



Bluetooth RC Car with Remote Arduino

A Bluetooth-enabled RC Car that uses Remote Arduino to allow another Windows device to control it.

In this project, we will modify an RC car to allow it to be controlled using Windows Remote Arduino via a Bluetooth connection! With Remote Arduino, it is possible to build a Universal Windows Application which will direct the car in any direction. We used a Lumia phone running Windows 10 to control the car with the built-in accelerometer!

This sample is "remote-controlled-car" inside the Win10 folder. Make sure to clone the repository recursively so that you also obtain a copy of the library (more info in the readme)!

Hardware Set Up

See the pictures below for an image-based overview of the set up process.

We're going to remove the existing circuitry and add our own Arduino and motor controller (shield). You can technically use the existing circuitry to drive the motors, but the work involved in doing so can be much more complicated and varies wildly from car to car! This solution, while requiring more parts, is much more general. At any rate, make sure to remove the battery before taking the car apart!

- Unscrew the bottom frame from the rest of the car
- Remove the wire connectors from the circuit board while removing the top of the frame from the bottom
- Unscrew the circuit board and cut the wires away. You can discard the board if you wish, we won't need it again.
- Prepare the Arduino and motor shield by sliding the shield down directly over the Arduino.
- Verify that all jumpers on the motor shield are in the correct orientation. The Velleman ka03 I am using has five total jumpers! Four are used to select the control pins for the motors. The most important one controls where the motor power comes from, so you'll need to decide if the RC car battery will continue to power only the motors (EXT) or the Arduino as well (INT). If you use internal power, you'll need a barrel jack to hook the power/ground wires up to before inserting it onto the Arduino's DC port.
- Attach the leads from each motor to a terminal on the motor shield. I've hooked up the front (left/right) motor to terminal 1 and the back (forward/back) motor to terminal 2.
- Attach the power leads to either the external power terminal on the motor shield (EXT) or to a barrel jack which will be plugged into the Arduino to power both devices at the same time (INT). I am using internal power, which is more than sufficient for the car that I am using!

- Last, hook up the Bluetooth device to the Arduino by connecting a wire from TX on the Arduino to RX on the Bluetooth device and vice-versa. Also connect it to power and ground!

That is the last of the hardware setup steps! You'll eventually want to secure the board down using a custom mount or zip ties!

Note: This RC Car that I am using uses a small motor for left/right which is designed to turn until it stalls, which then draws a lot of current. It is technically working as intended, and is OK to leave the way it is as long as your motor shield can supply the power it draws. However, I've gone ahead and added a 5w 10 Ω resistor to the front motor to reduce the amount of current required.

Software

You'll find "remote-controlled-car" inside the W10 directory. This sample uses a Windows 10 phone's accelerometer to drive the car! If you are using a different pin configuration to control the motor shield, you'll want to locate and change the pin values for the following variables inside [ControlPage.xaml.cs](#):

- FB_DIRECTION_CONTROL_PIN = 8;
- FB_MOTOR_CONTROL_PIN = 9;
- LR_DIRECTION_CONTROL_PIN = 2;
- LR_MOTOR_CONTROL_PIN = 3;

Or just modify the source to use the default (zero-argument) constructor of BluetoothSerial which will attempt to connect to each Bluetooth device you are paired to until it is successful with one.

As usual with Windows Remote Arduino, you only need to program your Arduino device to run "StandardFirmata" (you can find this in the "Firmata" folder in the included libraries. Just make sure to change the baud rate in the sketch to match your device. The BlueSmirf runs at 115200 by default.

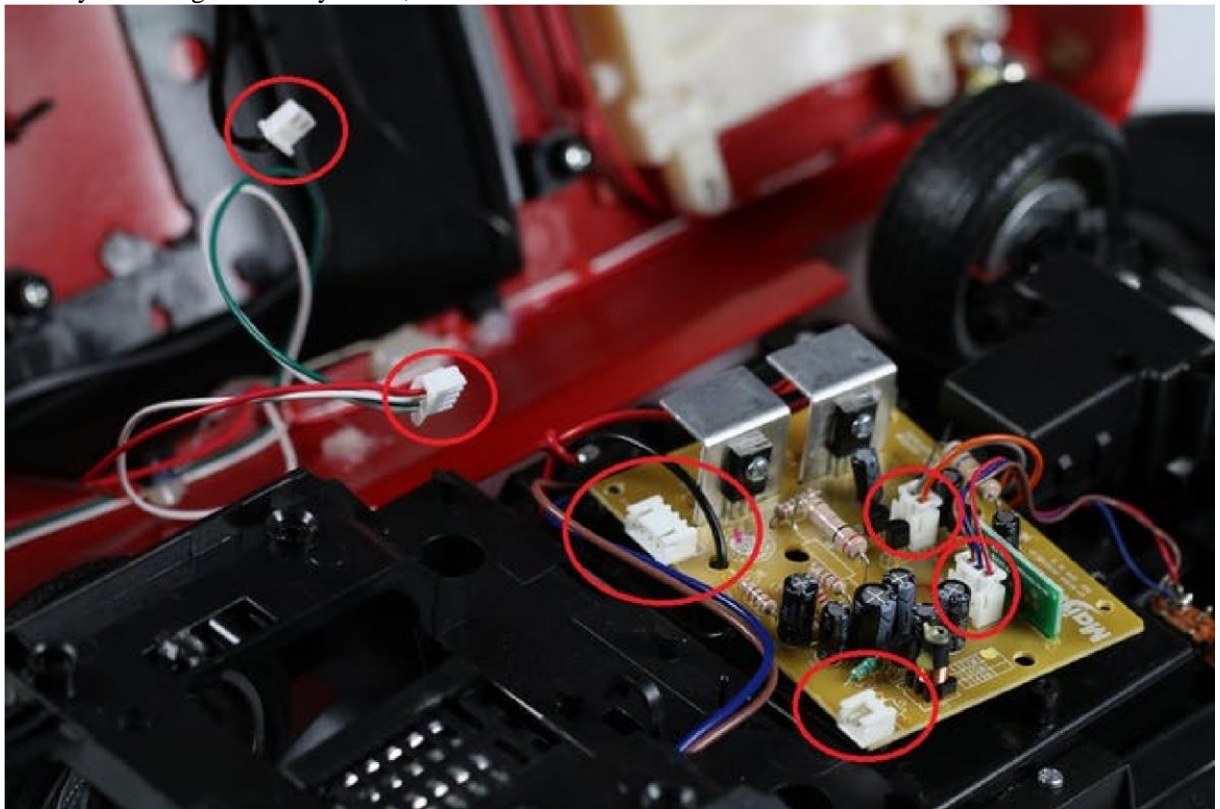
Pictures from the process

Here is the car right out of the box.

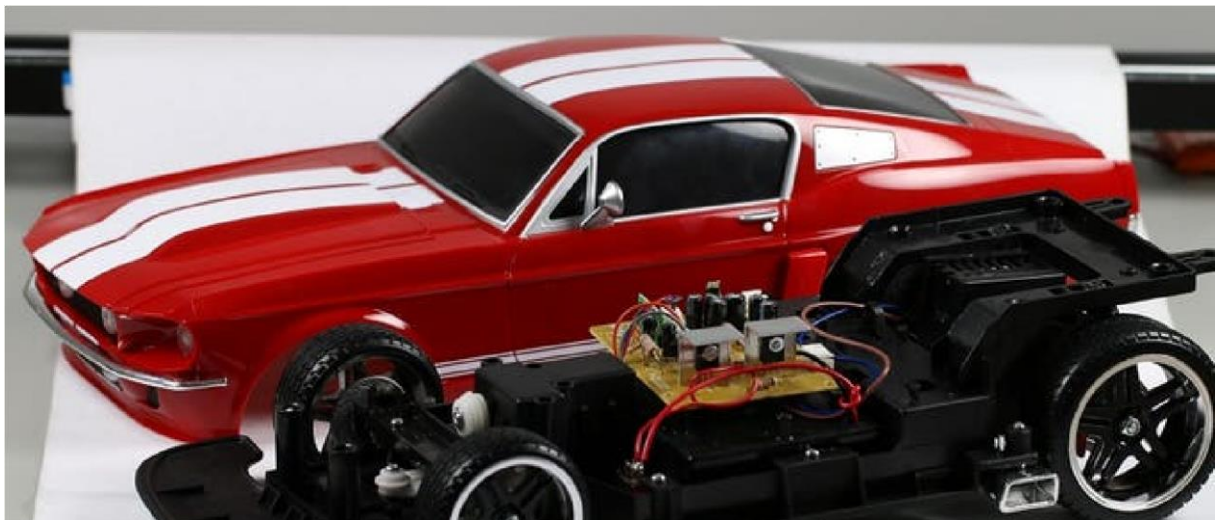




Start by removing the battery. Then, unscrew the bottom frame.

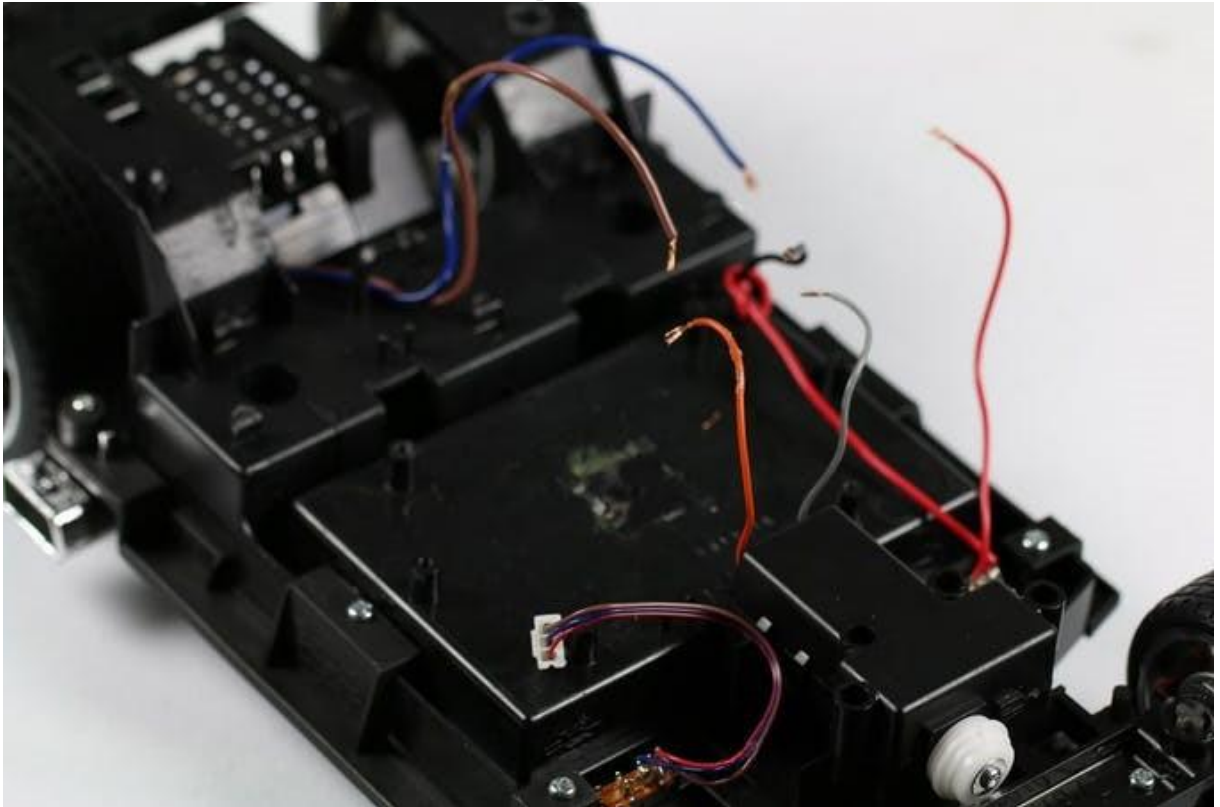


As you pull the frames apart, you'll find several wire connectors attached from the frame to the circuit board. Remove these!

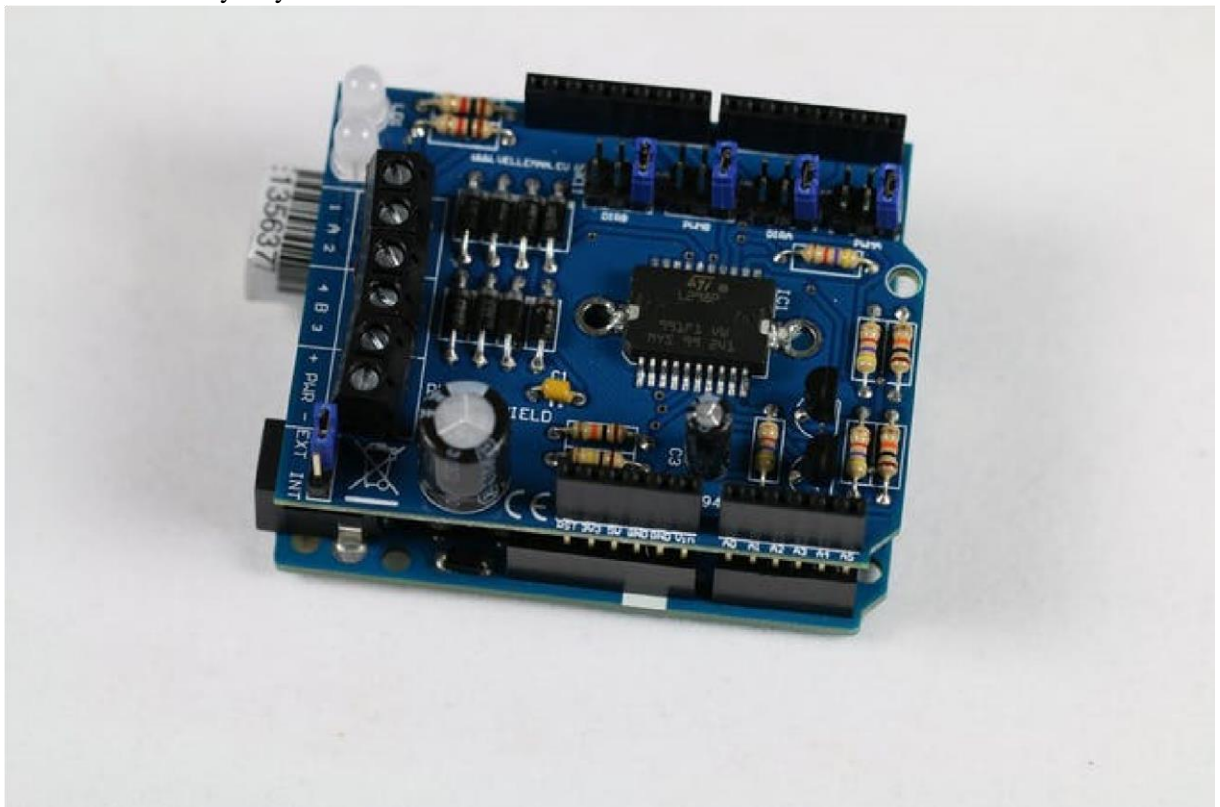




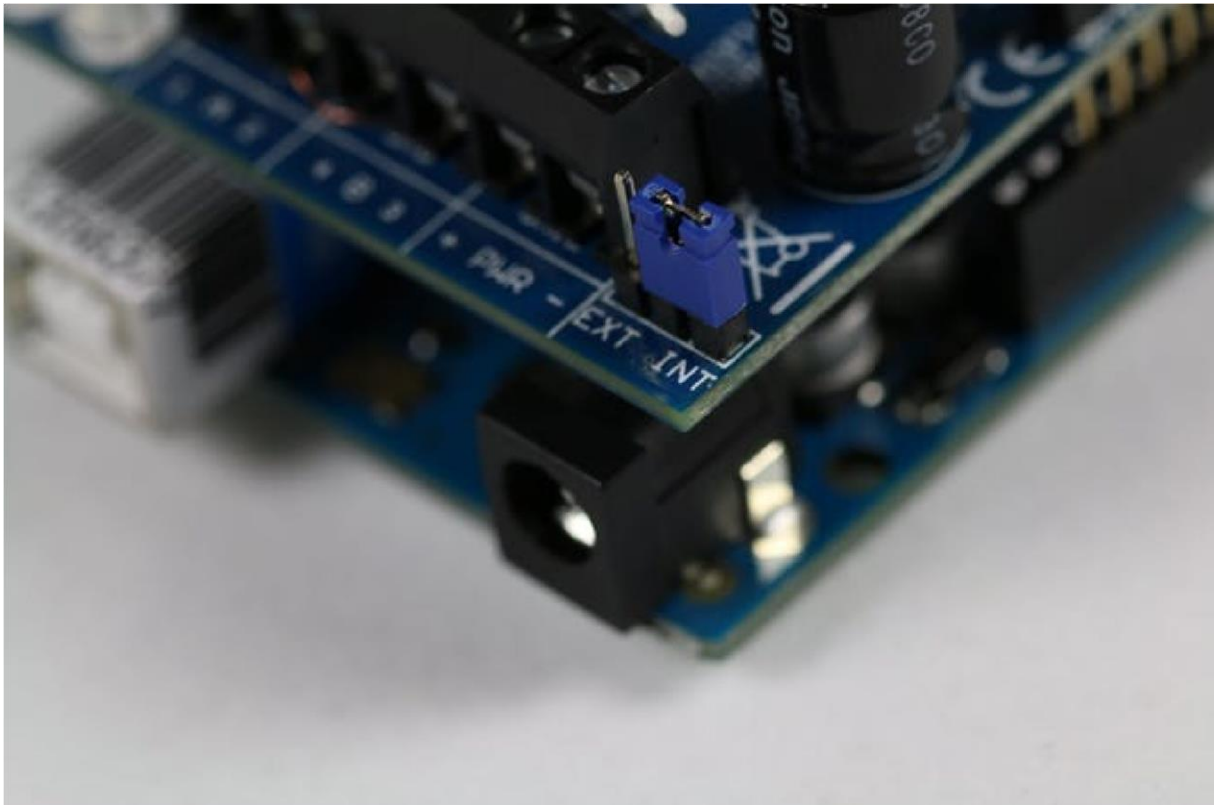
Here are the two frames after they have been separated.



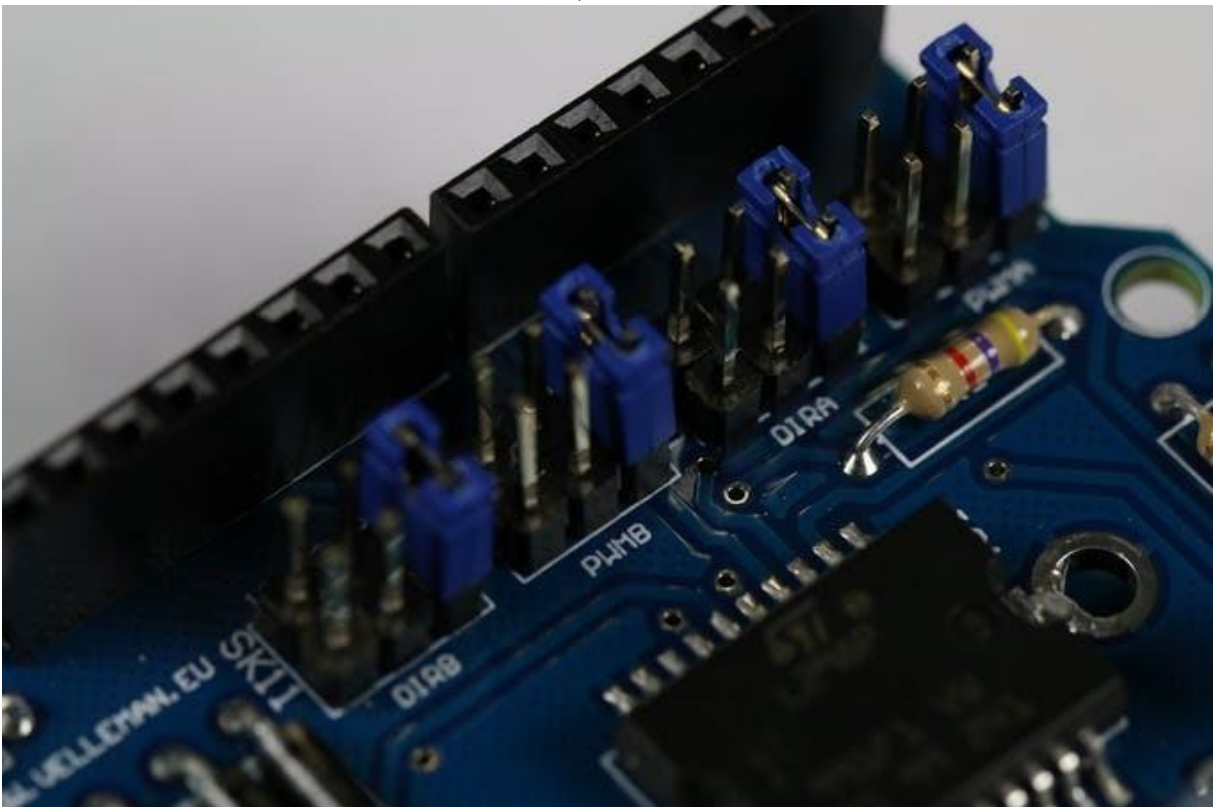
Unscrew the circuit board and cut away the wires. I've also stripped the power and motor wires from the ends so that they may be attached to the Arduino / motor shield.



Here is a photograph of the motor shield mounted on top of the Arduino.



You'll notice that there is a jumper near the power terminal which allows you to select from internal (INT) or external (EXT) power. Since our RC car uses a 7.2v battery and the current load is more than sufficient for both the motors and the Arduino, ...

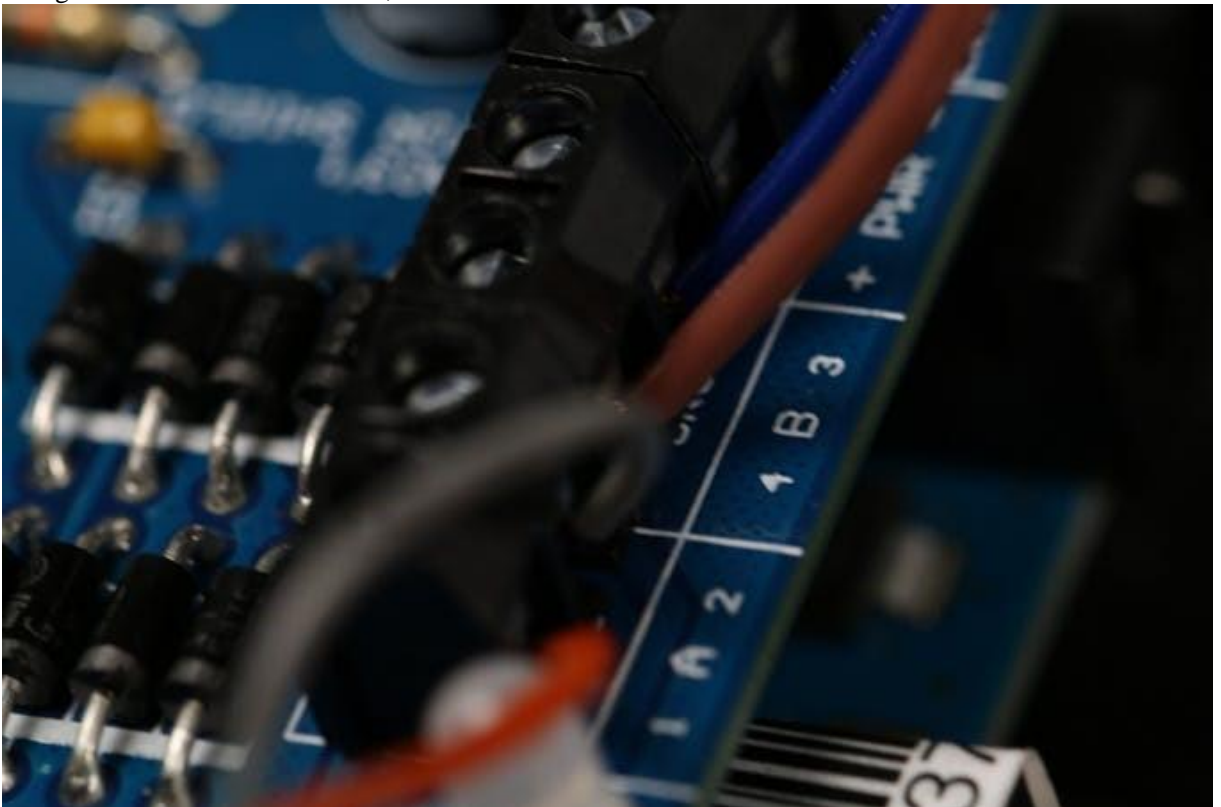


You'll also notice several jumpers near the right pinouts which allow you to choose which control pins you want to use to drive the motors. I am using 2 + 3 for one motor, and 8 + 9 for the other.

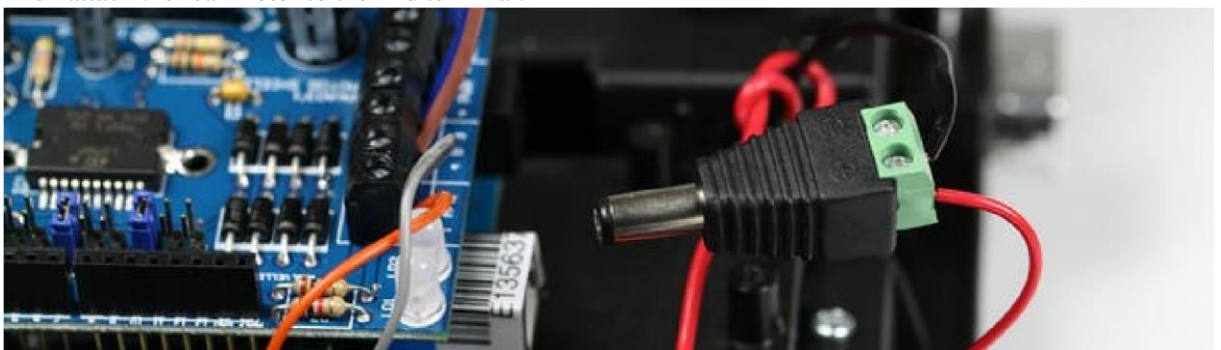


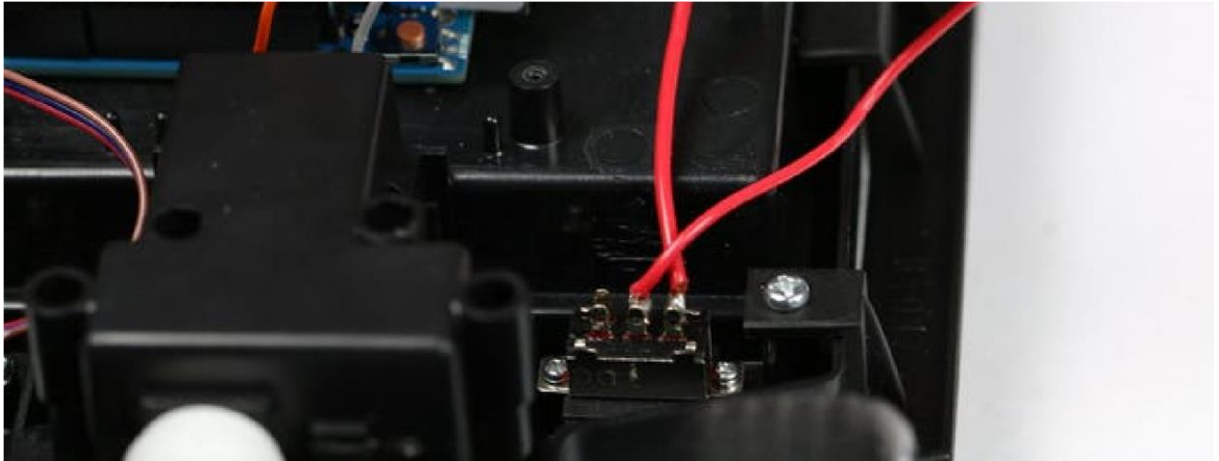


Using a mini-flathead screwdriver, attach the front motor wires to the nearest terminal.

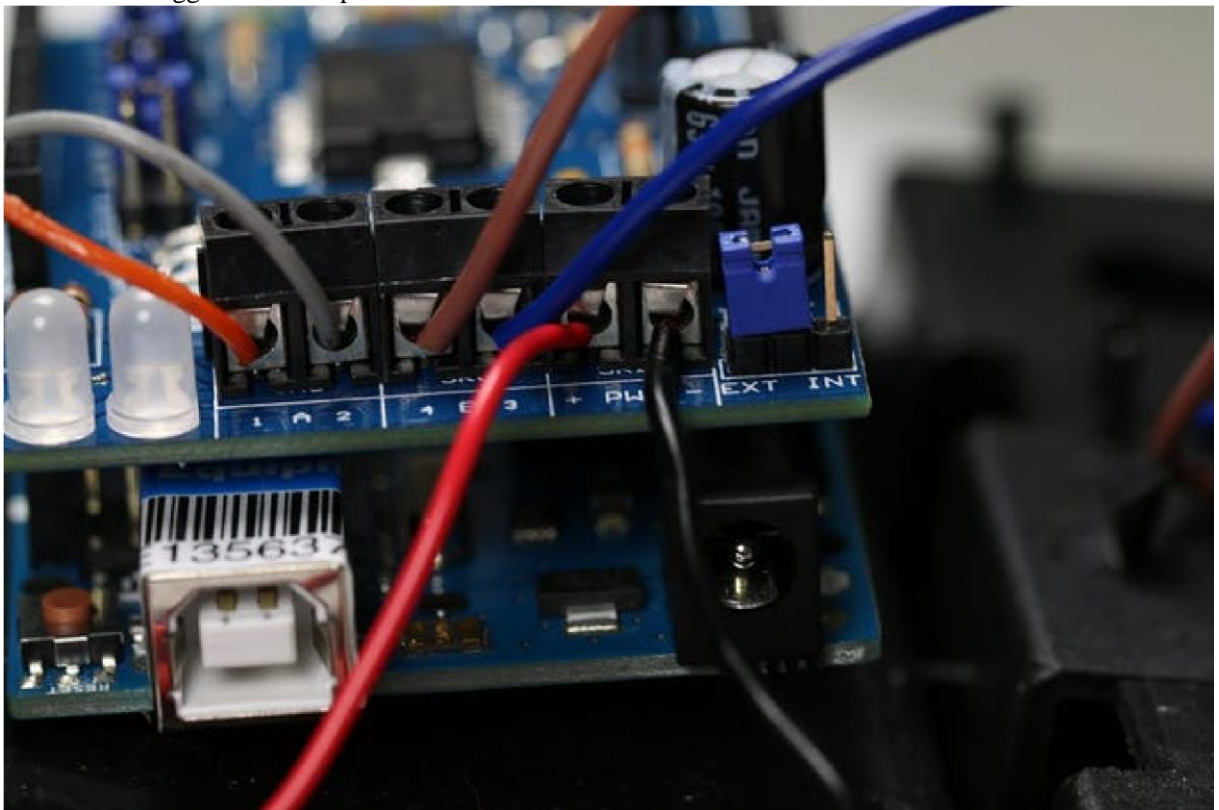


Then attach the rear motor to the 2nd terminal.

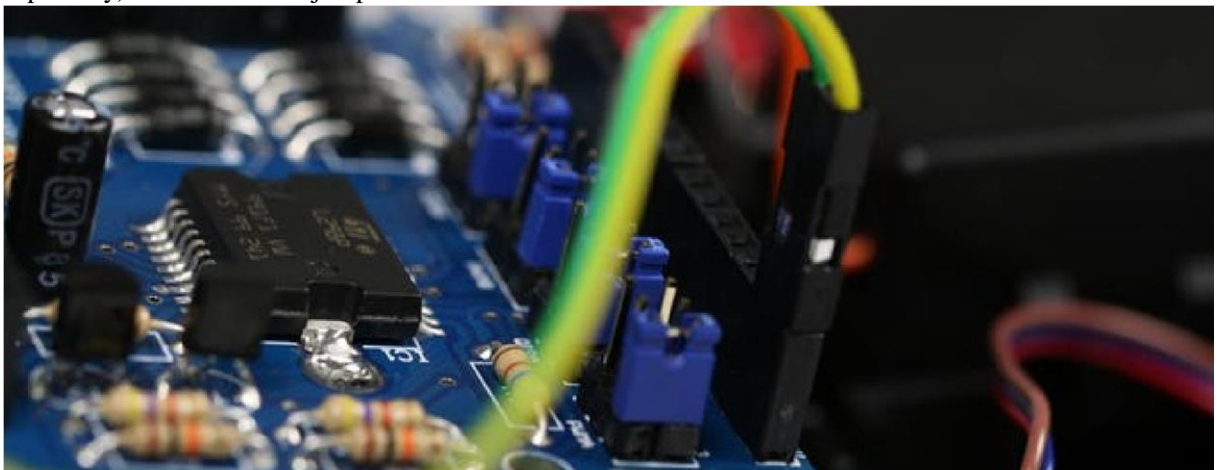


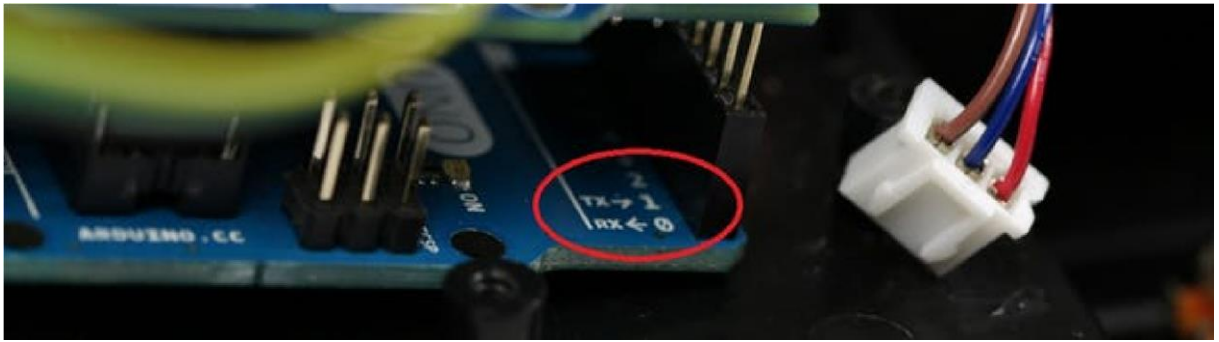


Since we are using internal power, we've attached a barrel jack to the power leads. Notice that we used the positive lead output from the power switch. This allows both the Arduino and the motor shield / motors to be toggled with the power switch.

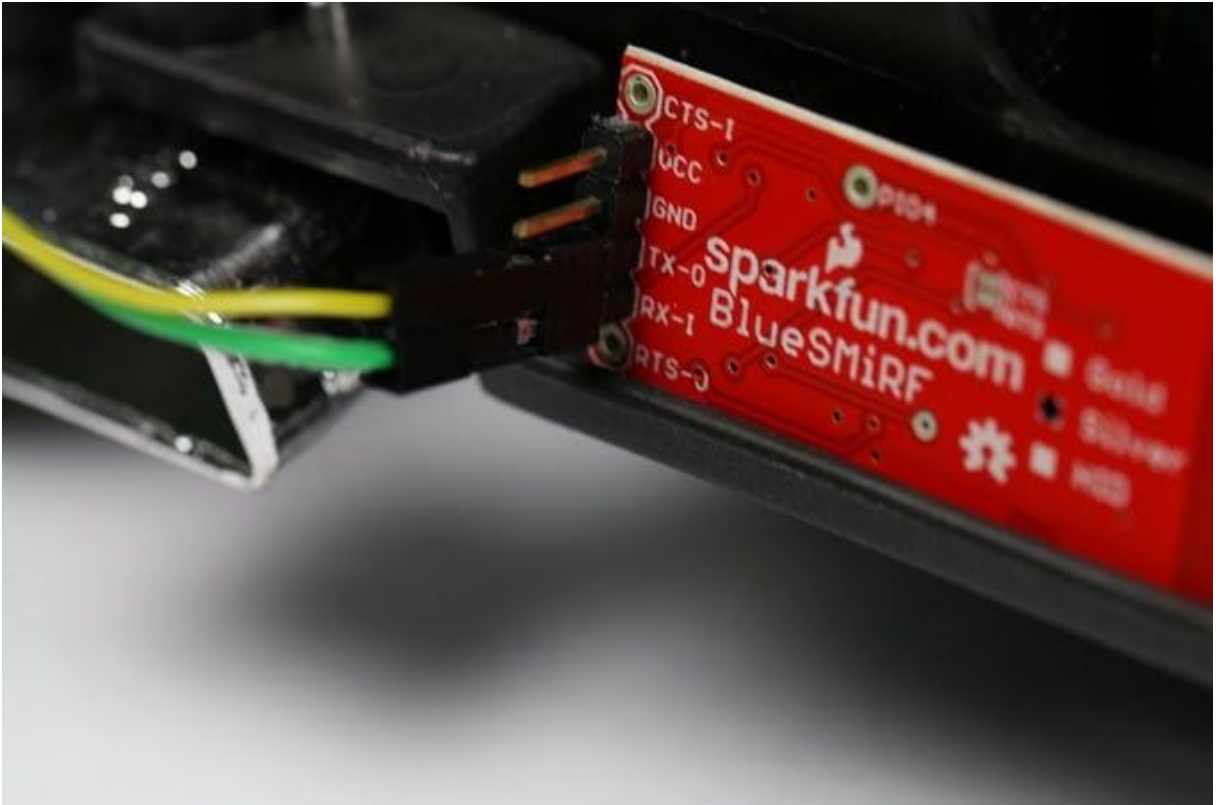


Here is what the attachments would look like if using external power (you'll need to power the Arduino separately). Notice that the jumper is now on the EXT side.

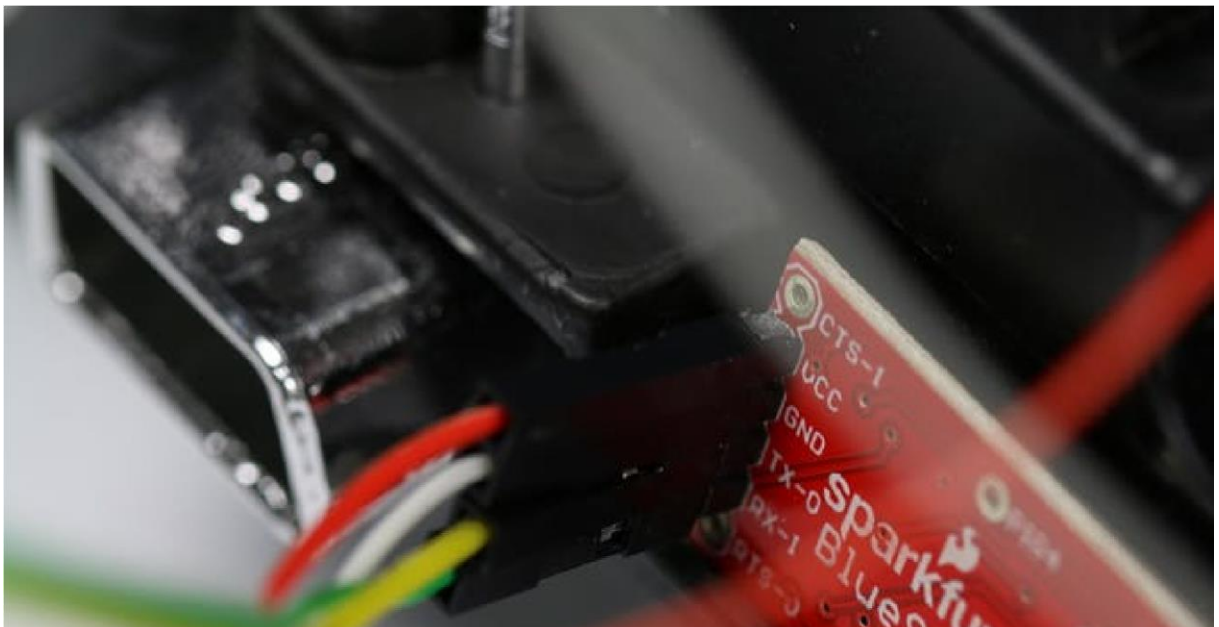




Attach two wires to pins 0 + 1 for the Bluetooth device. Using different colors will help to differentiate the wires. Make note of which color is which for the Arduino.



Attach the wires to their opposite-corresponding pins on the Bluetooth device. RX -> TX and TX -> RX.





Dont forget to also attach the Bluetooth device to power and ground on the Arduino.