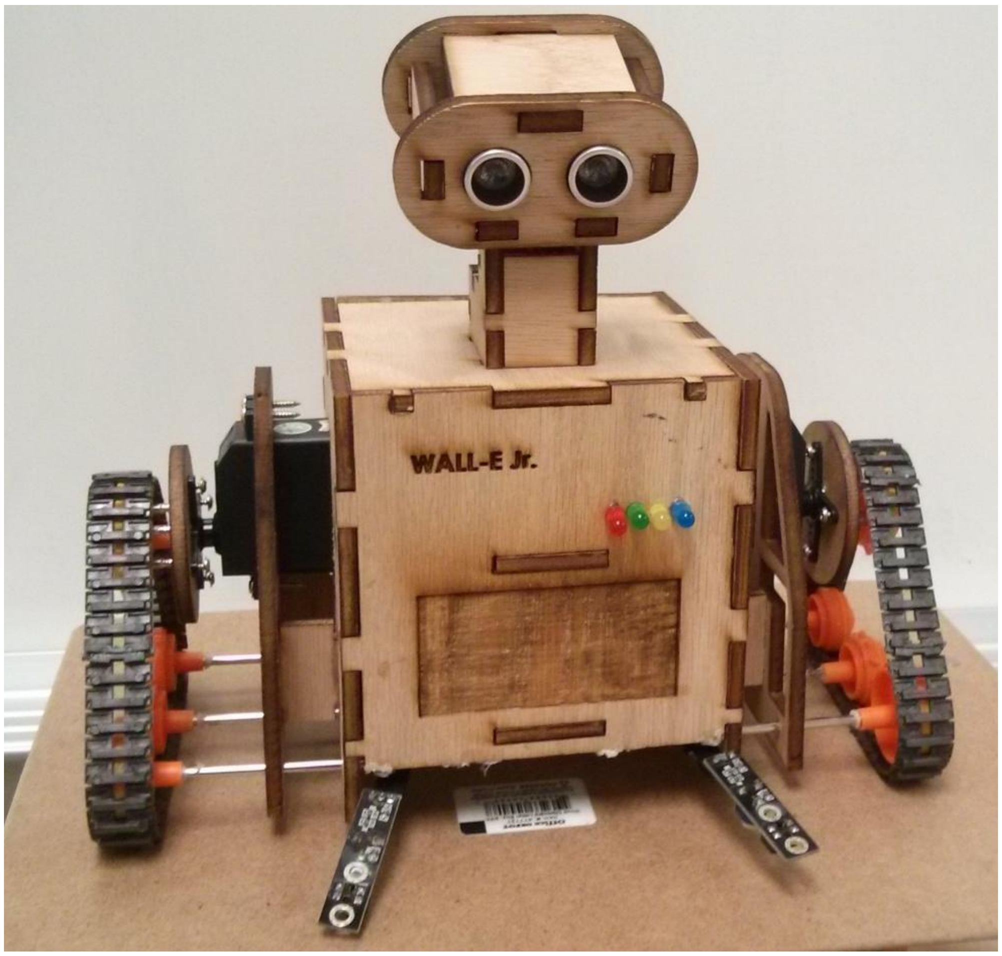
ENME 408 Mechatronics

Final Project Report

Jonathan Shulgach

12/11/15



Introduction:

Line following robots have been used for school workshops and competition robotics events throughout the world for years. These robots can be created from microcontroller parts for educational platforms and STEM programs, such as Arduino workshops and home projects. The Microcontrollers can also be used for advanced programs such as FIRST Lego League, Seaperch, and Robosub competitions. For the ENME 408 class, the assignment was to make a line following robot with a unique feature. For this robot project, the robot was designed to follow a black line on a light surface. The unique element included in the design was an uncanny resemblance to the delightful animated character from the animated Disney motion picture character "Wall-E" as well as extra sensory features. The robot performs well with minimal need for repair, and is very entertaining to observe and play with.

List of Parts:

HC-SR04 Ultrasonic sensor

5mm Red LED

5mm Blue LED

5mm Green LED

5mm Yellow LED

Arduino Uno R3

Tamiya Track and Wheel Set

(2x) HS-422 Servo Motor

400 Tie-points breadboard

(5x) 1100ohm resistor

Piezo buzzer

Robot frame pieces

4xAAbattery case holder

(4x) AA battery

50pcs Female-Male wire pack

50pcs Male-Male wire pack

50pcs Female-Female wire pack

(2x) IR infrared obstacle avoidance sensor module

(8x) 1/2 in. #8 Phillips pan-head screw

(8x) 1/4 in. #6 Phillips pan-head screw

Small Arduino button counter

9V battery

9V battery head connector

Tools:

Philips head screwdriver

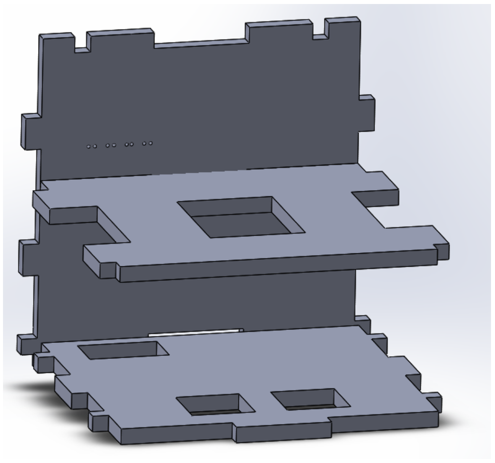
Flathead screwdriver

Wire stripper

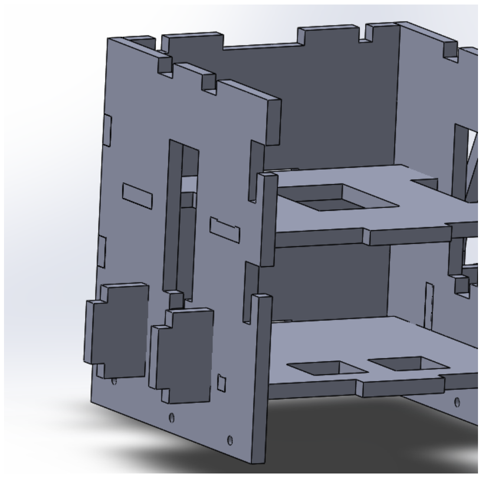
Elmer's glue

Step 1: Assembly

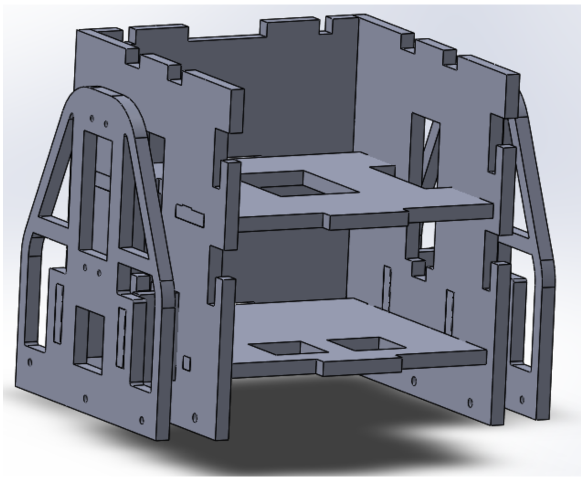
The building instructions for the part diagrams for the robot assembly pieces are shown in steps. The front and inside pieces are glues together, combined with the side box parts and connectors. In Step 2, place the motors inside the large rectangular slots with the side holes outside the box part and the axel motors on the lower side of the slot. The holes can be screwed with the heads facing the side of the motor axel. The screws will be tightened through the wood using the Philips head screwdriver.

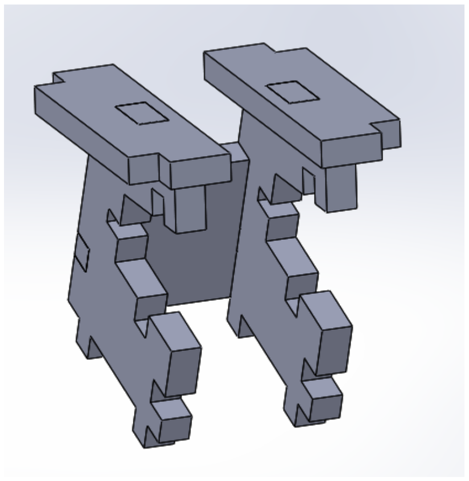


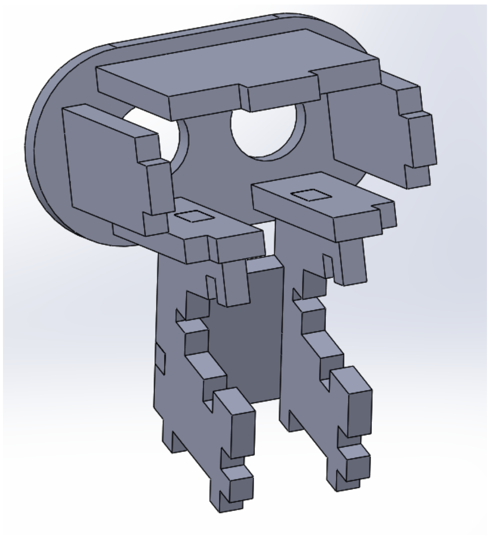
Step 1: Front and inside box parts

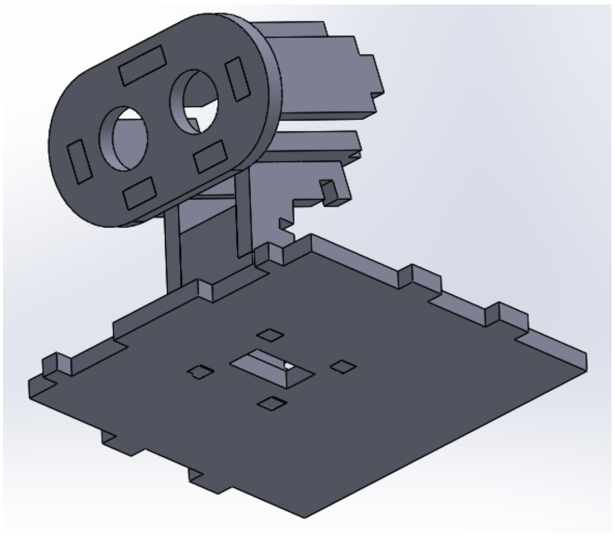


Step 2: Side box and connector pieces


Step 3:Motor mount parts attached


Step 4: Assembly of the neck

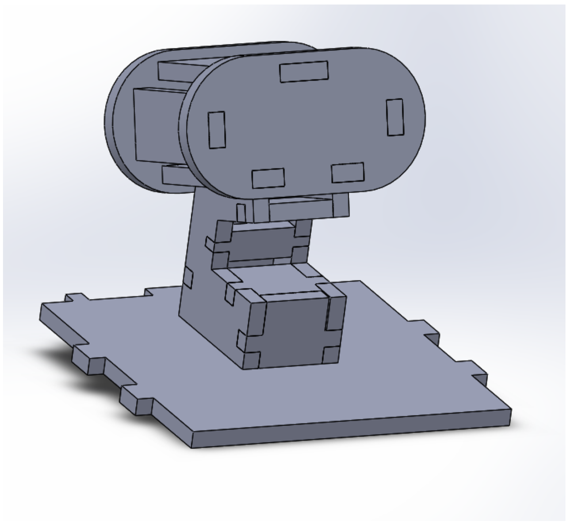


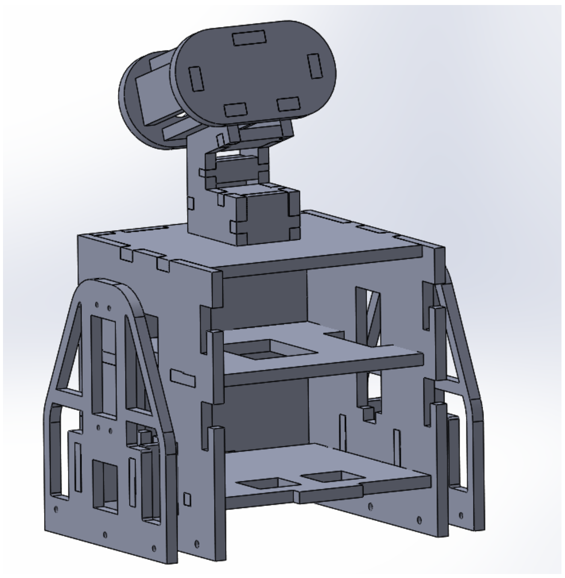


Step 5: Assembly of the head

Step 6: Attachment of top box part to neck

Before the remainder of the head and neck pieces are attached, install the ultrasonic sensor into the eye slot, gluing it into place. Once the two main components are completed, the head component can be attached to the body.


Step 7: Assembly of remainder of head and neck


Step 8: Attachment of both main components

A close up inspection of the wheels and motors show in detail the orientation of the screws and the connections of principle components. The cross piece should be screwed first to the wooden wheel with the 1/4 in screws. Take a black 1/2 in screw to connect the cross piece and wheel to the motor axel. When adding the wheels, place a axel through each of the three hole sets and placing the wheel stopper bearings on with glue to the wheel to prevent them from slipping. The larger wheel is placed on the front axle, the gear wheel is placed in the middle axle, and the big gear wheel is screwed onto the wheel and motor component. The robot runs on a custom tread wheel system, and when the wheels are in place, the tread can be attached to the system.

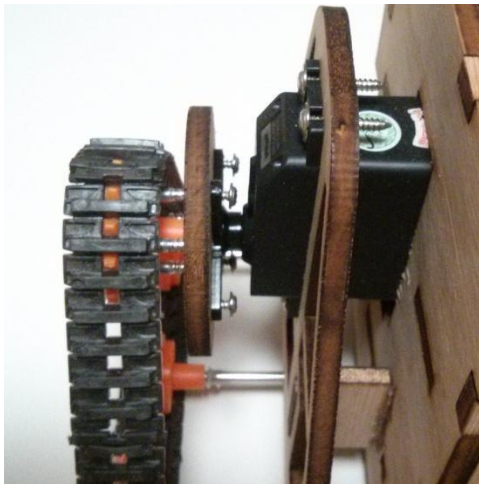


Figure 1: Orientation of screws with parts

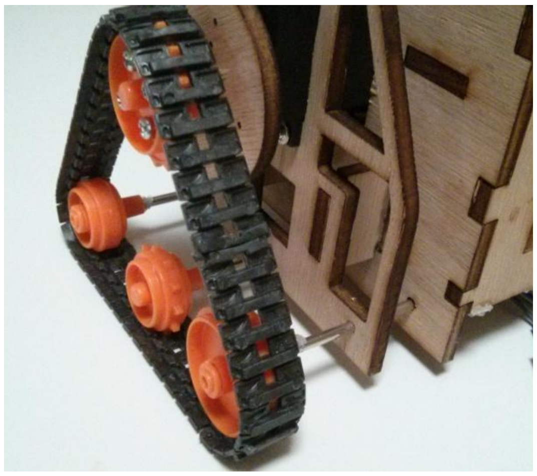
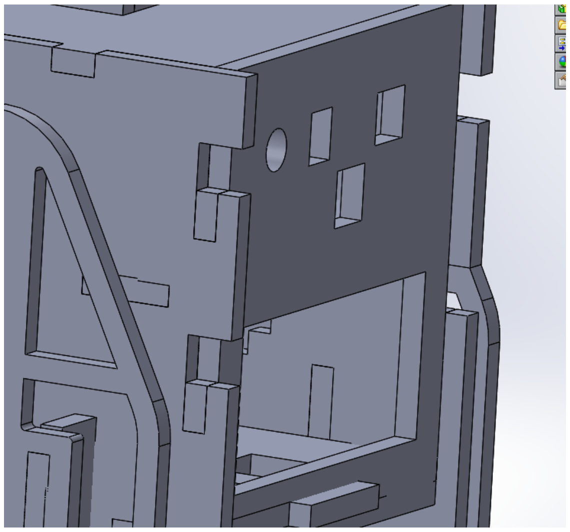
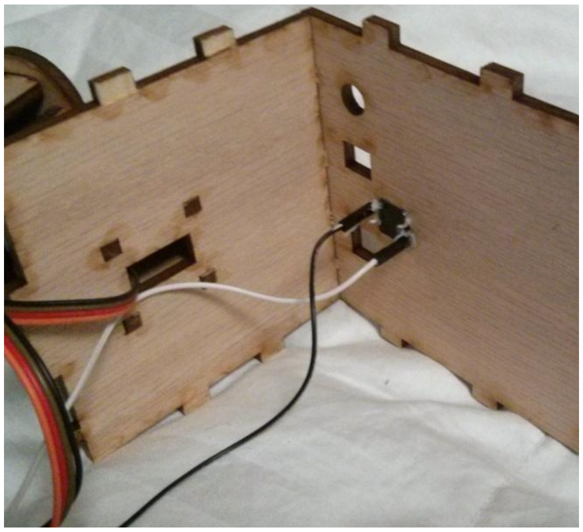


Figure 2: Placement and orientation of wheels

Step 9 can be completed when the electrical components are installed in the robot. The battery case sat in the lower section of the box with the power switch exposed through the rectangular hole near the front. The breadboard sits on top of the battery case, with room for the wires to be funneled through the middle inside box part. The back box component can then be slid into place when the installation of components is finished. Step 10 is the installation of the button and involves gluing the button into the square hole between the three-holed set and the opening along with the opposite corners for the wires. The remaining square holes are to be used for the Arduino connector ports, and the circular hole is for the 9V battery supply cable to the Arduino.



Step 9: Installment of the back box



Step 10: Installation of button with glued wires

The front of the robot contains four indicator LEDs in the order red, green, yellow, and blue from left to right as shown in Figure 3. The LEDs can be glued onto the front part if necessary.

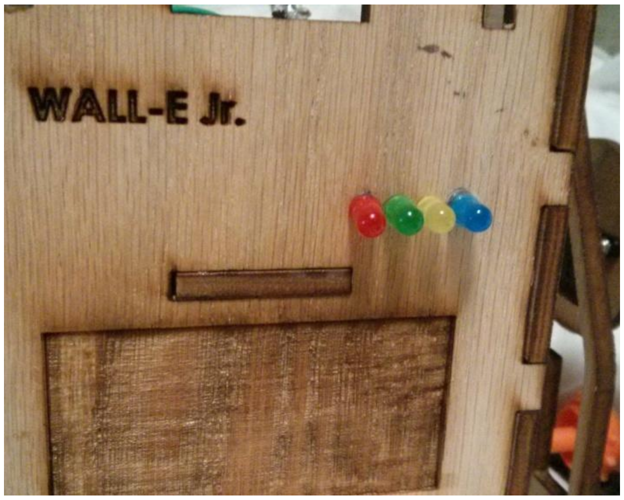


Figure 3: Arrangement of LEDs on front panel

Underneath the robot there are two infrared light diodes and infrared receivers that are funneled through the bottom holes of the robot. The wiring allows for each sensor to be able to be glued to the outside of the robot. The battery case switch is visible and accessible as well. The IR sensors are housed over the front wheel axles and glued to the front panel of the robot facing the ground. The amount of glue that is applied can vary depending on the distance to the ground the user would like the sensors to be at. The sensitivity of the sensors must be changed by using a flat head screwdriver to turn the potentiometers as shown in Figure 4. The potentiometer should be set to the position right before the IR sensor light becomes red.

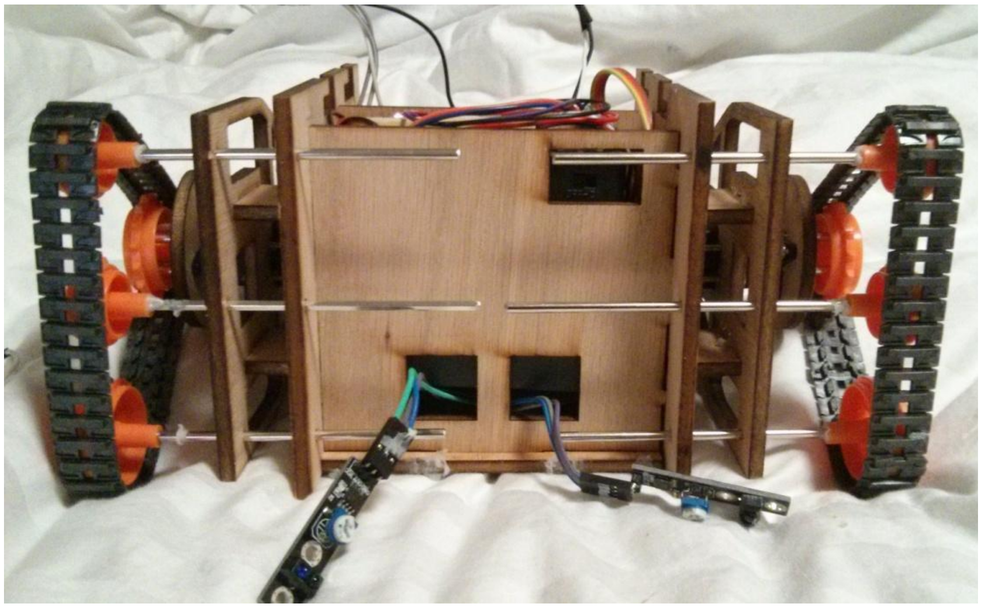


Figure 4: IR sensor placement through side openings

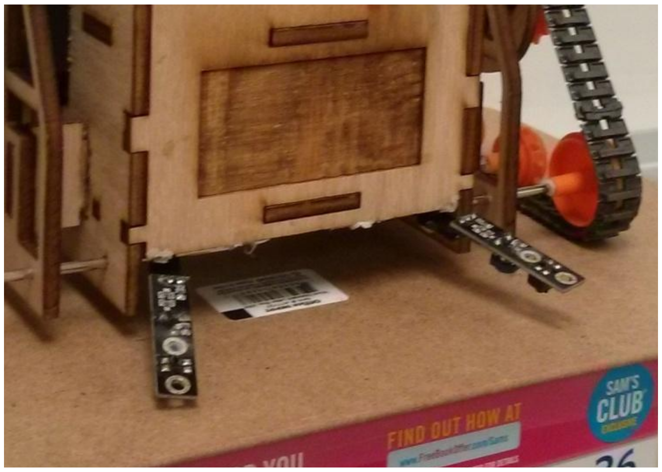


Figure 5: Placement of IR sensors

When assembly of the robot is complete, and the breadboard, battery pack, and Arduino are connected, the wires will be funneled through the inside of the box as shown in Figure 6.

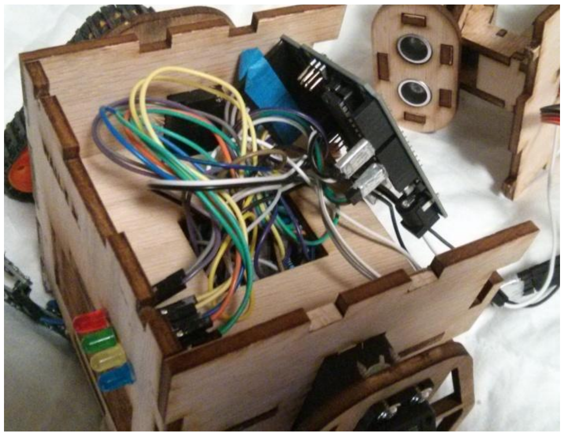


Figure 6: Close up of the top wiring configuration

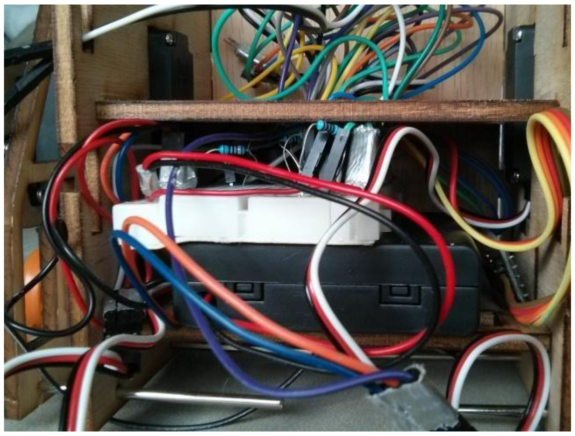


Figure 7: Close up of bottom wiring configuration

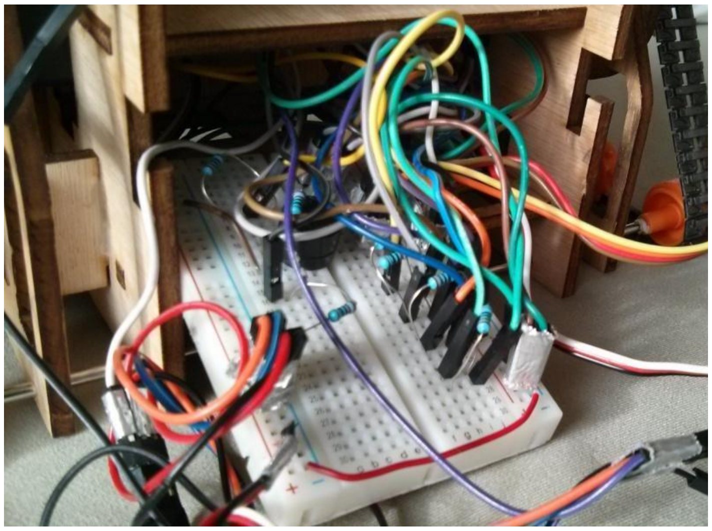
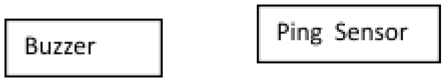
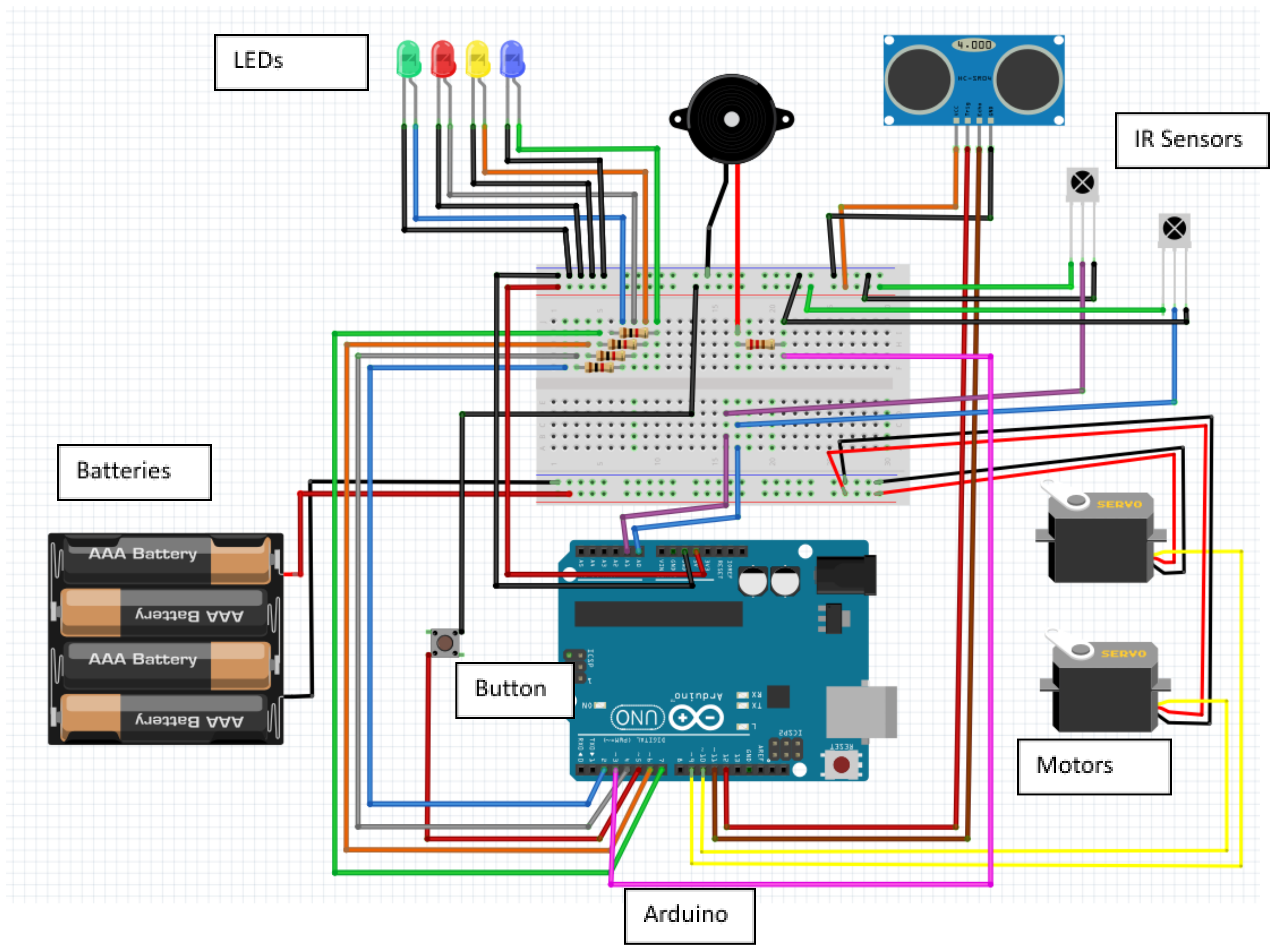


Figure 8: Close up of breadboard wiring configuration

Step 2: Wiring:

The following is a wiring diagram of the robot using the program "Fritzing". The colors of the wires have no actual importance to the function of the robot. However, it is recommended that wires connected to positive leads are red and wired connected to ground leads are black. The wire cutters can be used on the battery pack and buzzer to allow for easier placement of the wire ends into the breadboard. A separate 9V battery placed inside the robot is needed to power the Arduino since the motors require their own power supply provided by the battery pack.





Step 3: Programming

```
 /*
* ENME 408
* Line Following Program
* Jonathan Shulgach
* 10/4/15
*
* This code allows the robot to follow a line using a PID controller and calibration component
```

```
 * to detect variances between light intensities of the surface. The code also allows the robot
 * to distinguish distances between objects the robot faces, and creates a sound when an object
 * is detected in its path.
 *
 * Credit to www.instructables.com for line following robot design project ideas and for code
 * examples. Acknowledgements to Dr. Stephen Wilkerson for providing examples of photoresistor,
 * LED, motor, and variable initialization code examples.
 */


//Initialize variables and libraries
#include <Servo.h>
Servo Left;
Servo Right;

#define sensorLeft 15 //A1
#define sensorRight 14 //A0
#define trigPin 12
#define echoPin 11
#define buzz 8
#define ledB 7 //Blue
#define ledY 6 //Yellow
#define btn 3 //button
#define ledG 2 //Green
#define ledR 4 //Red



const int sens2 = 14;
const int sens1 = 15;
int pos = 0; //initial conditions
int lightread[150];
int sensorValue1 = 0;
int sensorValue2 = 0;
int sensorMax = 0;
int sensorMin = 1023;
int mid = 0;
int mybtn = 0;
int g_state = 0;
int dx = 0;
int dxd = 0;
char RUN =0;
int j = 0;
int k = 0;
int rlog = 0;
int llog = 0;


//Begin set up loop by setting pins to input and output modes
void setup() {
 Serial.begin(9600);
 pinMode(trigPin,OUTPUT);
 pinMode(echoPin,INPUT);
 pinMode(ledB,OUTPUT);
 pinMode(ledY,OUTPUT);
 pinMode(ledR,OUTPUT);
 pinMode(ledG,OUTPUT);
 pinMode(btn,INPUT_PULLUP);
 Right.attach(9, 800, 2200);
 Left.attach(10, 800, 2200);
}



//This is the main loop which allows the robot to run its line following and
//pinging functions by pressing the button
void loop()
{
 mybtn=digitalRead(btn);
 //Serial.print(mybtn); //These can be initialized during testing of the code to
 //Serial.print(",");
 //Serial.println(g_state);

 if((mybtn==0) && (g_state==0))
{
 g_state=1;
}
else if((mybtn==0) && (g_state==1))
{
g_state=2;
}
else if((mybtn==0)&&(g_state==2))
{
g_state=0;
}

 if (g_state==1)
 {
  digitalWrite(ledB,HIGH); //If button is pressed, line following program will execute
  linefollow(10);
  Serial.println(dx);
 }

 if (g_state==2) //If button is pressed, robot will enter into entertain mode with the lights acting in
sequence
 {
  digitalWrite(ledR,HIGH);
  delay(50);
  digitalWrite(ledG,HIGH);
  delay(50);
  digitalWrite(ledY,HIGH);
  delay(50);
  digitalWrite(ledB,HIGH);
  delay(50);
  digitalWrite(ledR,LOW);
  delay(50);
  digitalWrite(ledG,LOW);
  delay(50);
  digitalWrite(ledY,LOW);
  delay(50);
  digitalWrite(ledB,LOW);
  delay(50);
  j=0;
  k=0;
  Right.write(90);
  Left.write(90);
 }

 if (g_state==0) //If button is pressed, robot will enter standby mode with the yellow light flashing
 {
  digitalWrite(ledY,HIGH);
  delay(100);
  digitalWrite(ledY,LOW);
  delay(100);
  j=0;
  k=0;
 }
}



void linefollow(int del) //This is the line following function which tells the robot to follow a line for either
a duration or until told to stop
{
 if (j<1)
 {
Serial.println("Calibrating max and min sensor values. Please cover photoresistor for a few seconds");
  k=0;
  while (k < 1000) {       //calibrate sensor to record max and min values for 1000 increments, around 10
seconds
   sensorValue1 = analogRead(sens1);
   digitalWrite(ledB,HIGH);
     if (sensorValue1 > sensorMax) {
        sensorMax = sensorValue1;
     }
     if (sensorValue1 < sensorMin) {
        sensorMin = sensorValue1;
     }
     Serial.println(k);
     k=k+1;
  }

  digitalWrite(ledB, LOW);
```

```
   mid = (sensorMax+sensorMin)/2;


 Serial.println(sensorMax); //All lights will flash once when calibration sequence is finished
 Serial.println(sensorMin);
 digitalWrite(ledG,HIGH);
 digitalWrite(ledR,HIGH);
 digitalWrite(ledY,HIGH);
 digitalWrite(ledB,HIGH);
 delay(1000);
 digitalWrite(ledG,LOW);
 digitalWrite(ledR,LOW);
 digitalWrite(ledY,LOW);
 digitalWrite(ledB,LOW);
 delay(1000);
 j=j+1;
}

  sensorValue1 = analogRead(sensorRight); //read port A0
  int rightmapval = map(sensorValue1, sensorMin, sensorMax, 0, 100); //display sensor reading between 1
and 100
  rightmapval = constrain(sensorValue1, 0, 100);

  sensorValue2 = analogRead(sensorLeft);
  int leftmapval = map(sensorValue2, sensorMin, sensorMax, 0, 100); //display sensor reading between 1
and 100
  leftmapval = constrain(sensorValue2 , 0, 100);

  Serial.print(rightmapval);
  Serial.print(" ");
  Serial.println(leftmapval);

  dx=(leftmapval-rightmapval); //This is the change in value between the sensors which controls the PID
controller

if ((dx<=50) && (dx>=-50)) //If there's no significant difference in color, the robot will go forward.
{
  Right.write(0);
  Left.write(180);
  digitalWrite(ledB,LOW);
}

else if (dx>50) //controls right wheel power (smaller than -50 makes wheel go backward
{
  Right.write(0);
  llog=90 + (dx/3 + (dxd-dx)/5)-90;
  Serial.println(llog);
  Left.write(llog);
  Right.write(0);
  digitalWrite(ledB,HIGH);
  dxd=dx;
  delay(50);
}
else if (dx<-50) //controlls left wheel power (greater than -20 makes wheel go backward
{
  Left.write(180);
  rlog=90 - (dx/3 + (dxd-dx)/5) + 10;
  Serial.println(rlog);
  Right.write(rlog);
  Left.write(180);
  digitalWrite(ledB,HIGH);
  dxd=dx;
  delay(50);
}
else
{
  Right.write(90); //Robot will stop for emergency
  Left.write(90);
}
ping(10); //This runs the pinging subfunction which can be turned off here to decrease tthe latency of the
line following function
}


void ping(int del) //This is the ultrasonic sensor function which tells the robot to buzz when a object
comes within 10 cm from the robot.
{
  long duration, distance;
  digitalWrite(trigPin,LOW);
  delay(10);
  digitalWrite(trigPin,HIGH);
  delay(10);
  digitalWrite(trigPin,LOW);
  duration=pulseIn(echoPin,HIGH);
  distance=(duration/2)/29.1; //Conversion from ultrasonic sensor values to centimeters
  if (distance<10) {
    digitalWrite(ledY,HIGH); //If objects out of range, green LED stays on
    digitalWrite(ledG,LOW);
    tone(buzz,800,100);
    delay(10);
  }
  else {
    digitalWrite(ledY,LOW); //If objects come within 10cm, yellow LED comes on.
    digitalWrite(ledG,HIGH);
    delay(10);
  }
  if (distance>=300||distance<=0){
    //Serial.println("Out of range"); //This can be activated during the testing of the code
  }
  else{
    //Serial.print(distance); //display distance if necessary to test code
    //Serial.println(" cm");
  }

  delay(del);
}
```

Step 4: Calibration

1) Make sure the 9V battery is placed inside the robot, while connected to the Arduino, and the motor power supply switch is turned on.

2) The robot will enter standby mode when turned on. Press the button again to begin the calibration function.

3) While the blue light is on, hold the robot in a way to measure the light intensity of the ground and the line with both sensors.

4) When the calibration sequence ends, place the robot onto the ground with the line to follow between the IR sensors. Press the bottom on the back panel to stop the line following program.

Conclusion:

The robot should be able to make smooth turns when following the line. It has a small turning radius due to a tread going backwards to assist in its line following track, and can be adjusted by making changes to the code. When done playing with the robot, the battery pack underneath needs to be turned off, and the 9V battery connector must be disconnected to the Arduino. The back panel is the only part that does not need to be glued. However, the user can choose to glue the back and top box parts together as one component with the head and keep the remainder of the robot a separate component. Please play responsibly with the robot, and have fun with "Wall-E Jr".