Analysing Signal Frequencies

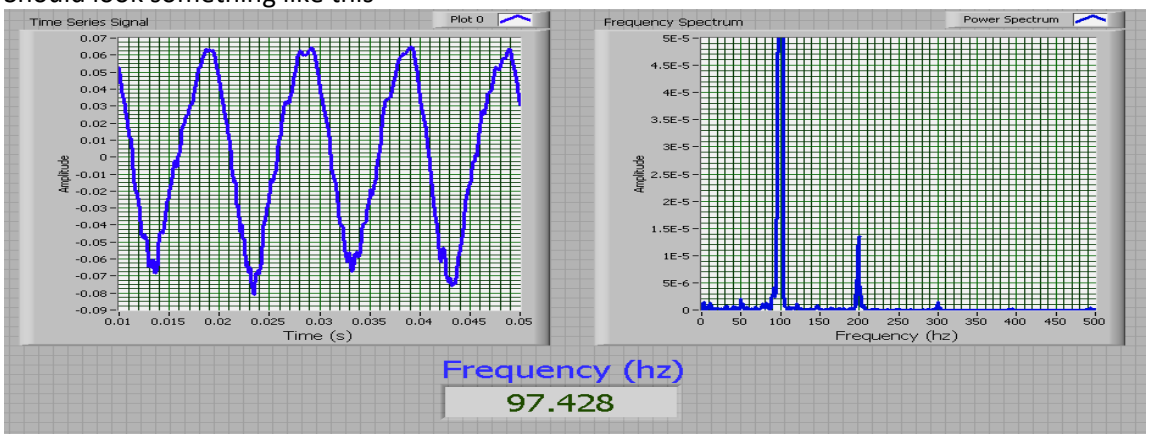
In this experiment we'll look in detail at the frequency content of our signals. We'll discover that for any time varying signal, there are two ways of looking at it; the time domain and the frequency domain. Both can carry significant information.
We're going to build a sound recorder that captures the signal from a microphone and displays it on screen. It then does a spectrum analysis of the signal showing us the frequency content of the sound. We'll be able to see how rich the sound is, how strong the harmonics are, or how spectrally pure it is. We'll analyse the spectrum to pick out the strongest frequency.

Physical Circuit

Just use the microphone provided. Do this before making the LabVIEW circuit so that LabVIEW can detect the device automatically.

The Front Panel

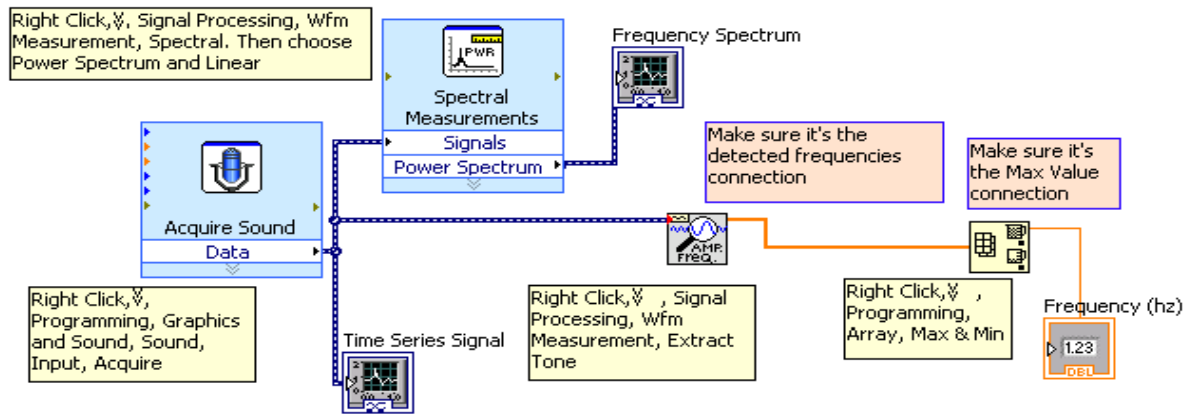
Should look something like this



The two spectra come from **GRAPH** then **WAVEFORM GRAPH**. To adapt the appearance of the spectra (getting the correct labels on the axes etc) we should right click on it and then choose ***properties***. Under the ***scales*** tab you can change the labels on the x and y axes. You should also change the ranges on the x axes of both graphs to get the best appearance (knocking off the *Autoscale X* by right clicking on the graph and choosing x scale is also good).

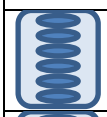
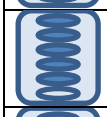
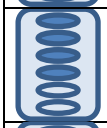
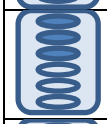
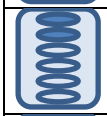
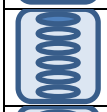
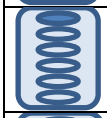
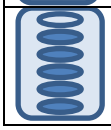
Block Diagram

Should look something like this (the yellow and pink boxes are just for your information):



Experiment

1. Hum a note and press the Run button (). Set the x-axis of the time spectrum to a small range to make sure it is a good sine wave. Use the time scale on the horizontal axis to estimate the period for the wave. Calculate the frequency from f = 1/T. Draw up a table including the frequency from the above analysis, the frequency from the numerical indicator, and the frequency from the maximum of the frequency spectrum
2. Note that the frequency analysis reveals that the sound contains more than one frequency, it also contains higher harmonics. These are the smaller bumps in the frequency spectrum above. Measure the frequencies of these harmonics. Verify that they are integer multiples of the fundamental frequency.
3. Play an octave scale into the microphone. Measure and take note of the fundamental frequency for each note. Each note should have a frequency 1.059 (or 1.059^2 if there is a sharp between them) times the frequency of the preceding note. Over the entire octave the frequency should double. Fill out the table below:

	Note Number	Note	Textbook Freq. (hz)	Measured Freq. (hz)	Log_{10} (f)
	0	D	587.33		
	2	E	659.26		
	4	F#	739.99		
	5	G	783.99		
	7	A	880		
	9	B	987.77		
	11	C#	1108.73		
	12	D	1174.66		

Write-Up

Your write-up should include:

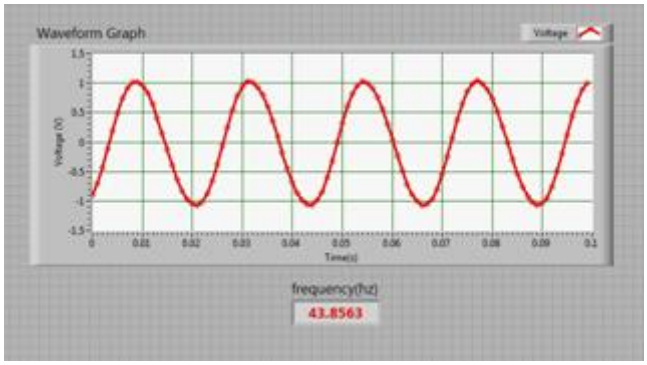
- Printouts of your front panel and block diagram, preferably showing a nice spectrum with clean harmonics.
- The two sets of results of the three frequencies from the time graph, the frequency spectrum, and the fundamental frequency box. These should all be similar. Present these in a table.
- For part 3, a graph of log frequency vs note number. Explain the slope of the straight line
- Submit your write-up in softcopy to the 01 Frequencies link on brightspace within 7 days.

Capturing pan AC Signal

A key feature of computer measurement systems is the conversion of the signal from analogue to digital format. This process includes sampling, and the sampling rate is critical in order to achieve correct conversion. We're going to investigate this by creating a new instrument. In previous experiments we have made thermometers and a sound recording device, this week we'll make an oscilloscope.

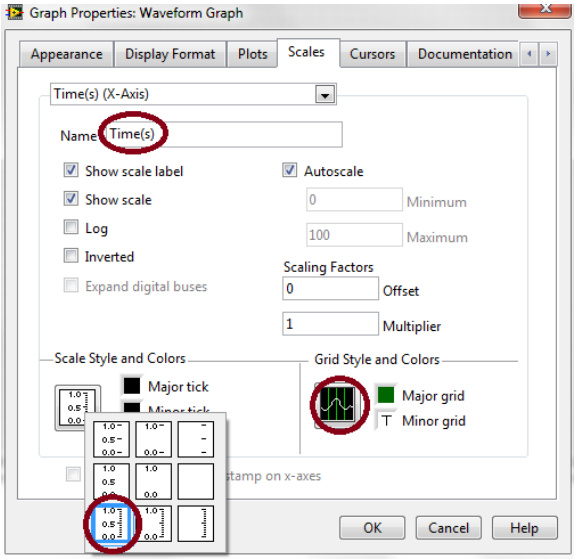
The Front Panel

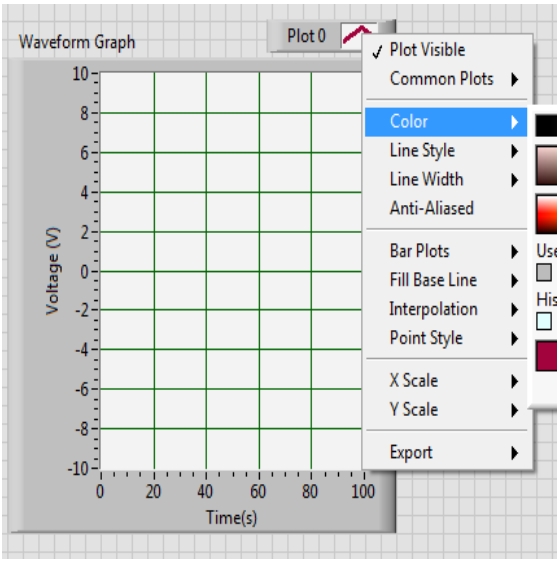
Should look something like this



The waveform graph comes from **GRAPH**. The **detected frequency** is a **NUMERIC INDICATOR**. The other boxes are **NUMERIC CONTROLS**.

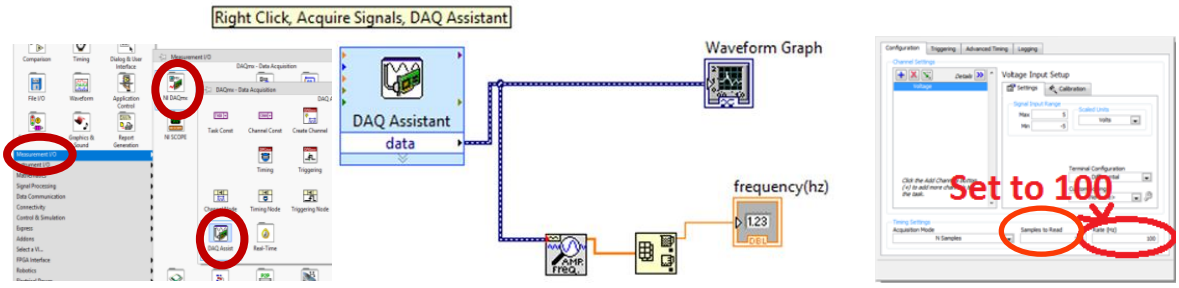
To adapt the appearance of the waveform chart (getting the correct labels on the axes etc) we should right click on it and then choose ***properties***. Under the ***scales*** tab you can change the labels on the x and y axes. You should also adapt the presentation of the axes.





Block Diagram

Should look something like this:

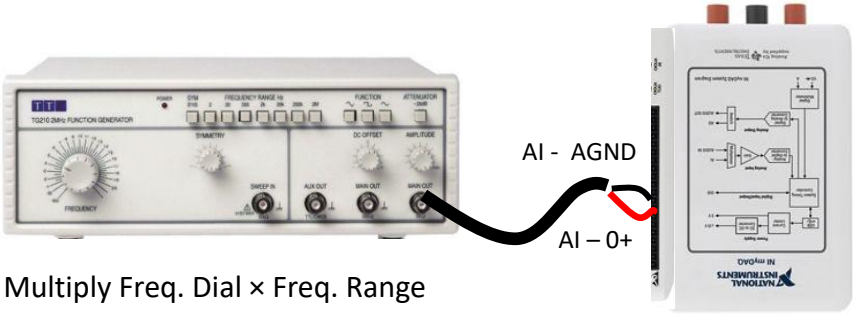


When you first pop the **DAQ Assistant** in place it will run you through a configuration procedure automatically. Choose *Acquire Signal, Analogue Input*,*Voltage, ai0*. Set the sampling rate to be 100 (not 1k)

The **Amp Freq** and the **Array Max&Min** are features we have used before.

Physical Circuit

Connect a coax cable from the function generator to the connector box as shown below:



Multiply Freq. Dial × Freq. Range

Experiment

We want to investigate the Nyquist criterion for sampling waveforms. This states that to correctly measure the frequency of a waveform you must sample at least twice frequency of the waveform. We're going to fix the sampling rate in LabVIEW, change the input frequency from the signal generator, and see what happens to the measured frequency as we scan across the Nyquist level.

- We have set the Sampling Rate to 100 samples/s. By Nyquist this suggests that the highest frequency we can measure is 100/2 = 50Hz.
- Also, we have set the number of samples to 100 (so each run will take 1s).
- On the signal generator set the input frequency to 10Hz.
- Run the programme using the *Run* button ( ), note not *Run Continuously*.
- Record the detected frequency.
- Repeat this with frequencies of 20Hz, 30Hz….120Hz, from the signal generator.
- Plot a graph with the signal generator frequency on the x axis and the detected frequency on the y axis.
- Explain your results using the Nyquist theorem and the idea of aliasing.

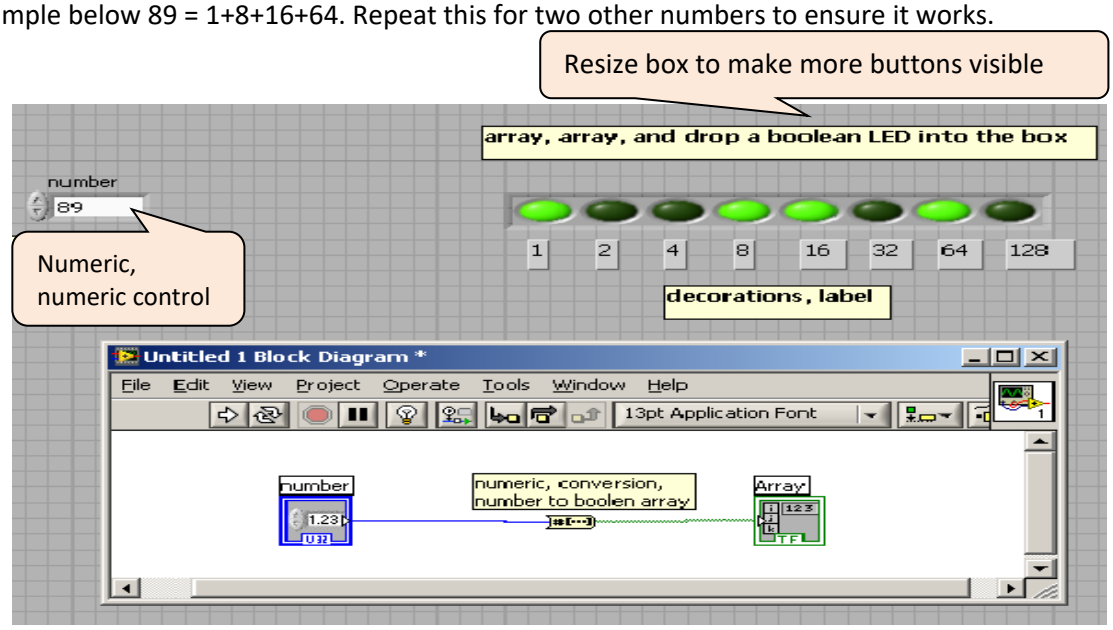
Write-Up

Your write-up should include:

- Printouts of your front panel and block diagram
- Data table with input and measured frequencies
- A graph of input frequency versus detected frequency
- An explanation of your results using the Nyquist theorem and aliasing
- Submit your write-up in softcopy to the 02 Nyquist link on brightspace within 7 days.

1. Binary and Digital Numbers

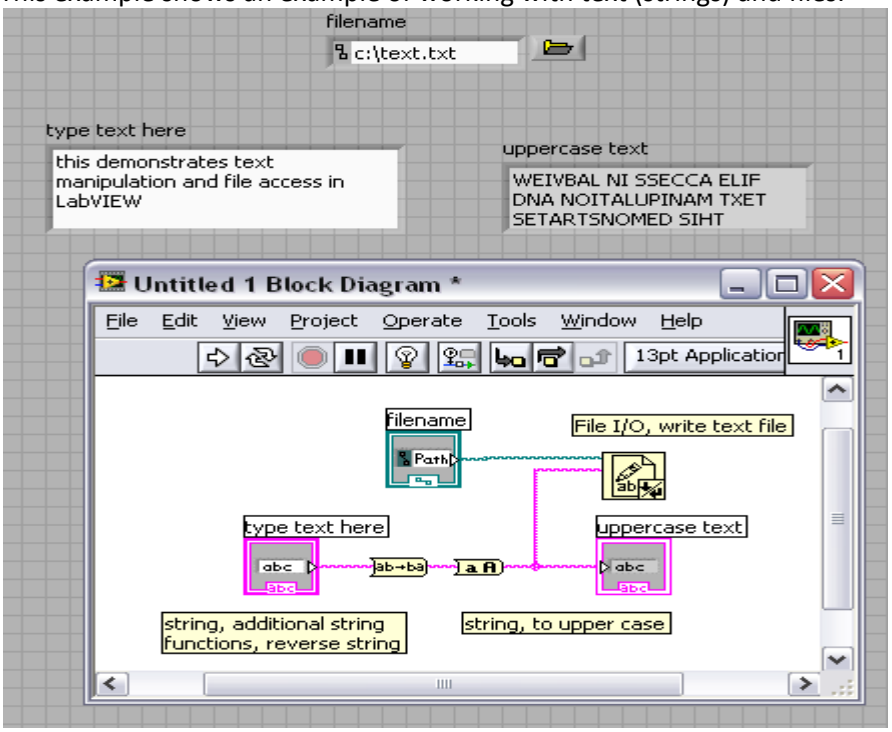
The programme below converts a digital number into a binary number. For example, note that in the example below 89 = 1+8+16+64. Repeat this for two other numbers to ensure it works.



Print out your front panel with an example of one number being converted from digital to binary.

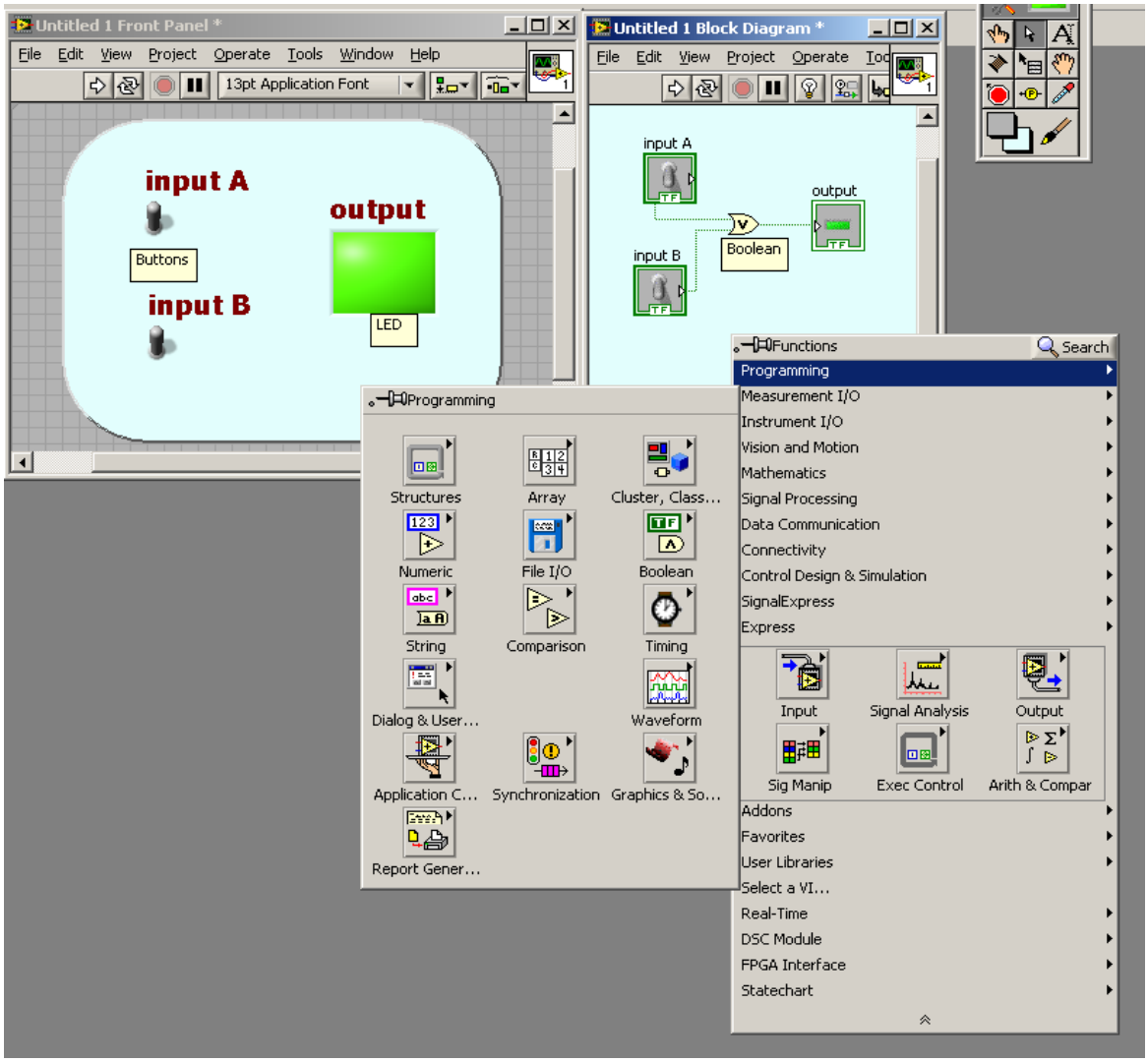
2. Text and File Control

This example shows an example of working with text (strings) and files.



3. Logic Operators

There are a series of Boolean logic operators in LabVIEW. Boolean means an on/off or true/false variable. The operators give a Boolean output based on one or more Boolean inputs. An OR circuit is shown below.



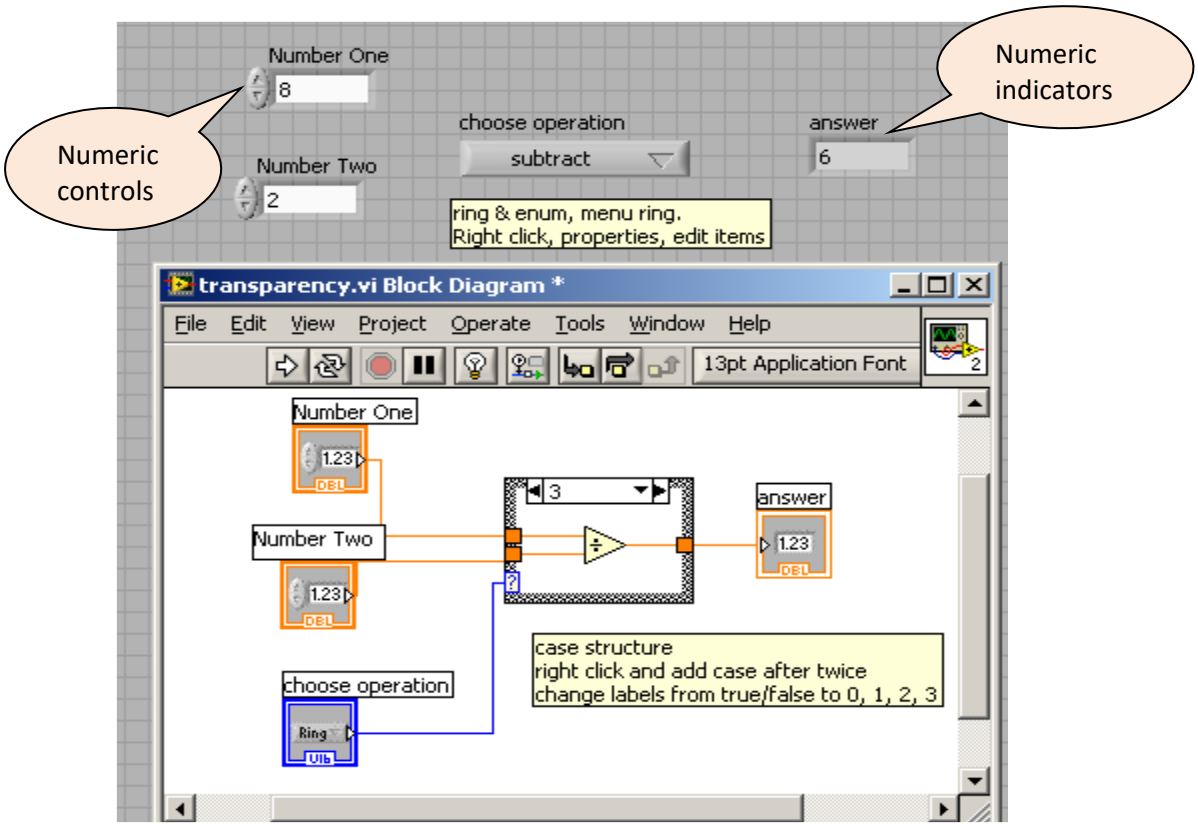
Do not just repeat the programme shown above, rather create a similar programme for a NOT_AND gate and investigate its truth table. A truth table show you the output for every possible combination of inputs. The truth table for the OR is shown below.

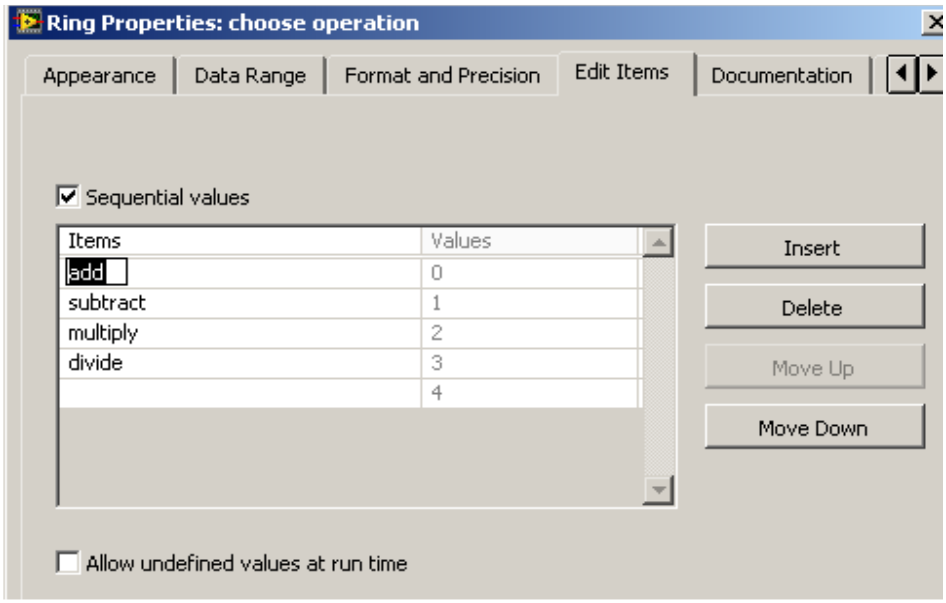
Input A	*Input B*	*ouput*
Off	Off	Off
Off	On	On
On	Off	On
On	On	On

For your truth table, the first two columns will be as above, the last column you have to figure out by building and testing the circuit.

4. Loops and Selection Control

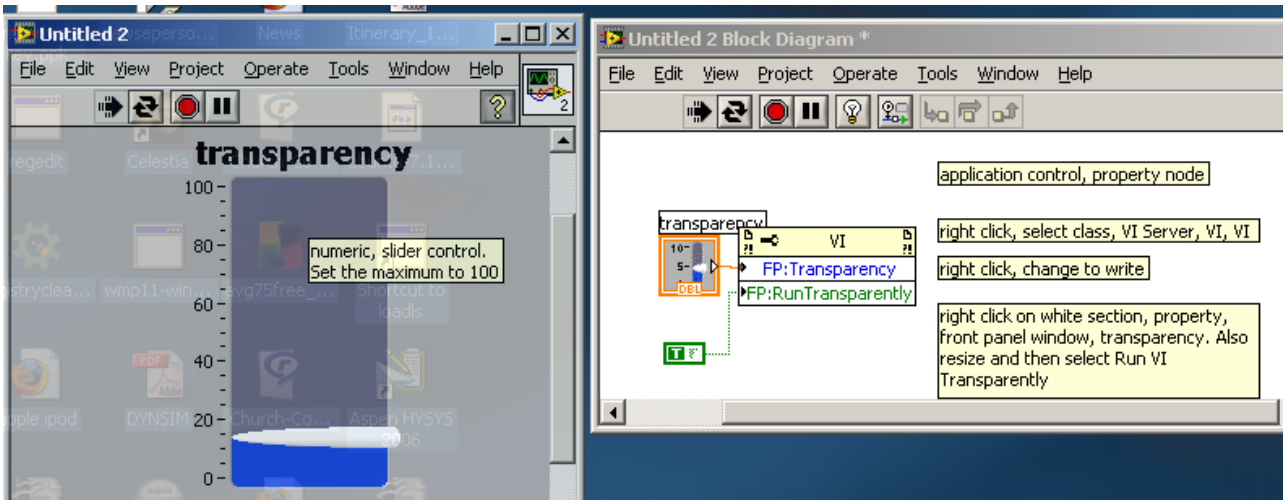
We'll look at a case structure, an item that can choose any one from a menu of items. In this case we will give it two numbers and we'll choose from a menu to add, subtract, multiply, or divide them. Note, the case structure consists of four different options (the fourth one is shown below). Each of the four have to be set up separately, i.e. there is one case structure but it consists of four different "cards" that you toggle between with the arrows.





5. Controlling Appearance of Front Panel

We want to influence the way we present our front panel. We can resize, minimise, and move it around from within the LabVIEW programme itself. The one we'll showcase here is the transparency of the front panel.



As a final exercise, you can adjust the items available on the menu bar in LabVIEW. Reconfigure the VI above so that the "Run Once" button is no longer visible. Do this by clicking on *File*, then *VI Properties* then choosing *Window Appearance* and unclicking the Run button.

6. Filters

We want to investigate the effect of filters on signals, looking at ways to reduce the impact of noise on our signal.

- We will use audio signals that we record.
- We will artificially introduce some noise onto these signals.
- We will set up different filters to try to curtail this noise.
- We will listen back to the sounds and gauge the efficacy of our filter.

Method

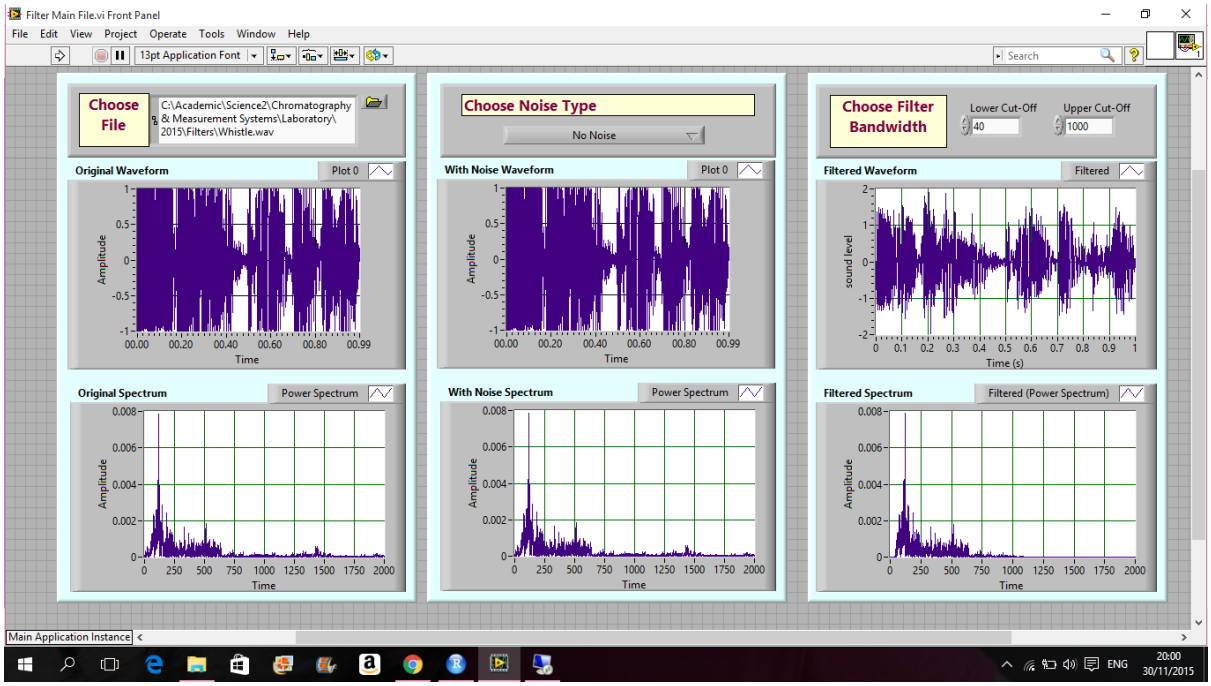
First thing to do is plug in your headset. It can take a while Windows to download and install the necessary drivers, this will happen automatically once the headset is plugged in. Go to the moodle page for this course and download the LabVIEW Filters files. The are seven files in a folder called ***Filter***. You'll need to download all of them. The folder contains:

- Filters Recording, a LabVIEW programme that will enable us to record sounds
- Filter Main File, a LabVIEW programme that will let us add noise, implement filters, and play the sounds back.

In General

Open Filters Recording.
Pick one of the three file paths; song.wav, whistle.wav, or voice.wav
Make the relevant sound into the microphone and at the same time run the programme
This will capture about 1s of audio data. We'll probably need to do this a few times during the experiment.

Open Filter Main File. Pick the file path for the sound you've recorded into the file dialog box. Pick a noise type. Set up the filters with lower and upper frequency limits



(this is a band pass filter; if you set the lower limit to 0Hz you make it a low pass filter, if you set the upper limit to 11024Hz you make it a high pass filter).

Specifically

- Record your voice in the microphone
- Add white noise
- What is the smallest bandwidth you can select and still be able to work out what you are saying?
- Repeat this for pink noise and for 50Hz noise
- For the three sound clip (original, with-noise, filtered), relate what you hear to what you see in the waveforms and spectra, to the type of noise added and to the filter bandwidths chosen.

Write-Up

- Include screen shots of the front panels and block diagrams of parts 1-5.
- Include the binary number calculation from part 1.
- Include the logic truth table from part 3.
- Include your frequency analyses from part 6.
- Submit your write-up in softcopy to the 03 Filters link on brightspace within 7 days