

# **LabVIEW Course**

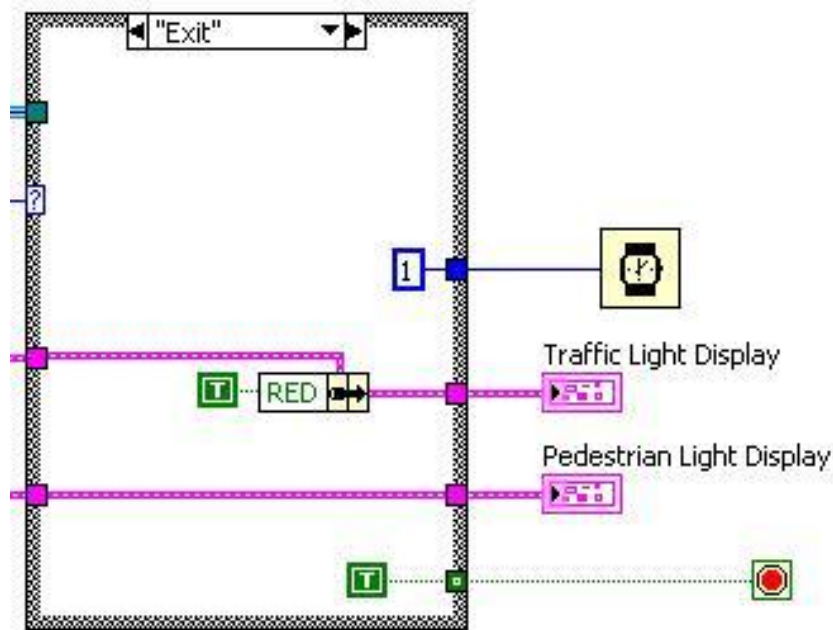
## **Exercise 16**

**G Boorman 2011**

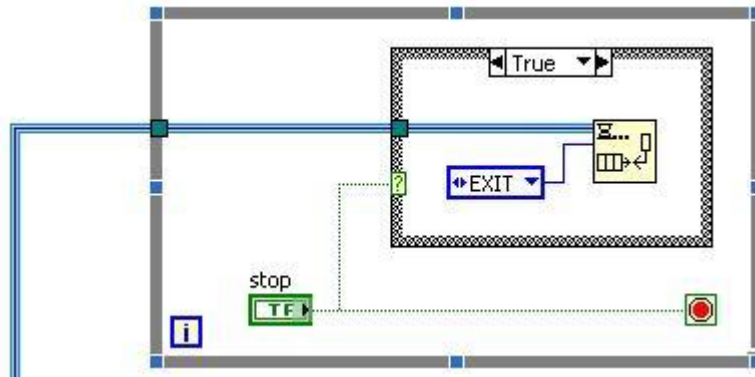
## Exercise 16 – Queuing from a Second Loop

This exercise takes the *Multiple Queued Pedestrian Lights.vi* produced at the end of Exercise 15 and adds to the Queue from a second loop.

- 1) Make a copy of *Multiple Queued Pedestrian Lights.vi* and name it *Queued Pedestrian Lights Second Loop.vi* and add to the Traffic Lights project.
- 2) Edit the *Traffic Light States.ctl* and add another item called 'Exit'. Do *Apply Changes* and save the control (note: previous VIs will now show a broken arrow and not run. How can they be modified to work again?)
- 3) Go to the 'Amber' case in the Case structure, right-click on the case Label and do *Duplicate Case*. A new case 'Exit' will appear. Edit this case so that the Red traffic light will be shown and the Wait time is 1ms. Add a Boolean 'True' constant in the lower right-hand corner of this case.
- 4) Disconnect the 'Stop' control from the While loop's Loop Condition terminal, and wire from the Boolean constant in the Exit case to the Loop Condition terminal. Right-click the tunnel created and set to 'Use Default if Unwired'.



- 5) Add a second While loop above the existing loop. Wire the Queue Ref created by the 'Obtain Queue' function to the left-hand edge of the loop. Move the 'Stop' control into this loop and wire it to the Loop Condition.
- 6) Add a Case structure into the new loop, and wire the 'Stop' control to the case Selector. Place an 'Enqueue Element at Opposite End' function and wire the Queue Ref to the *Queue* terminal. Create a constant linked to the *Element* terminal, and make it 'Exit'. Leave the False case empty.



- 7) Check the VI works as expected and save it.

## End of Exercise