
USING THE MOBILE STUDIO I/O BOARD WITH LABVIEW

Created by:
Michael Kleinigger

The Mobile Studio I/O Board is a powerful device for data acquisition; signal generation and control (see <http://www.mobilestudioproject.com> for further details and instructions describing how to download the supporting software). While the Mobile Studio Desktop software (developed by Jason Coutermarsh) allows users to both acquire and generate signals, it is not designed to use board for system control. This is where LabVIEW comes in. LabVIEW is a powerful, graphical programming language that makes it easy to implement custom control systems for lights, motors, automated sensing, home automation and more.

This tutorial describes the steps needed to operate the I/O Board within LabVIEW. We will begin with a simple example, creating a LED voltage indicator. A single-ended analog voltage is read through the I/O Board and processed by LabVIEW. Based on the level of this signal, one or more of the board's digital outputs will be set high. The outputs can be used, with buffers, to drive LED indicators.

In order to perform the steps outlined in this tutorial, you will need:

- **The Red2 I/O Board** (note: the Blue I/O Board will also work with this tutorial, but it does not support a number of other LabVIEW interfaces, such as arbitrary waveform generation and PWM output)
- **Mobile Studio Desktop Software**
(http://mobilestudio.rpi.edu/downloads/MobileStudioDesktop/Production/mobile_studio_desktop.exe)
- **The LabVIEW 8.5 I/O Board interfaces** (available via: <http://www.mobilestudioproject.com/mobilestudiolabviewutilization> - Please note that LabVIEW 8.0 and 8.2 can also be used if version 8.5 is not available.)
- **National Instruments LabVIEW 8.5** (available to RPI students as part of the default laptop image; or it can be via download at: http://www.rpi.edu/dept/arc/web/licenses/labview_license.html)

Utilization Steps

1. To begin, open LabVIEW 8.5. Once the “Getting Started” screen has appeared (as shown in Figure 1), under the “New” menu select “Blank VI”. You are now presented with a blank front panel and behind it, an empty block diagram (Figure 2). The front panel is the user interface for your LabVIEW program, and the block diagram is where the program is built, using various types of program blocks.

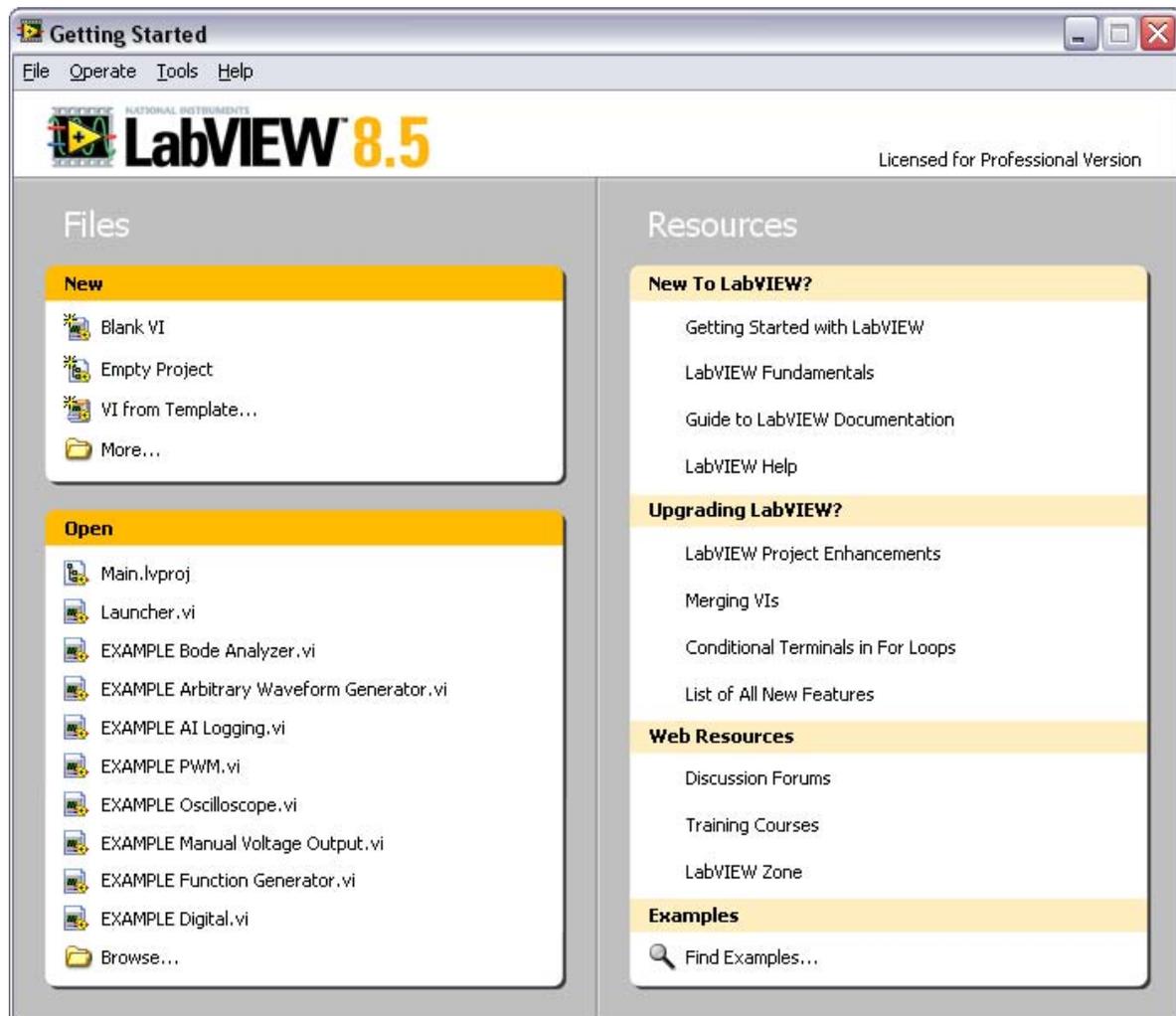


FIGURE 1 - LABVIEW "GETTING STARTED" SCREEN

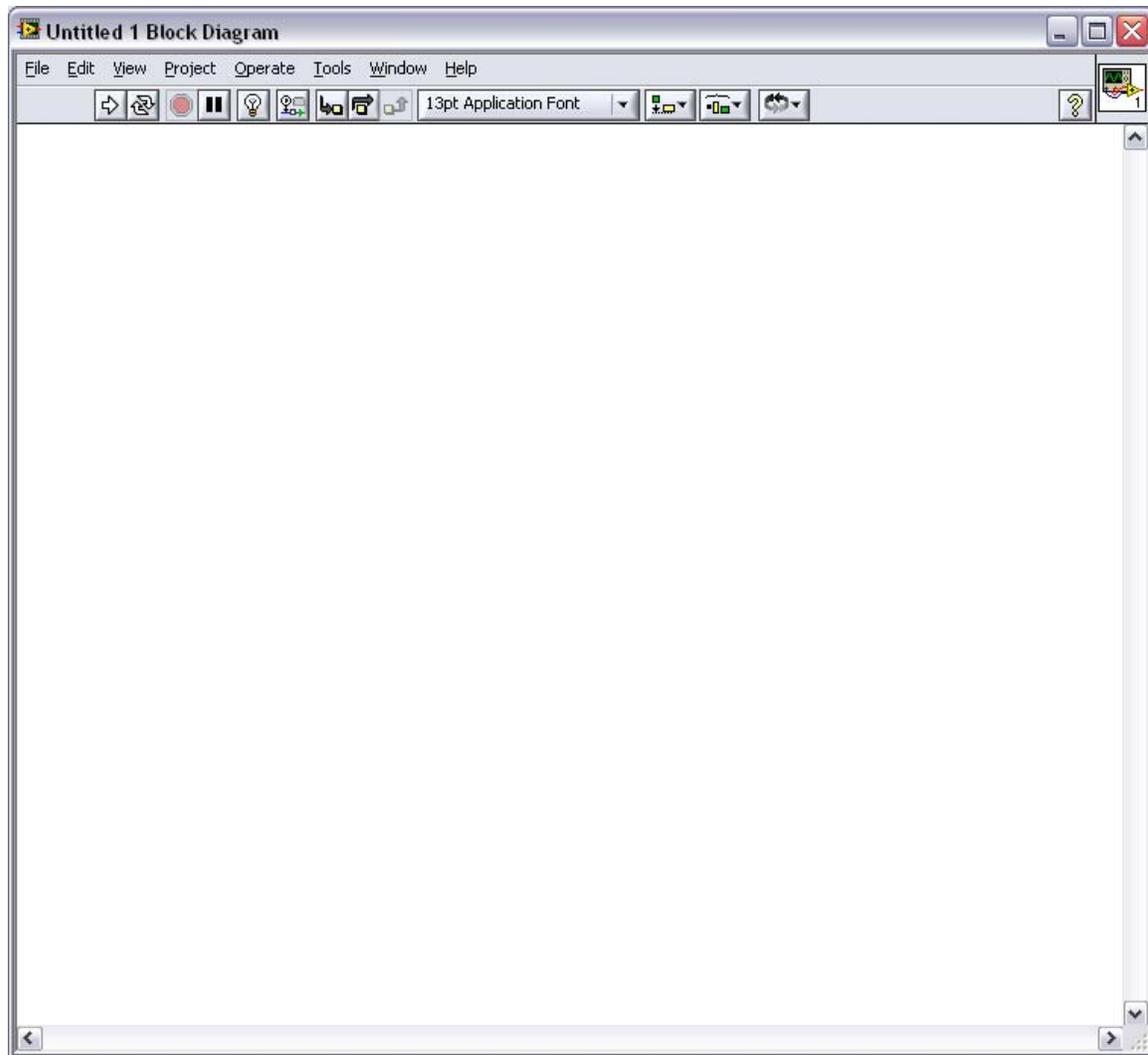


FIGURE 2 – BLANK LABVIEW BLOCK DIAGRAM

2. Select the block diagram window (located behind the grey front panel). To do this you may either click the window's title bar, or simply press Ctrl+E. This shortcut can also be used to switch back to the front panel when needed.
3. The first step for any I/O Board program is to add the block which will initialize our communications with the board. To do this, open the file folder where you've saved the LabVIEW I/O Board interface VIs. Resize this window and the LabVIEW block diagram so they are both visible on your screen. Locate the file named "**OpenBoard.vi**", then click and

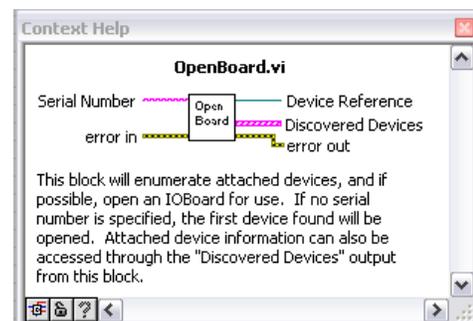


FIGURE 3 - CONTEXT HELP

drag the filename onto the block diagram. When you release the left mouse button, you will see that the “Open Board” block has been added to your program. If it is not already visible, open the context help window by pressing Ctrl+H (Figure 3). This window displays valuable information for whatever block you move your cursor over. Notice the colored, labeled lines connecting to the block. These are called wires, and are used to pass data between blocks within LabVIEW. However, not all inputs and outputs from certain blocks need to be wired (LabVIEW will complain if you try and run a program without all necessary wiring). The “Open Board” block does not require any inputs, but can be optionally wired to a serial number so that if you’re using more than one I/O Board, you can choose between them.

NOTE: If using the LabVIEW 8.0 interface VIs, you will need to make the following adjustment before the VIs recognize your attached I/O Board. Open **OpenBoard.vi** by double-clicking the block, then hit Ctrl+E to switch to the block diagram. Find the leftmost block labeled “I/O Board” (Figure 4) next to the “error in” control. Double-click this block, then select “I/O Board” from the objects list and click OK. Run the VI to make sure your device is found, then save and close.



FIGURE 4 - FIXING OPENBOARD.VI

4. For this program, we are only concerned with a single output from the “Open Board” block: “Device Reference”. This wire contains data for communicating with the I/O Board and is used by all other I/O Board function blocks. Before closing the folder with your I/O Board function VIs, drag “**AnalogInputConfig.vi**”, “**AnalogInputRead.vi**”, and “**DigitalIO_Single.vi**” onto your block diagram. Connect the “Device Reference” output from the “Open Board” block to the “Device Reference Input” of the “Analog Config Start” block by clicking once on the terminal of each block. Similarly, connect the “Device Reference Output” line from “Analog Config Start” to the “Device Reference Input” terminal of “Analog Input Read” (Figure 5).



FIGURE 5 - I/O BOARD BLOCKS

Next, move your cursor over the left edge of the “Analog Config Start” block until you see the words “Channel Configuration” appear. Right-click on this terminal and select “Create” → “Constant”. A box will appear with several options (Figure 6). Click and drag this box to move it to a more convenient location.

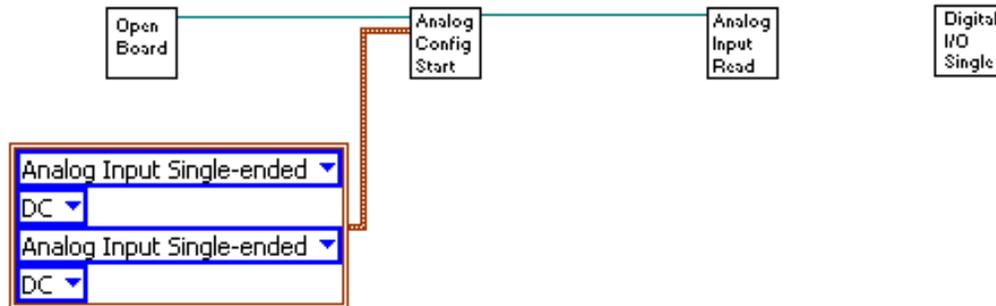


FIGURE 6 - ANALOG CONFIGURATION CONSTANT

For this program, the default values that appear inside this box (called a cluster) will work fine. By clicking the text within the blue boxes, you can adjust the mode (single-ended, differential, or function generator) and coupling (AC or DC) of the analog inputs. The other terminals on the “Analog Config Start” block will not be used in this program, but can be used for more complex data acquisition as in the oscilloscope example program.

- Next, we will add a while loop so that our program will run continuously. To do this, right-click on any blank part of the block diagram. This will cause the function palette to appear. The “Programming” section should already be visible, so move your cursor over the “Structures” icon. Click on the “While Loop” icon. Finally, click and drag a box around the “Analog Input Read” and “Digital I/O Single” blocks (Figure 7) and a grey box will appear, as seen below.

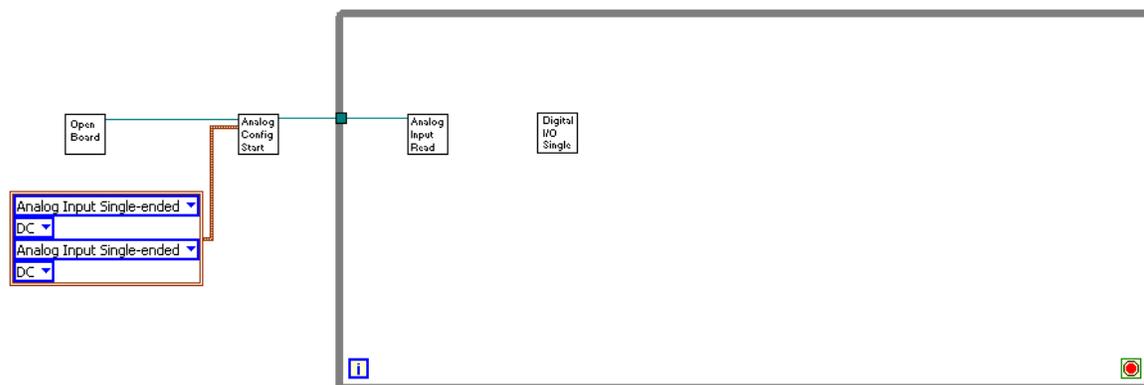


FIGURE 7 - WHILE LOOP ADDED

Next, right-click the “I/O Channel” terminal of the top “Digital I/O Single” block and create a constant (which will be an integer, a blue wire). Set this constant to “1”, then create constants for each of the other digital I/O blocks, set to “2”, “3”, and “4” in order from top to bottom. Create constants for the “State Input” terminal of these four blocks and set them all to “False”. Next, create constants for the “Output?” terminals and set them all to “True”. Optionally, you can create a single “True” constant and connect it to all four terminals (do not do this for the “State Input” terminals). This is done by simply clicking on a terminal, then clicking on the wire coming from the “True” constant.

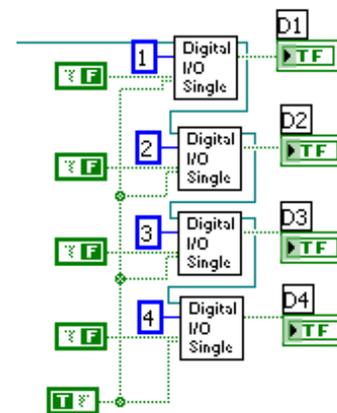


FIGURE 10 - DIGITAL I/O

Finally, right-click on the “State Output” terminal on the right side of the digital I/O block and select “Create” → “Indicator”. Do this for each block and label them as shown here. You may also right-click on the indicator icons and select “View as Icon” in order to reduce their size on the block diagram. Notice that there are now four dark-green indicator lights on the front panel. You may click and drag these indicators to rearrange them as you see fit.

- Returning to the “Analog Input Read” block, right-click the “Channel 1 Data” terminal, then select “Array Palette” → “Index Array”. Place this block just to the right, and wire its “array” terminal to the “Channel 1 Data” terminal (Figure 11). Since the “Analog Input Read” block is also used when acquiring large arrays of data (as in the case of the oscilloscope example program), the single point voltage we are interested in must be picked out of the array using the “Index Array” block. Right-click the “element” terminal on the right side of this block and select “Create” → “Indicator”. Label this indicator as “Voltage”.

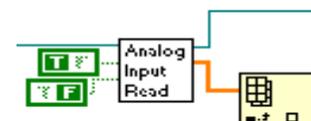


FIGURE 11 – ANALOG INPUT SETTINGS AND INDEX ARRAY

Next, right-click in any of the whitespace on the block diagram to bring up the function palette. Add the “Multiply” block under “Programming” → “Numeric” and the “Round to Nearest” block, also located under “Programming” → “Numeric”. Create a numeric constant of “1.2” for the lower terminal of the multiply block and then wire them together as shown in Figure 12.

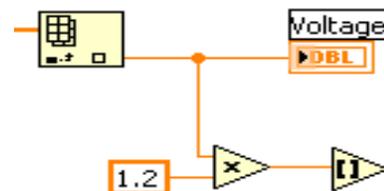


FIGURE 12 - READING VOLTAGE

- Next, we will create a case structure to select different LED output patterns based on voltage. To add this structure, right-click in any blank portion of the block diagram and select “Programming” → “Structures” → “Case Structure”. As you did with the while loop, click and drag a box around only the digital I/O blocks (Figure 13).

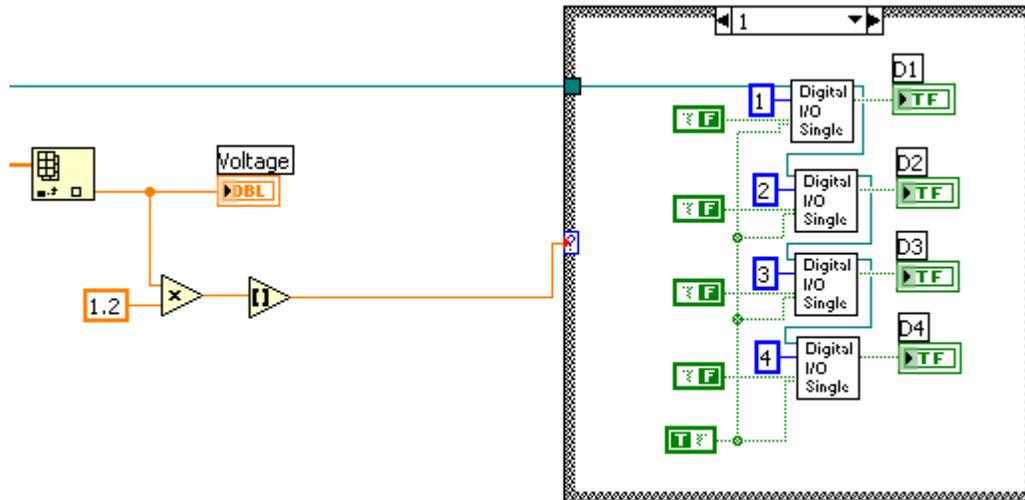


FIGURE 13 – CASE STRUCTURE ADDED

Connect the output from the “Round to Nearest” block to the “?” terminal of the case structure, as shown above. The input to this terminal will determine which case of the case structure is executed. By default, the digital I/O blocks have been grouped under case “1”. Right-click the numbered drop-down menu at the top of the case structure (which is also used to navigate between cases by clicking the black arrows) and select “Duplicate Case”. Next, select and delete the four Boolean indicators (“D1 2” through “D4 2”) from this new case. If any broken wires remain, you can quickly remove them by pressing Ctrl+B.

Now return to case “1” by using the drop-down menu at the top of the case structure, and right-click on any of the Boolean indicators. Select “Create” → “Local Variable”, then place this block in any convenient space outside the case structure. Move back to case “2”, then place copies of the local variable block inside the case structure by dragging them while holding down the control key. Set their values by clicking as shown in Figure 14.

Duplicate this case twice, as described above. Once you have a case “4” and have deleted the indicators, left-click and drag a box around the contents of the case structure, then press Ctrl+C to copy the selected blocks. Navigate to case “0” by clicking

the drop-down menu at the top of the case structure, then press Ctrl+V to paste a copy of the digital I/O blocks. Be sure to reconnect the top “Device Reference Input” terminals. You will also need to re-set the values of the local variables and delete the extra indicators which LabVIEW automatically creates.

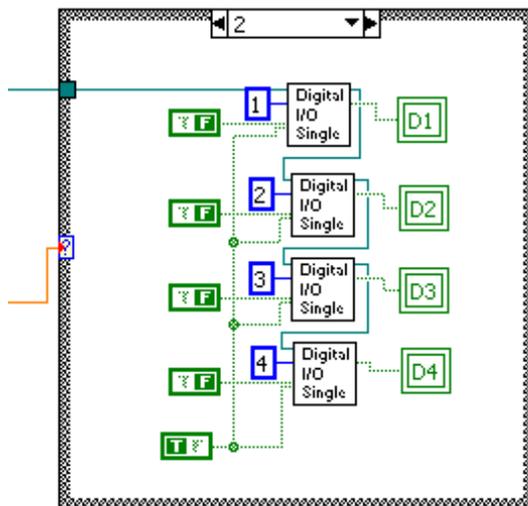


FIGURE 14 – LOCAL VARIABLES

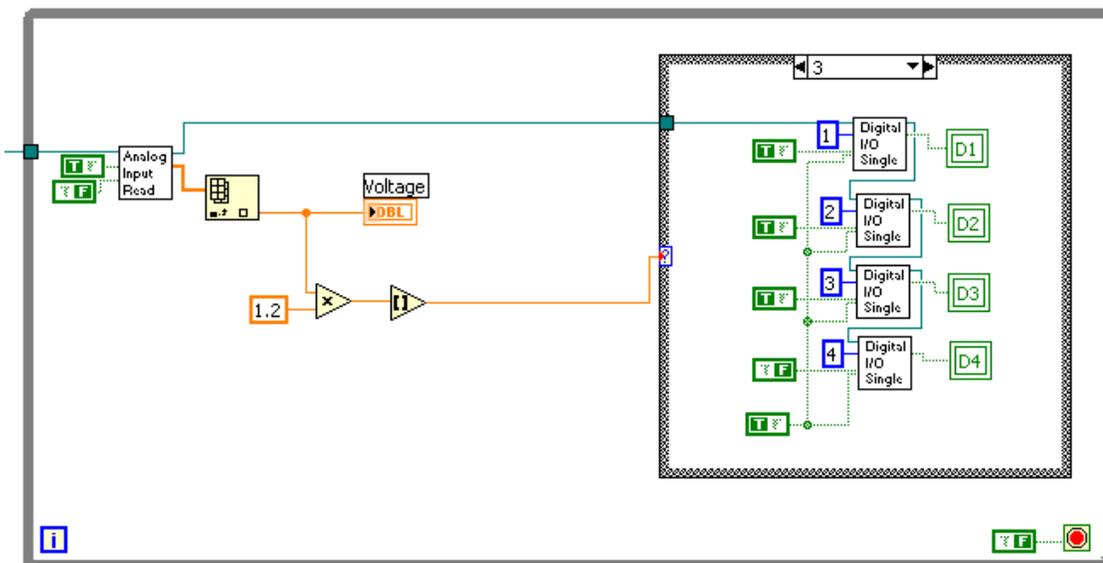


FIGURE 15 - "STATE INPUT" SETTINGS, CASE "3"

Finally, we need to set the values of the “State Input” Boolean constants so that they are correct and unique for each case. Case “0” is correct already – all “State Input”

terminals should be wired to “False”. Set only the top digital I/O block’s “State Input” in case “1” to “True” by clicking on its value. In case “2”, set only the top two blocks to “True”. In case “3”, the top three blocks are “True” (Figure 15), and in case “4”, all “State Input” terminals are wired to “True” constants.

10. The demonstration program is now complete! Return to the front panel by pressing Ctrl+E. To run the program, press Ctrl+R or click the “Run” arrow icon in the upper-left corner of the screen. The “Voltage” field will display the measured voltage and the indicators will light accordingly. Optionally, you may wish to change the look of the voltage indicator. This can be done by right-clicking the “Voltage” field and selecting “Replace” → “Numeric” then choosing from the available options. You may also change the scale of this indicator by double-left-clicking the scale numbers and typing in new values. The final result is shown below.

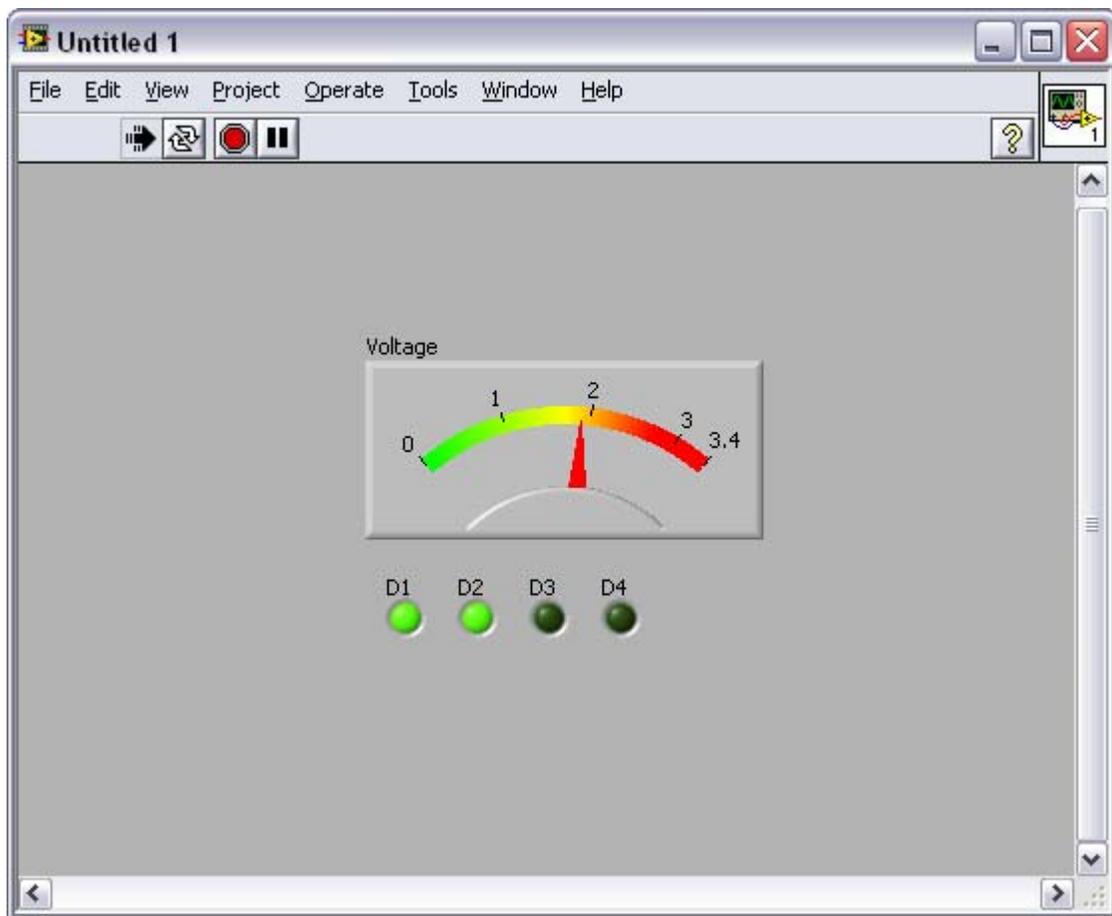


FIGURE 16 - COMPLETED VI RUNNING

Final Notes

- Other I/O Board blocks work very similarly to the ones presented in detail in this passage. Documentation for each block is available in the context help window. Once you've added a block to the diagram, press Ctrl+H (assuming the context help window is hidden), then simply place your cursor over the block in question to view its details.
- Don't forget to save your work often! LabVIEW is normally very stable, but has been known to crash, especially when working with the Simulation Module package (not addressed here).
- See the following link for a list of helpful LabVIEW tips and tricks:
<http://www.mobilestudioproject.com/Top10-LabVIEW.pdf>