

Part III Exercise 1: Controlling LEDs

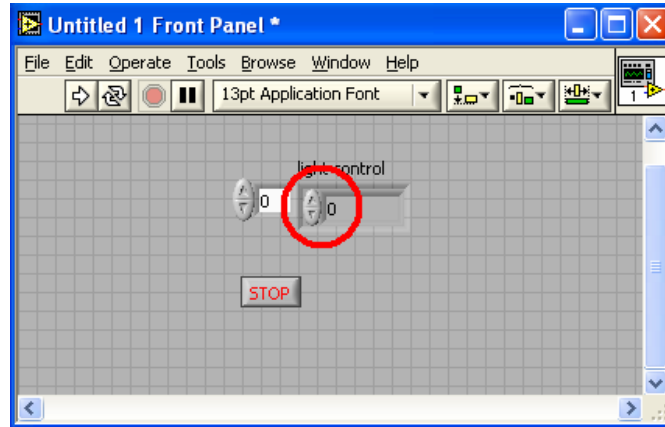
In this exercise, you use the DAQ Assistant and automatic code generation to control the 8 digital I/O lines to send 5V TTL signals to an LED connected to each line.

Note: This exercise uses simulated hardware. You will not be able to verify that the data you output since you have no physical way to connect the output to an input or to a physical indicator, such as an LED. However, this exercise will show you the process for developing a digital output application using NI data acquisition hardware and NI LabVIEW, specifically using the automatic code generation functionality of the DAQ Assistant.

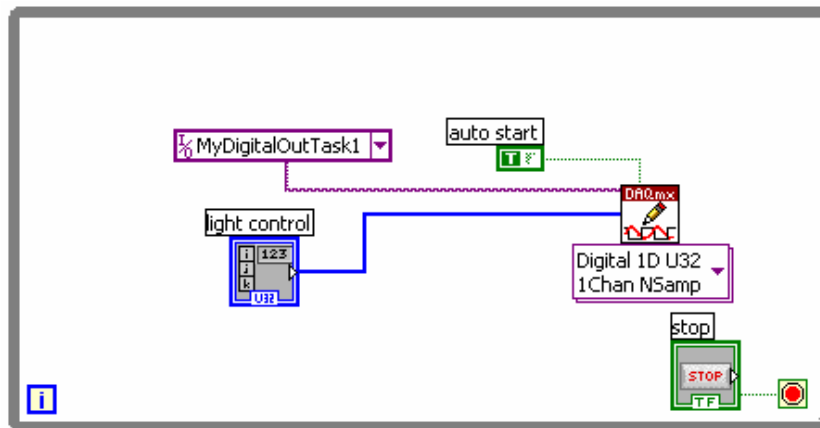
1. Visit <http://ni.com/trylabview> and click on the **Try LabVIEW Online** link to access the LabVIEW Online Evaluation. Once your browser has been tested, click Continue and you will be directed to the remote desktop.
2. If you have just logged on to the LabVIEW Online Evaluation, you will first need to configure your measurement system. Refer to Part I, Exercise 1 Configuring a DAQ System in Measurement & Automation Explorer (MAX).
3. Launch LabVIEW from the desktop.
4. Click **New** to start a new program. At the next splash screen, select **Blank VI** and click **OK**.
5. Press <Ctrl-E> to switch to the LabVIEW Block Diagram.
6. Right-click to bring up the **Functions** palette, and place a **DAQmx Task Name Constant** on the Block Diagram from the **All Functions » NI Measurements » DAQmx** subpalette.
7. Right-click on the **DAQmx Task Name Constant** and select **New Task (DAQ Assistant)**. The DAQ Assistant window should appear.
8. Select **Digital I/O** as the Measurement Type, and in the next window, select **Port Output**. In the next window, click the + sign next to Dev1, select **port0**, and click the **Finish** button.
9. Enter a name for your task. In the next window, notice that you have the option to invert the digital line logic. Leave this box unchecked, and click the **OK** button.
10. Use this task that you just created to generate modifiable LabVIEW code by right-clicking on the **DAQmx Task Name Constant** and selecting **Generate Code » Example**. The NI-DAQmx VIs require to perform this task should appear on the block diagram.
11. You will need to modify this code slightly to work with your application. Delete the **data** control connected to the VI, and use the polymorphic selector on the **DAQmx Write VI** to choose **Digital » Single Channel » Multiple Samples » 1D U32 (port format)**. This allows you to write our digital data to the entire port of digital lines.
12. You will now need to create a new control to create the data that the **DAQmx Write VI** will output to the digital lines. Right-click the **data** input terminal of the **DAQmx Write VI** and select **Create » Control**.
13. Right-click the data control you just created and select **Properties**. On the **Appearance** tab, change the label to `light control`. Click **OK** to close the Properties window.

Note: The light control is an array of numeric controls. The outer (white) number and up/down arrows control the index of the array. Because you are using only one digital port, you have only one element in

the array (element 0). The inner (gray) numeric control changes the value of element 0 in the light control array.

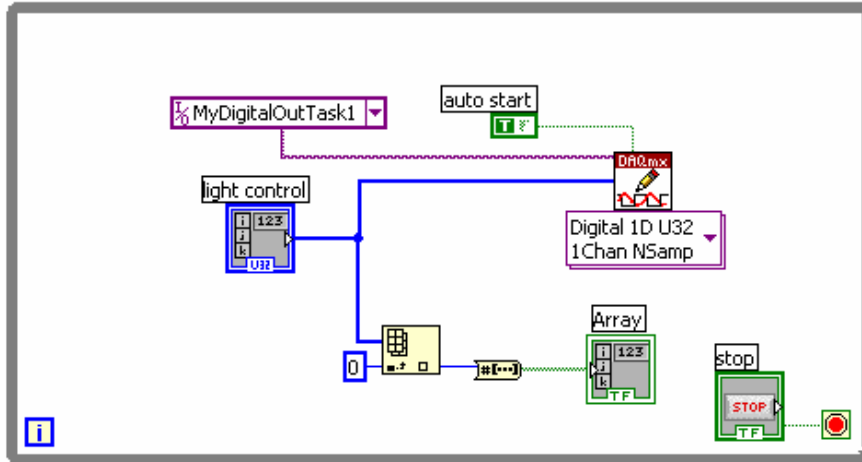


14. Draw a While Loop around all of your code. (A While Loop is found on the **Exec Ctrl** subpalette of the **Functions** palette.) Your block diagram should resemble the following figure:

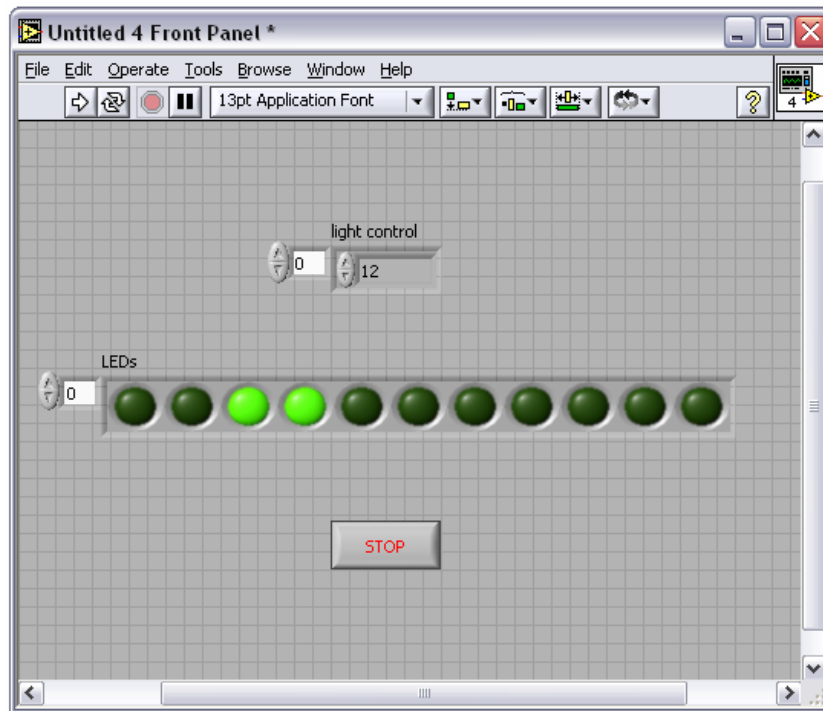


15. Since this exercise uses simulated hardware, you will need a way to visualize what would be happening to a set of LEDs if they were to actually be connected to the digital port you are writing to. Steps 16-21 detail the additional code you will need to add in order to see this.
16. On the Block Diagram, right-click to bring up the **Functions** palette, and navigate to the **Array** subpalette under **All Functions**. Select the **Index Array** function and place it on the block diagram inside the while loop.
17. Wire the output of the light control to the **array** input of the **Index Array** function. Right-click on the **index** input of the **Index Array** function, and select **Create » Constant**. This function will take just the first number of the array out. Since we are only using one number, this is all we need.
18. Right-click again on the Block Diagram to bring up the **Functions** palette, and navigate to the **Boolean** subpalette under **All Functions**. Select the **Number to Boolean Array** function and place it on the block diagram inside the while loop. Wire the output of the **Index Array** function to the **number** input of this function. This will convert the number into Boolean representation.
19. Press Ctrl+E to switch to the Front Panel, and create LED indicators for this program by dropping an **Array Control** onto the block diagram from the **All Controls » Array and Cluster** subpalette.

20. Drop a **Round LED** indicator from the **Controls » LEDs** subpalette into the empty array you just created. Expand the array by clicking on the right edge of the array and dragging until eight LED indicators are visible. Each of these LEDs will be used to display the state of a single digital line.
21. Press Ctrl+E to switch back to the Block Diagram, and wire the **boolean array** output of the **Number to Boolean Array** function to the input of the Boolean array you just created. Your final Block Diagram should resemble the following:



22. Switch to the Front Panel and run the program. Adjust the numbers on the light control to create the Boolean representation of that number in the LED indicators. If you had physical LEDs connected to the digital lines you are writing to, those LEDs would be changing state just as the indicators on the Front Panel are. Press the Stop button on the Front Panel to stop the program.



End of Exercise 1