
48	Good (Powering Motors)
54	Good (Powering Motors)
60	Bad (Not Powering Motors)

The test here tells us that the 9V battery (which powered the sensors) only lasted 60 full hours before dying. It was a good solution for the client because it powered the sensors for a good amount of time before dying.

Durability Test (Acrylic)

To test the durability of my model, I dropped a scape of aluminum that weighed approximately 5 pounds directly onto the car from a height of 10 feet. These photos show the results.

- Initial Design Acrylic Piece: The wheels came off (inexpensive repair) the chassis cracked (end of useful life of the robot)



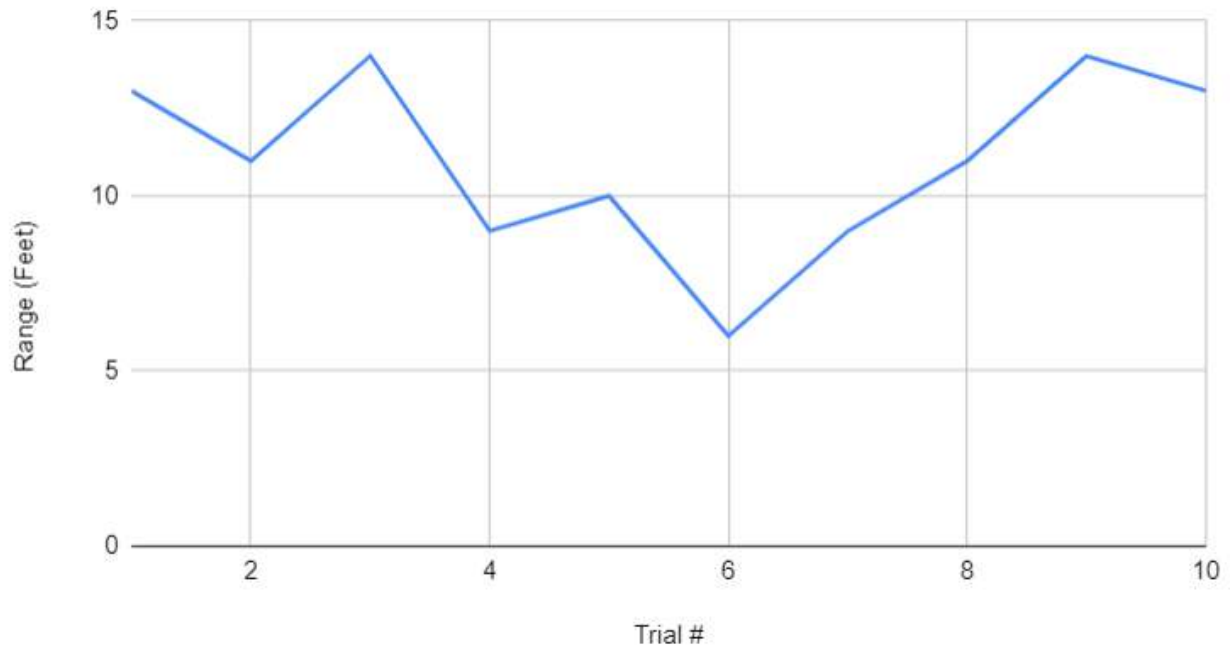
- Final Design Acrylic Piece: The $\frac{1}{4}$ " piece of acrylic got a small crack from the metal scrap.



The final design acrylic piece was more durable than the initial design acrylic piece. They only made a very small crack, unlike the initial design which shattered into pieces. The final design will do a good job protecting the Arduino components.

PIR Range Test

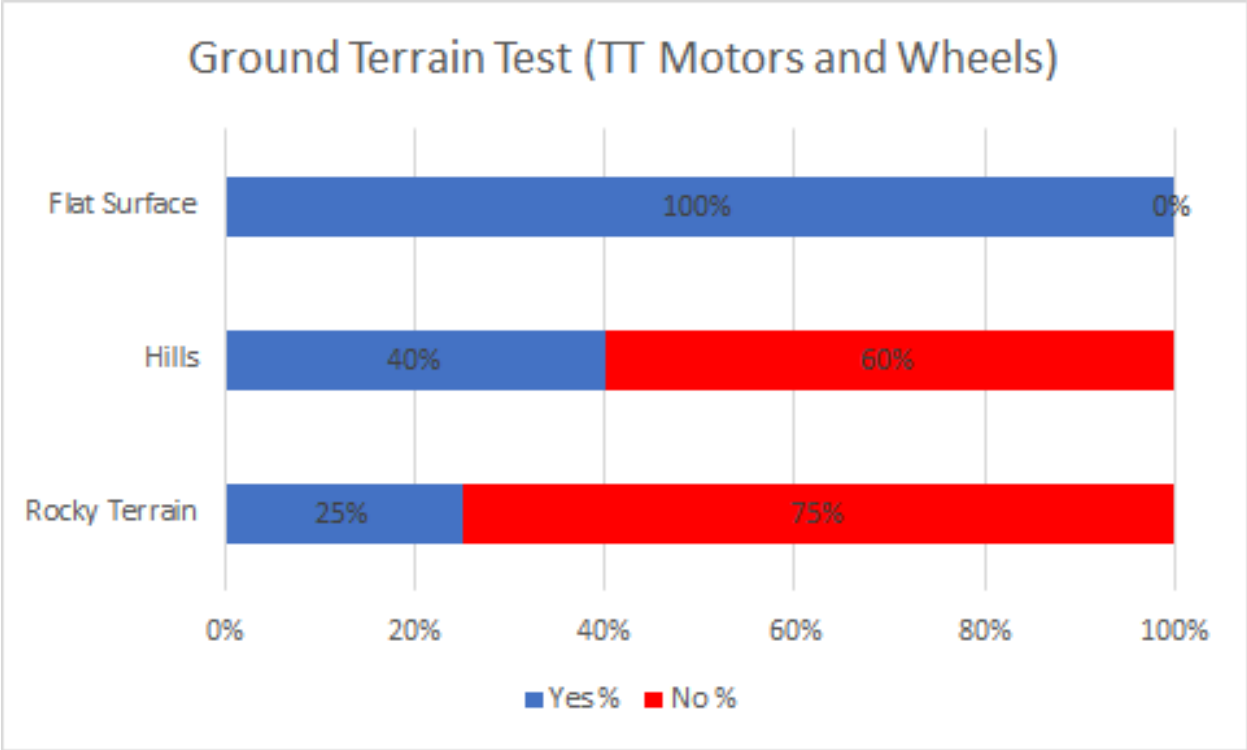
PIR Sensor Human Detection Range



The x-axis is the trial number and the y-axis is the range at which it is activated in feet.

The PIR sensor covered a good amount of area for detecting humans. The only issue, though, is that it does not detect people from the back. That is why I put a PIR sensor in the back for the final design.

Ground Terrain Test (TT Motors and Wheels)



The graph shows the chance (%) that the TT motors and wheels will work on that terrain. The bar shaded in blue symbolizes the percentage that it will work, and the red bar symbolizes the percentage that it won't work

Type of Terrain Being Covered	Did It Successfully Function In That Terrain?
Rock Terrain	No
Hills	No
Flat Surface	Yes

The TT motors and wheels are not a good choice for the car because it barely covers any terrains that the first responders will experience. The only terrain that it covered was a flat cement surface.

Assessment of the Product

When we completed the prototype, we conducted several tests to see the effectiveness of our robot. I tested the gas sensors, the durability of the robot, and PIR sensitivity. According to “How MQ2 Gas/Smoke Sensor Works? And Interface it with Arduino,” the gas sensor “...can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrations anywhere from 200 to 10000ppm.” With this information, I assessed the distance needed to set off the PIR sensor, as seen in the appendix. I increased the sensitivity to 300 to make sure it can detect lower amounts of toxic gas.

We also tested how hard it would be to destroy the robot's chassis when we conducted this experiment. It was made of a weak material that broke easily. After I tested it, we realized it wasn't strong enough for its purpose, so we replaced the chassis material with a harder acrylic (1/4" thick).

The PIR sensor, a digital sensor, was also tested to see its range (which we estimated to be around 9 meters after I did 10 trials). A variety of other tests were conducted to make sure that our search and rescue robot will be as effective as possible.

Modifications and Improvements (Final Design):

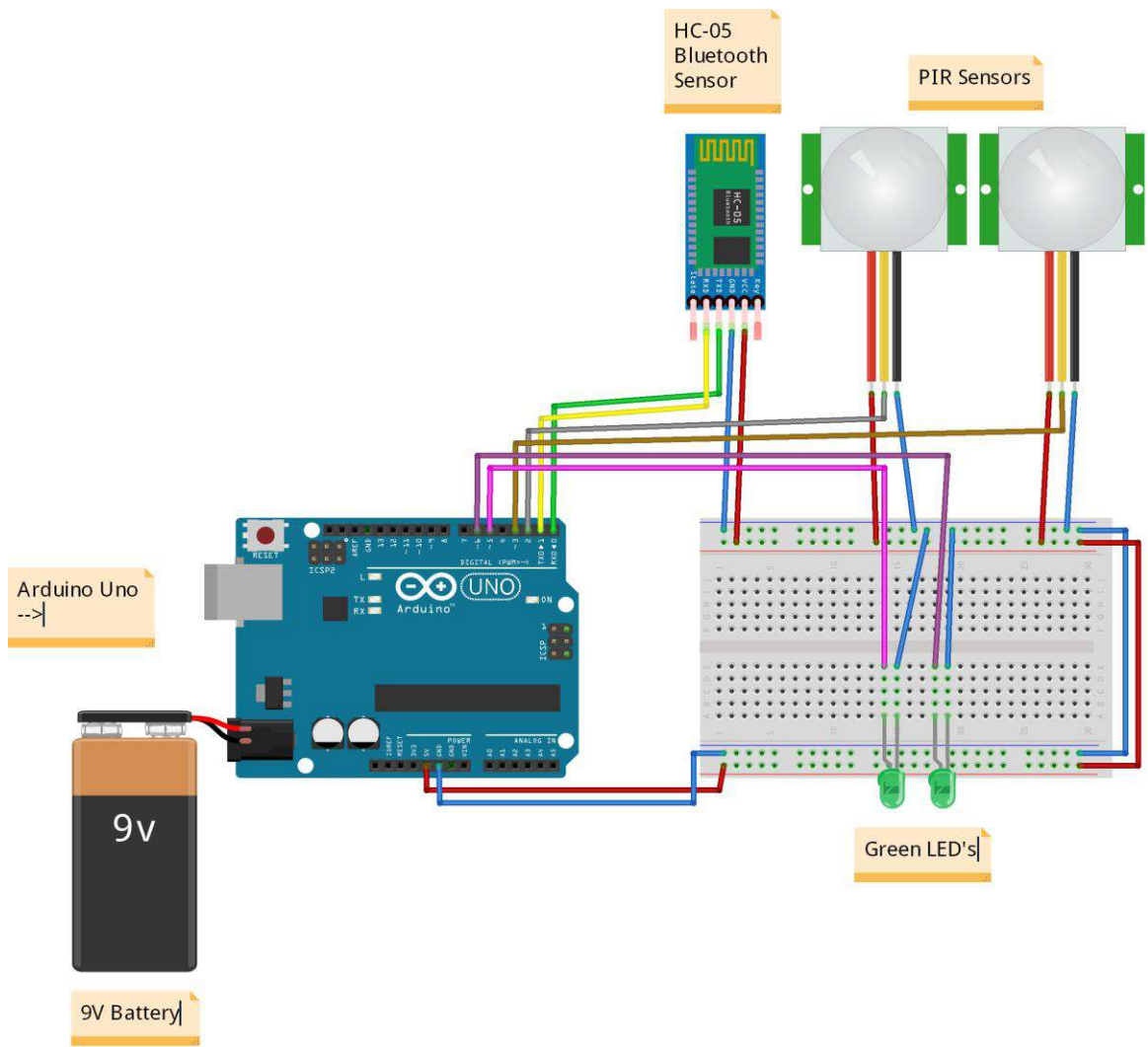
- Include three Arduino Uno's. One will be for the motors (AU1), the other will be for the PIR sensor (AU2), and the last one (AU3) will be for the MQ-2 sensor. This will allow the Arduino sensors and motors to run smoothly without any issues.
- Include thicker acrylic (1/4" thick) to protect the components from getting damaged. The acrylic should enclose the project to protect it from falling items
- Include a louder piezo buzzer, so that people from far can hear the alarm for when dangerous smoke or gas has been detected.
- Include a PIR sensor in the back, so that if a robot passes by the human, and it isn't seeing the person with the front PIR sensor, the back one can cover it. Also, more land will be covered.
- Include VEX 393 motors, since they are stronger than the TT motors.
- Include VEX sprockets and tread links to give the car a tank-like traction, while giving it better mobility. It will be able to run smoother on more dangerous and rough terrains.
- Include 1 more LED of each color. 1 green LED and 1 red LED will be in the front of the car and 1 green LED and 1 red LED will be in the back so that victims can have a better chance of seeing the signs, if they did not see them from the front or back of the car.

- Include rechargeable batteries for the Arduino in charge of the motors because the batteries die in less than 20 minutes. The other Arduino have regular batteries that can last for a few days before dying completely.

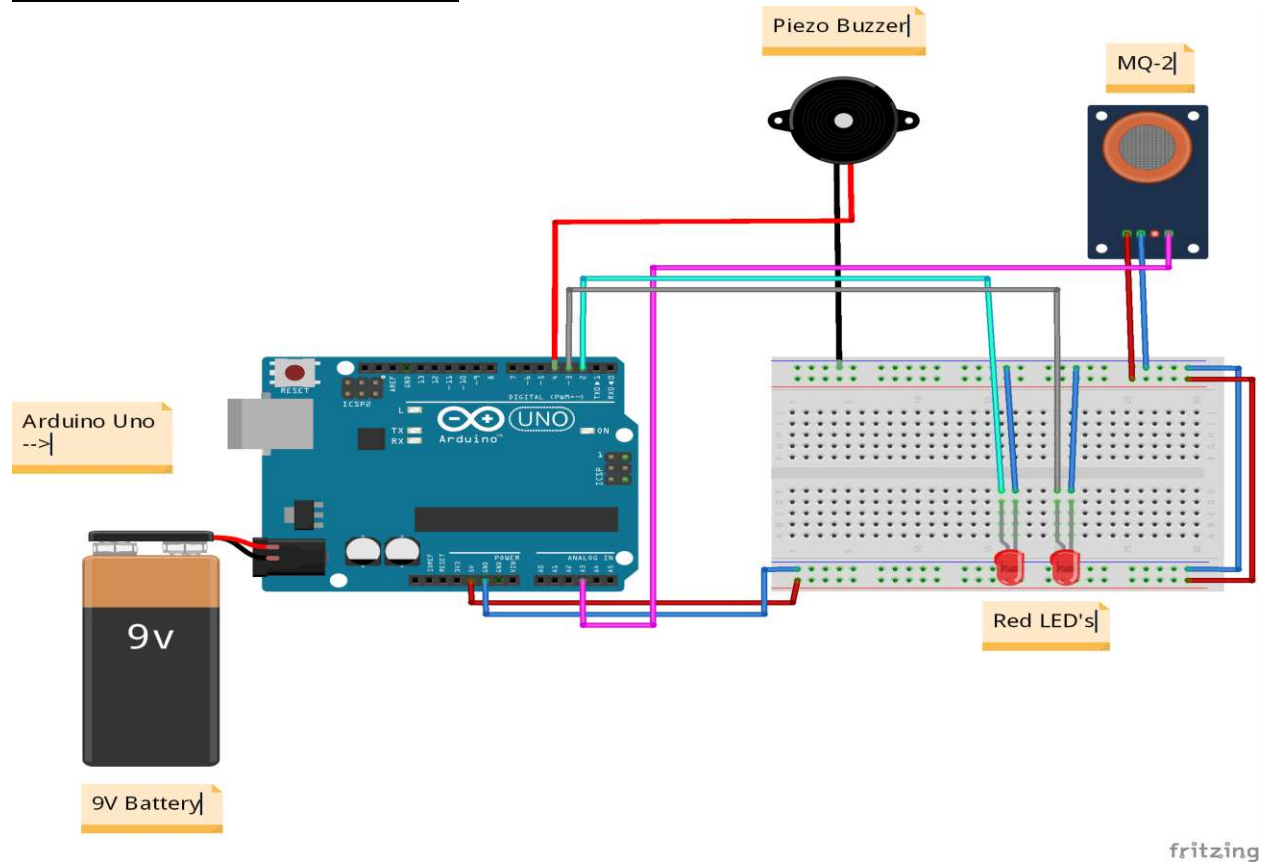
Final Design Sketches:

***We used an Arduino simulator online to make sure that the wiring we did will work fine on the final design once we build it. That is how we got the sketches done.

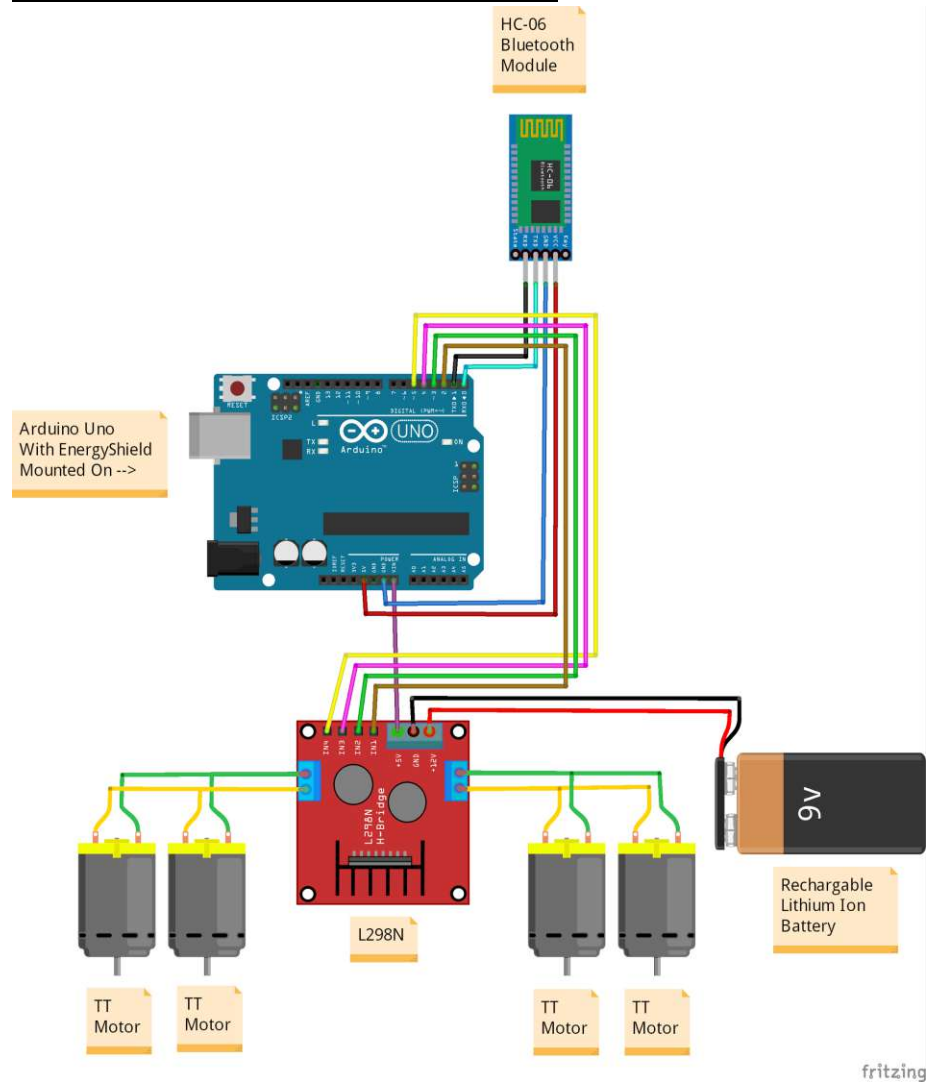
PIR Sensor, HC-06, and LED's



Buzzer, Smoke Sensor, and LED's



Vex Motors and HC-05 Bluetooth Module



Final Design Code

Code 1- Wheels with Bluetooth Module:

```
char t; // The HC-06 will store a character value

void setup() {
  pinMode(2, OUTPUT); // Left motors forward (Pin 2) is output
  pinMode(3, OUTPUT); // Left motors reverse (Pin 3) is output
  pinMode(4, OUTPUT); // Right motors forward (Pin 4) is output
  pinMode(5, OUTPUT); // Right motors reverse (Pin 5) is output

  Serial.begin(9600); // Get the Arduino ready to communicate with the serial monitor
  // At a rate of 9600 bits per seconds

}

void loop() {
  if (Serial.available()) { // If serial monitor readings are available regarding the motors
    t = Serial.read(); // The HC-06 will read the readings
    Serial.println(t); // The Hc-06 will print the readings on the Arduino Bluetooth Control
  }

  if (t == '1') { // Move forward (all motors rotate in forward direction)
    digitalWrite(2, HIGH); // Left motor will be going forward
    digitalWrite(3, LOW); // Left motors will not go reverse
    digitalWrite(4, HIGH); // Right motors will be going forward
    digitalWrite(5, LOW); // Right motors will not go reverse
```

```
}
```

```
else if (t == '2') { // Move reverse (all motors rotate in reverse direction)
```

```
    digitalWrite(2, LOW); // Left motors will not be going forward
```

```
    digitalWrite(3, HIGH); // Left motors will be going reverse
```

```
    digitalWrite(4, LOW); // Right motors will not be going forward
```

```
    digitalWrite(5, HIGH); // Right motors will be going reverse
```

```
}
```

```
else if (t == '3') { // Turn right (left side motors rotate in forward direction, right side motors  
doesn't rotate)
```

```
    digitalWrite(2, LOW); // Left motors will not be going forward
```

```
    digitalWrite(3, LOW); // Left motors will not be going reverse
```

```
    digitalWrite(4, HIGH); // Right motors will be going forward
```

```
    digitalWrite(5, LOW); // Right motors will not be going reverse
```

```
}
```

```
else if (t == '4') { // Turn left (right side motors rotate in forward direction, left side motors  
doesn't rotate)
```

```
    digitalWrite(2, HIGH); // Left motors will be going forward
```

```
    digitalWrite(3, LOW); // Left motors will not be going reverse
```

```
    digitalWrite(4, LOW); // Right motors will not be going forward
```

```
    digitalWrite(5, LOW); // Right motors will not be going reverse
```

```
}
```

```
else if (t == '5') { // Stop (all motors stop)
```

```
    digitalWrite(2, LOW); // Left motors will not be going forward
```

```
    digitalWrite(3, LOW); // Left motors will not be going reverse
```

```
    digitalWrite(4, LOW); // Right motors will not be going forward
```

```
    digitalWrite(5, LOW); // Right motors will not be going reverse
}
delay(100); // Wait 0.1 seconds before repeating the code from void loop again
}
```

Code 2- PIR Sensors, Green LED's, and Bluetooth Module:

```
#include <SoftwareSerial.h> // Library used for serial communication with a digital pin other
than that of the serial port

SoftwareSerial mySerial(0, 1); // RX, TX pins for HC-05 module

//PIR Sensor

long unsigned int lowIn1; //the time when the sensor outputs a low impulse
long unsigned int lowIn2; //the time when the sensor outputs a low impulse
long unsigned int pause = 5000; //the amount of milliseconds the sensor has to be low before we
    // Assume all motion has stopped

int pirPin1 = 2; // The digital pin connected to the PIR sensor 1's output
int pirPin2 = 3; // The digital pin connected to the PIR sensor 1's output

boolean lockLow1 = true;
boolean takeLowTime1;
boolean lockLow2 = true;
boolean takeLowTime2;

int greenLed1 = 5; // The digital pin connected to the green LED
int greenLed2 = 6; // The digital pin connected to the green LED

void setup() {
```



```
pinMode(greenLed1, OUTPUT); // Set the green LED as output
pinMode(greenLed2, OUTPUT); // Set the green LED as output
```

```
mySerial.begin(9600); //Send the MQ-2 readings to the HC-05 serial monitor found on the
Arduino IDE
```

```
pinMode(pirPin1, INPUT); // Set the PIR sensor as input
pinMode(pirPin2, INPUT); // Set the PIR sensor as input
```

```
}
```

```
void loop() {
```

```
if (digitalRead(pirPin1) == HIGH) { // If the readings of the PIR sensor is high
```

```
if (lockLow1) {
```

```
    lockLow1 = false; // Human motion was detected
```

```
    mySerial.println("Living Thing Found! (PIR 1)"); // The HC-05 serial monitor will print
```

```
                // Out that a human was found by the PIR sensor
```

```
    digitalWrite(greenLed1, HIGH); // The green LED will turn on
```

```
    digitalWrite(greenLed2, HIGH); // The green LED will turn on
```

```
    delay(1000); // Wait 0.2 seconds before moving on to the next line of code
```

```
}
```

```
takeLowTime1 = true; // Make sure this is only done at the start of a HIGH phase
```

```
}
```

```
if (digitalRead(pirPin1) == LOW) {
```

```
if (takeLowTime1) {
```

```
    lowIn1 = millis(); // Save the time of the transition from high to LOW
```

```

    takeLowTime1 = false; // Make sure this is only done at the start of a LOW phase
}

if (!lockLow1 && millis() - lowIn1 > pause) { // If the sensor is low for more than the given
pause
    lockLow1 = true; // Assume that no more motion is going to happen
    mySerial.println("Living Thing Lost (PIR 1)"); // The HC-05 serial monitor will print
        // Out that a human was lost according to the PIR sensor
    digitalWrite(greenLed1, LOW); // The green LED will turn off
    digitalWrite(greenLed2, LOW); // The green LED will turn off

    delay(1000); // Wait 0.2 seconds before moving on to the next line of code
}

if (digitalRead(pirPin2) == HIGH) { // If the readings of the PIR sensor is high
if (lockLow2) {
    lockLow2 = false; // Human motion was detected
    mySerial.println("Living Thing Found! (PIR 2)"); // The HC-05 serial monitor will print
        // Out that a human was found by the PIR sensor
    digitalWrite(greenLed1, HIGH); // The green LED will turn on
    digitalWrite(greenLed2, HIGH); // The green LED will turn on

    delay(1000); // Wait 0.2 seconds before moving on to the next line of code
}

takeLowTime2 = true; // Make sure this is only done at the start of a HIGH phase
}

if (digitalRead(pirPin2) == LOW) {

```

```

if (takeLowTime2) {
    lowIn2 = millis(); // Save the time of the transition from high to LOW
    takeLowTime2 = false; // Make sure this is only done at the start of a LOW phase
}

if (!lockLow2 && millis() - lowIn2 > pause) { // If the sensor is low for more than the given
pause
    lockLow2 = true; // Assume that no more motion is going to happen

    mySerial.println("Living Thing Lost (PIR 2)"); // The HC-05 serial monitor will print
                                                // Out that a human was lost according to the PIR sensor
    digitalWrite(greenLed1, LOW); // The green LED will turn off
    digitalWrite(greenLed2, LOW); // The green LED will turn off

    delay(1000); // Wait 0.2 seconds before moving on to the next line of code
}
}
}
}
}

```

Code 3- Smoke Sensor and Red LED's

```

//Smoke Sensor

int redLed1 = 2; // The digital pin connected to the red LED
int redLed2 = 3; // The digital pin connected to the red LED
int buzzer = 4; // The digital pin connected to the buzzer
int smokeA0 = A3; // The analog pin connected to the MQ-2 sensor
int sensorThres = 200; // The minimum value that the MQ-2 sensor needs to detect smoke and/or
harmful gases

void setup() {

```

```

pinMode(redLed1, OUTPUT); // Set the red LED as output
pinMode(redLed2, OUTPUT); // Set the red LED as output
pinMode(buzzer, OUTPUT); // Set the buzzer as output
pinMode(smokeA0, INPUT); // Set the MQ-2 sensor as input

}

void loop() {
int analogSensor = analogRead(smokeA0);

    Serial.println(analogSensor); // Print the MQ-2 Sensor's readings onto the serial monitor
    if (analogSensor > sensorThres) // If the MQ-2's readings of the air is greater than the
sensorThres

        // (200), then there is smoke and/or dangerous gas in the air.
    {
        digitalWrite(redLed1, HIGH); // The red LED will turn on
        digitalWrite(redLed2, HIGH); // The red LED will turn on

        tone(buzzer, 800, 800); // The buzzer will make a sound from tone 800 and 800
        delay(200); // Wait 0.2 seconds before moving on to the next line of code
        tone(buzzer, 600, 800); // The buzzer will make a sound from tone 600 and 800
        delay(200); // Wait 0.2 seconds before moving on to the next line of code

    }
    else
    {
        digitalWrite(redLed1, LOW); // The red LED will be off; it will not turn on
        digitalWrite(redLed2, LOW); // The red LED will be off; it will not turn on
        noTone(buzzer); // The buzzer will give off no tone
    }
}

```

```

}

delay(100); // Wait 0.1 seconds before moving on to the next line of code. In this case,

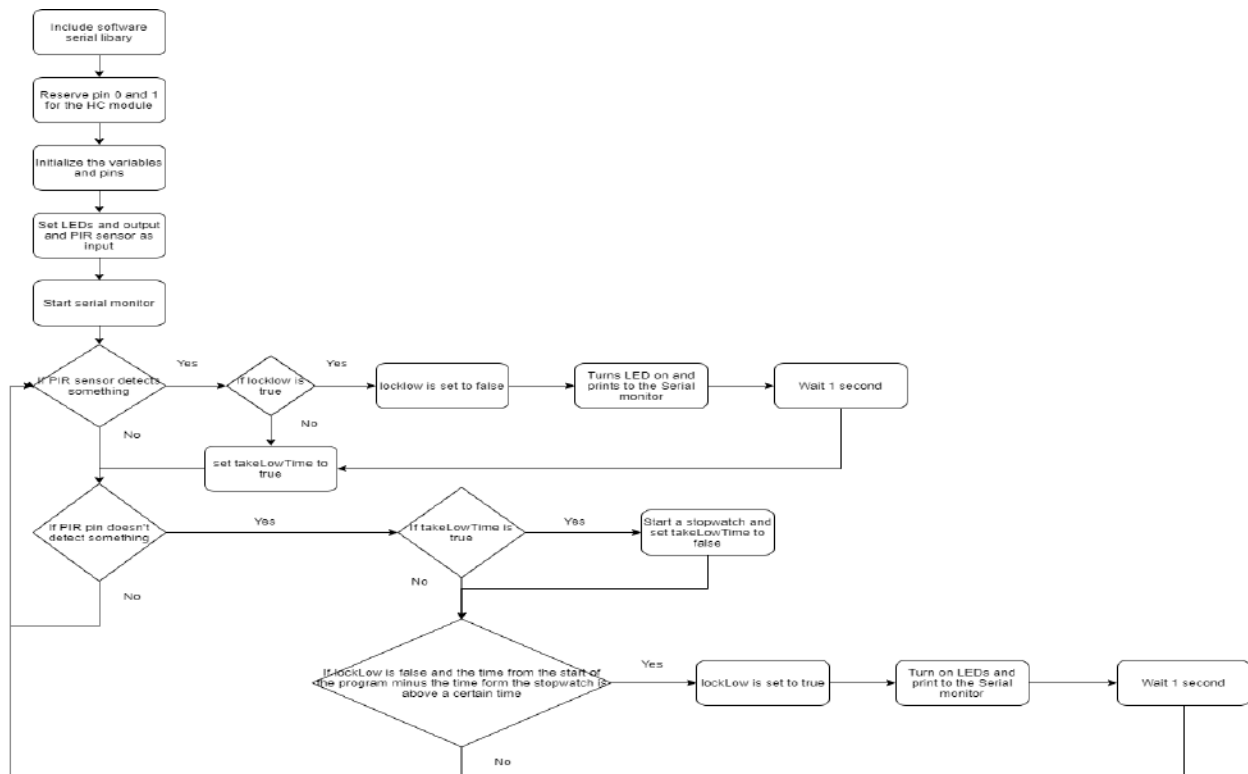
//it will repeat the void loop again

}

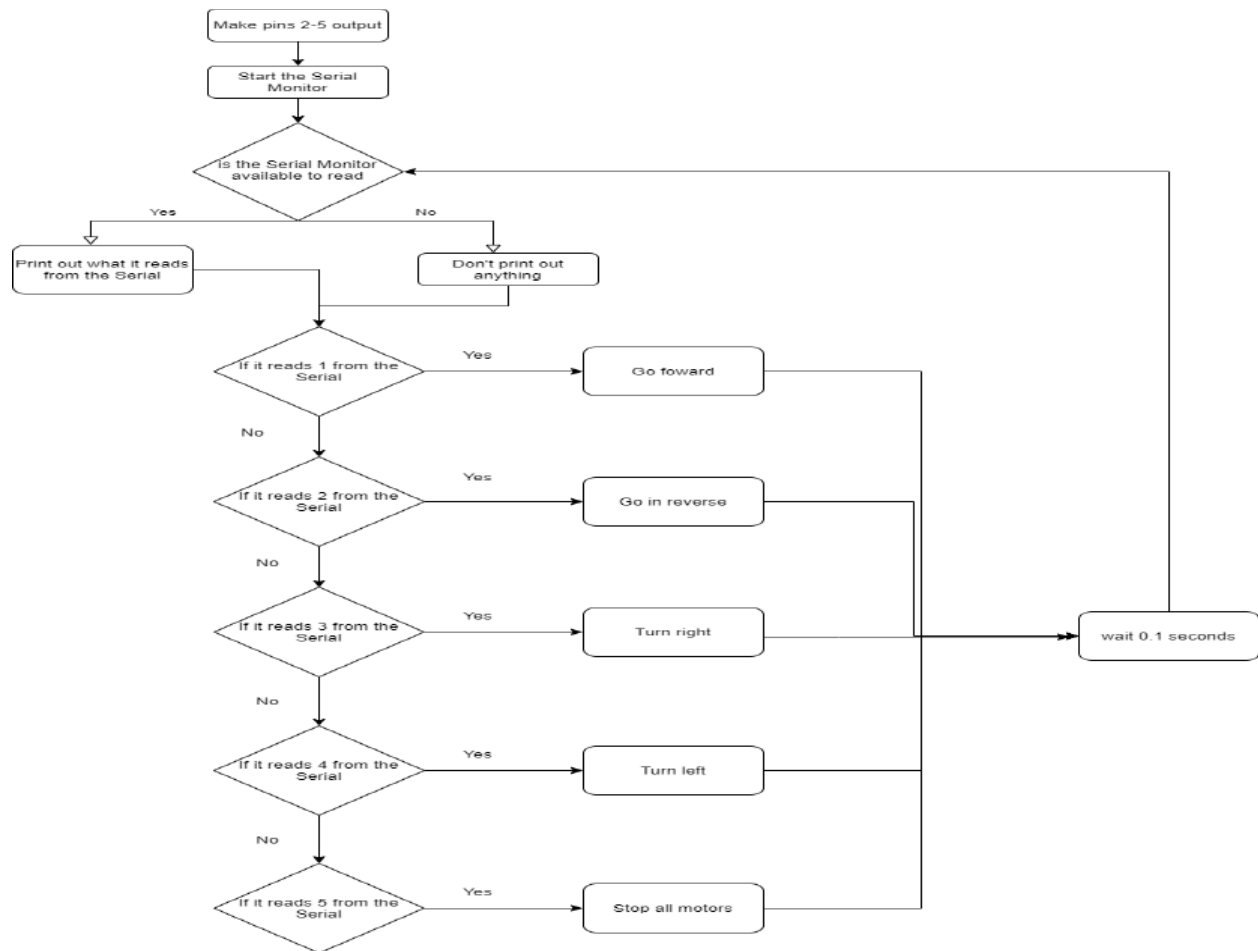
```

Final Design Block Logic Diagram

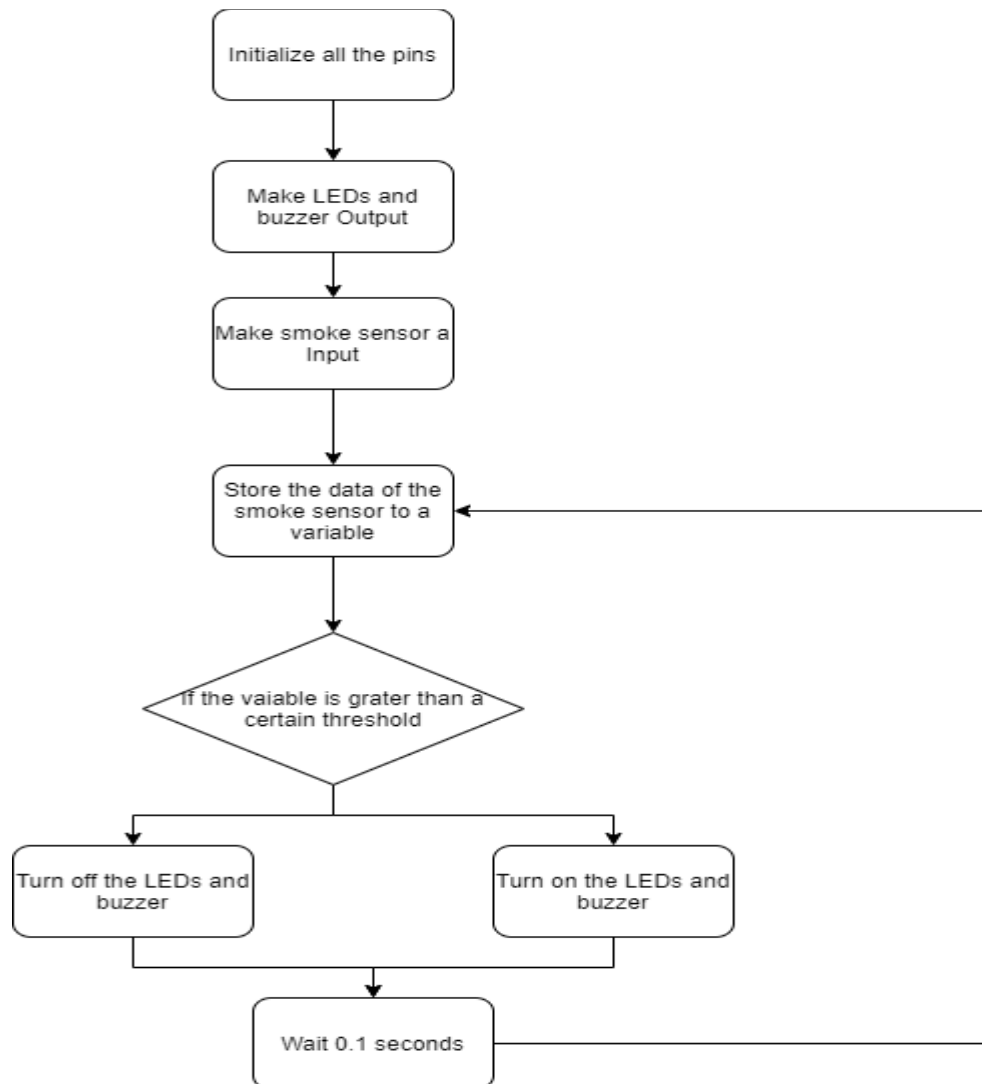
PIR Sensor with Bluetooth Module and Green LED's



Vex Motors with Bluetooth Module



Smoke Sensor, Buzzer, and Red LED's



Final Design PseudoCode:

PIR Sensor with Bluetooth Module and Green LED's


```

Include the software serial library
Reserve pin 0 and 1 for the HC-06 module
Initialize variable
Assign some variables a pin
Setup:
    Make the green LED pins as outputs
    Start the serial monitor
    Make the PIR pins as output
Loop:
    If PIR pin detects something
    Set locklow variable to false
    Print to the serial monitor
    Turn on the green LED's
    Wait 1 second
    Make takelowtime true
    If PIR pin does not detect something
    If takelowtime is true, start a timer
    If locklow is false and the timer went longer for a specified
time
    Locklow is set to true, a message is sent to the serial
monitor, and the LED's are turned off

```

Smoke Sensor, Buzzer, and Red LED's

```

Initialize all the pins
Setup:
    Make the LED's and buzzer output and the smoke sensor input
Loop:
    Store the data from the smoke sensor to a variable
    If the signal from the sensor passes a certain threshold
    Turn on the LED's and activate the buzzer
    If not turn off the LED's and deactivate the buzzer
    Wait 0.1 seconds

```

Vex Motors with Bluetooth Module

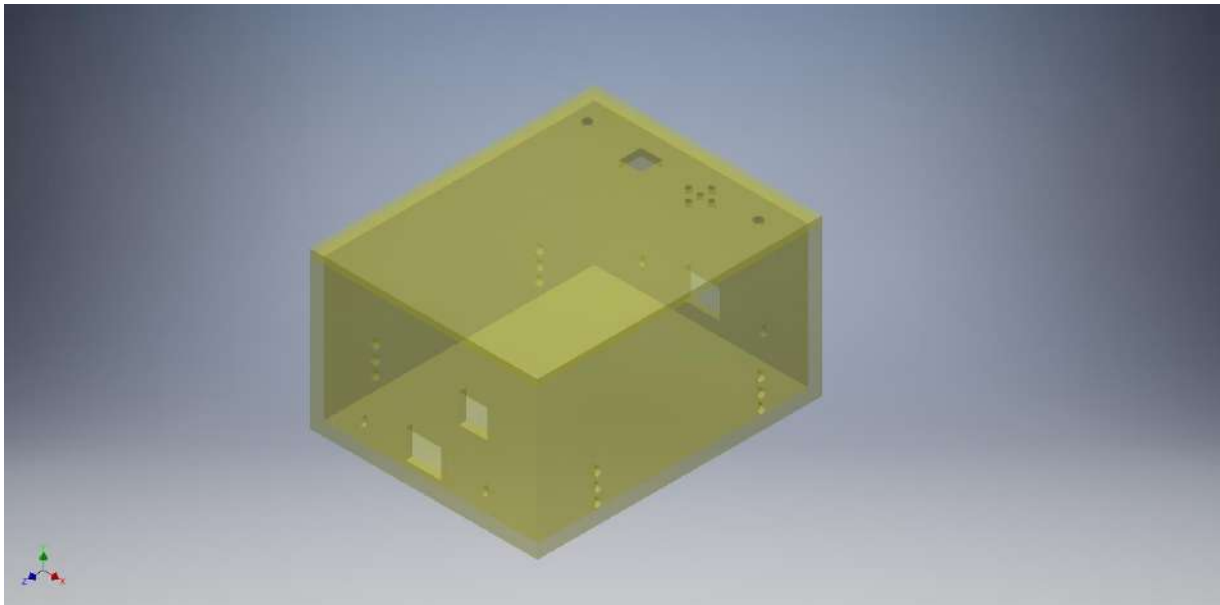
```

• • •
Setup:
  Make pin 2-5 outputs
  Start the serial monitor
Loop:
  If the serial monitor is available read and store what is in the
  serial monitor
  If it reads 1 make all motors go forward
  If it reads 2 make all motors go in reverse
  If it reads 3 make the robot turn right
  If it reads 4 make the robot turn left
  If it reads 5 stop all motors
  Wait 0.1 seconds

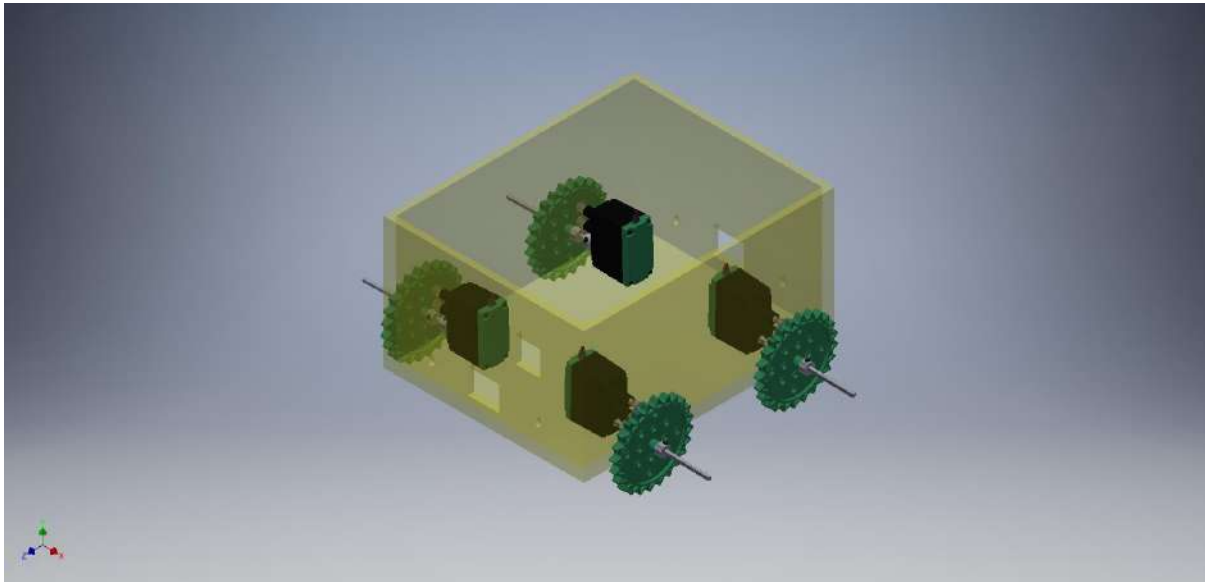
```

Proposed Final Design Model:

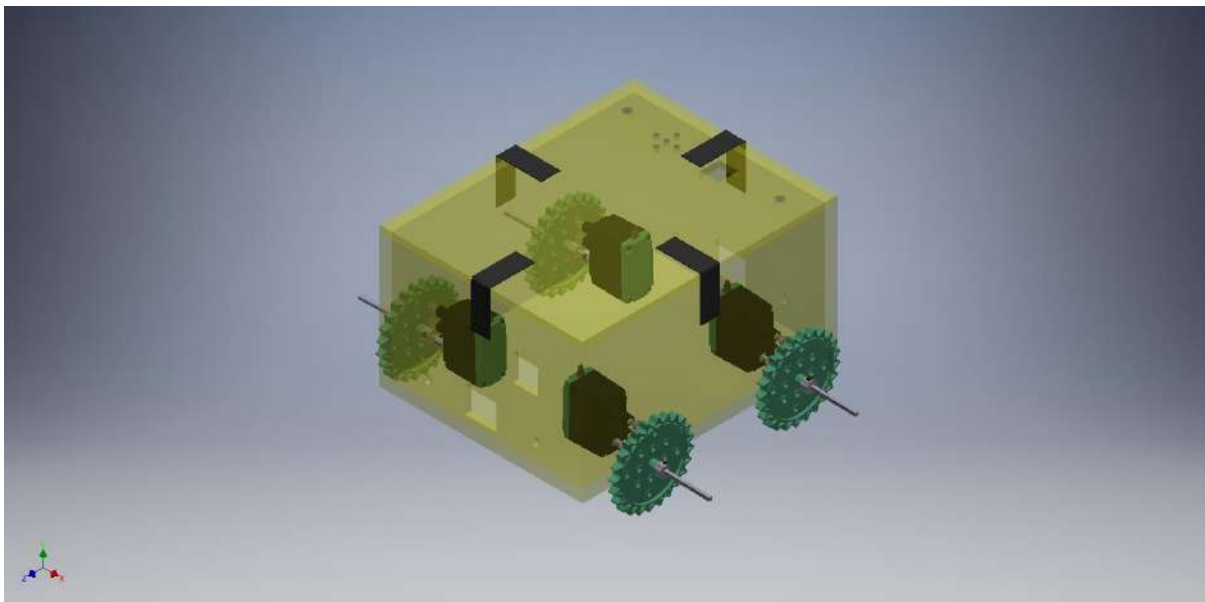
Acrylic Base with Top:



Acrylic Base (Not Including Top) With Vex Motors, Gears, and Axles:



Acrylic Base (Including Top) With Vex Motors, Gears, Axles, and Velcro:



Proposed Final Design Budget Sheet:

Part	Unit Dimension s	Retail Price \$	Price Per Unit \$	Quantity Used	Total Cost \$	Retail Source
Arduino Uno R3 (Clone)	3" x 2.1"	\$2.98	\$2.98/ Board	3 Boards	\$8.94	AliExpress.com
Half Breadboard	2.2" x 7"	\$1.65	\$1.65/Breadboard	1 Bread board	\$1.65	AliExpress.com
Acrylic Sheet	¼" x 36" x 10"	\$23.00	\$23.00/ Sheet	1 Sheet	\$23.00	Ebay.com
VEX 2-Wire 393 Motors (Comes With 2 6-32 x ¼" Screws)	2.19" x 1.82" x 0.93"	\$10.00	\$10.00/ Motor With 2 Screws	4 Motors and 8 Screws	\$40.00	The Roadium Open Air Market in Redondo Beach
VEX Axles (4 Pack)	Each Axle: 4"x 0.125"	\$4	\$1/Vex Axle	4 Vex Axles	\$4	The Roadium Open Air Market in Redondo Beach
VEX Shaft Collar	0.433" x 0.260"	\$8.99	About \$0.56/Shaft Collar	12 Shaft Collars	\$6.72	Store.robotmesh.com
VEX 24 Tooth High Strength Sprocket	Approximately 3.16" Outer Diameter	\$14.49	\$3.62/ Sprocket	4 Sprockets	\$14.49	Vexrobotics.com
110 VEX Tank Tread Links	Each Tread Link: 0.69" x 1.5"	\$15.00	About \$0.14/ Tread Link	110 Vex Tread Links	\$15.00	The Roadium Open Air Market in Redondo Beach
Energy Shield 2 Basic-Rechargeable Battery	2.8" x 2.1"	\$29.95	\$29.95/1 Rechargeable Battery	1 Rechargeable Battery	\$29.95	Amazon.com

for Arduino						
Lithium Ion Battery Pack - 3.7V 6600mAh	69mm x 54mm	\$29.50	\$29.50/1 Lithium Ion Battery	1 Lithium Ion Battery	\$29.50	Adafruit.com
HC-06 Bluetooth Module	26.7 mm x 13 mm	\$2.23	\$2.23/Bluetooth Module	1 Bluetooth Module	\$2.33	AliExpress.com
L298N Motor Driver	43 mm x 43 mm	\$1.34	\$1.34/ Motor Driver	1 L298N Motor Driver	\$1.34	AliExpress.com
PIR Sensor with Holder	1.27" x 0.96"	\$1.10	\$1.10/ PIR Sensor With Holder	1 PIR Sensor With Holder	\$1.10	AliExpress.com
100 Piece 5mm LED Assorted Kit	5mm x 31.7 mm/ LED	\$1.55	About \$0.02/ LED	4 LED's (2 Green and 2 Red)	\$0.08	AliExpress.com
Piezo Buzzer	1.18" in diameter 0.43" in depth	\$0.84	\$0.84/ Buzzer	1 Buzzer	\$0.84	AliExpress.com
9V Battery Button Power Cable Buckle Snaps Power Cable Connector	6" x 0.5"	\$1.25 for 10 Battery Buckle Snaps Power Cable Connector	\$0.13/Battery Buckle Snaps Power Cable Connector	2 Battery Buckle Snaps Power Cable Connector	\$0.26	AliExpress.com
Male to Male	20 cm x 0.2 cm	\$0.96 For 65 Male to	\$0.01/ Wire	28 Wires	\$0.28	AliExpress.com

Jumper Wires		Male Wires				
Male to Female Jumper Wires	Assorted Sizes	\$3.36 for 120 Wires	\$0.03/Wire	25 Male to Female Wires	\$0.75	AliExpress.com
9V Battery	4.5" x 2"	\$1.00	\$1.00/Battery	2 9V Battery	\$2.00	City Fashion Center in Lynwood California
2 Pin Wire Connectors	20mm x 17.5mm x 13.5mm	\$1.96/20 pcs	About \$0.10/Wire Connector	18 Wire Connectors	\$1.80	Aliexpress.com
Electric Tape	¾" x 20 yds	\$2.29	About \$0.04/Foot of Tape	4 Inches of Tape	About \$0.01	Ekono 99 Cents Store
Velcro	32" x 1"	\$6.65	About \$0.21 Per Inch of Velcro	32 Inches of Velcro	\$6.65	Ekono 99 Cents Store
Total Cost					\$190.69	

Bibliography:

- “American Red Cross.” American Red Cross, www.redcross.org/. “Arduino - PIR Sensor.”
- Tutorialspoint, - www.tutorialspoint.com/arduino/arduino_pir_sensor.htm. Bolles, Bob. “Weight Distribution - Analyzing Your Setup For Faster Lap Times - Circle Track Magazine.” Hot Rod, Hotrod, 14 June 2016, www.hotrod.com/articles/ctrp-0706-weight-distribution-transfer/.
- Doroftei, Daniela, et al. “Introduction to the Use of Robotic Tools for Search and Rescue.” IntechOpen, IntechOpen, 23 Aug. 2017, www.intechopen.com/books/search-and-rescue-robotics-from-theory-to-practice/introduction-to-the-use-of-robotic-tools-for-search-and-rescue.
- Last Minute Engineers. “In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface It with Arduino.” Last Minute Engineers, Last Minute Engineers, 13 Dec. 2019, lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/.
- Soffar, Heba. “Rescue Robots (Mobile Robots) Types, Uses, Advantages and Disadvantages.” Science Online, 29 Aug. 2019, www.online-sciences.com/robotics/rescue-robots-mobile-robots-types-uses-advantages-and-disadvantages/.
- Walker, Jon. “Search and Rescue Robots – Current Applications on Land, Sea, and Air.” Emerj, Emerj, 21 Nov. 2019, emerj.com/ai-sector-overviews/search-and-rescue-robots-current-applications/.
- “Wheels vs Tracks: Advantages and Disadvantages.” Lite Trax, 1 Mar. 2018, litetrax.com/wheels-vs-tracks-advantages-disadvantages/.
- Hek. “Gas Sensor: MySensors - Create Your Own Connected Home Experience.” MySensors, www.mysensors.org/build/gas.
- RobbeRobbe 3911 gold badge11 silver badge55 bronze badges, et al. “Can I Make Multiple Void Loops with Arduino Uno?” Arduino Stack Exchange, 1 Apr. 1967, arduino.stackexchange.com/questions/37684/can-i-make-multiple-void-loops-with-arduino-uno/51723.
- Drake, Dan. “Autodesk Inventor.” Autodesk Inventor, <https://www.autodesk.com/products/inventor>.