**Beam Degrader Electrical Configuration:**

This repository contains the PCB and Schematic for an 8 channel H-bridge system intended to drive 8 Actuators for the Beam Degrader project. There will be an input of 12volts and 3.3volts.

"PCB1" is the 4 layer Beam Actuator Driver Board (GND Plane and Power Plane)

"PCB1 -2Layer" is the 2 layer version of the Beam Actuator Driver Board where there is no GND or Power Plane. In order to accommodate the missing 2 planes, we made the trace thickness to be 30mils(which is still changeable)

1. When you push the button, then the Actuator will extend or retract based off the reading from the actuator state

* The actuator state is determined by the internal potentiometer within the Actuator. The Wiper of the potentiometer is fed as an Analog input into the Arduino (A4) and the two references for the potentiometer are 3.3v (yellow) and GND (white).
  + When the ~3.30v is read it means actuator is FULLY EXTENDED
  + When ~ 0.00v is read it means actuator is FULLY RETRACTED
* The button is connected into a pull up resistor fashion, with a 1.2k resistor and the 3.3v as the input voltage being fed into the Arduino
  + When the button is NOT pressed, the button input should read HIGH
  + When the button IS pressed, input should read LOW (since pull up resistor)
* After the potentiometer state is determined based off of the wiper input to the Ardunio, the logic is sent out to either extend or retract the actuator
  + If it is already extended(voltage > 2v), then to retract it, it sends out logic 01
  + If it is retracted(voltage < 1v), then we need to extend it, sends out logic 10

1. H-bridge: it has seven legs coming out.

1-Output 1 (output to motor wire RED)

2-Error Flag

3-Input 1 (input to H-bridge from Ardunio Digital pin 4)

4-GND (connected to ardunio ground)

5- Input 2(input to H-bridge form Ardunio Digital pin 5)

6-Vs (voltage that needs to be applied to power the motor-coming out from pins 1 and 7) the 12 volts coming from the power supply

7-Output 2(output to motor wire BLACK)

* Logic input 00/11 remains same position
* Logic input 01 retracts actuator: output1 = 0v and output2 = 12v
* Logic input 10 extends actuator: output 1 = 12v and output 2 = 0v

\*Troubleshoot with H-bridge, we had everything connected, but the output pins were not changing polarizations after reading input was retracted and need to extend the actuator. We were getting a constant 3.3vs no matter whether we switched the actuator position.

-We realized after that we were simply not inputting enough voltage from the power supply. Since there was no load outputting from the outputs then the voltage was not enough. It worked after we supplied 12 volts to the h-bridge.

-While debugging this error, we noticed that two of the h-bridge legs were touching the whole time and were probably shorting each other out(legs 6 and 7) so we decided to solder on some wires to make it easier to separate.

\*Error Flag: in order to correctly make use of the error flag we need to attach it using a pull up resistor connected to the 3.3v. When there is no error, the input to the arudino will be a logic high. When there is an error detected, the error flag will ground the 3.3v and the arduino will output a logic low. A 2.2k resistor was used. (\*\*Remember never to input anything greater than 3.3v to the arudino because the maximum it can input is 3.3v since that’s the voltage it is powered at!)

1. Arduino code

-working with the Arduino program to read the correct inputs and outputs to the h-bridge to control the actuator.

3 digital I/O pins used: - 2 for outputs to the h-bridge input pins to assign the logic

* + - * 1 for the input of the error flag

1 analog input used: for the wiper from the potentiometer

3.3v used to power the push button and the error flag inputs

The Arduino GND is used as the common ground for everything

\*Trouble Shoot 2 with outputs from H-bridge, after configuring everything correctly and the outputs are both read off the multi-meter correctly, one 12v and the other GND. When connected to the motor black and red wires, the voltage is reading 5.8v on each wire. Then we are receiving a SET error flag. So basically the outputs from the H-bridge are getting turned OFF (and grounded).

-I cannot figure out why the outputs are setting the Error Flag. But if the H-bridge manual goes over all the reasons for the error flag going off.

-After some investigation we were able to determine that as the 12v was connected as an input to the H-bridge (Vs) the ERROR flag was getting set off. As soon as we turned off the power supply, the Error flag turned off. A possible reason the h-bridge was interpreting this as an error was because we thought that as we connected the load, the voltage drop was very great because of the natural resistance and impedance in the wires of the overall circuit, this resulted in a voltage drop across the output of the actuator’s motor (and could have been interpreted as an error because of the voltage difference). In order to prevent this we need to add in the two parallel capacitors which was shown on the data sheet.

* We added in a 100nF capacitor to h-bridge legs 4 and 6
* We added another 100uF capacitor in parallel
* 4=GND and 6=12v(voltage input)

- **BYPASS CAPACITORS**: Adding these bypass 2 capacitors allows us to address the issue of the voltage drop across the motor. When we add these 2 capacitors, we allow the energy stored in each capacitor to act as a “temporary power source” when there is a drop in power coming from the power supply.

-**SLEW RATE**: During the rising edge of a power supply there is what is called a “Slew Rate” which is the (change in Voltage) / (change in Time). This is where the applied voltage is not directly vertical (instantaneous) during the voltage change and has a slight slope to it. In order to accommodate this slight slope we use the bypass capacitor to apply the voltage to push the slew rate higher so that it has a steeper slope to its voltage change rate.

-Esr: Zc = R +sL + 1/sC. This is done by something called Esr, where it is when the impedance from the capacitor and the impedance from the inductor cancel out so that the bypass capacitor functions the best it possibly can. We want this point to be at the same frequency at where the slew rate is so that during this Esr period it can supply optimal voltage to the change in logic voltage.

-Two Bypass CAPS: Adding in 2 bypass capacitors allows us to make the gap for the frequency bigger and will accommodate a larger frequency area where the slew frequency begins.

-The ideal bypass capacitor **Impedance** is: Zc = 1/sC

-The real bypass capacitor **impedance** is: Zc= R + sL + 1/sC, the R and sL are caused by the impedance in the wire and legs of the capacitor. The R is the resistance of the wire and the inductance is caused by the current traveling through the wire which causes some energy to be stored in the magnetic field of the wire. These two impedances are unavoidable in circuits that have wires and play a factor in the voltage loss across the entire circuit.

-After we discovered that the H-bridge may have been destroyed since the outputs were not outputting correctly, I went and used a new H-bridge and soldered on two new capacitors (100nF and 100uF). After connecting this new H-bridge to the circuit the Actuator finally turned correctly.

-ERROR FLAG lighting up initially: However, with this connected we evaluated that if the 12v was initially connected to the H-bridge Vs pin, and then the power supply was turned on, the RED error flag would be set and an error would be detected. But, if we disconnected the 12v coming in and then reconnected it, it would fix the error and no longer read an error. Since the data sheet says that once an error is detected it will keep that output unless a new input is received, we were able to fix this error by adding a portion into the Arduino code whenever there was an error detected. We added that if the error flag is set, then it will output logic 11 to the H-bridge inputs to reset the input. This input was able to fix the error we were receiving.

**HEATSINK**:

On the H-bridge IC, there is a heatsink tab that is mounted to the back of the chip. This can be used to dissipate the heat that builds up in the chip when too much current is running through the chip.

There are three ways to cool down an IC chip. If there is no heatsink tab, the two ways are forced convection and natural convection. Forced convection is where we have a fan or something forcing cool air toward the chip and forcing the hot air away from the chip. This can be a fan, or a hand waving motion or anything that forces cool air toward the chip. The natural convection is due to the earth’s gravitational forces, where if the PCB is open to the air and not closed off inside of a box, then naturally the heat that is coming from the IC is spread to the air above it, then this hot air which is lighter than the cool air flows above the cool air which takes over its place, repeating the cycle and cooling off the chip. The heatsink tab is usually used when the PCB is secluded inside of a box where no natural air can access the chip and the heat must dissipate through another method. If we mount the tab to the PCB and have it dissipate through a copper panel and then to the box that is connected to the copper panel, this way we are allowing the heat to flow from the chip silicon to the heatsink tab, to the copper panel, then to the outside box—leaving the chip cooler.

-We were able to determine that the heatsink tab is also ground by using the multimeter and measuring the resistance of the ground pin to the heatsink tab. The multimeter showed that the resistance was 1.0ohms which was simply the resistance in the wires. If we touched the two wires together we would also see 1.0ohms because touching the tab to ground is simply shorting the two wires together. Because of this, we can simply mount the tab to the board and connect the tab to a large ground trace.



