

LabView programming I

Equipment needed: PC with LabView 8 installed. Lecture note for handy reference

預習報告： Please read the lecture handout and find out answers to the following questions. It will be even better if you can get acquainted with the LabView environment before you come in for this laboratory session.

1. What is a Front panel and what is the Block diagram?
2. What is the difference between a For-loop and a While-loop?
3. What is an VI and what is a SubVI?
4. What is a tunnel in the loop structure?
5. What is dataflow programming?
6. What is a Case structure and what is a Sequence structure?

Notes:

1. Please create the following subdirectory in the hard disk: `C:\temp\B94522xxx&B94522xxx`, where `B94522xxx&B94522xxx` are the id numbers of the members in your group. Save your work in this directory for marking.
2. After you finish each experiment, please ask the Teaching Assistant to verify it.
3. In each experiment, follow the step by step guideline to complete your VIs and save them into your directory for later use and grading purpose.

1. Experiment 1 – Convert °C to °F and K.
2. Experiment 2 – Create and make use of a SubVI.
3. Experiment 3 – Number matching and time to match .
4. Experiment 4 – Simulate a data acquisition task.
5. Experiment 5 – Analyzing Logged Data

Experiment 1 – Convert °C to °F and K

Complete the following steps to create a VI that takes a number representing degrees Celsius and converts it to a number representing degrees Fahrenheit and also in Kelvin(absolute temperature).

Front Panel

1. Select **File»New** to open a new front panel.
 2. (Optional) Select **Window»Tile Left and Right** to display the front panel and block diagram side by side.
 3. Create a numeric digital control. You will use this control to enter the value for degrees Centigrade.
 - a. Select the digital control on the **Controls»Numeric Controls** palette. If the **Controls** palette is not visible, right-click an open area on the front panel to display it.
 - b. Move the control to the front panel and click to place the control.
 - c. Type `deg C` inside the label and click outside the label or click the **Enter** button on the toolbar. If you do not type the name immediately, LabVIEW uses a default label. You can edit a label at any time by using the Labeling tool.
 4. Create a numeric digital indicator. You will use this indicator to display the value for degrees Fahrenheit.
 - a. Select the digital indicator on the **Controls»Numeric Indicators** palette.
 - b. Move the indicator to the front panel and click to place the indicator.
 - c. Type `deg F` inside the label and click outside the label or click the **Enter** button.
- Your front panel should now look something like:



Carry on to add a further display for the absolute temperature.

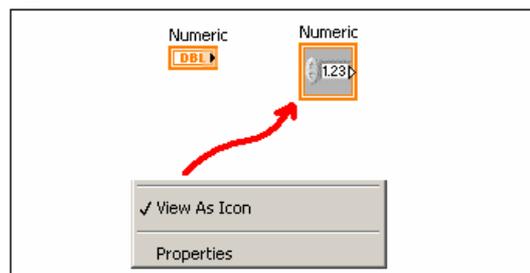
LabVIEW creates corresponding control and indicator terminals on the block diagram. The terminals represent the data type of the control or indicator. For example, a DBL terminal represents a double-precision, floating-point numeric control or indicator.

Note Control terminals have a thicker border than indicator terminals.

Block Diagram

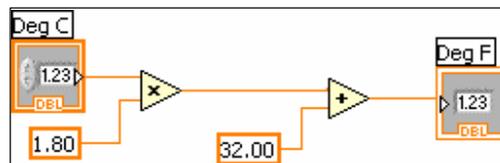
1. Display the block diagram by clicking it or by selecting **Window»Show Diagram**.

Note: Block Diagram terminals can be viewed as icons or as terminals. To change the way LabVIEW displays these objects right click on a terminal and select **View As Icon**.



2. Select the Multiply and Add functions on the **Functions»Numeric** palette and place them on the block diagram. If the **Functions** palette is not visible, right-click an open area on the block diagram to display it.
3. Select the numeric constant on the **Functions»Numeric** palette and place two of them on the block diagram. When you first place the numeric constant, it is highlighted so you can type a value.
4. Type 1.8 in one constant and 32.0 in the other.
If you moved the constants before you typed a value, use the Labeling tool to enter the values.
5. Use the Wiring tool to wire the icons as shown in the previous block diagram.
 - To wire from one terminal to another, use the Wiring tool to click the first terminal, move the tool to the second terminal, and click the second terminal, as shown in the following illustration. You can start wiring at either terminal.
 - You can bend a wire by clicking to tack the wire down and moving the cursor in a perpendicular direction. Press the spacebar to toggle the wire direction.
 - To identify terminals on the nodes, right-click the Multiply and Add functions and select **Visible Items»Terminals** from the shortcut menu to display the connector pane. Return to the icons after wiring by right-clicking the functions and selecting **Visible Items»Terminals** from the shortcut menu to remove the checkmark.
 - When you move the Wiring tool over a terminal, the terminal area blinks, indicating that clicking will connect the wire to that terminal and a tip strip appears, listing the name of the terminal.
 - To cancel a wire you started, press the <Esc> key, right-click, or click the source terminal.
 - From time to time, back up your work by saving the VI in **C:\temp\Bxxxxxx&Bxxxxxx**

When you finish the degC to degF part, your block diagram should look somewhat like:



Go on with the absolute temperature part.

6. Enter a number in the digital control and run the VI.
 - d. Use the Operating tool or the Labeling tool to double-click the digital control and type a new number.
 - e. Click the **Run** button to run the VI.
 - f. Try several different numbers and run the VI again.
7. When completed, save your VI and name it “temp convert.vi” for later use.
8. Did you notice that your conversion run only once after you press the run button. Try to use the loop structure make it run continuously so that whenever your input degC is changed, the outputs follow. You can also try to play around with different indicators and controls (dials, slider etc.).
9. Save your final version as “C to F and K conversion.vi”

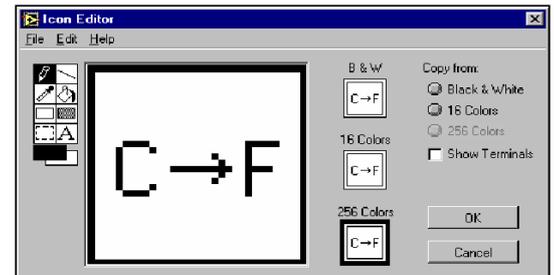
End of Exercise

Experiment 2 – Convert your VI in experiment 1 into a SubVI and make use of it

Front Panel

1. Open your “temp convert.vi” and select front panel.
2. Right-click the icon in the upper right corner of the front panel and select **Edit Icon** from the shortcut menu. The **Icon Editor** dialog box appears.
3. Double-click the Select tool on the left side of the **Icon Editor** dialog box to select the default icon.
4. Press the <Delete> key to remove the default icon.
5. Double-click the Rectangle tool to redraw the border.
6. Create your own icon. An example is as follows:

- a. Use the Text tool to click the editing area.
- b. Type C and F.
- c. Double-click the Text tool and change the font to **Small Fonts**.
- d. Use the Pencil tool to create the arrow.



Note To draw horizontal or vertical straight lines, press the <Shift> key while you use the Pencil tool to drag the cursor.

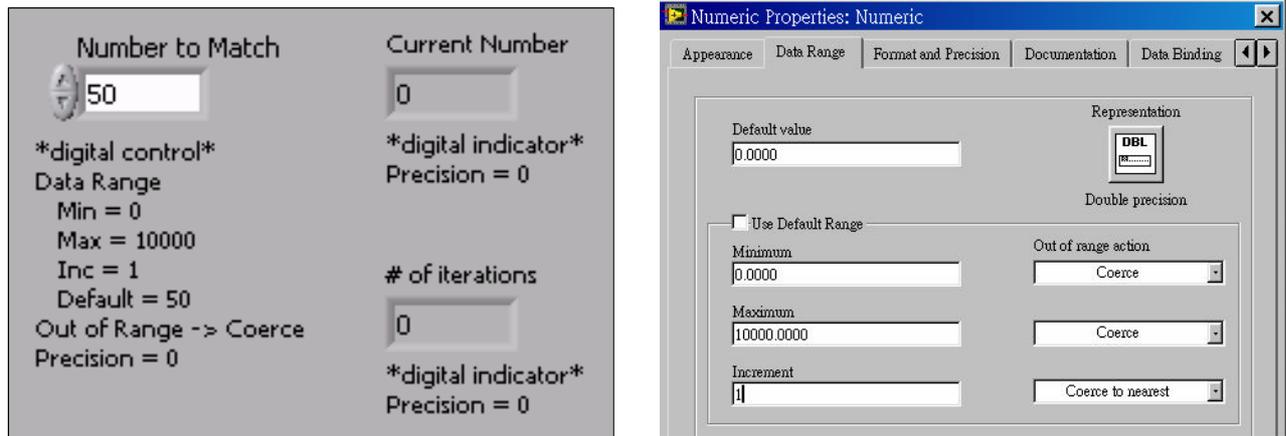
- e. Use the Select tool and the arrow keys to move the text and arrow you created.
 - f. Select the **B&W** icon and select **256 Colors** in the **Copy from** field to create a black and white icon, which LabVIEW uses for printing unless you have a color printer.
 - g. When the icon is complete, click the **OK** button to close the **Icon Editor** dialog box. The icon appears in the upper right corner of the front panel and block diagram.
7. Right-click the icon on the front panel and select **Show Connector** from the shortcut menu to define the connector pane terminal pattern.
LabVIEW selects a connector pane pattern based on the number of controls and indicators on the front panel. For example, this front panel has two terminals, **deg C** and **deg F**, so LabVIEW selects a connector pane pattern with two terminals.
 8. Assign the terminals to the digital control and digital indicator.
 - a. Select **Help>Show Context Help** to display the **Context Help** window. View each connection in the **Context Help** window as you make it.
 - b. Click the left terminal in the connector pane. The tool automatically changes to the Wiring tool, and the terminal turns black.
 - c. Click the **deg C** control. The left terminal turns orange, and a marquee highlights the control.
 - d. Click an open area of the front panel. The marquee disappears, and the terminal changes to the data type color of the control to indicate that you connected the terminal.
 - e. Click the right terminal in the connector pane and click the **deg F** indicator. The right terminal turns orange.
 - f. Click an open area on the front panel. Both terminals are orange.
 - g. Move the cursor over the connector pane. The **Context Help** window shows that both terminals are connected to floating-point values.
 9. Save your VI as “temp convert subvi.vi” for later use.

End of Exercise

Experiment 3 – Number matching and time to match

Complete the following steps to build a VI that generates random numbers until the number generated matches a number you specify. The loop count terminal records the number of iterations before a match occurs.

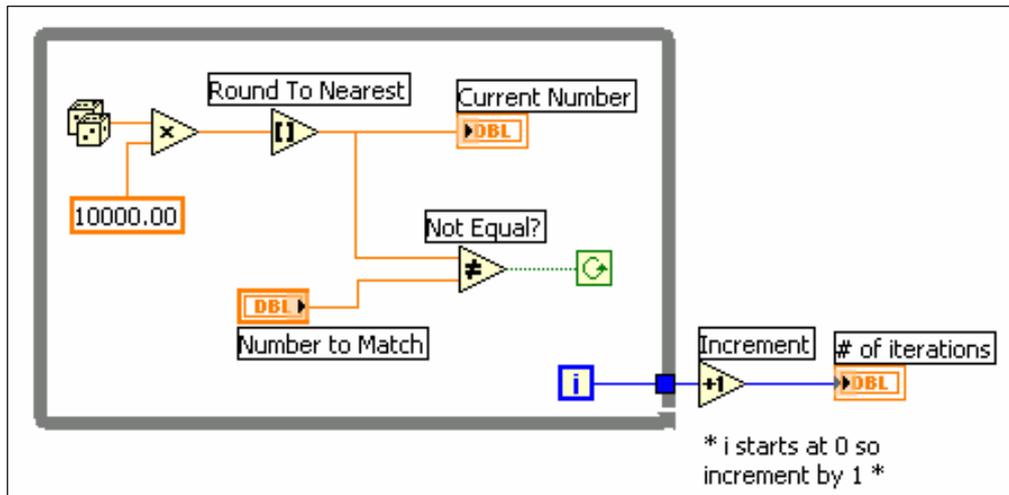
1. Open a new front panel.
2. Build the following front panel and modify the controls and indicators as shown and described in this exercise.



The **Number to Match** control specifies the number you want to match. The **Current Number** indicator displays the current random number. The **# of iterations** indicator displays the number of iterations before a match.

Use the **Data Range** option to prevent the user from selecting a value that is not compatible with a range or increment. You can ignore the error or coerce it to within range. Complete the following steps to set the range between 0 and 10,000 with an increment of 1 and a default value of 50.

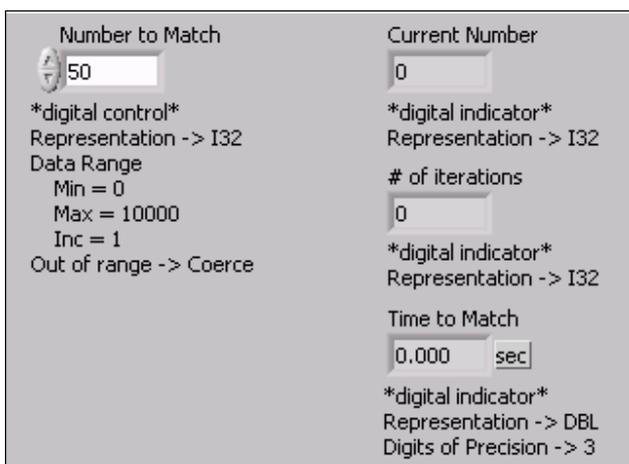
3. Right-click the **Number to Match** control and select **Data Range** from the shortcut menu. The **Data Range** dialog box appears.
4. Remove the checkmark from the **Use Defaults** checkbox, select the options as shown in the following dialog box and click the **OK** button.
5. By default, LabVIEW displays numeric controls and indicators in decimal notation with two decimal places, such as 3.14. Use the **Format & Precision** option to change the precision or to display the numeric controls and indicators in scientific, engineering, or hour/minute/second notation. Complete the following steps to change the precision to 0.
6. Right-click the **Current Number** indicator and select **Format & Precision** from the shortcut menu. The **Format & Precision** dialog box appears.
7. Type 0 in the **Digits of Precision** text box and click the **OK** button.
8. Repeat steps 7 and 8 for the **# of iterations** indicator.
9. Build the following block diagram. Try to locate the corresponding nodes from the relevant palettes. Use context help (Ctrl H) whenever you need explanation on the use of the nodes.
10. Place a While Loop on the comparison part of your program as shown.



The blue square that appears on the While Loop border is called a tunnel. Tunnels feed data into and out of structures. Data pass out of a loop after the loop terminates. When a tunnel passes data into a loop, the loop executes only after data arrive at the tunnel.

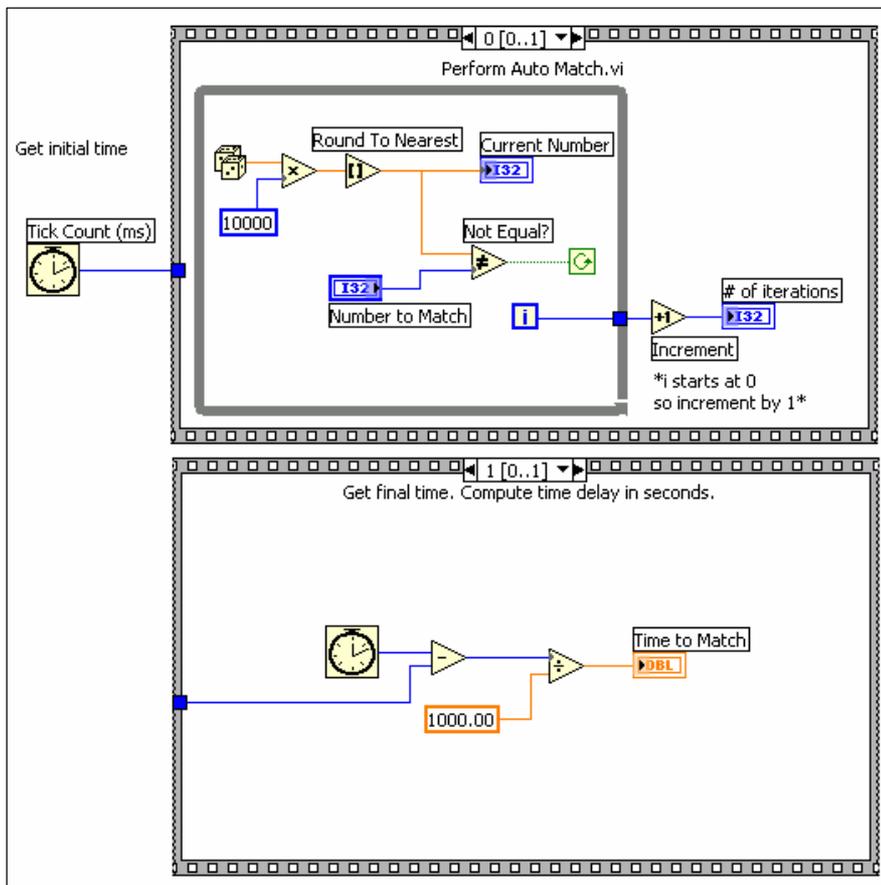
The loop executes while no match exists. That is, the Not Equal? function returns TRUE if the two numbers do not match. Each time the loop executes, the iteration terminal increments by one. The iteration count passes out of the loop upon completion. This value increments by one outside the loop because the count starts at 0.

11. Save the VI as “Match.vi”.
12. Test run your VI by changing the number in **Number to Match**.
13. Modify the front panel as follows.



- a. Change **Number to Match**, **Current Number**, and **# of iterations** to I32 representation.
- b. Change **Time to Match** to DBL representation and 3 digits of precision.

14. Modify the block diagram as follows.



- a. Place a Sequence structure located on the **Functions»Structures** palette. Find out the use of a Sequence structure.
- b. Right-click the structure border and select **Add Frame After** from the shortcut menu to add a frame.
- c. Place the Tick Count (ms) function located on the **Functions»Time&Dialog** palette. This function reads the current value of the operating system clock and returns the value in milliseconds.

6. Display the front panel, test run your VI by entering a number in **Number to Match**, and run the VI.

In frame 0, the VI executes the While Loop while **Current Number** does not match **Number to Match**. In frame 1, the Tick Count (ms) function reads the operating system clock. The VI subtracts the new value from the initial time read and returns the elapsed time in seconds.

7. Save the VI as “Match time.vi”.

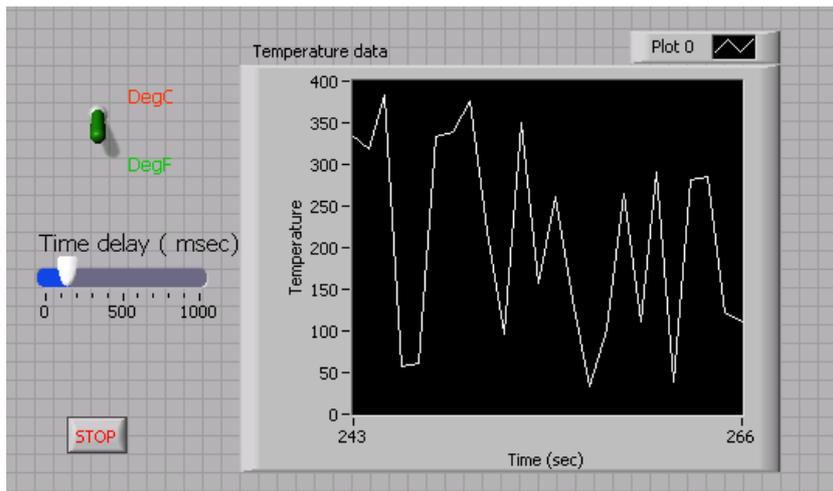
End of Exercise

Experiment 4 – Data Acquisition simulation

In this experiment, we simulate the acquisition of a temperature source by using a random number. We will display the data on a waveform screen in real time. We also logged the data and write it to a file for later use.

Front Panel

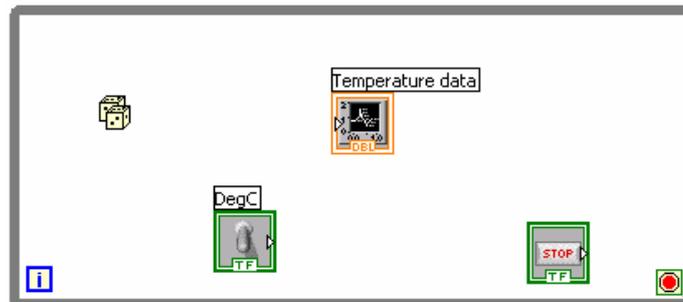
1. Open a new VI.
2. Build the following front panel.



Select a waveform chart on the **Controls» Graph Indicators** palette and place it on the front panel. The waveform chart will display the data in real time. Give it the correct labels and min./max. values for the x and y-axes.

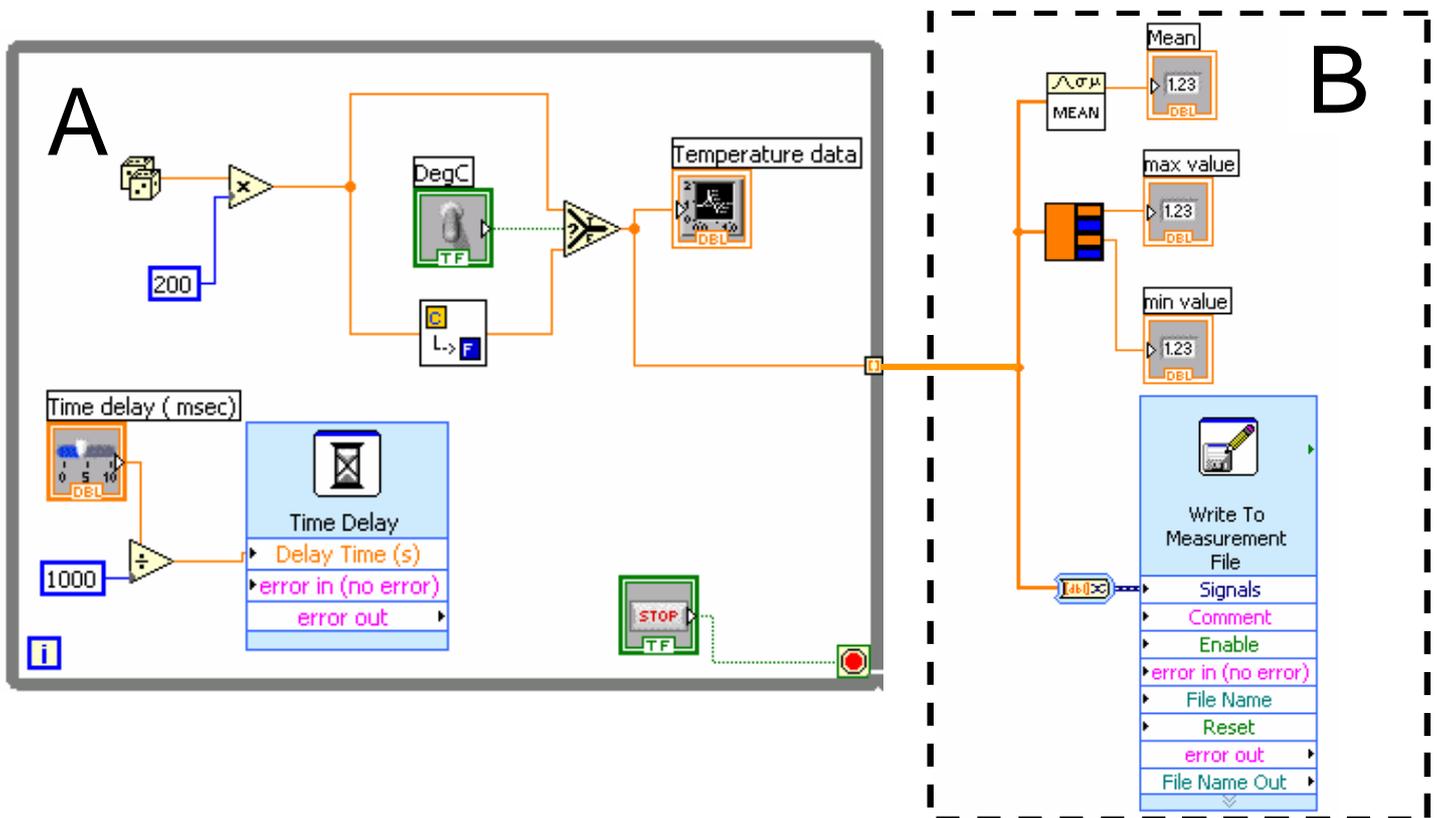
Block Diagram

3. Select **Window»Show Block Diagram** to display the block diagram.
4. Enclose the terminals in a While Loop as follows.



5. Make the necessary connections to let the random number generator generates a temperature in the range 0 to 200 °C.
6. Suppose the temperature generated is in °C. Add some selection logic to let the toggle switch choose whether we want to display °C or °F . You can make use the subVI “temp convert subvi.vi” in experiment 2.
7. Test run your VI at this stage and note the result.
8. Add a time delay that can be controlled on the front panel in unit of millisecond. Try different values for the millisecond delay and run the VI again. Notice how this affects the speed of the number generation and display.
9. Test run your VI again and save it as “data gen.vi”.

10. Now add the part on the right of the above diagram (in broken line box) to display the mean, max and min. value of your data. In the meantime, write the data to a data file.
11. In linking the input of part B to that of part A, you need a tunnel in the while loop. Enable auto-indexing of this tunnel. Auto-indexing allow the signals to be put into successive cells of an array.
12. Note that the mean, max. and min. functions takes on an array. Dataflow in an array is represented in a thicker line.
13. When you add the write to file node, a “convert from Dynamic Data” function will be automatically inserted. Data generated or acquired from instrument will be changing and are known as Dynamic Data, in contrast to the Static data in a numerical constant.
14. Display the front panel and test run your VI
15. After pressing STOP a dialog box will appear. Enter the name of the file to save the spreadsheet. Values will also be displayed in the mean, max and min. box. Record these value for later use.
16. Open the spreadsheet to make sure the file was properly created by using Notepad.
17. Save your VI as “data logging.vi”

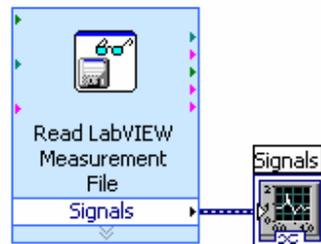


End of Exercise

Experiment 5 – Analyzing Logged Data

Block Diagram

1. Create the following block diagram



2. Place the Read LabVIEW Measurement File Express VI located on the **Functions»Input** palette.
3. Configure the VI to ask the user to choose the file to read and change the delimiter to Tab
4. Right click on the Signals Output and choose create graph indicator
5. Test run the VI
6. Compute the max., min. and mean values from this logged data. Compare them with the values you get in the last experiment.
7. Save the VI as “[Read and analysis.vi](#)”.

End of All Exercises

Announcement

LabView Lab Two will consist of the following:

1. Writing a [Function Generator VI](#). The output will be converted into analogue voltage and displayed on a hardware oscilloscope.
2. Writing an [Oscilloscope VI](#) and use it to display the output from a hardware function generator.
3. Writing a [Mechanical Properties Testing VI](#) to interface with a desktop universal testing machine.

You can try to explore possible functions and techniques for these VIs or even build up part of the VIs first before you come next week.