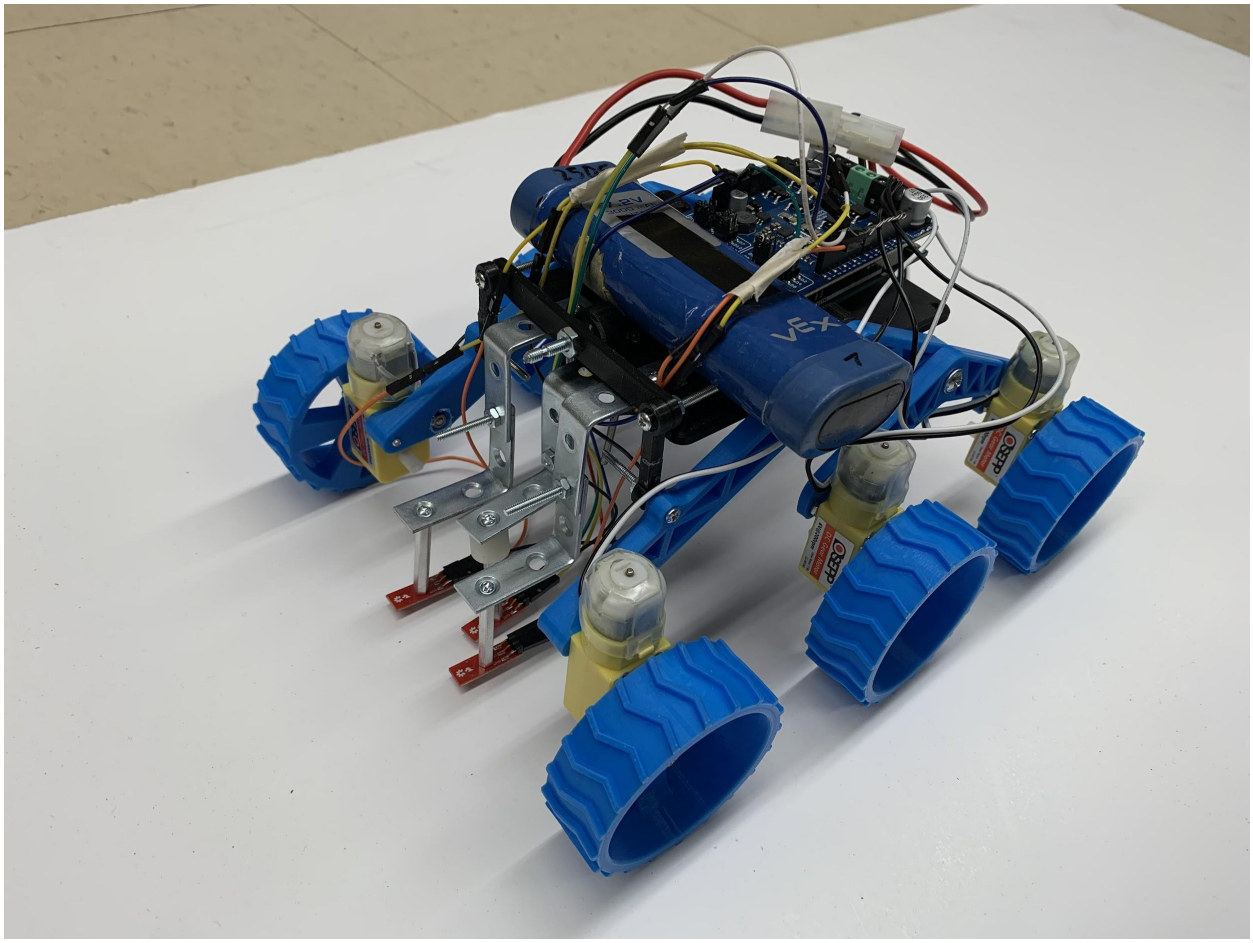
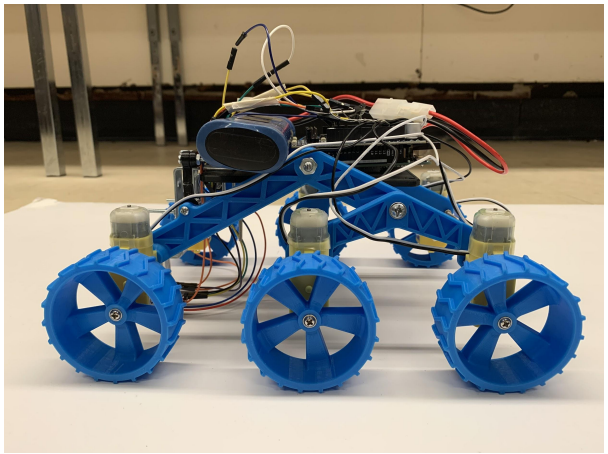
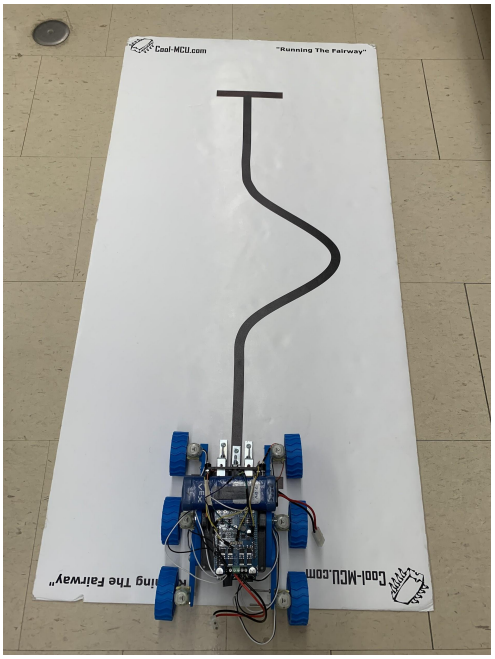
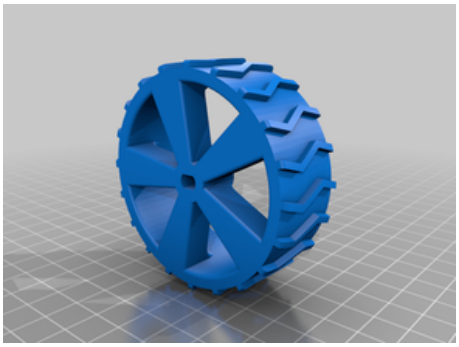
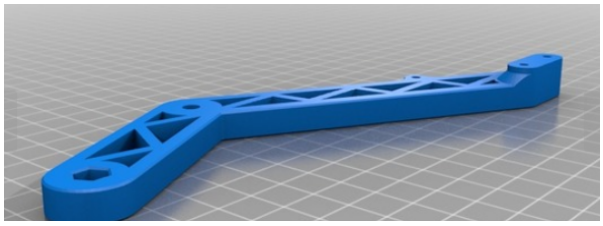
How To Build Your Own Line-Following Automated Rover

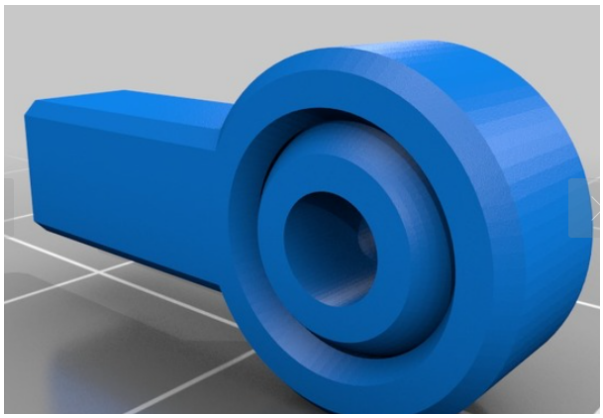


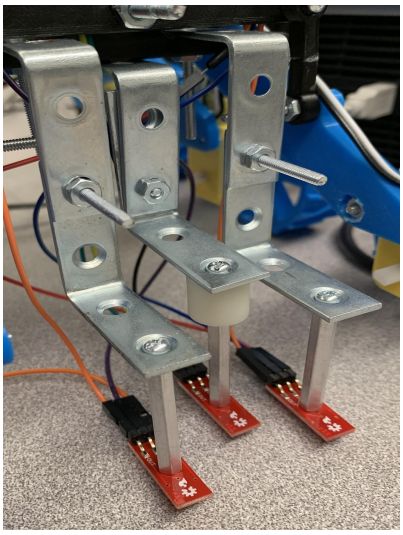


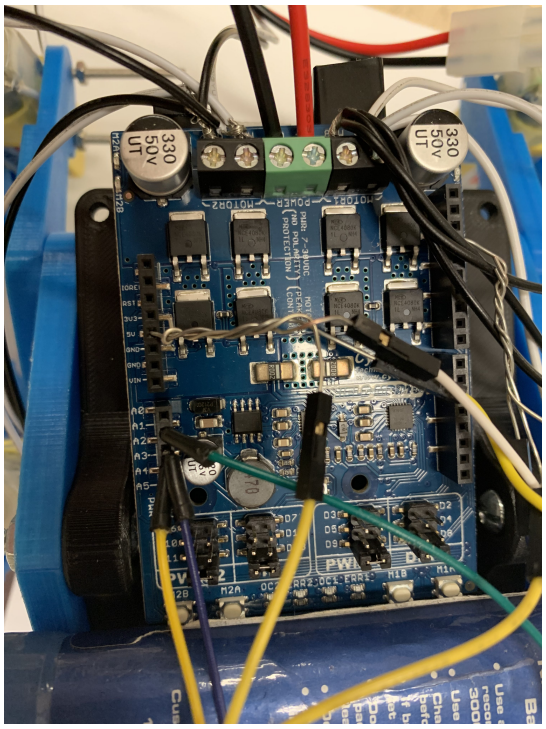


PARTS LIST

Item	Image
Motors x6	
Wheels x6	
Chassis Part For The Back Wheels x2	
Chassis Part For Connecting The Front Wheels To The Back Wheels x2	

Main Body Of The Chassis x1	
Chassis Rod To Connect Support Links To The Wheels x1	
Chassis Support Links x4	

Battery 7.2 Volts	
Velcro Strips x2	
Angle Brackets x6 Spacer x1 Standoffs x3 Analog Line Sensor x3	

4-40x1 Bolts x28 4-40 Hexagon Nuts x19 8-32x3/4 Screws x6 8-32x1 Bolts x3 8-32 Hexagon Nuts x3 10-32x5/4 Bolts x2 10-32 Hexagon Nuts x2	
Arduino Uno x1 Motor Shield x1	

Goal: To Build An Automated Robot Which Can Follow A Path

Process For The Hardware Section

Step 1: First Connect Each of the 6 Wheels to the Motors using the 8-32 screws.

Step 2: Connect the Motors to each end of the Chassis Parts for the wheels using the 4-40x1 bolts.

Step 3: Connect the Chassis part for the back wheels to the larger part for the front wheels using the 8-32 bolts and nuts.

Step 4: Attach the Chassis Parts for the wheels to the Main Body of the Chassis using the 10-32 bolts and place a nut on the opposite end.

Step 5: Using the 8-32 bolts and nuts, attach and secure the Chassis Rod to the front of the Main Body of the Chassis.

Step 6: Connect two Chassis Support Links together, making pairs of two. This can be done using glue and a small wire as a stabilizer.

Step 7: Attach both pairs of Chassis Support Links to the ends of the Chassis Rod using the 4-40 bolts and nuts. Then connect the other ends of the support links to the chassis part supporting the front wheels. Rotate it until the holes line up, and then use the 4-40x1 bolts and nuts to make it all stable. The frame of the rover should be now completed.

Step 8: Form 3 pairs of a ㄣlike shape using the 6 Angle Brackets you have, with one having a shorter length than the other two and connect them together using the 4-40 bolts and nuts.

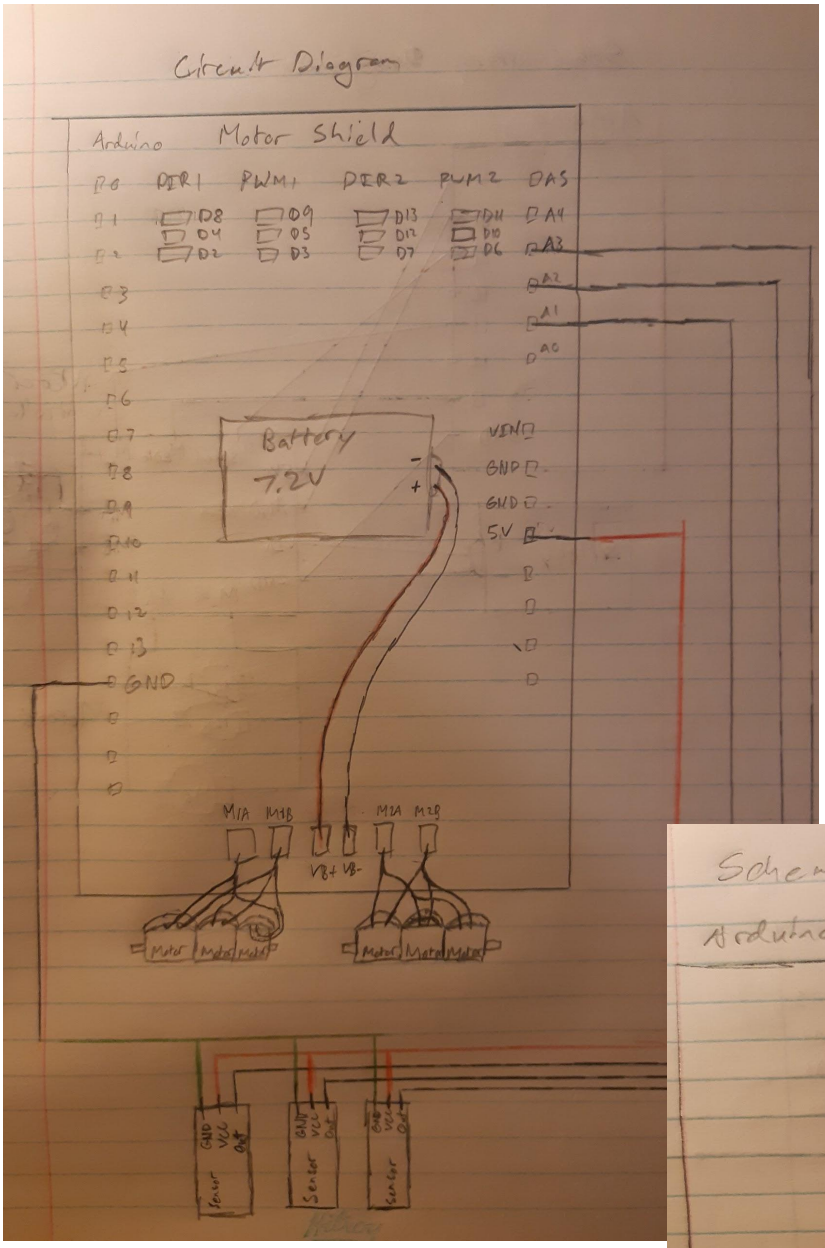
Step 9: Attach your 3 pairs of Angle Brackets to the Main Body of the Chassis. Make it so that the middle one is underneath the section connecting the middle of the Chassis Rod, and the two outer ones are beside that section which is on top of the Main Chassis.You can do this by drilling a hole through the Main Chassis section, and then using the 4-40 bolts and nuts to make them stable.

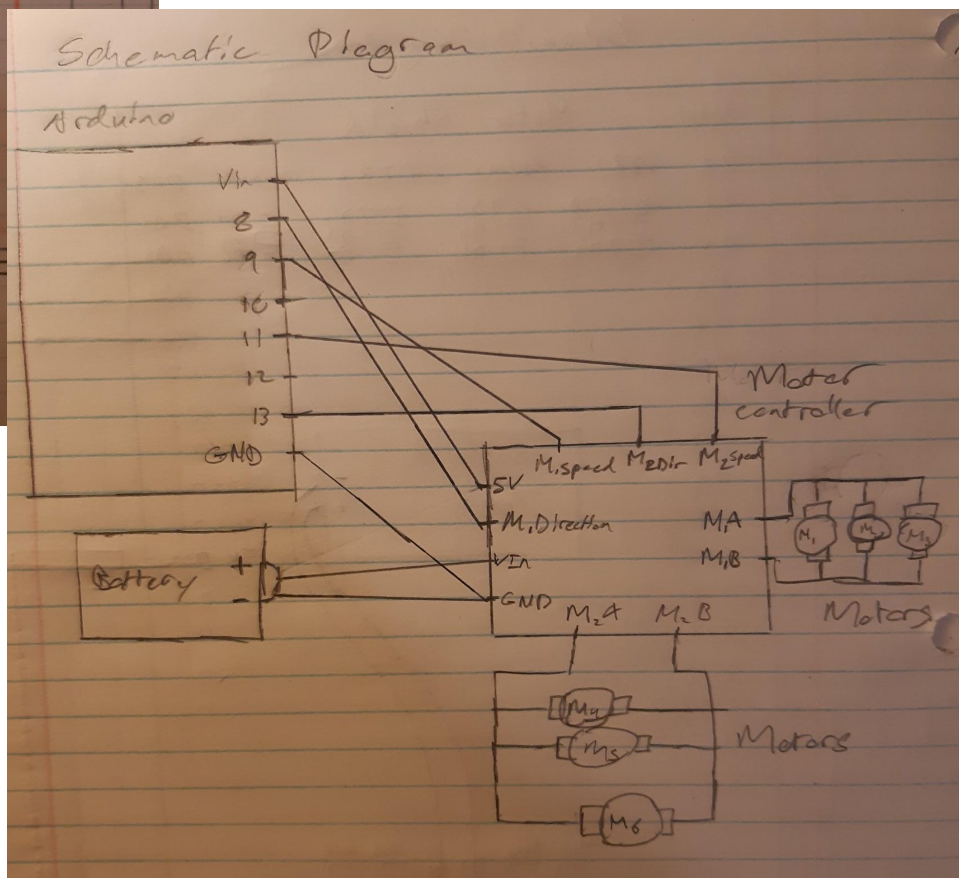
Step 10: Then add a standoff under each pair of Angle Brackets, and then attach the Analog Line Sensors underneath the standoffs so that the camera is facing forward. This can be done using the 4-40 x1/4 and x1 length bolts (A spacer and standoff must be added to the middle sensor since it is shorter in height)

Step 11: Using the Velcro, stick one end of the velcro to the Arduino Uno and the other end to the rear section of the chassis on your rover. Stick them together so that the Arduino stays on the chassis when the rover is moving.

Step 12: Place the Motor Shield on your Arduino Uno, as this will allow you to code the movement of the rover. And then power everything up with your 7.2V battery

Step 13: Follow the diagrams below for the wiring and the circuitry between the Arduino and the rover.





The Code

Arduino Pinout:
Pin 9 - motor 1 speed
Pin 8 - motor 1 direction
Pin 11 - motor 2 speed
Pin 13 - motor 2 direction
Pin A1 - sensor 1
Pin A2 - sensor 2
Pin A3 - sensor 3

Forward: motor 1 - LOW, motor 2 - HIGH
Backward: motor 1 - HIGH, motor 2 - LOW
Left: motor 1 - HIGH, motor 2 - HIGH
RIght: motor 1 - LOW, motor 2 - LOW
Stop: set speed = 0

Using the serial monitor: sensor detecting black ~ 700, sensor detecting white ~ 50, median = 375

The code below does the following:

- *Turns 180 degrees once reaching and endpoint*
- *Moves forward when both left and right sensors detect white*
- *Moves slightly left when left sensor detects black*
- *Moves slightly right when right sensor detects black*

```
void setup() {

  // Motor_1 control pin initiate;
  pinMode(9, OUTPUT); // Speed control
  pinMode(8, OUTPUT); // dir

  // Motor_2 control pin initiate;
  pinMode(11, OUTPUT);  // Speed control
  pinMode(13, OUTPUT); // dir

  //Enable the Motor Shield output;
  pinMode(6, OUTPUT);
  digitalWrite(6, HIGH);

  // sensor inputs
  int l_sensor = 1; // left sensor
  int r_sensor = 3; // right sensor
  Serial.begin(9600);
  pinMode(l_sensor, INPUT);  // input sensors
  Serial.begin(9600);
  pinMode(r_sensor, INPUT);
```

```
}

void loop() {

  int median = 375; // middle sensor value between white and black (white ~ 50, black ~ 700)
  int l_sensor = 1;
  int r_sensor = 3;
  int l_s = analogRead(l_sensor);  //Reading the value of each sensor
  Serial.println(l_s);
  int r_s = analogRead(r_sensor);
  Serial.println(r_s);

  if(l_s > median && r_s > median)  // if both left sensor and right sensor detect black, turn 180
degrees
  {
          stop(); // stop for 2 seconds once it reaches an endpoint
          delay(2000);
          left(100);
          delay(2000);
  }
  else if(l_s > median) // if left sensor detects black, move left
  {
          left(200);
          delay(50);
  }
  else if(r_s > median) // if right sensor detects black, move right
  {
          right(200);
          delay(50);
  }

  else  // if both sensors detect white, move forward
  {

          forward(25);  // slower speed
          delay(50);
  }

  stop();

}

void forward(int speed)
{
```

```
  analogWrite(9, speed);  // Set the speed of motor_1
  digitalWrite(8, LOW);  // Set the rotation of motor_1
  analogWrite(11, speed);  // Set the speed of motor_2
  digitalWrite(13, HIGH);  // Set the rotation of motor_2
}

void backward(int speed)
{
  analogWrite(9, speed);  // Set the speed of motor_1
  digitalWrite(8, HIGH);  // Set the rotation of motor_1
  analogWrite(11, speed);  // Set the speed of motor_2
  digitalWrite(13, LOW);  // Set the rotation of motor_2
}

void stop()
{
  analogWrite(9, 0);  // Set the speed of motor_1
  analogWrite(11, 0);  // Set the speed of motor_2
}

void left(int speed)
{
  analogWrite(9, speed);  // Set the speed of motor_1
  digitalWrite(8, HIGH);  // Set the rotation of motor_1
  analogWrite(11, speed);  // Set the speed of motor_2
  digitalWrite(13, HIGH);  // Set the rotation of motor_2
}

void right(int speed)
{
  analogWrite(9, speed);  // Set the speed of motor_1
  digitalWrite(8, LOW);  // Set the rotation of motor_1
  analogWrite(11, speed);  // Set the speed of motor_2
  digitalWrite(13, LOW);  // Set the rotation of motor_2
}
```

Problems that may have occurred during the building of this robot

- If the motor is very loose and can fall off easily from the chassis part, you can drill a hole through the chassis just enough for a nut to fit inside on the opposite end of the bolt head, or use a larger size 4-40 bolt (like a 4-40x2 and then place a nut at the end)

- Occasionally some of the bolts we used were too long, like the 8-32x1 or the 10-32x5/4 were too long, and was blocking the wheels from turning all the way around. In this case, we used pliers to cut through the bolts and then just smoothened out the cuts afterwards

- When testing the robot, it was difficult to make sharp turns smoothly. This was because the speed value of the turns was slower with a longer delay, so we increased the speed value of the turns (sharper turns). It would successfully make the sharp turn, however, it would oscillate left and right while making the turn.

- Occasionally after reaching the end of the path, the robot would not turn enough so that it can complete a 180 degree turn to go back on the line, this problem happens because of the types of wheels you have and if they have a good grip or not, and how much each individual motor spins compared to the rest. To fix this we had to use trial and error to get our desired value.

Future Additions/Ideas We Can Make To The Rover

- We can add another chassis to the original one and then with the additional chassis we can put some materials on it. If the rover was more enlarged, this would allow a rover to move from one assigned destination to another, while stopping for a few moments to allow the removal and addition of materials onto its chassis.

- We could decide to make it remote controlled using a keyboard, while being able to stream video using the raspberry pi.

Citations

The Hardware portion was inspired by: https://www.thingiverse.com/thing:5169882

The Parts list was taken from the original: https://www.thingiverse.com/thing:3576552