Mauran Pathmakanthan & Aman Ahmed

I. INTRODUCTION

In order to walk, run, or perform any kind of motor activities, the human body relies on its muscles.. When one desires to move a limb, neural inputs from the brain travel down to the spinal cord and synapses onto the target muscle fibers responsible for controlling the limb. The electrical impulses cause the specific muscle to contract. Muscles are attached to bones with tendons, so when the muscle contracts, the tendon pulls on the bone resulting in movement. The mechanism responsible for contracting muscles is the actional potential. Action potentials can be analyzed by attaching electrodes on the surface of the skin above a target muscle which is best known as an electromyogram. Electromyography, also known as EMG, is used to measure these signals and convert them into physical diagrams. EMGs are heavily used to generate analyzable signals to measure the health of a patient's skeletal and muscular system. The use of EMG has resulted in a better understanding of muscle properties, provided information about neuromuscular disorders, and provided insights on how different muscles work together to coordinate activities. However, the signals produced may sometimes contain unnecessary noise or unwanted parts of the signal which may also be collected. To mend this issue, multiple signal processing softwares is used in hand with microcontrollers. The specific software being used to perform frequency analysis on an EMG signal this time will be labview along with an Arduino microcontroller board.

II. BACKGROUND

The components of an EMG include an input voltage supply of 50 uV to 30 mV, adhesive electrodes to attach to the muscles of the patient, two op amps, and a bandpass filter. The op amps used will be differential amplifiers and the bandpass filter will be combining an active high pass and active low pass filter. For the high pass filter, the amplifier used will be a non inverting op amp, and for the low pass filter it will be an inverting amplifier. Both op amps for the filter most commonly use LM741 op amps. In addition, the amplifiers used alongside the bandpass filter can vary depending on the specifications of the system. Examples of op amps that can be used include; AD620, AD822, AD8248, INA118, and the INA128. Each of these op amps have their own specifications and can be used for the EMG signal depending on what is required of that signal. The specifications and justifications for which amplifiers are used will be conducted in the following methods section. The microcontroller that will be used to store the signal data pre-processing will be the Arduino UNO board. The arduino IDE is what will be used to code the MCU and it will be programmed to read the analog signals produced by the EMG and convert it to analog signals before sending it to labVIEW through a serial connection. LabVIEW is a software used to process and complete multiple analyses of different produced signals.

III. METHODS

The circuit designed used 3 op amps, and a safety component using diodes and a low pass filter. In the original proposal, the safety component was listed as the optoisolator, however, once the actual implementation had begun, the optoisolator was not seen to be impractical. The actual optoisolator was much too big to be portable along with the circuit, and it was not cost efficient either. Thus, to adjust this, and also provide a degree of safety for the user, diodes and the low pass filter were combined to reduce high voltage shocks. Although this method does not provide the same safety as the optoisolator, due to the cost efficiency and overall portability of the circuit, this method was much simpler. It was derived from the previous labs that were conducted, along with a combination of external research.

The design of the circuit was overall not extremely complicated. The input stage of the circuit used two AD620 op amps to act as differential amplifiers for the inputs. The inputs were passed through the first AD620, and then they were amplified through the second AD620, before passing through to the output stage of the circuit. The output stage of the circuit made use of a LM741 op amp and passed the amplified inputs through the amplifier and outputted it into the microcontroller (Arduino UNO). The output of the LM741 was connected to a specific analog pin on the MCU, in this case the A0 analog pin, and the circuit was grounded through the arduino. Every amplifier had its own power supply, which was set to be +15V and -15V. This method of power supply was chosen over powering the circuit through the MCU, due to the Arduino board not containing a sufficient amount of power to power the op amps and generate a signal. Thus, the only hardware connections that were attached to the MCU were the ground and the analog input pin. On top of grounding the circuit through the Arduino, the circuit was also grounded through the function generator. Once this was done, the final step was to attach the electrodes to the input pins of the first AD620. The positive input was connected to the elbow/forearm region. The image below provides a more specific description of the placement of the positive electrode:

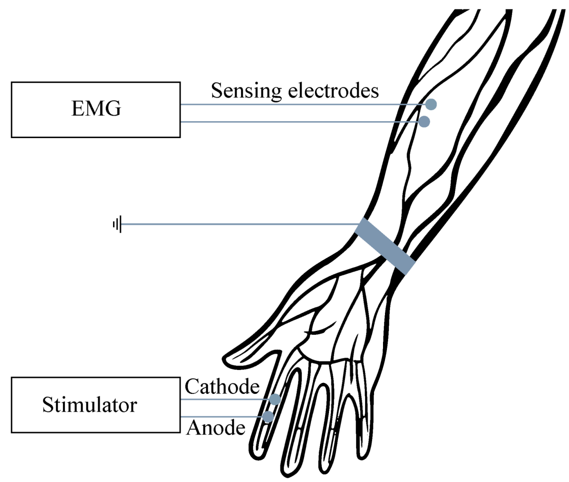

Figure 1: The figure above displays the area where the EMG's positive sensor was attached.

For this circuit, only two electrodes were used, instead of the usual 3. In a general EMG, one electrode is placed as the reference point or ground, one is placed on the area aforementioned, and the third is placed on the biceps. However, in the case of the circuit built, there was only the ground and the aforementioned electrodes that were placed. The electrodes on the biceps were excluded due to a multitude of reasons, some of which included, overall complexity of the circuit, economic and environmental efficiency, and because it was simply not necessary. Unlike the positive input, the negative input was attached to the ground electrode. The ground or reference point was placed on the bony part of the elbow as shown below.

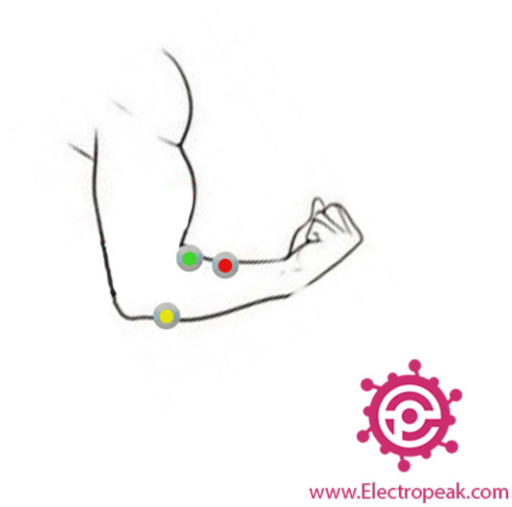

Figure 2: A display of the ground placement. The bottom green electrode was what was used as the reference point in this circuit.

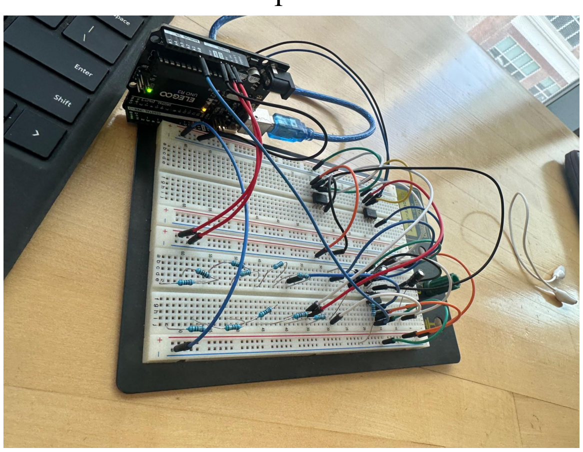

Figure 3: The figure above is a picture of the circuit built during this project. On the left side of the breadboard, the input stage can be seen with the use of the two AD620 op amps, and on the right side of the breadboard would be the output stage where the LM741 and the safety component reside.

As mentioned in the proposal and above, all components excluding the optoisolator were included. The bandpass filter was designated to be the job of the second AD620. The bandpass was used to control and maintain the noise and unnecessary frequency outputs that may have been seen without it. On the other hand the safety component that used the low pass filter and diodes as a combination, was both for an extra filter of the noisy signal and to protect users from high voltages and power leaks running through the circuit.

The specific values used for the resistors and capacitors depended on the used Gain and necessary cutoff frequency values. The expected gain as mentioned in the proposal for the op amps was 1000. To accommodate this gain, a 50.5 ohm resistance was connected to both AD620 op amps. In addition, the bandpass filter used a 1.59K ohm resistance, and the LM741's low pass filter used a 1.89K ohm resistance that was rounded to 2K ohms. As can be seen within the circuitry, to avoid excessive use of resistor components, one resistance was connected to all the components that required it. The cutoff frequency of the low pass filter that used the 2K ohm resistance and the 10uF capacitor had an expected cutoff of 7.96Hz. The cutoff frequency was decided on beforehand to be around 8Hz and the resistance was assumed to be 2K ohms. The reason for using a smaller capacitor was to allow for a higher cutoff frequency. On the other hand, if a lower cutoff frequency was required, then a higher capacitance would be used. In addition, for the bandpass filter 1.59K ohms was assumed to be the resistance and the expected frequency cutoff was 10.01Hz. Using this information the capacitance was again calculated to be 10uF. The reason a higher cutoff frequency was used was to allow for higher frequency components to pass through the filters which would in turn provide benefits to signal, such as improving the bandwidth.

Once the circuit was created the next step was to code the microcontroller to transfer the signal from the circuit to the labview software. The labview block diagram was created using the hobbyist add on, refer to appendix for diagram. The diagram was based on the one built in the tutorial provided. The read and write blocks were the main outputting blocks, along with the while loop that was created in the front panel. Labview used the diagrams and blocks to simulate a code like software that then converted those diagrams to actual code and outputted it. All parameters were set beforehand in the back panel, and the output waveform graph block was attached to the output of the front panel to acquire the signal.

The code provided within the appendix is the arduino code that was used to both transfer the signal from analog to digital and transfer it to labview, and also to calculate the resulting frequency and peak to peak values of the signals. The signals were made to be displayed in both the arduino serial plotter and the labview waveform graph. Due to this the calculation of the frequency of the signals and the amplitude of the peak to peak of the signals was attainable using simple arduino code.

IV. RESULTS

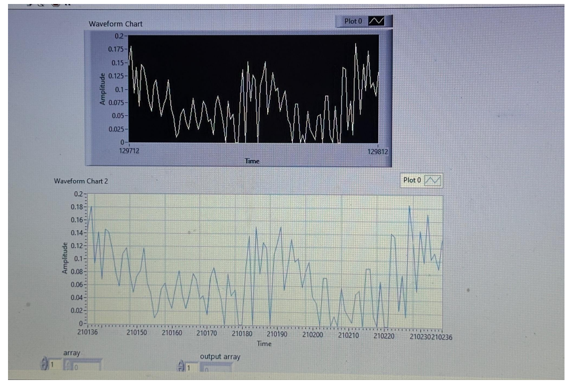

Figure 3: Plot of the subject at rest..

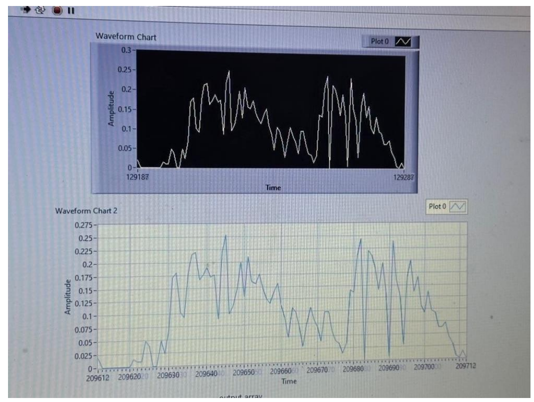

Figure 4: Plot of subject flexing.

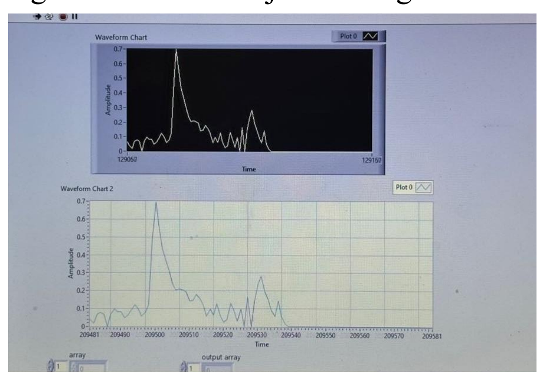

Figure 5: Plot of subject curling arm.

```
Peak to Peak when at rest = 15uV
Peak to Peak when flexing = 30uV
Peak to Peak when curling = 35uV
Frequency when at rest = 500Hz
Frequency when flexing = 500Hz
Frequency when curling = 500Hz
```

Figure 6: Frequency domain analysis

V. DISCUSSION

The circuit designed was expected to work on the first try. However, like most electrical aspects and circuits, the circuit did not work on the first try. When it was first plugged in, there was no signal being acquired. The problem seemed to be that there were connections on the circuit that were improperly connected. One of these improper connections was the connection of the 1.59K ohm resistance to the filter. There was no resistance going through the filter due to this improper connection. To fix this, the connections were redone and many components of the circuit were rebuilt to make sure all connections were correct. Another problem that was encountered was that the circuit was unable to properly protect from high voltages. When attaching the circuit to the function generator and running it, there was an incident where a spark was generated. However, this was fixed through the modification of the low pass filter and diodes that were used in the LM741. Another error that was encountered was in the arduino code itself. When attempting to convert the code from analog to digital using the /1023 method, the signal was not appearing. This resulted in multiple attempts of trial and error that led to finally deciding that it was more beneficial to not exclude that line. Excluding that line did not make any negative change, on the contrary it allowed an actual signal to form and completed the circuit and design.

When building and designing the circuit, 4 factors were considered; signal factors, environmental factors, economic factors, and medical factors. One of the first things that was considered was the signal factors. Aspects such as the expected frequency response, the expected component values, the expected frequency ranges, the accuracy and the reliability were all considered. As mentioned in the proposal, the expected frequency range was between 10-500Hz. The resulting frequency of the signal was calculated using code and the MCU to be different when the subject was at rest and when the subject was performing an activity such as flexing or curling their arm. In addition, when conducting frequency domain analysis, the peak to peak is also necessary to calculate. However, in the EMG signals acquired, it was difficult to tell the peak to peak from the signal itself, thus, the peak to peak was calculated using the MCU and code. It was 15 microvolts for the subject when at rest, around 30 microvolts when the subject was flexing, around 35 microvolts when the subject was curling. Another frequency domain response aspect that was discussed within the proposal was the shape of the signal. When looking at the previous results, the shape of the signal looks extremely similar to that of a general EMG signal. Judging by the specific shape of the signal that was acquired, however, it can be seen that when the subject is flexing or curling, the resulting amplitude becomes higher than that of when the subject was at rest. In addition, the accuracy of the signal was also considered, which was why only two channels, and a tighter set of wires was used to make sure that the signal would not be disturbed and that the accuracy of the signal produced would be as maximized as possible. The inputs for the signal were also considered, or in other words, what activities to do when recording the input EMG signals was considered. For example, as seen in the results section, one plot was when the patient was at rest, one was when the forearm muscle was being flexed, and one was when the forearm muscle was being curled. These experimental inputs were decided beforehand to be able to get a varied set of data where the subject is performing different activities.

The next factors that were considered were the medical factors. The medical factors in this circuit mainly included the user's safety. In general medical factors can include anything from invasive or non invasive procedures to tissue sensor interface requirements to electric safety. When considering the EMG circuit built, the two biggest factors considered were the electrical safety factors and the procedure factors. The electrical safety as mentioned before was considered to make sure the user did not suffer from power leakages or high voltage shocks. Instead of using an opto isolator, the reason why being a part of the economic factors, a full wave rectifier consists of low pass filter and diodes. This method provides a lower degree of safety when compared to the opto isolator, however, this was required to maintain the cost and economic efficiency. However, this is not a declaration that the rectifier does not provide safety. It provides a high degree of user safety, however, it is just not as comparable to that of the opto isolator. The other factor considered was the procedural factors. When considering medical factors, it was important to make sure the collection of data did not disturb or affect the patient in a negative manner. To obtain this ideal, the electrodes used were attachable electrodes that did not require the use of attachable gel. Gel may sometimes irritate the skin or cause issues in the area where it is applied to the patient. Thus to avoid this, the electrodes were used without the need for the gel.

The next set of factors that were considered were the economic factors. The economic factors such as cost efficiency, availability, and compatibility of already existing equipment. The biggest economic factor considered was cost. The materials that were already provided in previous kits were what were preferred instead of buying new components. The circuit was built around already acquired components such as capacitors and diodes. The only thing that was purchased was the arduino starter which contained the arduino UNO board and a multitude of other components such as a pack of resistors. Cost efficiency could have been improved if only the board was purchased on its own, however, it was decided that having a new pack of resistors that contained every resistor value would be beneficial to the implementation of the circuit. In addition, the availability was also considered mainly when buying the new components or in this case the starter kit. The starter kit was ordered online, and thus it was available to be shipped in a day or two, which is why it was considered to be bought in the first place. The last economic factor considered was the compatibility with existing equipment. Some of the equipment that was considered was the equipment provided within the lab, such as the function generator and the oscilloscope. The material used and the material that the circuit was built on, the breadboard was able to have enough space to completely encase the entirety of the circuit and also be able to adeptly connect to the equipment within the lab.

The last set of factors that were considered were the environmental factors. Examples of these factors would be things like the signal to noise ratio, stability, pressure, acceleration, shock, and power requirements. It is widely known that every signal that is collected will have some sort of noise within the signal that is unwanted and unnecessary. Thus to combat this, the signal was adjusted as much as possible using filters within the circuit to generate as little noise as possible. In addition, the factors such as stability, acceleration, shock, and pressure were all considered during the data collection. For example, when collecting the data for a flexing arm, the pressure being applied during the flexing is affected, and thus must be maintained to understand the signal acquired. On the other hand, if the flexing or contractions of the muscle are too fast, or accelerate too much, the signal will come out as an incorrect signal. Due to this reason, the contractions or the muscle flexing had to be maintained to a reasonable rate to make sure the signal was able to be collected in an organized manner. Moreover, any type of shock would have resulted in extra noise within the signal or disrupted the original frequency domain response specifications. For example, if while the subject was at rest, and an external shock was applied to the patient's forearm, the EMG signal would then pick up the signal generated by that shock instead of the actual at rest signal.

VI. CONCLUSION

After completing this final project, we now have a good understanding of how to design and build an EMG. We were able to successfully program the arduino uno microcontroller. As a result, we transmitted our acquired EMG signals into Labview to be viewed on a waveform graph. Furthermore, we were successfully able to perform a frequency signal analysis on the signal obtained from the EMG. We found the peak to peak and frequency values at different subject positions such as from resting, flexing, and curling the arm. Overall, electromyography is important in determining the health of muscles and the motor neurons that control them. After running an EMG test, doctors can determine if symptoms are a result of nerve issues or muscle disorder.

VII. REFERENCES

[1] Dr A. Tavallaei "AD620ANZ Datasheet", D2L content, Toronto metropolitan university, 2023.

[2] P. JH; "AAEM minimonograph #3: Motor Unit Recruitment2018," *Muscle & nerve*, 2014. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/1852155/#:~:text=Motor%20unit%20recruitment%20is%20the,of%20individual%20motoneurons%20(MNs). [Accessed: 18-Mar-2023].

[3] "Muscle Fibre types," *Physiopedia*, 2021. [Online].Available: https://www.physio-pedia.com/Muscle_Fibre_Types#:~:text=The%20three%20types%2

0of%20muscle%20fiber%20are%20slow%20oxidative%20. [Accessed: 18-Mar-2023].

[4] Dr A. Khademi “AD620ANZ D”, D2L content, Toronto metropolitan university, 2023.

VIII. APPENDIX

Circuit Design Calculations

The resistor values were calculated based on the expected gain of 1000.

Calculations of resistor values in the circuit:

$Gain = 1 + (50.5K\Omega/Rg)$

$Rg = 50.5k\Omega, /(Gain - 1)$

For a Gain of 1000:

$Rg = 50.5k\Omega, /(1000 - 1)$

$Rg \approx 50.5\Omega$

Capacitor 1 Calculation

Using the expected cutoff frequency of 7.96 Hz and assuming 2kΩ:

$Fc = 1/(2 \cdot \pi \cdot R \cdot C)$

$C = 1/(2 \cdot \pi \cdot R \cdot Fc)$

$C = 1/(2 \cdot \pi \cdot 2k\Omega \cdot 7.96Hz)$

$C = 10uF$

Capacitor 2 Calculation

Using the cutoff frequency of 10.01Hz and assuming 1.59kΩ:

$Fc = 1/(2 \cdot \pi \cdot R \cdot C)$

$C = 1/(2 \cdot \pi \cdot R \cdot Fc)$

$C = 1/(2 \cdot \pi \cdot 1.59k\Omega \cdot 10.01Hz)$

$C = 10uF$

Resistor Values For The Center Frequency and Bandwidth of the Low-Pass Filter

$Fc = 1/(2 \cdot \pi \cdot R2 \cdot C1)$

$R2 = 1/(2 \cdot \pi \cdot Fc \cdot C1)$

$R2 = 1/(2 \cdot \pi \cdot 100Hz \cdot 10uF)$

$R2 \approx 1.59k\Omega$

.

Microcontroller Final Code

```
#include <FFT.h> // Include the
FFT library

// Constants
const int EMG_PIN = A0; // Analog
input pin for EMG signal
const int SAMPLE_SIZE = 256; //
Number of samples for FFT
const int FFT_MAX_FREQ = 500; //
Maximum frequency for FFT plot
const double SAMPLING_FREQUENCY =
1000.0; // Sampling frequency of
EMG signal
const double VREF = 5.0; //
Reference voltage of Arduino (5V)

// Variables
double EMGArray[SAMPLE_SIZE]; //
Array to hold EMG samples
double peakToPeak = 0.0; //
Peak-to-peak amplitude of EMG
signal
double frequency = 0.0; //
Frequency of EMG signal

void setup() {
  Serial.begin(9600); //
Initialize Serial communication



  // Set up FFT
  FFT.useAVRfix(); // Enable AVR
fix for Arduino boards

FFT.Windowing(FFT_WIN_TYP_HAMMING,
FFT_FORWARD); // Set Hamming
window for FFT
  FFT.ComputeN(); // Compute N for
FFT
```

```
FFT.SetInputScaling(FFT_INPUT_ON); // Enable input scaling for FFT
}

void loop() {
  // Read EMG signal from array or input stream (replace with actual EMG samples)
  for (int i = 0; i < SAMPLE_SIZE; i++) {
    // Use EMGArray[i] = EMGValue; to assign EMG samples to array
    // or use EMGArray[i] = analogRead(EMG_PIN) * (VREF / 1023.0); to read from analog input pin
    // Replace with your own EMG acquisition code
    // Example: EMGArray[i] = ...;
  }

  // Perform FFT

FFT.Windowing(FFT_WIN_TYP_HAMMING, FFT_FORWARD); // Apply windowing to samples
  FFT.Compute(EMGArray, SAMPLE_SIZE); // Compute FFT
  FFT.ComplexToMagnitude(); // Convert complex data to magnitude

  // Find peak-to-peak amplitude of EMG signal
  peakToPeak = 0.0; // Initialize peak-to-peak amplitude
  for (int i = 0; i < SAMPLE_SIZE; i++) {
    double amplitude = EMGArray[i];
    if (amplitude > peakToPeak) {
      peakToPeak = amplitude; // Update peak-to-peak amplitude
    }
  }

  // Find frequency of EMG signal
  frequency = FFT.MajorPeak(); // Get major peak frequency from FFT

  // Print frequency domain specifications
  Serial.print(frequency);
  Serial.print(",");
  Serial.println(peakToPeak);

  delay(1000); // Delay for visualization purposes (adjust as needed)
}
```

Labview Diagrams

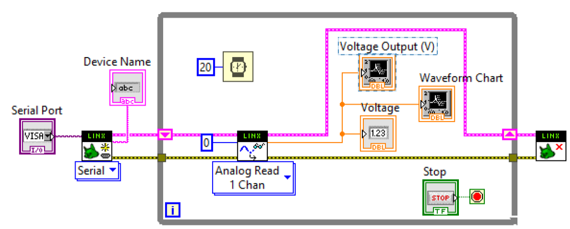

Figure 7: Labview Block Diagram

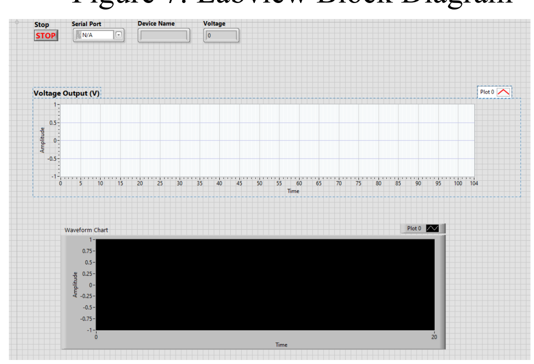

Figure 8: Labview Front Panel.