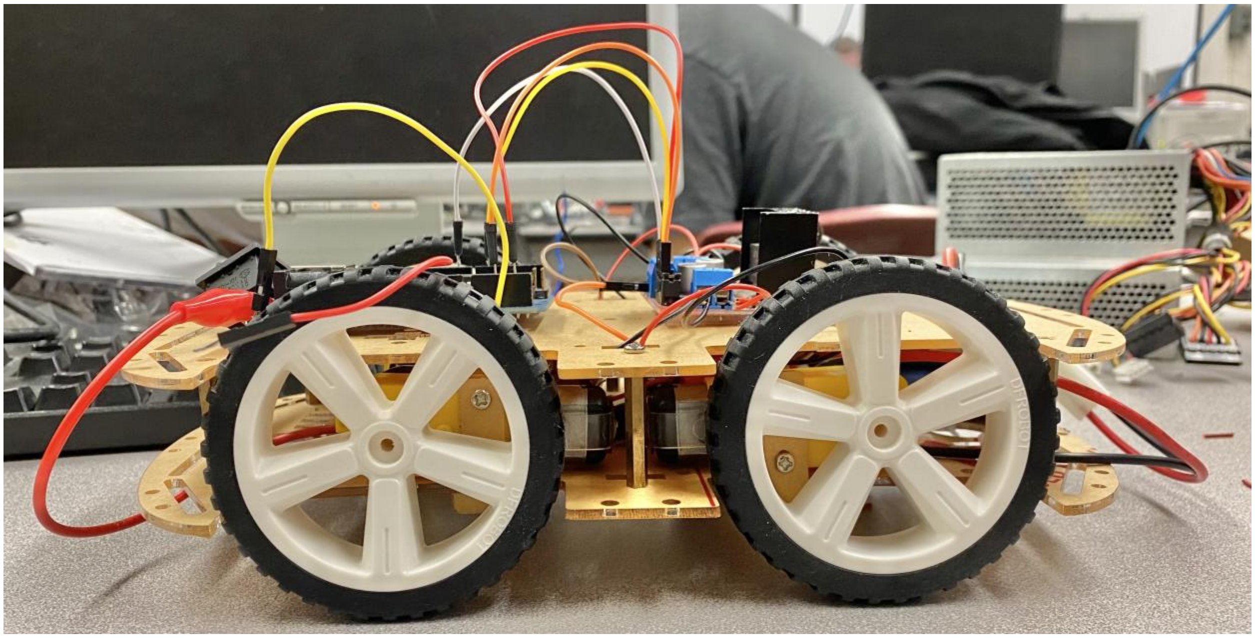
4-wheel motorized Car



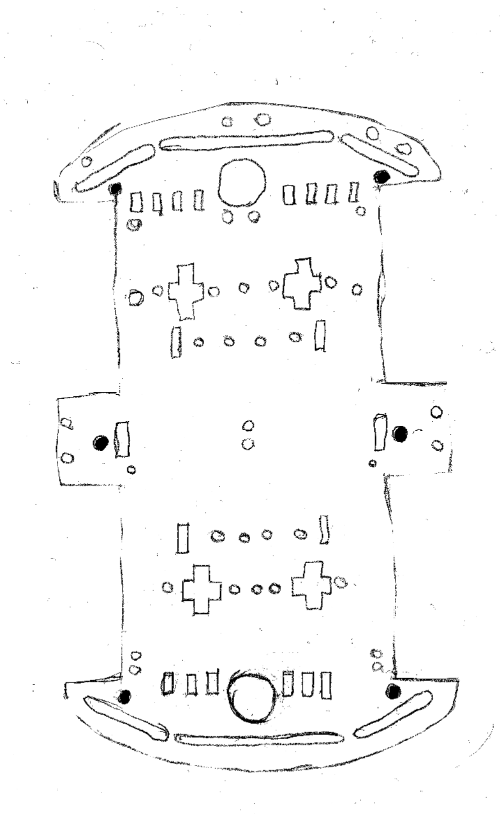
Goal: The goal of this project is to create a four-wheel car that, at a starting point, follows a non-linear line and returns back to the starting point as soon as it reaches the end of the line. The car's code allows the vehicle to repeat its goal an infinite amount of times until the switch is off or the battery runs out of charge. The two-line sensors will adjust the direction of the car to allow it to remain on the line.

Components	
Electronic	Physical
Audriono Duemilanova	Standoffs (X6)
7.2 V Nickel-metal hydride battery	2 mm acrylic paper (X2) (one for top one for bottom)
Rocker switch	2 screws for sensors
L298 Dual Motor Driver - 2A	velcro (X4)
5 V DC motor (X4)	6 Hex nuts for sensors
Analog Line Sensors (X2)	Rubber wheels
Breadboard	12 small screws for the standoffs
Alligator Clips	

Copper Wires	
Jumper Wires	

<u>Step 1: Preparing the chassis</u>

1. Create holes in the two acrylic sheets to get a structure like this:



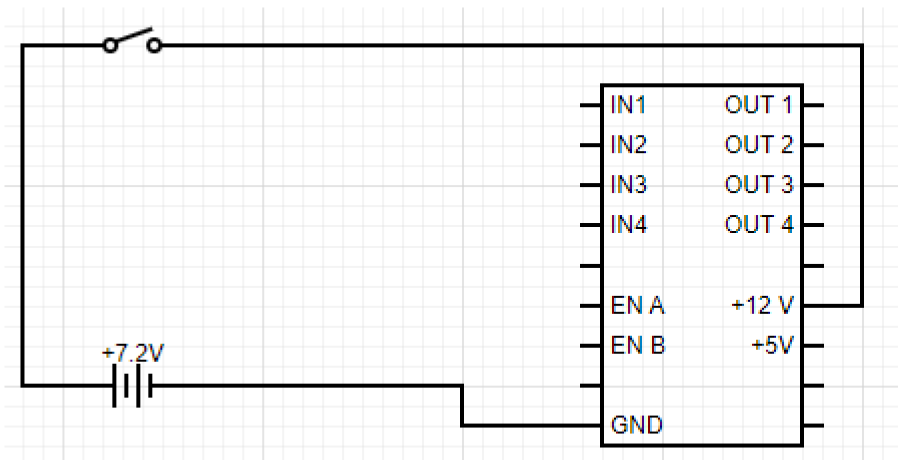
2. On the top side of the bottom acrylic sheet hot glue, the 4 motors on the board so that the wheels fit the open gaps and attach to the motors.
3. Attach the wheels to the motors
4. Fit the wires through the holes of the top acrylic sheet and solder the corresponding wires by the side of the car they are in and its colour
    a. It is to thicken the wires so that they do not become loose when inserted into the output pin
    b. So all 4 motors can spin instead of 2 due to the lack of output pins
5. Attach the sensor inside of a screw

    a. Add a hex bolt so that the sensor is secure against screw
6. Place the screw in the front part of the bottom acrylic sheet and secure it with 2 hex bolts
7. Attach the entire screw on the front part of the bottom acrylic sheet
8. Repeat steps 5 and 6 and attach the sensor on the opposite side of the car
9. Attach velcro to the battery, driver, Audrino, and breadboard to solidify its placement on the acrylic sheet
    a. All the electronic components go on the top side of the top acrylic paper with the breadboard at the front. The Audrino is behind it, followed by the driver and the battery in the back
10. Join the top and bottom acrylic sheets together by placing the standoffs in the holes with the shaded regions and connecting them with screws

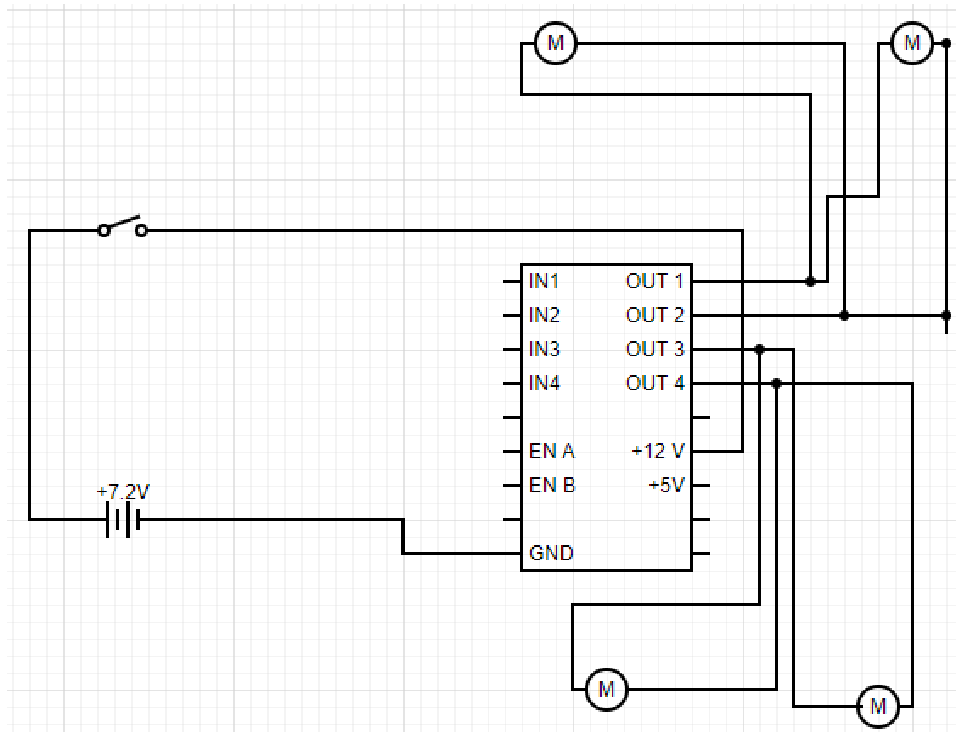
Step 2: Create the electronic circuit

1. Attach the battery to the Motor Driver
    a. Attach the positive terminal of the battery to the switch
    b. Attach the other side of the switch to the +12V on the Driver
    c. Attach the negative terminal of the battery to the ground of the Driver

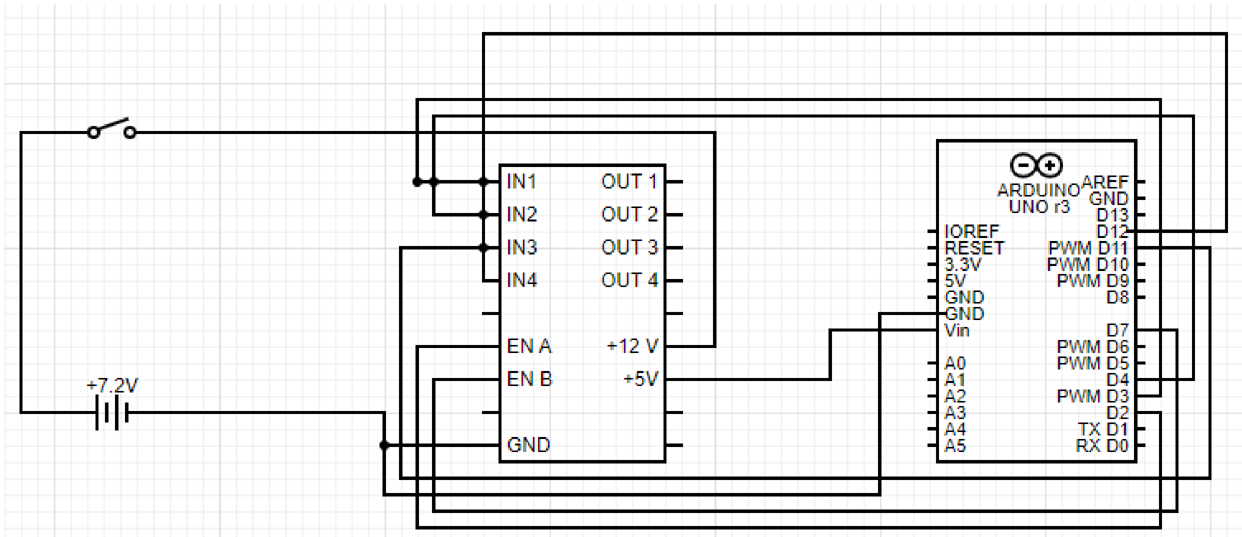
Note: keep the switch open to stop the current flow while building the rest of the circuit



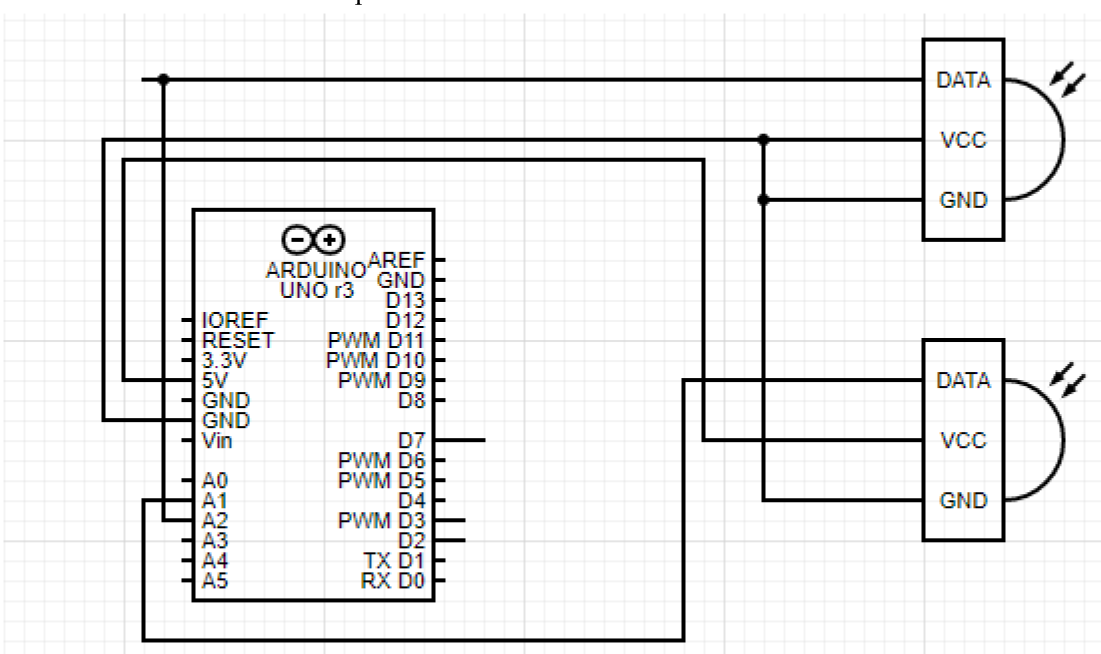
2. Attach the motors to the Driver
    a. Attach the wires of the right side of the motors to pin OUT 1 and OUT 2
    b. Attach the wires to the left side of the motors to pin OUT 3 and OUT 4



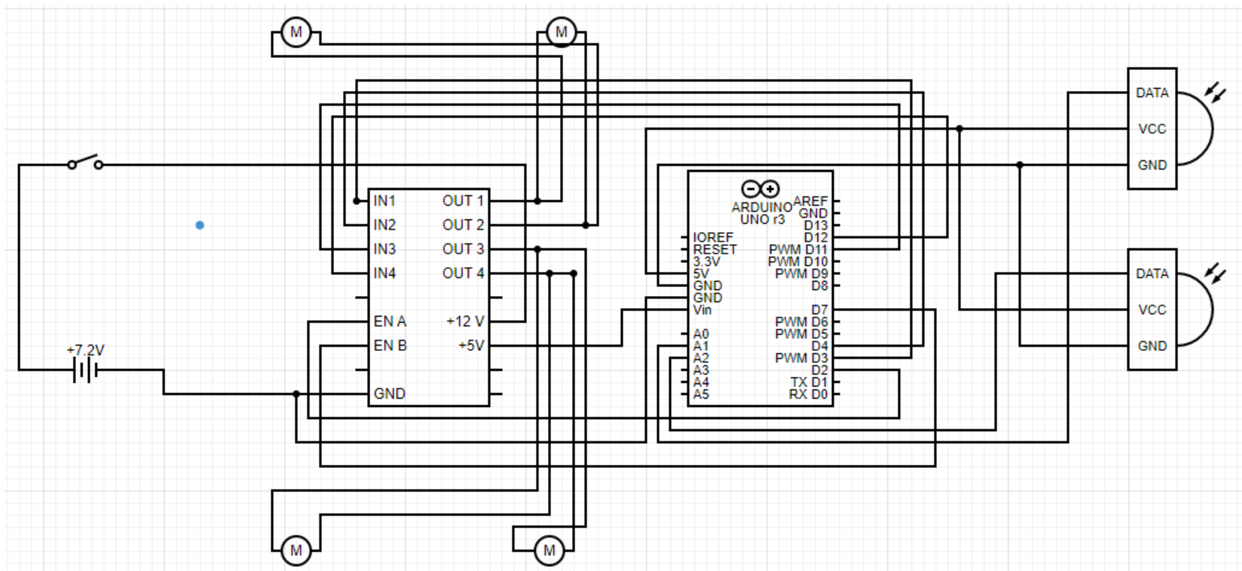
3. Attach wires from the battery and the Driver to the Audrino
   a. Connect the +5V pin on the Driver to the VIN pin on the Audrino
      i. This gives power to the Audrino from the Driver
   b. Connect the ground from the battery to the GND pin on the Audrino
      i. Note: the ground from the battery is in parallel with the Driver and Audrino
   c. Connect the IN1 pin on the Driver to pin D3 on the Audrino
      i. This controls the speed of motor 1 (motors on the right side of the car)
   d. Connect the IN2 pin on the Driver to pin D4 on the Audrino
      i. This controls the direction of the wheel spin of motor 1
   e. Connect the IN3 pin on the Driver to pin D11 on the Audrino
      i. This controls the speed of motor 2 (motors on the left side of the car)
   f. Connect the IN4 pin on the Driver to pin D12 on the Audrino
      i. This controls the wheel spin of motor 2
   g. Connect pin EN A of the Driver to pin D2 of the Audrino
      i. Enables the PWM signal for motor 1
   h. Connect pin EN B of the Driver to pin D7 of the Audrino
      i. Enables the PWM signal for motor 2



4. Attach wires from the Audrino to the sensors
    a. Connect the VCC pins of both sensors to the positive column of the breadboard
    b. Connect the positive column of the breadboard to the 5V pin of the Audrino
    c. Connect the GND pins of both sensors to the negative column of the breadboard
    d. Connect the negative column of the breadboard to the GND pin of the Audrino
    e. Connect the DATA pin of the right sensor to A1 of the Audrino
    f. Connect the DATA pin of the left sensor to A2 of the Audrino



Note: This is what the entire circuit should look like



Note: Circuit still uses Arduino Duemilanova despite the diagram showing an Arduino Uno.
The unnamed circuit is the Driver.

<u>Step 3: Setting up code</u>

First step is to install Arduino IDE version 1.8. Once this is downloaded, open a new file and select Arduino Duemilanove as the board.

```
int pin_2 = 10; //input from r_sensor
int pin_1 = 10; //input from l_sensor
int value_1 = 10; //output to motors
int value_2 = 10; //output to motors
```

```
void setup() {

  // Motor_1 controll pin initiate;
  pinMode(3, OUTPUT) ;  // Speed control
  pinMode(4, OUTPUT) ;  // dir

  // Motor_2 controll pin initiate;
  pinMode(11, OUTPUT);  // Speed control
  pinMode(12, OUTPUT); // dir

  //Enable the H bridge output for Motor 1;
  pinMode(2, OUTPUT);
  // Enable the H bridge output for Motor 2;
  pinMode(7, OUTPUT);

  //sensor inputs
  int r_sensor = 2;
  int l_sensor = 1;
  Serial.begin(9600);
  pinMode(r_sensor, INPUT);  //input sensor
  Serial.begin(9600);
  pinMode(l_sensor, INPUT); //input sensor

}
```

The following code setups the values from the Arduino to the driver. Motor 1 will receive a PWM signal from pin 3 which controls the speed of the motor. It will receive a digital signal from pin 4 which controls the direction of the motor. Motor 2 receives a PWM signal from pin 11 which controls the speed of the motor while getting a digital signal from pin 12 which controls the direction of motor 2. Pin 2 is the input of the Arduino which will allow the Driver to receive a PWM signal for motor 1 while pin 7 is the input of the Arduino which allows the driver to receive a PWM signal for motor 2. Analog pin 1 is the input where the right sensor receives its values while analog pin 2 is the input where the left sensor receives its values.

```
void forward(int speed = 255)
{
  analogWrite(3, speed);  // Set the speed of motor_1
  digitalWrite(4, HIGH);  // Set the rotation of motor_1
  analogWrite(11, speed);  // Set the speed of motor_2
  digitalWrite(12, HIGH);  // Set the rotation of motor_2
}

void stop ()
{
  analogWrite(3, 0); // Set the speed of motor_1
  analogWrite(11, 0); // Set the speed of motor_2
}

void left (int speed = 100)
{
  analogWrite(3, speed);  // Set the speed of motor_1
  digitalWrite(4, LOW);  // Set the direction of motor_1
  analogWrite(11, speed);  // Set the speed of motor_2
  digitalWrite(12, HIGH);  // Set the direction of motor_2
}

void right (int speed = 100)
{
  analogWrite(3, speed);  // Set the speed of motor_1
  digitalWrite(4, HIGH); // Set the direction of motor_1
  analogWrite(11, speed); // Set the speed of motor_2
  digitalWrite(12, LOW); // Set the direction of motor_2
}

void backward(int speed = 255)
{
    analogWrite(3, speed);  // Set the speed of motor_1
    digitalWrite(4, HIGH);  // Set the direction of motor_1
    analogWrite(11, speed); // Set the speed of motor_2
    digitalWrite(12, LOW);  // Set the direction of motor_2
```

The following car movement defines the functions in the loop function. The left and right functions move at a slower speed for the sensors to properly detect the lines and move accordingly to the path with accuracy.

```
void loop ()  {

  int median = 375;

  int r_s = analogRead(pin_2);  // Reading the values of the senors
  Serial.println(value_2);
  int l_s = analogRead(pin_1);
  Serial.println(value_1);
if(r_s and l_s > median)  // if left and right senors detects white, move foward a bit
{
    forward(50);
    delay(100);
   }
  else if(r_s > median) // if right sensor detects black move right a bit
   {
    right(50);
    delay(100);
    forward(50);
    delay(100);
   }
  else if(l_s > median) // if left sensor detects blacks move left a bit
   {
    left(50);
    delay(100);
    forward(50);
    delay(100);
   }
}
```

The initial function is for the car to move forward. Both motors spin forward for 50 milliseconds and then delays for 100 milliseconds and repeat unless it reaches another state. If the right sensor detects a black line the car is supposed to turn right, meaning that motor 1 is spinning forward while motor 2 stops moving. Then it’ll stop for 100 milliseconds, go forward 50 milliseconds and delay 100 seconds. It’ll repeat until the sensor stops detecting black. If the left sensor detects black the car is supposed to turn left, meaning only motor 2 is spinning forward for 50 milliseconds. Then it’ll delay for 100 milliseconds, move forward 100 milliseconds and delay for another 100 milliseconds.