PHY 406 - Microprocessor Interfacing Techniques
LabVIEW Tutorial - Part IX
Sequences

Sequences

LabVIEW works on the concept of *dataflow* - a function or VI executes when the inputs are available. This is generally OK since we are processing a flow of data, but there are occasions when this is not the case. One of the principal areas where we get into trouble is timing.

In a typical timing experiment things go like this:

i. Set up the system
ii. Start the clock
iii. Send the event trigger
iv. Wait for the ending event
v. Stop the clock
vi. Tidy things up

Now we can debate fiercely whether we should start the clock before sending the trigger or aftwerwards, but there is a much deeper problem in LabVIEW and that is that there is no obvious dataflow in the pattern. Nothing flows through “start the clock” to get to “send the trigger”. These events (starting the clock, sending the trigger, stopping the clock) could all occur at very different times and in different sequences, but we wish to impose a sequenc on them. In LabVIEW this is the job of the *sequence* structure.

In common with the *case* structure, the *sequence* structure has a number of panels lying on top of one another. The difference for the sequence function is that they all execute one after the other.

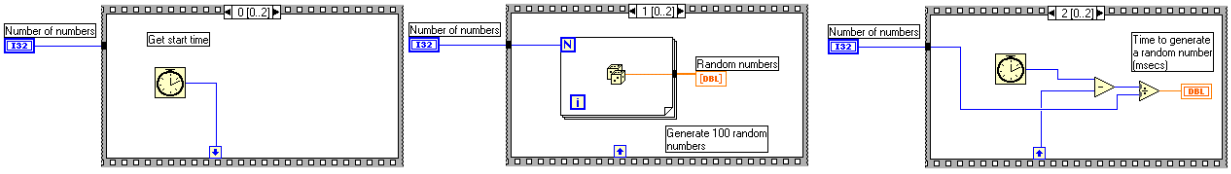
We also need a way of transferring information between the panes as well as to the outside world. Communication with the outside world can be made through *tunnels* which permit a wire to corss the boundary of the sequence. A *tunnel* can come onto or off any pane, but as usual input tunnels can be used in multiple panels and can be ignored at will on other panels, whereas output tunnels must have one and only one connection.

We also need a way of communicating between panes. This is supplied by a *sequence local* which is created anywhere on the perimeter of the sequence panel and can then be fed from an output on one panel to input(s) on other panel(s). *Sequnce locals* are created from the pop-up menu on the edge of the sequence. This menu also allows you to add panels before and after the current one as well as to delete panels.

Moving between panels can be accomplished by **clicking** on the arrows at the top of the sequence.

Here is a simple example of a sequence. We wish to find out how long it takes to generate a random number. We need to get the time before we start, generate a lot of numbers and then find the time when we stop - from the difference we can find the time to calculate one random number (plus a bit of "end effect" - but we'll ignore that at the moment)

Here are the three panels which make up this sequence:



(Note that this diagram is made up from three separate views of the sequence - you would normally only see one of these panels at a time)

The first panel collects the time in msecs (since some arbitrary start which needn't concern us here). This panel transfers the start time to other panels using the sequence variable at the bottom.

The second panel computes the required number of random numbers. It uses the input control to tell it how many numbers to compute and display. This tunnel input is available on all panels, but is only used in two panels

Finally the third panel takes the start time, subtracts it from the end time and divides the result by the number of random numbers calculated to give us the time per random number.

The sequencing of the panels ensures that things are done in the right order.

Many instrument set-up operations require that some things are done before others, and this is another useful place for a sequence. There are other ways of achieving this end, but this is often one ofhte clearest and simplest.

Summary

- Sequences can be used to force things to happen in a given order if the dataflow is not sufficient to do so.
- Sequence locals permit varaibles to be trasnferred between the panes of a sequence
- Tunnels permit variables to be brought in and out of the sequence

Exercise

Make a sequence to measure your reaction time to a light going on. (You will only be able to do this to about 1mS, but the exercise should be instructive)