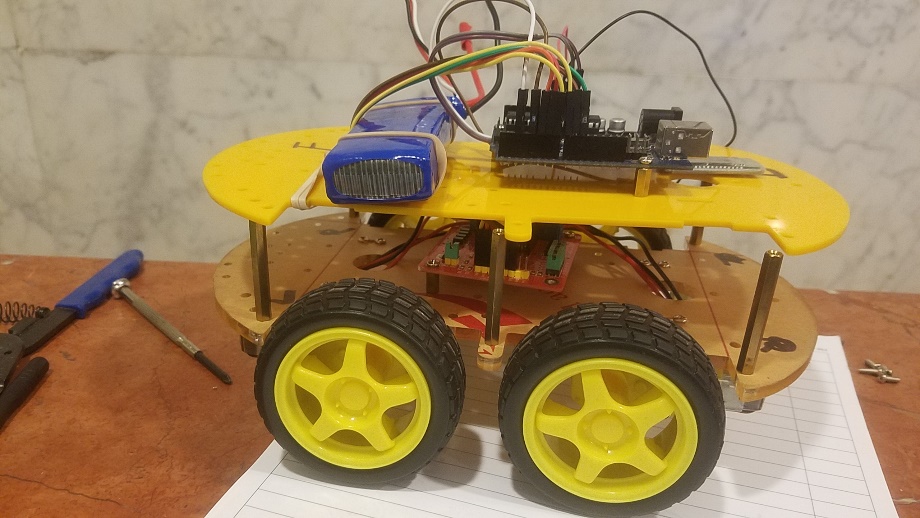
Moises Beato Nuñez

moises.beatonunez@gmail.com

**Bluetooth Car Instructions**

This document contains images and hyperlinks, so please use an appropriate word processor.



**Introduction:**

The idea here is to use an android app to control a robotic car via Bluetooth. The implementation is very simple, and the code is already written, tested and commented. The code is written using the Arduino IDE, so it is very simple to understand and modify to meet your requirements.

**How it works:**

The Arduino connects to your smart phone via the HC-06 Bluetooth to Serial module. The HC-06 is a slave Bluetooth module, so it only receives data. Via the Arduino Bluetooth App, we send signals to the Arduino, which in turn, send commands to the motor driver. Currently, the only commands the Arduino recognizes are Forward, Back, Left, and Right. The chassis used (image above) uses the [skid-steer](https://groups.csail.mit.edu/drl/courses/cs54-2001s/skidsteer.html) drive configuration, which means that the wheels have no axial rotation.

The Bluetooth app has more buttons and, therefore, capable of more functionalities. For example, you can add head and rear lights, a horn, or the ability for axial rotation of the wheels.

Note: If you use a different Bluetooth app, make sure you change your code accordingly. The current code is expecting to receive characters from the app (i.e. ‘F’ for forward, ‘B’ for back, etc).

**Parts requirement:**

*Software requirements*

* [Arduino IDE](https://www.arduino.cc/en/Main/Software)
* [Arduino Bluetooth RC Car Android App](https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller&hl=en)

*Hardware requirement*

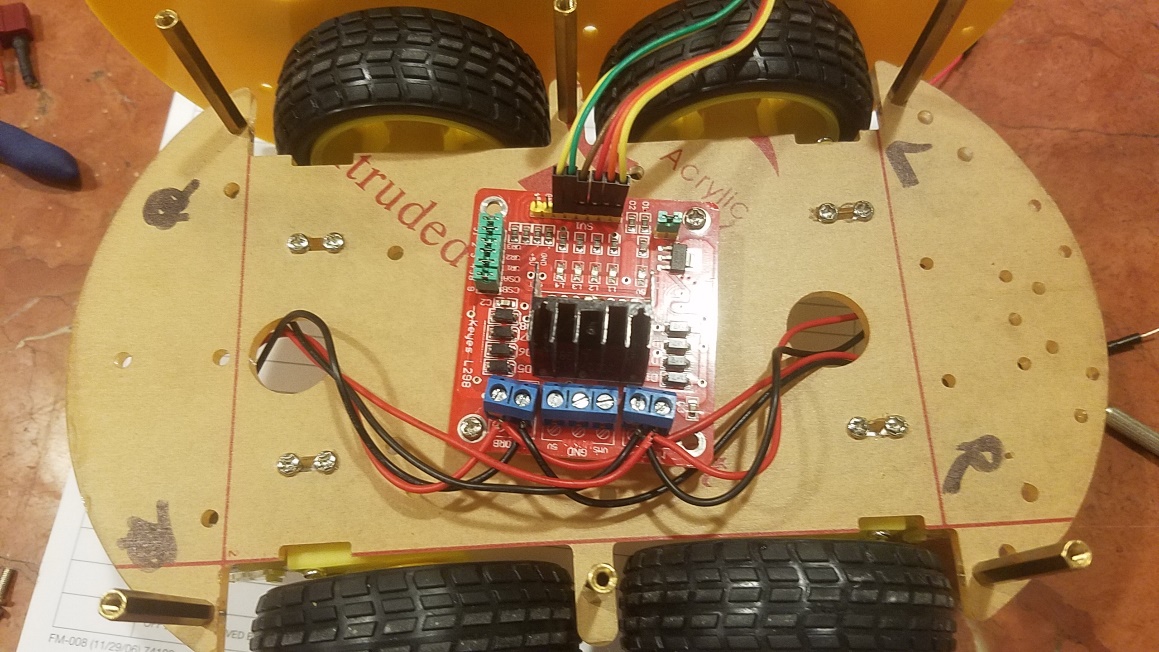
* [Arduino Uno](https://www.arduino.cc/en/main/arduinoBoardUno)
  + You may use a Duo, or a Mega if that’s what you have on hand, but the Uno is enough.
* [HC-06 Bluetooth to Serial module](https://www.amazon.com/Pass-Through-Communication-Compatible-Atomic-Market/dp/B00TNOO438)
* Any robot chassis kit ([here](https://www.amazon.com/ContempoViews-Robot-Smart-Chassis-5033/dp/B01LWYJX0Y/ref=sr_1_8?s=toys-and-games&ie=UTF8&qid=1494729203&sr=1-8&keywords=4WD+ki) is a cheap one), as long as it comes with DC motors.
* [L298N Dual H-Bridge Motor Driver](https://www.amazon.com/Qunqi-Controller-Module-Stepper-Arduino/dp/B014KMHSW6/ref=sr_1_1?ie=UTF8&qid=1494729360&sr=8-1&keywords=L298)
  + Some chassis kits come with a motor driver.
* [7.4v 2000mah lipo Battery](https://www.amazon.com/dp/B06X6JP5R3?psc=1) (only need one, and a [charger](https://www.amazon.com/GoolRC-Battery-Balancer-Charger-7-4-11-1V/dp/B00CF93RF4))
* SPST switch or a SPDT switch
  + Any type will do. I use one of [this](https://www.amazon.com/SupportTM-Miniature-Toggle-Switch-Dashboard/dp/B01GZS7FJO/ref=sr_1_3?ie=UTF8&qid=1494730090&sr=8-3&keywords=spst+switch).
* [Soldering iron](https://www.amazon.com/Weller-WLC100-40-Watt-Soldering-Station/dp/B000AS28UC/ref=sr_1_3?ie=UTF8&qid=1494757716&sr=8-3&keywords=weller+soldering+iron) (with [solder](https://www.amazon.com/Mandala-Crafts-0-6mm-Solder-0-6mmm/dp/B06XWJ8JS9/ref=sr_1_2?ie=UTF8&qid=1494757771&sr=8-2-spons&keywords=rosin+core+solder&psc=1))
  + I use a this [Solder Tip Cleaner](https://www.amazon.com/Hakko-599B-02-Solder-Cleaning-Holder/dp/B00FZPGDLA/ref=pd_bxgy_469_img_3?_encoding=UTF8&pd_rd_i=B00FZPGDLA&pd_rd_r=MKM6P5QD32Z2VCQTNH7G&pd_rd_w=dhhr0&pd_rd_wg=mWZIH&psc=1&refRID=MKM6P5QD32Z2VCQTNH7G), but any kitchen sponge will do.
* [Heat shrink tubing](https://www.amazon.com/JawayTool-Assortment-Electrical-Connection-Durable/dp/B018G6CQAG/ref=sr_1_2?ie=UTF8&qid=1494732086&sr=8-2-spons&keywords=heat+shrink+tubing&psc=1) or electrical tape (though tubing is cleaner)
* Jumper wires ([here](https://www.amazon.com/Haitronic-Multicolored-Breadboard-Arduino-raspberry/dp/B01LZF1ZSZ/ref=sr_1_3?ie=UTF8&qid=1494732286&sr=8-3&keywords=jumper+wires) is a kit), or some [22 AWG wires](https://www.amazon.com/StrivedayTM-Flexible-Electric-electronic-electrics/dp/B01LH1FYHO/ref=sr_1_1?ie=UTF8&qid=1494751734&sr=8-1&keywords=22+AWG+wires)
* 3 to 5 inches of [16 or 18 AWG wire](https://www.amazon.com/Bullz-Audio-BPES16-25-Gauge-Speaker/dp/B0149VCC10/ref=sr_1_2?ie=UTF8&qid=1494733390&sr=8-2&keywords=16+AWG+wires) (to connect from the battery to the switch)
* [Wire stripper](https://www.amazon.com/Wire-Stripper-Cutter-Crimper-Multi-Function/dp/B00ZYQEPEC/ref=sr_1_12?s=hi&ie=UTF8&qid=1494733204&sr=1-12&keywords=wire+stripper)
* Set of M3 nuts and screws
  + Standoffs would also help, but are not necessary.
  + [This kit](https://www.amazon.com/gp/product/B01LW4A34E/ref=oh_aui_detailpage_o07_s00?ie=UTF8&psc=1) comes with both

**Connections:**

***Motor Driver***

The L298N Dual H-Bridge motor driver is the middle man between the DC motors and the Microcontroller (Arduino). Assemble the chassis kit, use the M3 nuts and screws (and standoffs if you have them) to bolt the motor driver in a location where all the DC motors can reach it. If you are not using standoffs, use additional M3 nuts to elevate the motor driver off the surface of the chassis kit. Then connect the DC motors to the motor driver.

Connecting the DC motors

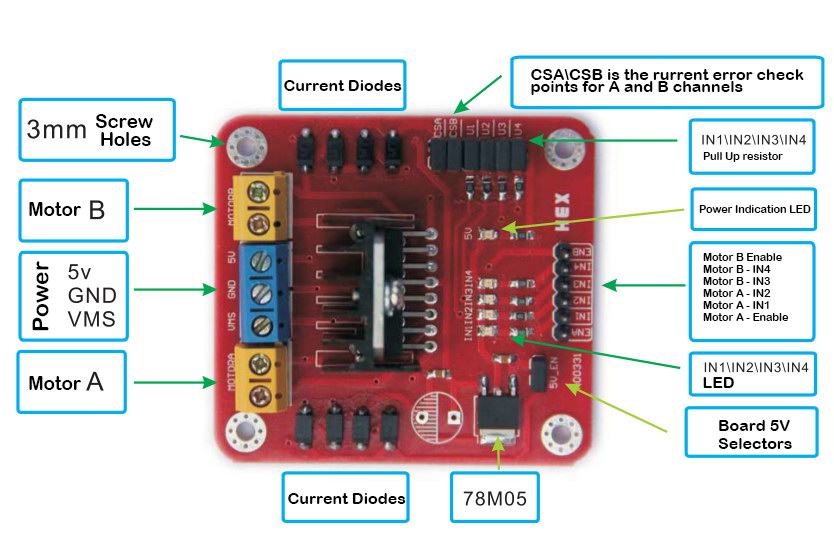


Here we are connecting two DC motors per H-bridge. The left side goes to the “MOTOR A” output port and the right side goes to the “MOTOR B” output port.

* Left side motors
  + Twist the positive and negative wires together
    - If you need to, use wire stripper to expose more of the cable
  + Connect the wire pairs to the blue output port labeled MOTOR A
    - red wires on the outside, black wires on the inside
* Right side motors
  + Twist the positive and negative wires together
    - If you need to, use wire stripper to expose more of the cable
  + Connect the wire pairs to the blue output port labeled MOTOR B
    - red wires on the outside, black wires on the inside

Connection with Arduino

This Dual H-bridge motor driver requires 6 pins to be connected to the Arduino; 2 enable pins, and 4 signal pins.

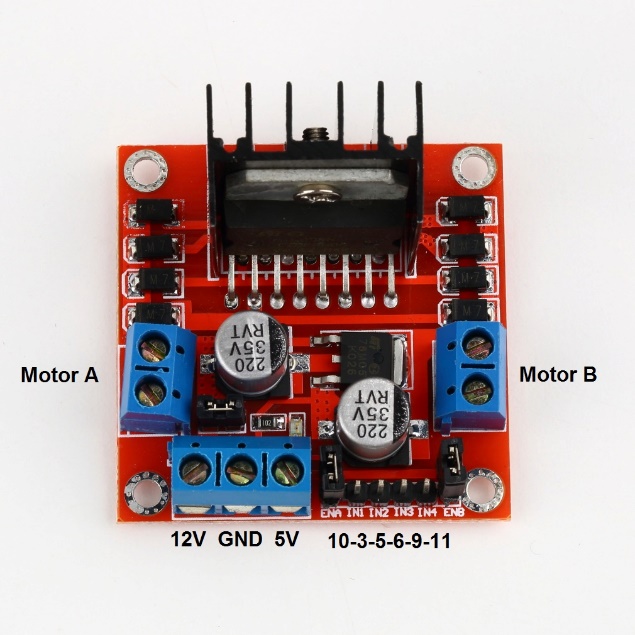


The Enable pins do just that, enable. ENA and ENB enable MOTOR A and MOTOR B, respectively. All they need is a constant signal (5v) to remain enabled. Connect ENA and ENB to Arduino pins 7 and 8. The order doesn’t matter since we are just setting them to “HIGH” internally.

Make the following connection between the motor driver and Arduino:

* IN1 → PWM pin 5
* IN2 → PWM pin 6
* IN3 → PWM pin 9
* IN4 → PWM pin 10

In the Arduino UNO, the PWM pins are identified by the “ ~ “ next to the number. The reason we use PWM pins is because we can control the speed of the motors. Click [here](http://www.circuitstoday.com/pwm-generation-and-control-using-arduino) for a general explanation on Pulse Width Modulation and the Arduino.



Alternatively, if you own a motor driver like the one above, you can completely ignore the enable pins as they are already connected to a 5v line via jumper. The other connections remain the same.

***Connecting the Power***

To power this project, we are using a rechargeable, 7.4v lipo, 2200mAh battery.



The battery comes with two sets of wires. The thinner three wire set connects to a [charger](https://www.amazon.com/GoolRC-Battery-Balancer-Charger-7-4-11-1V/dp/B00CF93RF4), and the thicker wire pair connects a RC or Quadcopter connector. We will be dealing with thicker wire pair.

*Adding Power to the motor driver*

Here we will connect the battery to the motor driver, but before we do that, we will add a switch to ensure we can turn the power on and off without having to remove the battery every time.

Things you will need:

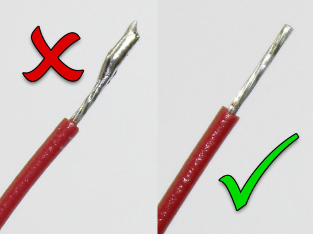
* 7.4v Lipo battery
* Switch (I am using a SPDT switch, but a SPST one will work as well)
* Soldering iron
* Heat shrink tubing, or electrical tape
* 16 or 18 AWG wire
* Wire stripper



Using the wire stripper, cut the banana connector from the battery.

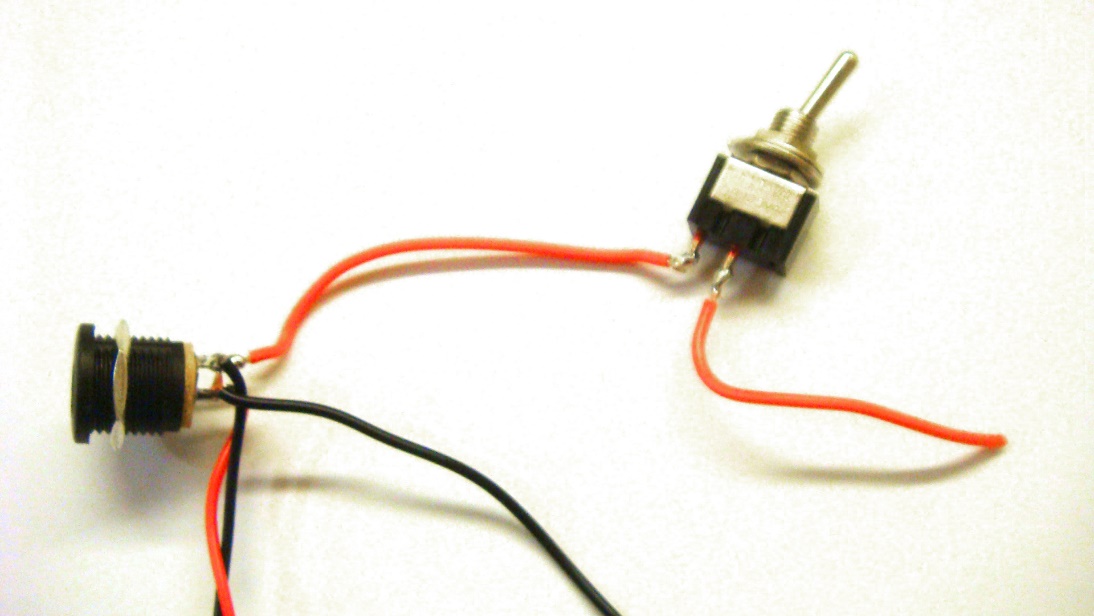


While making sure the red and black wires do not touch, strip about ¼ inch of wire from both.

Heat up your soldering iron and add a little bit of solder to the wire tips to make sure the wire doesn’t come apart later, this also helps when adding the switch. This process is called “tinning the wire”.

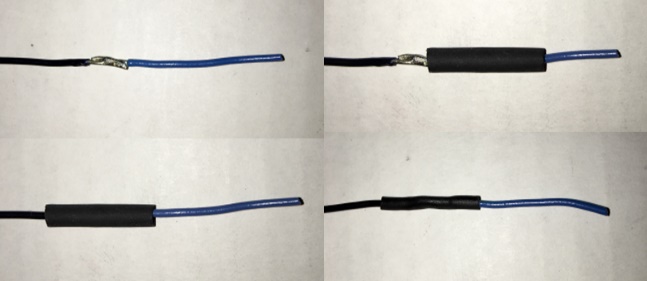
Repeat the same process with about 5 inches worth of 16 AWG wire pair. It can be more, or less than 5 inches, just make sure that there enough slack between where you plan to place the switch and where the motor driver is located.

Now, before we solder the wires to the switch, cut 3 pieces of about ½ inch of heat shirk tubing. Insert 1 piece per battery terminal wires (one on red and one on black). And 1 piece through the red piece of 16 AWG wire. DO NOT APPLY HEAT.

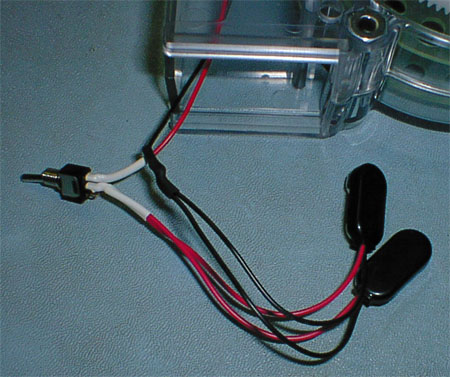


Now, solder the red battery terminal to one of the side terminal of the switch, and the red 16 AWG wire to the middle terminal. This can be inverted, just make sure that one of them is soldered into the middle terminal.

Move the heat shrink tubing over the soldered points, leaving no open wire exposed, then apply heat.

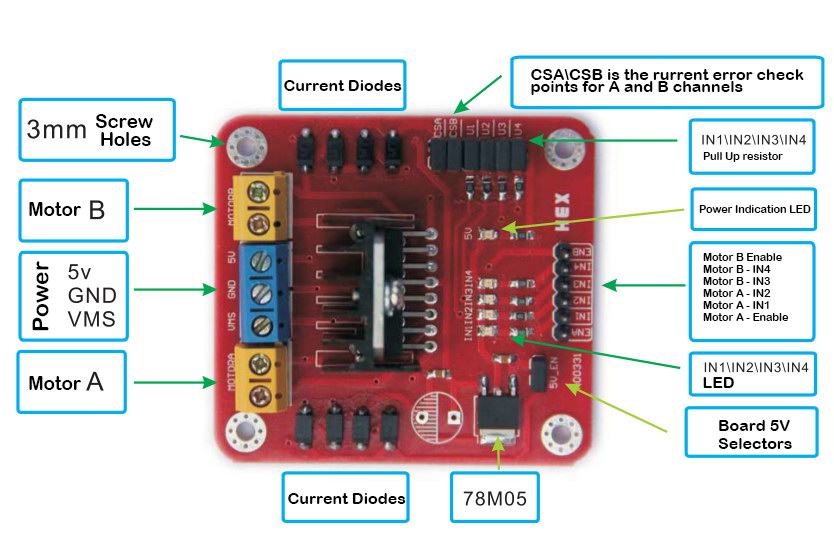


Solder the black wire battery terminal to the 5-inch piece of black 16 AWG wire. Place the heat shirk tubing over the exposed area and apply heat.



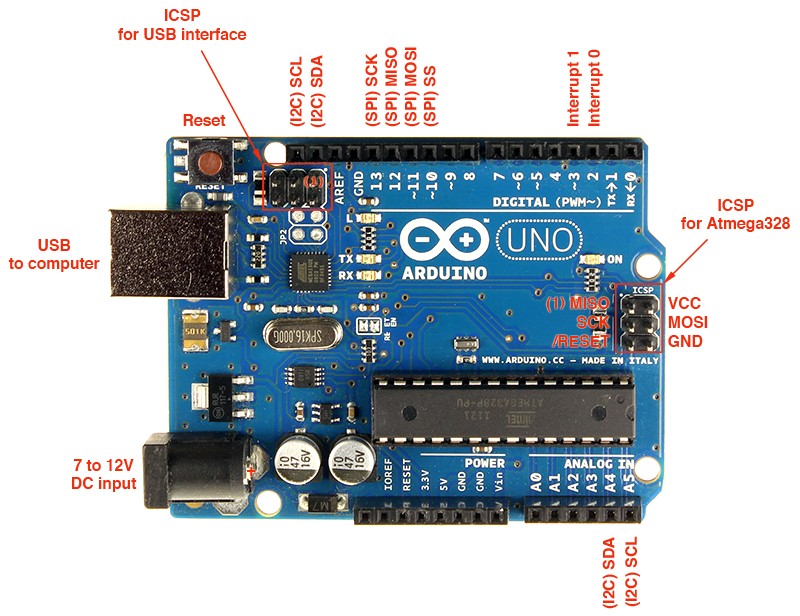
The result should look like this. You may place your switch wherever you want, just keep in mind the length of the cable and the placement of the battery. Now you are ready to power the motor driver and Arduino.

*Adding Power to the Arduino*



Connect the red wire coming out of the switch into the VMS (or 12v) port of your motor driver, and the black cable into the GND (ground) port.

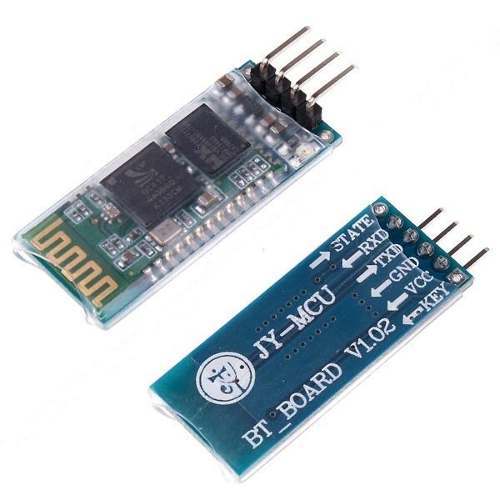
Before tightening the screws on the port, grab two male-to-male jumper wires from your kit (red and black) or two pieces of 22 AWG wire with their tips stripped. Making sure the red and black wires do not touch, wedge in the black wire into the GND port and the red wire into the VMS port. Now you may tighten the screws. We will be using these two wires to power the Arduino.

Connect the red jumper into the VIN port of the Arduino, and the black jumper into the GND port. This will ensure that the Arduino is powered directly by the battery and, therefore, will have enough current to operate.

***HC-06 Bluetooth Module***

The HC-06 Bluetooth module is a slave device, meaning that it can only receive signals. It will allows us to communicate with our Bluetooth app. It has 6 pins, but we will only be using 4. Connecting this is simple.

*Adding the Bluetooth Module*

The Arduino Uno have 1 hardware serial connection (UART), pin 0 for RxD and pin1 for TxD. Regrettably, these are also the same connections used when uploading code via USB or communicating to the Serial Monitor. To avoid conflict, we will be using the SoftwareSerial.h library, which allows us to use regular digital pins as Serial connectors.

Thing you will need

* HC-06 BT Module
* 4 female-to-male jumpers

Connections from HC-06 to Arduino:

* VCC → 5v pin
* GND→ Any open GND pin
* TxD → Pin 2 (defined as software RxD)
* RxD → Pin 4 (defined as software TxD)

*Testing the Bluetooth Module*

To test and see if the module work, first download the android [app](https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller&hl=en) on your smart phone, then open a new sketch in the Arduino IDE and run the following code. If you have never used the Arduino IDE before, follow [this guide](https://learn.sparkfun.com/tutorials/installing-arduino-ide).

#include <SoftwareSerial.h>

#define RxD 2 // connect to Tx pin on bluetooth receiver

#define TxD 4 // connect to Rx pin on bluetooth receiver

SoftwareSerial blueToohSerial**(**RxD**,**TxD**);** //software serial object

char recvChar**;** // saves the value received from the HC-06

void setup**()**

**{**

Serial**.**begin**(**9600**);**

pinMode**(**RxD**,** INPUT**);**

pinMode**(**TxD**,** OUTPUT**);**

blueToohSerial**.**begin**(**9600**);** // connect to HC-06

blueToohSerial**.**flush**();** // flush whatever was there before

**}**

void loop**()**

**{**

**if(**blueToohSerial**.**available**())** // if there is a connection, read what is being sent

**{**

recvChar **=** blueToohSerial**.**read**();**

Serial**.**println**(**recvChar**);**

**}**

**}**

Open the app, click on the settings icon (gray gear) and select “Connect to Car”, look for HC-06 and select it. Once the bluetooth connection is stablished, the red icon on the app will turn blue, and the HC-06 module will stop blinking. Go back to the computer and open the Serial monitor. You should see the screen populated with “S” all the way down, until you press a different key.

* Up arrow → “F”
* Dn arrow → “B”
* Right arrow → “R”
* Left arrow → “L”
* Up-Right → “I”
* Dn- Right → “J”
* Up-Left → “G”
* Dn-Left → “H”
* No button → “S”

**The Code:**

This is, by far, the easiest part of this whole thing. The code has already been written and should be in the same directory as this document. Simply open it with the [Arduino IDE](https://learn.sparkfun.com/tutorials/installing-arduino-ide) and upload it into your board.

If for whatever reason the code is not included, you can find it in my GitHub page (<https://github.com/momonarie22/Arduino-Bluetooth-RC-car>)