Electrical Components Report

Last Updated – April, 14, 2021

Group Big Gain, ELEC 391, University of BC, Vancouver, BC, Canada

Abstract

Arduino output pins with voltage range of 0-5 V were amplified using op amps circuits. The amplification enables Arduino control of four independent H-bridges. Each H-bridge drives a motor of nominal voltage 12V. This report outlines the circuit design in section 1, to the measurements that took place for meeting safety and design requirements in section 2, then discusses interfacing MATLAB with the circuit response for computer aided calculations in section 3, and finally summarizes PCB layout design of the finalized circuit in section 4.

Nomenclature

OTS	Off the Shelves
FD	Forward Driving
BD	Backward Driving
TRP	Transient Response Plot
ACA	Alternating Current Analysis
SIA	Step Input Analysis
TFEST	Transfer Function Estimate (function in MATLAB)

1. Designing Circuit Layout & Picking Components

Early in the design process, the team decided to keep the design parameter of nominal operating voltage fixed across all motors as it makes the design process easier to validate. In the Maxon catalog, OTS motors that work reasonably well in 12V condition were selected to be used in this project. The selected motors were two DCX 8 M, one for gripper, one for wrist, one DCX 14 L for elbow of the arm, and DCX 32 L for the base or shoulder of the arm. [1]

The first step was to amplify two Ardiuno pins from 0-5V to 0-12V for each motor. All amplification circuit layout consists of a simple op amp OPA544T, rated for 2A and 35V maximum. [2] OPA544T is chosen for all four circuits as the performance is high and the saturation voltage encompasses the power amplification required for driving each individual motor H-bridges, which ranges from 13-30V (due to voltage drop across diodes and BJTs, it has to be bigger than nominal 12V). Each H-bridge requires two amplification circuits to achieve both FD and BD modes of the motor. R1/R2 ratio is first calculated by hand then tweaked to achieve near 11.9V driving voltage across the motor resistance at the H-bridges.

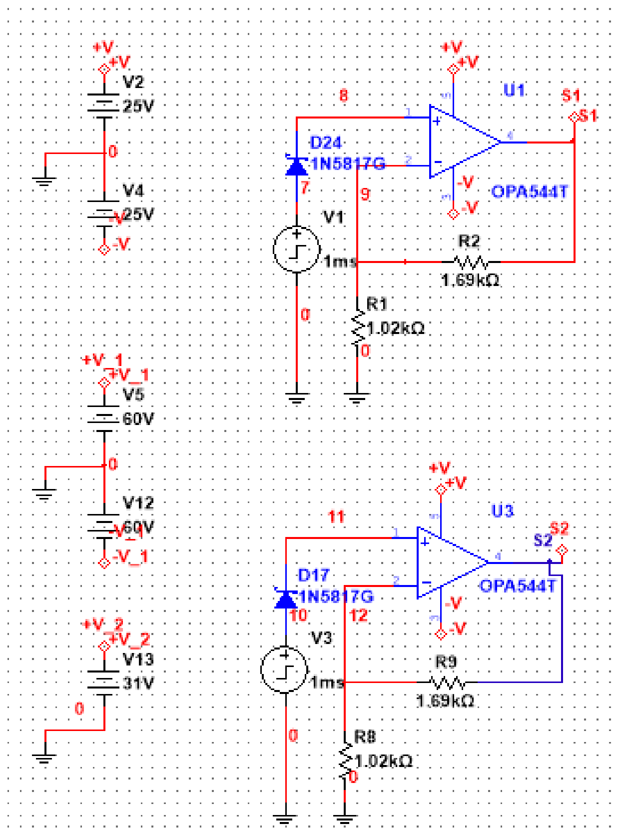


Figure 1.1. Dual Amplification of Arduino pins for FD (bottom) and BD (top) modes of Gripper Motor H-Bridge

H-bridges are also designed with reference to standard H-bridges. With chosen buffer resistors of low resistances of 7.87 to 10Ω that feed current into the BJTs that control the path of power flow of FD and BD modes. The BJTs are chosen for their power rating and least voltage drop while maintaining low cost. For the gripper motor, wrist motor, and the elbow motor H-bridges, PNP BJTs 2N5684G - rated for 50 A, 80 V - are used, and NPN BJTs 2N5686G - also rated for 50 A, 80 V- are used. For only the shoulder/base motor H-bridge, a more powerful PNP BJTs FZT968 for high performance under high current was used to ensure safety under high current draw. All H-bridges and power sources are equipped with a safety schottky diode 1N5817G, rated for 1A and 20V.

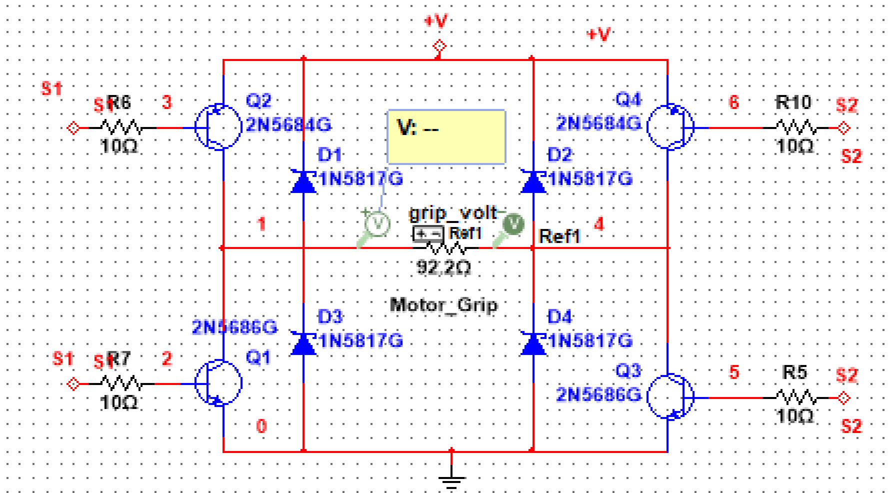


Figure 1.2. Gripper Motor H-bridge

2. Testing Circuit Output & Safety Check

After the circuits are completed, an ACA was conducted on circuit in figure 1.1. with all FD Arduino pins set to AC voltage source 0-5 V. the results of the individual amplifiers are attached below, showing perfectly sinusoidal output response to sinusoidal input. No clipping effects were observed in the amplifiers tested.

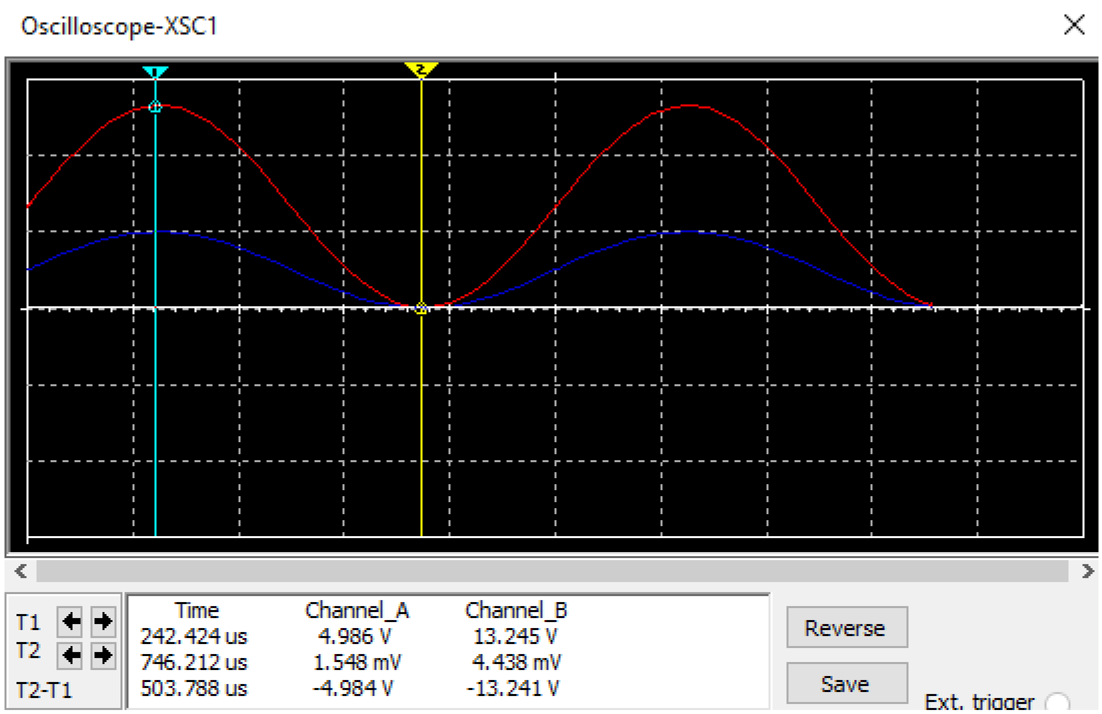


Figure 2.1. ACA of Gripper Amplifier (5V to 13.3V) - Input Blue (Ch A), Output Red (Ch B)

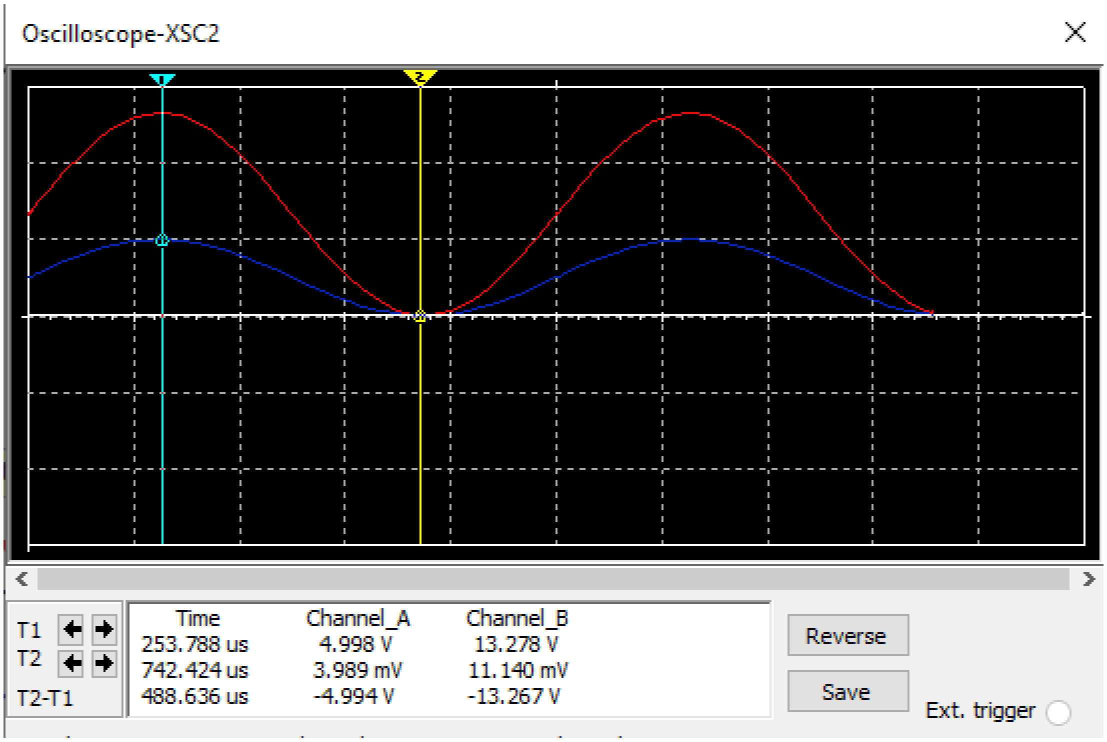


Figure 2.2. ACA of Wrist Amplifier (5V to 13.3V) - Input Blue (Ch A), Output Red (Ch B)

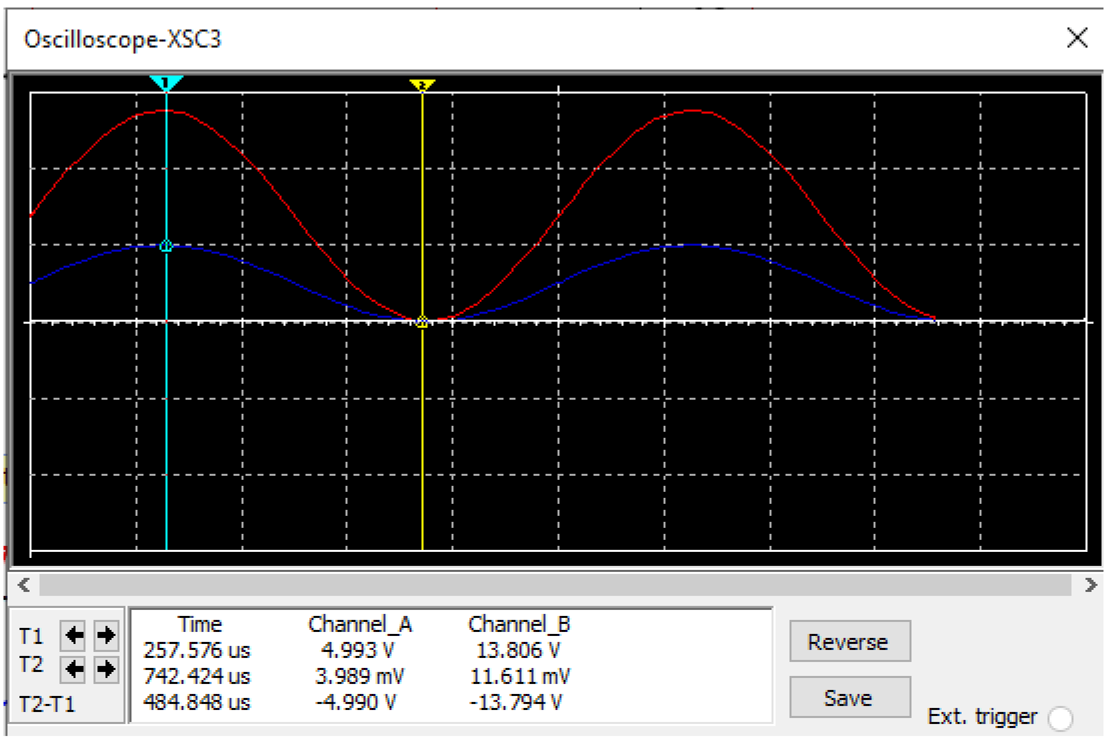


Figure 2.3. ACA of Elbow Amplifier (5V to 13.8V) - Input Blue (Ch A), Output Red (Ch B)

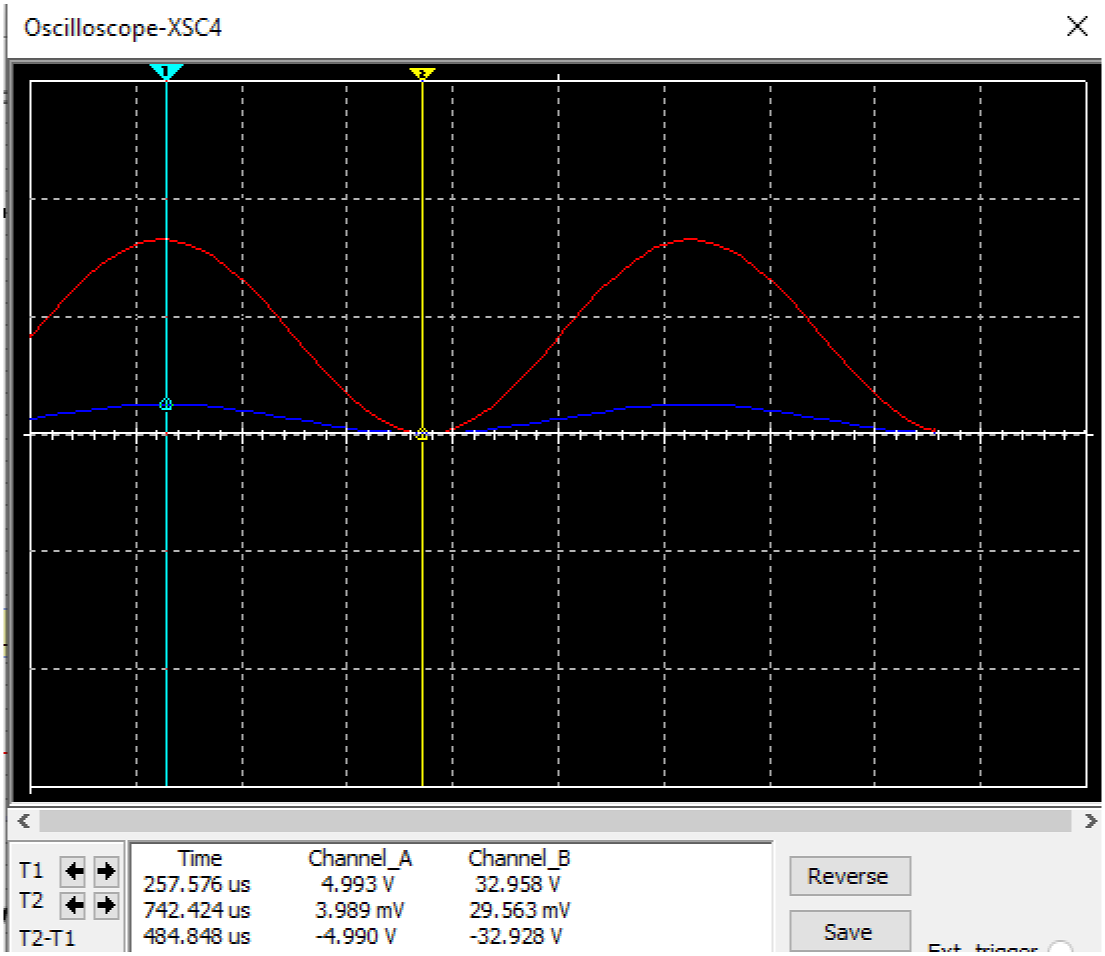


Figure 2.4. ACA of Base Amplifier (5V to 33V) - Input Blue (Ch A), Output Red (Ch B)

Along with the ACA, a SIA of max input voltage 5V in FD mode is conducted. Figures attached below.

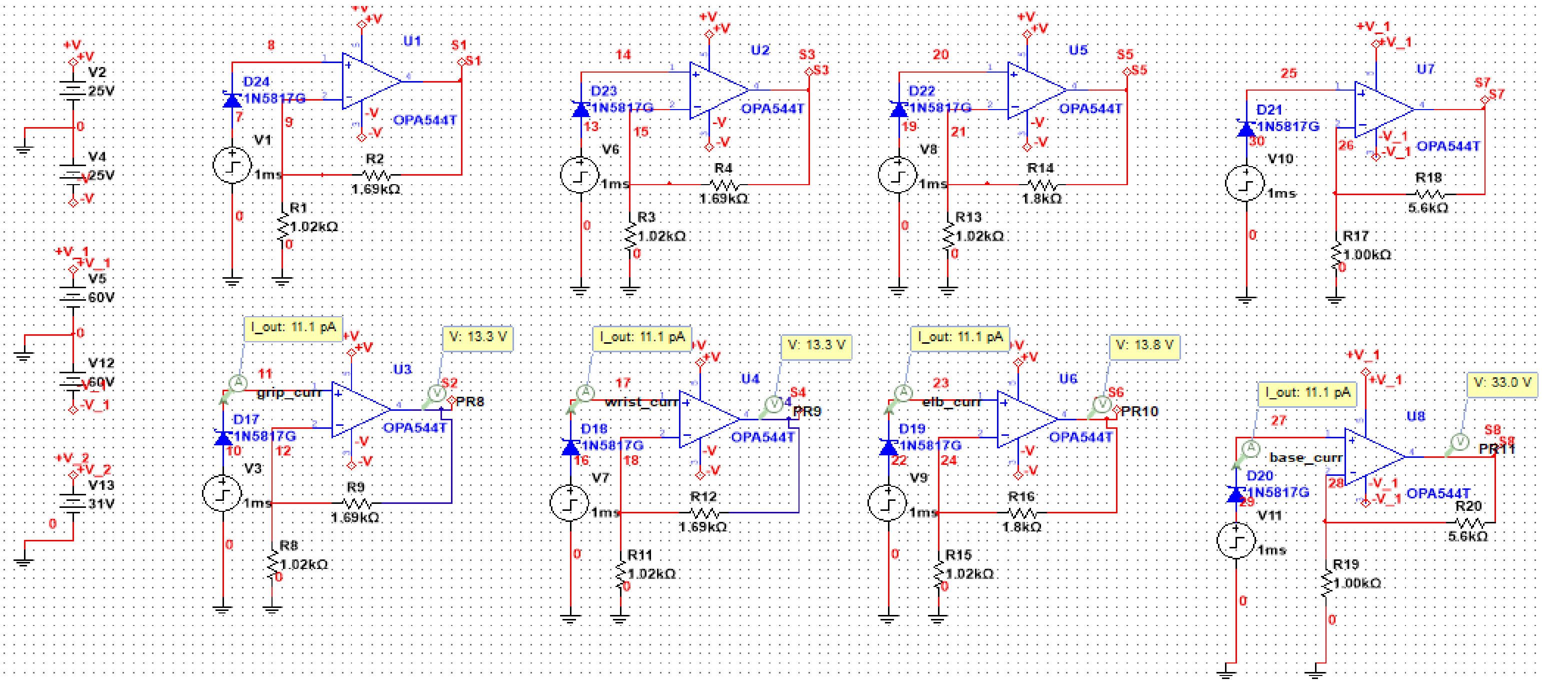


Figure 2.1. SIA of Voltage Amplified for Each H-bridges from 5V Arduino Step Input
(From left to right: gripper, wrist, elbow, base)

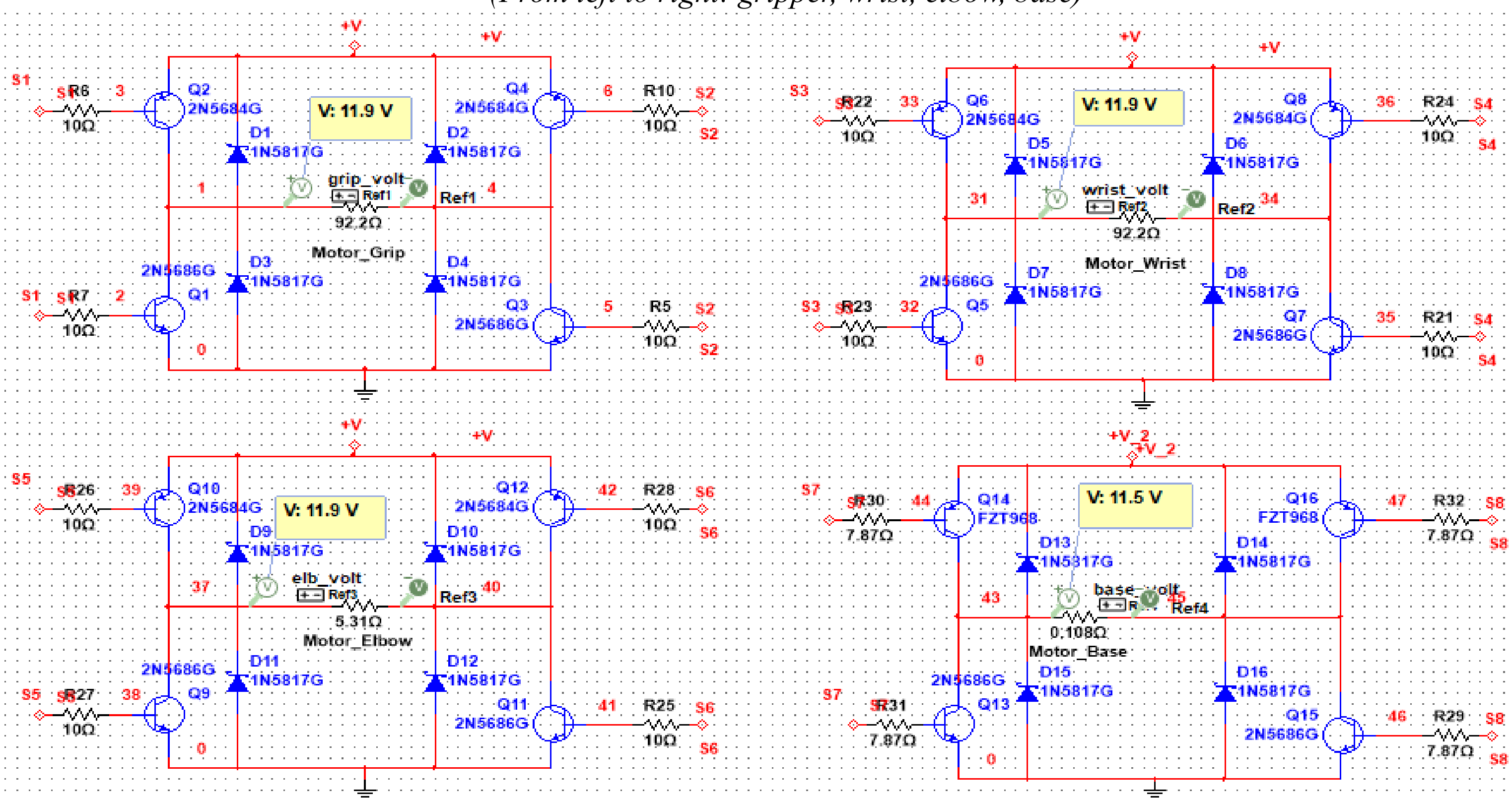


Figure 2.2. SIA of Voltage Delivered to Motors from 5V Arduino Step Input
(From left to right on top then left to right on bottom: gripper, wrist, elbow, base)

To ensure safety of the Arduino, current draw from the Arduino was measured to be below rated current 40mA. In figure 2.1, the current draw, I_out were all measured to be below 40mA at around 11.1pA. [3]

Additional steps taken in ensuring the safety of the Arduino involves integrating a 1N5817G schottky diode to prevent backwards current flowing into the Arduino pins. (This can be found in figure 1.1)

3. Modeling Circuit TRP in MATLAB

During the SIA, a TRP of 1*10^-8 sec sample time is produced and used to develop a linear MATLAB model. Attached here are the TRP of each motor.

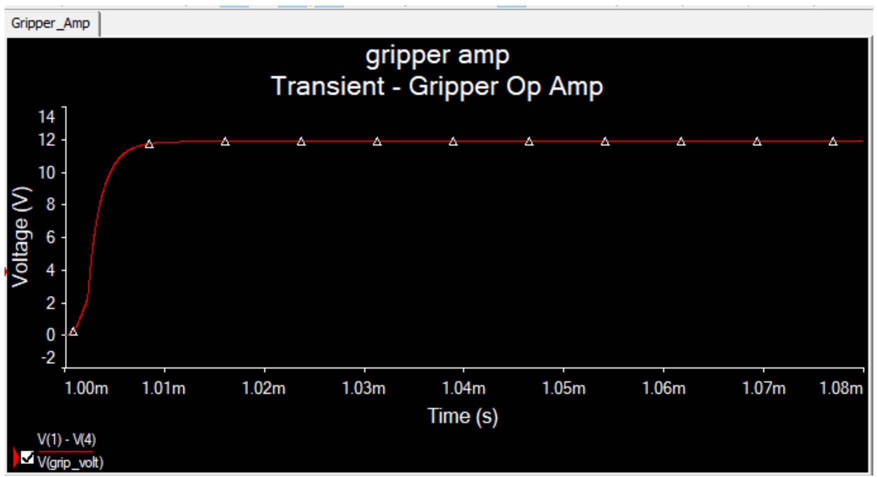


Figure 3.1. TRP of Gripper Motor from 5V SIA

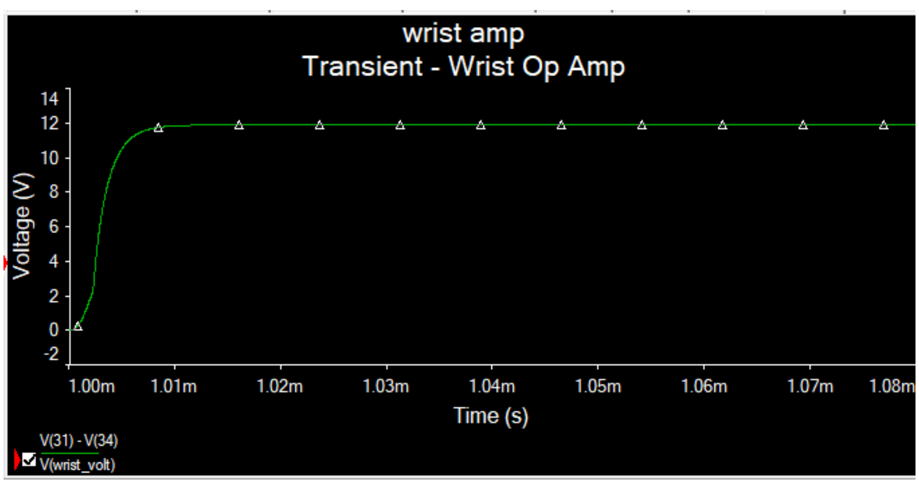


Figure 3.2. TRP of Wrist Motor from 5V SIA

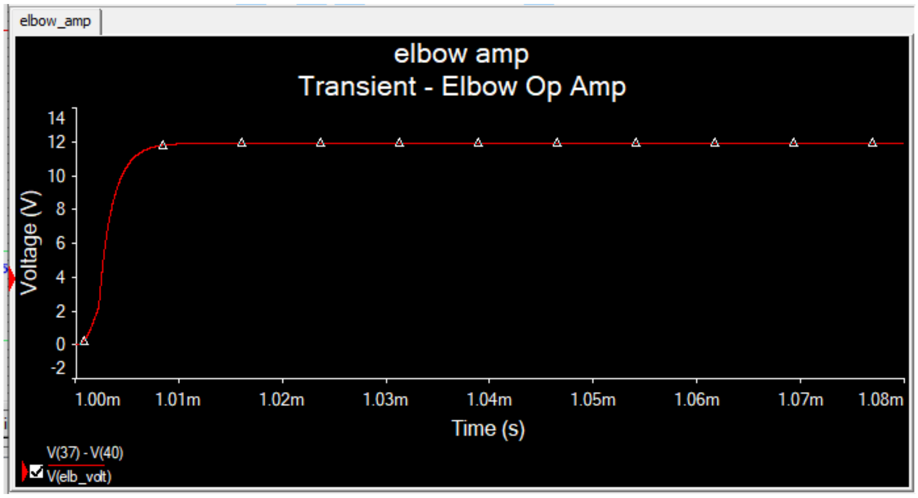


Figure 3.3. TRP of Elbow Motor from 5V SIA

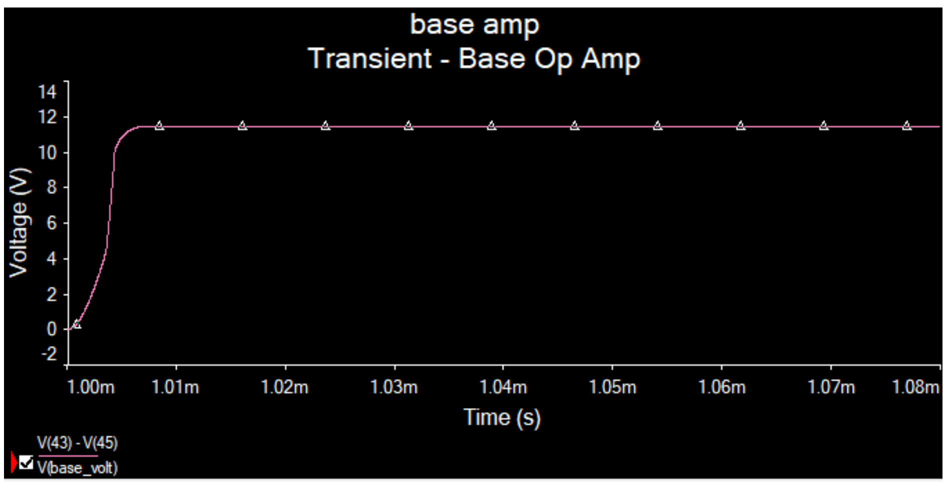
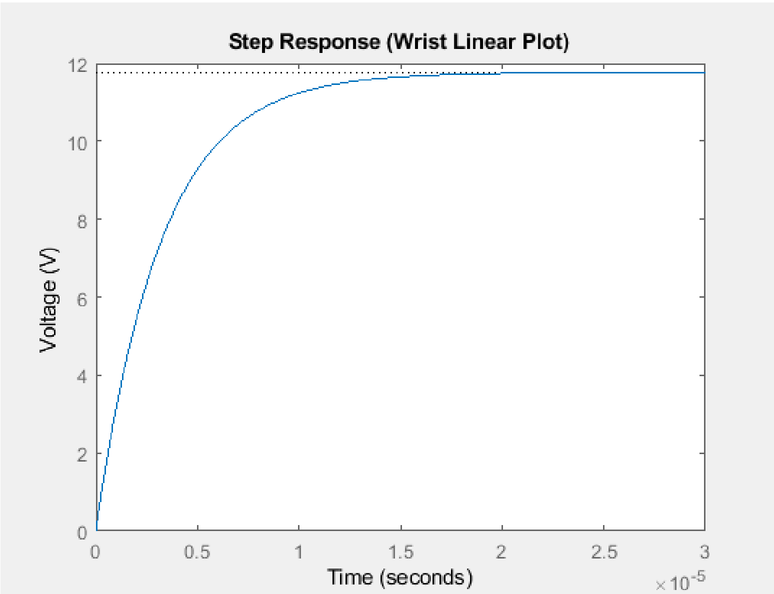


Figure 3.4 TRP of Base Motor from 5V SIA

The code used for linearizing the model in MATLAB is attached in appendix A. Below, images of the step response output obtained through each linear model are attached.



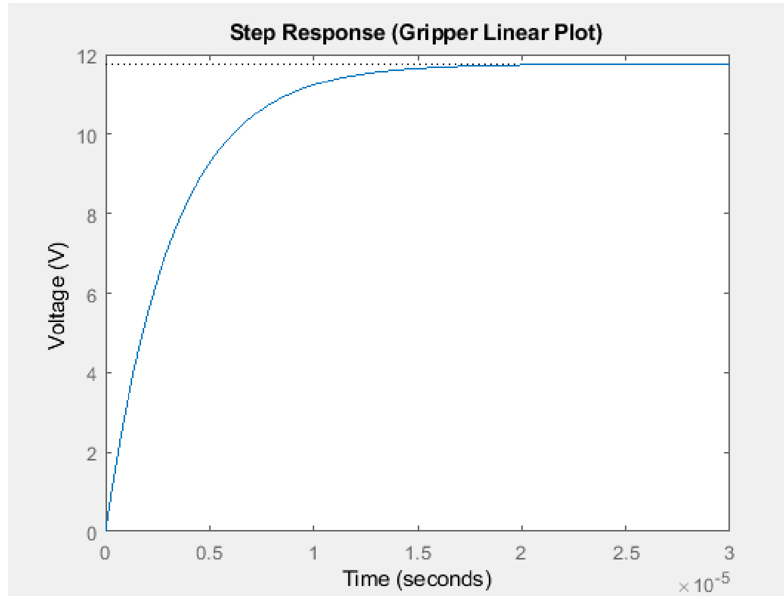
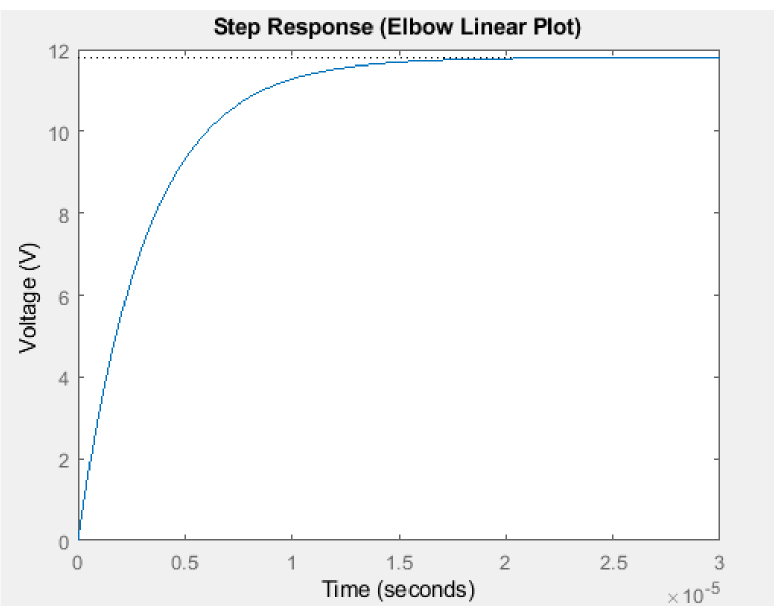


Figure 3.5., 3.6. - MATLAB TFEST Linearized Plots of the TRPs Above (Figure. 3.1, 3.2)



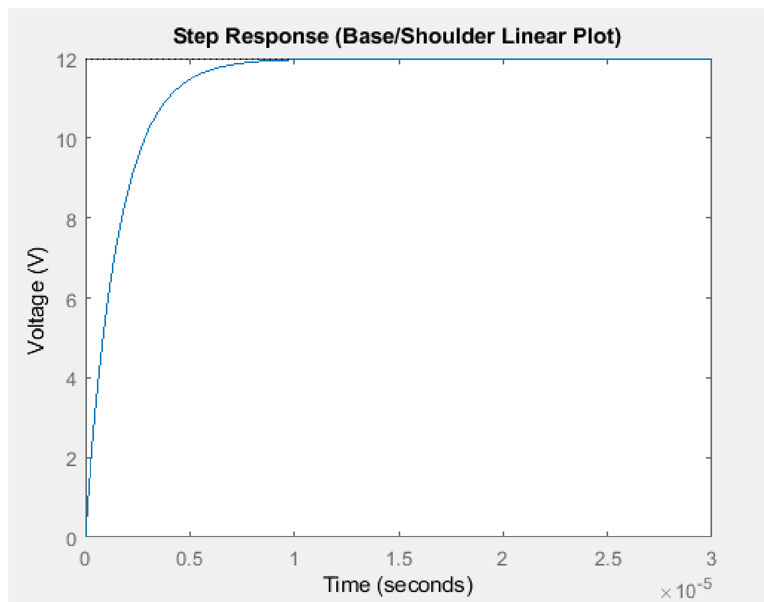


Figure 3.7., 3.8. - MATLAB TFEST Linearized Plots of the TRPs Above (Figure. 3.3, 3.4)

```
sys_grip =

  From input to output "y1":
    7.311e05
  ------------
  s + 3.107e05

Continuous-time transfer function.
```

```
sys_wrist =

  From input to output "y1":
    7.311e05
  ------------
  s + 3.107e05

Continuous-time transfer function.
```

Figure 3.9., 3.10. - MATLAB TFEST Linearized Equations of the TRPs Above (Figure. 3.1, 3.2)

```
sys_elbow =

  From input to output "y1":
    7.371e05
  ------------
  s + 3.124e05

Continuous-time transfer function.
```

```
sys_base =

  From input to output "y1":
    1.509e06
  ------------
  s + 6.291e05

Continuous-time transfer function.
```

Figure 3.11., 3.12. - MATLAB TFEST Linearized Equations of the TRPs Above (Figure. 3.3, 3.4)

4. PCB Layout Design

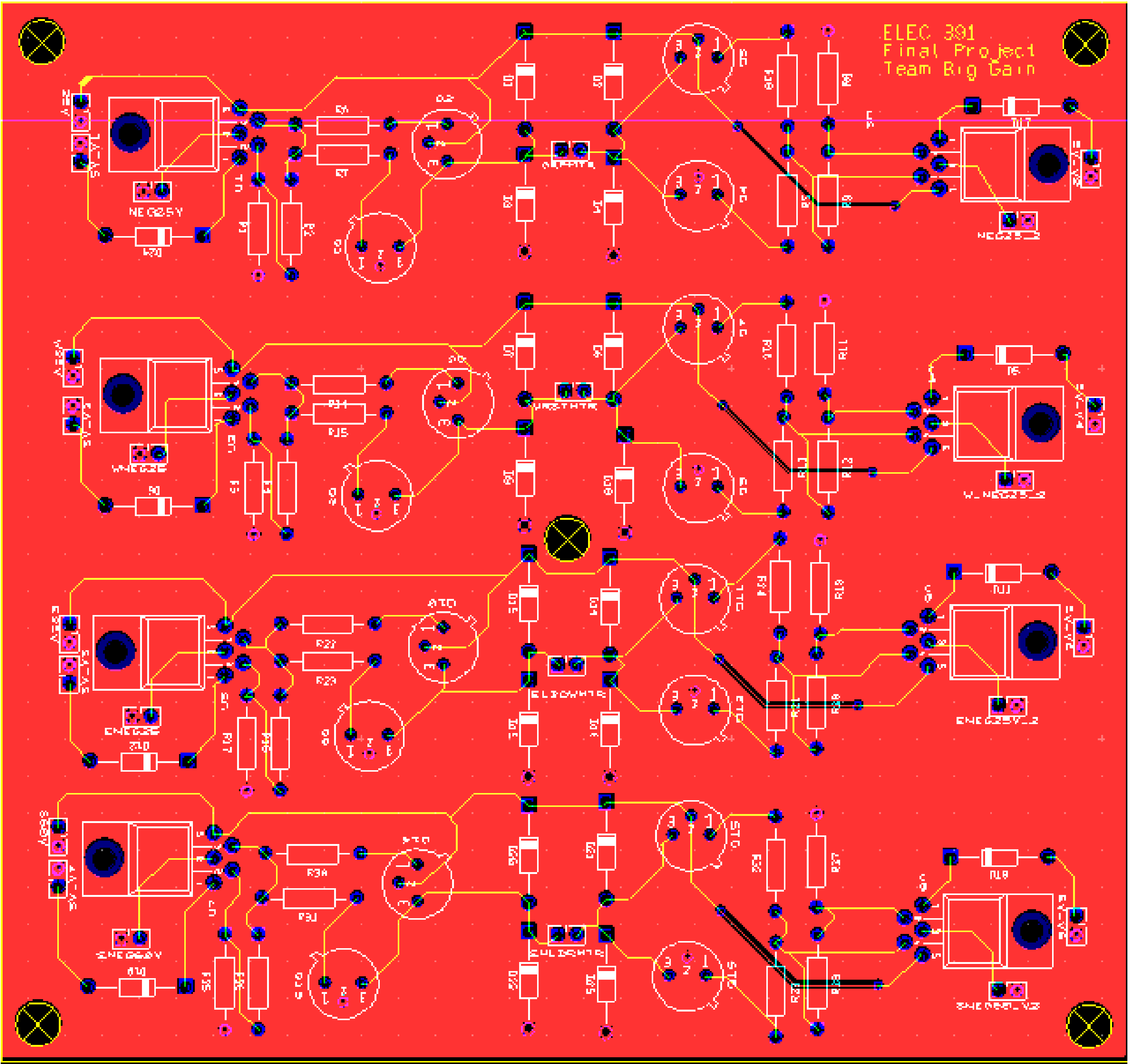


Figure 4.1. PCB of 4 H-bridge Motor Drivers with both Copper areas and Ground Plane on Bottom Layer

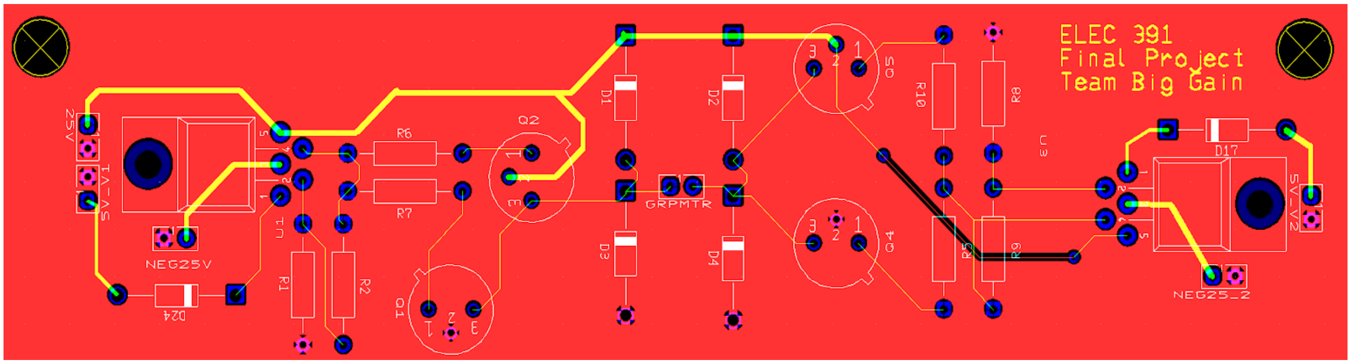


Figure 4.2. Zoomed in PCB for the gripper (top) circuit for better trace width and component clarity

The PCB uses a ground plane to provide a common ground for the whole board which prevents ground loops and prevents electrical noise. In Figure 4.1 and 4.2, the pins highlighted in magenta are connected to Net 0 hence the ground plane.

The Power lines (traces from the Arduino Pin, power supply to the Op-amps and to the H-bridge driver) are 25 mil thick because they typically route a higher current and the other traces are 15 mil thick for consistency. There are two copper layers in the PCB and the bottom layer is assigned the ground plane. A minimum number of vias (one) was used to connect two components and it was ensured that the via running through the bottom layer (black trace in Figure 4.2) was as small as possible because it runs through the ground plane. The dimensions of the PCB are 12.2 × 8.1 cm which is a total area of 98.8 cm^2 . This is less than the constraint of 10 × 10 cm = 100 cm^2 . M3 Drill sizes were used for the mounts for the PCB. It is as small in dimension as can be without compromising resemblance to the actual Multisim Circuit Schematic for easier debugging. The PCB uses stock components that can easily be obtained from DigiKey or a similar source. All the connectors are labelled descriptively (GRPMTR = Gripper Motor Connector, 5V pins for the arduino and appropriate power source connector labelling).

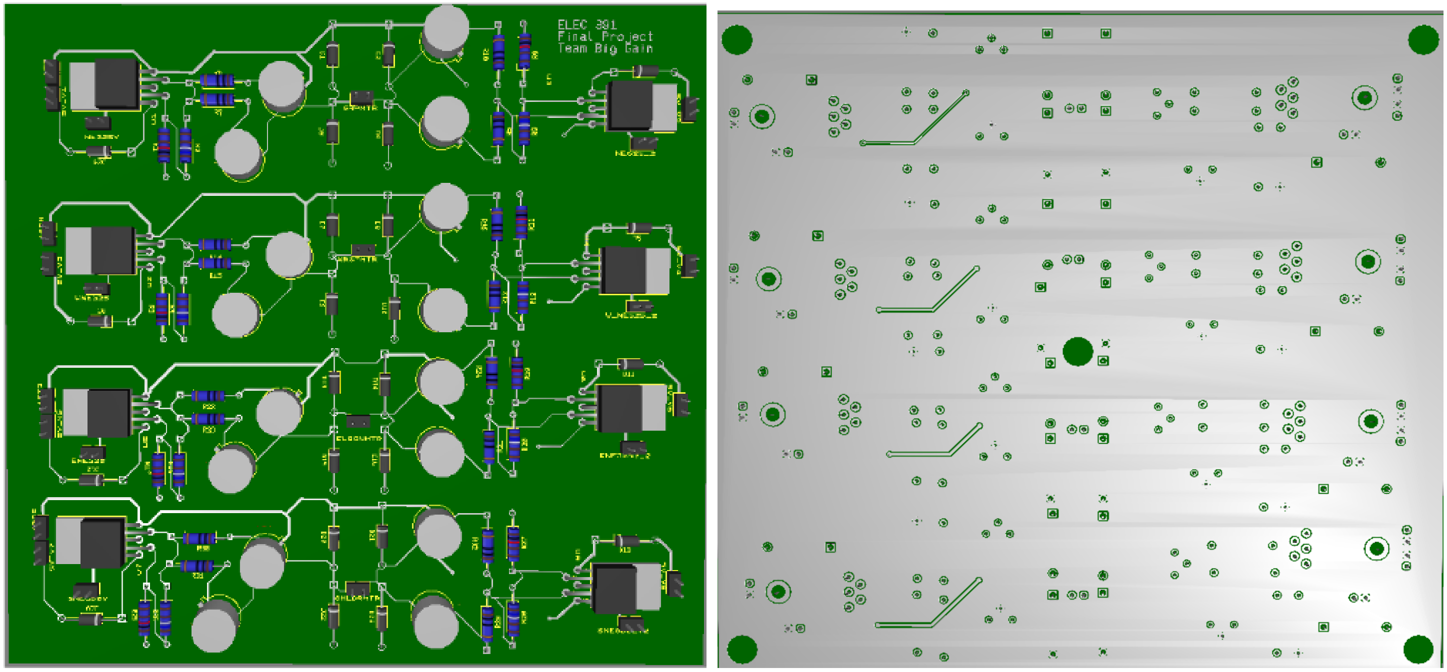


Figure 4.3. 3D view of the top and Bottom Layer

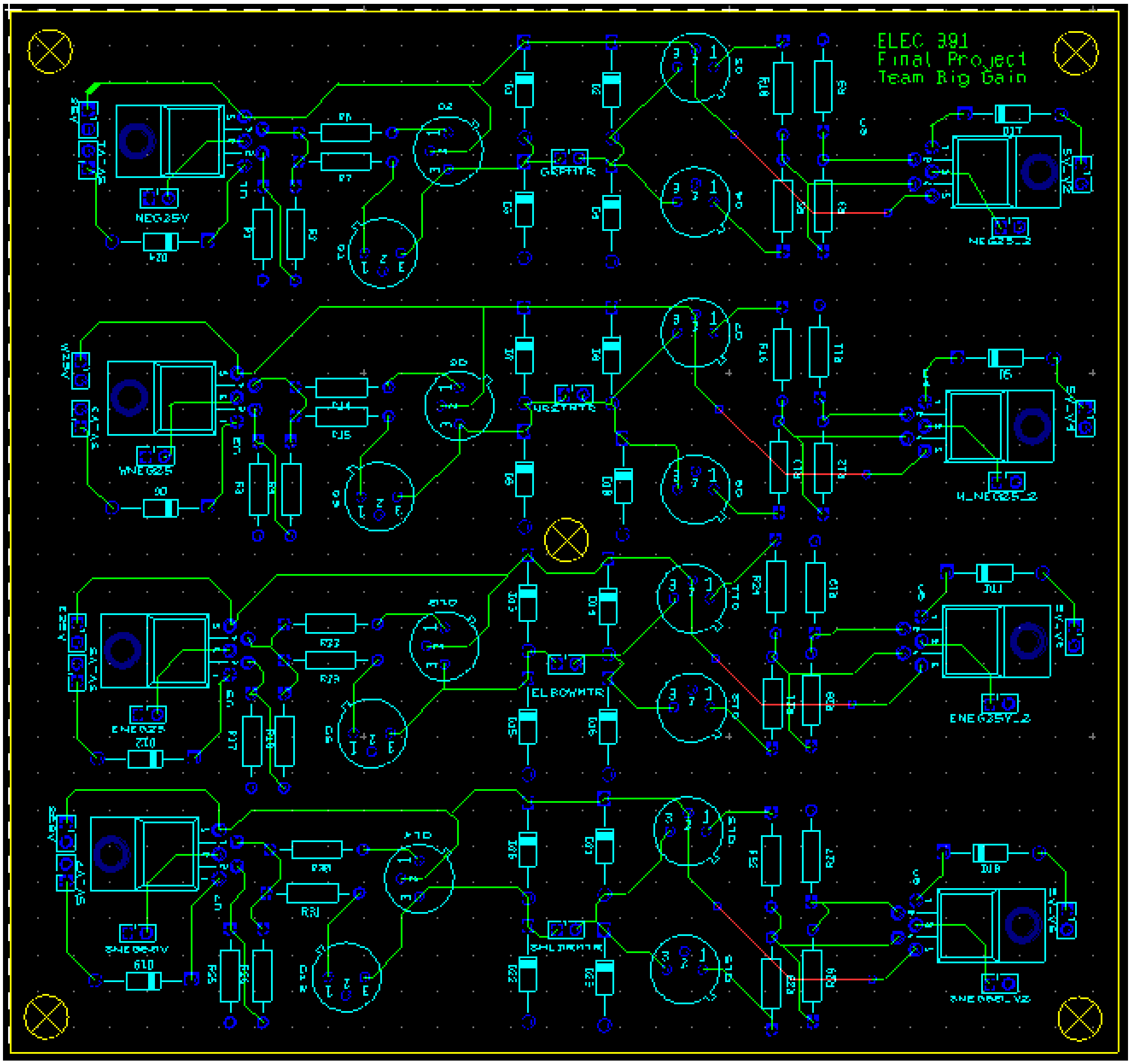


Figure 4.4. PCB layout with copper areas hidden

After finalizing the PCB design, a cost calculation was conducted using the finalized parts. The PCB area is calculated for estimating cost for printing PCB. Its size was measured in Solidworks to be 154.69 mm by 145.55 mm, which is equal to 34.9 square inches. Attached below is a spreadsheet calculator of the total cost estimate involved with producing this PCB. This PCB printing pricing was taken from ELEC 391 quota and cost for students page. [4]

Parts Name	Parts Unique ID	# used	Pricing (CAD)	$ for # parts used (CAD)
Diode	1N5817G	24	$0.511 for 1 $0.403 for 10	$10.10
PNP BJTs (small)	2N5684G	6	$4.04 for 1	$24.24
PNP BJTs (big)	FZT968	2	$1.43000 for 1	$2.86
NPN BJTs	2N5686G	8	$24.14 for 1	$193.12
Op amp	OPA544T	8	$7.40 for 1	$59.20
resistors	misc	48	~$0.30 for 1	$9.60
Printing Service	misc	1	$25 + $10/sq-in	$373.98
M3 10mm screw	91292A113	9	$5.19 for 100	$0.47
3D printed housing	misc	1	negligible	negligible
			Total:	**$673.57**

Table 4.1. PCB Production Cost Calculation

To see the price listing for each individual part, please see reference [5] through [11]. The encoders and motor cost are considered in the mechanical report and not considered in Table 4.1.

The PCB CAD file was exported as a DXF file, and extruded to 1mm to emulate the real PCB.

As seen in figure 4.5 there are 5 M3 screw holes for mounting the PCB to the housing, and there are 4 M3 screw holes for mounting the housing to the base plate.

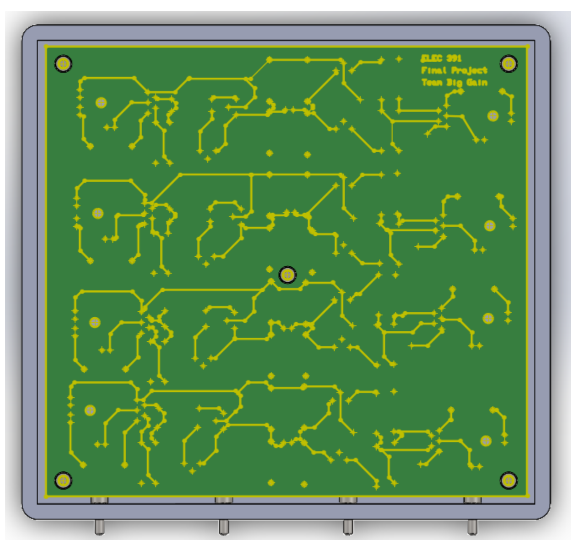


Figure 4.5. PCB housing, side view

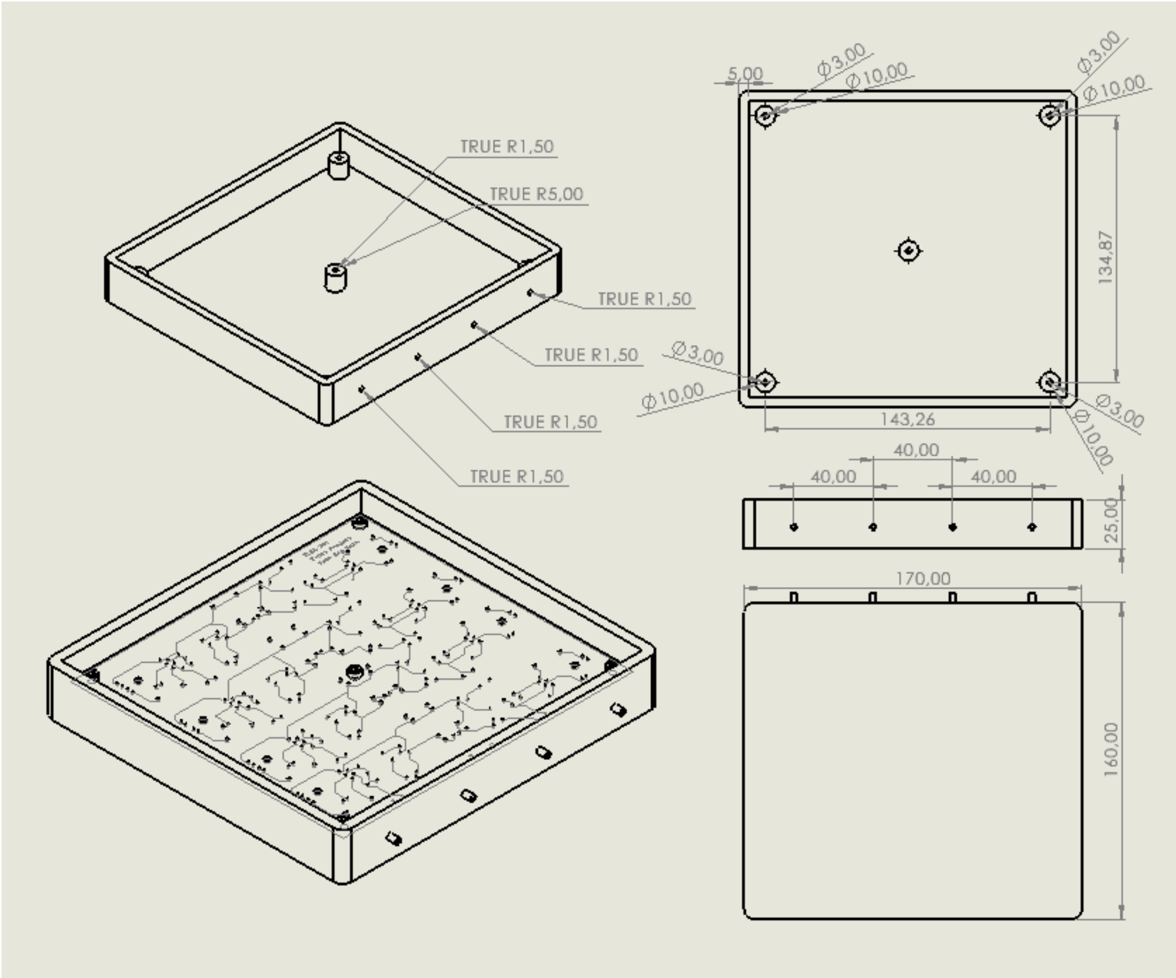


Figure 4.6. PCB housing drawing specifications.

References

[1] Maxon Motor Catalog, 2021, http://ece.ubc.ca/~leos/pdf/datasheets/Maxon/MaxonCat.pdf

[2] OPA544T Datasheets, September 1995, https://www.ti.com/lit/ds/symlink/opa544.pdf?ts=1618421514052&ref_url=https%253A%252F%252Fwww.ti.com%252Fstore%252Fti%252Fen%252Fp%252Fproduct%252F%253Fp%253DOPA544T%2526utm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dasc-null-null-OPN_EN-cpc-store-google-wwe%2526utm_content%253DDevice%2526ds_k%253DOPA544T%2526DCM%253Dyes%2526gclid%253DCj0KCQjwgtWDBhDZARIsADEKwgNl0fVW4d5PRYKlYxOxoQVqlG3C8ajWFLqeeWsjZVZmAlQX0SNN_AkaAix8EALw_wcB%2526gclsrc%253Daw.ds

[3] Arduino Leonardo Overview, 2021, https://www.arduino.cc/en/Main/Arduino_BoardLeonardo

[4] ELEC 391 Quota and Cost, accessed April 2021, https://eng-services.ece.ubc.ca/elec391-quota-and-costs/

[5] ON Semiconductor 1N5817G, accessed April 2021, https://www.mouser.ca/ProductDetail/ON-Semiconductor/1N5817G?qs=y2kkmE52mdMQJPppHZ2BuQ%3D%3D

[6] ON Semiconductor 2N5684G, accessed April 2021, https://www.infinity-component.hk/product/AMI-Semiconductor-ON-Semiconductor_2N5684G.aspx

[7] Diodes Incorporated FZT968TA, accessed April 2021, https://www.digikey.ca/en/products/detail/diodes-incorporated/FZT968TA/204155

[8] ON Semiconductor 2N5686G, accessed April 2021, https://www.mouser.ca/ProductDetail/ON-Semiconductor/2N5686G?qs=vLkC5FC1VN9aHVbAHQ6HAQ%3D%3D&gclid=Cj0KCQjwgtWDBhDZARIsADEKwgPJssamkIuP1gCgOt2hrJkwcfZTUguttFqcKOKKVbvpQ_d36r1HycIaArf9EALw_wcB

[9] OPA544T, accessed April 2021, https://www.infinity-component.hk/product/Luminary-Micro-Texas-Instruments_OPA544T.aspx

[10] Digikey Chip Resistor - Surface Mount Catalog, accessed April 2021, https://www.digikey.ca/en/products/filter/chip-resistor-surface-mount/52

[11] McMaster-Carr 91292A113, accessed April 2021, https://www.mcmaster.com/catalog/127/3366

APPENDIX A1 - LINEAR MODEL MATLAB CODE (GRIPPER)

```
% load your data (text file with your two columns)
load gripAmp.txt;

% sample index, reducing of input to get single step instead of square wave
x = gripAmp(:,1);

%{
% plot data
figure(1)
plot(x,gripAmp(:,2)); hold off
%}

%timing interval declare
Ts = 0.00000001025;

% prepare data for tftest, TS is a random chosen sampling time
data = iddata([zeros(3,1);gripAmp(:,2)],[zeros(3,1);x],Ts);

% estimate system, factor 2 -> number of poles (variable as desired)
sys_grip = tfest(data,1,0);

% Normalization - Multiply by sample rate
sys_grip = sys_grip*(Ts*2*10000);

%{
% plot step response (factor 5 comes from input - 5V)
figure(2)
step(5*sys_grip)
%}
```

APPENDIX A2 - LINEAR MODEL MATLAB CODE (WRIST)

```
% load your data (text file with your two columns)
load wristAmp.txt;

% sample index, reducing of input to get single step instead of square wave
x = wristAmp(:,1);

%{
% plot data
figure(1)
plot(x,wristAmp(:,2)); hold off
%}

%timing interval declare
Ts = 0.00000001025;

% prepare data for tftest, TS is a random chosen sampling time
data = iddata([zeros(3,1);wristAmp(:,2)],[zeros(3,1);x],Ts);

% estimate system, factor 2 -> number of poles (variable as desired)
sys_wrist = tfest(data,1,0);

% Normalization - Multiply by sample rate
sys_wrist = sys_wrist*(Ts*2*10000);

%{
% plot step response (factor 5 comes from input - 5V)
figure(2)
step(5*sys_wrist)
%}
```

APPENDIX A3 - LINEAR MODEL MATLAB CODE (ELBOW)

```
% load your data (text file with your two columns)
load elbowAmp.txt;

% sample index, reducing of input to get single step instead of square wave
x = elbowAmp(:,1);

%{
% plot data
figure(1)
plot(x,elbowAmp(:,2)); hold off
%}

%timing interval declare
Ts = 0.00000001025;

% prepare data for tftest, TS is a random chosen sampling time
data = iddata([zeros(3,1);elbowAmp(:,2)],[zeros(3,1);x],Ts);

% estimate system, factor 2 -> number of poles (variable as desired)
sys_elbow = tfest(data,1,0);

% Normalization - Multiply by sample rate
sys_elbow = sys_elbow*(Ts*2*10000);

%{
% plot step response (factor 5 comes from input - 5V)
figure(2)
step(5*sys_elbow)
%}
```

APPENDIX A4 - LINEAR MODEL MATLAB CODE (BASE)

```
% load your data (text file with your two columns)
load baseAmp.txt;

% sample index, reducing of input to get single step instead of square wave
x = baseAmp(:,1);

%{
% plot data
figure(1)
plot(x,baseAmp(:,2)); hold off
%}

%timing interval declare
Ts = 0.000000005125;

% prepare data for tftest, TS is a random chosen sampling time
data = iddata([zeros(3,1);baseAmp(:,2)],[zeros(3,1);x],Ts);

% estimate system, factor 2 -> number of poles (variable as desired)
sys_base = tfest(data,1,0);

% Normalization - Multiply by sample rate
sys_base = sys_base*(Ts*2*100000/5);

%{
% plot step response (factor 5 comes from input - 5V)
figure(2)
step(5*sys_base)
%}
```