

PHY 406 - Microprocessor Interfacing Techniques

LabVIEW Tutorial - Part II

A Simple Example

A Very Simple Example

A very long time ago (or yesterday, depending upon your age) it was suggested that if you could write a program to say “hello world” on the teletype (this was a long time ago!) you had probably done about 90% of the required work in learning the programming language. At a minimum you had mastered the basics. I am going to help you do the same thing in LabVIEW but the example is: $2 \times 3 = 6!$ This will take some time because I want to go into some details about things, so bear with me and please actually do the example!

Open LabVIEW. Swap to the front panel screen (grey background). The first thing we will need is the *tools palette* which is the available toolkit for making things. Select **Windows>>Show Tools Palette** and a tools palette should appear (it may already have been there - don't worry about that!). We will use most of the tools in the palette but at the moment we just need the selection tool which is the top middle one. Click on selection tool  to use it (you may need to click twice - once to activate the window and once to get the tool. I am now in the habit of doing two clicks to pick up a tool to make sure I get it - that's two single clicks, not a double click)



In order to do this example we will need two controls (for the “2” and the “3”) and an indicator for the “6”. Since the two inputs will be variables, you will be able to do all sorts of multiplications! We select a control by using the *controls palette* (**Windows>>Show Controls Palette**)

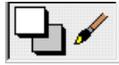


To use a palette is a bit tricky (at least I find it so!) There are ten panes on the control palette. You need the top left one. Click on it once and then drag the mouse directly onto the new pop-up window. This window shows some of the possible controls which supply numeric (as opposed to boolean, text, etc) inputs. The one we want is again in the top left. Click on it - and don't worry when the panel disappears!. Move the mouse to the front panel window - at which point a dotted outline appears - and position the control on the front panel. Click again and the control appears.

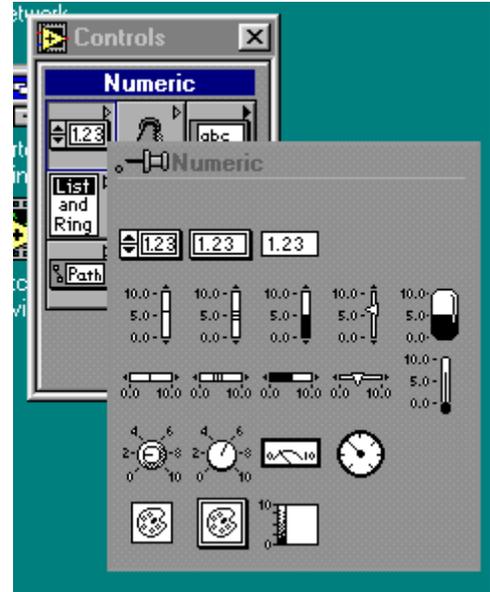
The control comes up with a small text panel ready for text input. **DO NOT PRESS ENTER.** Type in some text like “var a” and then press ENTER. This adds a label to the control on both the front panel and the block diagram. Now try selecting the label and moving it. Notice

that the control stays put. Move the label back. Now do the same to the control. Notice that the control and label move as one. The point is that moving a label moves it in relation to the control. Moving the control retains the relative spacing between the control and the label. You will therefore need to be careful which one you move.

Now I am going to show you how to “embellish” a control by changing the colours. I have no particular reason for this - but it’s fun! However it will help you to get some experience with the tools and it will also (I hope) give you some satisfaction in customising your front panels. The two tools you need for this are the colour pick tool  - third rank left on the tools palette and the painting tool

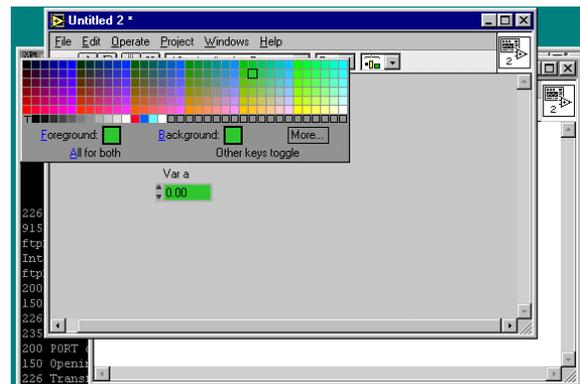


which is the bottom panel.



There are many pieces of the control which you can colour and I don’t want to do all of them, but here are two:

Get the colour pickup tool and click once on the background of the VI. Then get the paintbrush and place the tip at the background to the text label. Click once and the background changes to the same colour as the general background and disappears.



Now, still using the paintbrush, go to the number and click on the *right* mouse button and hold it down. A colour panel appears and by sliding around the colours you can change the background of the number - I generally pick white as a background, but that’s my preference.

To make the second control we will duplicate the control (thus keeping our colour changes and so on) and then modify it. Pick up the selection tool and select the control (remember that’s the *control*, not the *label*). Then do a Ctrl-drag (hold Ctrl and left mouse button down and drag the control to the new location). This places a new copy of the control on the front panel, but it is identical to the original - with the exception that the label changes. In order to distinguish it from the first one we will change the label. We do this using the text tool which is the upper-left tool . The pointer changes to a text tool and we can go and click on the control label and change it.

Next we need an indicator - which we get from the controls palette (which therefore should be the controls *and* indicators palette). Just for fun we’ll use a vertical slide type

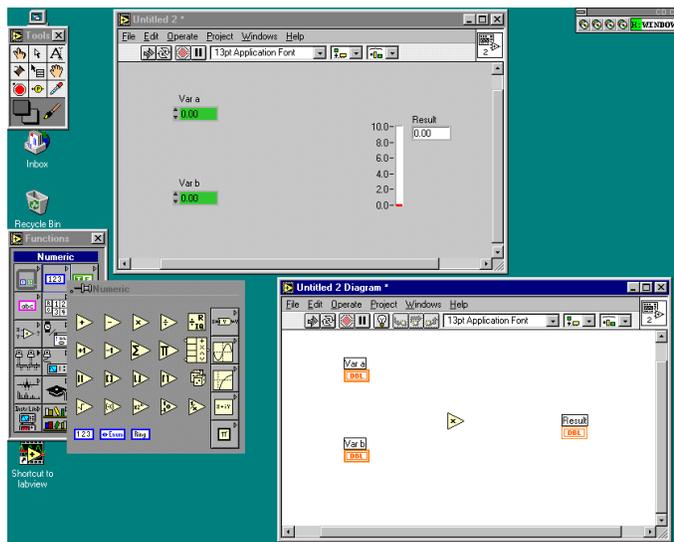
indicator which is on the same panel as the control you got before. (Watch out there are several things which look similar - the one you want is called “vertical fill slide”) When you have placed it, labelled it and coloured it, notice that there is a problem - this is a *control* and we need an *indicator*. No problem!. Using the select tool, select the control, press the *right* mouse button and the first menu item is - *change to indicator*. Notice that when you select that, the little arrows on the digital display disappear - you don’t need them for an indicator.

Now with a bit of judicious selecting and moving you can give the front panel a nice appearance. Try selecting the two controls (Using the select tool, start at upper left and drag the mouse across both controls to lower right whilst holding the left mouse button down) and then aligning them vertically using the alignment tool from the top of the window.

Now swap to the block diagram. You will see the backside of your two controls and an indicator. You can independently move these around to please yourself on this side (remember to move the control/indicator, not the label only!). An important aspect of LabVIEW programming is the concept of *dataflow programming* - things happen as the data for them are available, and by convention things are generally arranged to flow from left to right. Since the two input variables to our problem are available simultaneously, they should align vertically to the left and the result then appears to the right.

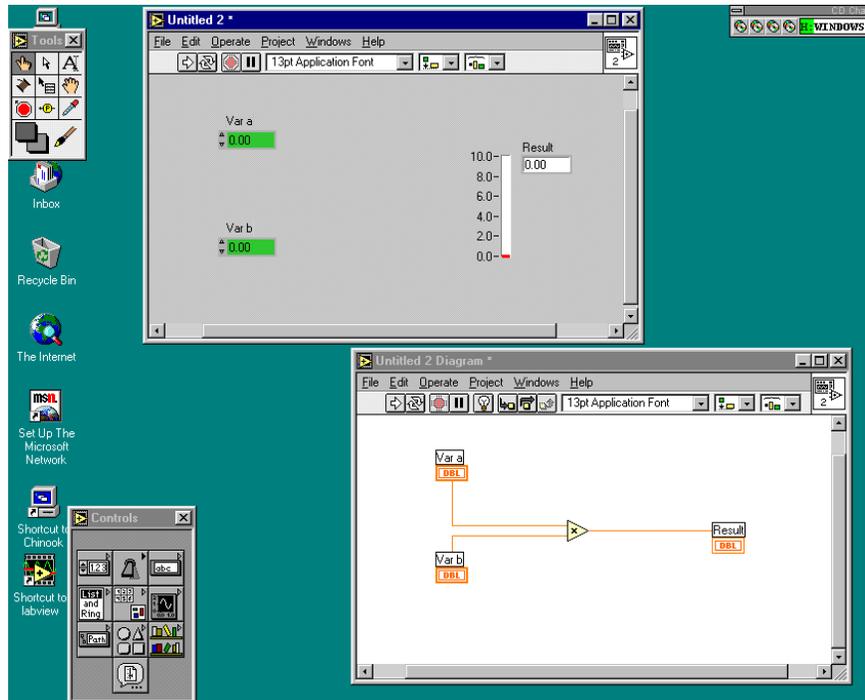
Between the inputs and the outputs we need a multiplication function. We get these naturally enough from the *functions* palette, **Windows>>Show Functions Palette**. Use **functions>>numeric** and get the multiply function. Put it in the middle of the block diagram.

Now we need to wire the diagram up. This is done with the spool of wire on the *tools* palette so select that tool and then go to the first input control. The *DBL* box will flash as you pass over it. To lay a wire, push the left mouse button while the box is flashing and hold it down until you reach the upper left point of the multiply symbol which will also start to flash and show a label *x*, then lift the mouse button and then click it. The wire should turn brown/orange. If it stays dotted, press **Ctrl-B** (remove bad wires) and try again. It is slightly tricky to get a wire to lay down properly, but watch for the upper left of the symbol to flash, then release the mouse button and the wire should stay there. Do the same for the lower input and the output indicator.



OK- you’re done - now your screen should look (somewhat) like this:

Go to the front panel and run the VI. Since the two inputs default to 0.0 - the answer is 0.0. To actually do the question asked, you need to change the two input values to 2.0 and 3.0. Make sure the VI is stopped (there is zero chance that it hasn't, but at this stage you should not change things while the VI is running) by making sure that the *start* indicator is in the "not-running" state which looks like this . Now use the *operating tool*  to change the values - either by clicking on the up arrow on the appropriate control or over-writing the value. Run the VI again and you should get the right answer!



Finally all this is no use unless you can save your work. To do this select **File>>Save As** and save the file in an appropriate directory under an appropriate name. (Important - you must add the ".vi" extension to the file - LabVIEW won't do that for you). Notice that this is slightly different to the way we loaded the thermometer example above, in that case we used a file which was a library of VIs and then loaded a single VI from the library. However I did say that I would only show you one way of doing any one thing when there might be many ways of accomplishing the desired end.

Please note that you should save all your own files in your own home directory which is the *F-Drive* on this system. When you get the file selection menu - press the up icon until you get to the screen which permits you to choose the F-drive and then select it. Within that drive, you can do what you like! Do not leave anything on the local drives of the PC - these drives may be cleaned out without any notice whatsoever.

Data Input and Display

Now let's try a few variations. One of the very strong points of LabVIEW is its ability to control the way in which input is authenticated and displayed. All of these are controlled by the pop-up menu which comes up when you click on the *right* mouse button when over the appropriate item. Some of the most useful things you can do are:

Change the default value	Select Data Range and then alter the <i>Default</i> value.
Change the rate at which the variable changes when you click the “up” and “down” arrows	Data Range and then alter the <i>Increment</i> value.
Change the allowable range of the data	Data Range and change the <i>Maximum</i> and <i>Minimum</i> values.
Change what happens at and beyond the limits	Data Range then select from “If value is out of range” - the options are: <i>Ignore</i> Don’t apply any limit checking <i>Coerce</i> If the value is out of range, reduce it to the nearest limit value. The increment/decrement arrows will not exceed the range <i>Suspend</i> If the value goes out of range the VI is suspended or will not run - you can also add an error message.
Change the way in which the data are presented	Format & Precision and make appropriate changes.

Try changing a few of these to see how it all works.

You can also change the range of the thermometer display simply by changing the maximum value on the display with the operating or text tools

Fun Stuff (aka Customisation)

Some more things you can do to brighten up the front panel are to add labels, colours and decorations.

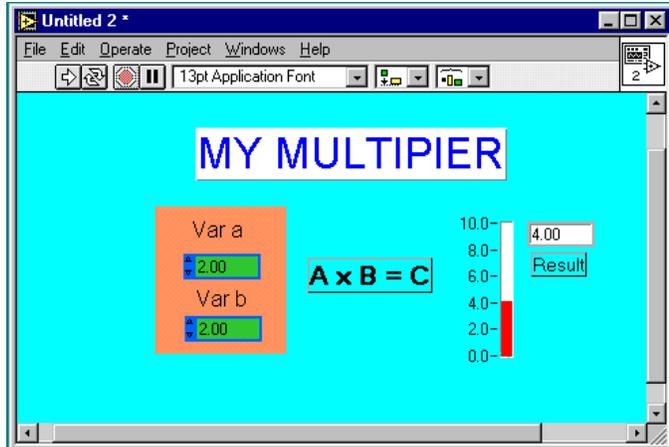
Labels are added with the text tool and any text item can be changed by selecting it and then using the font menu at the top of the screen.

Colours are added with the colour tool as I’ve already explained - you might already have accidentally discovered that you can change the background colour as well as any other colour by using the paintbrush away from a control or indicator.

Decorations are accessed through the decorations item on the controls menu  and can be independently coloured. Objects may be sized by selecting them, then dragging a corner to resize it. You will also need to cope with *layering* - which item lies on top of another - since only the top one of a stack of objects is visible. Hence if you have a decorative panel behind a control,

that is exactly what you need - a decorative panel *under* a control. Layering is controlled from the **Edit** menu and any selected item can be brought up a layer, down a layer, to the front or to the back. Usually it is only necessary to bring things to the front or push them to the back.

Anyway after I had played around a bit my front panel looks like this - which it must be admitted is unique!



Summary

- ▶ You can create your own VI which used *controls, functions and indicators*.
- ▶ You can load and save VIs
- ▶ We can alter the way in which data are displayed on controls and indicators.
- ▶ We can alter the default values and the rate of change of a variable
- ▶ We can alter the limiting values of a variable and the action when a limit is reached
- ▶ We can change the size, colour, font etc of text
- ▶ We can add decorations to the screen

Exercise

Try changing your VI to compute a quadratic (ax^2+bx+c) with controls for a,b,c and x and a sensible front panel layout.