

PHY 406 - Microprocessor Interfacing Techniques

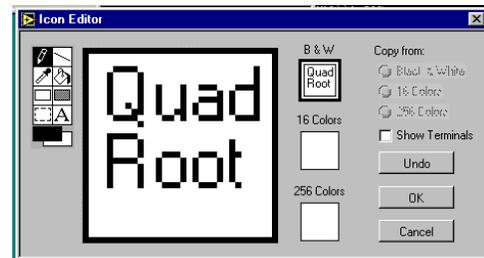
LabVIEW Tutorial - Part VII

Wrapping It Up - SubVIs

Hiding the Hard Work

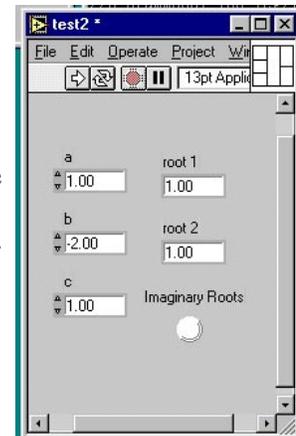
Encapsulation or the enclosing of a body of work into a sub-VI is an important part of serious LabVIEW programming. We are going to encapsulate this VI. To do this we will need to create an ICON and also to define the inputs and outputs required for this VI. When we encapsulate a VI the controls are replaced by the inputs to the sub-VI and the indicators are replaced by the outputs from the VI. It isn't necessary to have all the controls and indicators accessible from outside the sub-VI, but in the case of controls, you had probably have a good idea about what values they default to and whether these are the right values. It is probably better to use constants to prevent changes being made accidentally.

Go to the *front panel* and pop-up on the icon in the top right. Select **Edit Icon**. A little graphical editor should pop-up. I am assuming that you can run something like this. I used the select (dotted box) to clear the icon and then the text tool to write and got this:



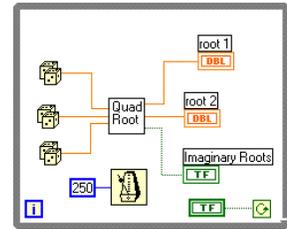
When you exit the icon editor you will notice that the icon has changed to your new one. This is the icon which will show when you use this entire VI as a sub-VI.

Now pop-up on the icon again and select **Show Connector**. LabVIEW assigns a connector pattern to this VI. The general rule in LabVIEW is that inputs are on the left, outputs are on the right. This connector pattern has only two slots on the right and I want three, so I can change it with the pop-up **Patterns** and get one with three slots on the left and three on the right. Notice also that the tool has changed to the wiring tool. To associate each of the inputs and outputs with a particular sector of the connector pane, click on the front panel item and then on the connector sector. When a connection is established, the sector turns dark.



Now would be a really good time to add descriptions to all the controls and the VI to enable the help facilities to work properly. When you have finished all this, save the VI (don't forget the ".vi" at the end of the name).

Now open a new VI and using the **Select a VI..** option you can put your new sub-VI in place. If you did add the help for it, you can now access all the narratives and your VI will look just like many other parts of LabVIEW



Here's a really simple use of our sub-VI

Summary

- ▶ Sub-VIs can be created from a conventional VI.
- ▶ When used as a sub-VI, inputs replace controls and outputs replace indicators
- ▶ The ICON can be edited to be more realistic
- ▶ The connector pane establishes which inputs and outputs are placed where
- ▶ Help is very, very useful in order to use a sub-VI

Exercise

Construct a sub-VI to compute $x!$ ($x(x-1)(x-2)\dots(1)$) from x as an input. You might want to think about the range and type of input you will accept and an efficient algorithm for computing the result.