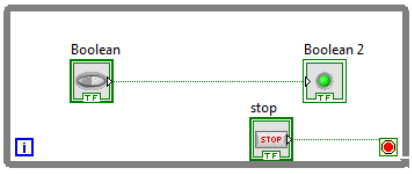
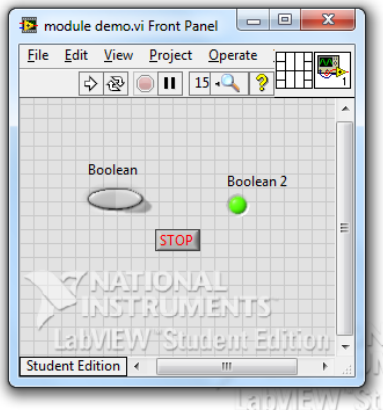
Module 11 – Introduction to LabView Programming Including Arduino Toolkit

Carey Williamson

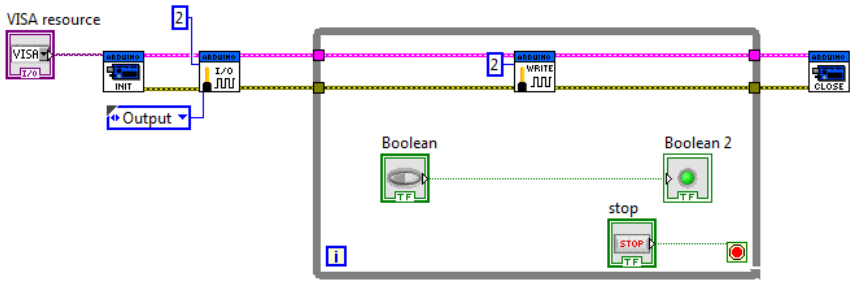
1. Develop a basic processing loop.
    a. Start LabVIEW 2011. Select New Blank VI. Save the file.
    b. Place a button on the front panel
    c. Place an LED on the front panel
    d. Switch to the block diagram and create a while loop (located in the express -> structures pallet) around the two Boolean VI's.
    e. Wire the two Boolean VIs together.



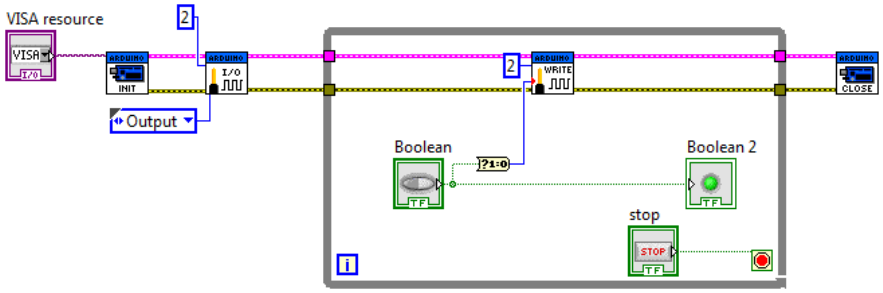


    f. Run the VI. Observe the operation. The Boolean indicator should illuminate when the Boolean button is pressed.
    g. Open the computer Task Manager and note the CPU usage here _____ . We will use this value in section 3.
    h. Press stop button to stop the loop.
    i. Add an Arduino INIT VI, Set Pin Mode VI, and a Close VI outside the loop.
    j. Add an Arduino Digital Write Pin VI inside the loop.

k. Add controls and constants as shown in the following image.

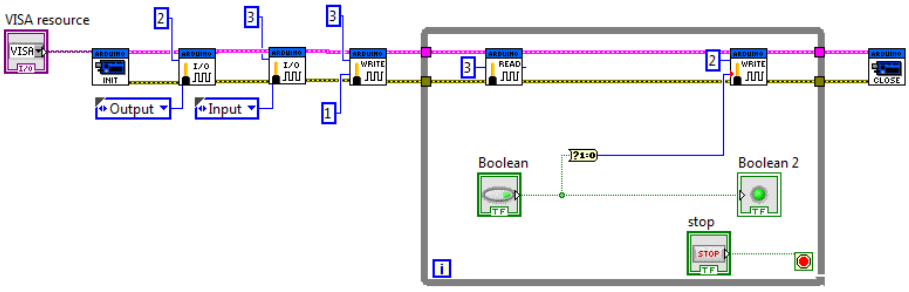


l. Add a Boolean To (0,1) VI inside the loop and connect it's input and output as shown.



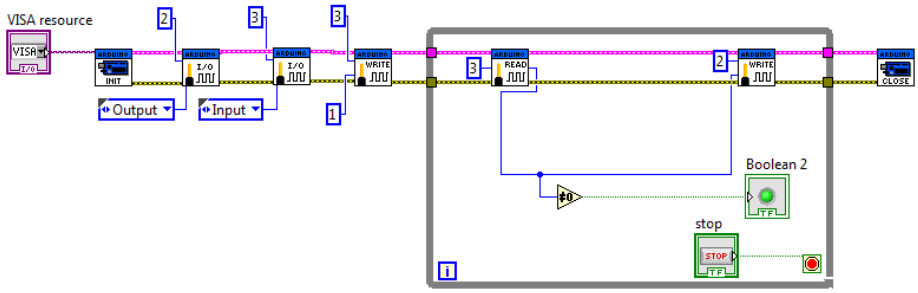
m. Connect the Arduino to the computer and an LED/resistor to pin 2 I/O and ground.
n. Run the VI. Observe the operation. Press the stop button to stop the loop.
o. Save the file.

2. Read the state of an input pin and control an output based upon that state.
   a. Add another Arduino Set Pin Mode VI in line with the first one.
   b. Add a Arduino Digital Write VI following the Set Pin Mode VIs. This enables a built-in pull-up resistor in the Arduino.
   c. Add an Arduino Digital Read Pin VI inside the loop in line with the Digital Write VI.
   d. Add controls and constants as shown in the following image.



   e. Delete the Boolean button and the Boolean To (0,1) VI.
   f. Connect the Arduino Digital Read Pin VI output to the Arduino Digital Write Pin input.

g. Add a Not Equal To 0? VI and connect it as shown



h. On the Arduino, add a normally closed pushbutton switch connected to pin 3 and ground.

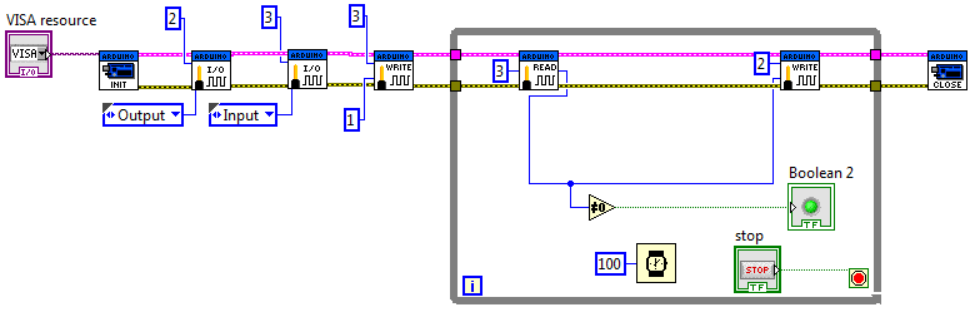
i. Run the VI. Press the switch. Observe the operation. The connected LED and Boolean indicator will illuminate as the pushbutton is pressed. Press the stop button to stop the loop.

j. Save the file.

3. Incorporate a delay timer in the processing loop.

a. Add a Wait (ms) VI inside the loop.

b. Give it a constant value of 100 as shown.



c. Run the VI and note the CPU usage here _____. Compare this to the value to the one found in 1g. The wait conserves CPU utilization.

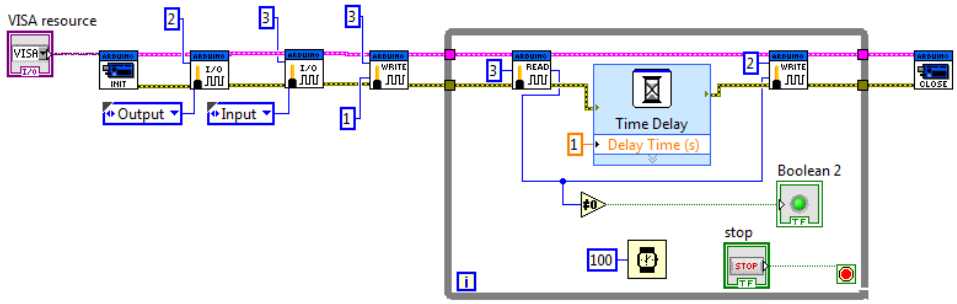
d. Adjust the Wait (ms) VI constant value to 3000.

e. Run the VI and observe the affect. The loop only runs once every 3 seconds. Press stop button to stop the loop.

f. Set the Wait (ms) VI constant value to 100.

g. Add a Time Delay VI in-between the Digital Read Pin VI and Digital Write Pin VI, connecting the Error In and Error Out in line.

h. Give the Time Delay VI a constant of 1 as shown.



i. Run the VI and observe the affect. This adds a time delay between the Digital Read Pin and the Digital Write Pin. Press the stop button to stop the loop.

j. Adjust the Time Delay VI constant to 3.

k. Run the VI and observe the affect. Press the stop button to stop the loop.

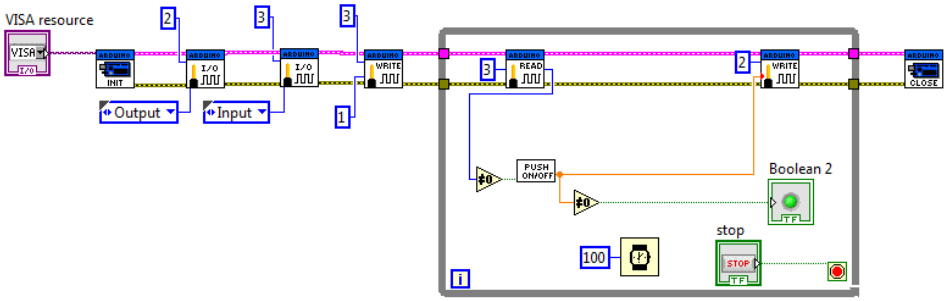
l. Save the File

4. Add a sub-VI to the processing loop.

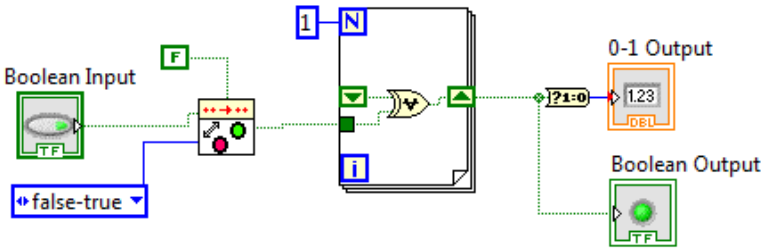
a. Delete the Time Delay VI and its constant.

b. Add the given push_on_off.vi to the loop.

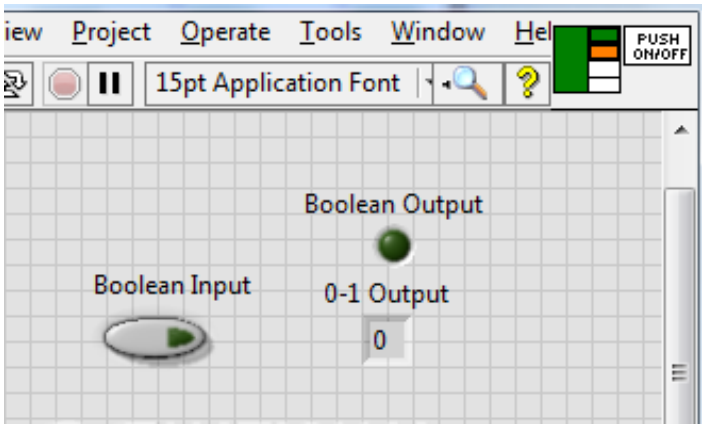
c. Add another Not Equal To 0? to the output of the Digital Read Pin VI and to the input of the push_on_off VI. Wire the output to the Digital Write Pin input as shown.



d. Open the push_on_off VI and inspect the subroutine. This subroutine finds the instantaneous moment the switch is pressed and maintains that state until the switch is pressed again.



e. Note the Terminal Pane on the Front Panel. This is how this VI interfaces with main program loop. Also note the VI icon and the ability to customize it.



f. Run the main VI. Press the switch. Observe the operation. Press the stop button to stop the loop.

g. Save the file.