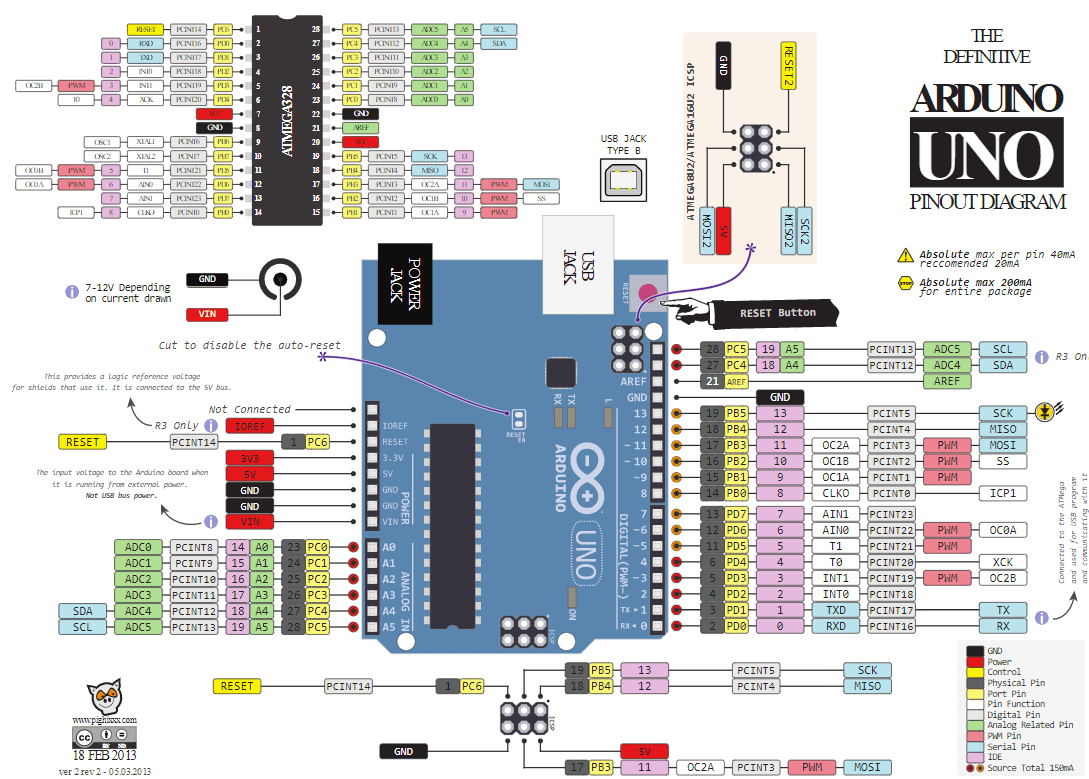
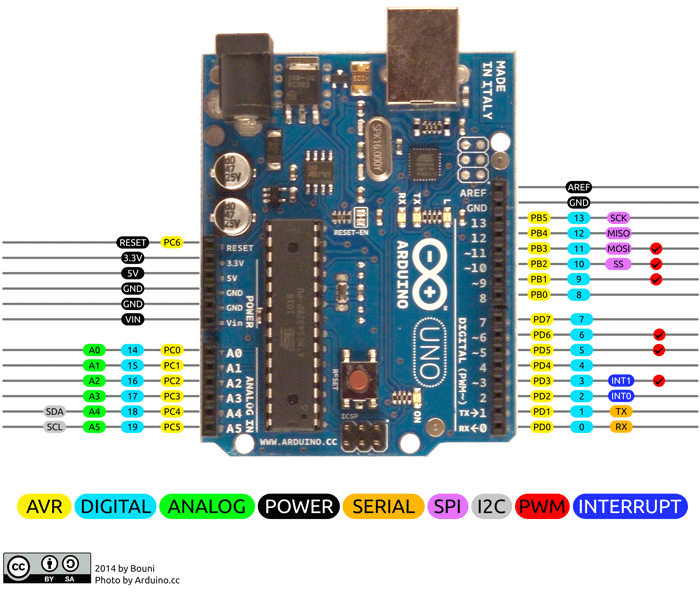
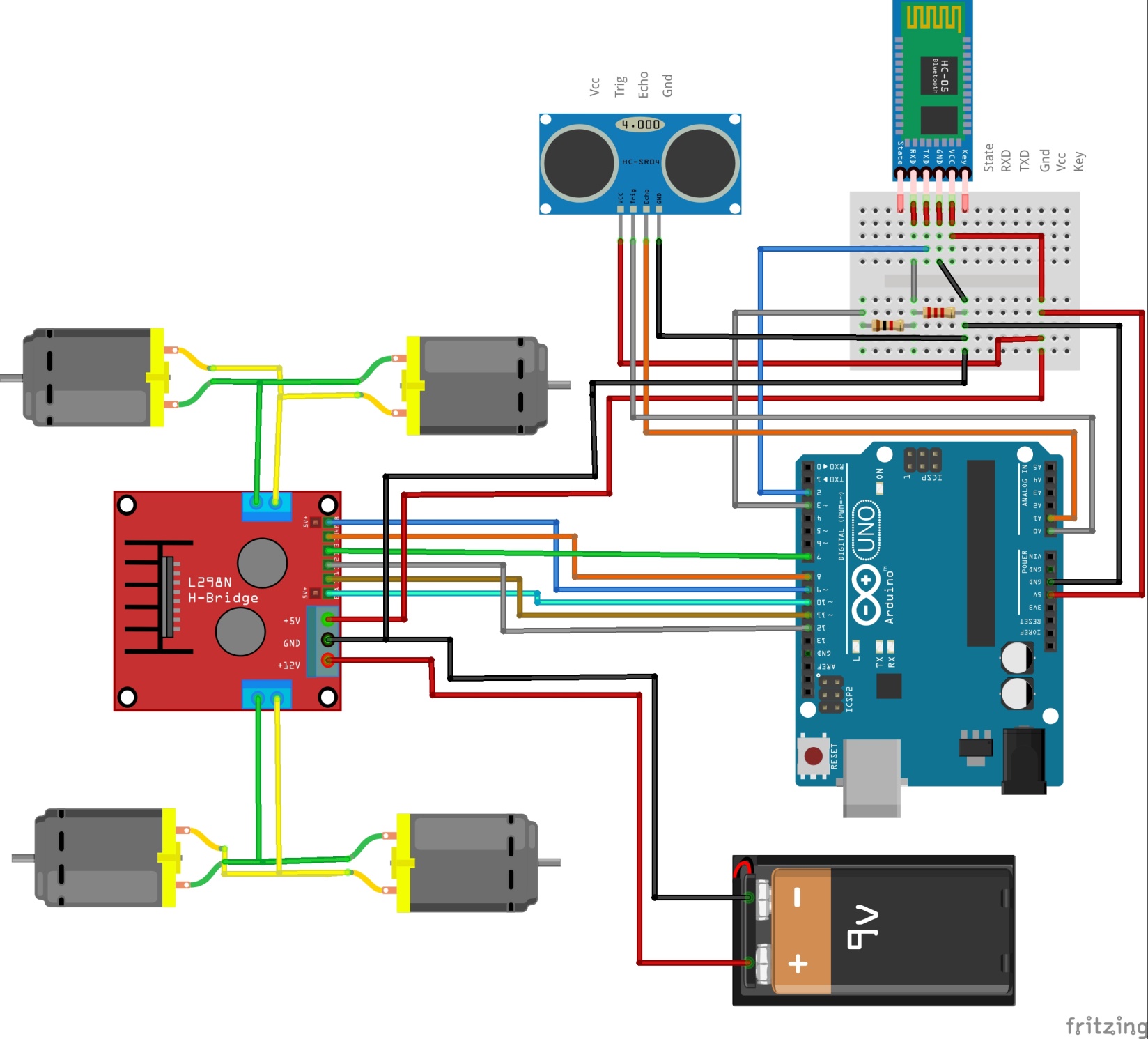
<https://www.circuito.io/blog/arduino-uno-pinout/>

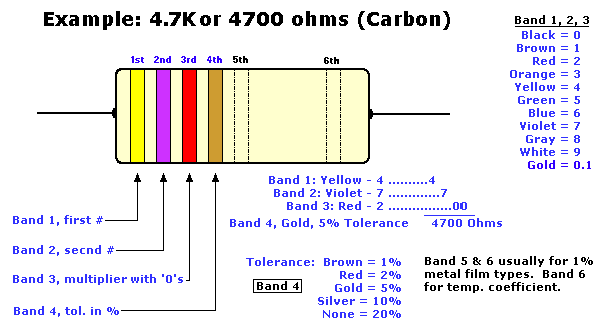




**Arduino Car Schematic**



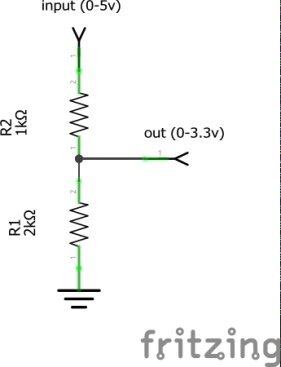
**Resistor Color Code and voltage divider Circuit**

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjBlcrj1uHgAhUR7J8KHSr3DVYQjRx6BAgBEAU&url=http://www.tonyvanroon.com/oldwebsite/gadgets/resistors/resistor.htm&psig=AOvVaw3XbtPGOK6fk2_Teybrk0B5&ust=1551554942963662)<http://www.tonyvanroon.com/oldwebsite/gadgets/resistors/resistor.htm>

**1k = Brown-Black-Red**

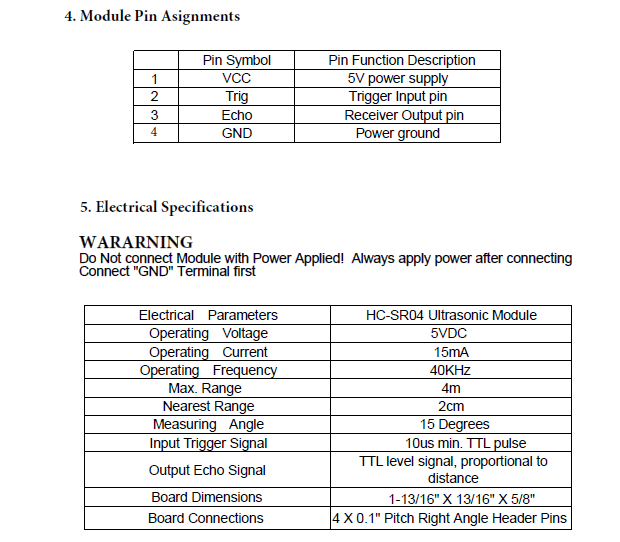
**2k= Red-Black-Red**

**Voltage Divider (5v to 3.3v)**



**HC-SR04 ULTRASONIC MODULE**

****



**Description of Operation**:

Set Trig and Echo Low to initialize module. Place a minimum 10us High

level pulse to "Trigger" (module will automatically send eight 40KHz acoustic

bursts). At the same time, Gate the microcontroller timer to start timing.

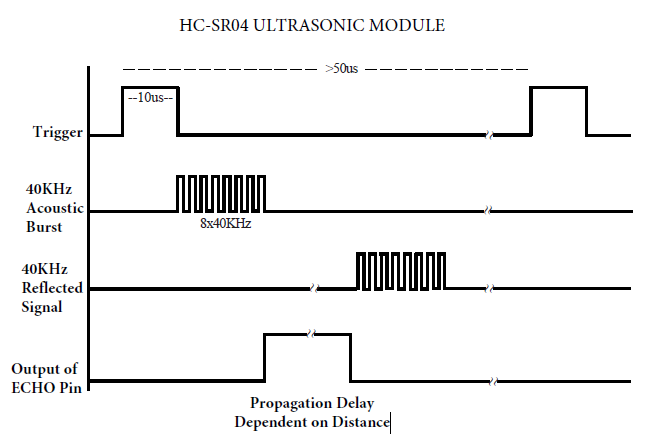
Wait to capture the rising edge output of ECHO port to stop the timer. Now

read the time of the counter, which is the ultrasonic propagation time in the air.

According to the formula: Distance = (ECHO high level time X ultrasonic

velocity (Speed of Sound in air 340m/sec) / 2, you can calculate the distance to

the obstacle.



**Trigger** 10us min. start measurement from microcontroller.

Max Rep. Rate: 50us

**ECHO** Output pulse to microcontroller, width is the time from last of 8 40KHz

bursts to detected reflected signal (microcontroller Timer gate signal)

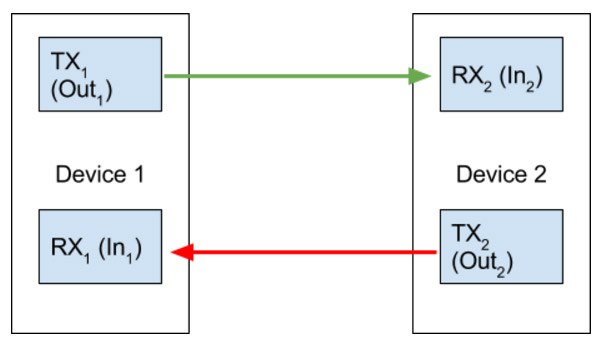
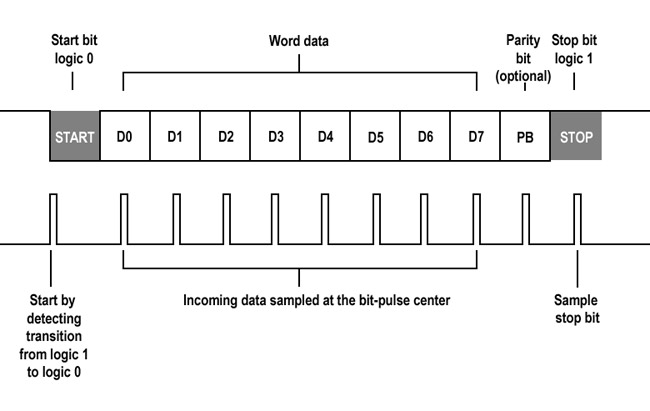
Distance in cm = echo pulse width in uS/58

Distance in inch = echo pulse width in uS/148

**HC-05 Bluetooth module**

[](https://wiki.eprolabs.com/index.php?title=File:HC-05.jpg)

Bluetooth uses UART (Universal Asynchronous Reciever/Transmitter) communication protocol.

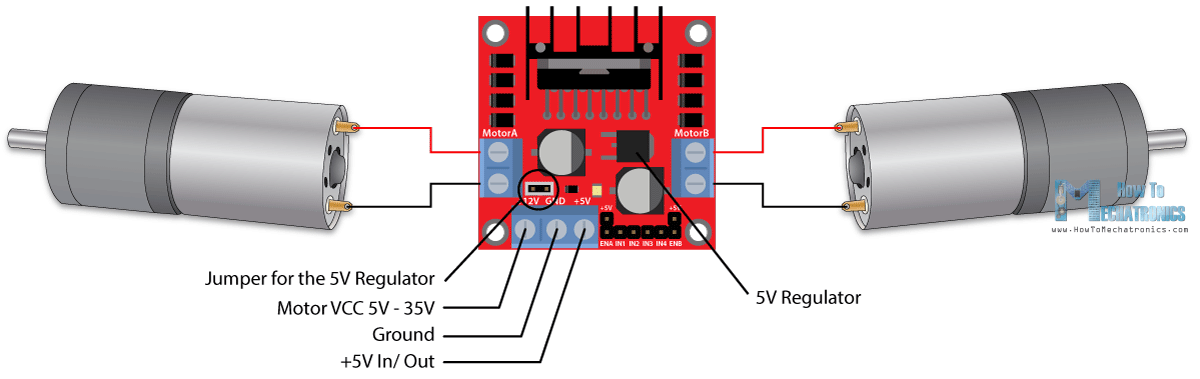


<https://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/>

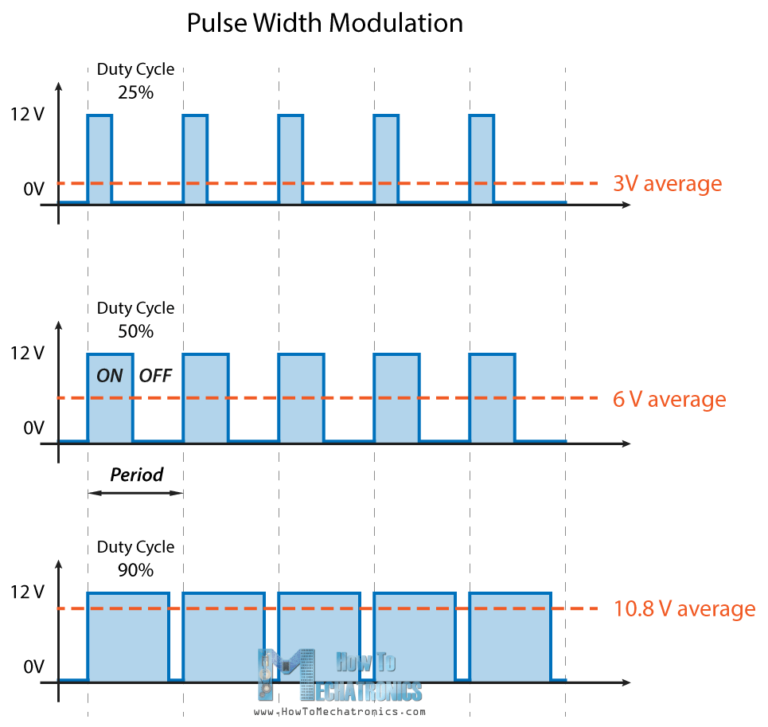
<https://alselectro.wordpress.com/2014/10/21/bluetooth-hc05-how-to-pair-two-modules/>

**LN298N Motor Driver**

<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>



Remove jumpers on enable pins if using PWM signal (jumpers connect Enable pins High).



|  |  |  |  |
| --- | --- | --- | --- |
| **EnA /EnB**  **Enable Motor** | **in1/in3**  **Motor Ctrl 1** | **in2/in4**  **Motor Ctrl 2** | **Status** |
| High | Low | Low | Off |
| High | Low | High | Clockwise |
| High | High | Low | Counter Clockwise |
| High | High | High | Off |
| Low | n/a | n/a | Off |

**Arduino Coding**

How to learn Arduino coding:

* <https://www.robotshop.com/community/tutorials/show/arduino-5-minute-tutorials-lesson-1-software>
* <https://www.makerspaces.com/arduino-uno-tutorial-beginners/>
* Run examples in Arduino IDE (File->Examples)

Reference:

<http://rosettacode.org/wiki/Rosetta_Code> <- great way to compare languages

<https://www.arduino.cc/reference/en/> <- Arduino language reference

**Complete build instructions:**

1. Configure Bluetooth Module (This step has already been done).
   1. Bring HC-05 into “AT” mode and set baud rate (38400), name (unique) and role (slave).
   2. <https://alselectro.wordpress.com/2014/10/21/bluetooth-hc05-how-to-pair-two-modules/>
2. Develop/install Arduino code (This step has already been done).
   1. Use Software serial for Bluetooth to eliminate need for unplugging when uploading new code.
   2. A valid command for Bluetooth data is <command><value>;
      1. Valid commands are f=forward, b=back, r=right turn, l=left turn, s=stop,o1= option1, o2=option2
      2. For driving commands, the “value” is the speed from 0% to100%,. Example: f75; <- forward 75%
      3. For the option commands, 0=off, 1=on. Example: o10; <- option 1 off
3. Configure jumpers on Motor Driver.
   1. The jumpers on ENA and ENB should be removed. This allows the use of a PWM signal to enable/disable the motors and control the speed. With the jumpers in place, the motors are always enabled and can not adjust speed.
   2. The power jumper should be enabled. This take the input “Motor VCC” power generates 5 volts from the onboard power regulator to the 5 volt terminal. This will be used to drive the Arduino and the motor controller circuit. If the power jumper is removed, the onboard regulator is disabled and the 5 volt terminal needs to be used as an input to power the motor controller circuitry and the “Motor VCC” will only be used to drive the motors.
   3. WARNING: You should not connect the 5+ terminal to external power when the power jumper is in place.
4. Assemble the chassis
   1. Mount the motors/wheels
   2. Mount the motor controller, Arduino, Ultrasonic sensor and battery pack.
      1. Use Velcro to mount the battery pack for easy removal.
5. Wiring
   1. Wire motors to the motor controller. Connect motors from left side to MOTORA terminals. Connect motors from right side to MOTORB terminals. (See schematics). If any motors are turning the wrong direction, swap the wires on the terminal for that motor.
   2. Wire the battery to the Motor control board (Motor VCC and GND).
   3. Wire only the ground wire from the motor controller to the Arduino.
   4. Wire the motor control pins from the Arduino to the motor controller. This will be a good time to test your wiring.
      1. Disconnect power jumper on Motor controller (if you have wired the Arduino 5 volts to the motor controller).
      2. Connect the Arduino to a computer (computer will be powering the Arduino).
      3. In IDE, select Arduino/Genuine Uno for the board and the proper port.. Open the Serial monitor and set to 115200 baud.
      4. Type commands into Serial monitor to test motor wiring and direction. Swap wires if backwards.
   5. Connect the 5 volt power wire from the motor controller to the Arduino. Not the Arduino will be powered by the motor controller regulator.
   6. It will be a good idea to connect power and ground from the Arduino to empty rows on the Breadboard. You will need multiple power connections and you can use this as a power bus.
   7. Mount HC-05 Bluetooth and resistors on breadboard. Wire as per the schematic. The resistors act as a voltage divider reducing the 5 volt signal output from the Arduino to the RX pin on the HC-05. This pin on the HC-05 can only support 3.3 volts.
   8. Test Bluetooth
      1. Double check wiring and that voltage divider is wired properly.
      2. Power the car and try connecting to Bluetooth via the phone app (Each car has a unique name);
         1. Pair your phone with the Bluetooth module.
         2. The passcode is: 1234
      3. Try sending commands.
   9. Wire Ultrasonic sensor and test.