**Build a RC car controlled via a mobile device (for robotics beginners)**

This tutorial will teach you how to build a small remote controlled vehicle that you will be able to control from a simple app on a bluetooth enabled Android or IOS tablet or phone. There is an additional section at the end if you are interested in including two servos to open and close a hatch.

**Materials:**

* A chassis with four motors (We used the [Half-Pint Rover](https://www.servocity.com/html/half-pint_runt_rovertm__637154.html#.V4FheJmDGko) from Actobotics)
* Arduino (we used an [Arduino Uno](https://www.arduino.cc/en/Main/ArduinoBoardUno))
* [HC-05 Bluetooth Module](https://smile.amazon.com/Wireless-Bluetooth-Transceiver-Module-Arduino/dp/B00PQ30CXQ/ref=sr_1_3?ie=UTF8&qid=1468097552&sr=8-3&keywords=hc05)
* [L298N Dual H-Bridge Motor Drive Controller](https://smile.amazon.com/Qunqi-Controller-Module-Stepper-Arduino/dp/B014KMHSW6/ref=sr_1_1?ie=UTF8&qid=1468098117&sr=8-1&keywords=l298n)
* A power source for the Arduino (We used a 9V battery and a [DC Jack Adapter](http://www.ebay.com/itm/like/121985101818?lpid=82&chn=ps&ul_noapp=true))
* A power source for the Motors (We used a [7.2V 1500mAh NiMH Battery Pack](http://www.duratrax.com/batteries/dtxc2010.html))
* A bluetooth enabled mobile device (IOS or Android)
* Breadboard
* Jumper Wires (Both male and female heads recommended)

**What are these electronic pieces for?**

Arduino:

The Arduino board is a microcontroller that serves as the brains for your car. You are able to upload a code onto the board and then it will be able to interpret signals from the bluetooth receiver and tell the motor driver what to do.

HC-05 Bluetooth module:

This receives the bluetooth signals from the mobile device and sends them to the Arduino.

L298N Motor Driver:

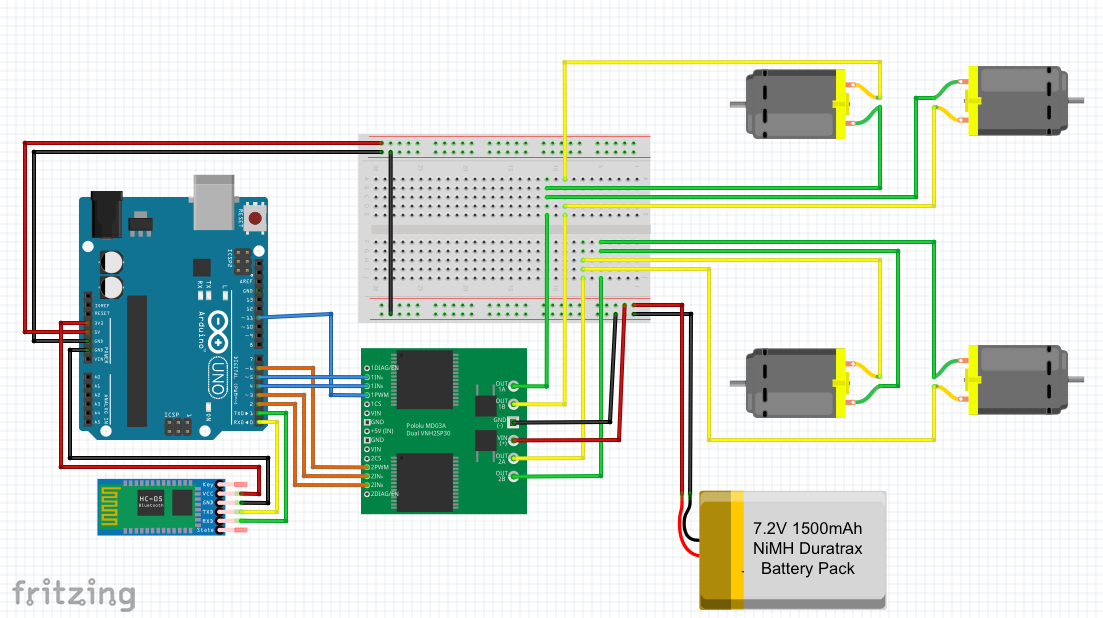
The motor driver takes a signal from the Arduino and allows a large current to flow to the motors. The motor driver is needed because the Arduino supplies only small amounts of current, roughly 40mA from its signal pins, which is plenty to turn on a small LED. A DC motor, however, will likely need a greater current in order to turn. The DC motors in the Half-Pint for instance operate on 190mA to 250mA, so we need the motor driver to take the Arduino’s signal and using the 7.2V battery, bump it up to a higher current for the motors.

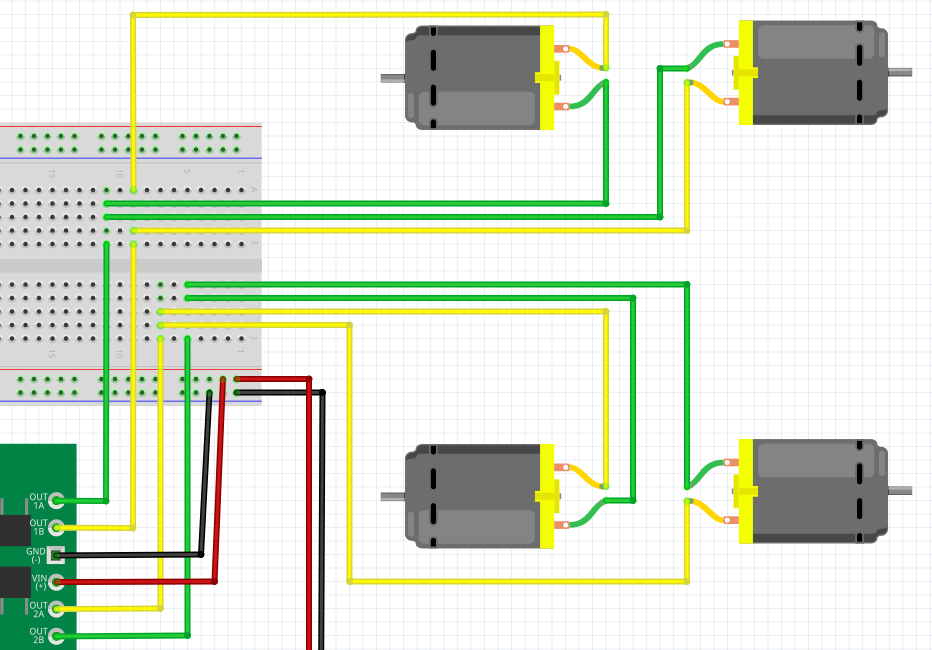
**A note about the materials:**

As far as the materials go there is a lot of variability in terms of what products will work. The links provided are simply guidelines for what we used. Any chassis can be used, as well as any DC motors as long as the motor controller supports the voltage and current specification of the motors. The L298N Dual H-Bridge Motor Drive Controller can take a 6-35V power supply. We used a 7.2V 1500mAh NiMH Duratrax battery because we found it left over from an old RC car. You can use a different type of battery (take proper precautions if using a LiPo battery), a different voltage rating seeing as the Half-Pint motors operate on around 4.5V, and a different capacity rating, for example a 3000mAh battery for a longer runtime. We powered the Arduino with a 9V battery because we had it on hand but there are other ways to do this as well, for example you could use a standard [USB power bank](https://smile.amazon.com/dp/B00LRK8EVO), or if you wanted you could consolidate down to one power source for the Arduino and the motors (If supplied with 12V, 5V can be pulled from the L298N motor driver to power the Arduino). You only need the receiver end of the HC-05 bluetooth module since the transmission will come from the Bluetooth built into your tablet or phone.

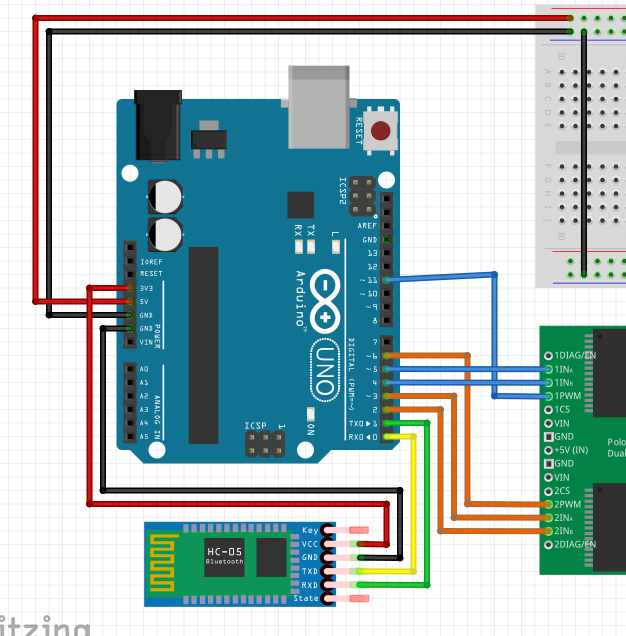
**Step 1: Wiring**

Use the following images as a guide to set up the wiring:



1. Connect the motor controller to the motors. If not done already, solder a wire to each terminal of the DC motors. Since the motor controller only has two outputs, the motors have to be wired in pairs so that we can move all four. The left two will be paired together and the right two will be paired together so the vehicle will be driven using differential steering, similar to how a tank drives. To do this, connect the two motors on right in parallel and the two motors on the left in parallel. By connecting the motors in parallel, both motors will experience the same voltage and turn at the same rate. If they were connected in series then the first motor would move faster than the second because there would be a voltage drop after the first motor. To attach the motors to the motor controller, connect the “positive” terminals of the first two motors to OUT1 (OUT1A) on the motor controller, and connect the “grounds” of the motors to the OUT2 (OUT1B) slot. Do the same for the other two motors, connecting the “grounds” to OUT3 (OUT2A) and the “positive” terminals to OUT4 (OUT2B). Since the DC motors do not actually have specified positive and ground terminals, you may end up with motors that spin the the wrong direction. To fix this you can just swap the wires and the motor will spin in the opposite direction. 
2. Connect the VCC and GND terminals of the motor controller to the breadboard but do not connect the 7.2V battery yet. The breadboard is used here so that the battery can be easily connected and disconnected. Unlike the motors, when the battery is eventually connected the distinction between GND and VCC is very important. Make sure the GND is connected to the ground of the battery and the VCC is connected to the positive terminal of the battery.
3. Connect the Motor controller to the Arduino in the following manner:

* IN1 to pin 2
* IN2 to pin 3
* IN4 to pin 4
* IN3 to pin 5
* ENA to pin 6
* ENB to pin 11



The ENA and ENB enable motors A and B to turn and determine their speed.

The IN1 and IN2 control the direction of motor A and the IN3 and IN4 control the

direction of motor B. By sending a HIGH value to IN1 and a LOW value to IN2, motor A will spin in one direction and by sending a LOW value to IN1 and a HIGH value to IN2 motor A will spin in the opposite direction.

1. Next connect the HC-05 Bluetooth module to the Arduino in the following manner:

* VCC to 3.3V of Arduino
* GND to GND
* TXD to RXD
* RXD to TXD

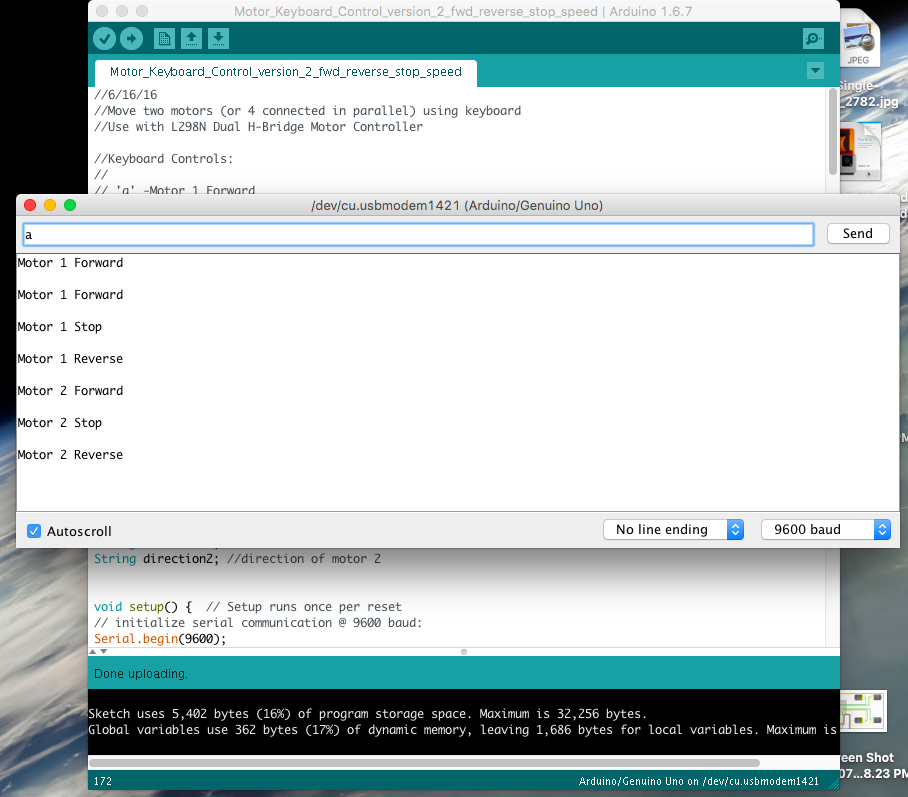
1. Finally connect all of the grounds together by running a wire from the Arduino ground to the ground on the breadboard. The additional wire connected to the 5V pin is for optionally adding servos later.

**Step 2: Uploading the code to the Arduino**

If it has not been downloaded already, go to the [Arduino website](https://www.arduino.cc/en/Main/Software) and download the Arduino Integrated Development Environment (IDE). Plug the Arduino into the computer and open up the Arduino IDE. Check to make sure that the board has been recognized by clicking Tools > Port > and selecting the /dev/ that corresponds to the Arduino. If it doesn’t show up at first try unplugging it and plugging it back into the computer in a different USB. Once you have selected the port, download our [repository](https://github.com/STEMInnovation2016/bluetooth_car_turorial) from github (Click “Clone or Download” and then “Download Zip”). Open the zip file and click on the “bluetooth\_car\_tutorial-master” folder. Then open the “bluetooth\_car.ino” file, which is an Arduino script (If it says that the file needs to be inside a sketch folder, then click “ok” to create the folder).

Now that you have the Arduino code,disconnect the TXD and RXD wires from the arduino. These wires have to be disconnected in order to upload anything to the Arduino. Once the wires are disconnected, click upload (the arrow pointing to the right) and the code will be stored on the Arduino. If it does not upload properly, try switching the port

To check that the code is working, in the top right of the Arduino IDE, click on the magnifying glass icon to open the serial monitor. In the bottom right of the serial monitor, set the baud rate to 9600 if it isn’t already. Then type the letter ‘a’ into the serial monitor and click enter. The monitor should respond with the message “Motor 1 Forward”. Enter any letter from ‘a’ through ‘l’ or the space bar to print different messages.



**Step 3: Testing the Bluetooth Module (Optional)**

With the serial monitor working, it is now time to test that the Bluetooth module works. Reconnect the TXD and RXD wires to the Arduino and make sure that the LED lights up on the Bluetooth Module. Turn on the computer’s bluetooth and pair the HC-05 device with the computer. If it does not pair at first, click on options and enter the passkey for the Bluetooth module which is either ‘0000’ or ‘1234.’ Once it is paired, go to the Tools > Port tab in the Arduino and select the port that corresponds to the HC-05. Then open the serial monitor again and now commands can be sent to the Arduino via bluetooth. To double check that the communication is happening over bluetooth, unplug the Arduino, power it via the DC jack and try sending serial commands from the computer again.

**Step 4: Download Evothings**

Evothings is the program that will be used to create the app for controlling the car. On a computer, go to the [Evothings Downloads page](https://evothings.com/download/) and download Evothings Studio. Evothings Studio is where the code will be saved for the app. On the Downloads page, click “Get token” to get a cloud token, open the Evothings Studio App, and enter the cloud token to connect to Evothings for the first time. Then go to the Apple store or Google Play store on your mobile device and download the Evothings Viewer app. On the Evothings studio app click “Get Key” and then enter that key into the Evothings Viewer app on the mobile device to connect the Viewer with the Studio.

Then download the following [repository](https://github.com/hammadtq/Bluetooth-LED-Remote-Control-Evothings-App) from github created by hammadtq. Open the “Bluetooth-LED-Remote\_Control\_Evothings-App-master” folder that you just downloaded and then open the “app” folder. Now go back to the “bluetooth\_car\_tutorial-master” folder that you downloaded from our github repository earlier. Replace the index.html file in the “Bluetooth-LED-Remote\_Control\_Evothings-

App-master” folder with the index.html file in the “bluetooth\_car\_tutorial-master” folder. We are replacing this index.html file because we have adjusted the buttons so that it will control our car instead of turning on and off an LED. Finally, on the Evothings Studio app on the computer, go to the “My Apps” tab and drag in the new index.html file.

**Step 5: Let’s Get Moving**

Power up the Arduino through the DC jack using the 9V battery. Then plug in the 7.2V NiMH battery to power on the motor controller (An LED on the motor controller should light up). Open up the Evothings viewer app on the mobile device so that it is connected to the Evothings Studio. On the computer click the “Run” button on the Bluetooth Remote Control Car app and it should run on the mobile device. If the bluetooth module doesn’t appear, click refresh and then connect to the bluetooth module. Once it has connected you should see the following buttons on the screen. Press these buttons to make the car move!

**Add Two Servos:**

Additionally, if you would like to control two servos, as we did to open and close a hatch, they can be added into the circuit as follows:

