Benzon Carlitos Salazar
CS412

Midterm

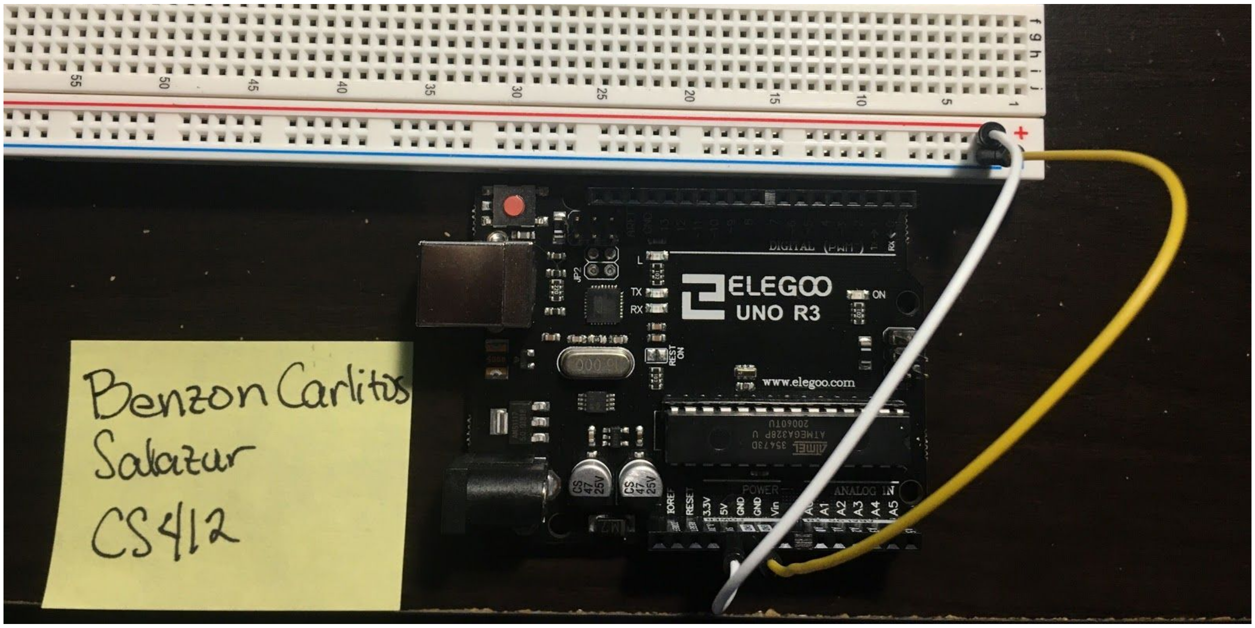
A quick note, my DC motor is completely broken, even when I try to use external jumper wires to act as the original wires of the DC motor. Nevertheless, the whole report is described as though we have a working DC motor.

Supplies:

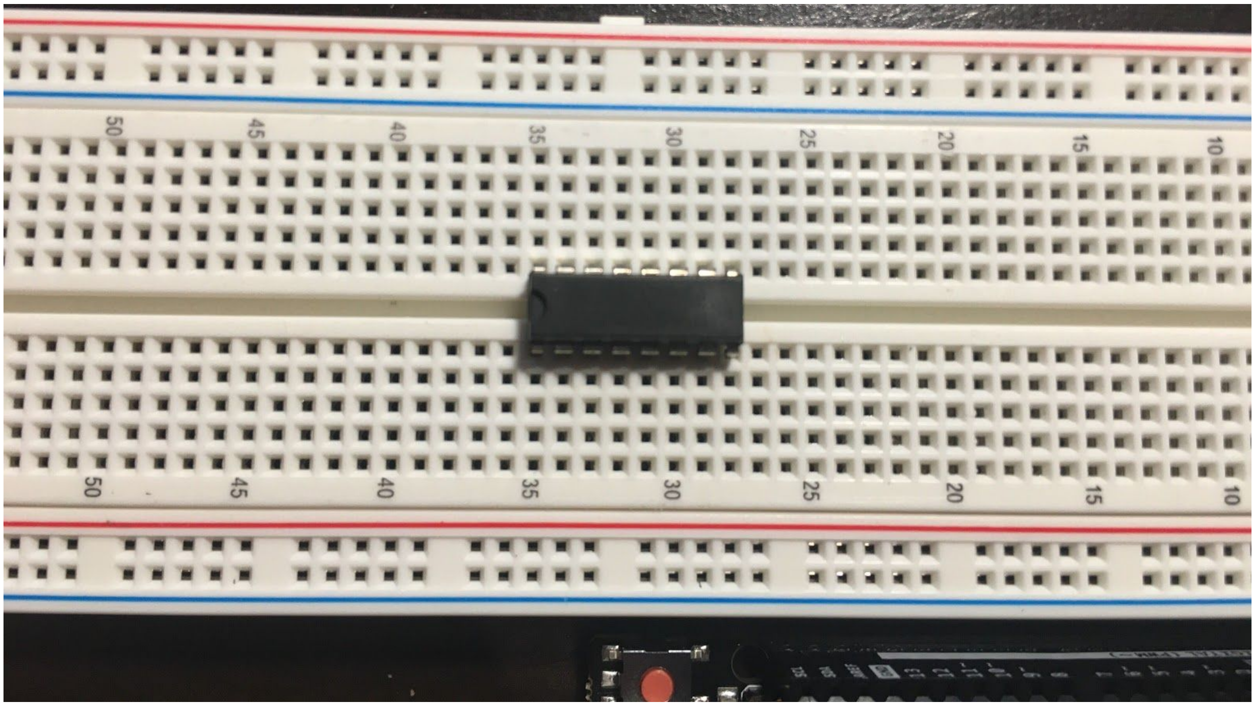
- 1 x 830 Tie-Points Breadboard
- 1 x UNO R3 Controller Board
- 1 x DC Motor -- ***Two blue jumper wires will represent the terminal wires of the DC motor***
- 1 x H-Bridge Circuit
- 1 x Button
- 9 x Breadboard Jumper Wires (***+2 more to represent the terminal wires of DC motor***)

The Circuit:

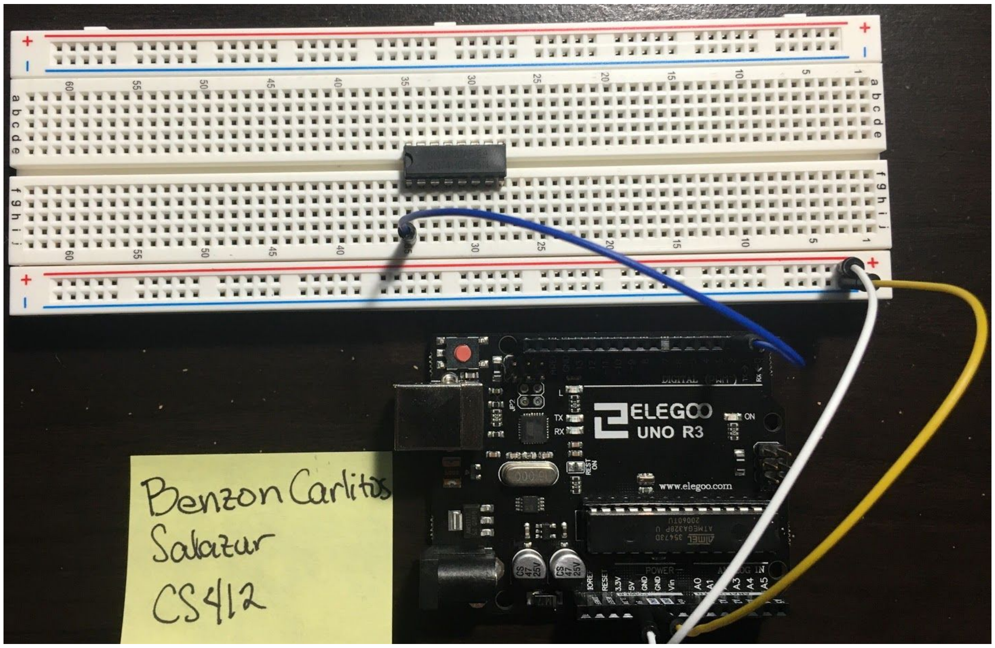
First, connect a jumper wire from the 5V pin of your Arduino to the positive horizontal row of your breadboard, and connect another jumper wire from the GND pin of your Arduino to the negative horizontal row of your breadboard.



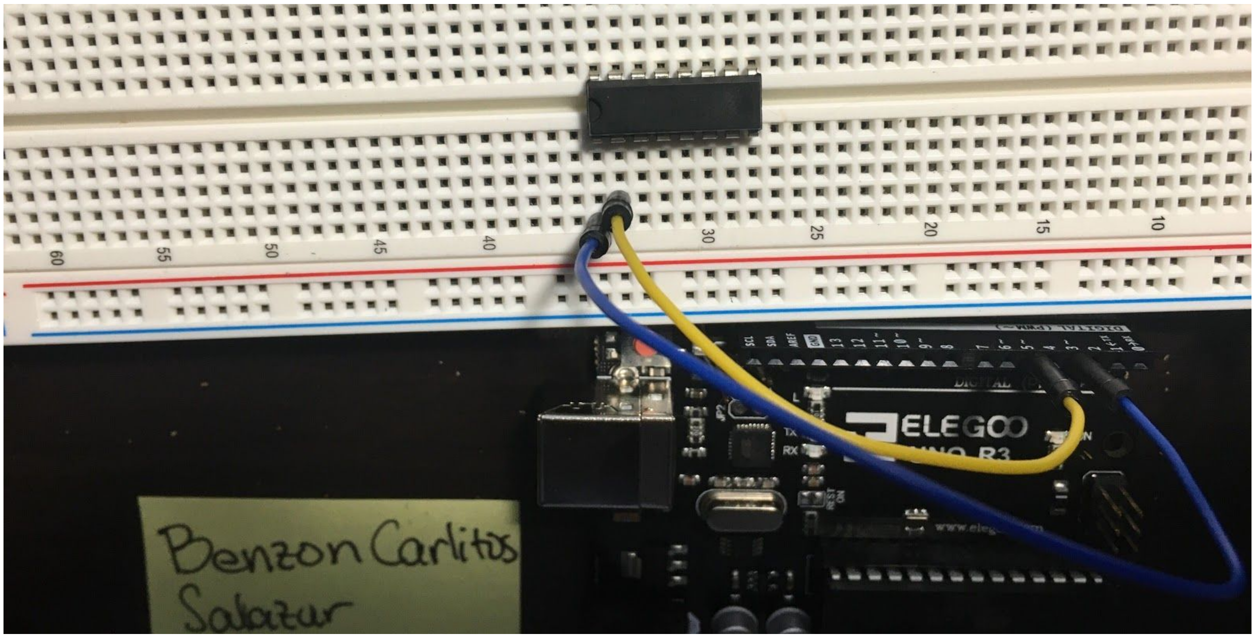
Next, place your H-Bridge Circuit on the breadboard such that the little half circle shape is on your left hand side.This is an important distinction because the bottom row pins are arranged from, left to right, pin 1 to pin 8.



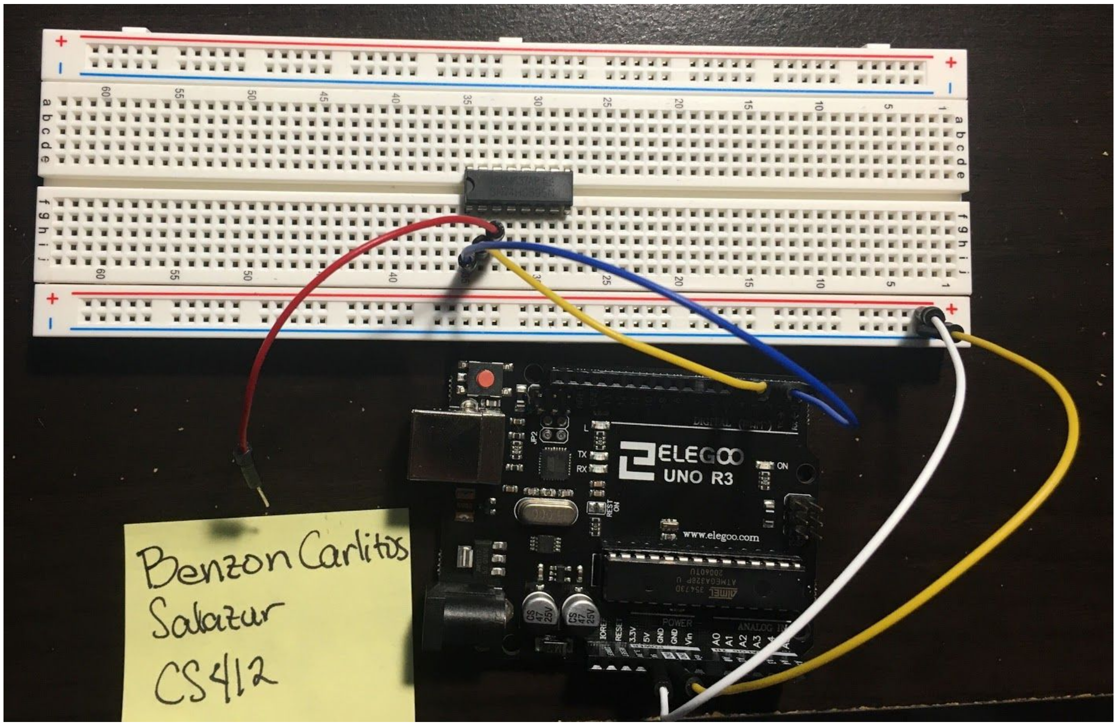
Next, using a jumper wire, connect pin 1 of the H-Bridge to pin 3 of the Arduino, this functions as the speed of rotation of our DC motor.



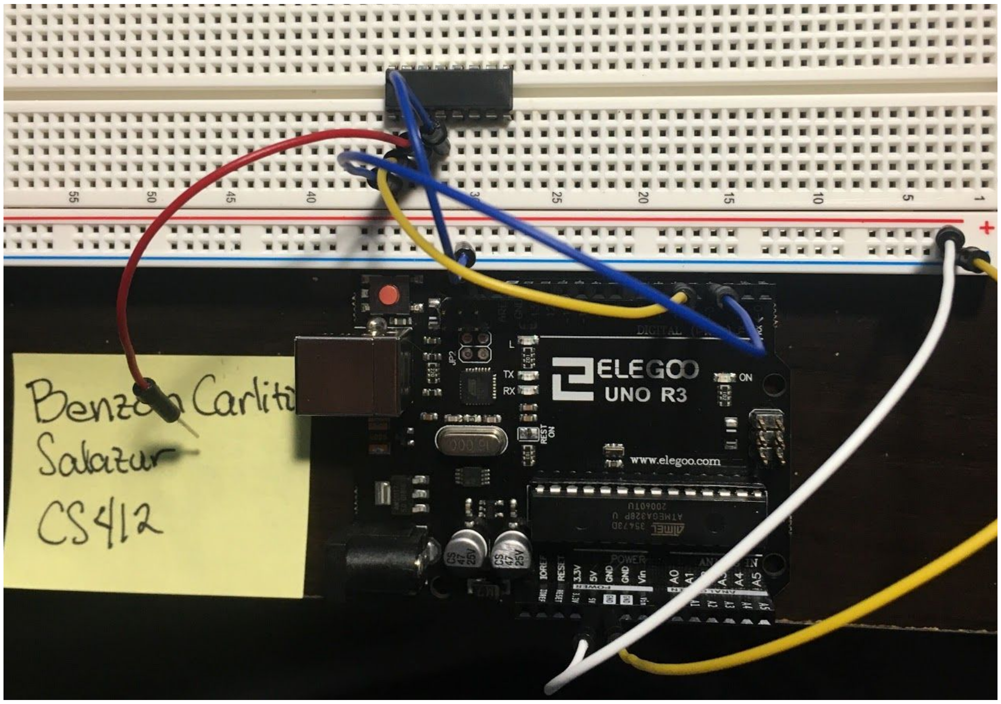
Next, connect pin 2 of the H-Bridge to pin 5 of the Arduino, this functions as one of our inputs.



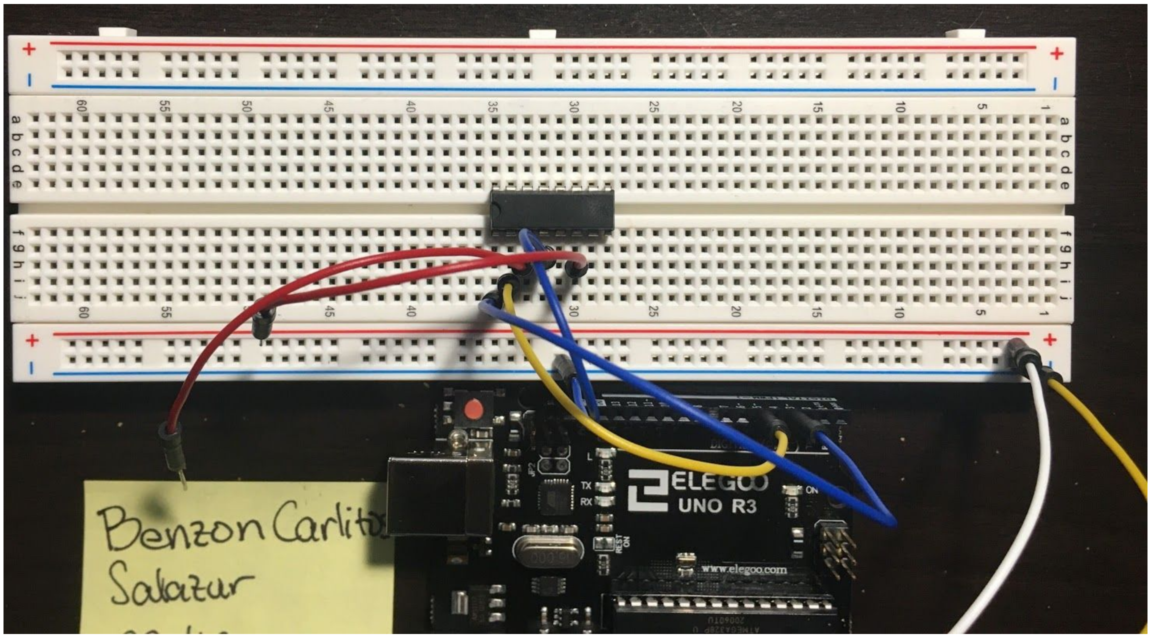
Next, connect one of the motor terminals to pin 3 of the H-Bridge, in our case, a **red jumper wire** will represent the terminal, this will function as one of the outputs.



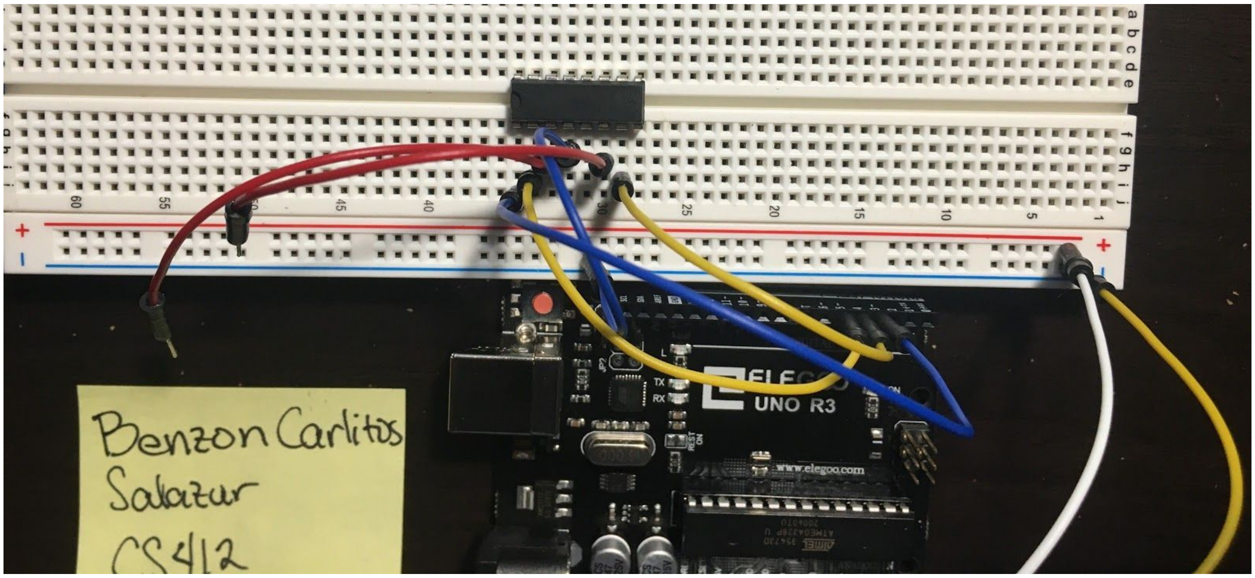
Next, connect pin 4 of the H-Bridge to the GND.



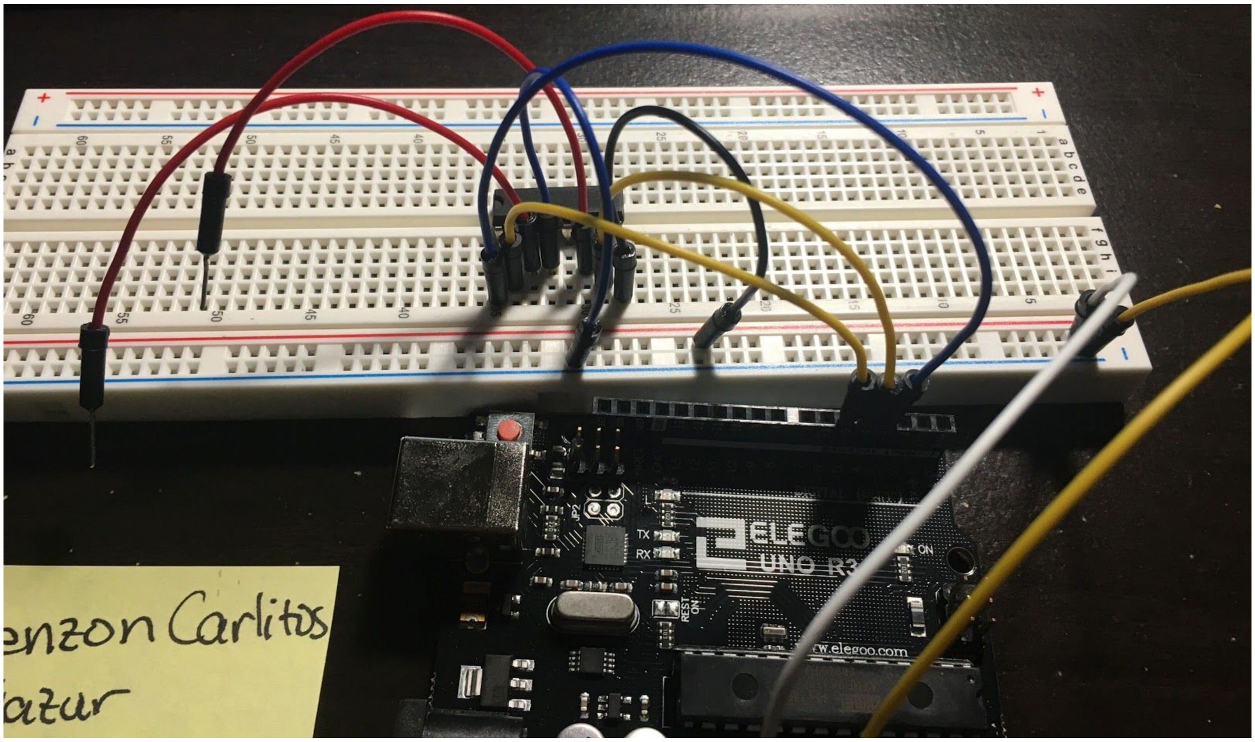
Moving along, connect pin 6 of the H-Bridge to the motor, in our case, a **red jumper wire** will represent the terminal, this will function as one of the outputs.



Next, connect pin 7 of the H-Bridge to pin 4 of the Arduino. This functions as one of our inputs.



Finally, connect pin 8 of the H-Bridge to the 5V, this functions as our power.

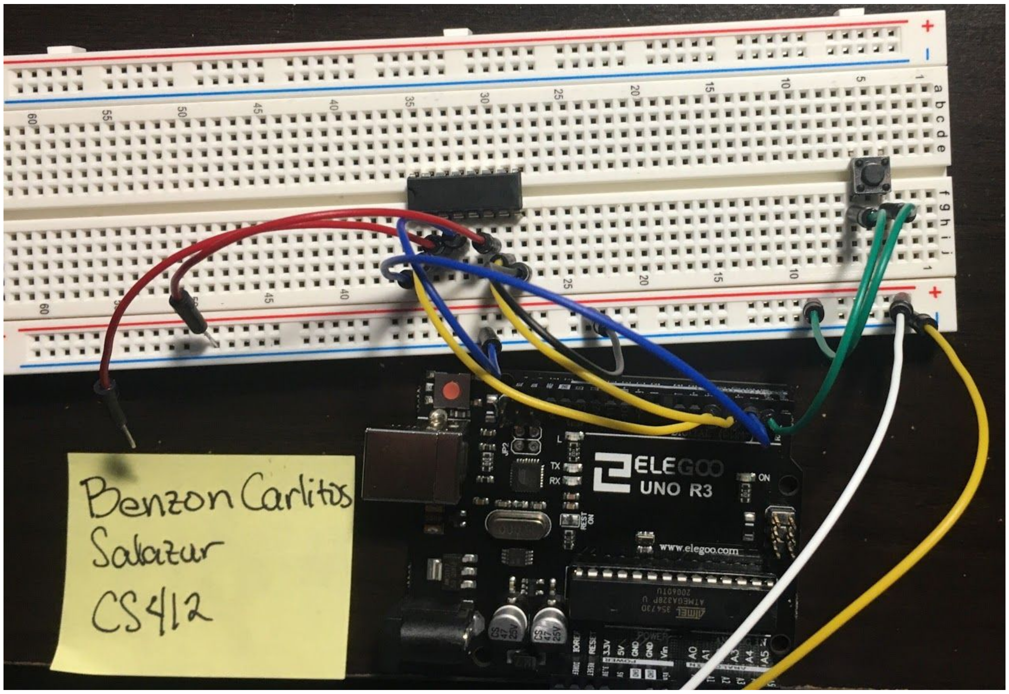


Here is a table summary of each of the connections from H-Bridge to DC motor and Arduino.

H-Bridge Circuit	**Arduino**	**Function**
Pin 1	Pin 3	Speed
Pin 2	Pin 5	In
Pin 3	Motor	Out
Pin 4	GND	GND
Pin 6	Motor	Out
Pin 7	Pin 4	In
Pin 8	5V	Power

Connecting the button:

Pin the button to the breadboard and connect one terminal C to pin 2 and terminal D to the 5V pin.



Connect your Arduino to your computer via the USB cable, and run the code below.

Code:

```
int enablePin = 3;
int in1Pin = 4;
int in2Pin = 5;
int switchPin = 2;

void setup() {
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(switchPin, INPUT_PULLUP);
}

void loop() {
  boolean reverse = digitalRead(switchPin);
  setMotor(reverse);
}

void setMotor(boolean reverse) {
  analogWrite(enablePin, 200);
  digitalWrite(in1Pin, !reverse);
  digitalWrite(in2Pin, reverse);
}
```

Explanation:

We can control the speed of the DC motor by simply controlling the input voltage to the motor and the most common method we do that is through the PWM signal. That is why we have `analogWrite(enablePin, 200).` This means that we want to enable the rotation of the motor at 200 RPM. On the other hand, when we want to control the motor's direction of rotation, we need to inverse the direction of the current flow through our DC motor, and we achieve this by using an H-Bridge circuit.

The H-Bridge circuit contains four switching elements, and by activating two particular switches together at the same time, we can achieve a change in the direction of the current flow, thus helping us change the direction of the rotation of the DC motor. When we want this change in flow, we do a push switch, which is what `switchPin` on pin 2 is. This variable is what literally changes the direction of current through our DC motor.

So, combining the PWM and H-Bridge, we can have complete control of the speed of our DC motor, as well as its direction of rotation.