NUST School of Electrical Engineering and Computer Science

Faculty Member:_____ Date: _____

Semester:_____ Section: _____

Department of Electrical Engineering

EE379: Control Systems

LAB 2: Introduction to LabVIEW

Student name	Reg. No.	Lab Report Marks / 10	Viva Marks/5	Total/15

LAB 2: Introduction to LabVIEW

1. Objectives

i. Introduction to LabVIEW
ii. Designing a simple VI
iii. Introducing Control and Simulation Module in LabVIEW

2. Introduction to LabVIEW

Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) is a platform and development environment for a visual programming language from National Instruments. It is a very popular software tool among electrical engineers. LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications.

There is also another important difference between text-based programming languages and LabVIEW. In most text-based programming languages instructions determine program execution. LabVIEW uses dataflow programming, where the flow of data determines execution. This point will become clearer as you will work in LabVIEW.

3. Virtual Instruments (VIs)

LabVIEW programs are called virtual instruments (Vis) because their appearance and operation imitate physical instruments, such as oscilloscopes and millimeters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data. There are also many tools in LabVIEW that help you troubleshoot your program.

There are two parts of a VI, the front panel window and the block diagram.

3.1. Front Panel

The front panel is the user interface of the VI. You build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, pushbuttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display the data generated by the block diagram.

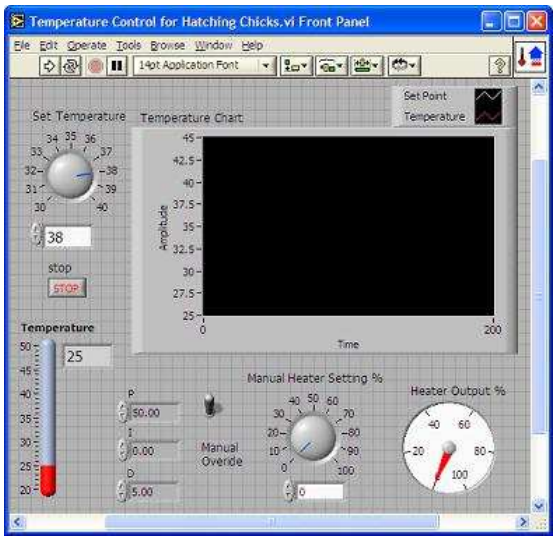


Figure: Front panel

3.2. Block Diagram

After you build the front panel, you need to add graphical code to describe what needs to be done when the controls/input devices are manipulated in the front panel. You also need to add graphical code for describing what information is displayed on the output terminals in the front panel. This graphical coding is done in the block diagram part of the VI.

The block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires are used to connect nodes in the block diagram, including control and indicator terminals, functions, and structures.

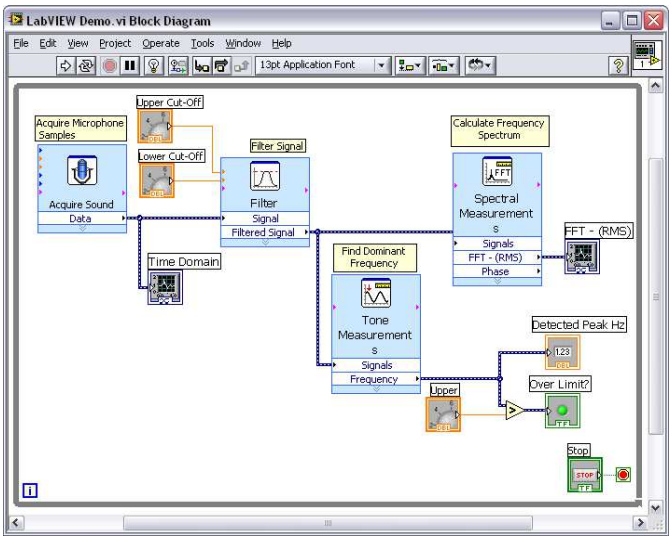


Figure: Block diagram

3.3. Controls and Indicators

You create the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data that the block diagram acquires or generates. The figures below show some of the controls and indicators

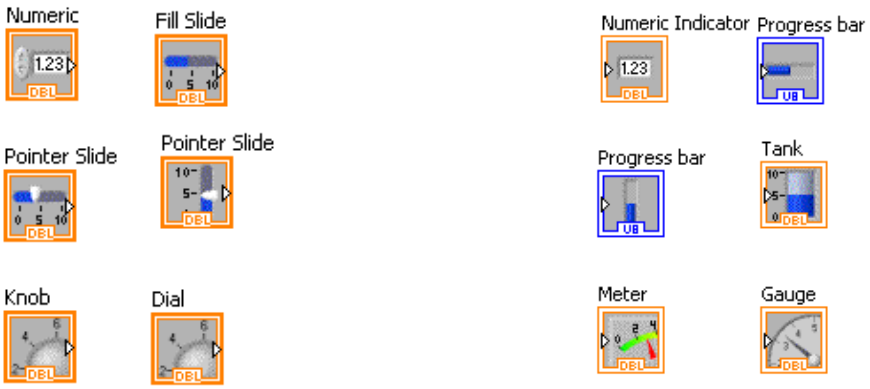


Figure: Controls and Indicator as seen in the block diagram

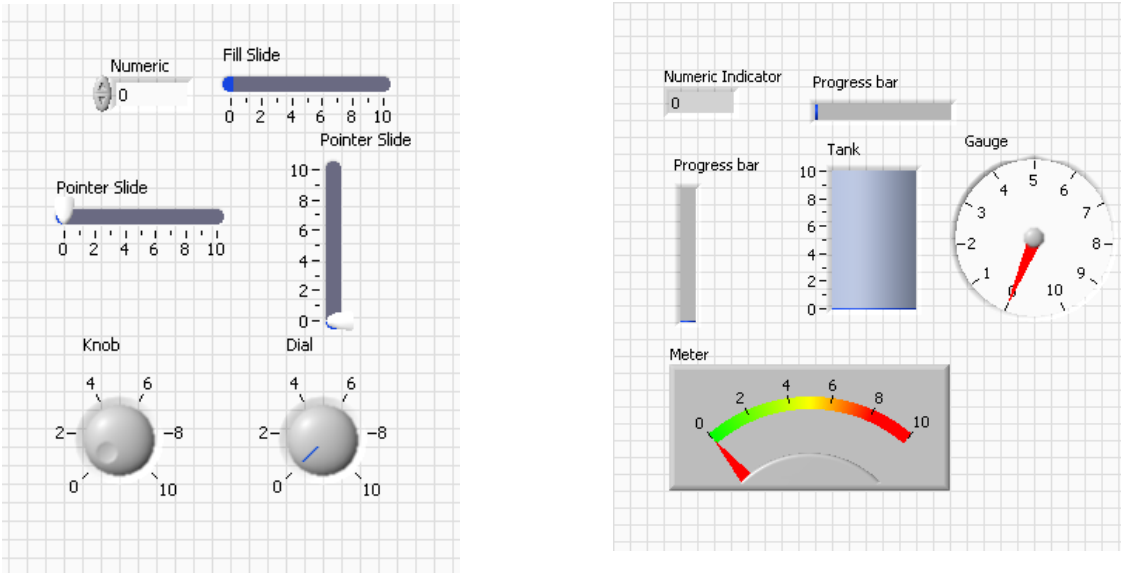


Figure: Controls and Indicator as seen in the front panel

4. Palettes

Palettes in LabVIEW are menus that contain commands to create or edit the front panel and the block diagram.

4.1. Tools Palette

The Tools palette is available on both the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. Use the tools to operate and modify front panel and block diagram objects.

Go to **View->Tools Palette** to display the Tools palette.

If automatic tool selection is enabled and you move the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the Tools palette. (It is recommended that automatic tool selection must always be enabled)

Figure: Tool Palette

4.2. Controls Palette

The Controls palette is available only on the front panel. The Controls palette contains the controls and indicators you use to create the front panel.

Go to **View->Controls Palette** or right-click the front panel workspace to display the Controls palette.

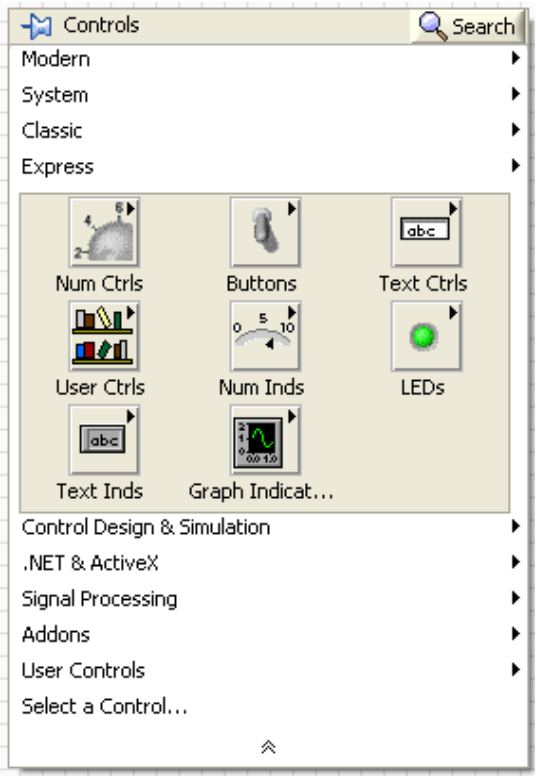


Figure: Control Palette

4.3. Functions Palette

The Functions palette is available only on the block diagram. The Functions palette contains the VIs and functions you use to build the block diagram.

Go to **View->Functions Palette** or right-click the block diagram workspace to display the Functions palette.

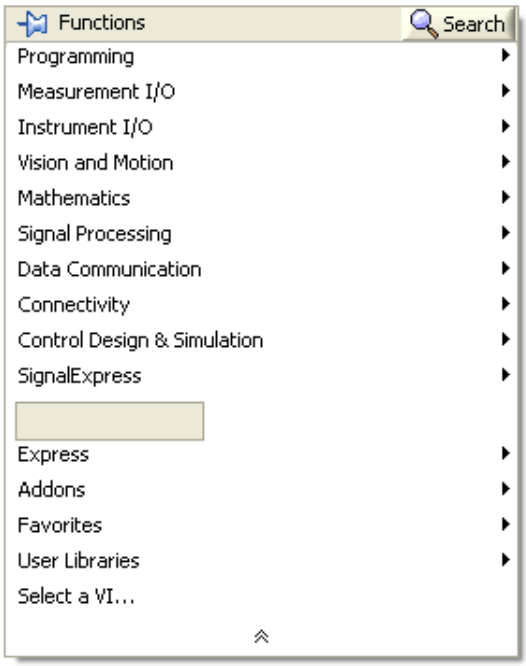


Figure: Function Palette

5. Wires

Data is transferred between block diagram objects through wires. Wires connect the controls and indicators. Each wire has a single data source, but you can wire it to many VIs and functions that read the data. Wires are of different colors, styles, and thicknesses, depending on their data type. The table below shows the most common types of wires.

Data Type	Scalar	1D Array	2D Array	Color
Numeric - Floating Point				Orange
Numeric - Integer				Blue
Boolean				Green
String				Pink

Table: Common Wire Types

There is also another type of wire called the broken wire. It appears as a dashed black line with a red X in the middle, as shown in the figure below. Broken wires occur for a variety of reasons, such as when you try to wire two objects with incompatible data types.

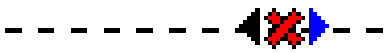


Figure: Broken wire

6. Shortcuts Keys

The table below lists some of the useful shortcut keys available in LabVIEW

Key	Function
Ctrl+N	New VI
Ctrl+Q	Quit LabView
Ctrl+B	Remove Broken wires
Ctrl+R	Run VI
Ctrl+.	Abort VI
Ctrl+E	Display Block/Front Diagram
Ctrl+T	See Block/Front diagram at same time
Ctrl+H	Display Context Help
Ctrl+Shift+L	Lock Context Help

Table: Common Shortcuts keys

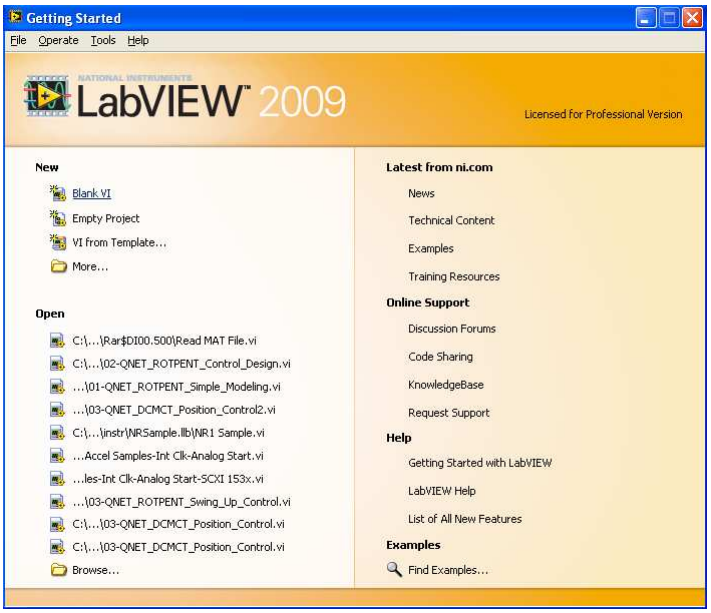
7. Creating a basic VI

The use of LabVIEW is best understood by creating a basic VI. We will proceed step by step to create a VI that performs some very simple functions. We will do the following tasks

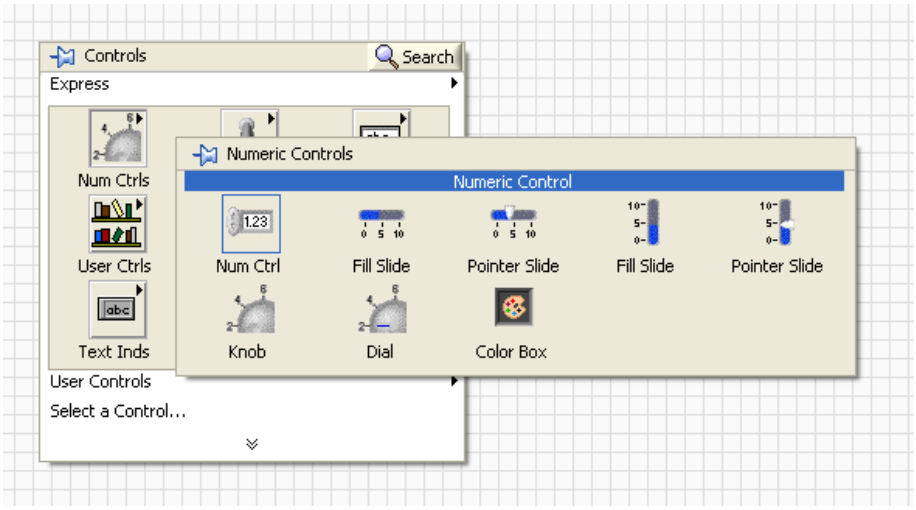
- We will start by adding two numbers entered by use.
- Then we will add two randomly generated numbers.
- How to continuously run the program in LabVIEW.
- Use of While Loop in LabVIEW
- Adding delay in While Loop
- Conversion from While Loop to For Loop.
- Using For Loop as conditional Loop
- Displaying the message as per the condition.
- Using Case Structure to glow the LED as per condition.

Follow the steps given below to create a basic VI:

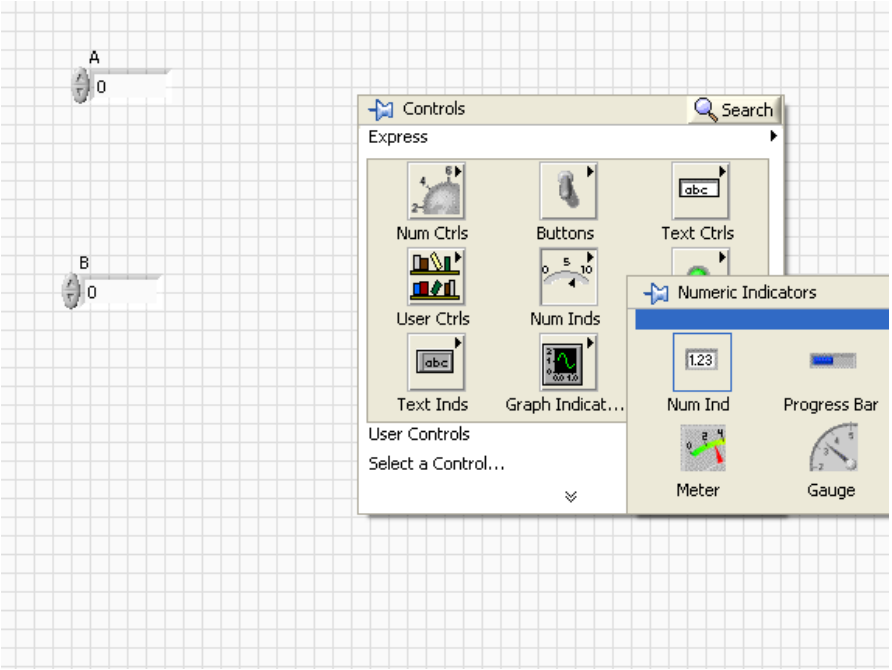
1. Go to Start >> Programs >> LabVIEW 2009. The window shown below will be displayed. Click on Blank VI to start programming.



2. In the front panel Right click >> Control Palette>> Express >> Numeric Controls. Drag two Numeric Controls to front panel and name them “A” and “B” respectively.

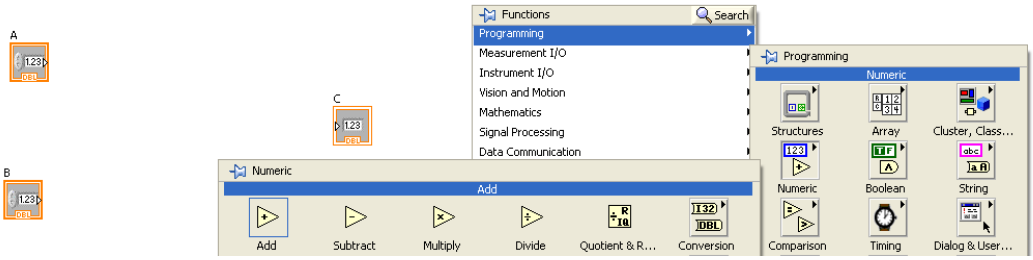


3. Go to Express >> Numeric Indicators. Drag a numerical indicator in front panel and name it “C”.

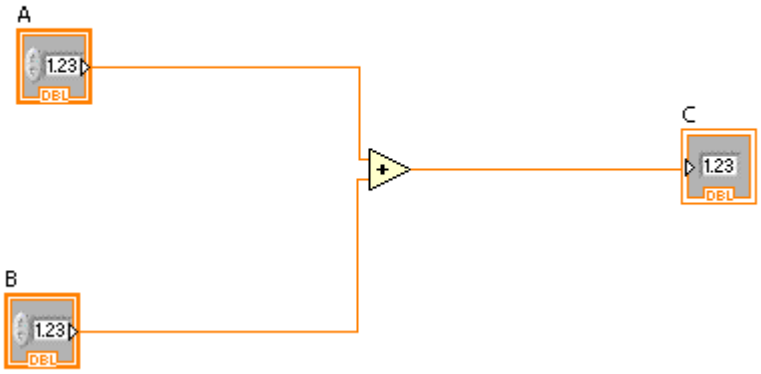


4. Switch to the block diagram. (Switch between front panel and block diagram by pressing Ctrl+E). The block diagram will look like the figure given below

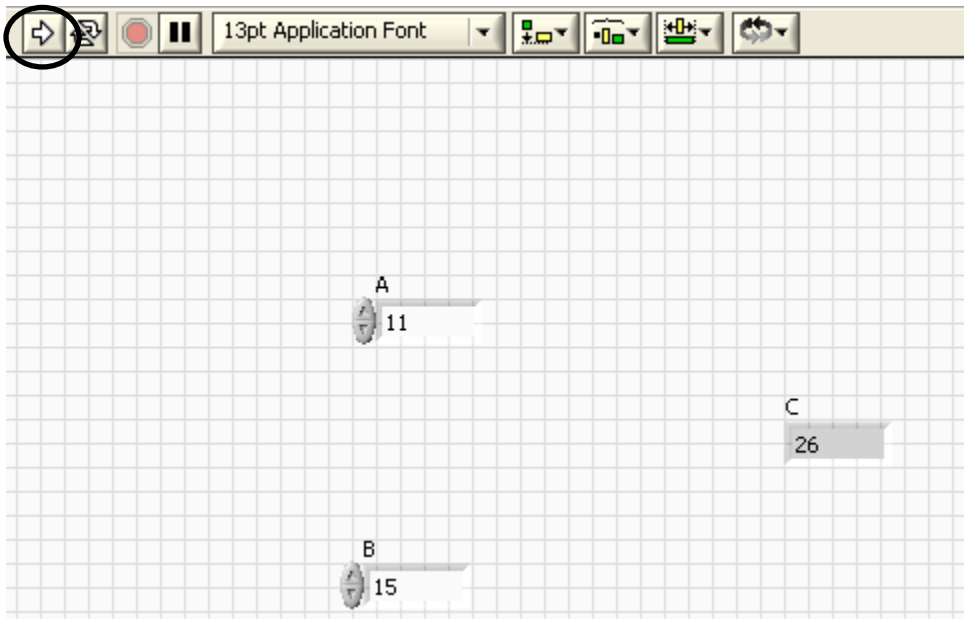
5. Now to perform the numeric operation Right click in block diagram, function palette will be displayed, go to Programming >> Numeric. Drag Add in your block diagram



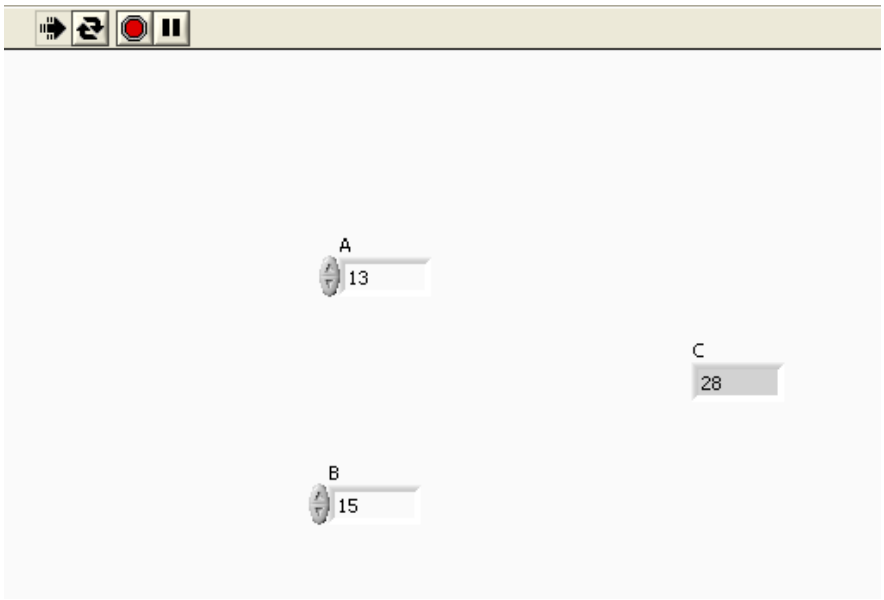
6. Now connect the wires in your Block diagram as shown in the figure below



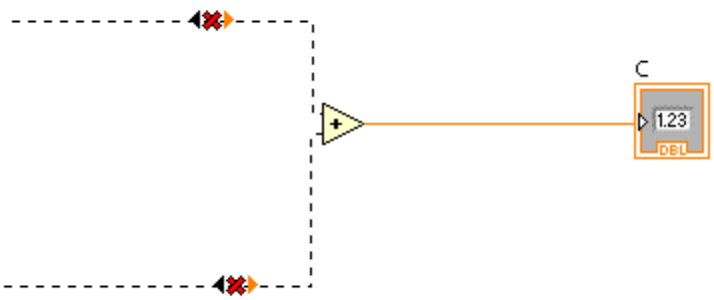
7. Switch to the front panel and Run the program by clicking the white arrow



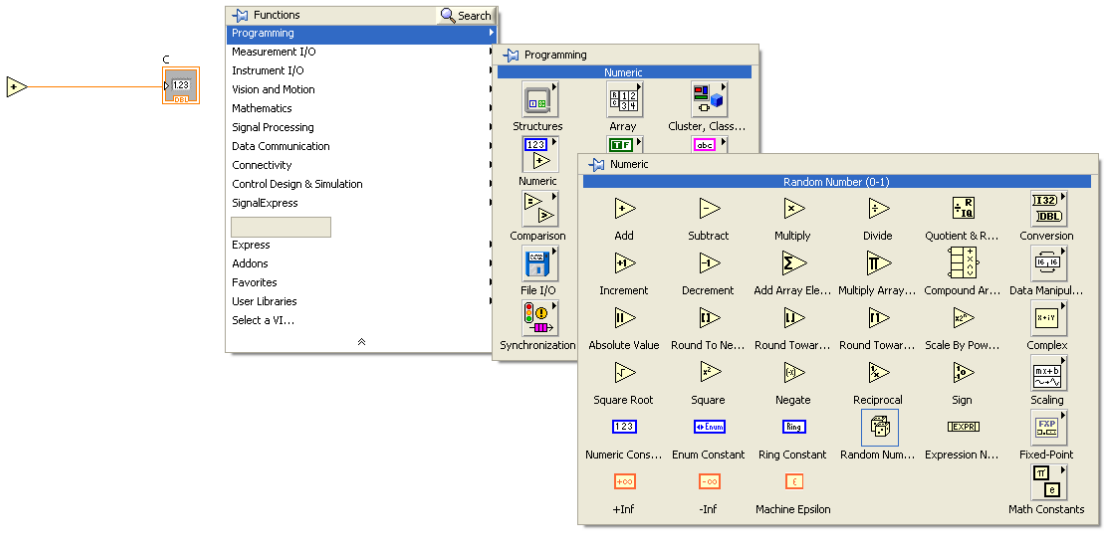
8. Enter the values of A and B and resultant C will be shown.



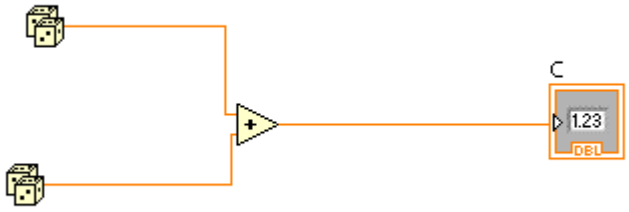
9. Delete A and B.



10. Drag two Random Numbers in your block diagram by going to Programming >> Numeric >> Random Number

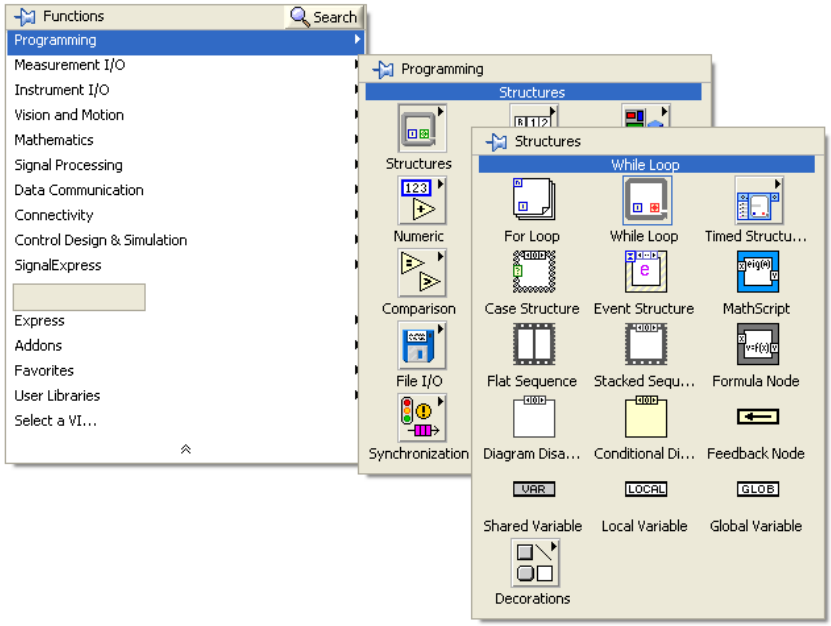


11. Connect the wires and block diagram will look like

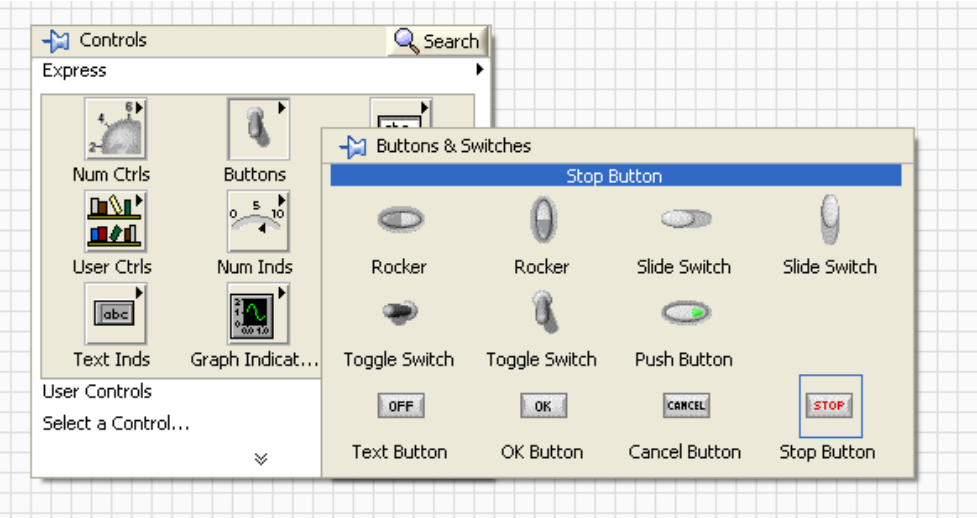


12. Run the program a few times and see what happens.

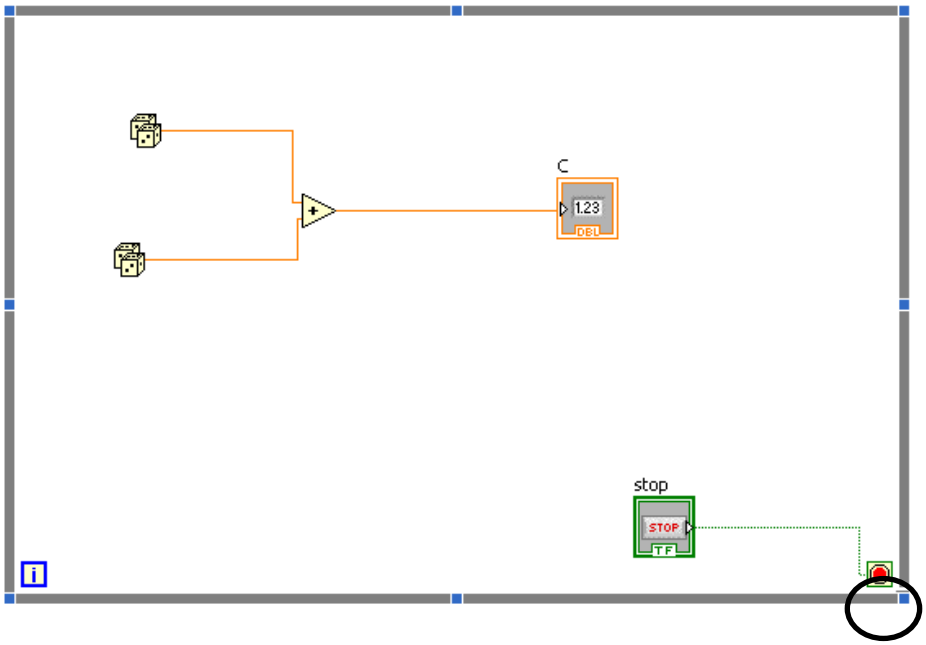
13. Now we will insert a loop in the program. To add the loop go to Programming >> Structures >> drag While Loop in your block diagram.



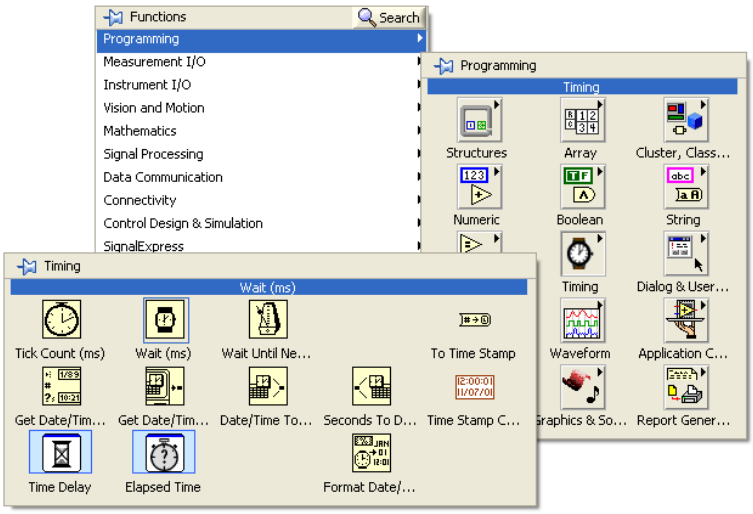
14. While Loop will only be terminated if the required condition is met, in our case we simply add a stop button by switching to the front panel and going to Express >> Buttons >> Drag Stop button in your front panel.



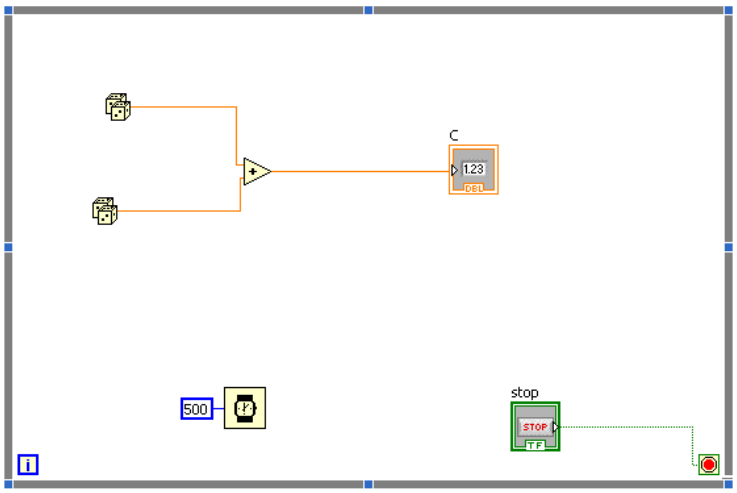
15. Connect the stop button with terminate point to complete the while loop.



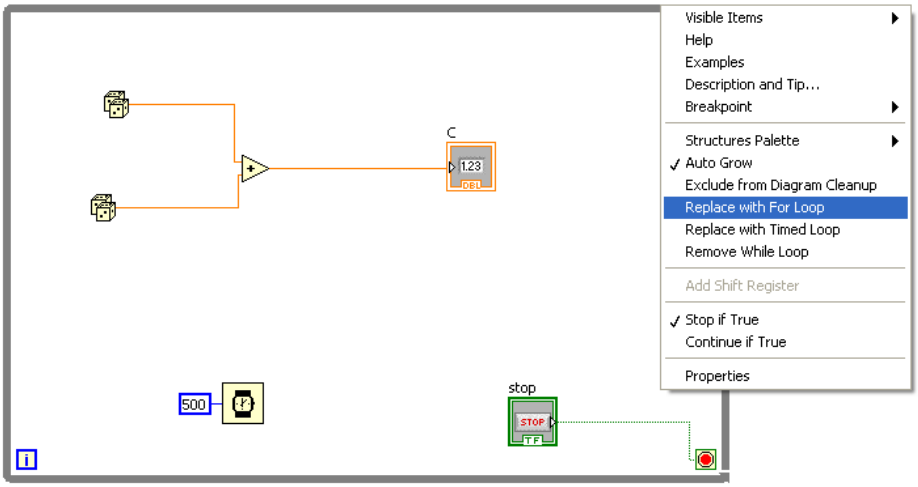
16. Now when we execute the program, we will notice that the program is running at very high speed and we cannot see the output. We can add delay to program by going to Programming >> Timing >> drag wait to your block diagram. (you can also use other blocks within Timing for adding delay)



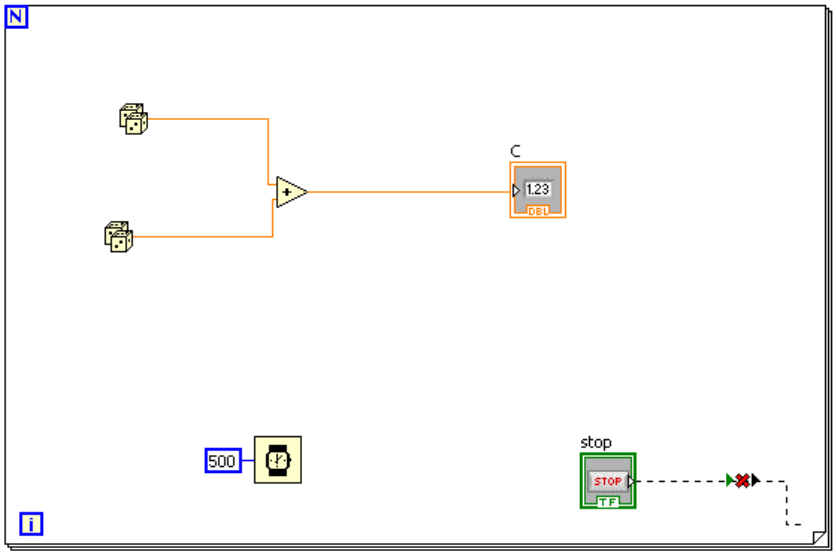
17. Right click the Wait and go to create >> constant and enter 500. (Delay of 500 ms will be added)



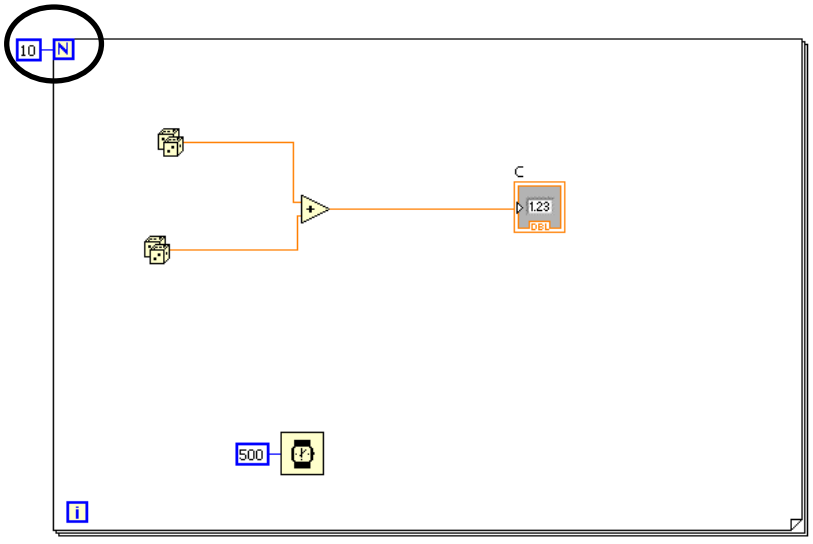
18. To replace while loop with For loop, right click the loop and select “Replace with For Loop”



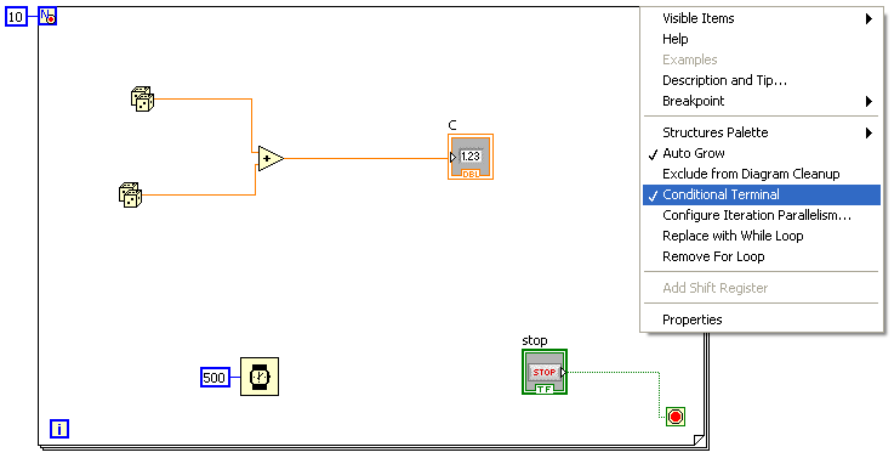
19. Now remove the conditional terminal as it may not be required in for loop.



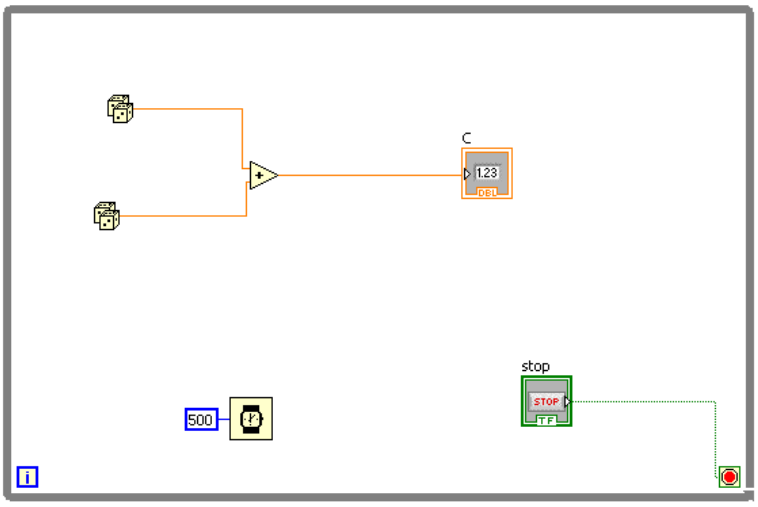
20. Add the no of times you want the loop to run by right clicking on “N” >> create >> constant and enter 10.



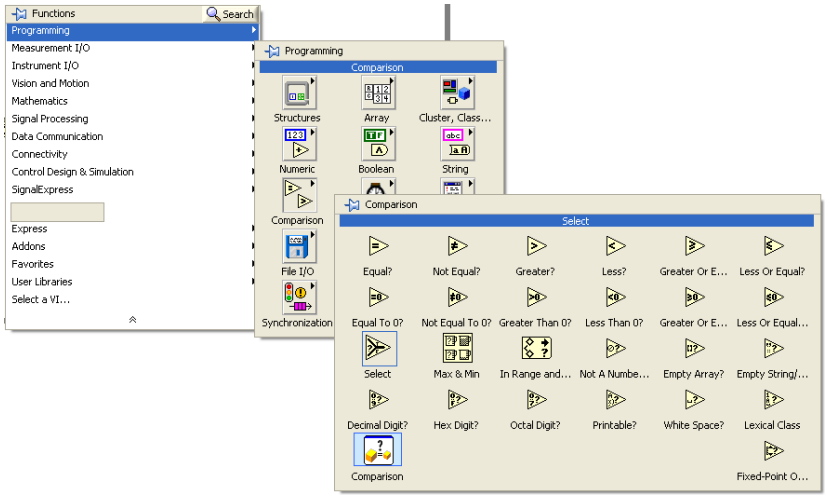
21. To run the for loop conditionally, right click the loop and select “Conditional Terminal” and connect the stop button with it. Now the loop will run 10 times but can be terminated by pressing the stop button.



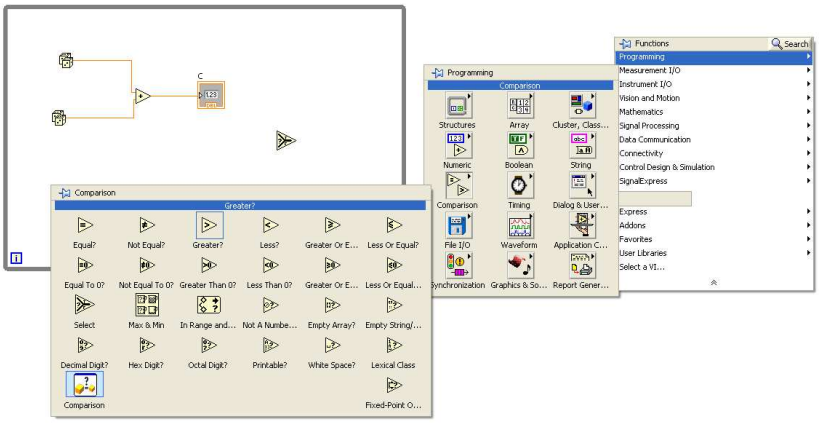
22. Replace the for loop with the while loop.



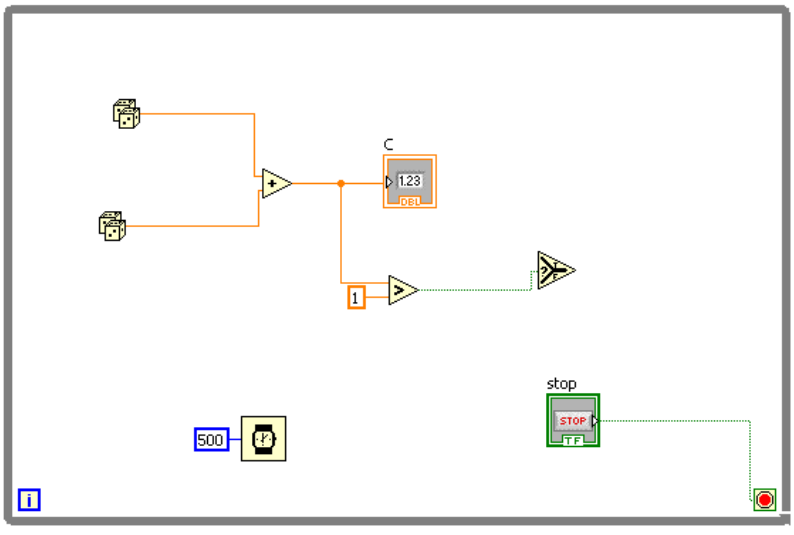
23. Now we want that if the sum of two random numbers is greater than 1 than a string message should be displayed on the front panel quoting “Greater than 1” or “Less than 1” for the other condition. We can do this by using the “Select” block in Comparison. Go to Programing >> Comparison >> drag select into your block diagram.



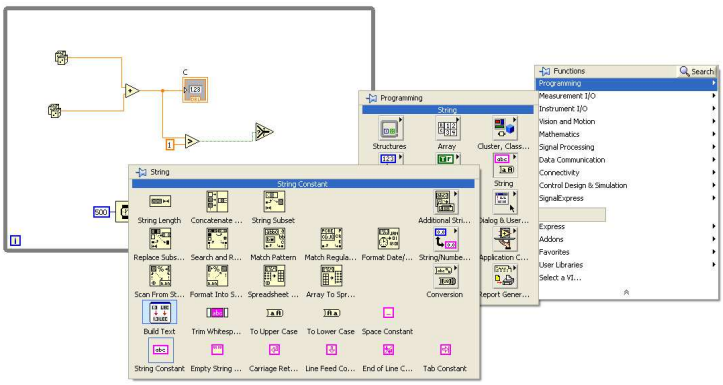
24. To check that whether the sum is greater than 1 or not select “Greater” from comparison under programming and connect one terminal of that with the output of the add terminal and other with a constant 1.



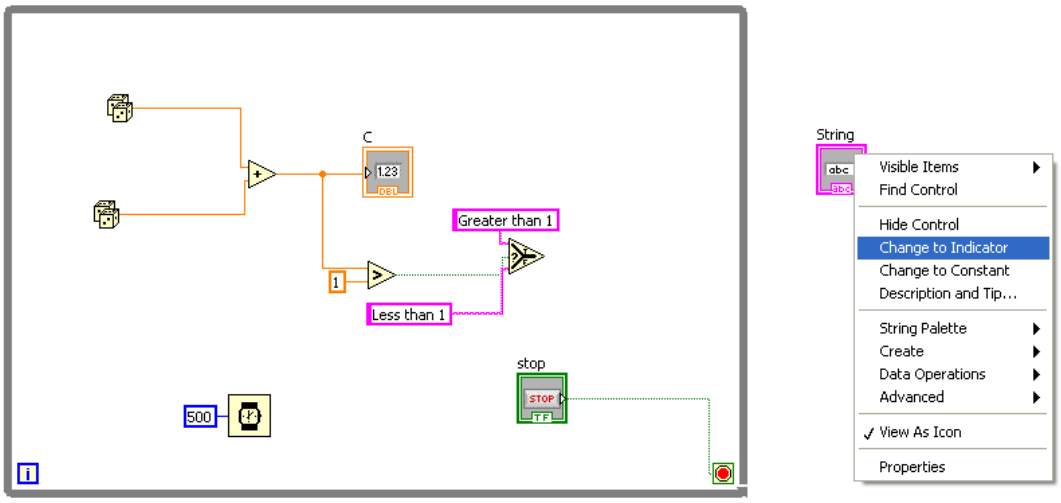
25. “Select” has three input terminal, select, true and false. Select terminal is the conditional terminal which checks which condition has been invoked, if true has been invoked the block attached with true terminal will be executed and false will be executed for the other condition.



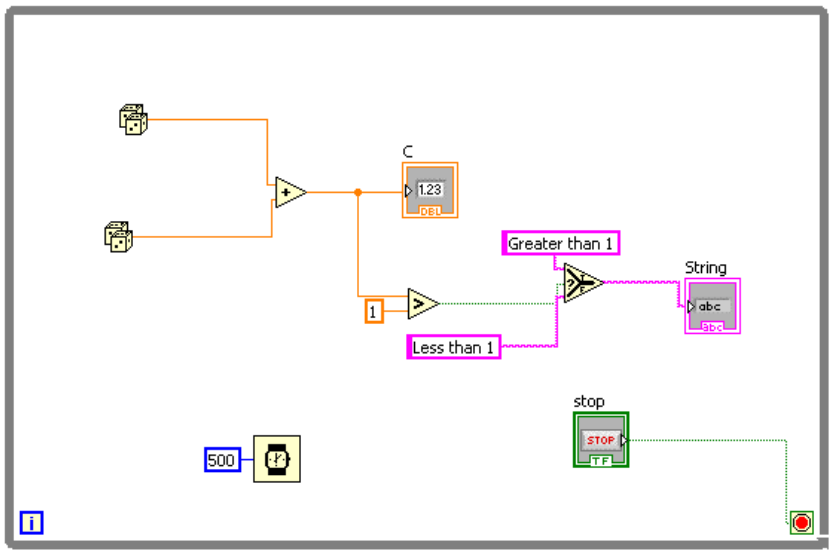
26. We will add the input string saying “Greater than 1” with true terminal and ‘Less than 1” with false terminal. To do this right click in block diagram and go to Programming >>Strings >>String constant. Drag two string constants in the block diagram and enter the respective messages.



27. From the front panel select a “String Control” within Express. Now to change the string control to indicator, right click on that block and select “Change to Indicator”

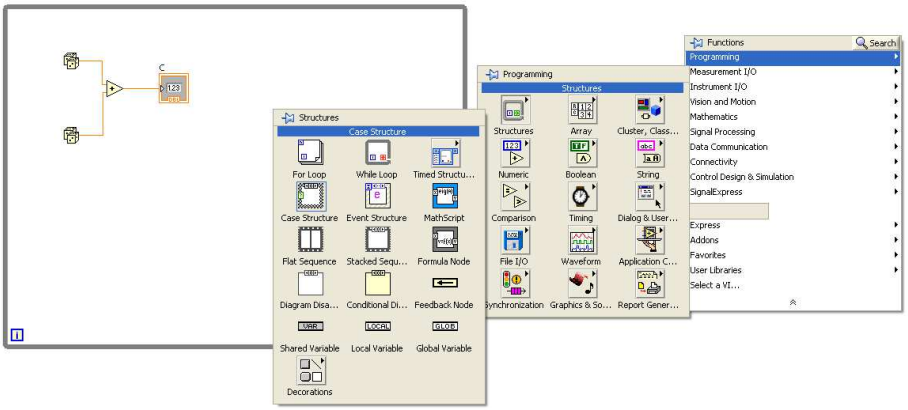


28. Wire the respective terminals to complete the program.

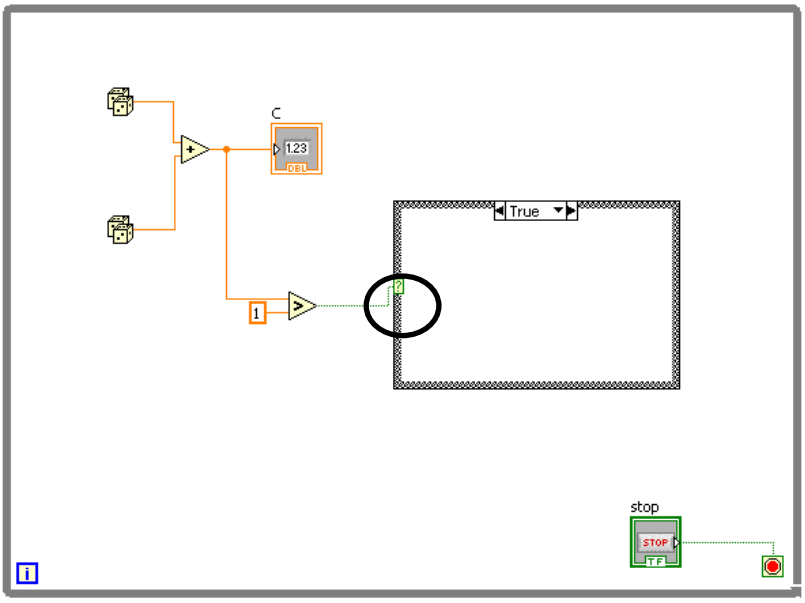


29. Now when you run the program, you will see the respective message as per condition.

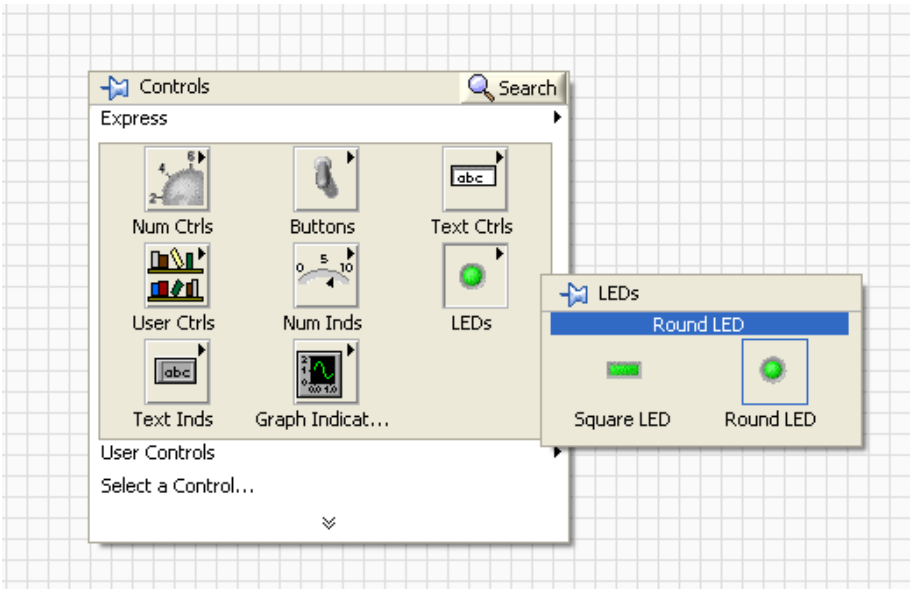
30. Now we will do the same by using the “Case Structure”, the only difference will be that instead of string message, a LED will show whether the sum is greater or less than 1. To do this go to Programming >> Structures >> drag Case Structure into the block diagram.



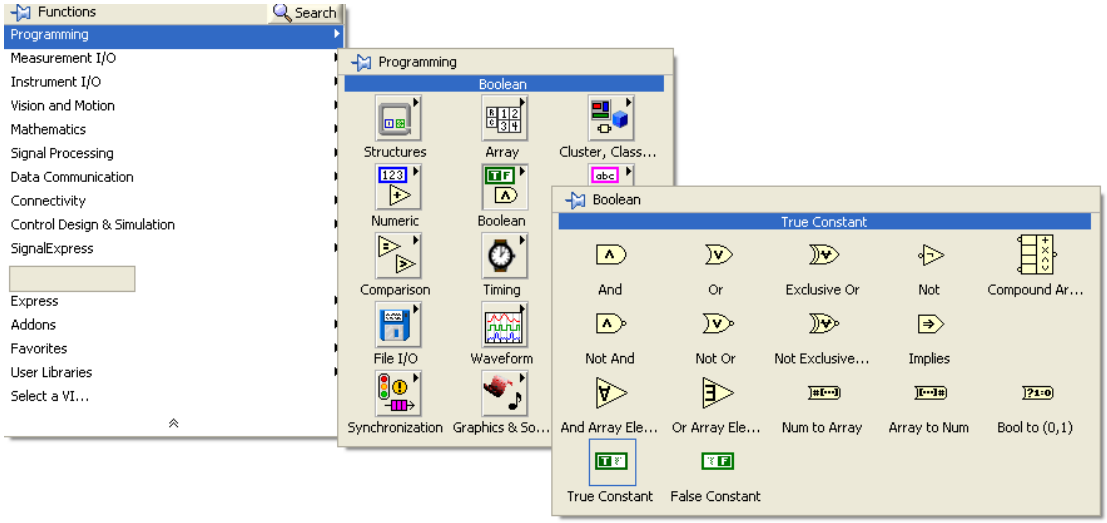
31. Wire the output terminal of “Greater” with the conditional terminal of Case Structure.



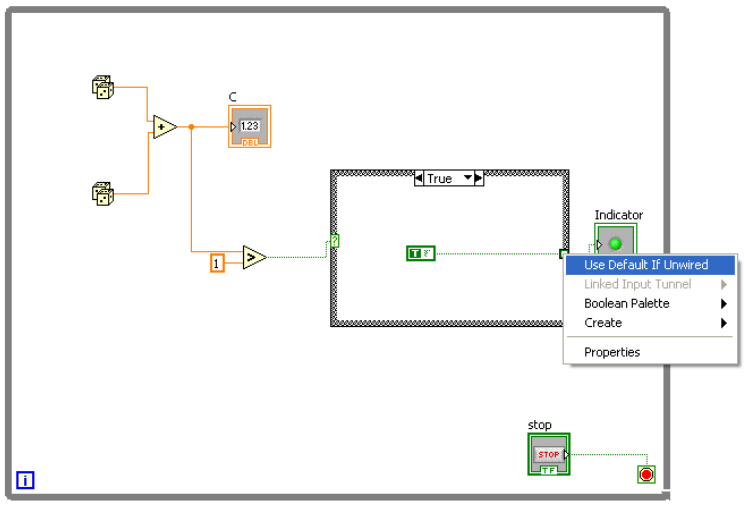
32. Select a LED from front panel by going to Express >> Buttons >>Round LED.



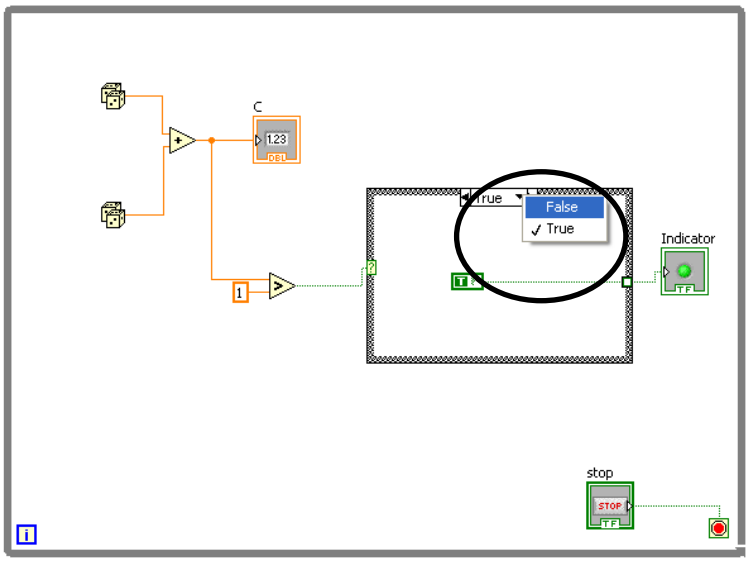
33. When the condition is true LED will glow to indicate it. To do that drag a true constant into the "TRUE" case of the structure by going to Programming >> Boolean >> True constant.



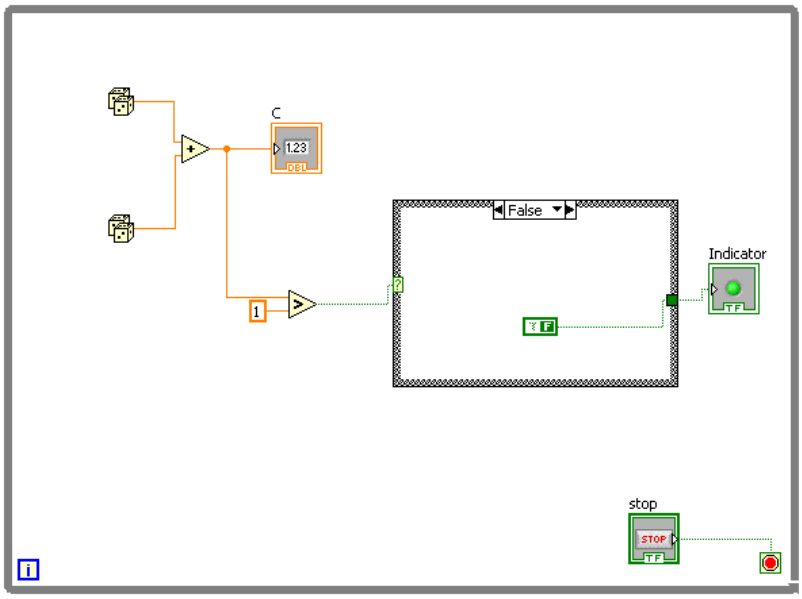
34. Connect the true constant with LED. (Right click the case structure at the terminal which is connected to LED and select “Use default if unwired”.



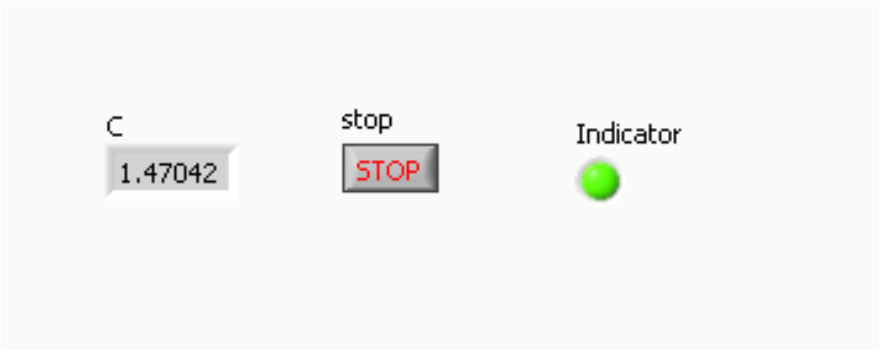
35. Now go to False structure.



36. Connect the false constant with LED.



37. Run the program, output will be displayed as per condition.



8. Transfer function LabVIEW

We have seen how to create a basic VI in LabVIEW. Now we will see how to create a transfer function in LabVIEW.

Transfer Function in LabVIEW can be inserted using “Control and Simulation Toolkit”.

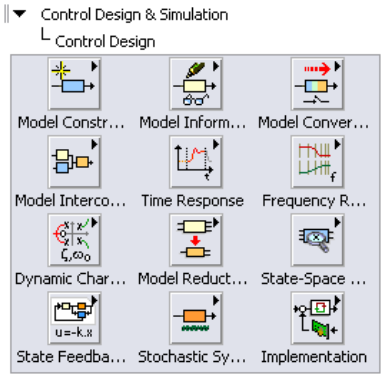
Use the following steps to construct an example transfer function:

$$\frac{K_m}{J_{eq}R_m s + K_m{}^2}$$

i. Open a blank VI and save it.

ii. Switch to the block diagram (<Ctrl+E>).

iii. Place the CD Construct Transfer Function Model.vi on the block diagram. From the functions palette, select Add-ons » Control Design » Model Construction » CD Construct Transfer Function Model.vi and place it on the block diagram. From the drop-down menu select Single-Input Single-Output (Symbolic).



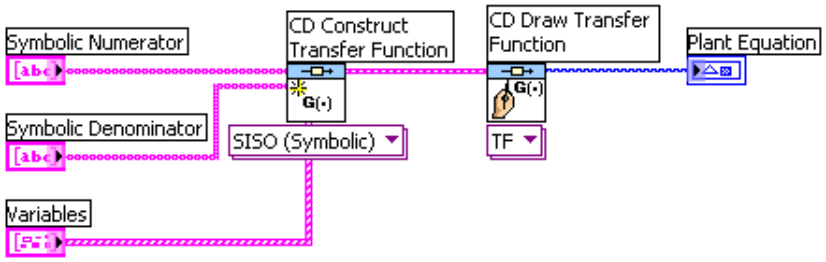
iv. Right-click the Symbolic Numerator input terminal of the CD Construct Transfer Function Model.vi and select Create » Control from the shortcut menu.

v. Right-click the Symbolic Denominator input terminal of the CD Construct Transfer Function Model.vi and select Create » Control from the shortcut menu.

vi. Right-click the Variables input terminal of the CD Construct Transfer Function Model.vi and select Create » Control from the shortcut menu.

vii. Place the CD Draw Transfer Function Equation.vi on the block diagram. From the functions palette, select Add-ons » Control Design »Model Construction » CD Draw Transfer Function Equation.vi and place it on the block diagram. From the drop-down menu select Transfer Function.

viii. Wire the Transfer Function Model output terminal of the CD Construct Transfer Function Model.vi to the Transfer Function Model input terminal of the CD Draw Transfer Function Equation.vi.



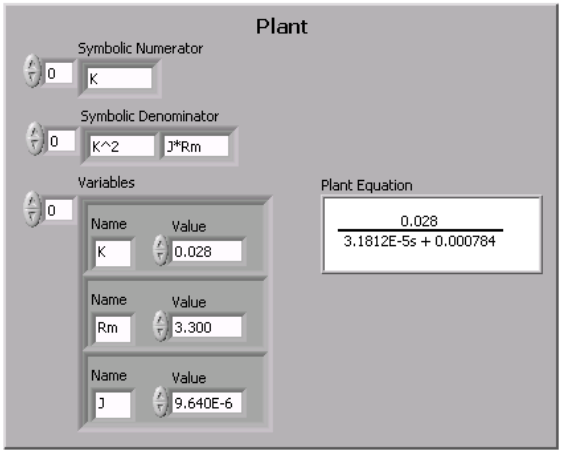
ix. Right-click the Equation output terminal of the CD Draw Transfer Function Equation.vi and select Create » Indicator from the shortcut menu.

x. Double-click the label of the picture indicator, and rename it Plant Equation.

xi. Save the VI.

xii. Run the VI.

xiii. On the front panel, change the Symbolic Numerator, Symbolic Denominator, and Variables controls to see the output.



Exercise 1

Create a VI for indicating extreme temperatures. Inputs, outputs and requirements of VI are:

Inputs

User will enter a Maximum Temperature, a Minimum Temperature and the Current temperature (Maximum and Minimum Temperatures will be in Celsius while the current temperature will be in Fahrenheit)

Outputs

i. Current temperature in Celsius
ii. String message, indicating the state of the system
iii. LED, indicating warning.

Requirements

i. If the current Temperature is below the **Minimum Value**: "Freezing Warning" message should be displayed.
ii. If the current Temperature is above the **Maximum value:** "Heat Stroke Warning" message should be displayed.
iii. If the current Temperature is within **limits**: "System OK" message should be displayed.
iv. A LED should turn red to indicate a warning message.

Exercise 2

Create a VI for the following transfer functions:

$$\frac{30s-180}{(s^3+4s^2+13s+7)}$$

$$\frac{s^3+s+1}{s^3+s^2+6}$$