University of Colombo
Department of physics

HARDWARE INTERFACING

Chandima Kasun Edirisinghe
Index 12033
July, 2017

Abstract

The purpose of this report is to effectively use the LabVIEW software in data acquisition and data manipulation. Data acquisition and data manipulation are very important aspects of many modern systems including scientific applications. The first part of the practical is based on basic operations in LabVIEW to get familiar with the GUI and the tools of the LabVIEW. The second part of the practical include applying the learnt techniques in signal generation, signal analysis and data acquisition and processing. In signal generation, a signal generator capable of producing sinusoidal, triangular, rectangular and saw tooth signals was created. In signal analysis, an oscilloscope was created. In data acquisition and processing, a resistance meter, a capacitance meter and a diode curve generating tool were created.

Contents

List of Figures

1 Introduction

The advent of the computer revolutionized the Science and technology vehemently. As a result of this, human thinking and intuition prospered limitlessly to an extent that flying which was thought of as an impossible feat was achieved. The computer became so popular and widely used to an extent that it became a mere necessity in the daily life.

As the computer was used in a myriad of applications, techniques were developed acquire and process data. For this process, data acquisition systems were invented. A data acquisition system (DAQ system) can be simply defined as a system developed to measure an electrical or physical phenomenon such as voltage, current, temperature, pressure, or sound with a computer.

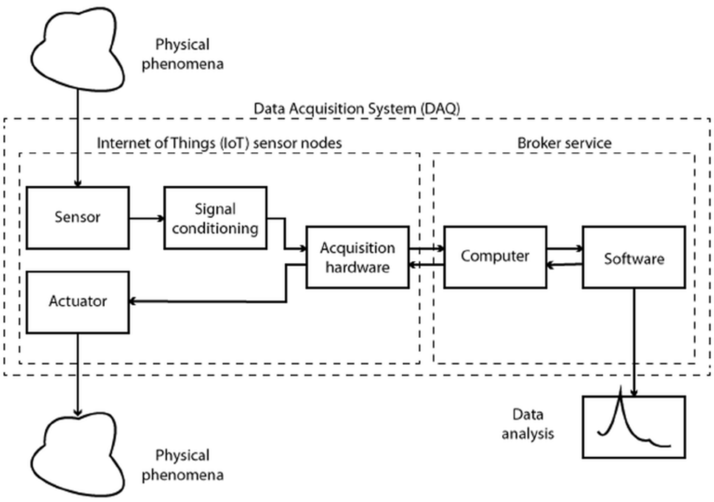


Figure 1.1: Components of a DAQ system

With the development of the data acquisition systems, software able to control these systems were introduced. LabVIEW is such a software. LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a system-design platform and development environment for a visual programming language. Unlike other programming languages like C or Java, LabVIEW uses a data-flow programming language consisting of icons instead of lines of text.

When using LabVIEW to acquire and manipulate data, a data acquisition hardware will have to be used to convert the physical parameter which is being measured to a signal which can be processed by the software through the computer. For this purpose, the most commonly used component is the DAQ card.

Figure 1.2: A DAQ system with a DAQ card and LabVIEW

2 Theory

In a data acquisition system, the analog signals of some physical parameters are digitized in order to be captured and manipulated through a computer. In this process, there are mainly 2 essential components required.

1. A software capable of data capturing and manipulation
2. A data acquisition hardware
3. A computer

2.1 LabVIEW software

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a platform and a development environment from National Instruments which comprises of a visual programming language. LabVIEW can be used in a myriad of applications. Some of them are as follows.

- As a DAQ system able to measure physical parameters.
- Validation or verification of electronic designs.
- Development of production test systems.
- Designing of smart machines or industrial equipment.

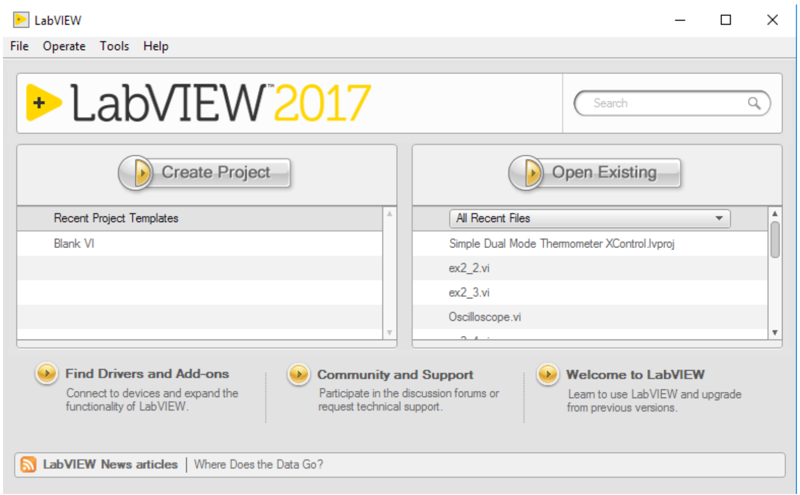
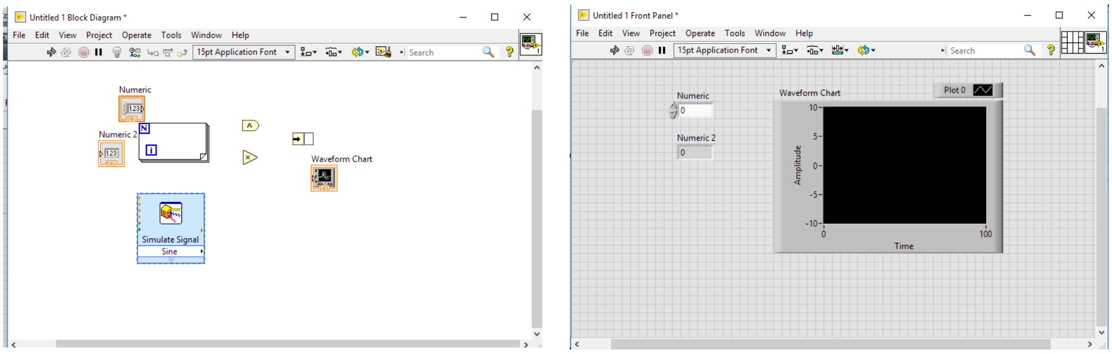


Figure 2.1: Main interface of LabVIEW

A new program in LabVIEW can be created by creating a new VI. A VI comprises of 2 interfaces.

1. Block diagram used to create the program.
2. Front panel used to execute the created program and display results.



(a) Block diagram (b) Front panel

Figure 2.2: The interfaces of a new VI

The program is created in the block diagram. The functions needed for a specific program can be directly inserted by using the function panel which can be accessed by right clicking on the block diagram.

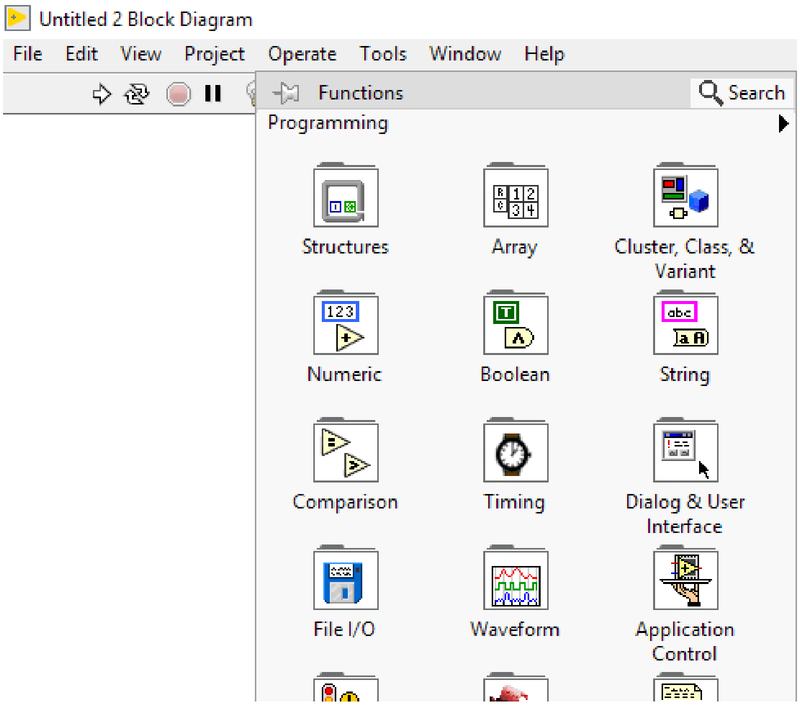


Figure 2.3: Function palette

The indicators and controllers can be used to display and control the processed output. The output can be viewed from the front panel using the indicators. The control panel is used to insert the needed indicators and they could be obtained by accessing the control palette by right clicking on the front panel.

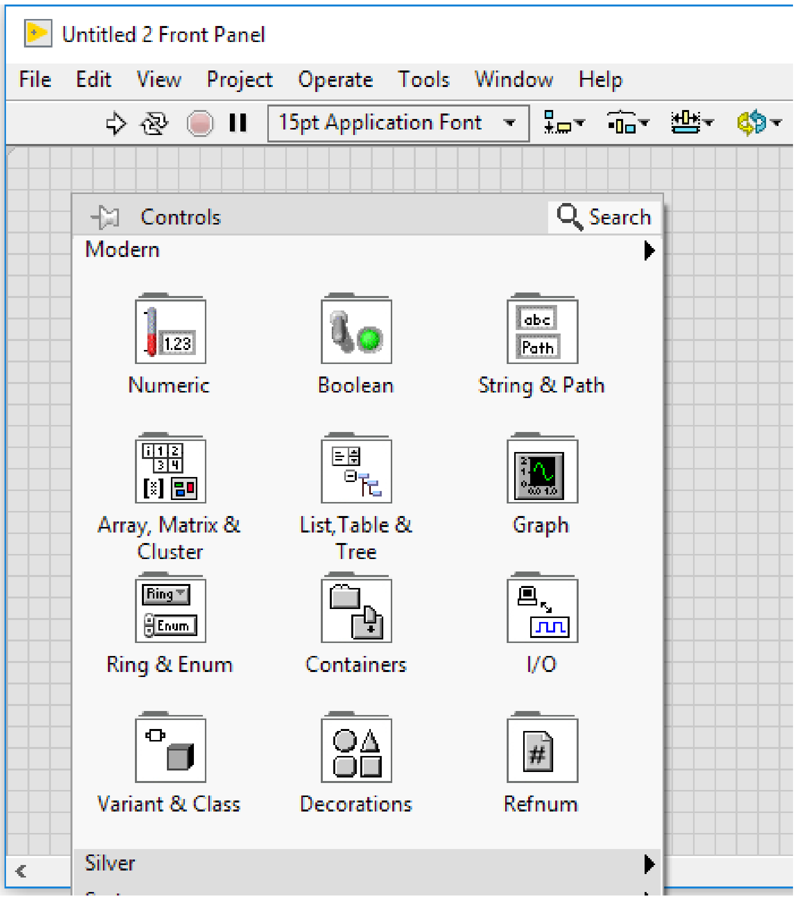


Figure 2.4: Control palette

2.2 DAQ hardware compatible with LabVIEW

The NI USB-6008 DAQ card is a DAQ hardware commonly used in tandem with LabVIEW in DAQ applications. The USB-6008 provides basic DAQ functionality for applications such as simple data logging, portable measurements and academic lab experiments. It is an affordable component which is powerful enough for more sophisticated measurement applications.

The USB-6008 comprises of the following features.

- 8 analog inputs (12-bit, 10 kS/s)
- 2 analog outputs (12-bit)
- 12 digital I/O

Figure 2.5: Front view of the NI USB-6008 DAQ card

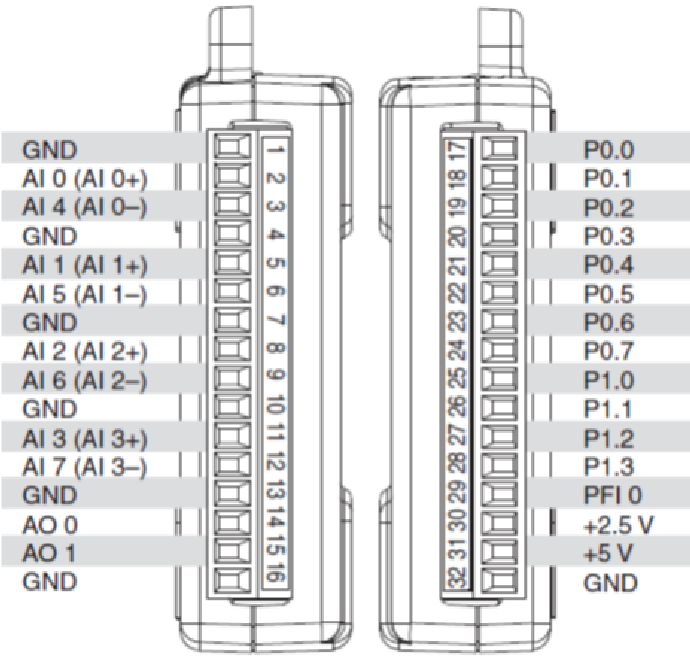


Figure 6. NI USB-6008/6009 Pinout

Figure 2.6: Pinout of the USB-6008

3 Methodology

3.1 Introduction to LabVIEW

In this section, the functions and indicators in LabVIEW were introduced.

3.1.1 Temperature unit converter

In this exercise, a temperature converter was interfaced which accepted any temperature in Celsius and converted it to a temperature in Kelvin or Fahrenheit. For this operation, the following equations were used.

1. To convert to Kelvin (K) from Celsius ($°C$)
   $K = C + 273.15$

2. To convert Celsius ($°C$) to Fahrenheit ($°F$)
   $F = \frac{9}{5} \times C + 32$

The block diagram was constructed as follows.

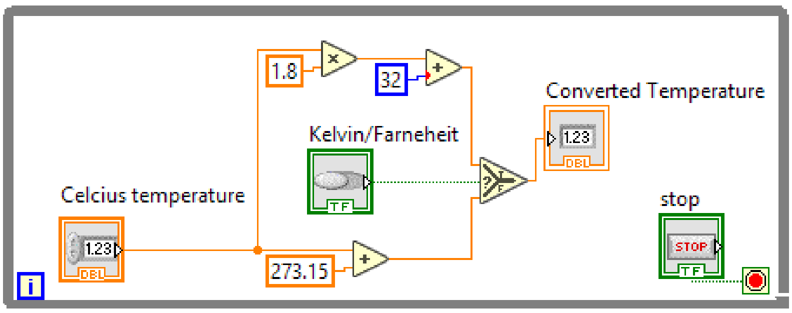


Figure 3.1: Block diagram for the temperature converter

In this exercise, a numeric control (Celsius temperature) was used to input the needed temperature and the output was displayed through the numeric indicator (Converted temperature). The switch was used to select the needed unit (Kelvin or Fahrenheit).

3.1.2 Wave mixer

In this experiment, a wave mixer was constructed by superimposing 3 sinusoidal signals. In this process, numeric controls were added to control the phase, frequency and amplitudes of the 3 signals separately. Then, the 3 created signals and the superimposed signal were plotted in waveform graphs. Finally, the Fourier transformed outputs of the 3 input signals and the superimposed signal were plotted in waveform graphs. In obtaining the FFT curves, the axes were calibrated suitably. In this process, the following steps were followed.

1. The FFT values of the signal were obtained using the FFT transform control in the signal processing tool box. Here, only the first 1000 samples of the signal was used for the FFT.

2. Then, the first 500 values were chosen (the next 500 values are the mirror image values of the first 500 values)

3. Then, the obtained 500 values were divided by the maximum value in the set (to normalize the y axis) and the result was multiplied by the amplitude of the signal to obtain the calibrated y axis

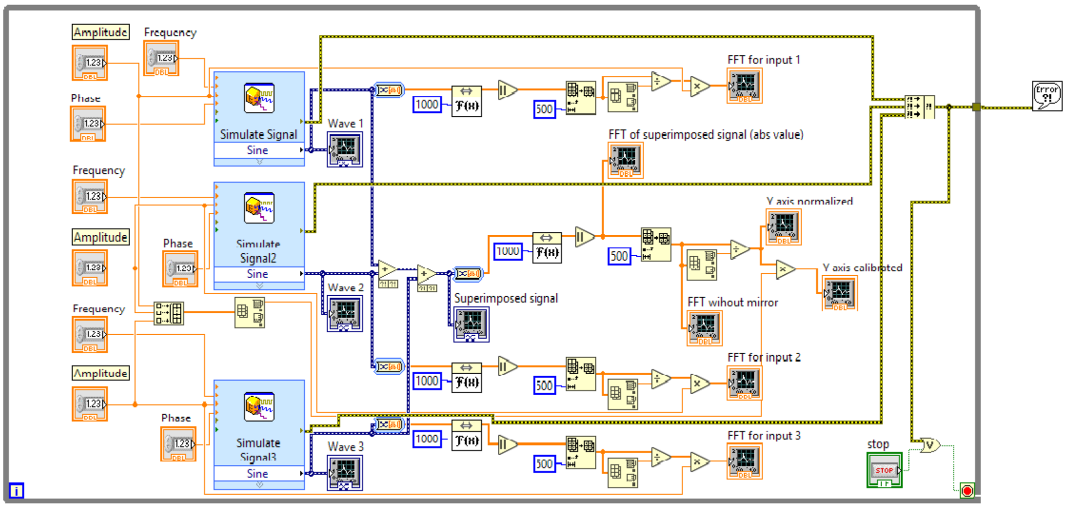


Figure 3.2: Block diagram for the wave mixer

3.1.3 Frequency sequence generator

This experiment comprised of 3 exercises.

1. Creating a sinusoidal signal and playing it through the speaker connected to the sound card of the computer.
2. Playing the pure notes of the C octave starting from the middle C (C_4).
3. Playing the melody of a song.

In the first exercise of this experiment, a sinusoidal signal was created with the ability to control its frequency and amplitude and output. Then, the created signal was played through the speaker. In this process, the **play waveform** control was used to output the signal. In sampling the created signal, the sampling frequency and the number of samples could be adjusted so that the playing tempo of the signal could be adjusted.

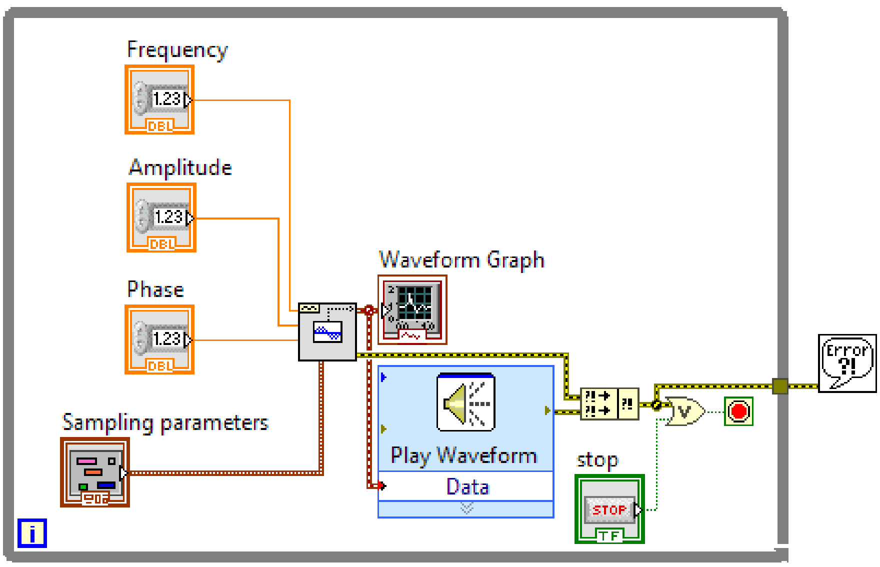


Figure 3.3: Block diagram for playing a signal

In the second exercise of this experiment, the pure notes of the C_4 octave was played through the speaker. In this process, the frequencies of the notes were stored in an array. In sampling the created signal, the sampling frequency and the number of samples could be adjusted so that the playing tempo of the signal could be adjusted.

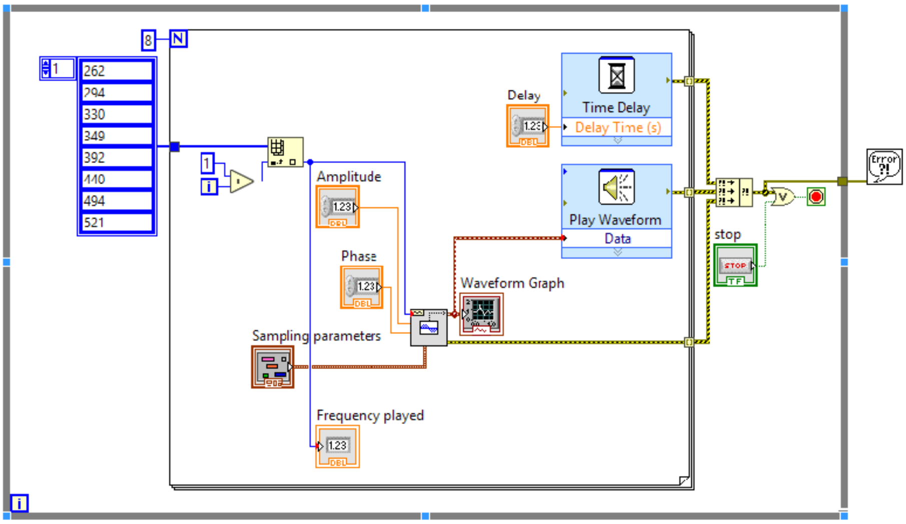


Figure 3.4: Block diagram for playing the pure notes of the C_4 octave

In the third exercise of this experiment, the melody of the song ”Diya goda sema thena” by Sunil Shantha was played using the speaker. In this process, the notes and timing of the song was stored in a 2 dimensional array. In sampling the created signal, the sampling frequency and the number of samples could be adjusted so that the playing tempo of the signal could be adjusted.

In addition to the changing of the amplitude and the tempo of the song, a control for shifting the pitch of the melody by transposing was added. From this control, the melody could be transposed either up or down by the needed amount of semitones. For example, the melody originally played on the C major scale could be easily transposed to D major scale by entering 2 in the **Transpose** numeric control.

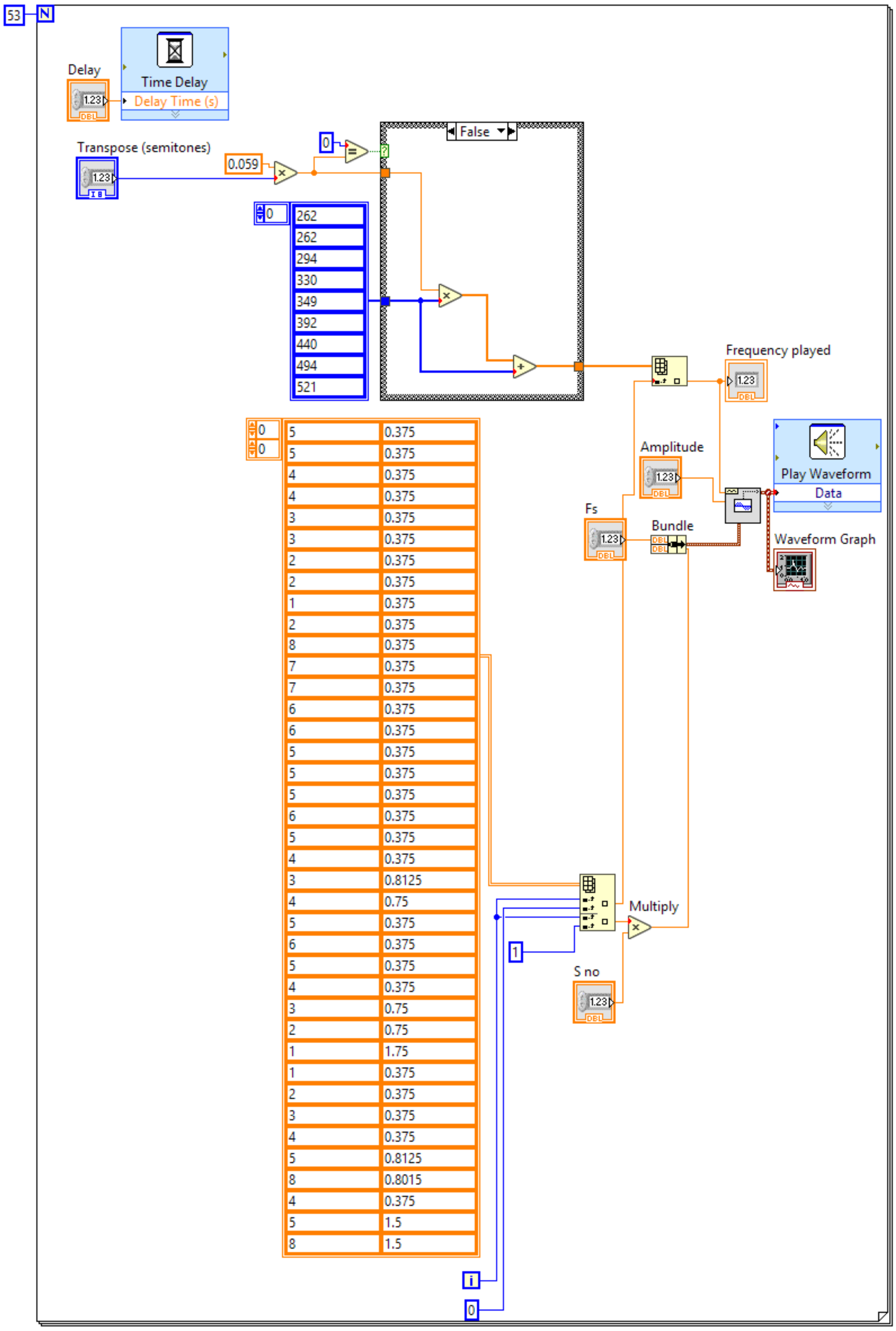


Figure 3.5: Block diagram for playing the melody of the song

3.1.4 DTMF decoder and encoder

In this experiment, as the first exercise, a DTMF encoder was constructed. In this process, the frequencies needed for the DTMF generation were obtained from 2 arrays and the sinusoidal signals with the corresponding frequencies were superimposed together to generate DTMF signals. Here, the time in which a sound played after a key was pressed was adjusted so that the sound played for 30 ms. It is the standard time for a DTMF signal. But this time could be adjusted by changing the sampling frequency and the number of samples.

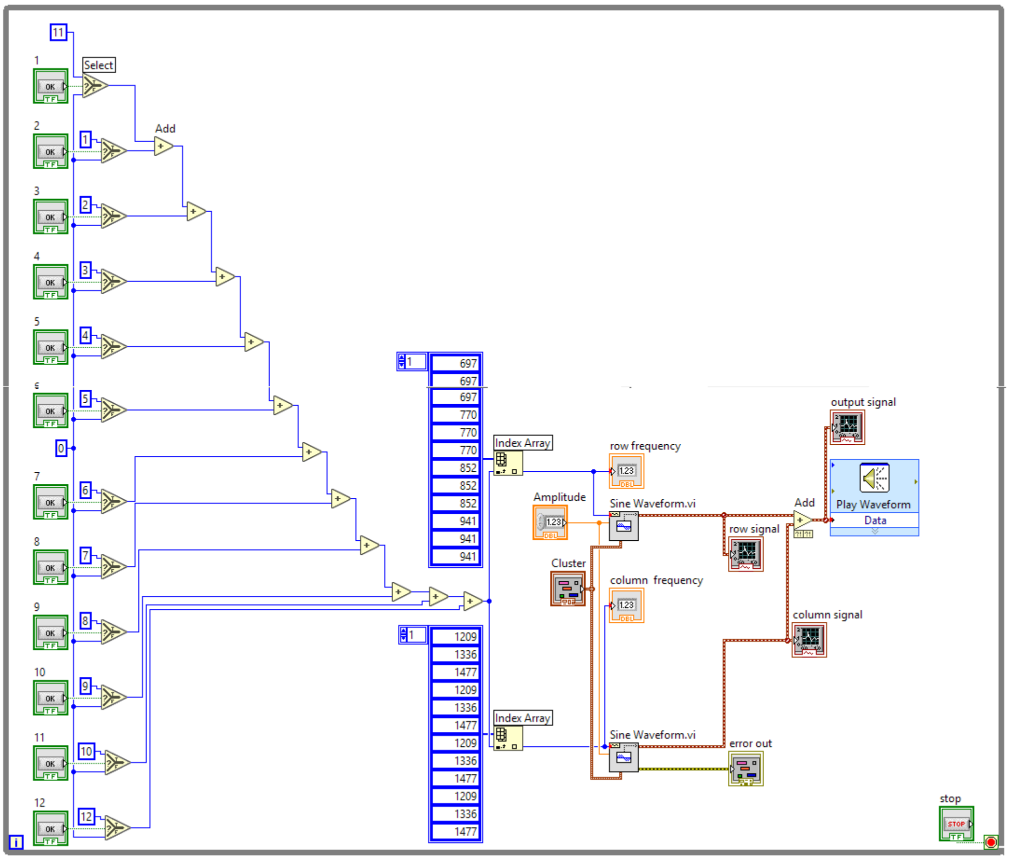


Figure 3.6: Block diagram for DTMF generation

In the second exercise of this experiment,a DTMF decoder was constructed. In this process, the signal was first filtered out using 2 Butterworth band pass filters of order 5 to separate out the low frequency and the high frequency components in the signal. In this process, the filter used to identify the low frequency had a lower cutoff frequency of 650 Hz and a higher cutoff frequency of 970 Hz. The filter used to identify the high frequency had a lower cutoff frequency of 1170 Hz and a higher cutoff frequency of 1500 Hz.

Then, the 2 frequencies of the 2 filtered signals were identified using the **Tone measurement** tool in the signal processing toolbox.

Finally the pressed key was detected and shown using the **Played button** indicator if the identified frequencies were between ± 20 Hz of the exact frequencies of the DTMF tones.

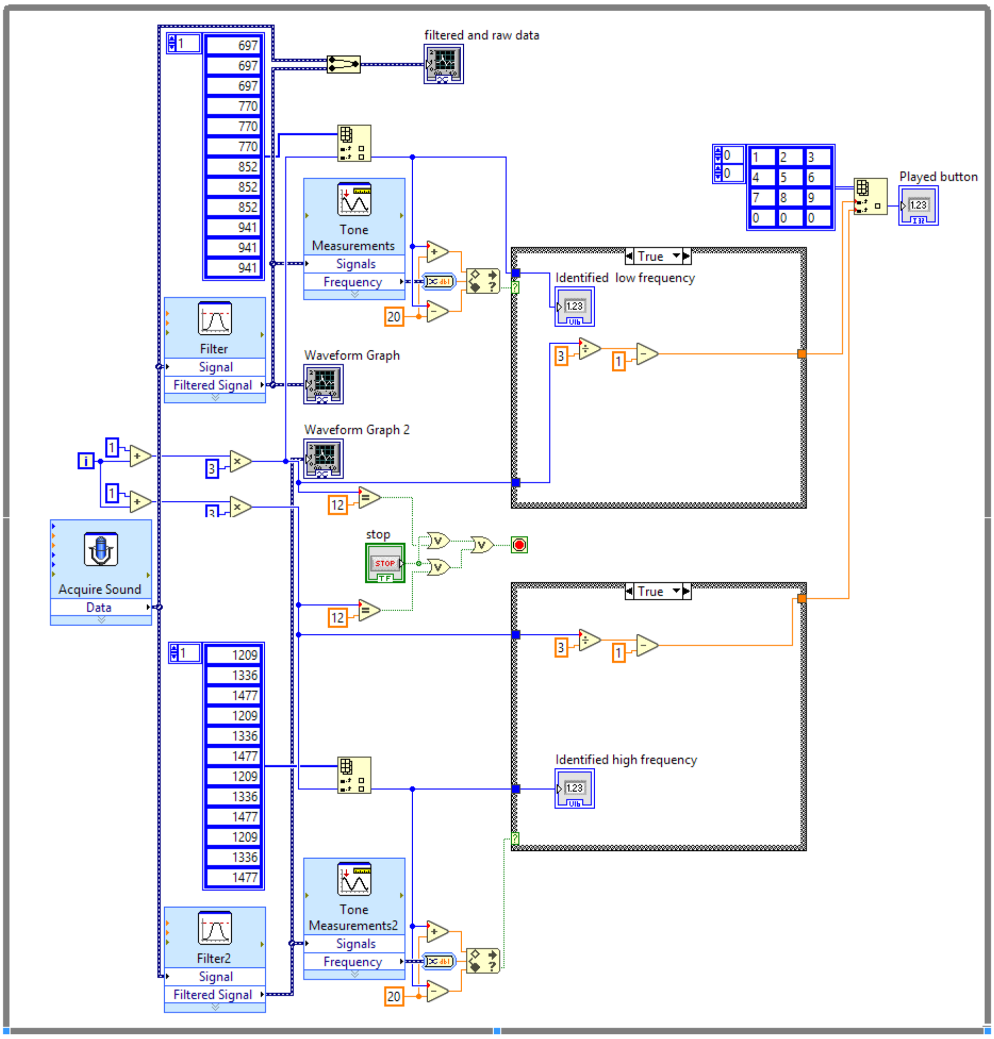


Figure 3.7: Block diagram for DTMF decoding

3.2 Interfacing the DAQ card

In this section, the USB-6008 DAQ card was used in various applications by interfacing it with LabVIEW.

3.2.1 Creating a function generator

In this exercise, a function generator capable of producing sinusoidal, square, triangular and saw tooth waves was created. In this process, the amplitude and the frequency of the signal could be adjusted by using the numeric controls.

A constant sampling frequency of 1000 Hz and a sample size of 100 was given to the signal generators as the sampling information cluster.

The input frequency was multiplied by a factor of 4 to get the output signal with the needed frequency.

An offset of 2.5 V was given to the signal as the DAQ card was only capable of producing voltages between 0 V and 5 V. Negative voltages couldn't be produced from the DAQ card.

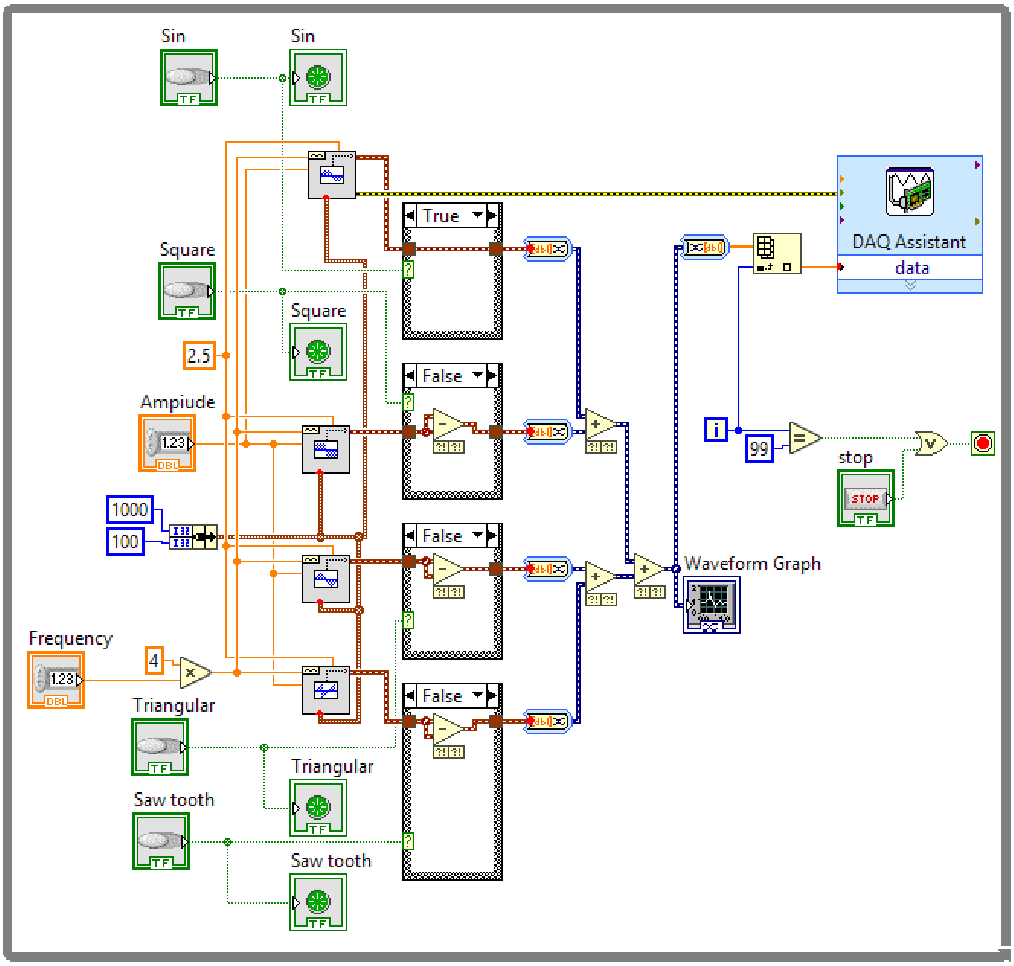


Figure 3.8: Block diagram for the function generator

3.2.2 Four channel oscilloscope

In this exercise, a four channel oscilloscope was constructed. In this process, the AI0, AI1, AI2 and AI3 analog inputs were used as 4 channels to input any needed signal. The signals corresponding to channel 1 (AI0), channel 2 (AI1), channel 3 (AI3) and channel 4 (AI3) were represented using the colours white, red, green and blue respectively. The needed channel could be selected by switching the boolean switch.

For each channel, a fixed offset value was found by trial and error was added in order to show the correct amplitude reading.

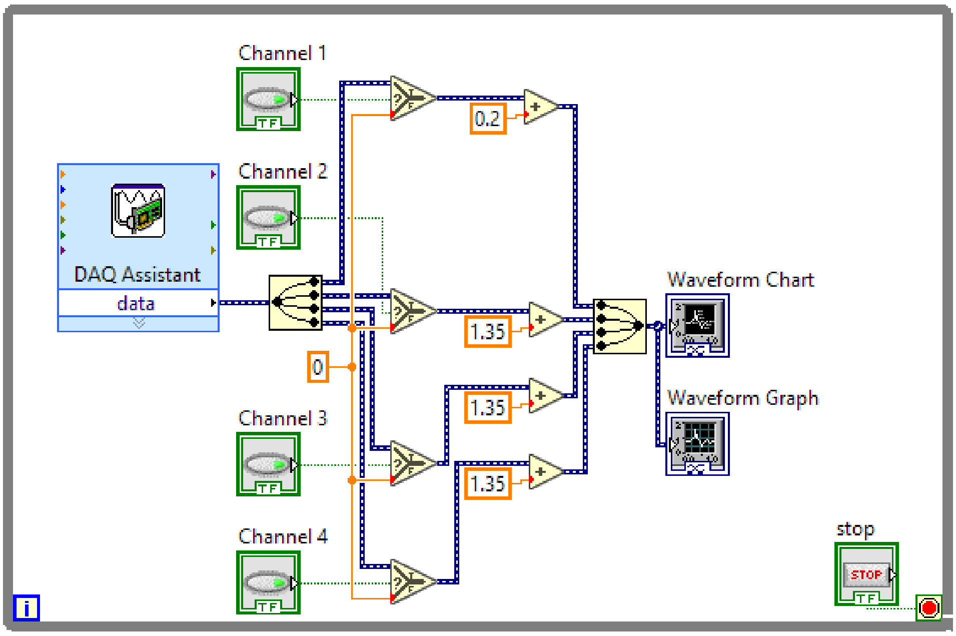


Figure 3.9: Block diagram for the four channel oscilloscope

3.2.3 Resistance meter

In this experiment, a resistance meter was created. In this process, the following circuit was used to voltage across the unknown resistance by the DAQ card and thereby the unknown resistance.

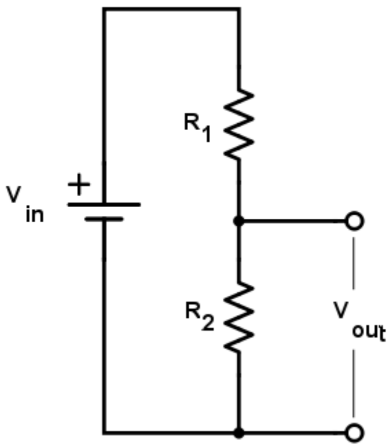


Figure 3.10: Circuit diagram of the resistance meter circuit

From the Ohm's law, an expression for the V_{out} in terms of the input voltage V_{in}, known resistance R_1 and the unknown resistance R_2 can be written as follows.

$$V_{out} = V_{in} \times \frac{R_2}{R_2 + R_1} \tag{1}$$

As the V_{out} can be measured using the DAQ card and LabVIEW, the unknown resistance can be found by the following equation.

$$R_2 = V_{out} \times \frac{R_1}{V_{in} - V_{out}} \tag{2}$$

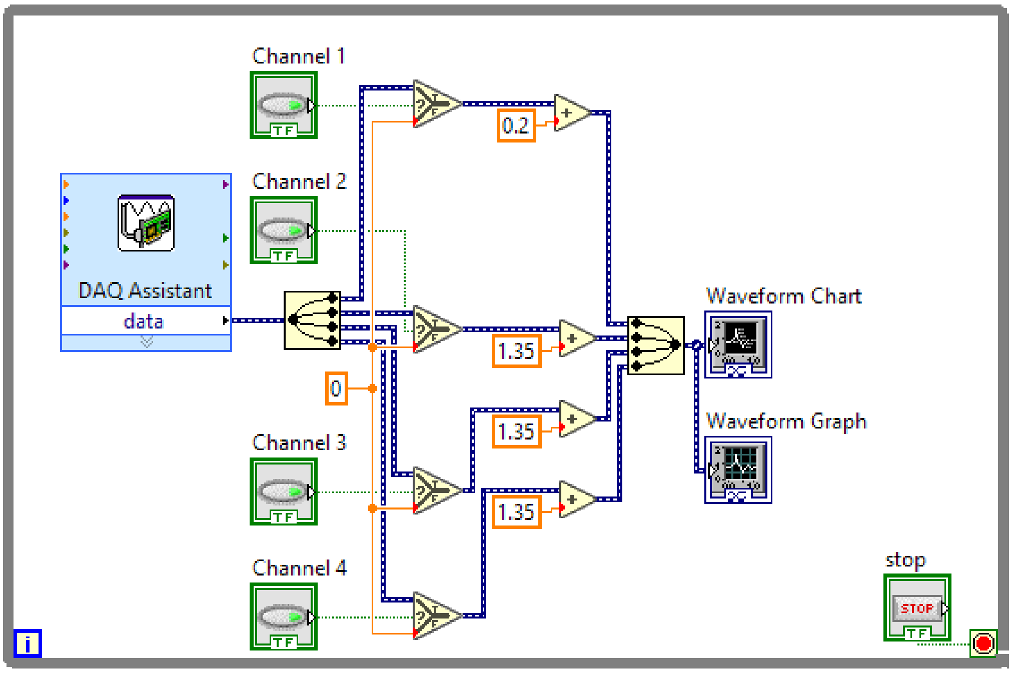


Figure 3.11: Block diagram of the resistance meter

3.2.4 Generation of the VI curve of a diode

In this experiment, the VI curve of a diode was generated. A LM 4007 diode was used to generate the VI curve. In this process, the following circuit was constructed.

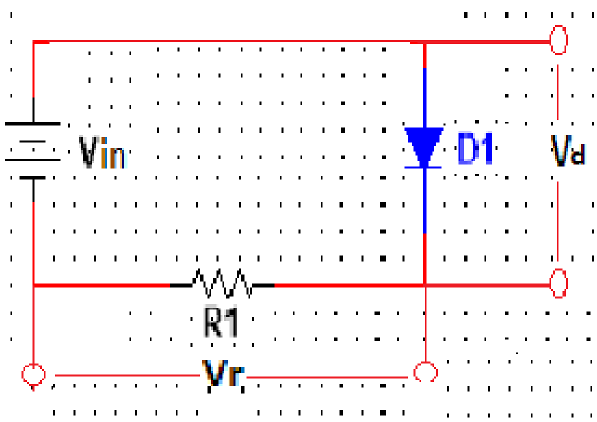


Figure 3.12: Circuit diagram for generating the VI curve of a diode

Here, the input voltage (V_{in}) was incremented in increments of any needed step size and that voltage was output through the DAQ card and connected in series to the resistor (R_1) as in the circuit. Here, the connected resistor's resistance must be input in the resistance control in the front panel of the LabVIEW application.

Then, the voltage through the resistor (R_1) is measured through the DAQ card. Then, the following equation was used to find the voltage through the diode (V_d).

$$V_d = V_{in} - V_r \tag{3}$$

The current through the diode was calculated from the following equation.

$$I_d = \frac{V_r}{R_1} \tag{4}$$

Then, the calculated V_d was plotted against the calculated I_d using a **xy graph** .

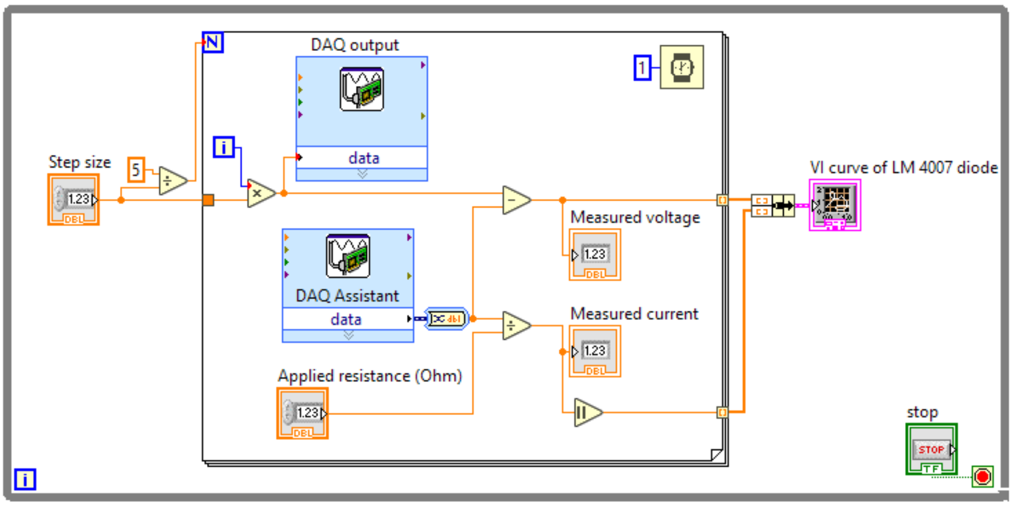


Figure 3.13: Block diagram for generating the VI curve of a diode

3.2.5 Capacitance meter

In this experiment, a capacitance meter was constructed. In this process, a charging capacitor was used to determine the time constant (τ) and thereby, the capacitance of the capacitor. The following circuit was constructed in order to measure the capacitance.

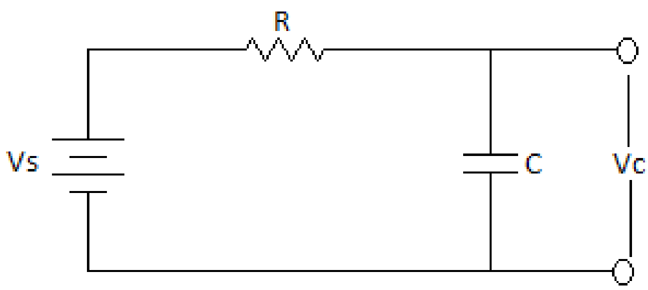


Figure 3.14: Circuit diagram for the capacitance meter

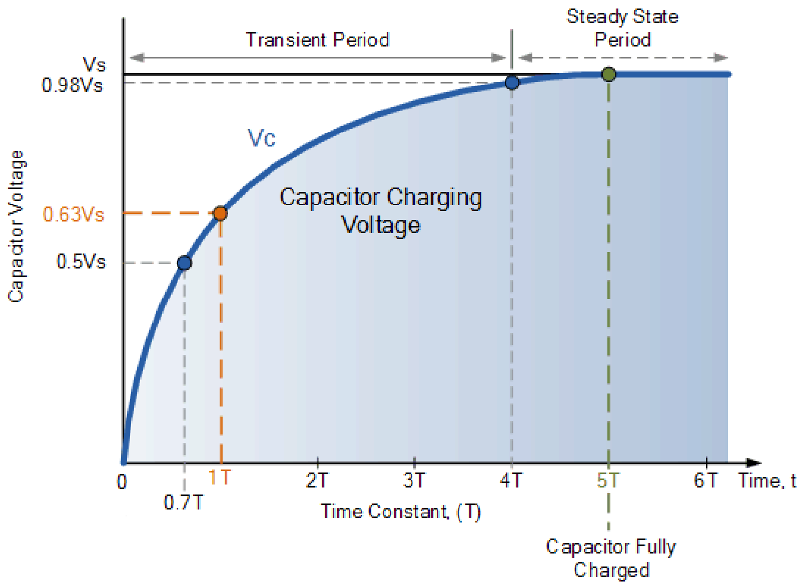


Figure 3.15: Charging curve of a capacitor

To find the capacitance, the following equation was used.

$$V_c = V_s\,(1 - e^{-\dfrac{t}{RC}}) \tag{5}$$

In calculating the time constant, the first step was to calculate the time taken until $V_c = 3.15V$ when the supply voltage was 5 V ($V_s = 5V$). Then, the above equation (5) can be simplified as follows.

$$\begin{aligned} 3.15 &= 5\,(1 - e^{-\dfrac{t}{RC}}) \\ e^{-\dfrac{t}{RC}} &= 1 - \frac{3.15}{5} \\ t &\approx RC \end{aligned} \tag{6}$$

So, from equation (6), when time is measured and the resistance is known, the capacitance can be calculated.

4 Results and Analysis

4.1 Introduction to LabVIEW

4.1.1 Temperature unit converter

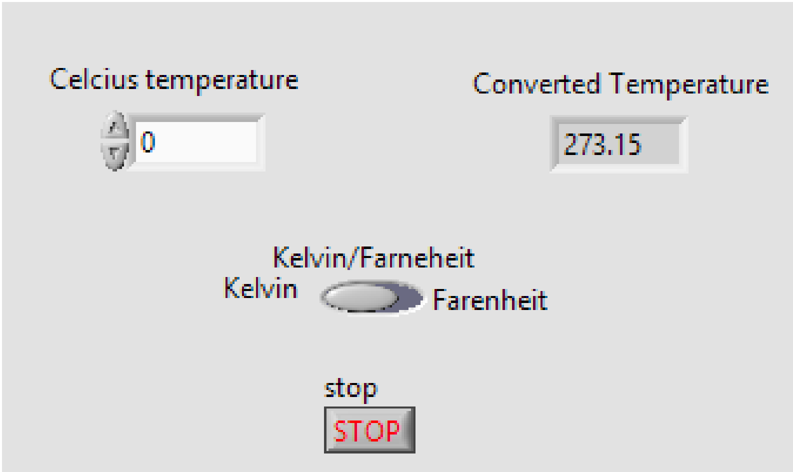


Figure 4.1: Temperature unit converter front panel

As shown above, using this application, any temperature could be converted from Celsius scale to Kelvin or Fahrenheit scale. The output scale could be controlled by the switch by selecting the Kelvin or the Fahrenheit scale.

4.1.2 Wave mixer

In this experiment, the FFT of 3 sinusoidal signals and the FFT of the signal formed by superimposing the 3 signals were obtained. When obtaining the FFT, the first 1000 samples of the respective signals were taken into consideration. There was no need to choose a larger samples as the frequencies and the amplitudes of the signal weren't fluctuating with time. The plotting of FFT curves were done using the first 500 samples (single sided amplitude spectrum) as the second 500 samples were the mirror image of the first set. The normalized output was obtained by dividing the FFT values by the maximum FFT value. For this process, an array of FFT values were created and the maximum of that array was obtained and all the FFT values were divided from that maximum value.

When obtaining the calibrated FFT outputs of the signals, the normalized values were multiplied by the maximum amplitude. To find the maximum amplitude of the superimposed signal, an array consisting of the 3 amplitude values of the 3 sinusoidal signals was created and the maximum of that array was obtained.

The front panel of the wave mixer was as below.

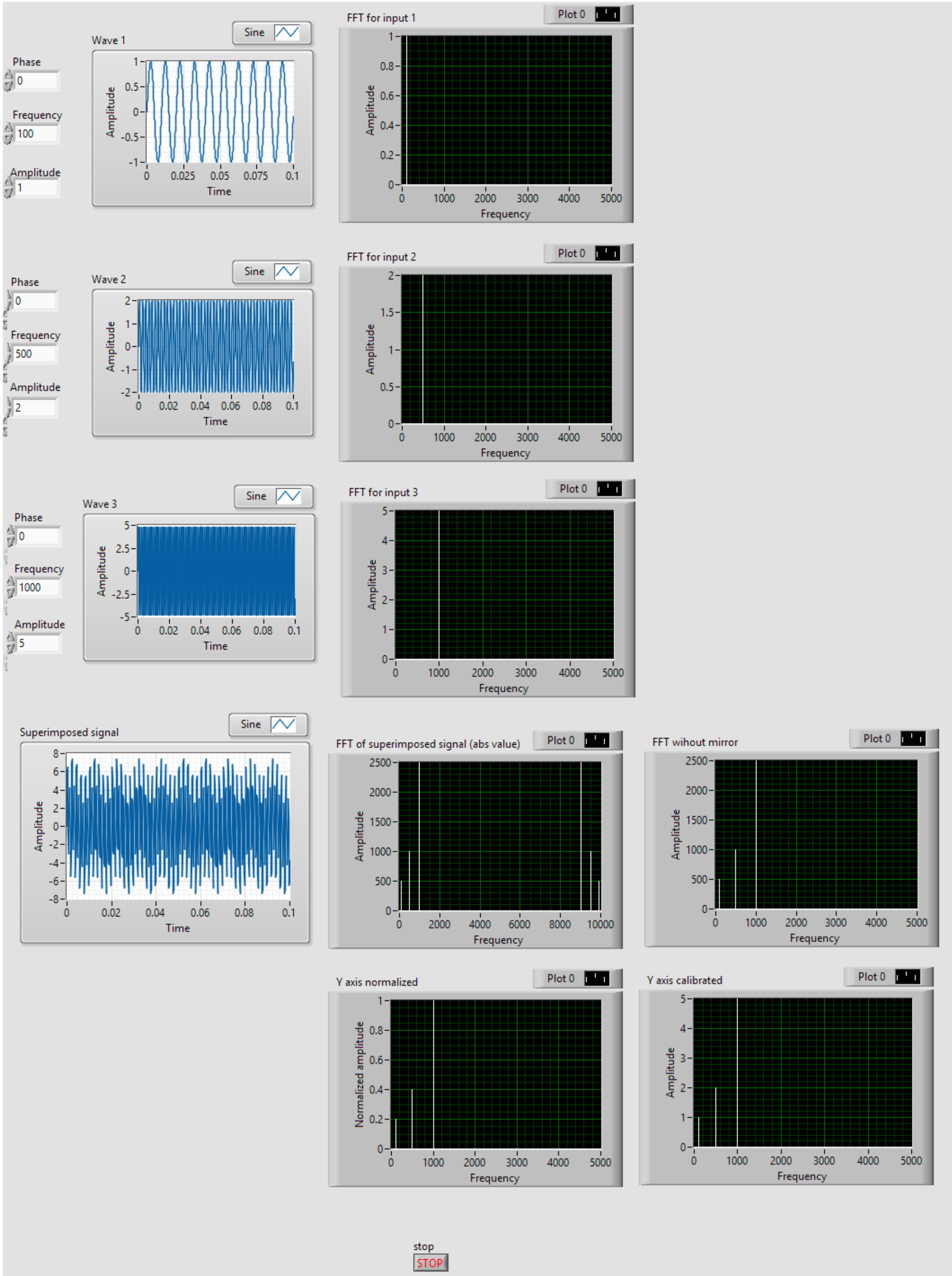


Figure 4.2: Wave mixer front panel

4.1.3 Frequency sequence generator

Sin wave generator

The front panel of the sin wave generator was as below.

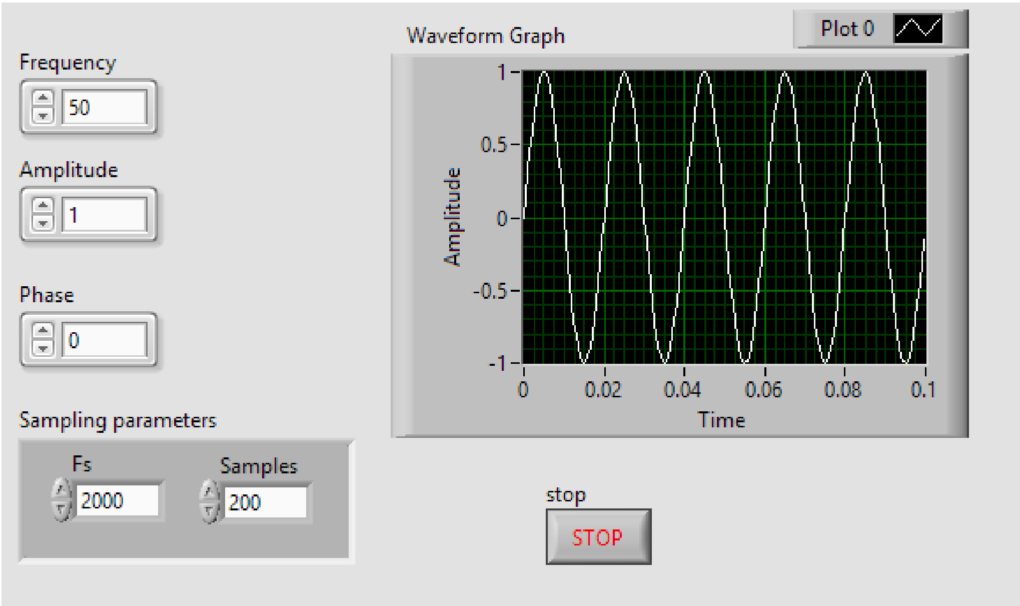


Figure 4.3: Sin wave generator front panel

In this part, the sinusoidal signal corresponding to the entered frequency, amplitude and phase was played through the speaker of the computer. There was stark difference in the sound heard through the computer speaker when the frequency changed. But when amplitude or phase was changed, there was no apparent change in the sound heard.

Playing the pure notes of the middle C (C_4) octave

In this part, the pure notes of the middle C octave was played through the speaker of the computer. The front panel of the middle C octave generator was as below.

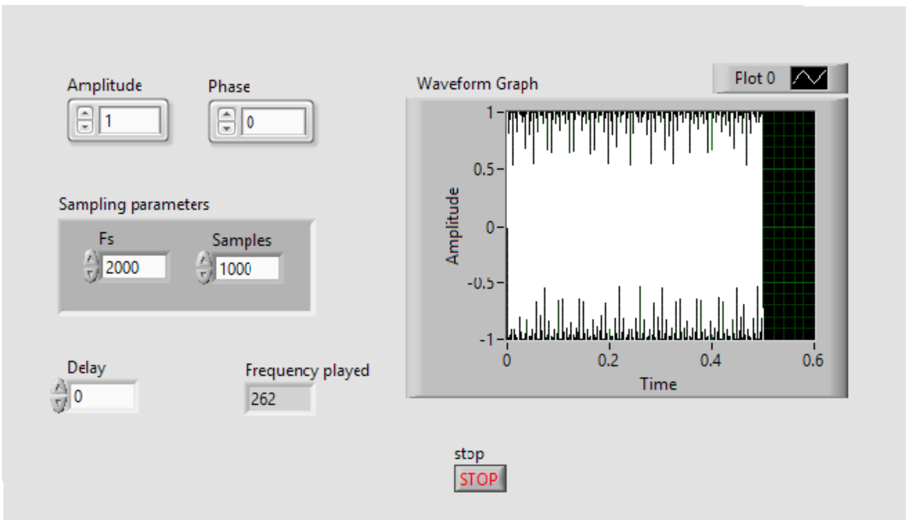


Figure 4.4: Middle C octave's pure note player front panel

Playing the melody of a song

The front panel of the melody player was as below.

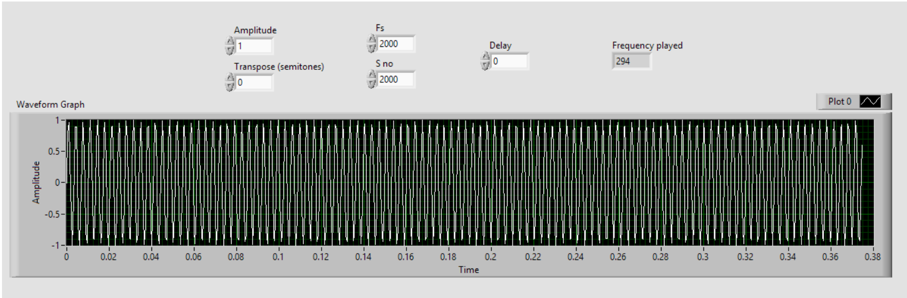


Figure 4.5: Song melody player front panel

In this part of the exercise, the above 2 parts were combined together with some additional blocks to play the melody of the song **Diya goda sema thena**. In this process, the notes and timing of the song was saved in an array. To find the relevant notes and timing, the tab sheet in the Appendix was used in tandem with a guitar.

By adjusting the sampling information namely sampling frequency and the number of samples, the tempo of the song could be adjusted. By adjusting the delay, the time gap between 2 notes could be adjusted. The transpose option could be used to transpose the key of the song to another note. Essentially, this application could be used as a simple music maker software capable of editing notes, timing, tempo and the pitch of the song.

4.1.4 DTMF decoder and encoder

DTMF encoder

The front panel of the DTMF encoder was as below.

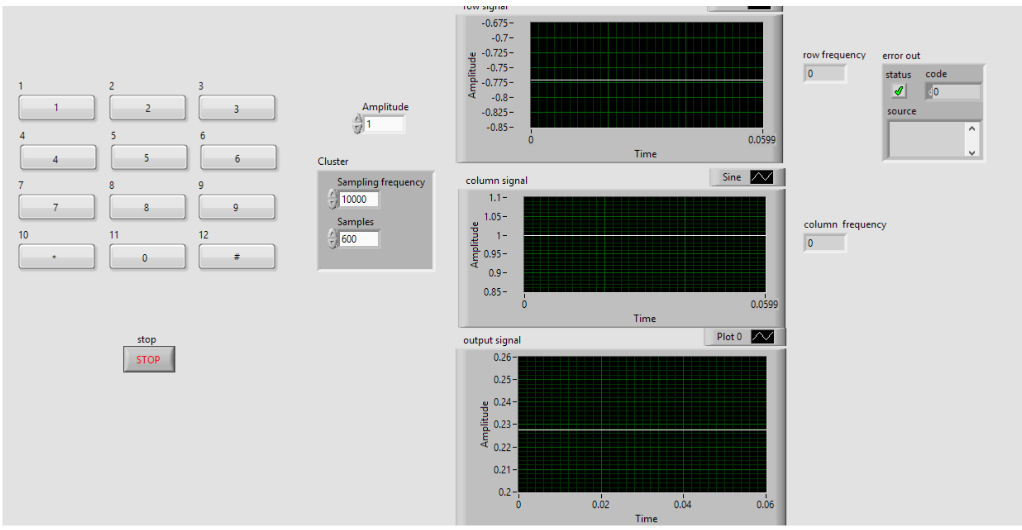


Figure 4.6: DTMF encoder front panel

In this part, a DTMF encoder was created. It could play the sound of the pressed key. The amount of time a sound was played could be adjusted by adjusting the sampling information. In this application, the time for which a sound was played was adjusted to be 30 ms as it's the international standard for DTMF playing time.

Through the 3 waveform graphs, the row signal, column signal and the superimposition of the row signal and the column signal were displayed.

DTMF decoder

The front panel of the DTMF decoder was as below.

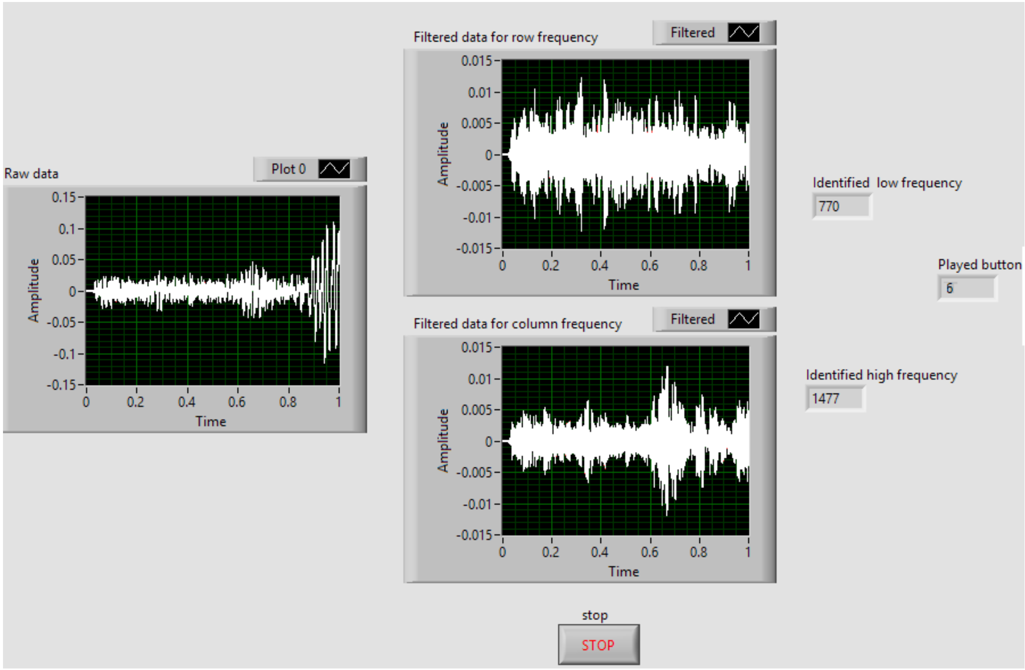


Figure 4.7: DTMF decoder front panel

In this part, a DTMF decoder was created. It filtered the sound obtained through the built in microphone of the laptop and identified the row frequency and the column frequency and thereby, the key pressed.

Although 2 filters of the order 5 were used to filter the noise, didn't work as expected all the time. When the background noise was high, it showed erroneous results.

4.2 Interfacing the DAQ card

4.2.1 Creating a function generator

The front panel of the function generator was as below.

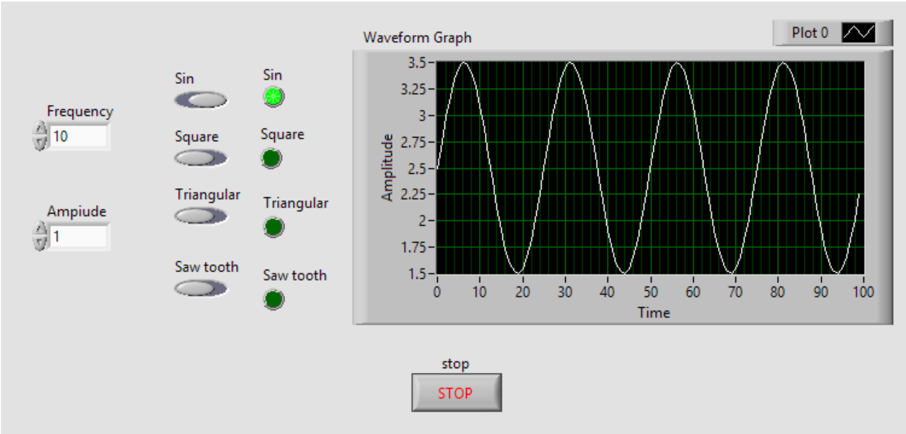


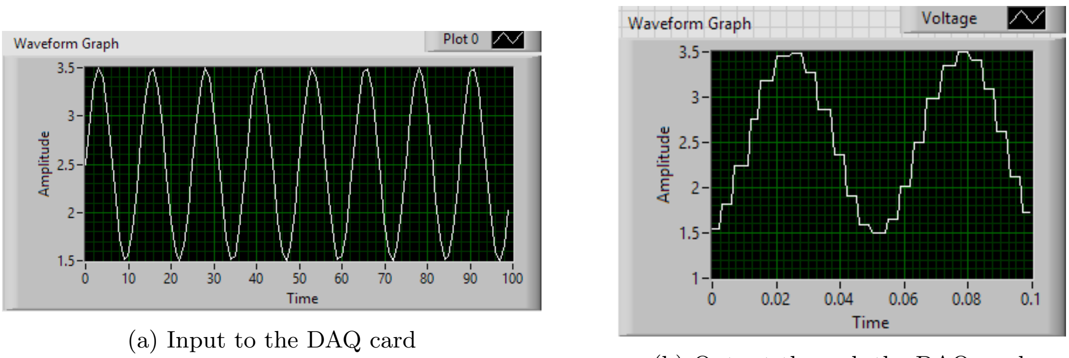
Figure 4.8: Function generator front panel

In this experiment, a function generator was created. In this process, the output of the signal was obtained from the AO0 pin of the DAQ card. The drawback in this application was that signals of frequencies higher than 500 Hz could not be produced. This was due to the fact that the sampling frequency of the DAQ card was set at 1000 Hz. Although the maximum sampling frequency of the DAQ card was at 10 kHz, at that frequency, the signals produced were incorrect in the sense that the frequencies and the type of the signal (sinusoidal, triangular, square or saw tooth) were very much different from the intended ones.

In the sampling frequency of 1 kHz, to get the correct output signal, the input frequency had to be multiplied by 4 to get the intended output frequency. This correction factor was found by trial and error.

The output signal resembled more like a digital signal with clear steps instead of showing an analog signal. The main reason for this behaviour was that the number of samples were set at 100.

Another drawback of this application was that only 1 type of signal could be generated at a given instance. This was due to the fact that superimposition of 2 or more signals in the same phase created a signal with an amplitude greater than 5 V. The maximum voltage that can be output from the DAQ card is 5 V. So, it is a physical limitation of the DAQ card.



(a) Input to the DAQ card

(b) Output through the DAQ card

Figure 4.9: The effect of the number of samples on the shape of the signal

4.2.2 Four channel oscilloscope

The front panel of the four channel oscilloscope was as below.

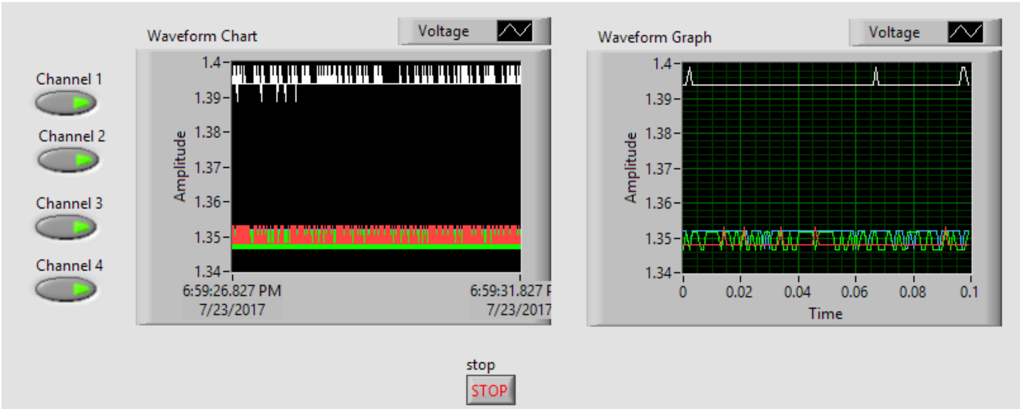


Figure 4.10: Four channel oscilloscope front panel

In this experiment, a four channel oscilloscope was created. Both a waveform chart and a waveform graph were used in showing the outputs. The main difference between the 2 graphs was the data acquisition method. In the waveform graph, evenly sampled measurements were displayed while in the waveform chart, data was typically acquired at a constant rate. Simply said, the waveform chart displayed a larger span of the signal on its display while the waveform graph dispalyed a much smaller sapn of the signal. By using both the plots, one could get a clear idea as to how the signal has varied with time and how the signal behaves presently.

As 4 colours were used for the 4 channels, the signals could be identified clearly and could be used as a simple oscilloscope where finding a digital signal oscilloscope (DSO) is impossible.

4.2.3 Resistance meter

The front panel of the resistance meter was as below.

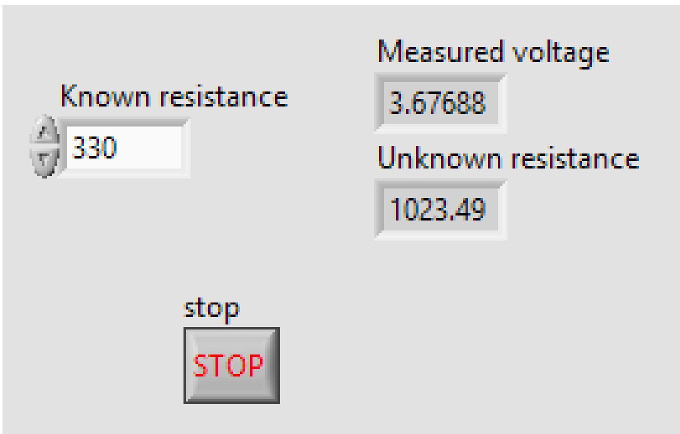


Figure 4.11: Resistance meter front panel

As shown above, when a resistor of 1.1 kΩ was used as the unknown resistor with the known resistor being 330 Ω, the calculated resistance value was 1023.49 Ω. So, it's evident that this resistance meter can calculate resistance values to about 8 % accuracy. Although the accuracy is not as high as a multimeter resistance reading, it can be quite useful in finding a rough estimate for an unknown resistance.

4.2.4 Generation of the VI curve of a diode

The front panel of the application used for generating the VI curve of a diode was as below.

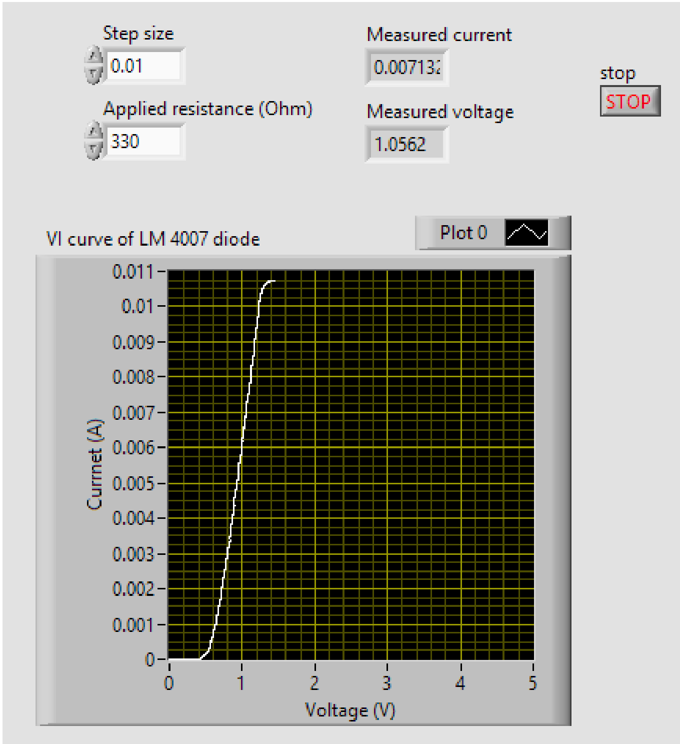


Figure 4.12: Generation of the VI curve of a diode front panel

In this experiment, by adjusting the step size, curves with higher accuracy could be obtained as the increments by which the input voltage vary equals to the step size. As a control for the resistance was added, the diode curve could be obtained using any needed resistor. The instantaneous voltage and the current readings were also displayed in order to get more details regarding the current voltage and current through the diode.

4.2.5 Capacitance meter

The front panel of the capacitance meter was as below.

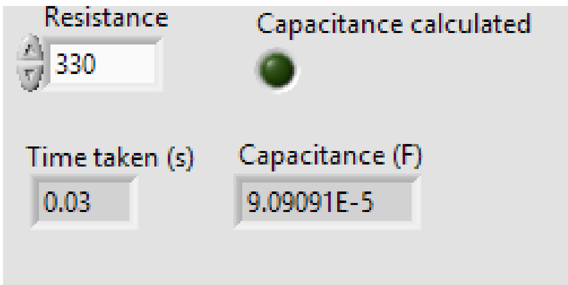


Figure 4.13: Capacitance meter front panel

The capacitance was displayed in Farads. The boolean switch named **capacitance calculated** was to indicate that the process has finished and the capacitance value has been calculated.

The resistance of the series resistor connected to the capacitor could be controlled by the resistance control in the front panel. It allowed choosing varoius resistance values for the needed circuit without limiting into 1 resistance value.

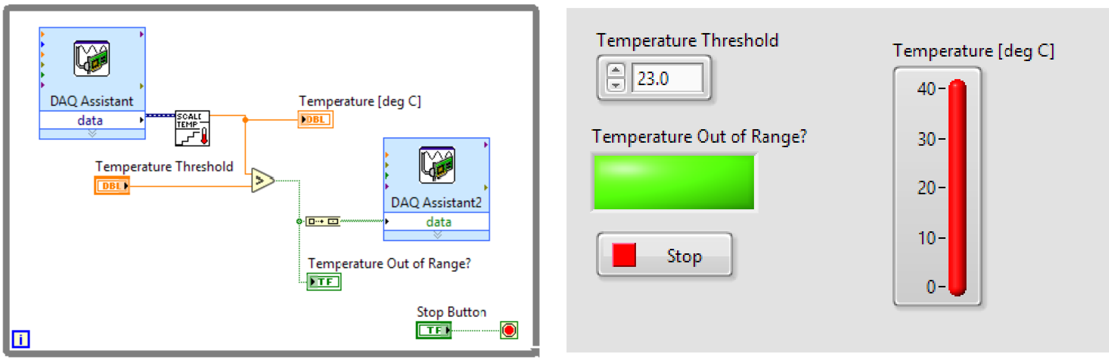
As the time calculation was done in ms, the resistance had to be chosen such that the time taken for the capacitor to charge to the predefined level (3.15 V) was in the scale of ms (milliseconds). So, a small resistor was chosen. (330 Ω).

The limitation of this application was that it could not be used to find a completely unknown capacitance. A rough idea about the capacitance is of utmost importance so that the resistor could be chosen such that the time constant τ is in the order of ms (milliseconds).

5 Discussion

In the first experiment, a simple temperature converter which could be used to change the scale of the needed temperature was created. In fact, this is just a preliminary step of a fully fledged temperature control system able to monitor the ambient temperature and take some actions based on some controlling systems. In fact, this application could be modified to control the AC operating in a closed room to keep the temperature constant at any desired level.

The following figure shows a simple temperature monitoring system developed using LabVIEW and the DAQ card. In here, a thermocouple attached to the DAQ card provides a voltage proportional to the temperature.



(a) Block diagram

(b) Front panel

Figure 5.1: A simple LabVIEW application capable of monitoring temperature

The melody player created in the third experiment could be used as a simple DJ application. By adding more features like superimposing the same melody at a higher pitch to create harmonics and adding distortions, a fairly advanced DJ system based on LabVIEW could be made. By integrating with Arduino, , a hardware controlled interface could be created instead of the instead of the software controlled interface in LabVIEW.

When using the DAQ card, it was important to ensure that the DAQ card worked without any hitch. For this process, the device monitor which opens automatically when the DAQ card is plugged to the computer was used. In this process, the test panels were used to test the input and output functionalities of the DAQ card.

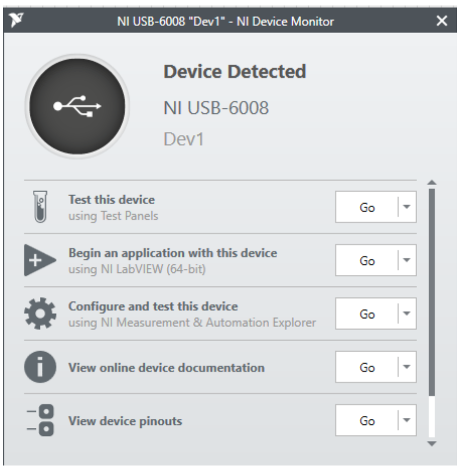


Figure 5.2: NI USB-6008 device monitor

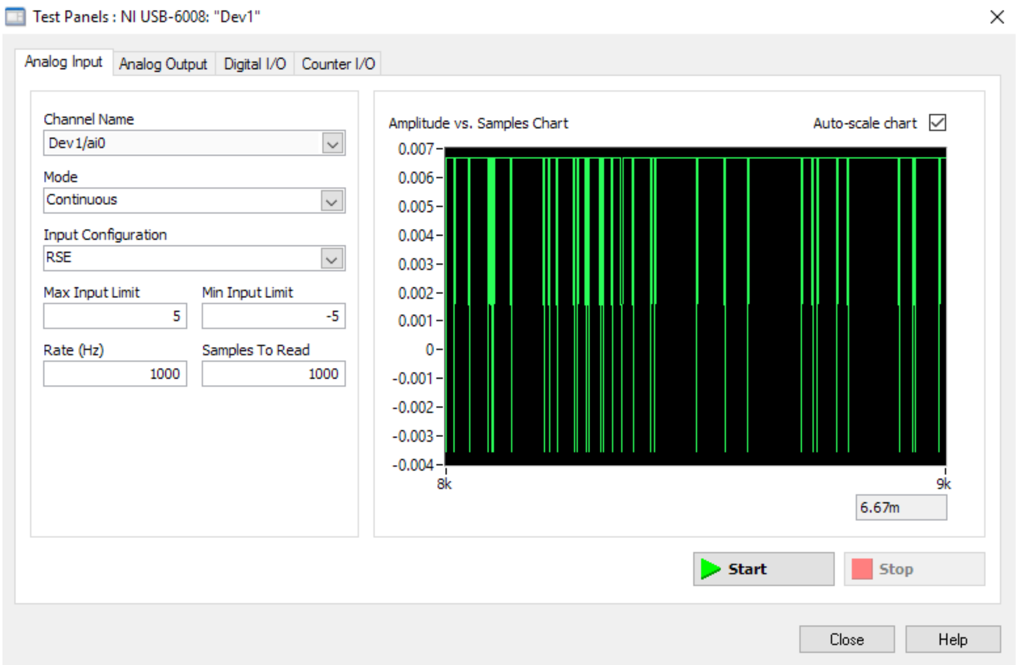


Figure 5.3: Test panel of the NI USB-6008 DAQ card

The resistance meter, the capacitance meter and the LabVIEW's ability to measure the voltage could be added together to form a digital multimeter. It would be much more advantageous as any of the above 3 measurement could be taken with the touch of a button.

The low sampling frequency (10 kHz) of the USB-6008 DAQ card was a limiting factor in practical use. So, this DAQ card could not be used in high speed applications where the need to sample signals at a much higher frequency arises. As predicted by the Nyquist theorem, the DAQ card can sample signals having frequencies below 5 kHz. Otherwise, various problems like aliasing, data loss will occur due to the under sampling.

6 Conclusion

In this practical, application of LabVIEW software in data acquisition and data manipulation was studied. Theoretical and practical knowledge needed in using LabVIEW in data acquisition through the DAQ card were acquired through this practical. Also the theoretical knowledge and practical limitations of digital data acquisition systems were practically understood. So, all in all, it can be concluded that through this practical, all the basic knowledge needed in the use of LabVIEW combined with a DAQ card in any needed application was achieved practically.

7 References

[1] "Data Acquisition (DAQ) - National Instruments", *Ni.com,* , 2017. [Online]. Available: `http://www.ni.com/data-acquisition/`. [Accessed: 01- Jul- 2017].

[2] "DTMF: Dual Tone Multi Frequency", *Engineersgarage.com,* 2017. [Online]. Available: `https://www.engineersgarage.com/tutorials/dtmf-dual-tone-multiple-frequency`. [Accessed: 02- Jul- 2017].

[3] D. Hodge, "Timing Is Everything - A Guide To Reading Musical Notation - Part Two - Guitar Noise", *Guitar Noise,*,2017. [Online]. Available: `http://www.guitarnoise.com/lessons/how-to-read-musical-notation-part-2/`. [Accessed: 05- Jul- 2017].

[4] "Diya Goda Sama Thana guitar tab Archives - Sinhala Guitar Lessons with Tabs and Chords", *Sinhala Guitar Lessons with Tabs and Chords,* 2017. [Online]. Available: `http://guitar.lk/tag/diya-goda-sama-thana-guitar-tab/`. [Accessed: 01- Jul- 2017].

[5] "RC Charging Circuit Tutorial & RC Time Constant", *Basic Electronics Tutorials,* 2017. [Online]. Available: `http://www.electronics-tutorials.ws/rc/rc_1.html`. [Accessed: 01- Jul- 2017].

Appendices

Appendix A - Diya goda sema thena guitar tab