Section 1 Guide

Background

Through the series of LabVIEW workshops you will be learning how to use and design in LabVIEW. In order to help you, a project was created that will challenge all your knowledge of LabVIEW, and yet it will still be flexible enough so that you could go beyond the scope of it and add more functionality or even add completely different features to the project to make it more interesting.

The project will be to implement the game *Nigel Says* (much like Simon Says found here http://www.miniclip.com/games/simon-says/en/) in LabVIEW. The objective of the game is to be able to memorize the sequence indicated by the LED's and then repeat it by pressing the buttons in the right sequence.

Fun Fact: Nigel is the name of National Instruments' eagle mascot!*

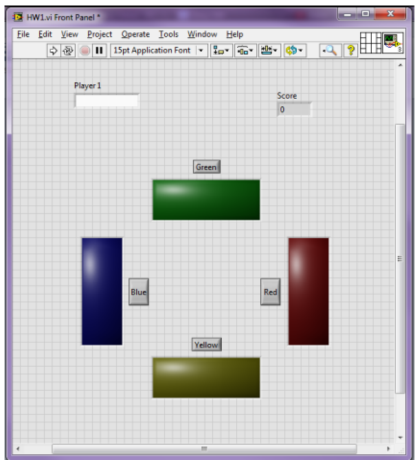
To help you complete this project, instructions are separated into sections corresponding to each Workshop (i.e. Section 1 can be completed after Workshop 1, Section 2 can be completed after Workshop 2, etc…).

Task

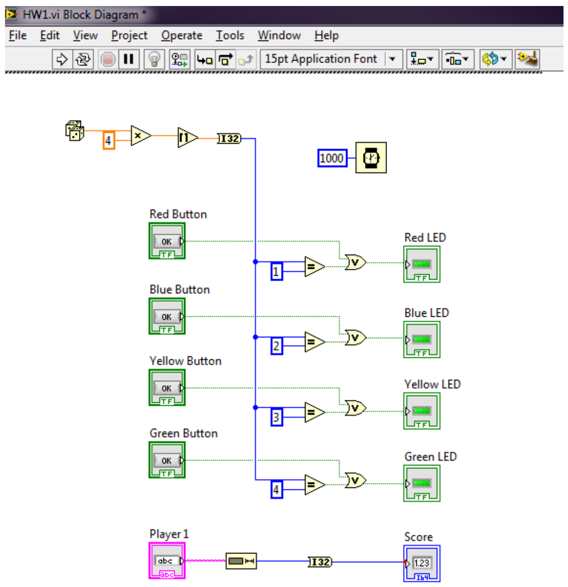
At the end of this section you will be more confident moving around LabVIEW. You will practice the skills you learned in Workshop 1 by replicating a VI, which will be the base for the Final project. The main objective of this section is to get familiarized with the LabVIEW environment, so don't be afraid to go beyond what this guide establishes and add as much stuff as you think might be helpful to the final project.

This task is divided in two main parts:

- Front Panel:



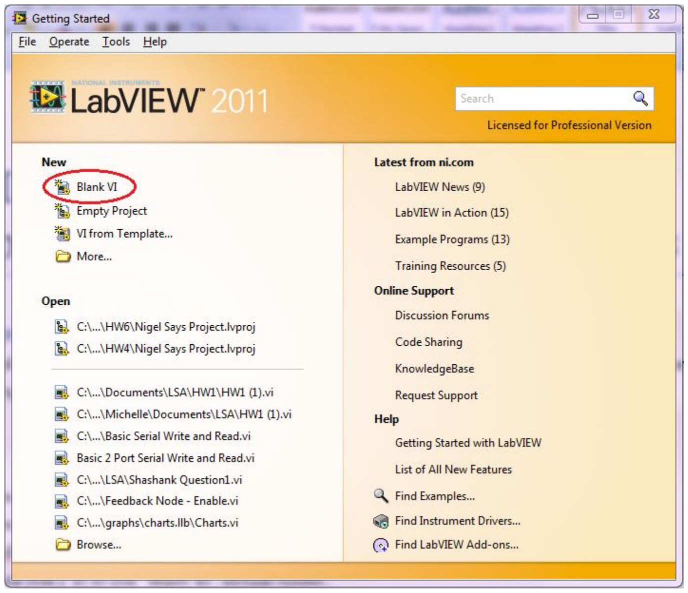
- Block Diagram:



Steps

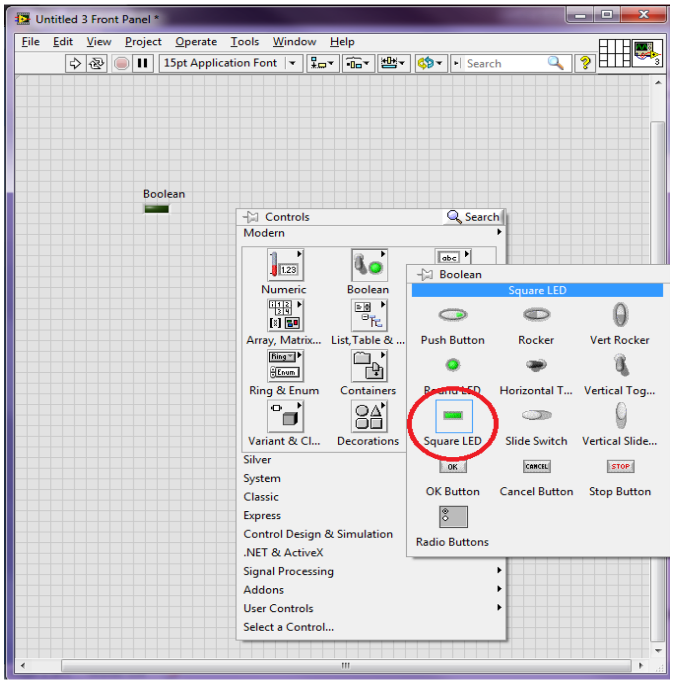
This part of the Guide provides you step-by-step instructions of how to obtain the expected results of this section. Feel free to follow all steps, or do even it by yourself and compare your solution at the end with the steps, filling in any pieces you may have missed. To maximize your learning experience, we suggest choosing the latter method. The first step is to create a blank VI, and then we will start adding items to the Front Panel.

1. Open LabVIEW and create a new blank VI:

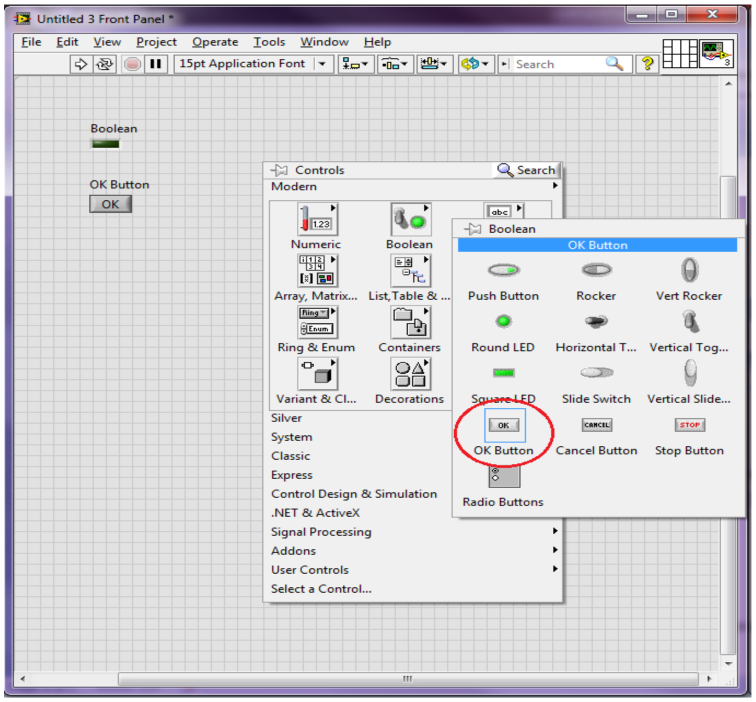

Getting Started
File Operate Tools Help
LabVIEW 2011
Search
Licensed for Professional Version
New
Blank VI
Empty Project
VI from Template...
More...
Open
C:\...\HW6\Nigel Says Project.lvproj
C:\...\HW4\Nigel Says Project.lvproj
C:\...\Documents\LSA\HW1\HW1 (1).vi
C:\...\Michelle\Documents\LSA\HW1 (1).vi
C:\...\Basic Serial Write and Read.vi
Basic 2 Port Serial Write and Read.vi
C:\...\LSA\Shashank Question1.vi
C:\...\Feedback Node - Enable.vi
C:\...\graphs\charts.llb\Charts.vi
Browse...
Latest from ni.com
LabVIEW News (9)
LabVIEW in Action (15)
Example Programs (13)
Training Resources (5)
Online Support
Discussion Forums
Code Sharing
KnowledgeBase
Request Support
Help
Getting Started with LabVIEW
List of All New Features
Find Examples...
Find Instrument Drivers...
Find LabVIEW Add-ons...

Front Panel

2. Place a Square LED on the panel (*Right Click > Controls palette > Modern > Boolean > Square LED*); this will later become the Green LED to display the color sequence in the game.

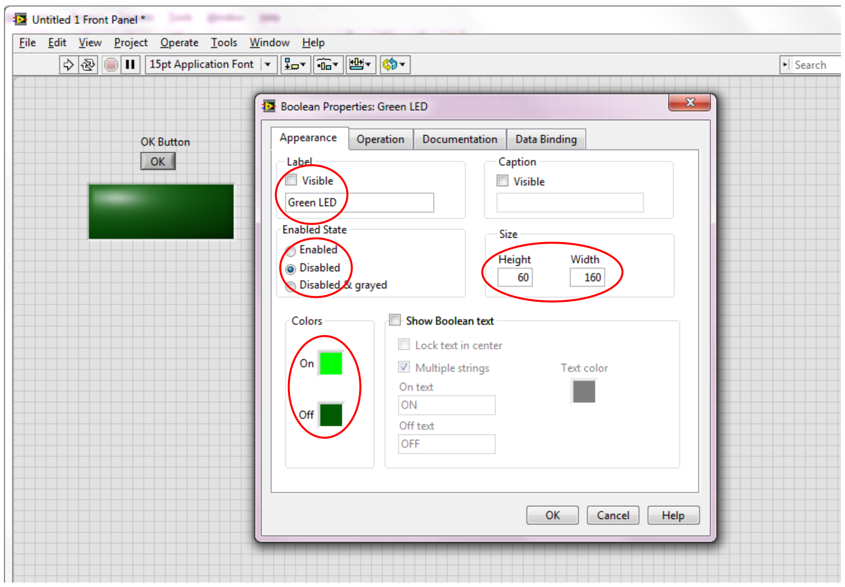


3. Place an OK Button on the panel (*Right Click > Controls palette > Modern > Boolean > OK Button*); this will later become the Green Button that the user clicks when replicating the LED sequence in the game.



4. Edit the Boolean LED properties (*Right Click on the LED > Properties*). Edit the label text box to say "Green LED". Uncheck the "visible" box for the label to hide the text. Change the Enabled State to Disabled since it is an Indicator and we don't want to modify its value by clicking it. Change the sizes and on and off colors to be as close to the RGB values as possible (see table).

Color	**On Color**			**Off Color**			**Sizes**	
	R	**G**	**B**	**R**	**G**	**B**	**Height**	**Width**
Green	0	255	0	0	91	0	60	160


Untitled 1 Front Panel *

File Edit View Project Operate Tools Window Help

15pt Application Font

OK Button

OK

Boolean Properties: Green LED

Appearance | Operation | Documentation | Data Binding

Label: ☐ Visible — Green LED

Caption: ☐ Visible

Enabled State: ○ Enabled ● Disabled ○ Disabled & grayed

Size: Height 60 Width 160

Colors: On 🟩 Off 🟩

☐ Show Boolean text

☐ Lock text in center

☑ Multiple strings

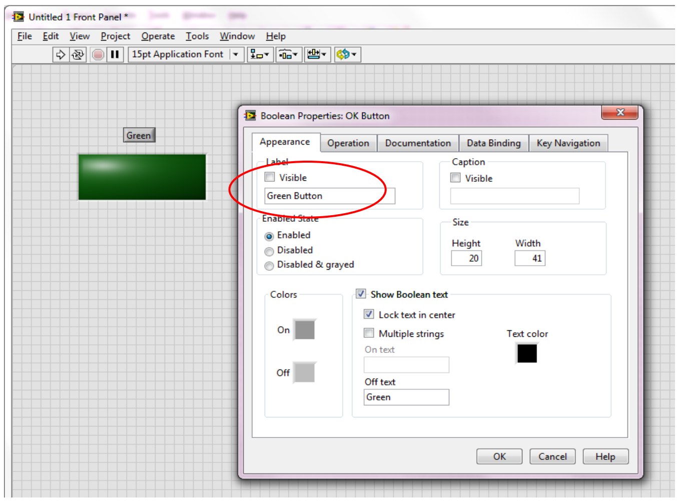
On text: ON

Off text: OFF

Text color

OK Cancel Help

5. On the front panel, double-click on the "OK" text on the actual OK button and type in "Green" instead. Also, edit the properties of the OK Button (*Right Click on the OK Button > Properties*) to edit the button name to "Green Button" and hide the label. Move the button so that it is on top of the LED.


Untitled 1 Front Panel *

File Edit View Project Operate Tools Window Help

15pt Application Font

Green

Boolean Properties: OK Button

Appearance | Operation | Documentation | Data Binding | Key Navigation

Label: ☐ Visible — Green Button

Caption: ☐ Visible

Enabled State: ● Enabled ○ Disabled ○ Disabled & grayed

Size: Height 20 Width 41

Colors: On Off

☑ Show Boolean text

☑ Lock text in center

☐ Multiple strings

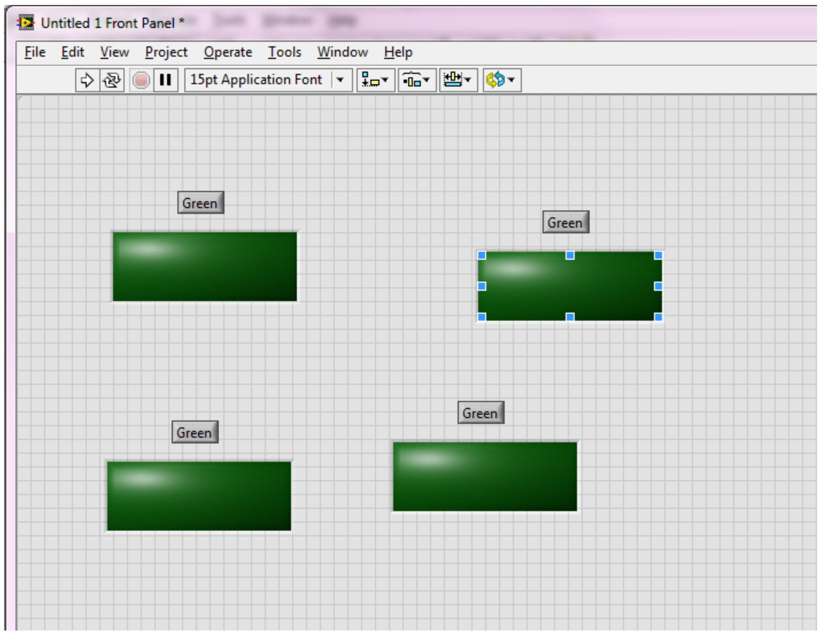
On text:

Off text: Green

Text color

OK Cancel Help

6. Copy and paste what we have, three times. (Select the LED and the Button by clicking on each item and holding down Shift; press Ctrl + Left Click while you drag the items to make one copy; repeat to make another copy). You can also use edit copy and edit paste along with Ctrl + c and Ctrl + v if you feel more comfortable using these methods. We are doing this to create more LEDs and buttons which will correspond with the Red, Yellow, and Blue colors of the sequence.



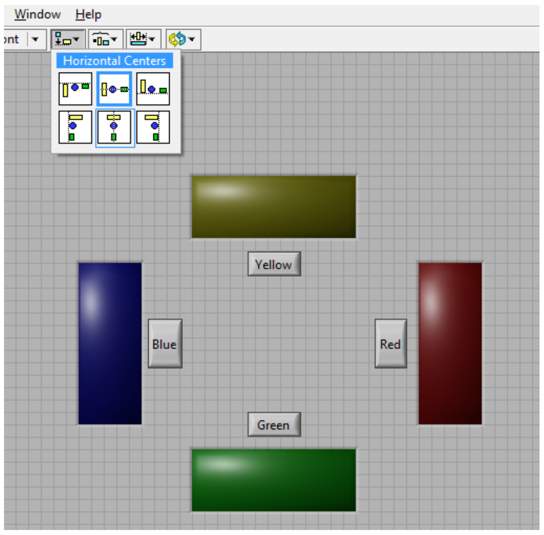
7. Change the new LEDs to have different colors (Red, Blue and Yellow), appropriate labels for LEDs and buttons, corresponding text on buttons, and correct sizes for LEDs. Be sure to change both the on/off colors for the LEDs. See Table below for RGB values of each of the on/off colors, and height and width of the LEDs. (All of these properties can be set in the LED properties box). (Hint: see steps 4 and 5 to see what we did for the Greed LED and button)

Color	**On Color**			**Off Color**			**Sizes**	
	R	**G**	**B**	**R**	**G**	**B**	**Height**	**Width**
Red	255	0	0	91	0	0	160	60
Green	0	255	0	0	91	0	60	160
Blue	0	0	255	0	0	91	160	60
Yellow	255	255	0	91	91	0	60	160

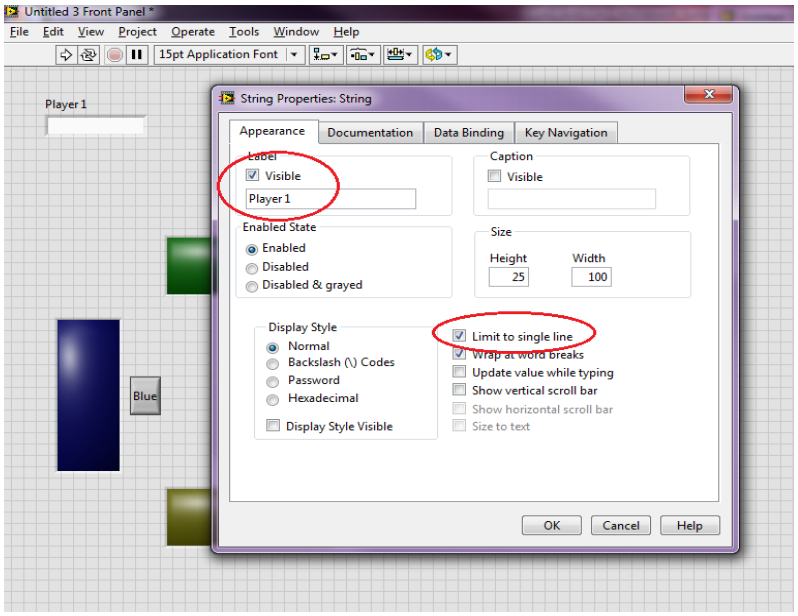
8. The dimension for the Blue and Red LEDs along with their respective Boolean buttons will have to be altered to give them the verticalcorrect shape as seen below. (*Right Click on the Button/LED > Properties*). The values for the Red and Blue LEDs and buttons height and width can be seen in the table below. Set these values in each control's properties.

Button Sizes		LED Sizes	
Height	Width	Height	Width
41	30	160	60

9. Select the LEDs and Buttons for both, Green and Yellow, and center them (*Align Objects>Horizontal Centers*). Repeat for the Blue and Red LEDs but with *Vertical Centers*.

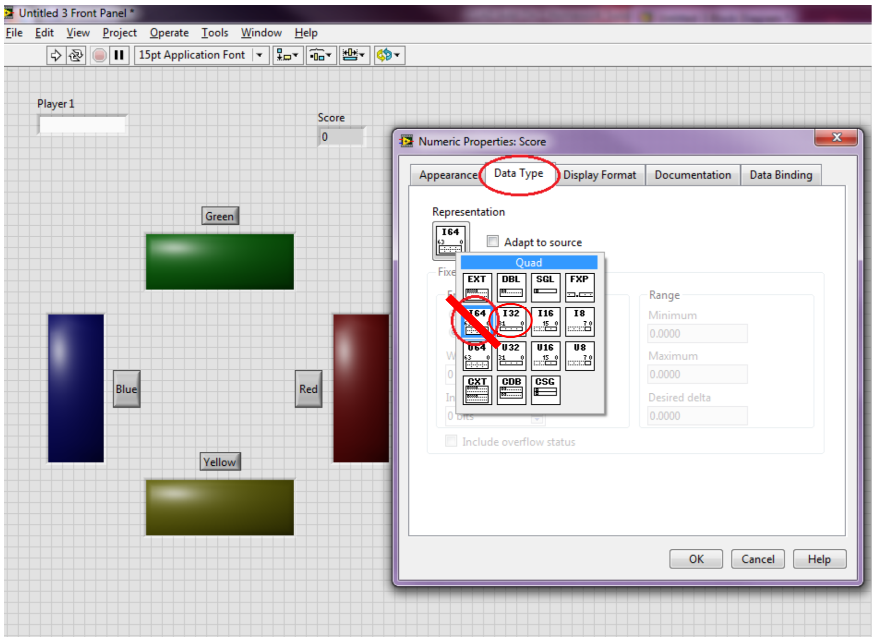


10. Now we will add a String Control (*Right Click > Controls palette > Modern > String & Path > String Ctl*) to hold the player's name and configure it to be limited to a single line (*Right Click on the String Control > Properties*).



Label
[✓] Visible
Player 1
Caption
[ ] Visible
Enabled State
(●) Enabled
( ) Disabled
( ) Disabled & grayed
Size
Height 25
Width 100
Display Style
(●) Normal
( ) Backslash (\) Codes
( ) Password
( ) Hexadecimal
[ ] Display Style Visible
[✓] Limit to single line
[✓] Wrap at word breaks
[ ] Update value while typing
[ ] Show vertical scroll bar
[ ] Show horizontal scroll bar
[ ] Size to text

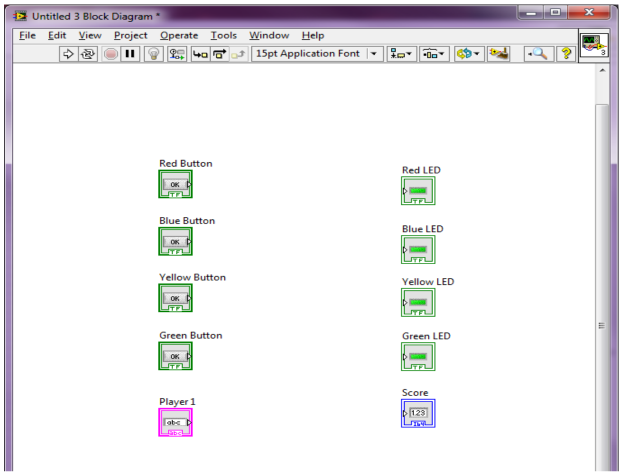
11. Finally, let's add a numeric indicator (*Right Click > Controls palette > Modern > Numeric > Numeric Indicator*), and configure it to be called score (this can easily be edited right when the indicator is placed on the front panel or after its initial placement by double clicking on the indicator) and to be integer. (To do the second part, go to *Data Type tab> Representation > I32)*



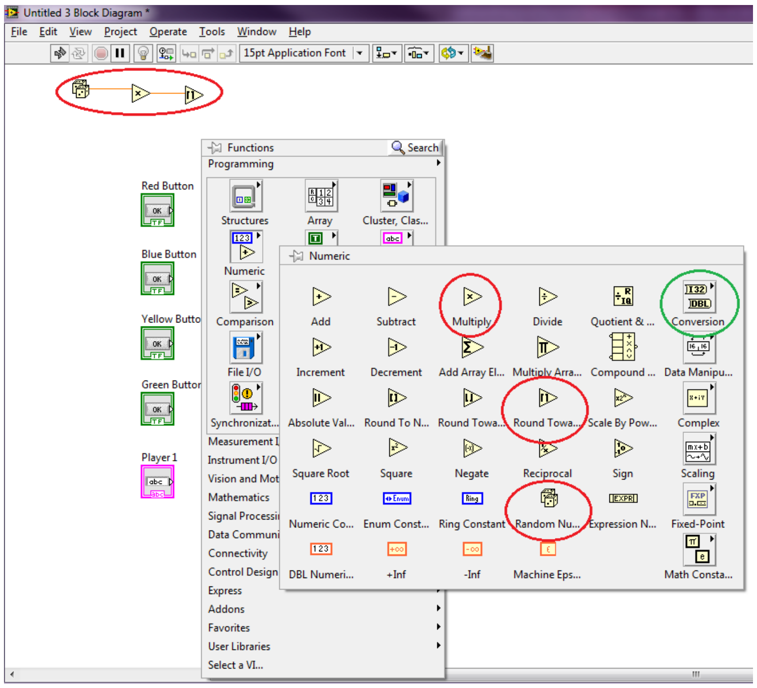
Representation
I64
[ ] Adapt to source
Range
Minimum 0.0000
Maximum 0.0000
Desired delta 0.0000
[ ] Include overflow status

Block Diagram

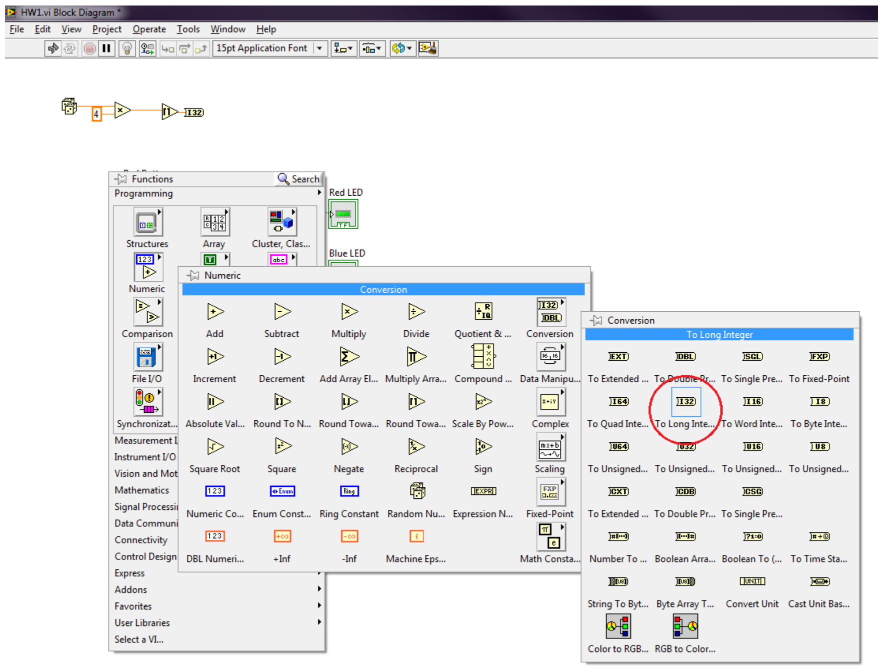
12. That was it for the Front Panel! Good job. Now let's work with the Block Diagram. First of all let's align all the controls and indicators (*Align Objects* on the toolbar like we did in past steps). Also, if your controls and indicators do not have the same labels as shown below, you need to edit their label names in the properties section. Do this by *Right Click on the Button/LED > Properties.*



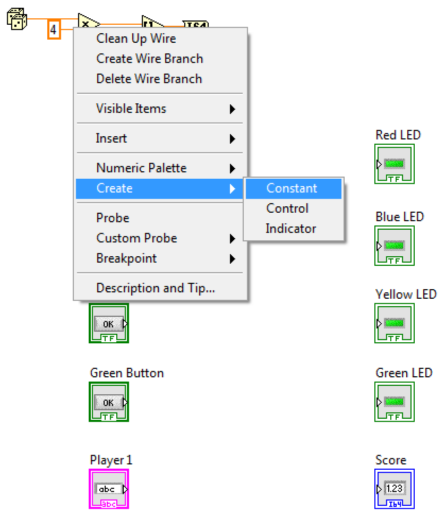
13. Add the functions Multiply, Round Toward +Infinity, and the Random Number (0-1) from the *Right Click > Function Palette > Programming > Numeric*; and connect them as shown in the figure below.



14. Add To Long Integer from *Function Palette> Programming > Numeric > Conversion > To Long Integer* (Click on the green icon from previous image and you will see the portion of the functions palette pictured below) and connect it.

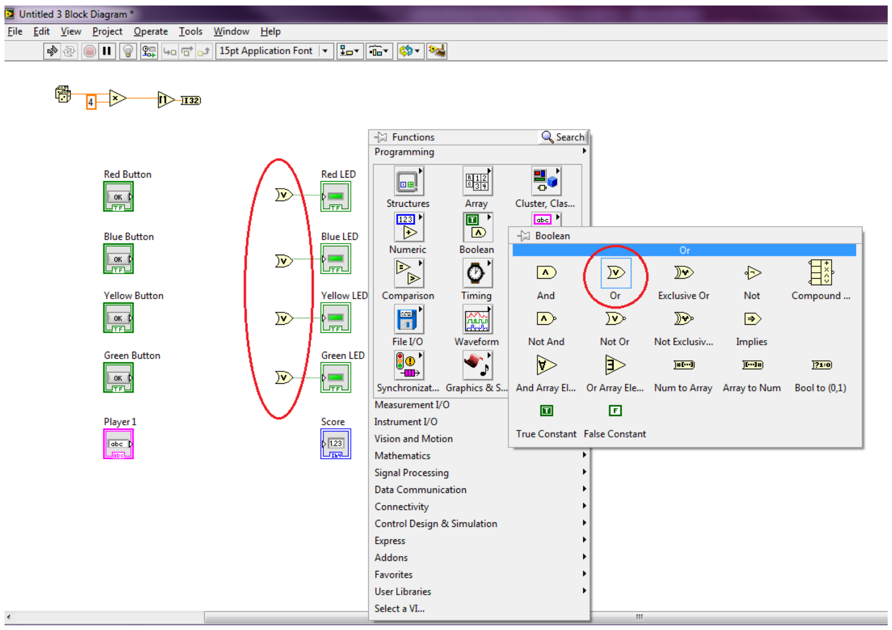


15. Add a Constant to the second input of the multiplier (*Right Click on Empty Input Terminal to Multiply > Create > Constant*) and write a 4 into it.

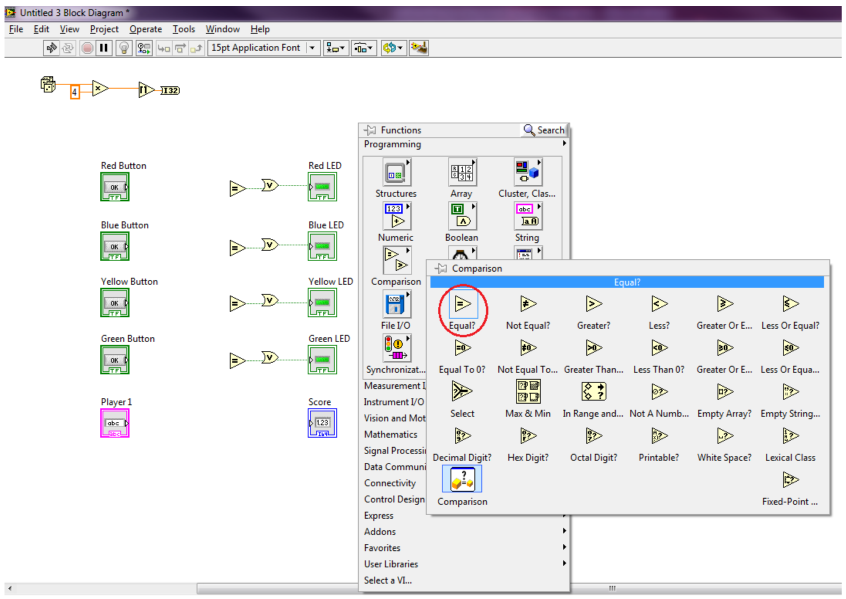


16. The section of code previously constructed allows us to generate a random number between 1 and 4 which is an integer with 32 bits of precision. It will be used to turn on one LED at a time

randomly. We also have buttons that the user can press to light up the LED. Therefore, we want the LED to turn on whenever the random number says so OR when the button is pressed. *Functions Palette > Programming > Boolean*, add the OR and connect it to the indicators

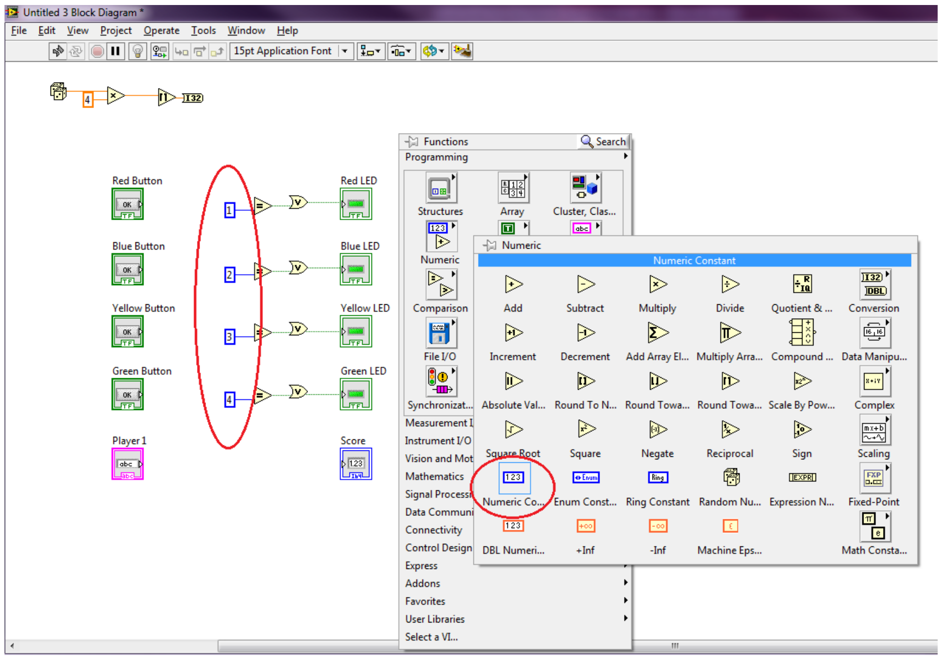


17. Since we have one random number selected in the range 1 to 4, we need to check which number was picked in order to turn on the correct LED. On *Functions Palette > Programming > Boolean*, add the Equal? for every single OR as seen below.

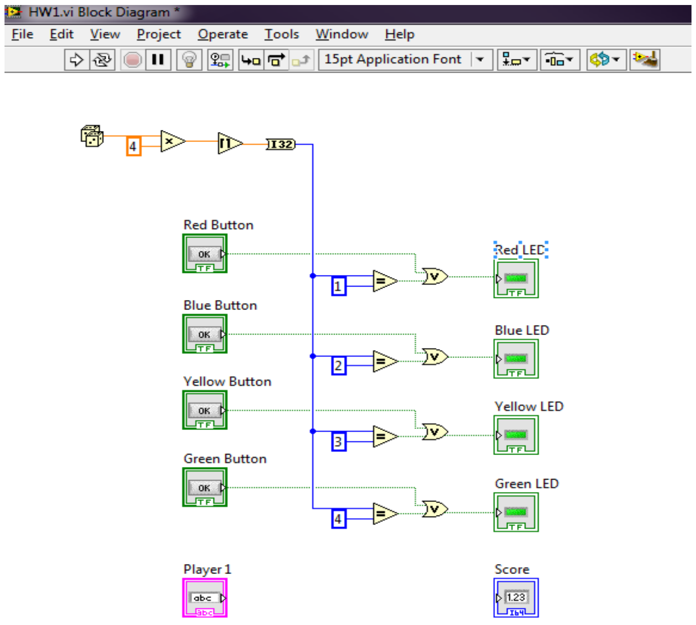


18. Next add a constant to the second terminal of the Equal? and connect them as shown in the image below. Since we need the type of the constant to be an integer, we have to manually

select the integer constant by *Palette> Programming > Numeric > Numeric Constant.* Be sure to make the numbers match those seen in the figure below.

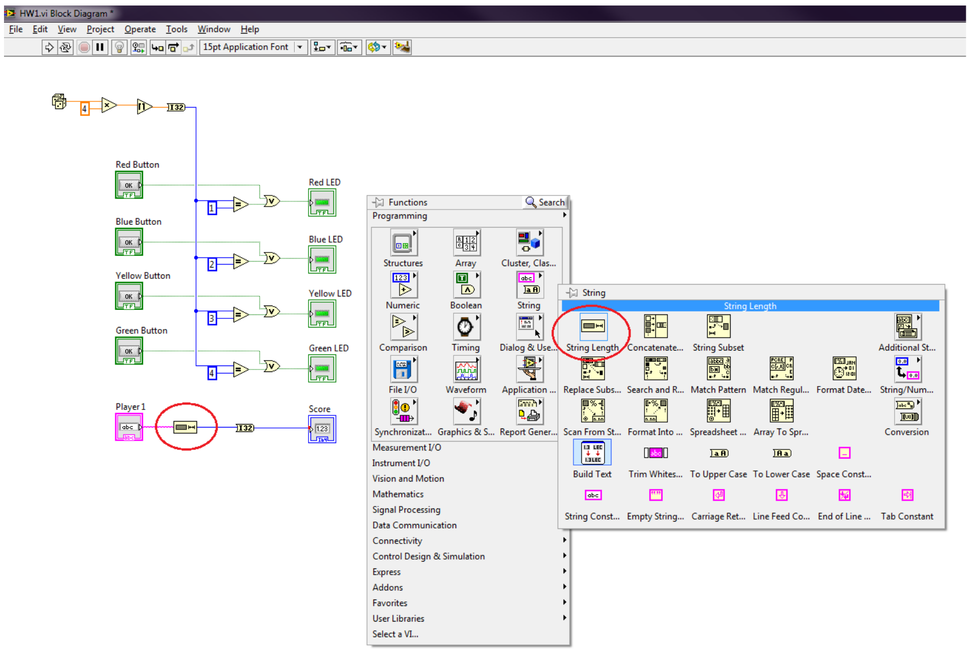


19. Finish connecting this section of the VI as seen below.

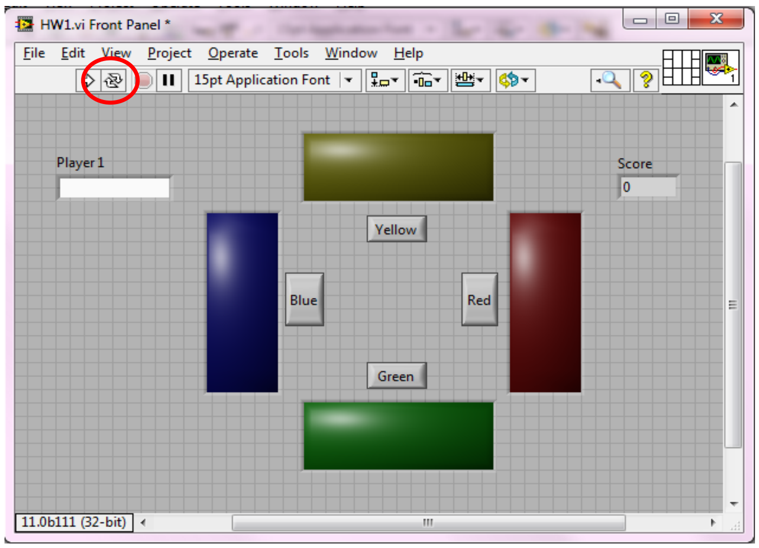


20. For the String control and the numeric indicator, we are going to count how many letters the String has and display it as the score. To do so add the String Length function from *Functions*

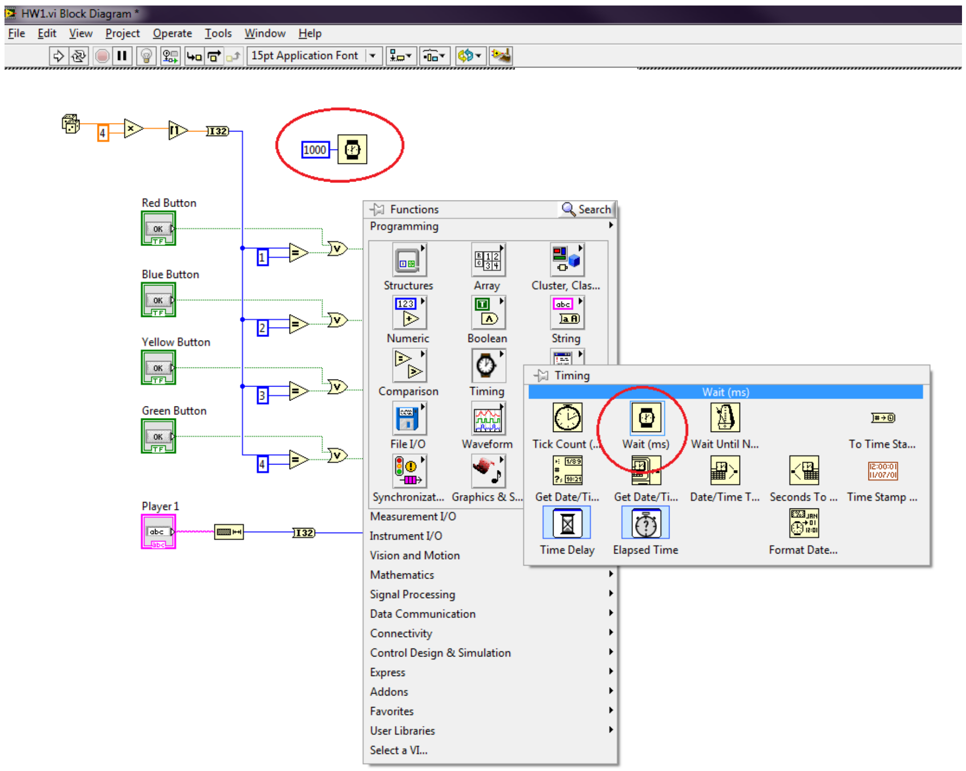
Palette > Programming > String > String Length. The result for this function is an unsigned number so we need to transform it to I32 (Add To Quad Integer from *Functions Palette > Programming > Numeric > Conversion, See Step 15*)



21. Now we are ready to try out our VI. Run it continuously by pressing the *Run Continuously* button. As you can observe, the simulation is too fast to be able to see the LEDs light up so we have to add some delay to it.



22. On the block diagram, add the Wait ms function from *Functions Palette > Programming > Timing > Wait ms* and add a Constant to it. Remember the constant will be in ms so to wait for 1 second you need to modify the constant to be 1000.



Optional

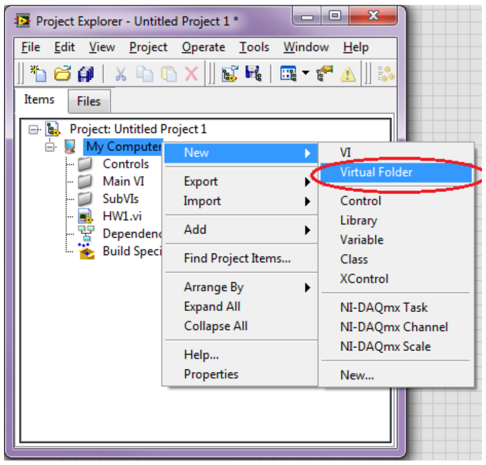
You can add as many features as you want, for example add a numeric control for the delay that accepts seconds and transforms them to ms on the block diagram. Modify the front panel by adding some decorations, a Background, or change the LEDs and buttons to be prettier. Have fun!

Creating a New Project

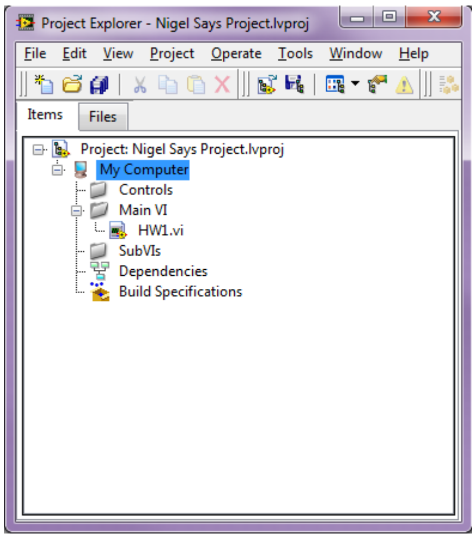
23. Save your project as Section1.vi.

24. Create a new project and add Section1 VI to it. With your VI still open go to *File > New Project > Add*.

25. Create three virtual folders under “My Computer” in the project window; name them “Main VI”, “SubVIs”, and “Controls”. A picture of this can be seen below.



26. Place Section1 VI in the “Main VI” virtual folder. The other two folders will be used in the future when you have SubVIs and Custom Controls within the Main VI. A Picture of what your final project window should look like can be seen below.


Project Explorer - Nigel Says Project.lvproj
File Edit View Project Operate Tools Window Help
Items Files
Project: Nigel Says Project.lvproj
My Computer
Controls
Main VI
HW1.vi
SubVIs
Dependencies
Build Specifications

27. Save your project as “Nigel Says Project”. From now on, you will be using this LabVIEW project to keep all of your code organized.