## Cushioned Start Modification

## Summary

As part of Mercer’s adaptive toy engineering team I was tasked with developing a cushioned start to control acceleration. This project uses an arduino and motor controllers to accomplish this allowing for further modification and control. In the following tutorial I will be using a red Maserati but this could easily be applied to most 6V and 12V vehicles.

## Parts List

Red Maserati

Arduino Uno/Nano

20-22 Ga wire

12V Motor Controller compatible with Arduino

* I used a Cytron MD10C DC Motor Driver

Modified Button

12V Step Down Regulator (Required for 12V vehicles)

Multi-position switch (Optional)

Wire Cutters

Soldering Iron and Solder (Recommended)

Electric Tape/Wire Nuts/Heat Shrink

A selection of screwdrivers

## Hardware Installation



First open up the seat and find all of your wires. Go ahead and put the battery in place and find which side the wires lead out to.



Next go to the underside of the car on the side where the wires are running and find the main power and ground. On the red Maserati this is underneath the black plate that is screwed on underneath.



Cut two wires of different colors roughly the distance from these main connections to underneath the seat following the other wires. I did this by feeding them through the seat then cutting them off at a generous length.



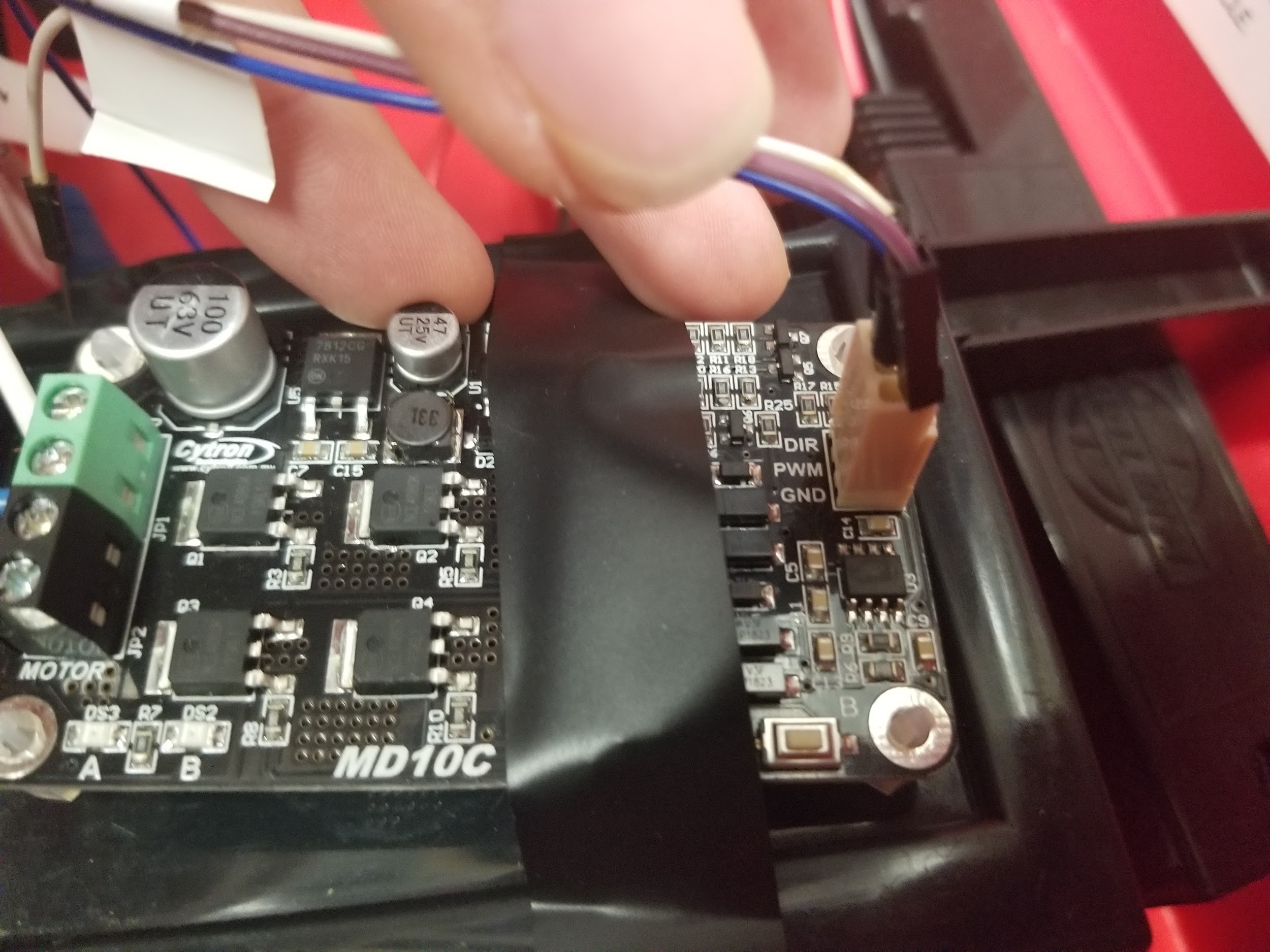
Mark these as Arduino power and ground. Now if you have a 12V vehicle this is where you would install the 12V dropdown and then feed power and ground back out labeling these new wires as ardu power and ground.



We will wire the motor controller next. The first thing to do is to cut the wires going into the motor. Ensure the positive end goes to the positive end of the motor controller. It is ok if you get this mixed up but your motor will drive the opposite direction of your code so keep this in mind.



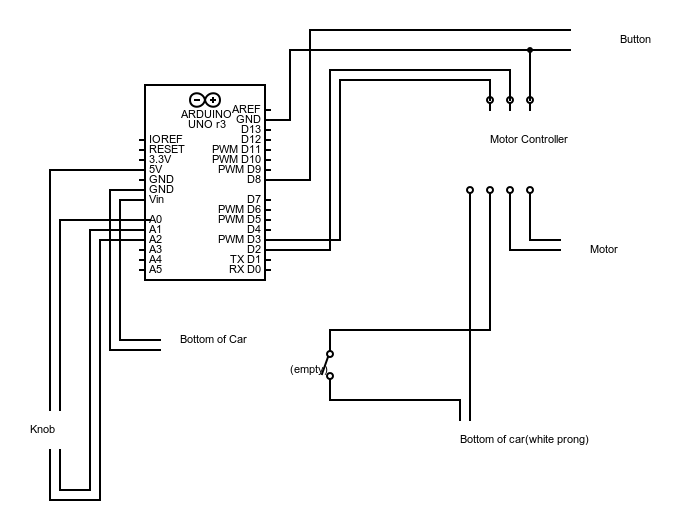
Next feed the two ends that were attached to the motor into the power supply of the motor controller. Remember to ensure positive to positive and ground to ground on the motor controller.



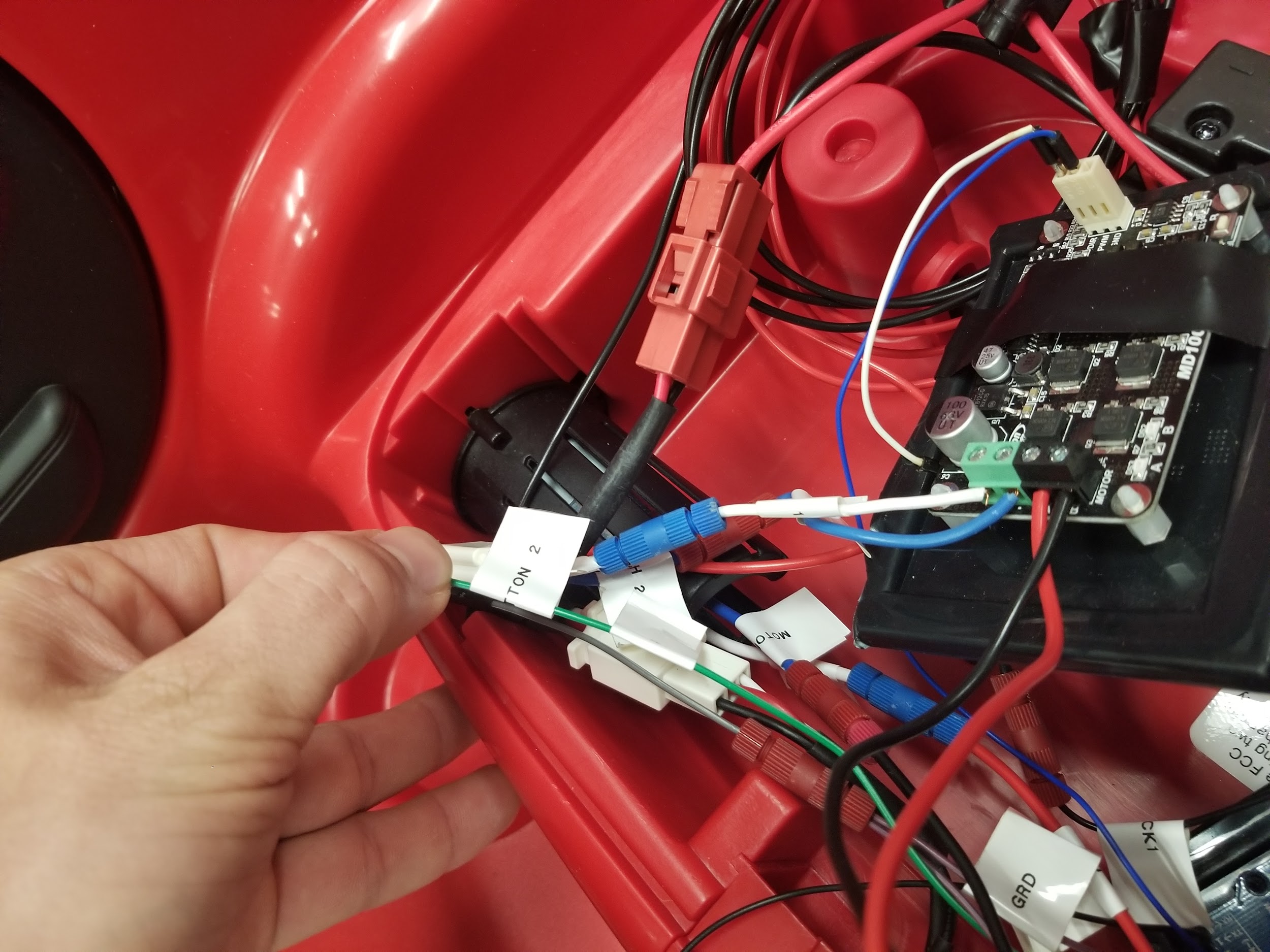
Now for the motor controller I used came with a special connector going to DIR, PWM, and GND from top to bottom in the picture to the left. Solder an arduino compatible wire to each port and plug it in. After this is done it simply plugs into the motor controller. On the arduino side you must ensure your PWM wire goes to a PWM pin denoted by a ‘~’. See Table 1 for the arduino pins I use and where they go.

| **Arduino Pin** | **Connection** |
| --- | --- |
| 2 | DIR pin on motor controller |
| 3 | PWM pin on motor controller |
| 8 | Connected to one side of button wire |
| A0 | Connected to analog multi-level switch |
| A1 | Connected to analog multi-level switch |
| A2 | Connected to analog multi-level switch |
| 5V | Connected to analog multi-level switch |
| Vin | Connected to wire pulled from car’s main ground ( Red ) |
| Gnd | Connected to analog multi-level switch |
| Gnd | Connected to GND pin on motor controller |
| Gnd | Connected to wire pulled from car’s main ground ( Black ) |

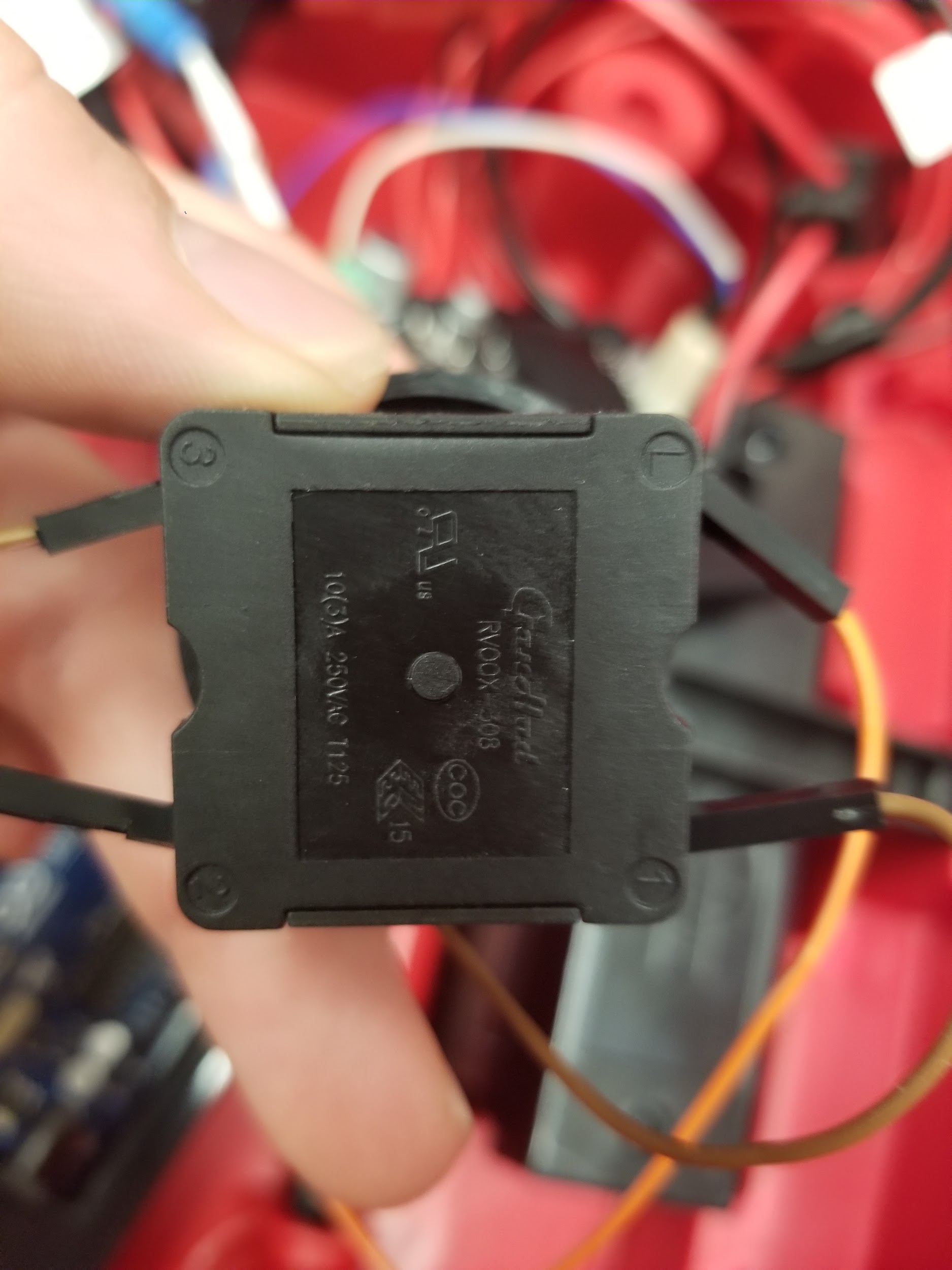
**Table 1: Connection Table**

****

Wiring diagram



Now I am using a modified button with two wires coming off of it. For me this was developed by another team and they fed small wire to the seat upon my request. These two cables go directly into the arduino and looking on Table 1 you can see my I/O pin mapping.



Now if you would like to add a multi-position switch to allow for different acceleration settings I used the switch seen to the left. And wired it into three different pin on the arduino and then a Vout pin. I have included this mapping on Table 1. You may need to branch a Vout pin if you include this to allow for each peripheral to be powered.



Finally wire the arduino power and ground wires into Vin and ground and all the hardware should be fully wired. At this point I would connect the dashboard and hit the power button to ensure the new electronics are powered.

Now we are ready to flash the arduino. Find a USB Blaster cable and hook it up to your PC.

## Software Installation

You may read <https://www.arduino.cc/en/main/howto> to learn how to load code onto an arduino.

A copy of the code we used can be seen below.

//Define Pins

#define pwm 2

#define dir 3

#define pot A1 //For State Switch

#define buttonPin 8

int state = 1;

int pinA = 7;

int pinB = 6;

int pinC = 4;

int i = 179;

void setup() {

pinMode(buttonPin, INPUT\_PULLUP);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(buttonPin, HIGH);

if(digitalRead(buttonPin) == LOW){

Serial.print("Button is Pressed");

digitalWrite(pwm,HIGH);

//PWM is valued between 0 and 255, must stay between these

if(state == 0)

{

//Cut the Motors

analogWrite(dir,0);

delay(1000);

}

else if(state == 1)

{

//For loop to ramp up speed

analogWrite(dir,i);

delay(100);

if(i > -255)

{

i--;

}

else

{

i = -255;

}

}

else if(state == 2)

{

//Fast State

//For loop to ramp up speed

analogWrite(dir,i);

delay(100);

if(i > -254)

{

i -= 2;

}

else

{

i = -255;

}

}

}

else

{

Serial.print("Buton is Not Pressed");

digitalWrite(pwm,LOW);

i = -179;

analogWrite(dir,0);

}

//while(1) continue;

}