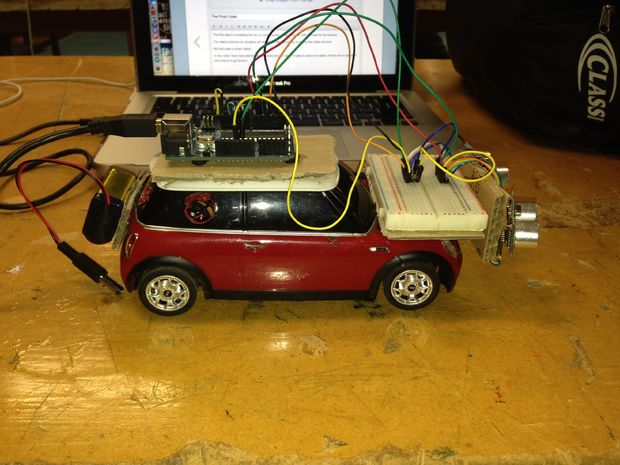
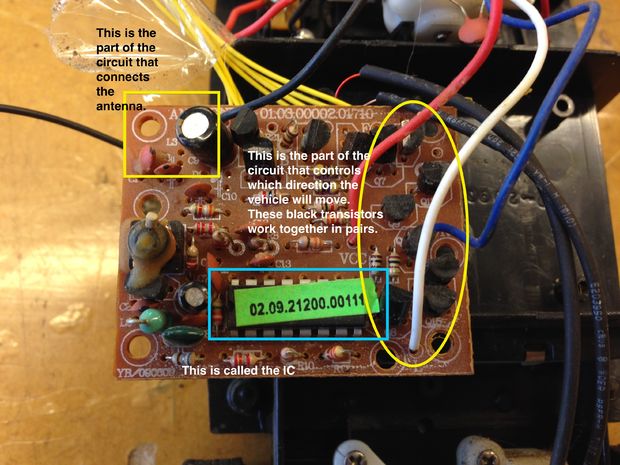
[](http://www.instructables.com/file/FGTOLPXHWJJW2BO/)

This instructable shows how to re-build a second hand remote control car. You will see how to re-wire the circuit board. As well, basic coding and Arduino based skills are demonstrated in this construction. The final product should be able to navigate its way through any environment it is placed in. This will be done by using small sensors.

**Step 1: Learning About the Circuit Board**

[](http://www.instructables.com/file/FV8V6PQHUOP8SHP/)

[](http://www.instructables.com/file/FB6KC7BHUOP8SHS/)

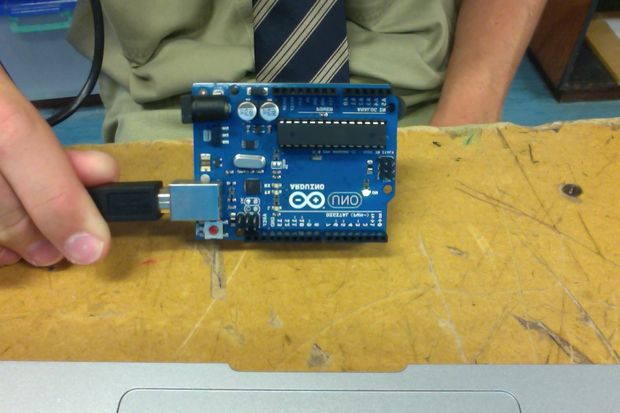
The first step you need to do is learn about the circuit board and understand its components. This was done by pulling our cars apart.

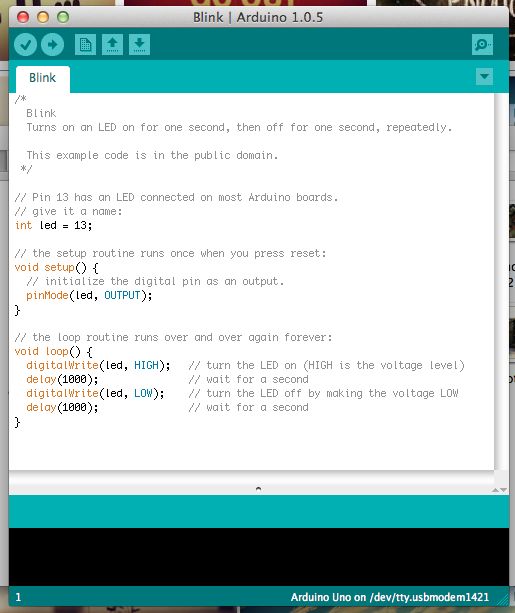
My car comprises of:

6V battery requirements

Different areas on the circuit board which control numerous things

**Step 2: Learn the Basics of Arduino**

[](http://www.instructables.com/file/FTQ50UAHUOP8SH7/)

[](http://www.instructables.com/file/FHDD1H1HUOP8TKM/)

During class we were taught the basics steps needed to use an Arduino these are below.

You can download the Arduino Zip file from: http://arduino.cc/en/Main/Software

Arduino Lesson

First expand the zip file.This will give you the Arduino application. (You can run this from anywhere)

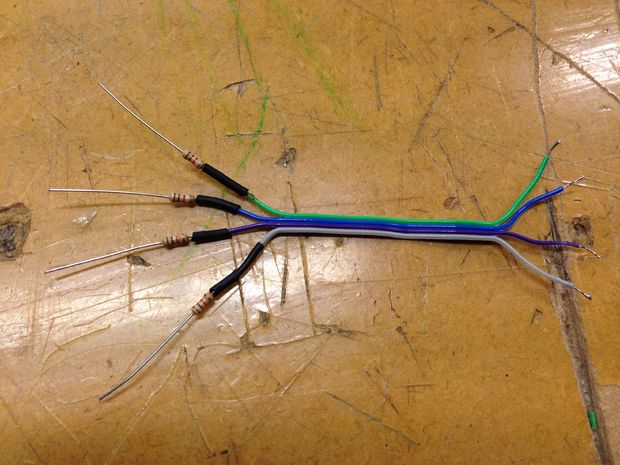
Next go to **tools, serial port** and select - **/dev/tty.usbmodem1411**

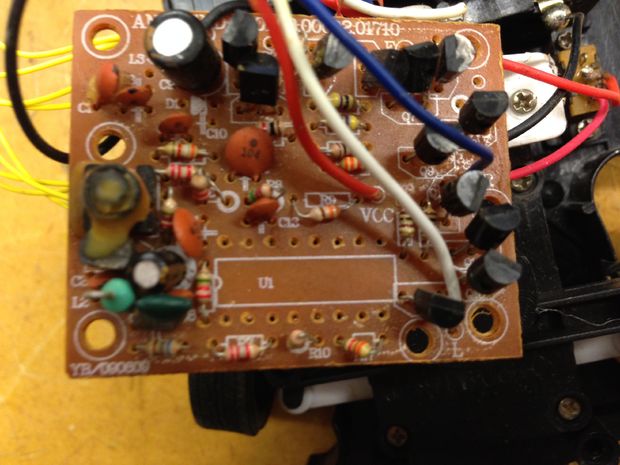
Click open and go down to **basics**, then load the control **blink**

You now **Verify** the code and then click **Upload.**

That is the Introduction to an **Arduino.**

**Step 3: Jumper Leads and Ports**

[](http://www.instructables.com/file/F06EAAUHUOP8SHW/)

[](http://www.instructables.com/file/F51CZZ6HUOP8SHX/)

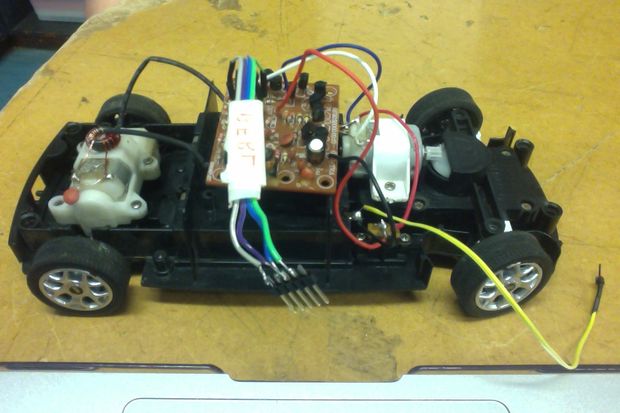
In this stage you will take apart some of the circuit board. To do this you use a solder remover. You should be able to remove the IC from my circuit board, enabling yourself to control different motions of the car manually. To do this you will need to make jumper leads. These leads connect power to the circuit board on desired ports. These ports were then tested to determine which ports control the cars ability to turn left or right and move forwards and backwards.

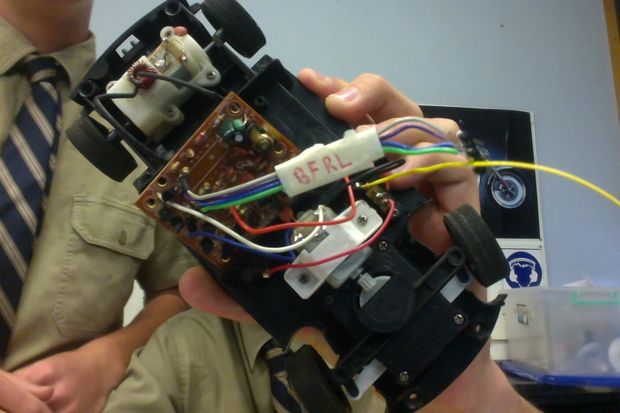
**Step 4: Connecting the Jumper Leads to the Ports**

[](http://www.instructables.com/file/F9AQQCTHUOP8SGW/)

Now you solder the jumper leads to the circuit board in the correct ports which control the movement of the car. After this you then solder together the pin stack to the other end of the jumper leads. This allows the Arduino to be plugged into the circuit board of the car

**Step 5: Directional Movement and Control**

[](http://www.instructables.com/file/F6B40O0HUOP8SHG/)

[](http://www.instructables.com/file/FLOJJFYHUOP8SHE/)

Now you need to determine which pins control which directional movement. You do this by testing each pin, with a wire connected to the battery. After you have figured out which pins do what, display this by using some tape and writing it on that. After this connect a wire to the earth pin. This then allows you to upload the code to your Arduino.

**Step 6: Figure Eight Movement**

Now that the car has directional movement, you can now test codes instructing the movement of your car. Below is the code for a figure eight movement and a video showing a test run.

This code came from https://www.instructables.com/id/Autonomous-Control-of-RC-Car-Using-Arduino/

/\*  
Car Test Makes the modified RC car go in a figure 8.

Plug the striped white wires into the Arduino pins as

\*/

int forward = 12; // forward pin

int reverse = 11; // reverse pin

int left = 10; // left pin

int right = 9; // right pin

// The setup() method runs once, when the sketch starts

void setup() {

// initialize the digital pins as an outputs:

pinMode(forward, OUTPUT);

pinMode(reverse, OUTPUT);

pinMode(left, OUTPUT);

pinMode(right, OUTPUT);

}

void go\_forward()

{

digitalWrite(forward,HIGH); // turn forward motor on

digitalWrite(reverse,LOW); // turn revers motor off

}

void go\_reverse()

{

digitalWrite(reverse,HIGH); // turn reverse motor on

digitalWrite(forward,LOW); // turn forward motor off

}

void stop\_car()

{

digitalWrite(reverse,LOW); // turn reverse motor off

digitalWrite(forward,LOW); // turn forward motor off

digitalWrite(left,LOW);

digitalWrite(right,LOW);

}

void go\_left()

{

digitalWrite(left,HIGH); // turn left motor on

digitalWrite(right,LOW); // turn right motor off

}

void go\_right()

{

digitalWrite(right,HIGH); // turn right motor on

digitalWrite(left,LOW); // tune left motor off

}

// the loop() method runs over and over again,

// as long as the Arduino has power

void loop()

{

go\_forward();

delay(1000);

go\_right();

delay(3000);

go\_forward();

delay(1000);

go\_left();

delay(3000);

go\_forward();

delay(1000);

go\_right();

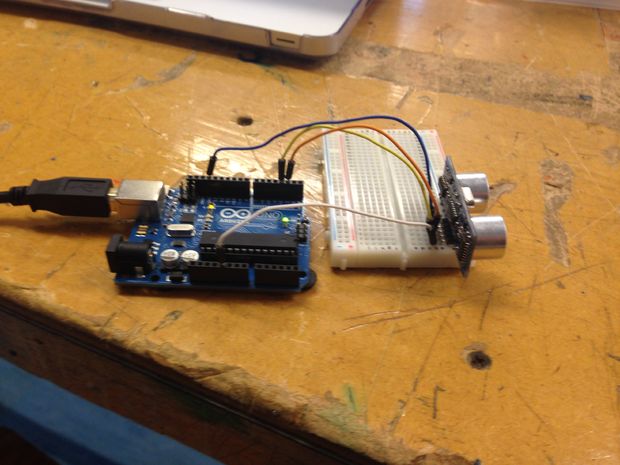
delay(3000);

}

* [[IMG_1930.MOV](https://cdn.instructables.com/ORIG/F4W/1GJL/HW96NITT/F4W1GJLHW96NITT.mov)**IMG\_1930.MOV**](https://cdn.instructables.com/ORIG/F4W/1GJL/HW96NITT/F4W1GJLHW96NITT.mov)

[Download](https://cdn.instructables.com/ORIG/F4W/1GJL/HW96NITT/F4W1GJLHW96NITT.mov)

**Step 7: Learning About Sensors**

[](http://www.instructables.com/file/FZM7LK7HW96NNBI/)

To enable the car to interact in any environment it will need sensors. The sensor I chose was the SRF 05 Arduino Sensor. This sensor has the ability to sense objects up to 2-3000cm away. Below is how to set up the sensor with the Arduino.

1.Download the source

2. Place the DistanceSRF04 folder in your Arduino1.0+ "libraries" folder

3. Open example sketch: "file", "Examples", "DistanceSRF04", "Centimeter" (or "Inch" or "Time")

4. Connect echoPin to pin 2, trigPin to pin 3 (and connect Vcc and GND)

5. Compile & upload code

6. Sensor data should be arriving over the serial port

Here is a photo of my setup:

The source is available below.

* [[DistanceSRF04_v0.1.1.zip](https://cdn.instructables.com/ORIG/F3E/51NO/HW96NMDQ/F3E51NOHW96NMDQ.zip)**DistanceSRF04\_v0.1.1.zip**](https://cdn.instructables.com/ORIG/F3E/51NO/HW96NMDQ/F3E51NOHW96NMDQ.zip)

[Download](https://cdn.instructables.com/ORIG/F3E/51NO/HW96NMDQ/F3E51NOHW96NMDQ.zip)

**Step 8: PWM**

Pulse Width Modulation is an important aspect of programming cars. This allows for you to manually control the amount of power given to certain motor, or more simply control the speed of the car. It changes the code from digital to analog.

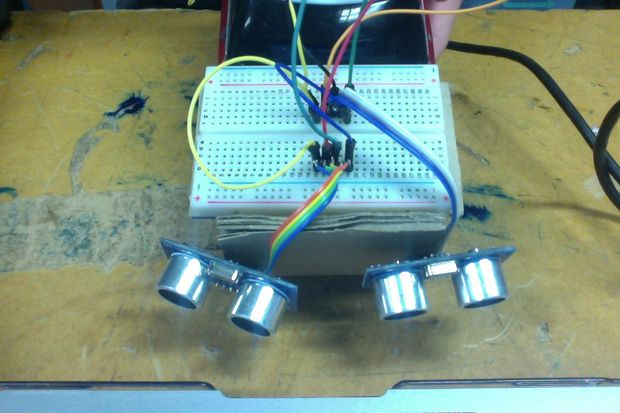
Below is the sample code to input pulse width modulation:

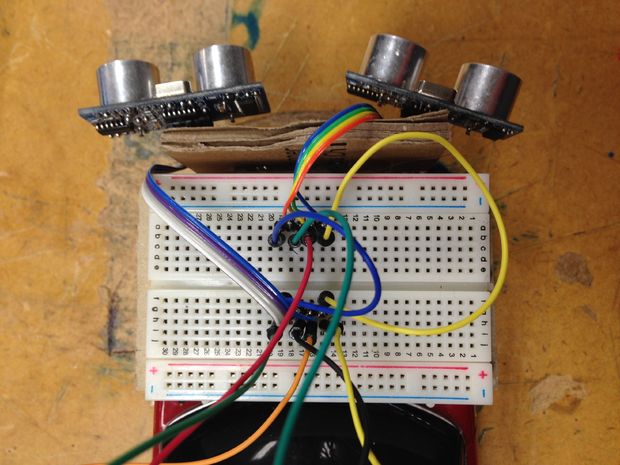
analogWrite(forward, **100**);

digitalWrite(reverse,LOW); // turn revers motor off

This allows you to change the amount of power between 1-255. The number in bold is one that can be

**Step 9: Using Two Sensors Instead of One**

[](http://www.instructables.com/file/F0HS8UPHWJJW1G0/)

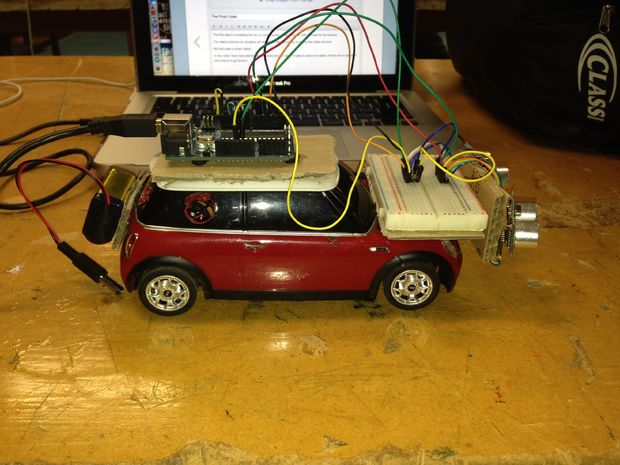
[](http://www.instructables.com/file/F3HXN0VHWJJW1GJ/)

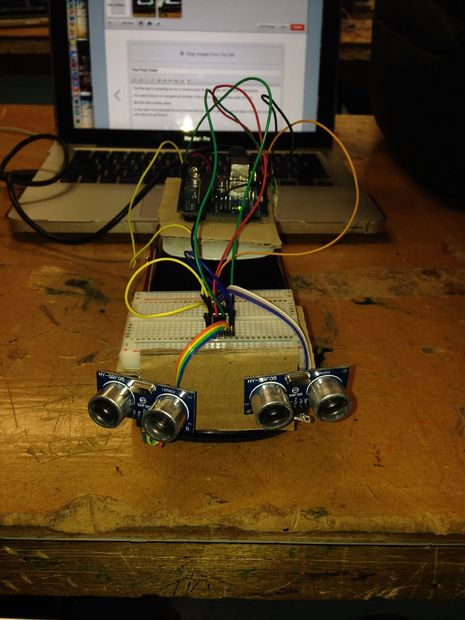
After some basic tests it was seen that the car, at certain angles, couldn't sense objects in front of it. To solve this problem, I have used two sensors, angled on the front of the car. This gives a wider view for the sensors to detect any objects. To do this, the second sensor needed to be added into the code. Below is how this is done:

All you need to do is connect the second sensor into the same GND and VCC pins, but add a second lot of pins for Echo and Trig.

I have used pins 5 and 6 for the additional Echo and Trig ports for the second sensor.

**Step 10: The Final Code**

[](http://www.instructables.com/file/FGTOLPXHWJJW2BO/)

[](http://www.instructables.com/file/F0X8KGWHWJJW2HR/)

The final step to completing the car, is combining the movement code, with the code for the sensors.

You need to ensure you recognise all variables in the void set up, to ensure the codes will work.

My final code is shown below:

In this code I have instructed the car to reverse and turn right, if it gets to close to an object. If there are no objects it will continue to go forward.

* [[Car_Codes_Two_Sensors.ino](https://cdn.instructables.com/ORIG/F0H/7RP5/HWJJW2O5/F0H7RP5HWJJW2O5.ino)**Car\_Codes\_Two\_Sensors.ino**](https://cdn.instructables.com/ORIG/F0H/7RP5/HWJJW2O5/F0H7RP5HWJJW2O5.ino)

[Download](https://cdn.instructables.com/ORIG/F0H/7RP5/HWJJW2O5/F0H7RP5HWJJW2O5.ino)

Top of Form