Lab #4
Beginning LabView programming

In this lab you will explore the LabView environment by programming some simple applications read and write data from the front panel, control some logic switches, and build arrays of data. You will also explore making simple Dialog boxes, and retrieving information as to which button was pushed in the Dialog box.

1) Create an Application which reads the current time from the computer, and pops up a simple dialog box with the time information.

   a) Create new LabView application, and go to Diagram, and show the Functions Palette.
   
   b) On the Functions Palette put a copy of the function "Get Date/Time in Seconds" (under "Time and Dialog" in the Functions Palette) into your diagram. Also get a copy of the function "Format Date/Time String" from the same Palette. Connect them up to create a text formatted string with the current time and date.
   
   c) Under 'String' get the string concatenation function. Concatenate the previous text string containing the current time and Date after a new text string constant. The new text string constant should contain the words "Current date and time is: ".
   
   d) Create a Single Button Dialog Box (function is under "Time and Dialog"). Use the concatenated text string from part (1c) as the Dialog Box. Use another text string constant to make a name for the Button of "Thanks".
   
   e) When you run the program, it should pop-up a single dialog box which says "Current date and Time is: XXXXXXXXX" where XXXXXXXX is the current date and time. Clicking on the 'Thanks' button will close the dialog box. Demonstrate the program to the TA.

2) Create an application which reads a text string from the front control panel, sends the text string to a text string display as well as to a dialog box.

   a) Create a new application. Go to Panel, add a String Control and a String Indicator to the front panel. Rename the String Control title as "input", rename the String Indicator as "output".
   
   b) On the Diagram, wire the String Control to the String Indicator. Go back to the Panel view, type some data into each little input area on both text controls (different text data in each), and then run the program. Which input text changes? Which text stays the same?
   
   c) On the front Panel, add a single LED indicator (under Boolean). Go to the Diagram view. Add a Two button Dialog. Connect return value from the dialog box (T?) to the LED indicator. Pass the string from the 'input' string Control into the name of the Dialog box, but do it so that all characters are capital letters. Do this by sending the string from the string control through the 'To Upper case" function in the String Palette. The output from this function gets sent to the Dialog box display. Make sure to save the application at this point, otherwise you may lose the LabView application if you click on the wrong 'run' button in part 2d.
   
   d) Test by entering some lower case data into the 'input' area on the front panel, then running the program using the 'run once' button. DO NOT use the continuous run mode of LabView. The dialog box should appear with the text string all in upper case. If you click on 'ok' the LED should turn on. If you click on 'Cancel' The LED will turn off.
   
   e) Be sure you have saved the application. After having saved the application, try hitting the 'continuous run' button. What happens? Is there any way to stop the program?
   
   f) Try hitting Ctrl-Alt-delete to bring up the Task Manager, and select LabView, and then select End Task. This is the only way to stop the process. Restart LabView and reload your saved Program, but do not run it.
   
   g) We will now 'fix' this lockup problem. In general, you should always test your software to make sure somebody cannot lock up the system by pressing the wrong run button. To fix the problem, you need to stop the program when the person clicks on one of the Dialog Box
Buttons. Go to the Diagram Window in the functions Palette, look in `Application Control' for a Stop Sign. Put the stop sign on the Diagram, and connect the input to the stop sign to the Boolean output from the Dialog box. The Stop sign will stop the program if it receives a true; the program keeps running if the Stop Sign has a false. Save the program and then try to run in both `run once' mode and in continuous mode. The program should exit nicely when you click on the correct dialog box button (OK).

3) A boolean to decimal converter, and the difference between I8 and U8 integers.

a) Create a new application. Place 8 vertical toggle switches roughly in a straight horizontal line on the front panel. Click with arrow to draw a selection box around all 8 switches. Now use "Distribute Objects" and "Horizontal Center" to distribute the switches evenly in the line, then use the "Align Objects" to align the switches on Vertical centers. Your resulting switches should all look very neatly aligned and evenly spaced.
b) Rename the switch titles, from left to right, as '128', '64', '32', '16', '8', '4', '2', and '1'.
c) Go to diagram view, and click with arrow to draw selection object around the 8 boolean outputs, and align neatly as you did in part 3a.
d) We will take these individual bits and concatenate them together to form a single byte. Each switch will then control the value of a different bit of the byte. To form the byte, use the "Build Array" function from the functions Palette when you are in the Diagram. Drop one of these objects into your Diagram. Now add 7 more inputs by placing the arrow tool into the left hand side of this function, then right-clicking with your mouse to select the `add input' function. Add seven more inputs so the total number of inputs is eight.
e) Wire the switch outputs, in sequential order, into the inputs of the build array. Note that the bottom-most input of the array will be the leftmost bit, the one corresponding to the 128's bit; the topmost input will be the rightmost bit, corresponding to the 1's bit.
f) Convert this array into a number using the "Boolean array to number" function under "Numeric" and submenu "Conversion"
g) Go back to the panel view and add a Digital Indicator (Under "Numeric").
h) Go to the diagram view, and you now need to set the type of data to be displayed by the numeric control. Point to the numeric indicator (In yellow, says "DBL") and right click and go down to submenu "Representation". This is currently set to "DBL" meaning Double Precision. Change this to "U8" meaning unsigned Byte.
i) Run the program from the front panel, and click the switches to different states Verify the numeric output is the proper sum of the individual bit values. In particular, see what happens when you click the 128 switch on and off. What is the biggest number you can get? What is the smallest?
j) Now go to the diagram view, and change the data type from U8 to I8. Point to the numeric indicator (In blue, says "I8") and right click and go down to submenu "Representation". Change this to "I8" meaning signed Byte.
k) Run the program from the front panel, and click the switches to different states Verify the numeric output is the proper sum of the individual bit values. In particular, see what happens when you click the 128 switch on and off. What is the biggest number you can get? What is the smallest?
m) In 3k, if I changed U8 to U16 or U32, would it make any difference? What would the minimum/maximum that could be displayed as I toggled my 8 switches?
n) In 3k, if I changed U8 to I16 or I32, would it make any difference? What would the minimum/maximum that could be displayed as I toggled my 8 switches?