Virtual instruments and introduction to LabView

(BME-MIT, updated: 26/08/2014 – Tamás Krébesz – krebesz@mit.bme.hu)

Introduction

The purpose of the measurement is to present and apply the concept of virtual instrumentation and to introduce one of its software (SW) platforms, LabVIEW, a data-flow type graphical programming language and integrated development environment.

The students learn the basics of LabVIEW graphical SW development environment, and the using of the sound card (as an analog-to-digital, digital-to-analog converter for the audio frequency band) installed inside the PC. Furthermore some practical problems have to be solved.

During the measurement the tutors act in an interactive manner and so they actively guide and help the solution of the tasks. The students are kindly asked to keep up with the others in solving the tasks and any time-lag should be reported to the leader of the measurement who can help in the solution. Any questions are welcome any time!

References, reading material

[1] BME MIT - LabView basics, 2008.,
Download the pdf from the homepage of the course

[2] LabViewQuickRef.pdf
Download the pdf from the homepage of the course

[3] Web page of National Instruments (international company who developed LabVIEW): www.ni.com where LabVIEW trial version can be downloaded. A lot of material related to electrical engineering can be found from the wide range of technical knowledge base presented there.

Preparing for the measurement

1. Read the material LabView basics [1]!
2. Read and consider the measurement tasks from this document!
3. Respond to the Test questions (which can be found at the end of this document)!
4. Recommended: learning the basics of LabVIEW SW platform by downloading LabVIEW via www.ni.com. (note: a new version of LabVIEW appers almost in every year but the handling of the necessary functions are the same; no matter which version is used. Be careful, backward compatibility is supported but not automatic!)

Instruments and tools

Oscilloscope	Agilent 54622A
PC	NEC TM600
Special SW	LabVIEW a version of
Special supplements	Jack – RCA cable, RCA – BNC converter

Hints for using LabVIEW

- Context Help (shortcut: Ctrl+h) provides information about a LabVIEW graphical element (graphical element is usually referred to as VI – Virtual Instrument in LabVIEW) above which the mouse is positioned. Context Help gives an answer to the question "What is the function of this element?" For the function (Equal?) used to determine equality Context Help looks like this (see Fig. 1):

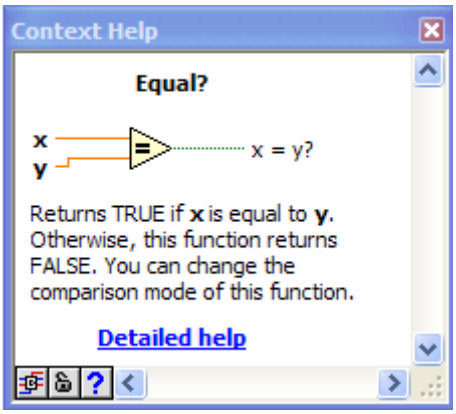

Figure 1. Context Help

- By left-clicking on an element we can move it by drag-and-drop. Fields can be edited by double clicking.
- While editing a VI, the mouse pointer constantly changes depending on where it is pointing. For instance if the cursor of the mouse is pointing at the output of an element, the pointer automatically changes its appearance indicating that wiring is now possible.
- LabVIEW has 2 panels: *Front Panel*, it is the user interface, and *Block diagram Panel* where the graphical code can be written (Fig.2). Fast switching between the two panels: **Ctrl+e**. By right clicking on the *Block diagram Panel*, the *Functions palette* can be reached and the graphical elements can be placed on the panel ~ writing the code. By right clicking on the *Front Panel*, the *Controls palette* can be reached to place input and output fields on the panel ~ creating the user interface.

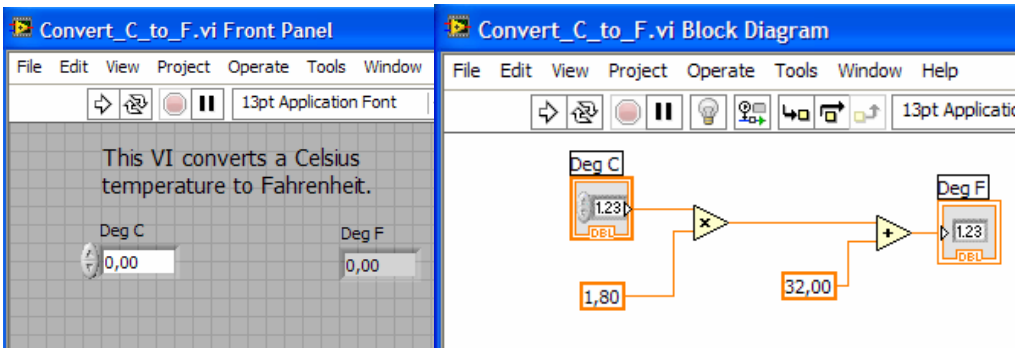

Figure 2. LabVIEW *Front Panel* on the left and *Block Diagram* on the right

- If we right-click on an icon at the Block Diagram, or on an input or an output a useful pop-up menu is invoked. The most useful is the **Create** menu item, clicking on this we can directly insert a *Constant* (only the developer has access to it on the Block diagram Panel), a *Control* (appears on both the Frant and Block diagram Panels) or an *Indicator*

(appears on both the Frant and Block diagram Panels) in the Block Diagram. This is simpler then choosing these from the Functions palette.

- If a wire is not looking "good" in the Block Diagram, or it is not clear where it is connected, we can right-click on it, and choose the **Clean up wire** function. This re-draws the same wire in a clearer, simpler way.
- There is the opportunity to clean up the whole graphical code of the Block Diagram by clicking on the **Clean Up Diagram**  button. This button is located among the elements of the upper graphical buttons of the Block Diagram.
- Not LabVIEW specific issue but it is useful to know: **Alt + print screen** copies only the active window into the clipboard, not the whole screen.

Measurement Tasks

1. Solving practical problems

1.1. Using the Celsius – Fahrenheit converter VI

Start LabVIEW by clicking on Windows Start menu *All Programs->NI LabVIEW 2013 SP1*! Click on *Open Existing* of the starter window of LabVIEW 2013 and choose the "**Convert_C_to_F.vi**" file from the folder: *D:\alaplabor\ml3\01_LabView*.

- Follow the explanation and instructions of the tutor to understand the principles of the LabVIEW SW platform.

1.2. Loops

Open the file "**FOR_Loop.vi**" from the directory *D:\alaplabor\ml3\01_LabView*!

Turn on the Highlight Execution function  by clicking on the light bulb icon! This helps to understand what happens during execution by showing the values, data “traveling” on the wires.

Check and understand the operation!

- Indicate the number of the iteration on the Front panel, then build up a vector whose elements are the numbers of the iterations!
- Replace the For loop with a While loop, and check whether it works correctly or not!

1.3. Building a Fahrenheit-Celsius converter VI

Open a new blank VI! (File menu / New / Blank VI)!

- Build the program performing the inverted function than Celsius – Fahrenheit converter performs, based on section 2.3 of the reading material "LabVIEW basics" [1]!

  Remark: subtract 32 from the value in Fahrenheit, and divide the result by 1.8 to get the final result in Celsius.

2. Virtual function generator

Build a function generator that plots the chosen waveforms on the monitor of the PC!

The main specification of the function generator is the following:

- Sine wave, square wave, and triangle wave can be chosen to plot
- In case of the square wave, the duty cycle can be changed
- The frequency and the amplitude can be changed
- The waveform is plotted on the monitor
- The user interface on the front panel is orderly, and simple to use

Hints: use the **Basic FuncGen** VI from the palette **Functions / Signal Processing / Wfm Generation**! This element already has most of the specified functionality, only the inputs and outputs have to be built around it. (Use Ctrl+h to have Context-help.) It is useful to place all of the elements that generates the function in a While loop, to ensure that the waveform generation is done continuously.

It is possible to visualize the inputs and outputs of a graphical element in an unambiguous way by right clicking on the element then uncheck the **View as icon** in the menu appears. If you do so the icon should look like this (Fig.3):

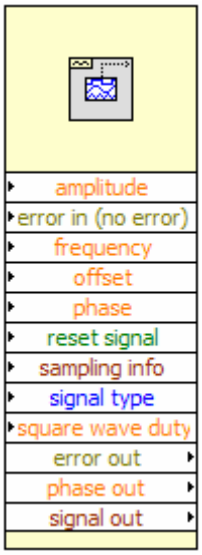

Figure 3. The icon of *Basic FuncGen* when the *View as icon* feature of the VI is unchecked

Remarks: It is important to know the relationship among the signal frequency (f), sampling frequency (Fs) and number of samples (#s). For example: let us take and plot 10 samples from a f=10Hz sine wave using sampling frequency, Fs=100Hz. One entire period can be seen since Fs/f=10 and so 1 period contains 10 samples. Think the situation over carefully!

3. Function generator implemented by a sound card

Extend the functionality of the virtual function generator implemented previously: beside plotting the waveforms on the screen, the waveforms should now appear at the output of the sound card of the PC!

Connect a Jack – RCA cable to the output of the sound card, and using a RCA-BNC converter, connect the cable to an oscilloscope! Check the correct operation and the signal characteristics on the oscilloscope!

Hints: use the **Sound Output Configure**, **Sound Output Write** and **Sound Output Clear** elements from the palette: **Functions** / **Graphics & Sound** / **Sound** / **Output**! Figure 4. helps in accessing the graphical elements. The **Configure** and **Clear** elements should be put outside the While loop, but the **Write** element has to be inside the loop. Make sure that the sampling rate is the same at the **sound format** input of the **Configure** element and at the **sampling info** input of the **Basic Function Generator**!

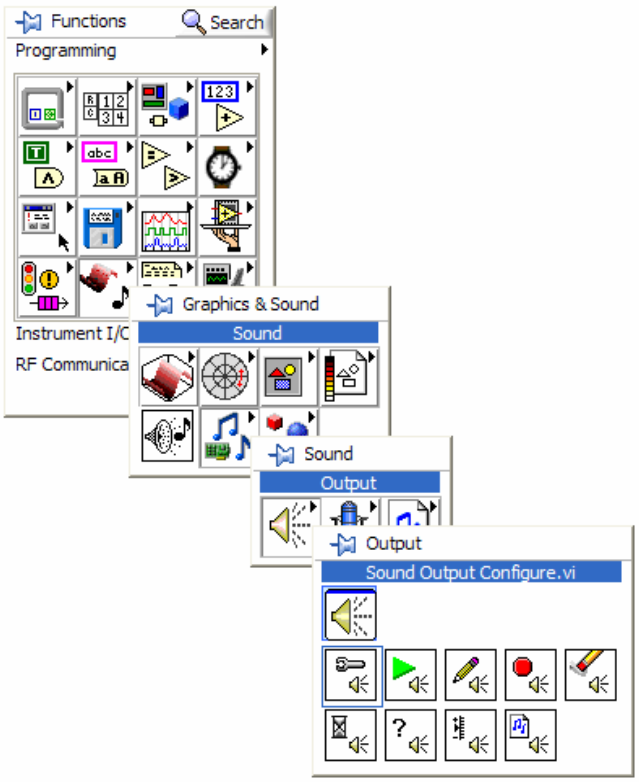

Figure 4. Functions of the *Functions* palette

4. Variable frequency function generator

- Modify the function generator to make the frequency of the output signal change in pre-defined steps. At the Front Panel we introduce new input values: the Start Frequency, the Frequency Step, and the Number of Steps. Each frequency has to be active for 1 second. (E.g. if the Start Frequency is 500 Hz, the Frequency Step is 100 Hz and the Number of Steps is 2, then a 500 Hz sine wave appears at the output for 1 second, and then a 600 Hz sine wave for the next 1 second.)

In order to accomplish this task, use a **Wait** element, which delays execution for a predefined amount of time. The suitable **Wait (ms)**  element can be found here: **Programming/Functions/Timing**. This element has only one input, which is the waiting time in milliseconds. The Wait element should be placed inside the loop and not to be connected to any other elements.

5. Visualization of the input signal

Generate a sine wave having 1 kHz frequency, and 0.5 V amplitude using the **Agilent 33220A** arbitrary function generator, and connect it to the **Line In** input of the PC sound card (**BLUE** connector)!

Before connecting the cable to the sound card, double check the signal using an oscilloscope! The Line In input of the sound card can be driven by 1 V RMS signal at maximum. Make sure that the cable is connected to the Line In input, and not to the Mic In input (**PINK** connector)!

- Plot the incoming signal on the PC screen similarly to an oscilloscope!
- Create an indicator to display the signal amplitude on the Front Panel of LabVIEW!

Test questions

1. What are the two main windows of a LabView Virtual Instrument (VI)?
2. What is the Front Panel? (What kind of objects does it show, who uses the Front Panel, the user or the developer, etc.)
3. What is the Block Diagram? (What kind of objects does it show, who uses the Block Diagram, the user or the developer, etc.)
4. Name 3 different LabView tools that help in debugging!
5. What is the difference between the Functions palette and the Controls palette?
6. What does it mean if the Run icon turns into a broken arrow?
7. What is the difference between FOR and WHILE loops?
8. How would you build an *if-then-else* structure in LabView?
9. Why should the use of *Continuous run* be avoided? Instead how this function should be implemented in a program?