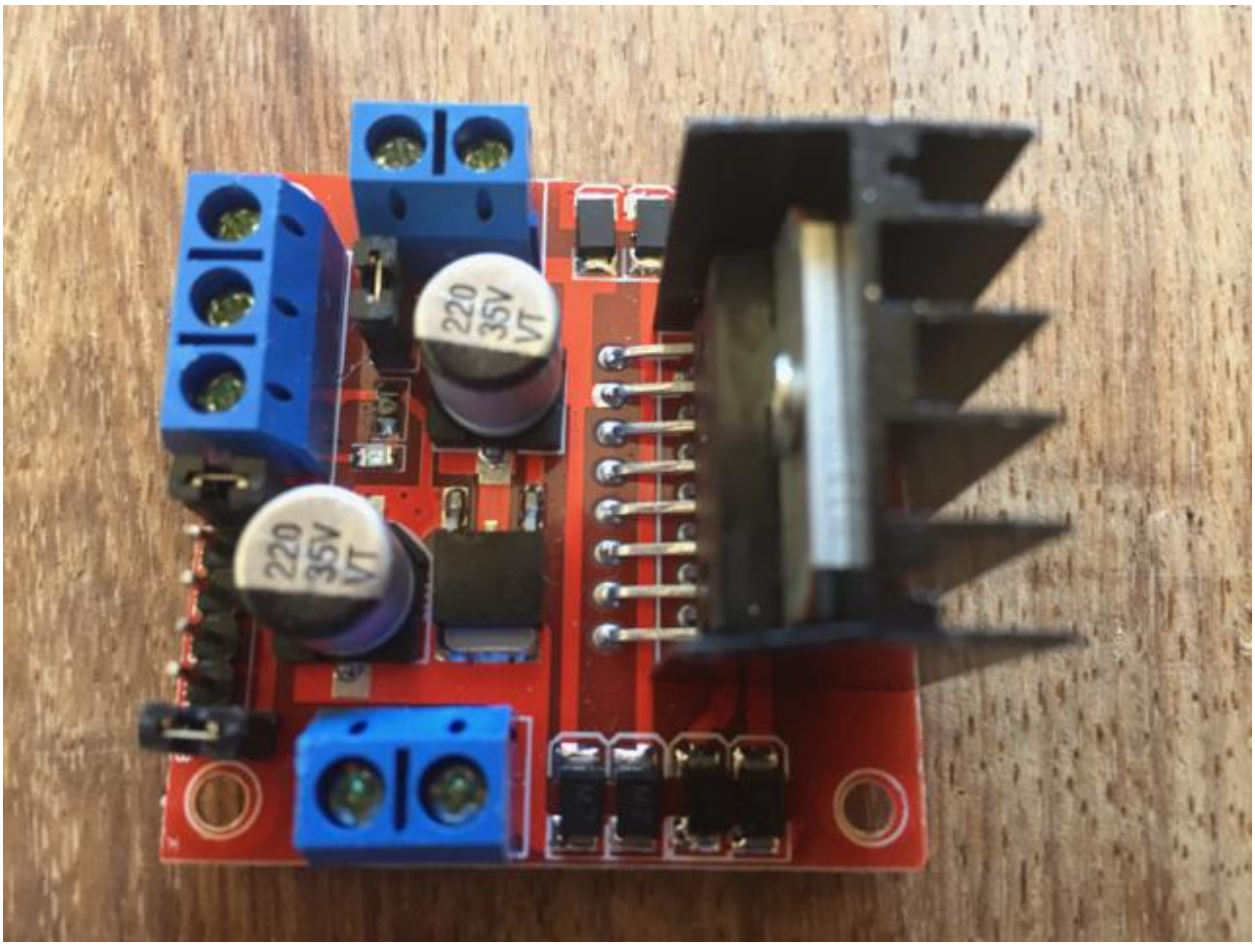
Robot Assembly Guide Beginner

You should by now have an assembled chassis for your robot. The next step is to mount each component one by one and then test each component. This guide instructs you to mount each part and then test them, however your instructor may have other ideas. Treat this as a general guide and not the single source of truth. That’s what your instructors are for

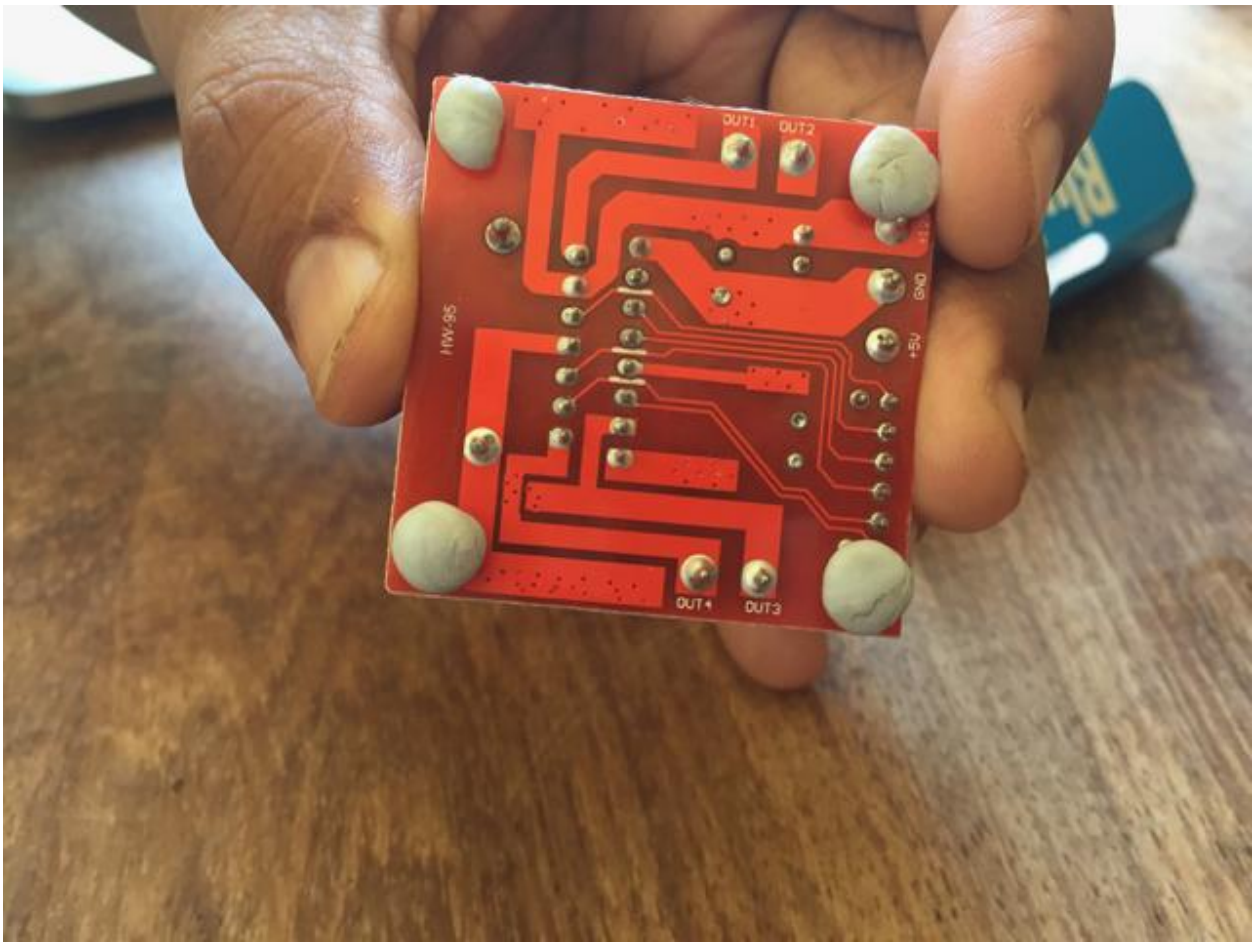
Mounting the L298N Motor Driver

Your instructors will give you the motor driver it looks like this:

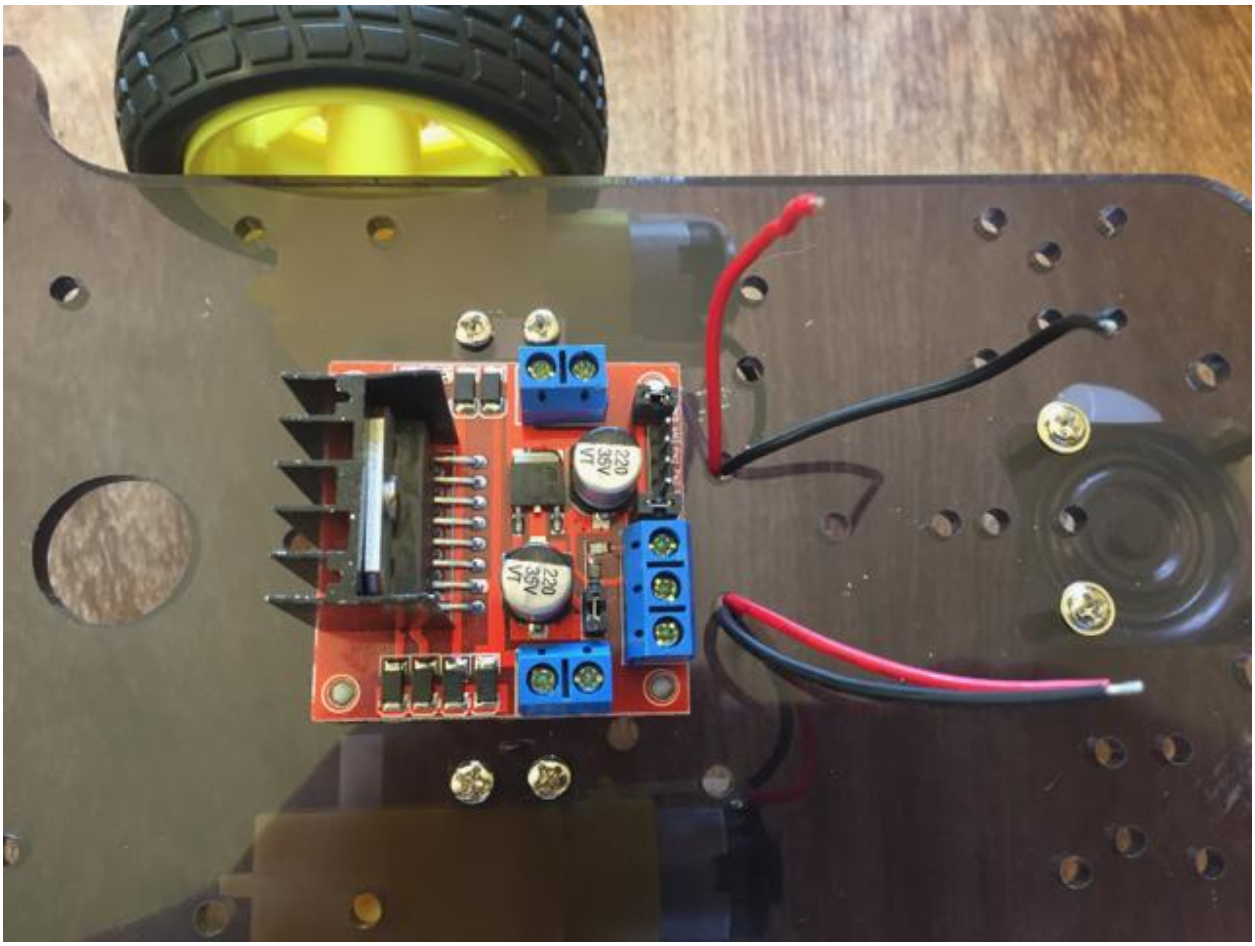


Essentially you will blue tack it down. Please be careful when you do this, taking care to always hold the motor driver on the edges of the board and not exerting pressure on any of the components on the board.

Note that your instructor may ask you to connect the wires from the motors up at this point also. Photos can be found on page 14 of this guide.

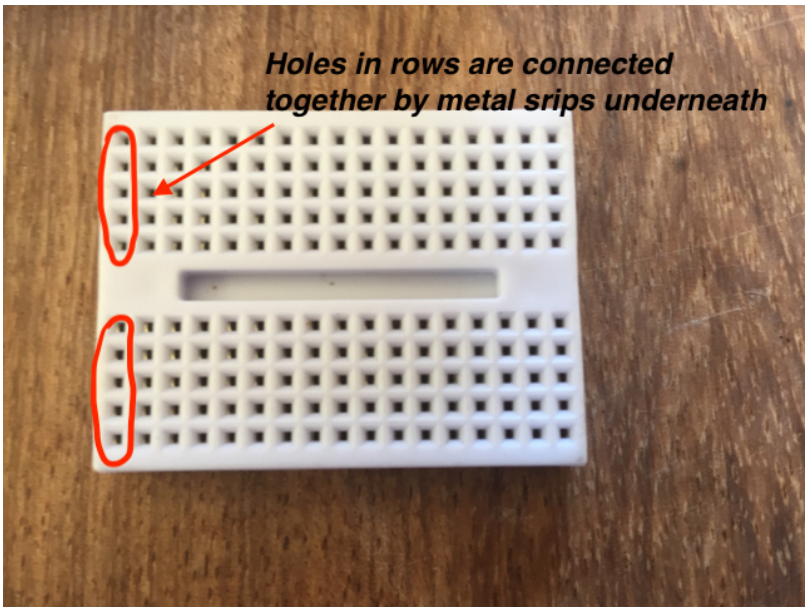


The next step is to get some bluetack (most probably from your instructors!) and attach it to the underside like so!



If your motor wires are not pulled through at this point, pull them through so that it looks like the photo above.

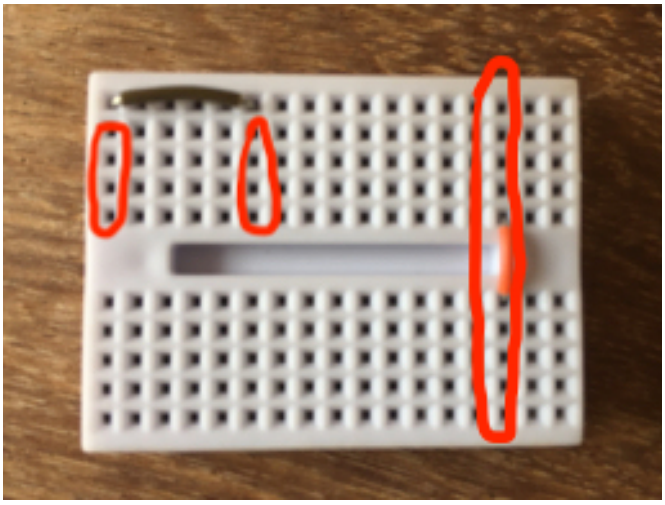
*** It may be a good time to attach the wires to the motor driver. This will enable the motor driver to ‘control’ the motors via the Arduino. This depends on whether your instructor decides that it is a good time to do it! ***



This is a breadboard. This is the next item that we are going to attach to our chassis. You will see that it is full of holes - this enables us to connect components together with wires. The useful thing about breadboards is that each row of holes are connected together underneath with metal strips that you can't see.

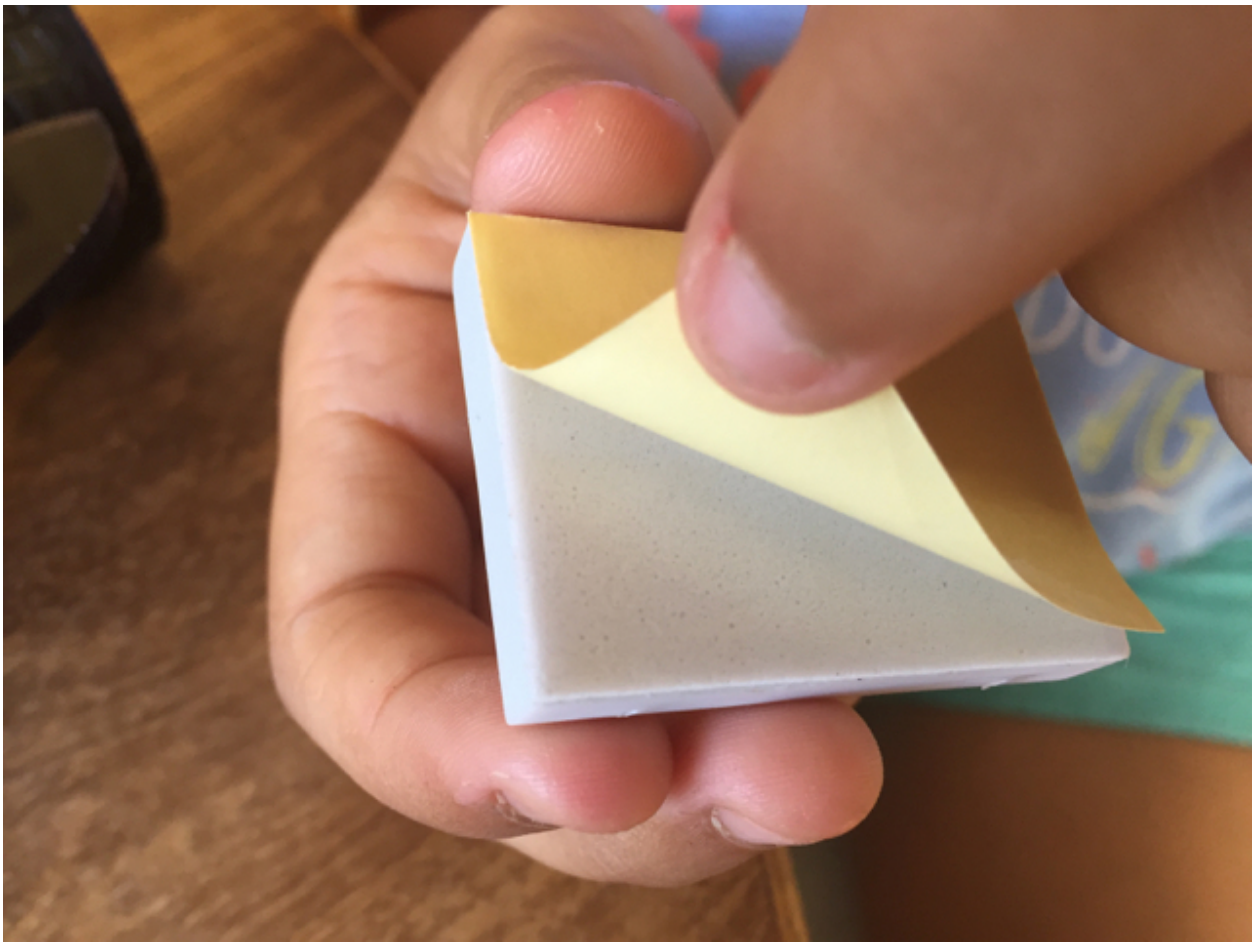
You can see that the breadboard has 2 sides - they are not connected to each other.

We can connect rows together and even the 2 sides together like this with jumper wires:

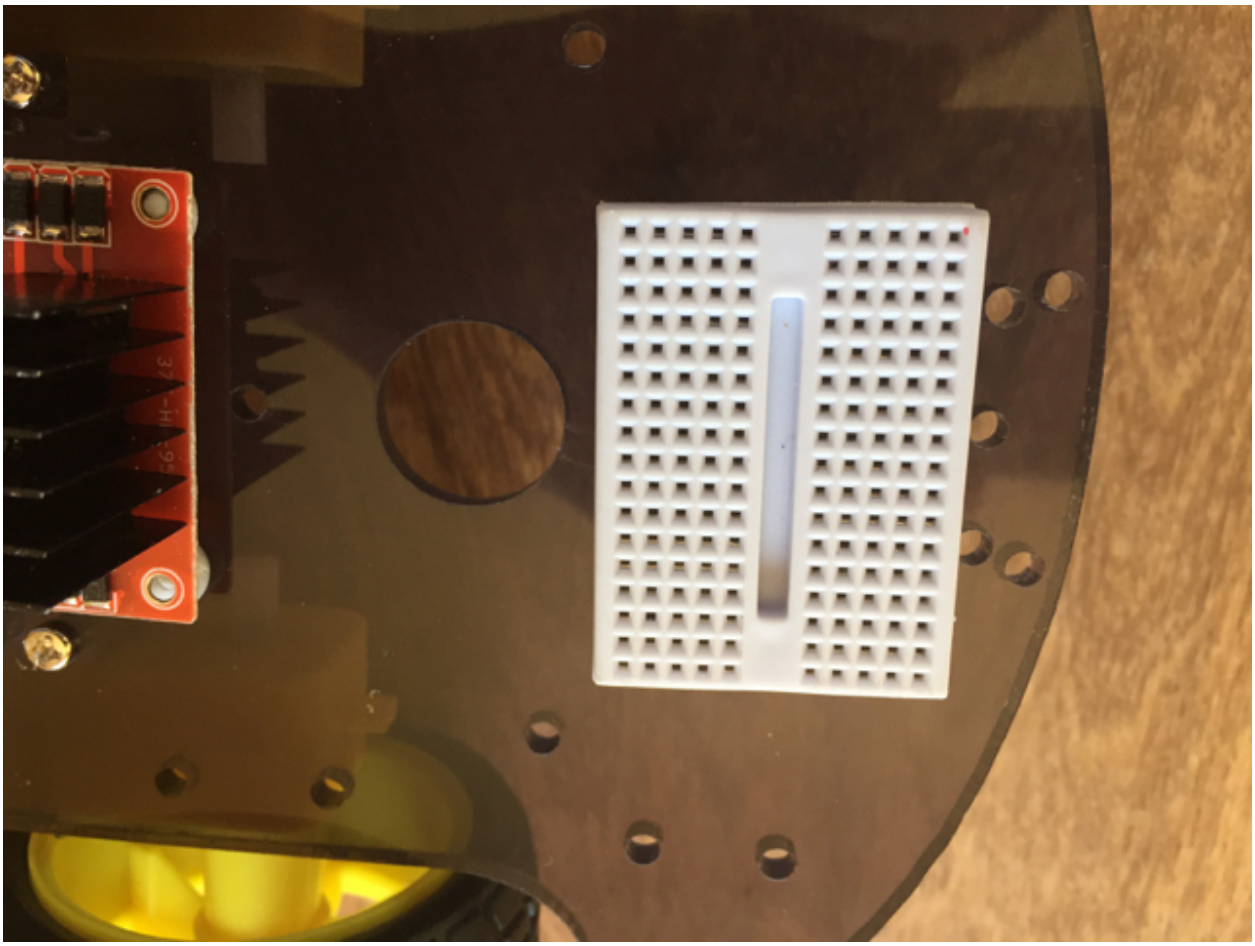


If you look at the underside of the breadboard you will see that it has a backing that can be peeled off. You will need to peel this off and stick this to the front of your chassis. HOWEVER we suggest that you DO NOT do this straightaway and use blue tack instead. Once your robot is

assembled completely, you can stick it down again properly. Blue tack gives you the option of re-attaching it. The sticky breadboard is so sticky that when you stick it down, you won't get it off again - therefore if you stick it down wrong - you may have a problem !



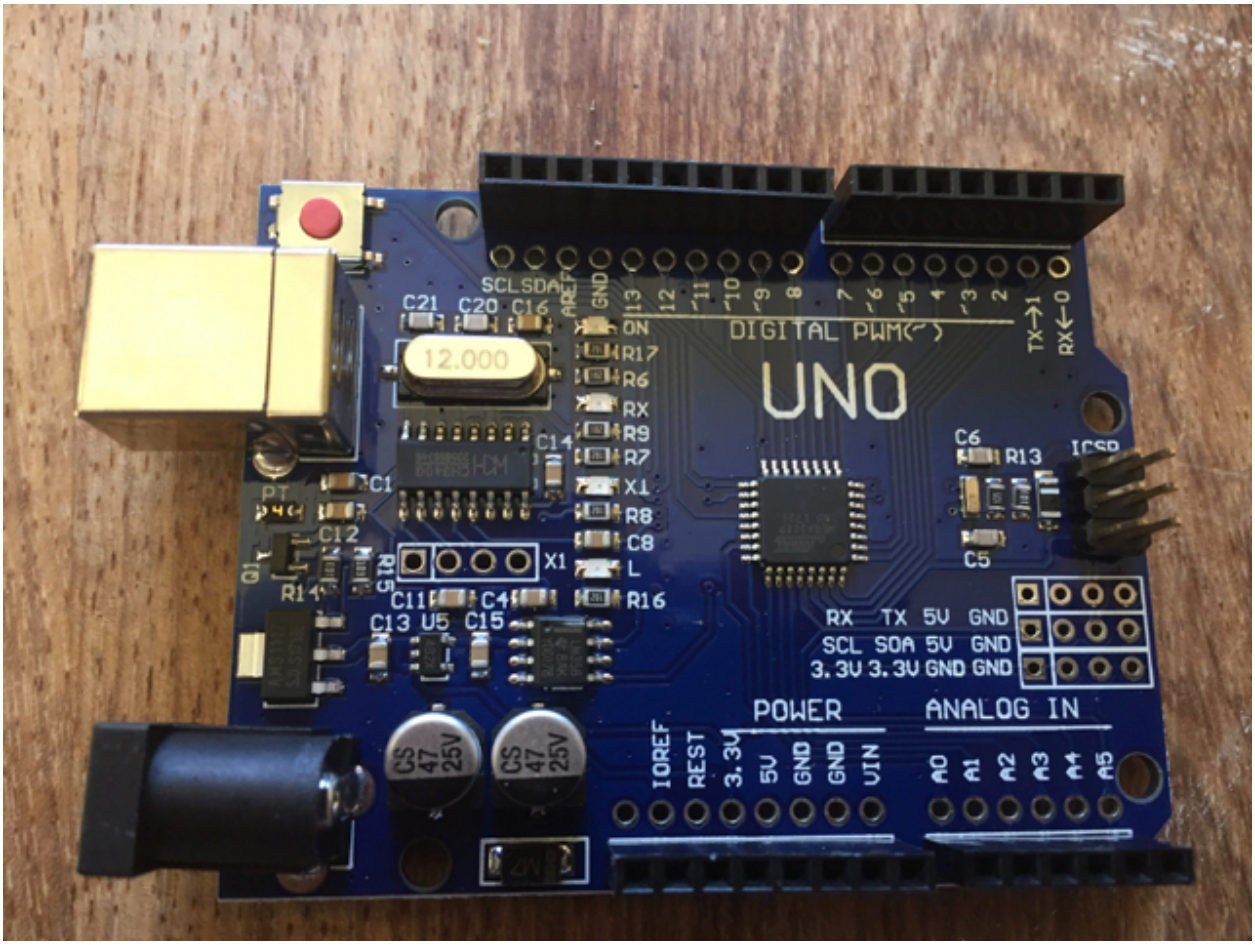
This is the breadboard stuck down at the front of the chassis.



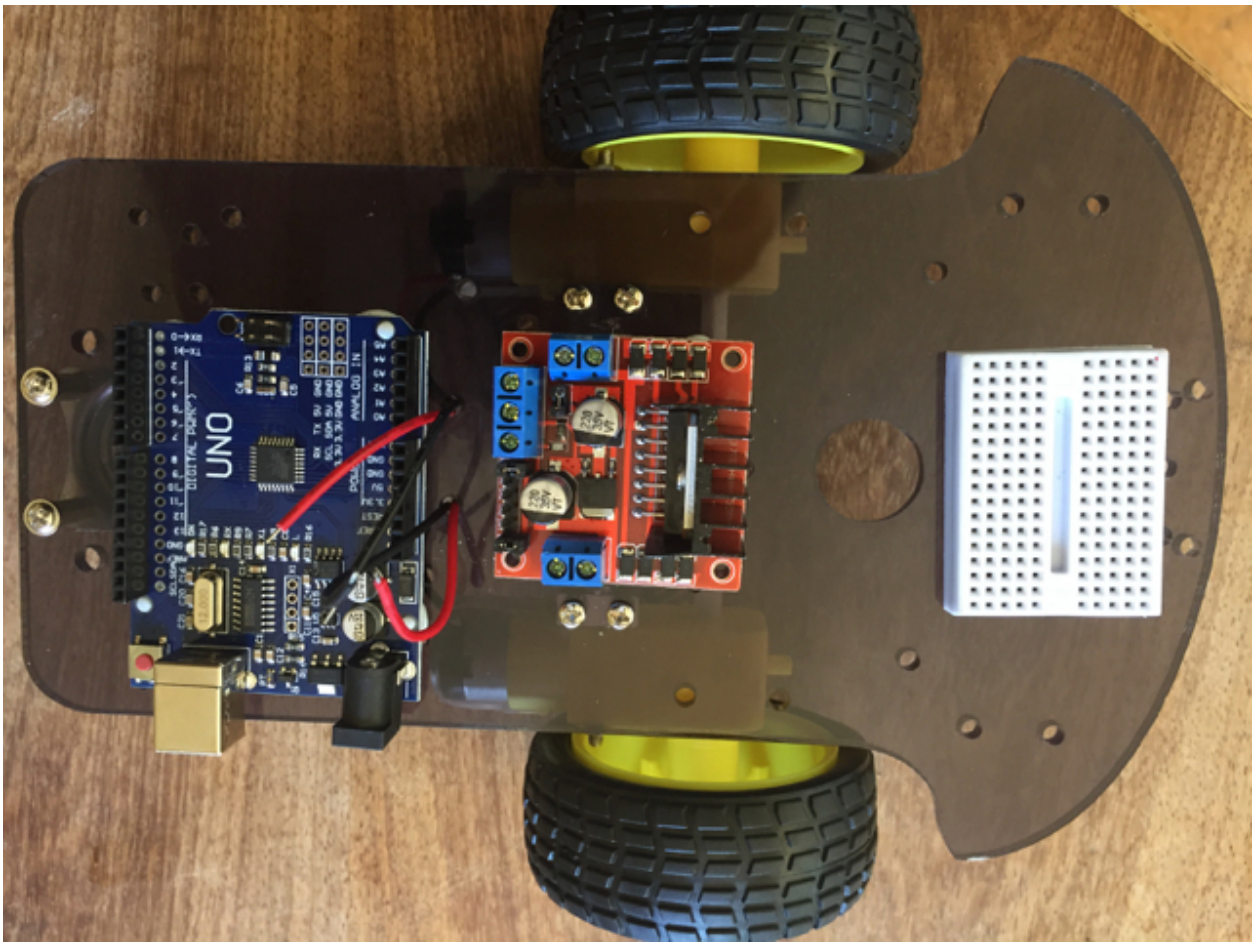
Next, we will be mounting the Arduino. The Arduino is effectively the ‘brain’ of your robot. Whilst it is not a fully fledged computer, it is like a basic computer that allows you to send input and read output from different devices.. Such as our motor driver above!

Please make sure that your instructors have handed out Arduinos to the class (obviously!) before performing this step.

We will be attaching blue tack to the underside as before and sticking the Arduino down onto the chassis. The only thing that you need to ensure you get right is the mounting position as you will need access to the USB port (large silver block below) so that you can upload code from your computer.

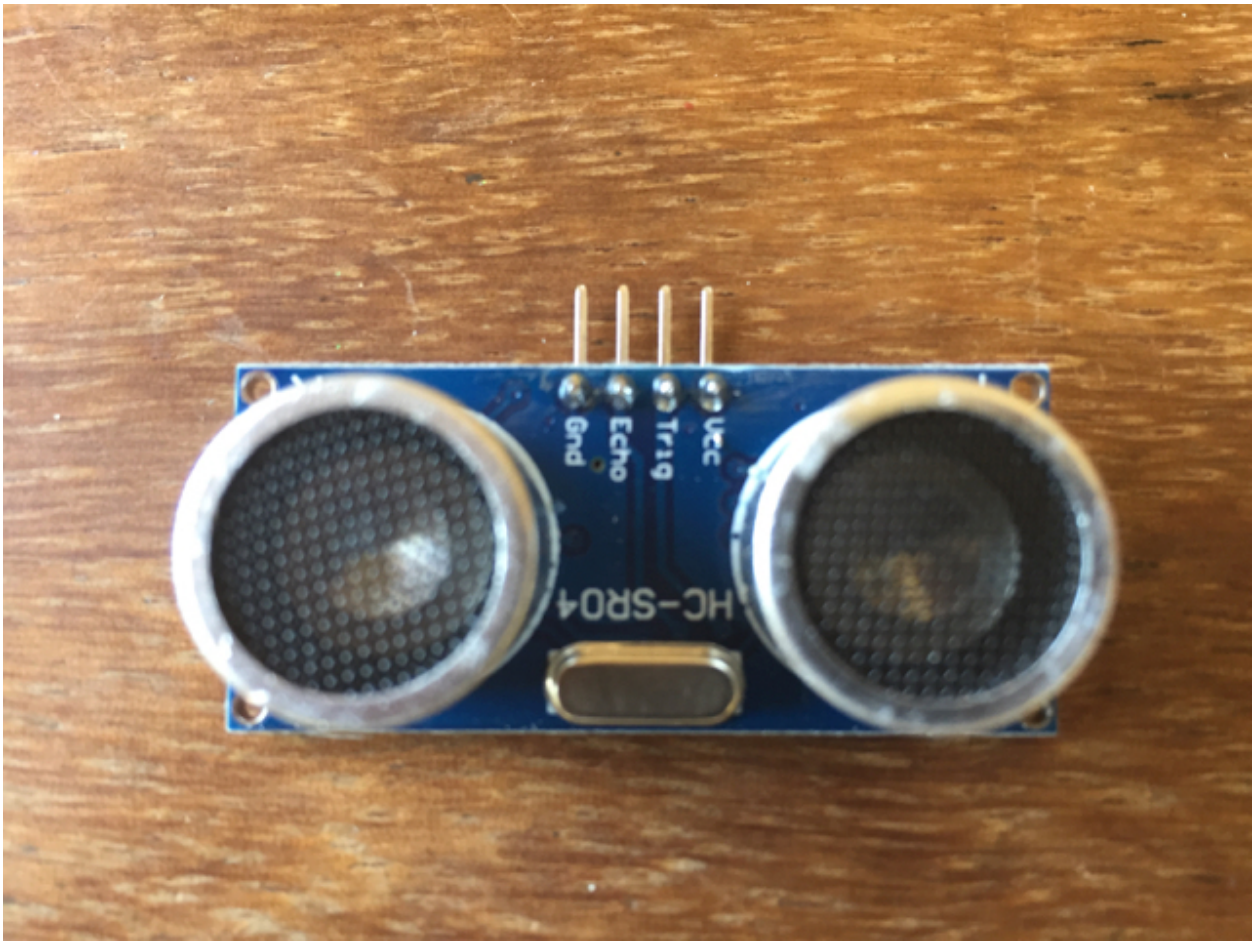


You can’t quite see it but we have blue tacked the underside 4 corners and stuck the Arduino down. Again be very careful that you are not putting pressure on the components/pins of the Arduino when you stick it down.



This is what you should have so far.

The next task is to mount the Ultrasonic sensor. This is what it looks like:

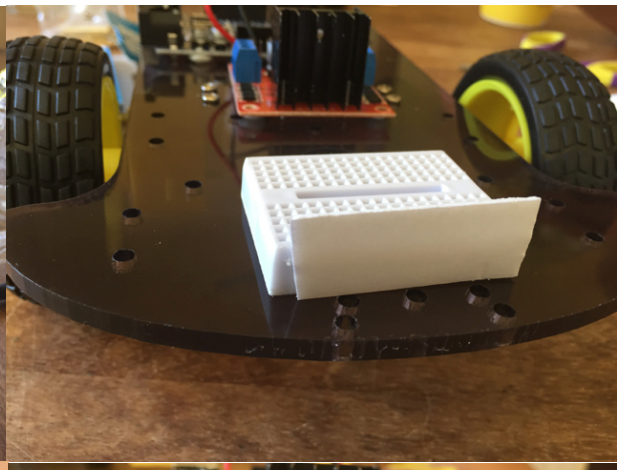


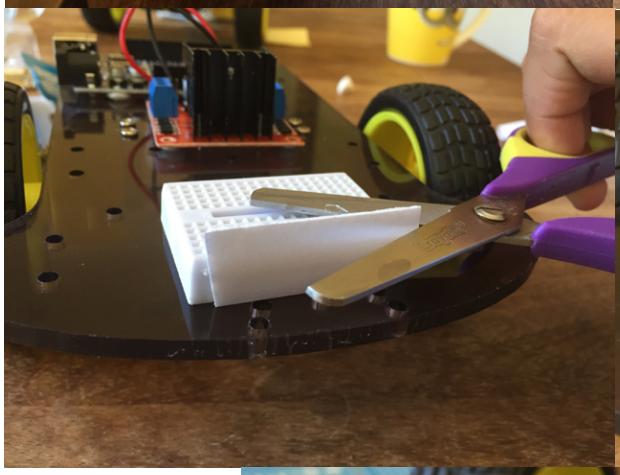
This is what you are about to do….

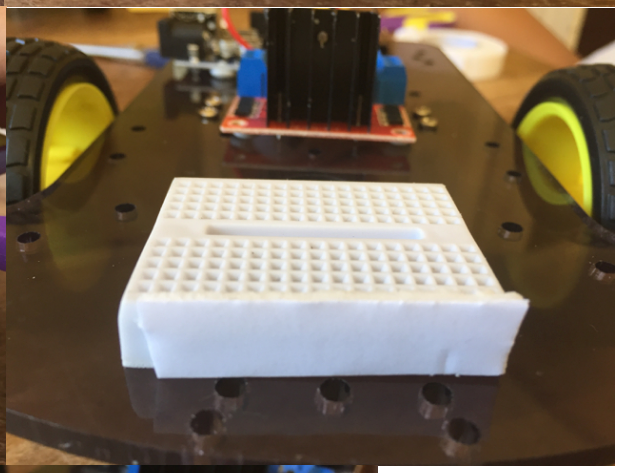
Mount some double sided sticky foam/tape to the front of your chassis on the side of the breadboard.

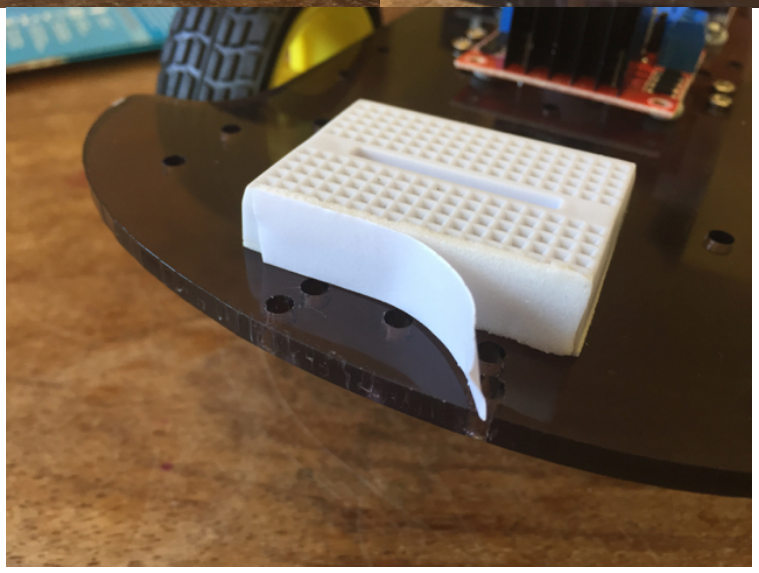
Attach/stick the sensor to the breadboard so that it looks like your robot has a pair of eyes!!

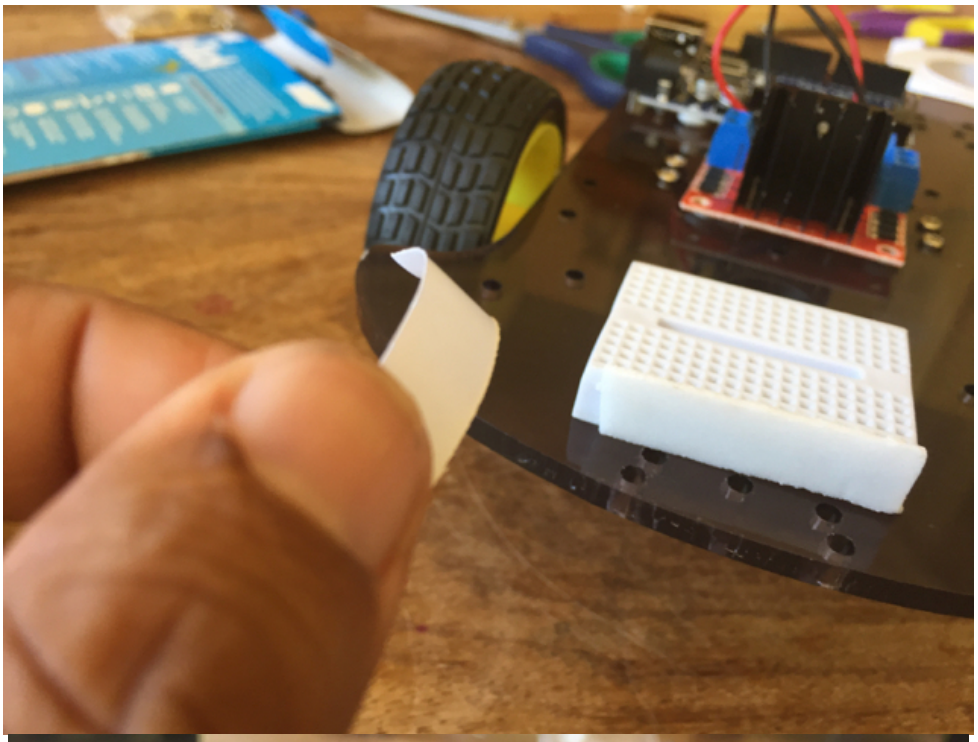
Check out the next few photos that illustrate this.

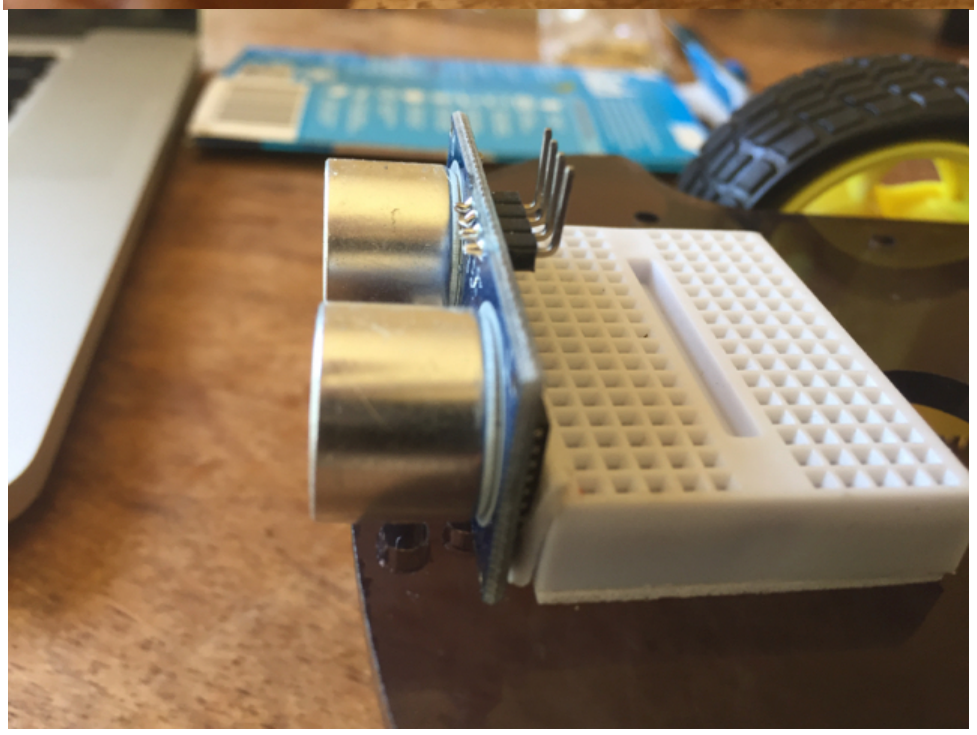




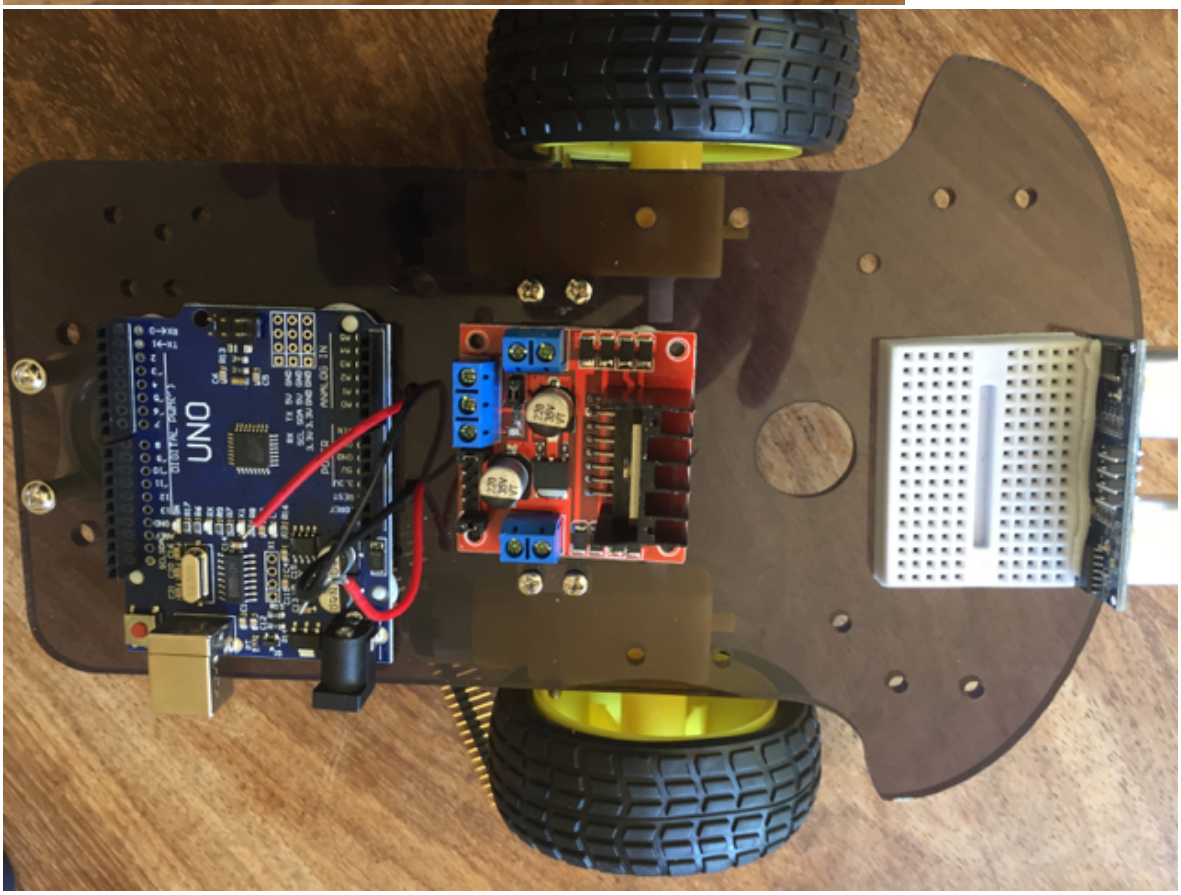






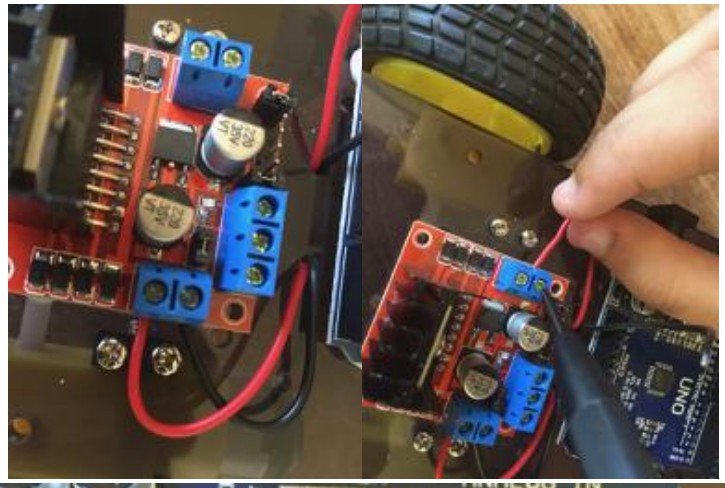


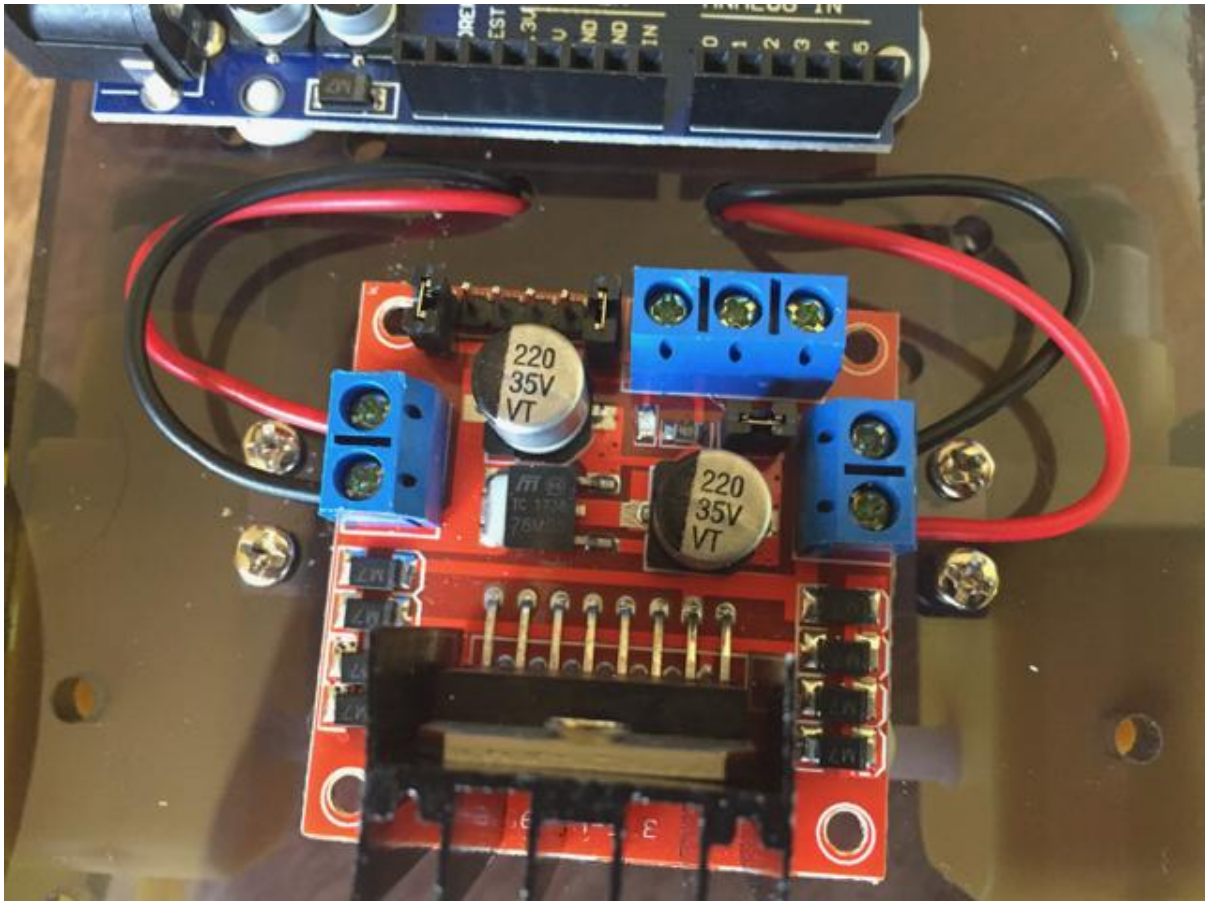
Your robot should now look a bit (or even a lot like this):



Wiring up the Motors

Wiring up the motors is pretty simple. You will need to get yourself a phillips screwdrive (cross-head screwdriver) because the red and black wires from each of your motors must be inserted into the motor driver terminals and screwed down. See below:

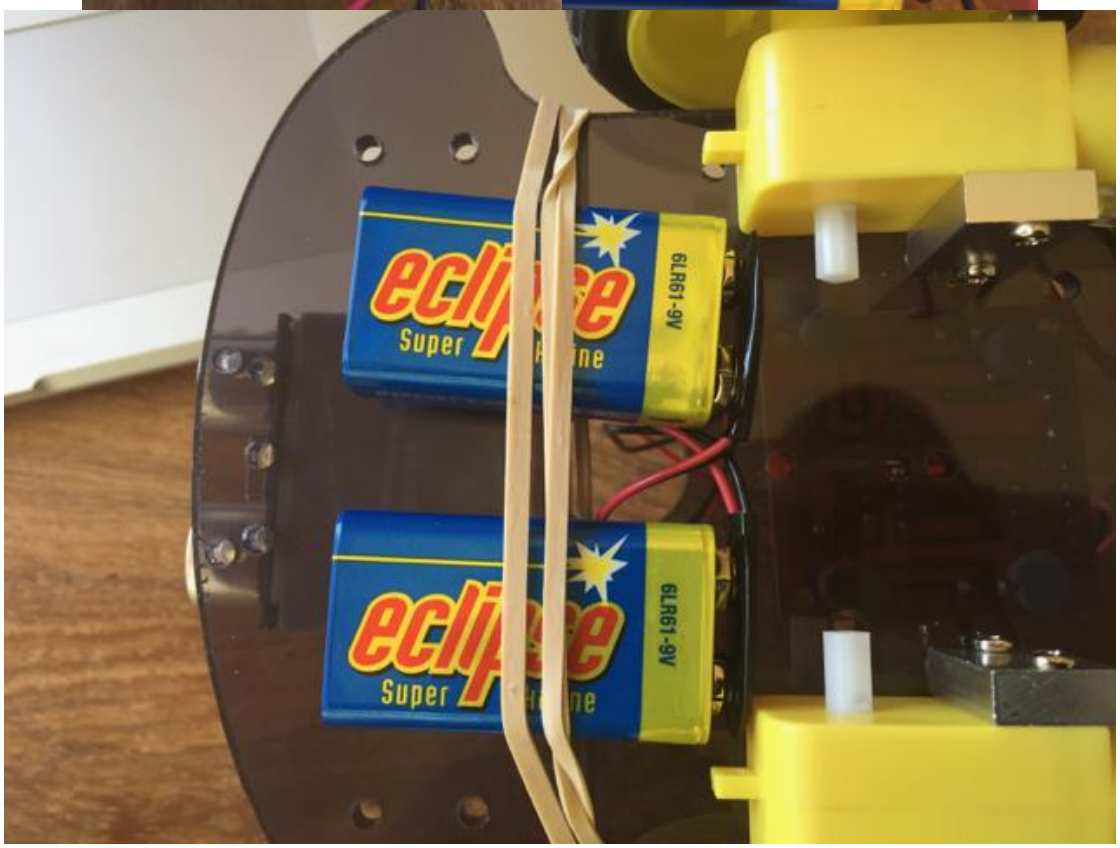




Mounting the Batteries

The next step is to mount our batteries so that we can power the motor driver and the Arduino. This is what you need. You will be attaching the batteries to the underside of the chassis at the front. *** NOTE THAT YOU MAY GET A DIFFERENT TYPE OF BATTERY HOLDER TO HOLD AA BATTERIES. SIMILAR INSTRUCTIONS WILL STILL APPLY ***

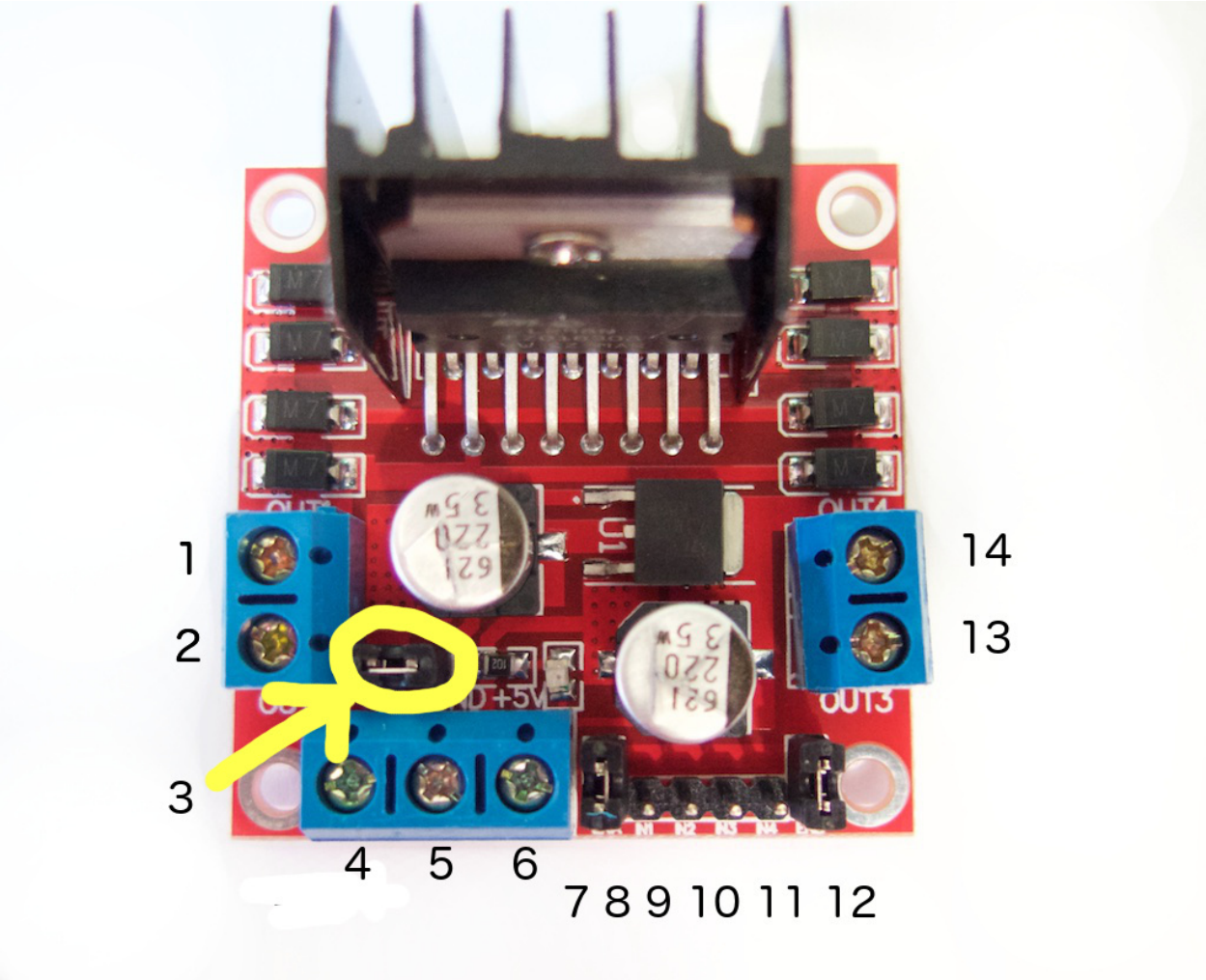
Attach the two battery connectors to the batteries (the AA holder will already have attached wires. You will just need to put the AA batteries in) and get 2 blobs of blue tack to initially secure the batteries to the underside of the chassis (front/middle so that the wires can go through the large hole in the middle).



Here is a picture of the other side with the wires pulled through!

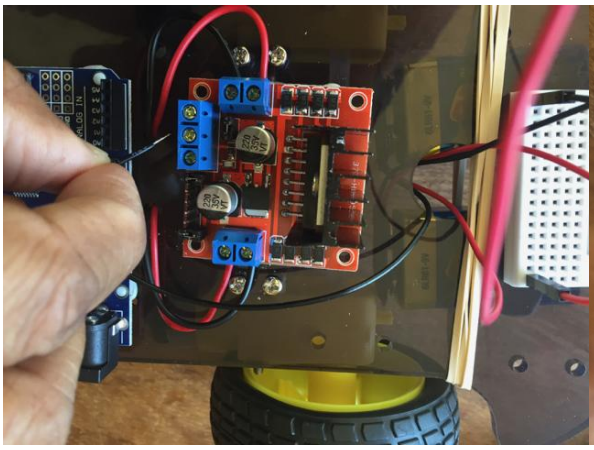
Add Power to the Motor Driver

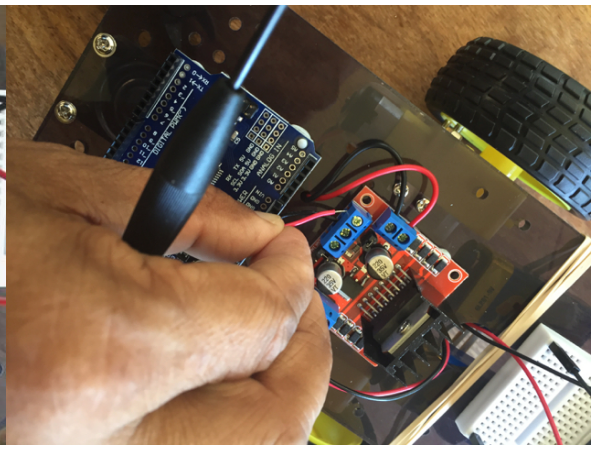
Now that you have your batteries installed, you can add power to the motor driver. For this, you will need to connect the negative (black) wire of one battery to the ground terminal 5 of the motor driver and the positive (red) wire to the supply voltage terminal 4 of the motor driver. See below:

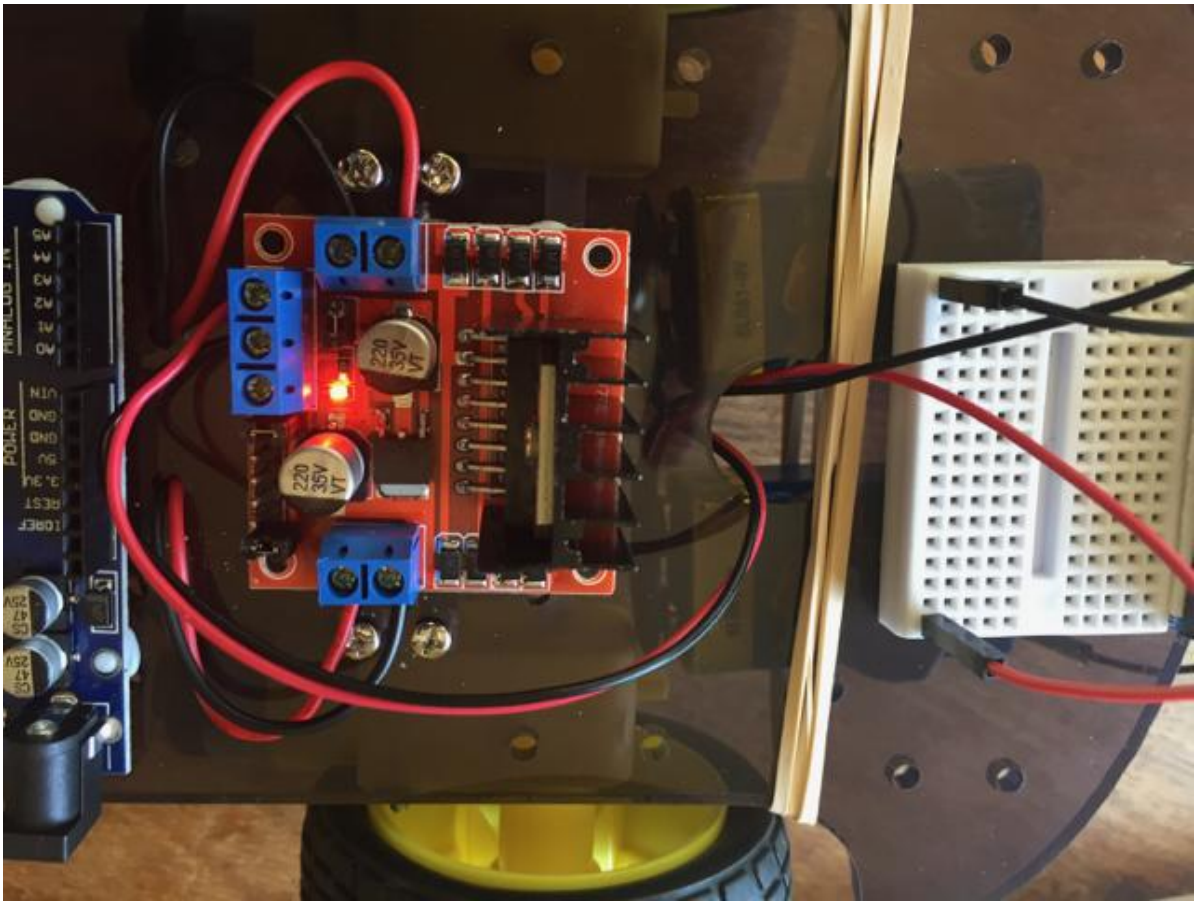


*** NOTE - you may actually connect the battery connector to the breadboard instead. In order to do this, you will need to create a ground rail and a power rail on your breadboard. Your instructors can show you how to do this and may well get the group to perform this step together ***

The next 3 photos show the battery black and red connector wires being attached to the motor driver:

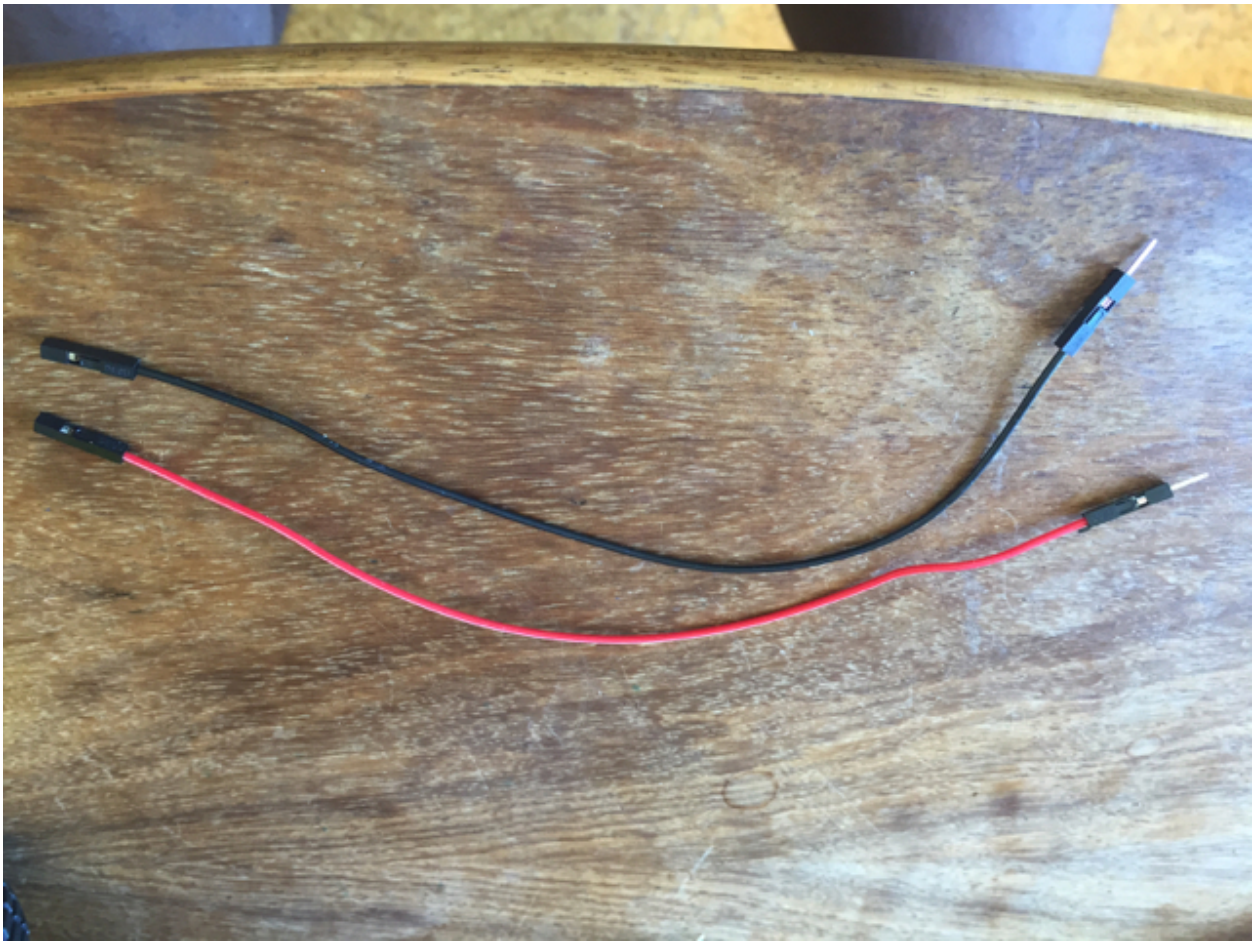




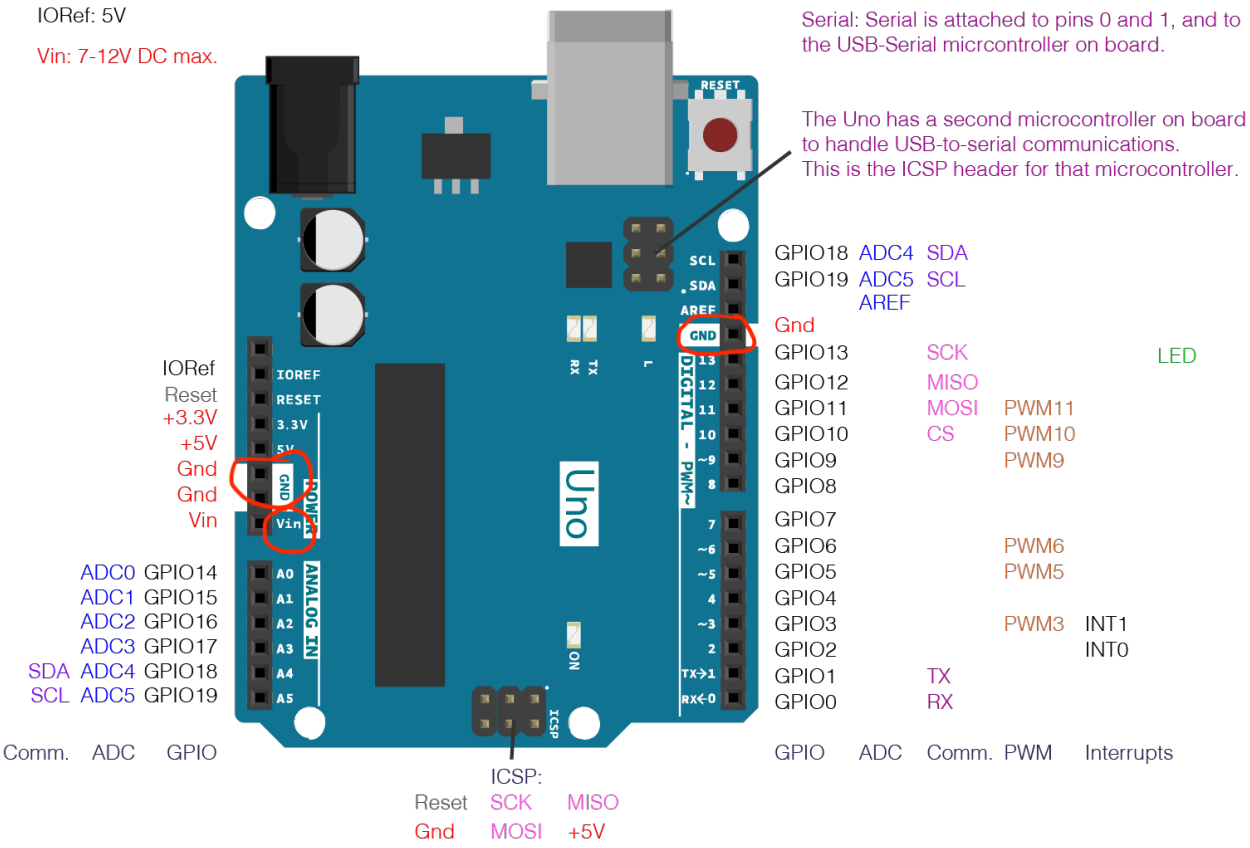


Powering the Arduino

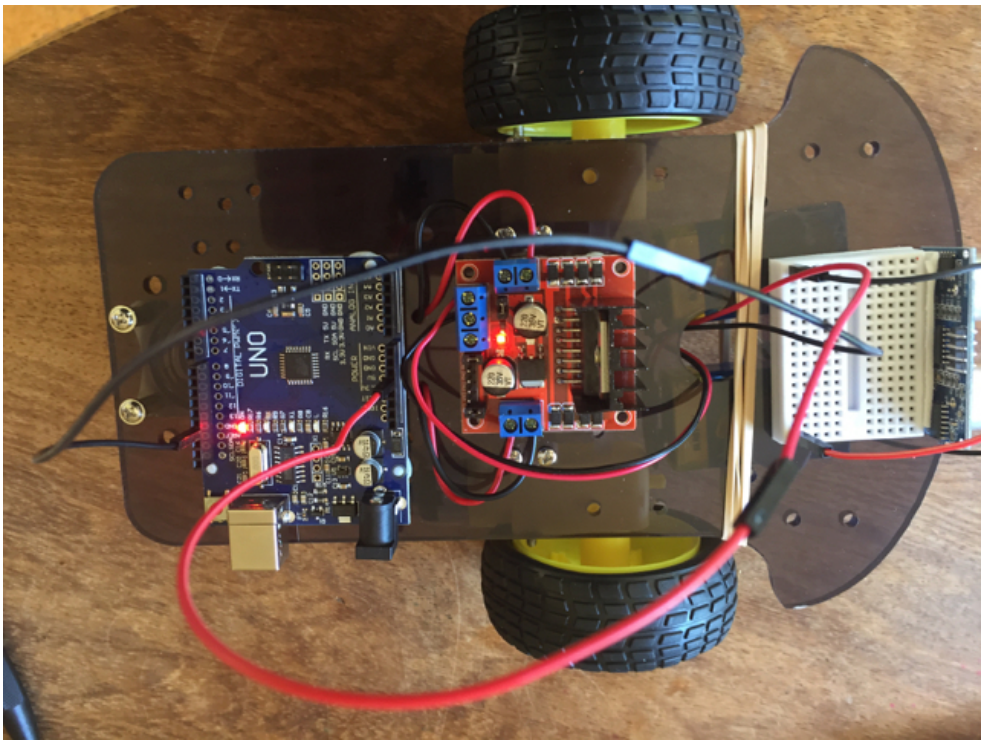
The next step is to use the remaining 2 battery leads to power the Arduino. You will find that the leads are two short so you will need to extend them with 2 jumper wires (Female - Male) like so:



Then you will connect the black lead to a GND (ground) input on the Arduino and the red lead to the VIN (voltage in) input on the Arduino. Please see the Arduino diagram on the next page.

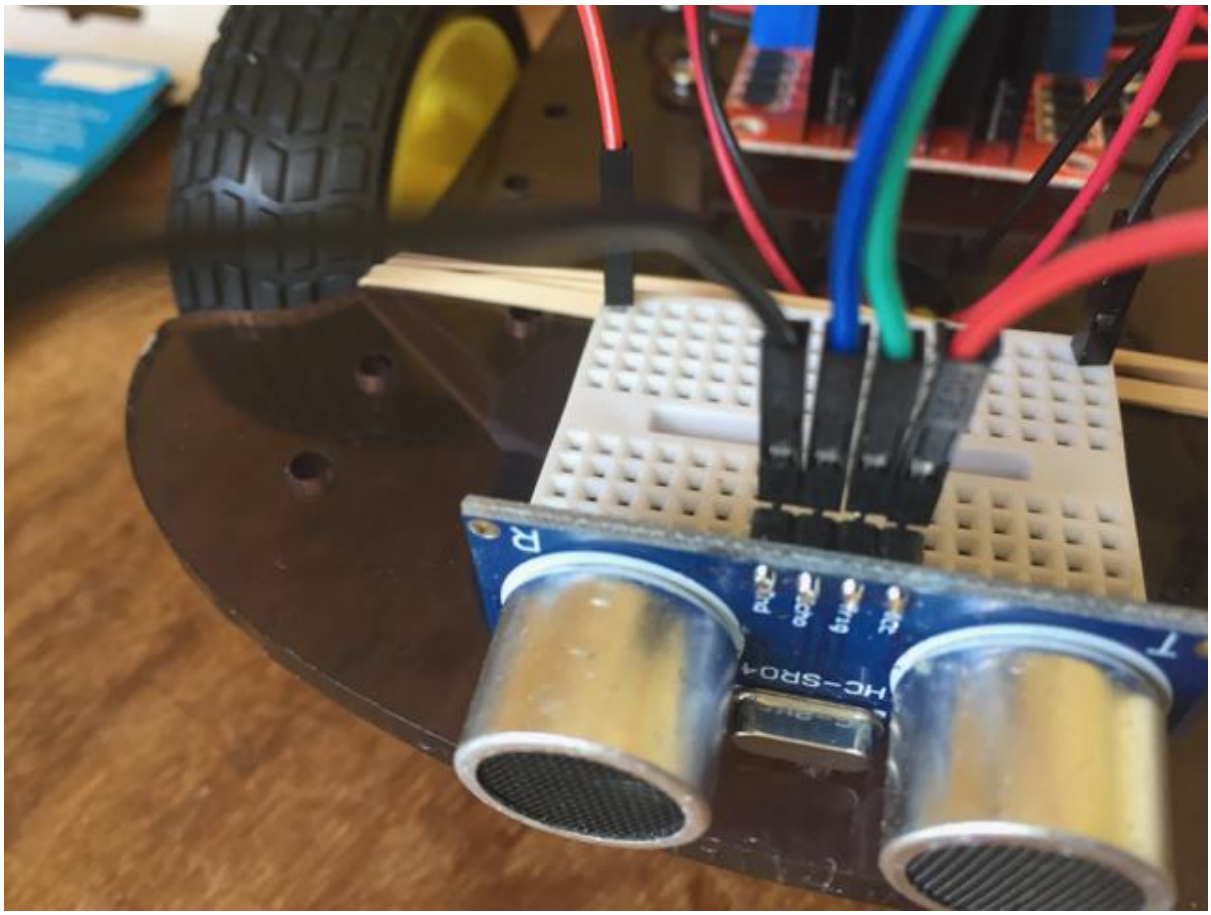


The next photo shows that battery wired up:



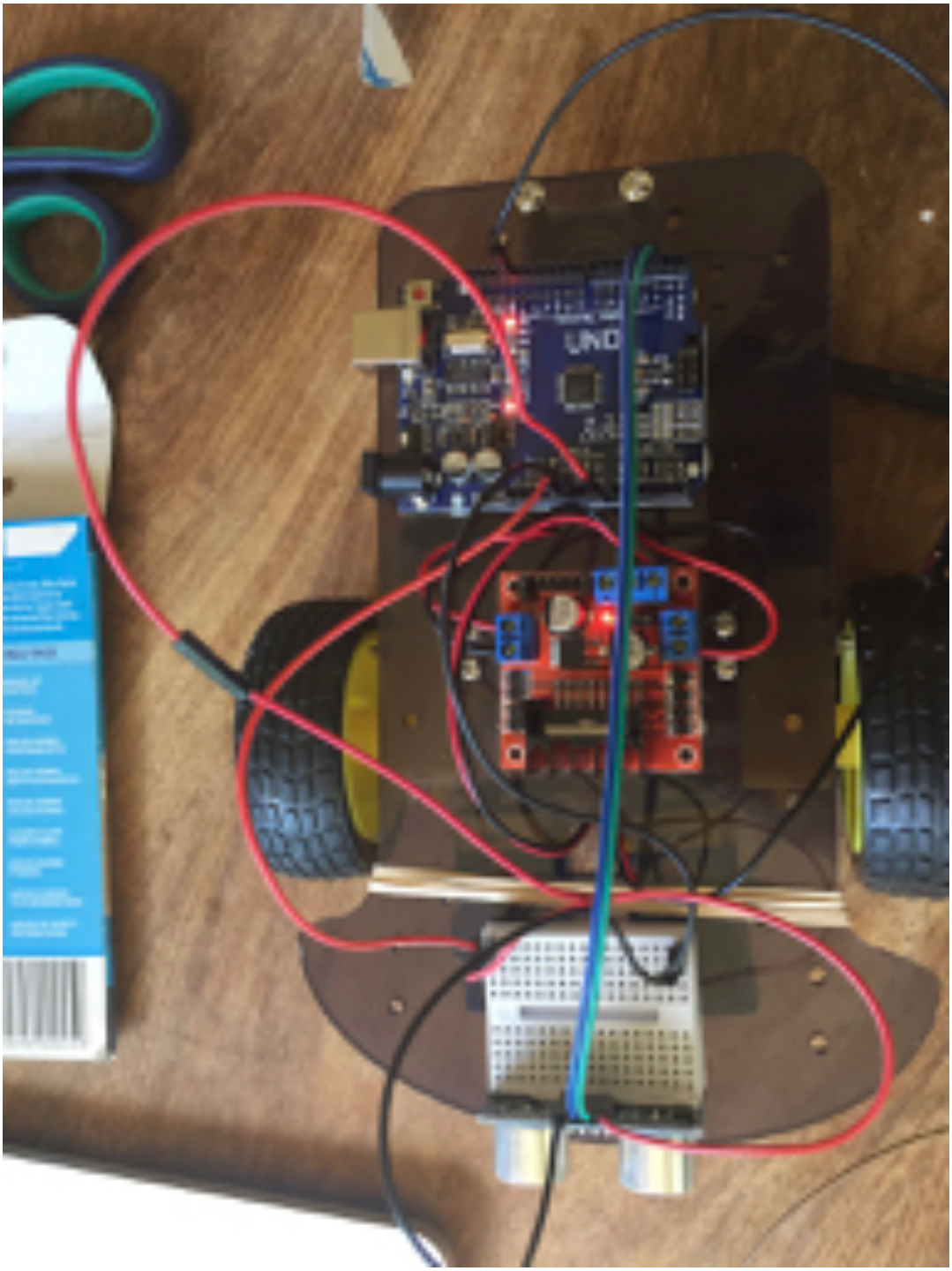
Last step before testing - Wiring up the Ultrasonic Sensor

You will need at least 4 jumper wires for the ultrasonic sensor (Female to Male). We will let your instructors tell you how to wire this up as there are at least 2 ways you could do it. If there are enough jumper leads you could make good use of the breadboard here by creating a ground rail and a power rail.



Here is a picture of a wired up robot. Yours may look a little different, however the result will be the same. The next step will be to test each part of the robot ie. motor driver and sensor.

This is the coding step!



Appendix A - Code to test the Motor Driver

```
//Motor A
const int motorPin1  = 9;  // Pin 14 of L293
const int motorPin2  = 10;  // Pin 10 of L293
//Motor B
const int motorPin3  = 6; // Pin  7 of L293
const int motorPin4  = 5;  // Pin  2 of L293

//This will run only one time.
void setup(){

    //Set pins as outputs
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);

    //Motor Control - Motor A: motorPin1,motorpin2 & Motor B: motorpin3,motorpin4

    //This code  will turn Motor A clockwise for 2 sec.
    analogWrite(motorPin1, 180);
    analogWrite(motorPin2, 0);
    analogWrite(motorPin3, 180);
    analogWrite(motorPin4, 0);
    delay(5000);
    //This code will turn Motor A counter-clockwise for 2 sec.
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 180);
    analogWrite(motorPin3, 0);
    analogWrite(motorPin4, 180);
    delay(5000);

    //This code will turn Motor B clockwise for 2 sec.
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 180);
    analogWrite(motorPin3, 180);
    analogWrite(motorPin4, 0);
    delay(1000);
    //This code will turn Motor B counter-clockwise for 2 sec.
    analogWrite(motorPin1, 180);
    analogWrite(motorPin2, 0);
    analogWrite(motorPin3, 0);
    analogWrite(motorPin4, 180);
    delay(1000);

    //And this code will stop motors
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 0);
    analogWrite(motorPin3, 0);
    analogWrite(motorPin4, 0);

}
void loop(){
}
```

Appendix B - Code to test the Ultrasonic Sensor

```
/*
* Ultrasonic Sensor HC-SR04 and Arduino Tutorial
```

```
*
* Crated by Dejan Nedelkovski,
* www.HowToMechatronics.com
*
*/
// defines pins numbers
const int trigPin = 2;
const int echoPin = 3;

// defines variables
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;

  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```

Appendix C - Code to test the Analogue Sound Sensor

```
void setup(){
  Serial.begin(9600); // open serial port, set the baud rate to 9600 bps
}

void loop(){
      int val;
      val=analogRead(0);   //connect mic sensor to Analog 0
      Serial.println(val,DEC);//print the sound value to serial
      delay(100);
}
```

Appendix D - Integrated Code to help get your robot working

```
// Motor A
const int motorPin1  = 9;  // Pin 14 of L293
const int motorPin2  = 10;  // Pin 10 of L293
// Motor B
const int motorPin3  = 6; // Pin  7 of L293
const int motorPin4  = 5;  // Pin  2 of L293

// Defines ultrasonic pins numbers
const int trigPin1 = 2;
const int echoPin1 = 3;

// Ultrasonic variables
long duration;
int distance;
long counter = 0;

// 0 = stopped
// 1 = moving
int robotState = 0;

void setup() {
  Serial.begin(9600); // open serial port, set the baud rate to 9600 bps

  //Set pins as outputs
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);

  pinMode(trigPin1, OUTPUT); // Sets the trigPin on forward sensor as an Output
  pinMode(echoPin1, INPUT); // Sets the echoPin on forward sensor as an Input

  forward();
}

void loop() {
  if(++counter % 10 == 0){
    int currentDistanceAhead = getDistance(trigPin1, echoPin1);
    Serial.print("Distance ahead: ");
    Serial.println(currentDistanceAhead);

    if(currentDistanceAhead < 50) {
      halt();
      delay(200);
      Serial.print("Counter: ");
      Serial.println(counter);
      int d = counter % 20;
      Serial.println(d);
      if(counter % 20 == 0){
        right();
```

```
        forward();
      }
      else{
        reverse(2000);
        left();
        forward();
      }
    }
  }

  // Delay of 100 milliseconds before next loop is run
  delay(100);
}


// Speed of motors can be controlled by the range 0-255
// 255 is the fastest, 0 means stopped.
void forward() {
    //This code will turn Motor B clockwise for 2 sec.
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 160);
    analogWrite(motorPin3, 180);
    analogWrite(motorPin4, 0);
}

void reverse(int seconds) {
    analogWrite(motorPin1, 180);
    analogWrite(motorPin2, 0);
    analogWrite(motorPin3, 0);
    analogWrite(motorPin4, 180);
    delay(seconds);
}

void halt() {
  //And this code will stop motors
  analogWrite(motorPin1, 0);
  analogWrite(motorPin2, 0);
  analogWrite(motorPin3, 0);
  analogWrite(motorPin4, 0);
}

void right() {
    //This code  will turn Motor A clockwise for 2 sec.
    analogWrite(motorPin1, 180);
    analogWrite(motorPin2, 0);
    analogWrite(motorPin3, 180);
    analogWrite(motorPin4, 0);
    delay(250);
}

void left() {
    //This code will turn Motor A counter-clockwise for 2 sec.
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 180);
    analogWrite(motorPin3, 0);
```

```
    analogWrite(motorPin4, 180);
    delay(250);
}


int getDistance(int tPin, int ePin) {
  // Clears the trigPin
  digitalWrite(tPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(tPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(tPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(ePin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;
  return distance;
}
```

Appendix E - Checklist of Parts

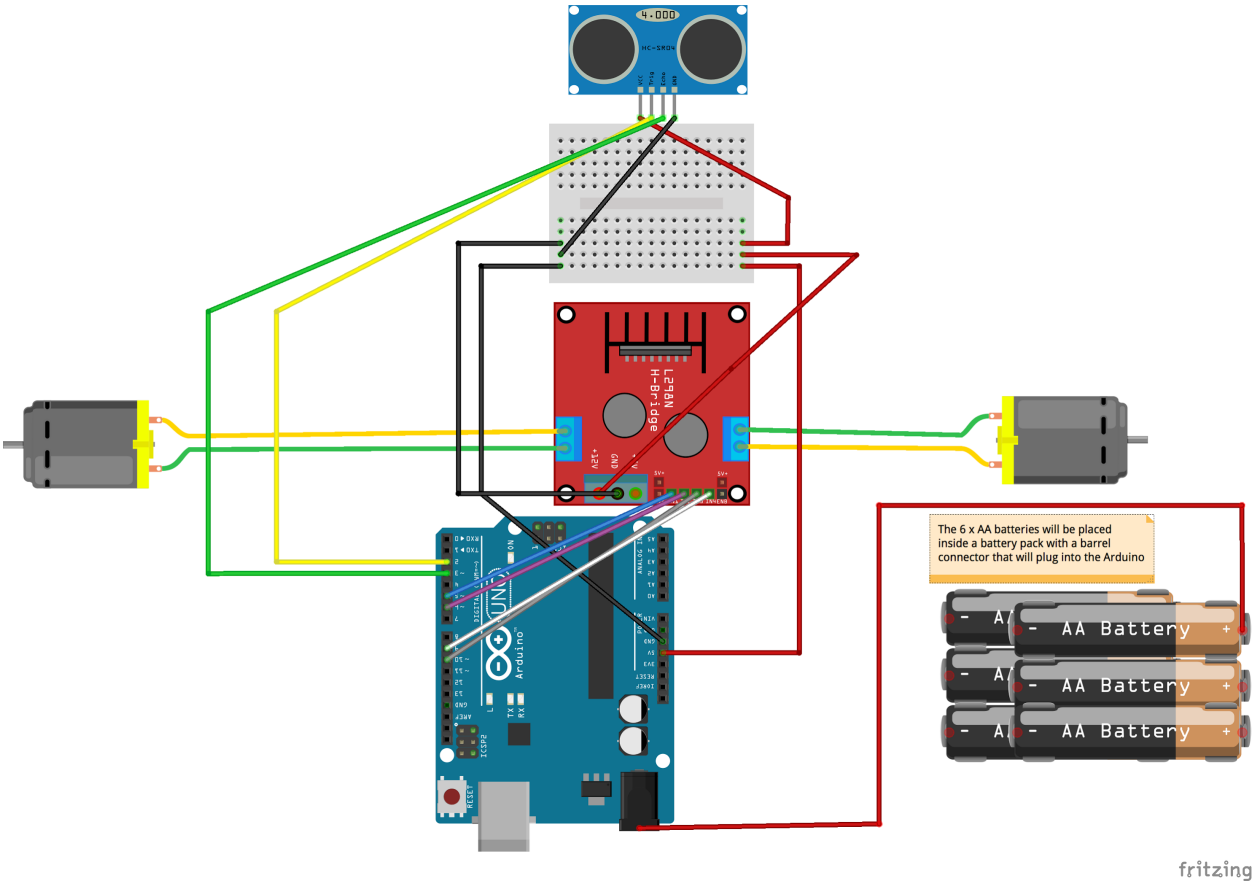
1 x Chassis
1 x L298N Motor Driver
1 x Breadboard
1 x Arduino
1 x USB Cable
1 x Ultrasonic Sensor
2 x 9v batteries and 2 battery connectors
OR
6 x AA batteries and 1 battery holder
2 rubber bands
4 x M-F jumper wires for motor driver
4 x M-F jumper wires for ultrasonic sensor
4 x M-M jumper wires for power/ground connections

Bluetack
Double sided sticky foam/tape

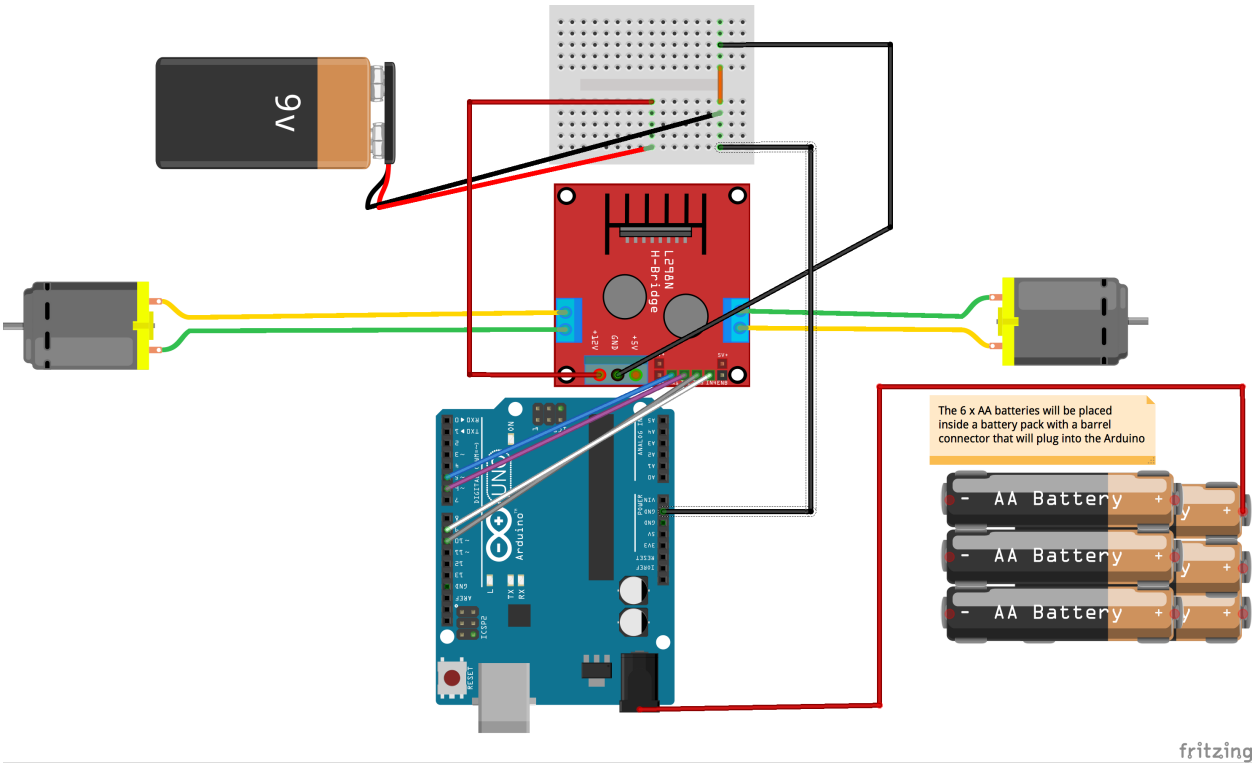
Extras
LEDs
Resistors

Appendix F - Diagrams

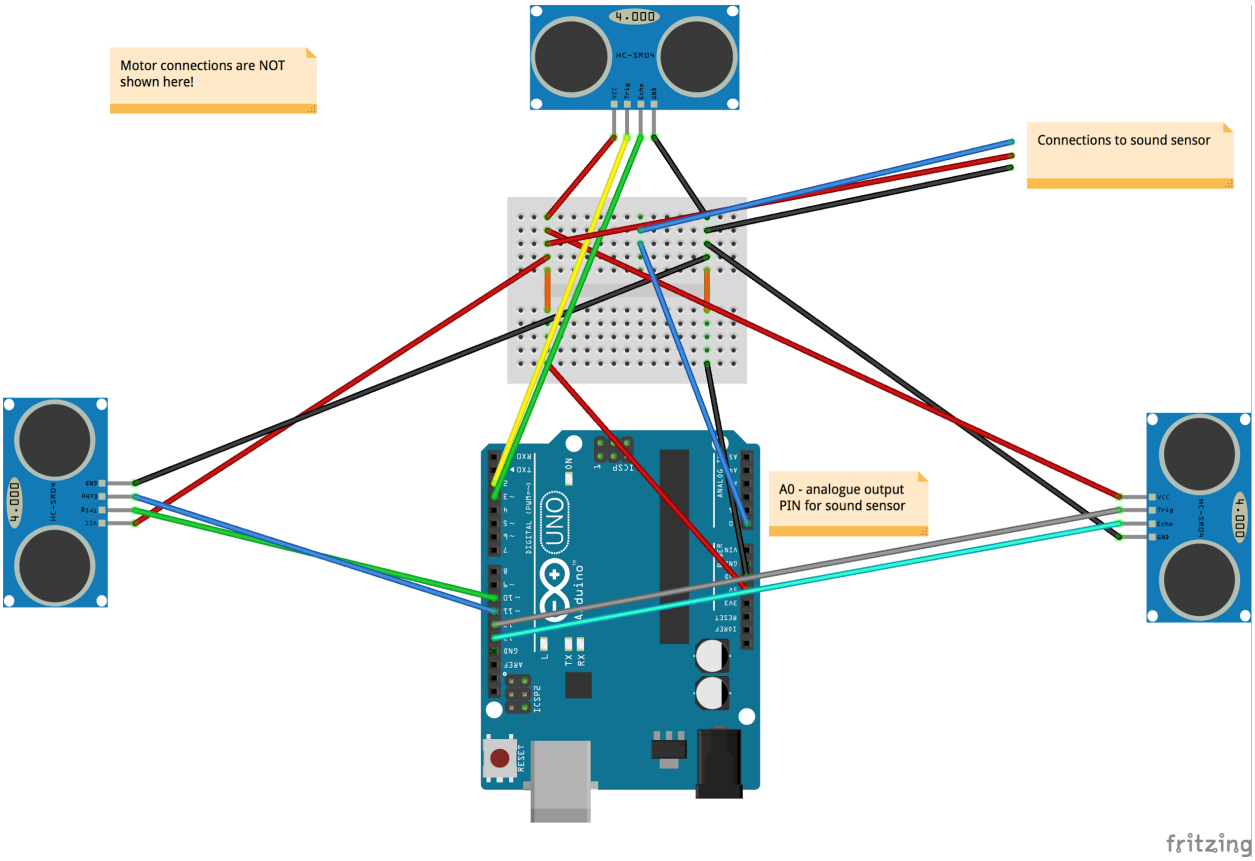
Example (beginner) with 1 Ultrasonic sensor



Example (skilled) – Motor Wiring

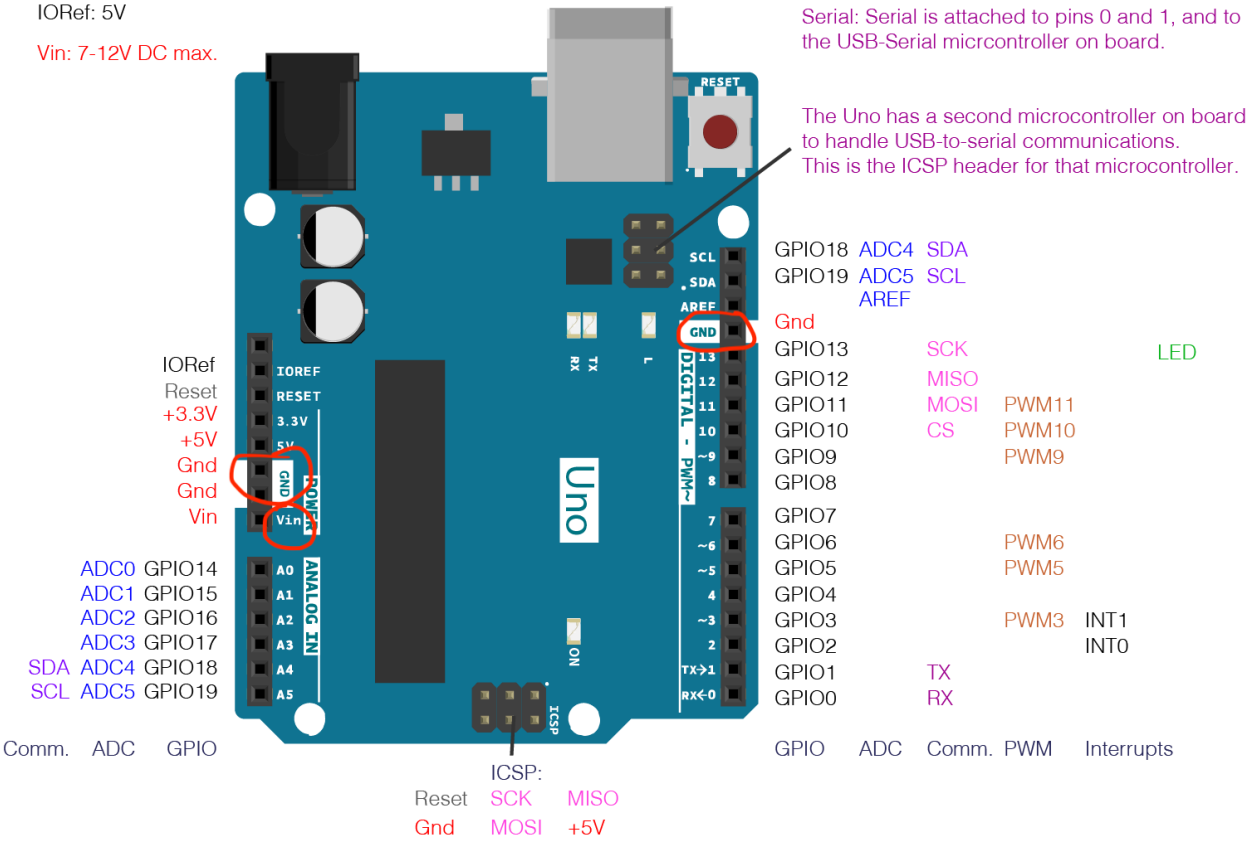


Example (skilled) - Ultrasonic Sensors Wiring



Appendix G - Arduino

UNO Pins



Blinking an LED