

LabVIEW – Variables

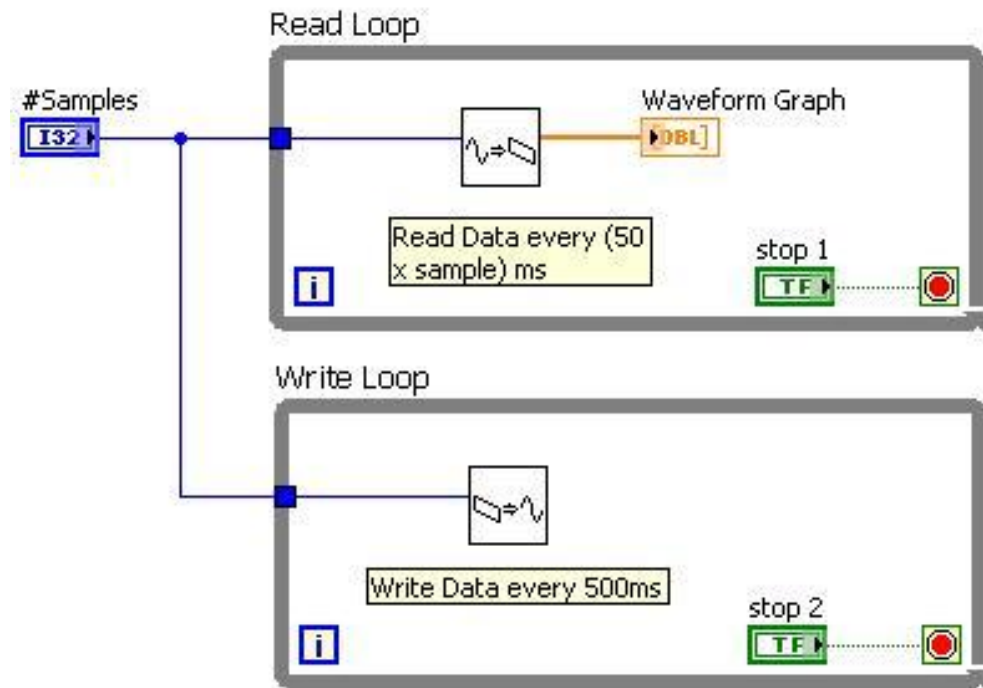


Gary Boorman



Dual While Loops

- Two While loops in parallel. Equal processor time given to each loop
- How to stop both loops with only one Stop button?



Local Variables

- A Local Variable can read or write to controls or indicators on the front panel of a VI
- Useful to communicate between structures within one VI
- Place Local Variable on diagram, select the variable to which to link and whether read/write

Local Variable
linked to 'stop 1',
direction 'read'

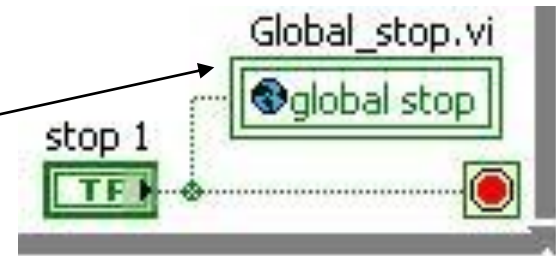


Global Variables

- A Global Variable is used to access and pass data among several VIs
- A global variable is a VI that has its own front panel, but no diagram



Global Variable
linked to 'global stop'
direction 'write'



Global Variable
linked to 'global stop'
direction 'read'



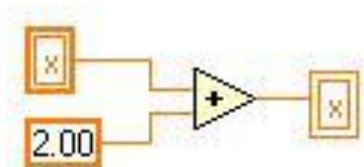
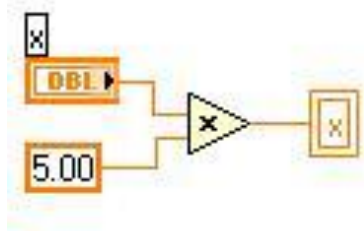
Variable Hazards I

- Both Locals and Globals must be initialised (must be written to first before being read).
- Variable will contain default data if uninitialised
- Be aware of Race Hazards – the variable being read before meaningful data has been written to it
- Each copy of a Local or Global creates a new buffer in memory – fine for scalar Booleans, not so good for large 3D arrays of Doubles...



Variable Hazards II

- Order of execution of variable reading/writing must be guaranteed



- Do not write to the same variable you read from



Functional Global Variables I

- Functional Global Variables (FGV) are VIs that use loops with un-initialised shift registers to hold global data
- Data is kept even when VI is not running, until FGV VI removed from memory
- Can be accessed by multiple VIs
- Can provide more functionality than just read/write



Functional Global Variables II

Implements initialise/read/increment/decrement

