Characterization and Optimization of Preamps for the Mu2e Tracker

Perianne Johnson

SIST Intern, Fermi National Accelerator Laboratory, Particle Physics Division

New Mexico Institute of Mining and Technology

August 4th 2014

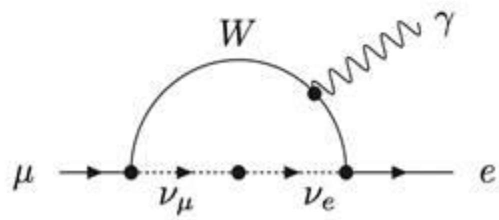
Abstract

Mu2e is a particle physics experiment hoping to observe charged lepton flavor violation currently in the design phase at Fermi National Accelerator Laboratory. The experiment utilizes a straw tracker, which uses small amplification devices known as preamps. The preamp is still being designed, with changes being made to increase gain, optimize the shape of the response, and minimize noise. This paper focuses on one round of modifications. At the end of this study, the current design of the preamp was exhibiting a gain of 52 dB at its peak of 100 MHz, while operating at an RMS noise level of 10 mV.

Introduction

The phenomenon of Charged Lepton Flavor Violation, in which electrons, muons, and tau particles can readily transform into one another, is not explicitly forbidden in the Standard Model, though it has never been observed, due to the low probability of the necessary interactions, of the order 10^{-50}. One such interaction consists of a muon changing flavor into an electron with no neutrino production. This rare event is the subject of study for the Mu2e experiment currently being designed at Fermi National Accelerator Laboratory.

Figure 1:



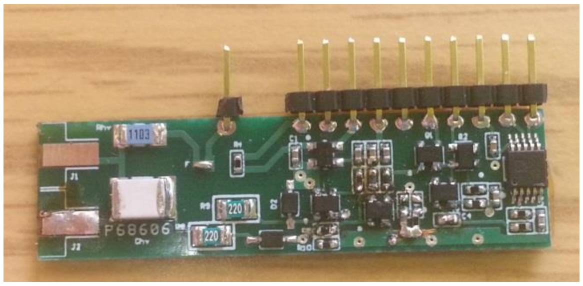
Feynman Diagram depicting a muon changing flavor into an electron. Credit: http://mu2e.phy.duke.edu/

Mu2e utilizes a straw tracking system to determine the path of particles produced. The straws are composed of a long thin cylindrical Mylar shell (5 mm diameter) with a concentric wire running through it. Many straws are placed together in a vacuum chamber, and the straws themselves are filled with $ArCO_2$ gas, which will become ionized when a particle passes through. A potential difference is established between the cylindrical shell and the wire, such that freed electrons move towards the wire and the ions are drawn towards the shell. This produces a detectable signal, signifying the particle was at that straw's location at that given time. Knowing where on the length of the straw the hit occurred is also important for determining the particle's path. This is achieved by having an amplification device called a preamp, at both ends of the straw. Since the current will move through the wire in each direction at the same speed, the time difference between when each end's preamp registers the hit can be used to determine the location of incidence.

Development of the preamp is ongoing, seeking to maximize gain and stability while minimizing noise and cross talk. Cross talk is an unavoidable fact of having the preamps mounted so close together on the motherboard; signal from one preamp is inevitably transferred in part to its neighbors.
Boosting the gain at key frequencies, known as shaping, is also an important goal. This optimization is achieved in general by making modifications to the current version of the preamp, sending the preamp a signal in the form of a sine wave, and characterizing the response, both quantitatively and qualitatively. Some example modifications include the amount of shaping, the values of certain electrical components, and the type of transistor used. This paper focuses on several versions of the preamp and the testing performed on them to optimize the gain and noise.

Apparatus

Figure 2:

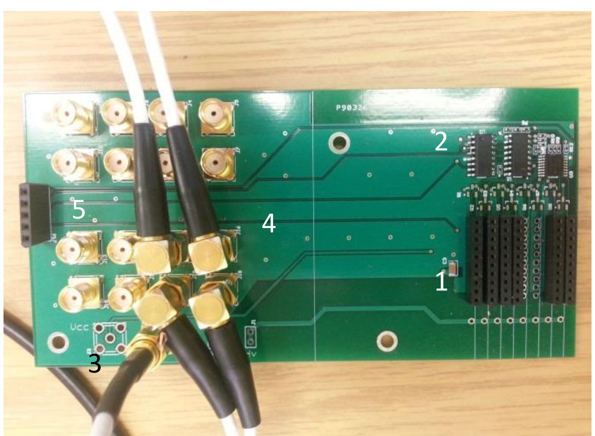


Front view of a preamp. Dimensions 4.0 cm by 1.1 cm

A preamp, or pre-amplification device, is a small printed circuit board that amplifies an input signal. Many preamps are mounted together onto a motherboard, which provides the preamps with power, programs them via a 1-Wire chip [1], as well as allowing their amplified signal to be seen on an oscilloscope. Each preamp has an inherent gain, measured in decibels (dB), which is a measure of the amplification factor related to the ratio of output and input signal amplitudes.

$$Gain_{dB} = 20\log_{10}\left(\frac{A_{out}}{A_{in}}\right) \qquad \text{(eq. 1)}$$

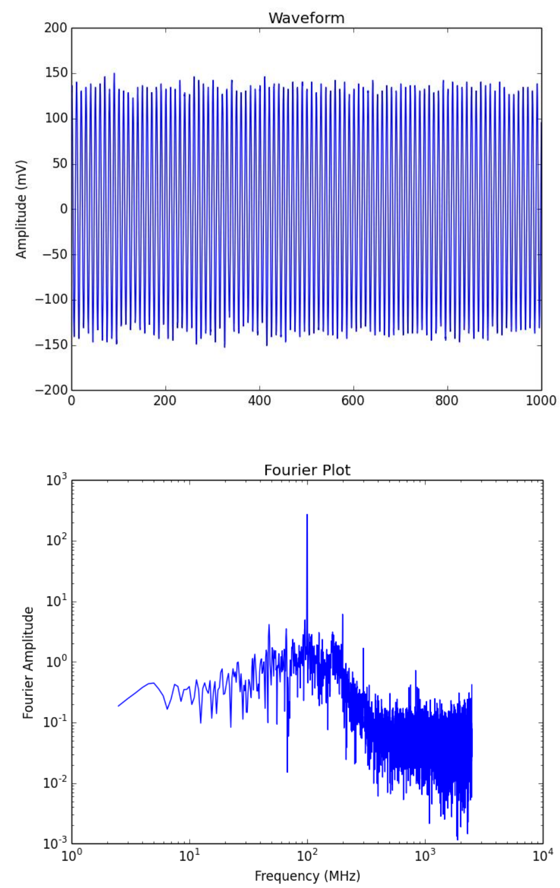
Figure 3:



Motherboard, showing eight preamp channels (1), 1-Wire Chips (2), power cable (3), four scope cables (4), and Arduino connector (5)

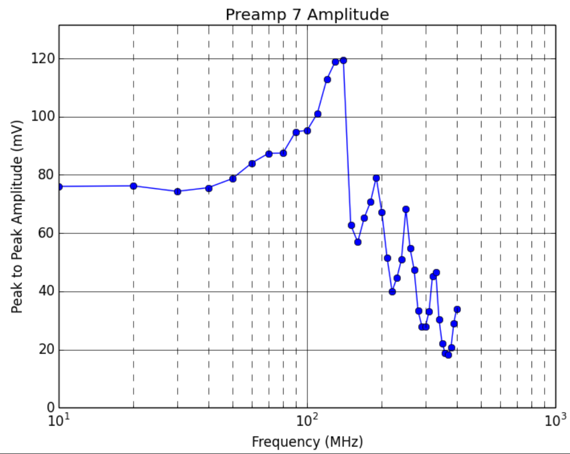
The motherboard is controlled using an Arduino Due microcontroller [2]. The Arduino provides information on which preamp to program, what to program it with, and also reports back if any preamps are unresponsive. The key piece of information shared by the Arduino is known as a balance number, so called because it controls how much current is on each leg of the output, which need to be balanced. The total current supplying the two legs is constant, and a transistor controls the amount of current flowing through each leg, depending on the voltages at points 1 and 2 on the preamp schematic (shown in Appendix A1). The balance number determines the voltage at point 1, chosen by the user such that both legs receive equal current when there is no signal. If the voltage at point 2 becomes higher than at

Figure 4:



Example plots for 100 MHz input signal.

Figure 5:



Output signal amplitude vs. frequency for Preamp 7

point 1, due to a signal or noise, more current will be directed to one of the legs. The difference between the two legs now shows a detectable, nonzero signal. The balance numbers, unique to each preamp are fed to the Arduino by a Python script (“balance_varies.py”, A2), and subtly changed until both output legs have the same current. The total current level for the preamps can also be adjusted in Python and is communicated to the preamps via the same Arduino program. The balance and current numbers are stored in a text file, which Python reads in and transmits to the Arduino when prompted by the user. The text file is easily modified by the user, and “balance_varies.py” can be run continuously in a background python command window, allowing the user to update the balance or current whenever appropriate. Python was also used to save the oscilloscope data to be used for offline analysis. The script “read.py” (AII) captures the output waveform from the scope and saves the data as a text file, which can be re-plotted or otherwise manipulated later on. This script displays the waveform plot as well as the Fourier transform of the wave. Examples are shown in Figure 4.

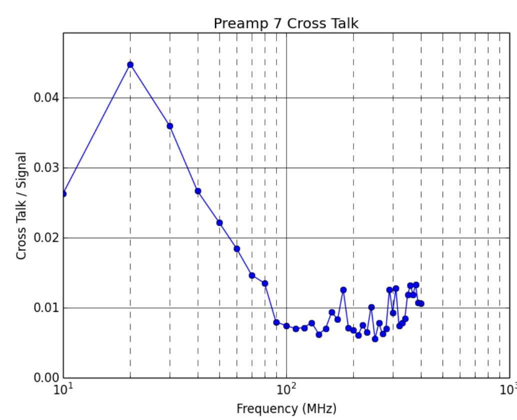
Methods and Results

Original Preamp

This study began by analyzing the current version of the preamp, looking at the gain and cross talk levels. Two preamps were mounted on the motherboard, in channels 6 and 7, and a 27 mV peak to peak amplitude sine wave signal was supplied from a signal generator. This signal was attenuated by 18 db, resulting in an amplitude of 2.5 mV. Their gain was analyzed for input signals in the 10 to 400 MHz range. Their cross talk was also measured in this range.

Figure 5 shows the amplitude versus frequency plot for the preamp in channel 7. This was produced by using the python program “read.py”. A computer is connected to the oscilloscope via a serial port, which allows the program to capture the output wave. This wave is then Fourier transformed, producing an amplitude versus frequency plot. The maximum amplitude occurs at the current frequency of the signal generator, with the other smaller amplitudes attributed to background noise. The maximum amplitude is determined for varying frequencies to produce a gain plot for each preamp, as shown in Figure 5. There is a fairly constant output amplitude of roughly 80 mV up

Figure 6:



Cross talk for Preamp 7

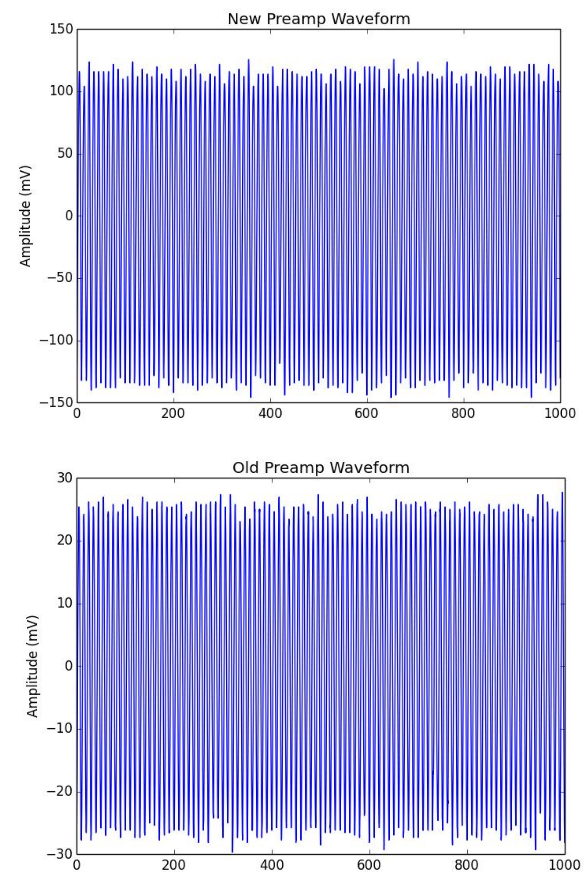
until roughly 140 MHz, after which the amplitudes drop off steeply, with small undulations attributed to various resonances of the preamp. At 100 MHz, where much of the signal is expected for the experiment, the amplitude is 95 mV, with a noise level of 0.6 RMS. The noise level was determined by finding the RMS amplitude of the output wave when the preamp was not connected to the signal generator.

To measure the cross talk, a second preamp was mounted on the mother board, into channel 6. A signal was sent through this new preamp, but the output wave was read off of preamp 7. In a perfect world, no signal would be seen in Preamp 7, but due to how close the two preamps are mounted, some of Preamp 6's signal gets transferred over to Preamp 7. The cross talk was measured just like the gain, at varying frequencies, shown in Figure 6. The cross talk amplitude is divided by the output amplitude measured earlier in Preamp 7, producing a plot of cross talk / amplitude versus frequency. The maximum acceptable amount of cross talk for the experiment is two percent, and the setup was modified until the cross talk was reduced to about that value. Modifications mostly consisted of adding copper tape to the backs of both preamps, as well as around the signal cable on Preamp 6. The copper tape acted like a Faraday cage, trapping field lines and preventing cross talk. Unexpectedly, lower cross talk was seen at higher frequencies, a phenomenon not fully understood at this point. However, after 60 MHz, the cross talk was well below two percent of the signal, a promising result.

The gain and cross talk measurements were repeated for Preamp 6. The noise level was about the same, but the gain was about half of Preamp 7's, with about twice the cross talk at a given frequency. Due to this discrepancy, as well as the difficulty in obtaining a low enough cross talk value, a new version of the preamp was made.

<u>New Preamps</u>

Figure 7:



Top panel: output signal amplitude of 50 mV. Bottom panel: output signal amplitude of 250 mV.

There were various changes made to the components on the preamp in this new version. One of the most significant was the addition of a "shaping capacitor," which increases the gain, boosting it most dramatically in the 100 MHz range. Additionally, a new socket and pin style connector (instead of soldering the preamp directly to the motherboard as before) was used to mount the preamp to the motherboard, allowing it to be easily removed. This easy removal allows modification to be made to a single preamp, and then for immediate testing and analysis of these changes by remounting the preamp. This revolutionized the testing methods contributing to accelerated optimization of the preamp.

Since the gain was so much greater on these new preamps, the input signal had to be adjusted. The signal now consisted of a 50 mV peak to peak sine wave from the generator, attenuated by 38 dB, producing a 0.63 mV signal on the scope.

Shaping Capacitors

To shape the gain response of the preamp, an RC circuit section was added to the preamp, denoted by “3” on the schematic in Appendix AI. At high frequencies, specifically higher than $1/2\pi RC$, the capacitor acts like a short in the circuit. This very low impedance allows for higher gain. In this case, a 100 pF capacitor was used to boost the signal everywhere, but most significantly at 100 MHz. This change is seen in the top panel of Figure 7, which shows the waveform of a new preamp for an input signal of 100 MHz. The output signal amplitude is greater by a factor of five, and a clear peak was observed near 100 MHz, in contrast with the bottom panel showing an old preamp at 100 MHz. Unfortunately, the shaping made the preamp very unstable, meaning it was sensitive any touching or movement. Touching the preamp with a finger or metallic tool adds capacitance to that point in the circuit, which can be the tipping point in an already unstable circuit. Additionally, the boosting greatly increased the cross talk.

Table 1:

Preamp #	Balance #	Voltage Drop	Output Amplitude (mV)	Noise RMS
0	22800	70	21	0.35
1	22600	74	23	0.38
2	22900	65	48	0.6
3	22400	74	21	0.34
4	22300	75	21	0.36
5	23100	70	24	0.36
6	21800	73	20	0.33
7	22300	71	22	0.35

The first solution, the simple solution, was to remove the boosting capacitor. This drastically reduced the gain, but this loss was balanced by an increase in stability. For the final experiment, where fifty thousand preamps will be used, stability is key, perhaps even more so than the gain.

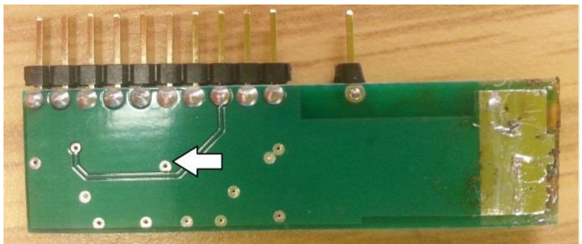
Eight preamps were made with the boosting capacitor removed, all of which were individually tested for gain and noise levels before being mounted together onto a motherboard. The individual gain and noise measurements were very consistent (Preamp 2 is a notable outlier), with a noise to signal ratio of under five percent, summarized in Table 1. Initially, the cross talk and noise values were very high when all eight preamps were mounted onto a single motherboard. This was easily mediated by covering the back of the preamp with a layer of insulating Kapton tape and then a layer of grounded copper tape, serving as a small Faraday cage. With these tapes in place, the noise to signal ratio dropped to eight percent. This is higher than the ratio for individually mounted preamps, which is an expected and unavoidable artifact of the close mounting.

While the stability issue was overcome by removing the boosting capacitor, this is not exactly an acceptable solution. While the preamps with no boosting were working with low noise and high stability, the possibility of achieving those same results with higher gain justified further testing to search for a solution that did not involve removal of the boosting capacitor.

BFP520 versus BFP720 Transistors

There are five transistors on the preamp, three of which are BFP520; these control the output stage of the preamp. The two which are originally BFP720 control the input stage. Both the 520 and 720 are NPN polarity RF transistors, but the newer 720, made of a silicon germanium material, has a faster transition frequency which allows for higher bandwidth. Unfortunately, the 720 transistors on the preamp added to the instability. By changing them to 520 transistors, we achieved the same gain, with significantly less noise and stability issues.

Figure 8:



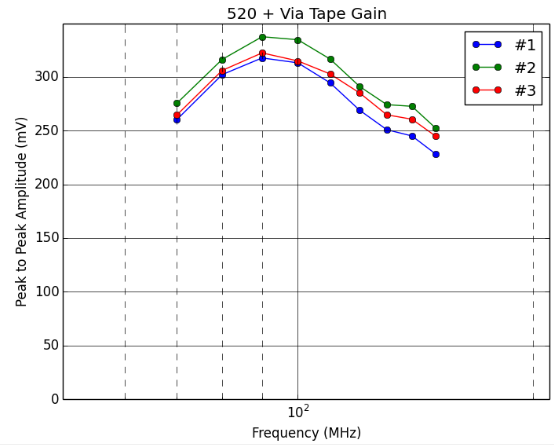
Back view of preamp: arrow denoted unstable via

Some instability remained, even after changing the transistors. The cause was tracked to a small via in the center of the preamp, shown in Figure 8. Touching this via with tweezers or other small metal tool, in other words adding capacitance, greatly reduced the noise, and increased the gain by about ten percent. Placing a small piece of copper tape over just this via on the back achieved the same results. This procedure was applied to three preamps, with all 520 transistors and the boosting capacitors. They all had consistent output signal amplitude of over 300 mV, with anoise amplitude of roughly 10 RMS. This is summarized in Figure 9.

Grounding Improvements

The small via causing the instability in the 520 preamps was originally designed to provide grounding for a resistor and capacitor, denoted R11 and C6. During testing, a better ground source was found for them in the form of a ground plane directly above them. When the green mask on the board was scraped off, it

Figure 9:

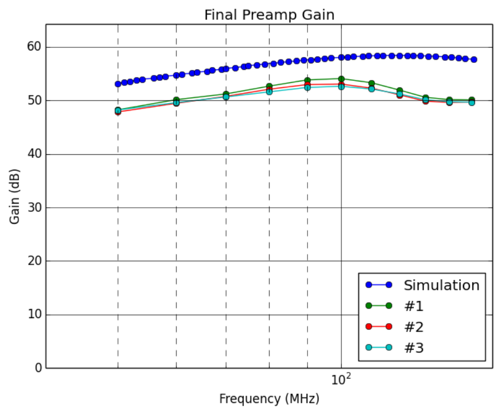


Output signal amplitude versus frequency for three preamps, with BFP520 transistors and a small piece of copper tape over via.

was possible to solder the two components from one of their designated pads onto the exposed copper. By providing this better ground source, the instability all but disappeared. For something to be well-grounded, it needs to be connected to a ground source directly, with as little impedance as possible. Since thin wires or traces and vias have small cross-sectional area, they have high impedance. This can cause small voltage variations in the supposedly constant ground source. It is best to connect one side of a component directly to ground or to use a thick trace. In this instance, by connecting R11 and C6 directly to ground instead of using a small via provided more stability.

Figure 10 shows the gain for three example preamps, with 520 transistors and the flipped resistor and capacitor. There is very good consistency across all the preamps, and a clear peak at 100 MHz as expected. The noise to signal ratio was also very low, with RMS noise levels of less than 8 mV when the preamps were mounted alone. Figure 10 also includes a SPICE simulation of the circuit's response, which matches very nicely with the experimental data.

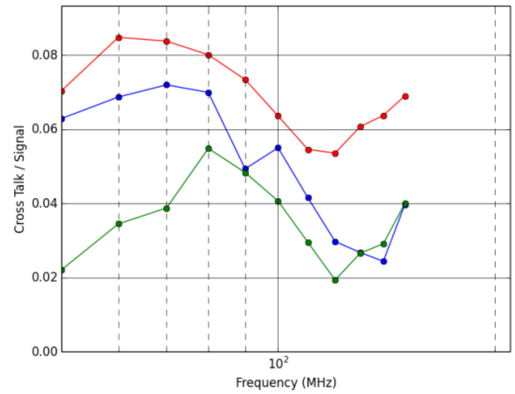
Figure 10:



Gain in dB versus frequency for three preamps, compared with simulated data (Blue)

The cross talk for three preamps is shown in Figure 11. While the three preamps had a moderately similar response, the cross talk to signal ratio was higher than the allowed two percent. At time of writing no measures had been taken to lower the cross talk.

Figure 11:



Cross talk to signal ratio for three preamps with 520 transistors and flipped components.

Conclusions

At the end of this study, the current best version of the preamp includes all 520 BFP transistors, a 100 pF boosting capacitor, and flipped resistor and capacitor. The gain of the original preamp version was 32 dB at its peak. This version had a peak gain of 52 dB, with a RMS noise level of 5 mV when mounted individually; an improvement in gain by almost a factor of two. The cross talk remains an issue, and future experimentation will have to be performed to determine a method of reducing the cross talk to signal ratio to the requisite two percent. Additionally, the layout of the preamp must be changed, so that the resistor and capacitor flipping is built in. While this is not yet the final version of the preamp ready for the experiment, the preamp in its current state is suitable for use in testing other equipment.

References

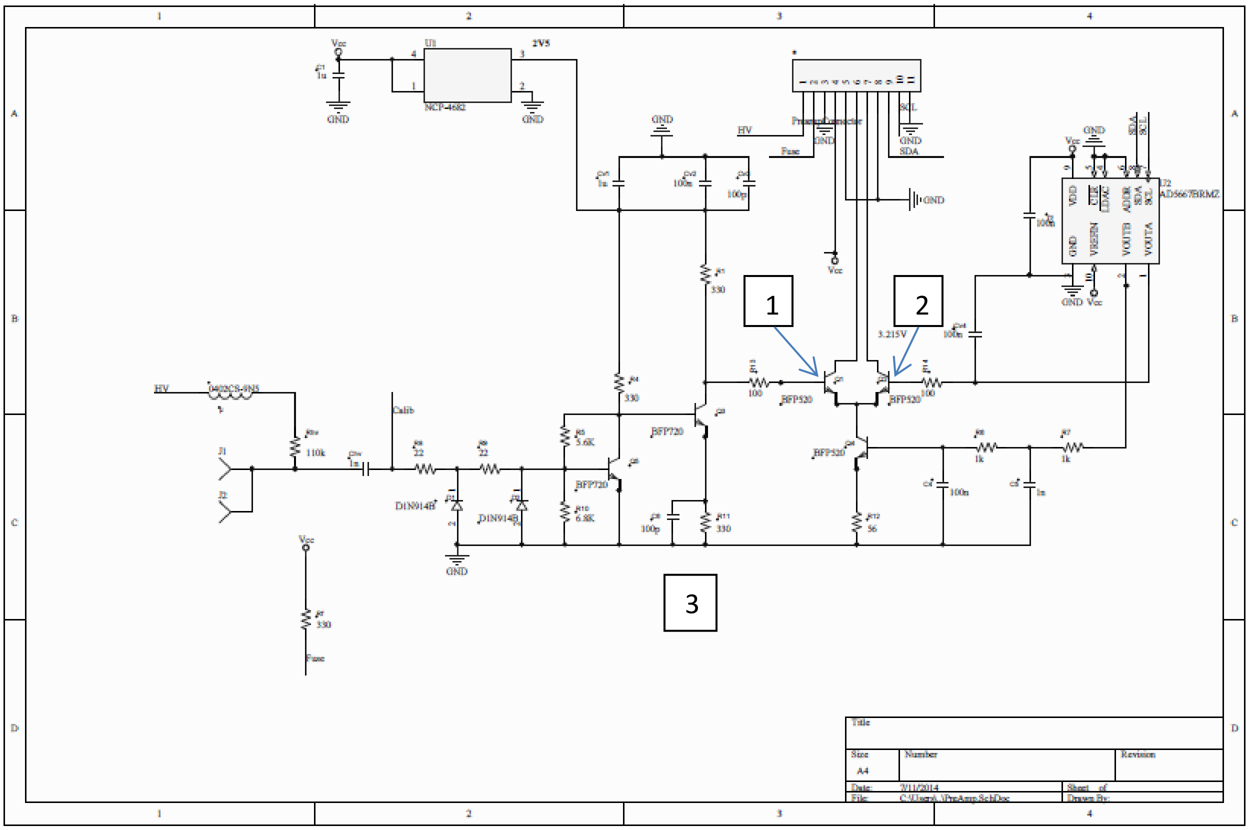
[1] DS2408: 1-Wire 8-Channel Addressable Switch
http://www.maximintegrated.com/en/products/digital/memory-products/DS2408.html
[2] Arduino Due Microcontroller
http://arduino.cc/en/Main/ArduinoBoardDue

Acknowledgments

Several parties are deserving of recognition for the success of this project and summer internship. I would first like to thank my supervisor, Dr. Vadim Rusu, for providing me with ample guidance and support throughout my project, while also allowing me to learn from my own mistakes. I would also like to thank my assistant supervisor, Dr. Aseet Mukherjee, and coworker Angela Yang for always lending her third hand. Finally, I would like to recognize Elliot McCrory, Dianne Engram, Linda Diepholz and the rest of the SIST Committee for selecting me to be a part of this program.

Appendices

A1: Preamp Schematic



A2: Python Code

“balance_varies.py”

```python
import serial
import time
import struct

def packIntegerAsULong(value):
	"""Packs a python 4 byte unsigned integer to an arduino unsigned long"""
	return struct.pack('I', value)
while 1:
	opt = input("Enter 1 to initialize (999), 2 to program: ")
	ser = serial.Serial(port="COM9",baudrate=115200, timeout = 2)
	time.sleep(1)
	a1 = []

	if opt == 3:
		#break out of loop
		ser.close()
		break
```

```
    elif opt == 1:
        #999 file
        with open('999') as f:
            for line in f:
            data = line.split()
            a1.append(int(data[0]))
        num = a1[0] #number of preamps
    elif opt == 2:
        #upload file to arduino on COM9
        with open('bal4') as f:
            for line in f:
                data = line.split()
                a1.append(int(data[0]))
                #print a1
            num = a1[0] #number of preamps
    num2 = packIntegerAsULong(num)
    ser.write(num2)
    #ser.write each line in file
    for i in a1[3:]:
        #print i
        val = packIntegerAsULong(i)
        #print binascii.hexlify(val)
        ser.write(val)
    time.sleep(.5)
    for i in a1[3:]:
        # print i
        line = ser.readline()
        print line
    #read each preamp
    if a1[3] != 999:
        for n in range(int(num)):
            #time.sleep(5)
            print ser.readline() #0 or 1 for talking to channel
            print ser.readline()
            print ser.readline() #DACA reading
            print ser.readline() #DACB reading
    elif a1[3] == 999:
        print ser.readline() #reading DACA
        print ser.readline() #DACA Data
        print ser.readline() #reading DACB
        print ser.readline() #DACB Data
        print ser.readline() #no more addresses
    ser.close()
    f.close()
```

"read.py"

```
import numpy as np
import matplotlib.pyplot as plt
```

```
import serial
import time

scope=1

ADCLSB = 100*0.0390625 #(10 mV)
N=5000
nplot=5000
frate=5000

if scope:
	ser = serial.Serial(port="COM5",baudrate=38400)
	ser.write("HEAD OFF\r\n".encode())
	ser.write("DAT:WID 1\r\n".encode())
	ser.write("DAT:ENC RPB\r\n".encode())
	ser.write("DAT:STAR 1\r\n".encode())
	ser.write("DAT:STOP 10000\r\n".encode())
	ser.write("ACQ:STOPA SEQ\r\n".encode())
	ser.write("DESE 1\r\n".encode())
	ser.write("*ESE 1\r\n".encode())
	ser.write("*SRE 32\r\n".encode())
	ser.write("*WAI\r\n".encode());
	ser.write("ACQ:STATE RUN\r\n".encode())
	ser.write("*OPC?\r\n".encode())
	ser.readline()
	ser.write("DAT:SOU CH4\r\n".encode())
	ser.write("CURV?\r\n".encode())
	seq1=[]

	for i in range (1,10009):
		c=ser.read()
		if i > 9 and i<10009 :
			seq1.append(128-ord(c))
			time.sleep(1)
else:
	seq1=[]
	with open("520_st1_100") as f:
		for line in f:
			numbers_float = map(float, line.split())
			seq1.append(numbers_float[0])
	f.close()

diff = [ADCLSB*float(m) for m in seq1]
diff=diff-np.mean(diff)
seq11 = [ADCLSB*float(i) for i in seq1]
print len(seq1)
print "RMS=",np.std(diff)
x=[0.2*float(i) for i in range(0,len(seq1))]
```

```
#PLOTTING
#wave plot
fig = plt.figure(0)
ax2 = fig.add_subplot(111)
ax2.plot(x[:N],diff[:N])
plt.title("New Preamp Waveform")
plt.ylabel("Amplitude (mV)")

#fourier plot
fig1 = plt.figure(1)
ax3=fig1.add_subplot(111)
ft0 = np.fft.fft(diff*np.hanning(len(diff))) / (len(diff)/4)
freqs = np.fft.fftfreq(len(ft0),1./float(frate))
logft0 = 2 * abs(ft0)
xft0=[i for i in freqs]
ax3.plot(xft0[5:nplot],logft0[5:nplot], label='')
ax3.get_yaxis().get_major_formatter().set_useOffset(False)
ax3.set_xscale('log')
plt.xlabel("Frequency (MHz)")
plt.ylabel("Fourier Amplitude")
plt.title("Cross Talk")
plt.title("Fourier Plot")
plt.show()

if scope:
	ser.close()
	f=open('outtest','w')
	for i in range(0,len(seq1)):
		buf = "%7.4f\n" % (seq1[i])
		f.write(buf)
	f.close()
```