Hands-on Lab 1:

LabVIEW – Basics 1

This lab reviews LabVIEW concepts needed to interface with the LEGO NXT. Specific LabVIEW concepts include: creating a VI (front panel and block diagram), case structures, while loops and shift registers.

LabVIEW 2012 for LEGO Mindstorms provides (1) many of the common controls found in standard LabVIEW; and (2) controls specific to the NXT like motors and sensors. The net effect is a rapid-development software platform to create NXT-based programs – especially those involving a PC-based GUI (Graphical User Interface). GUIs enable one to create dials (e.g. to adjust motor power), switches (e.g. to turn on a motor), and charts (to display sensor data).

Preamble: These steps will be common to all future NXT Lego LabVIEW programs. All programs begin by double-clicking the LabVIEW 2012 for LEGO Mindstorms icon. The following splash screen appears (**Figure A1**). To create a new program, click `File-New` (**Figure A2**)

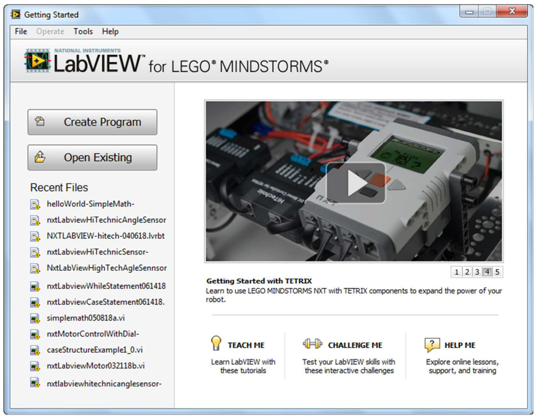

Getting Started
File Operate Tools Help
NATIONAL INSTRUMENTS LabVIEW for LEGO® MINDSTORMS®
Create Program
Open Existing
Recent Files
helloWorld-SimpleMath-
nxtLabviewHiTechnicAngleSensor
NXTLABVIEW-hitech-040618.lvrbt
nxtLabviewHiTechnicSensor-
NxtLabViewHighTechAgleSennsor
nxtLabviewWhileStatement061418
nxtLabviewCaseStatement061418.
simplemath050818a.vi
nxtMotorControlWithDial-
caseStructureExample1_0.vi
nxtLabviewMotor032118b.vi
nxtlabviewhitechnicanglesensor-
1 2 3 4 5
Getting Started with TETRIX
Learn to use LEGO MINDSTORMS NXT with TETRIX components to expand the power of your robot.
TEACH ME
Learn LabVIEW with these tutorials
CHALLENGE ME
Test your LabVIEW skills with these interactive challenges
HELP ME
Explore online lessons, support, and training

Figure A1: Splash Screen

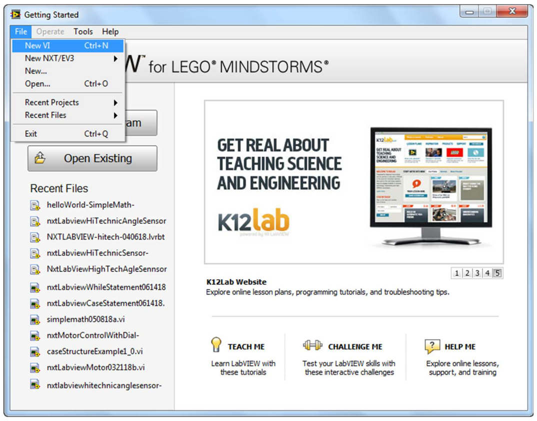

Getting Started
File Operate Tools Help
New VI Ctrl+N
New NXT/EV3
New...
Open... Ctrl+O
Recent Projects
Recent Files
Exit Ctrl+Q
W for LEGO® MINDSTORMS®
am
Open Existing
Recent Files
helloWorld-SimpleMath-
nxtLabviewHiTechnicAngleSensor
NXTLABVIEW-hitech-040618.lvrbt
nxtLabviewHiTechnicSensor-
NxtLabViewHighTechAgleSennsor
nxtLabviewWhileStatement061418
nxtLabviewCaseStatement061418.
simplemath050818a.vi
nxtMotorControlWithDial-
caseStructureExample1_0.vi
nxtLabviewMotor032118b.vi
nxtlabviewhitechnicanglesensor-
GET REAL ABOUT TEACHING SCIENCE AND ENGINEERING
K12lab
1 2 3 4 5
K12Lab Website
Explore online lesson plans, programming tutorials, and troubleshooting tips.
TEACH ME
Learn LabVIEW with these tutorials
CHALLENGE ME
Test your LabVIEW skills with these interactive challenges
HELP ME
Explore online lessons, support, and training

Figure A2: Click File-New to create new program

A front panel and block diagram will pop up. Depending on your monitor's size, you may wish to resize them with your mouse as shown in **Figure B1**. The front panel will be your GUI and serves as your canvas to add virtual controls like knobs, indicators, and charts. Programming occurs in the block diagram by wiring up virtual controls.

Now is the time to name your program. Refer to **Figure B2** (`File – Save As`) to choose and create a suitable folder where you'll keep all your future VI files (e.g. a USB thumb drive). Now you're ready to create new NXT LabVIEW programs.

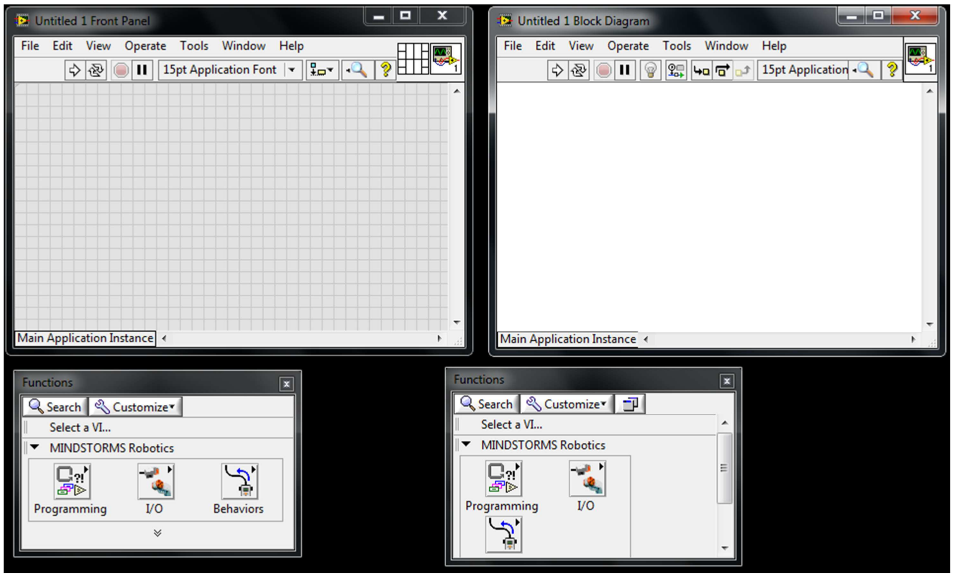
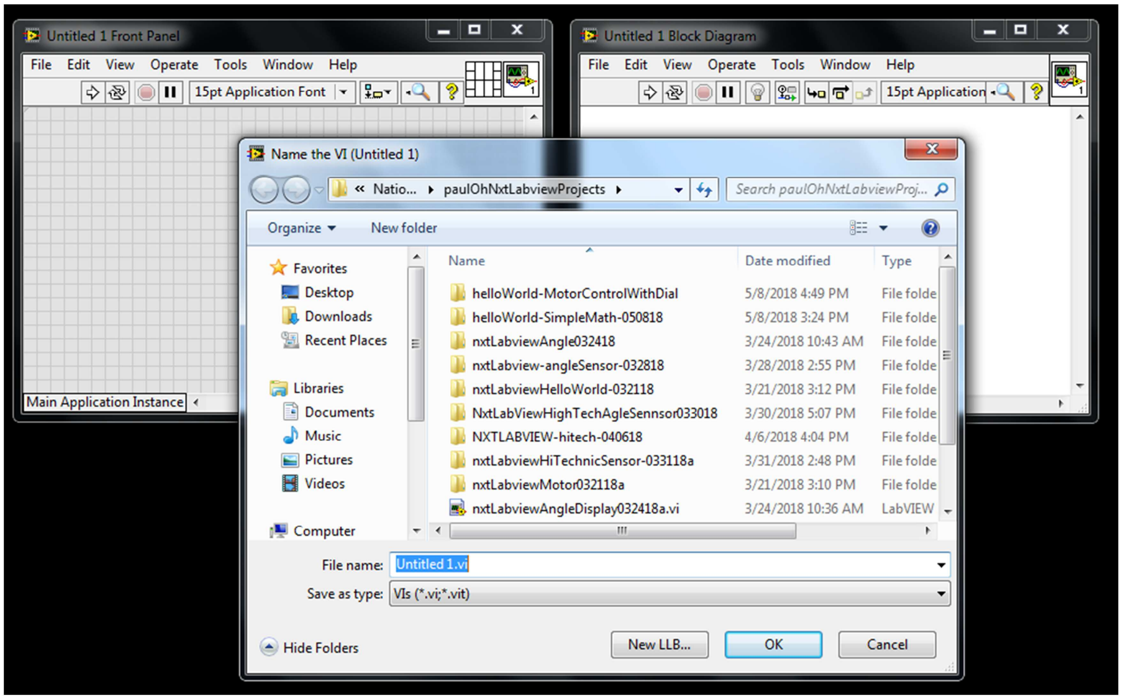


Figure B1: Front panel and Block Diagram



Name	Date modified	Type
helloWorld-MotorControlWithDial	5/8/2018 4:49 PM	File folde
helloWorld-SimpleMath-050818	5/8/2018 3:24 PM	File folde
nxtLabviewAngle032418	3/24/2018 10:43 AM	File folde
nxtLabview-angleSensor-032818	3/28/2018 2:55 PM	File folde
nxtLabviewHelloWorld-032118	3/21/2018 3:12 PM	File folde
NxtLabViewHighTechAgleSennsor033018	3/30/2018 5:07 PM	File folde
NXTLABVIEW-hitech-040618	4/6/2018 4:04 PM	File folde
nxtLabviewHiTechnicSensor-033118a	3/31/2018 2:48 PM	File folde
nxtLabviewMotor032118a	3/21/2018 3:10 PM	File folde
nxtLabviewAngleDisplay032418a.vi	3/24/2018 10:36 AM	LabVIEW

File name: Untitled 1.vi

Save as type: VIs (*.vi;*.vit)

Figure B2: Chose File – Save As to store your VI programs e.g. to a USB thumb drive

You'll be repeating these Preamble steps for all your future VI programs.

Concept 1: Creating your first VI: Front Panels and Block Diagrams

LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench. In LabVIEW, a virtual instrument has two parts, namely a *front panel* (the visual front-end) and a *block diagram* (underlying high-level graphics-based code). One creates the front panel and block diagram by clicking and dragging icons (called controls). The objective in this exercise is to create a simple program that adds displays the sum and product of two numbers. The LED glows when the two numbers are equal.

Step 1: Launch LabVIEW to create the following front panel (**Figure 1A** below). Save your VI as `nxtLabview-HelloWorld-SimpleMath.vi`.

On your front panel, click `View – Controls Palette` as seen in **Figure 1A** below. This pops up a Controls palette. This palette categorizes GUI components like readout displays (numeric and alphanumeric), knobs and graph charts. Clicking `Numeric` reveals specific components (**Figure 1B**).

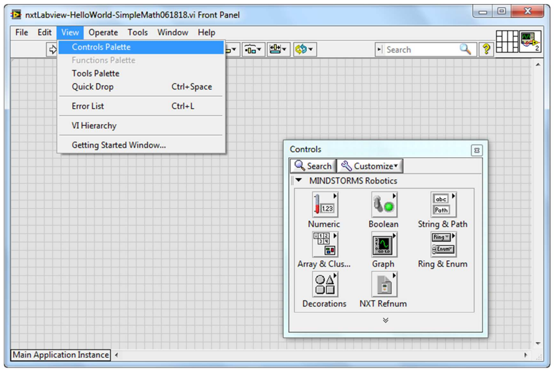


Figure 1A: On your Front Panel, click View-Controls Palette

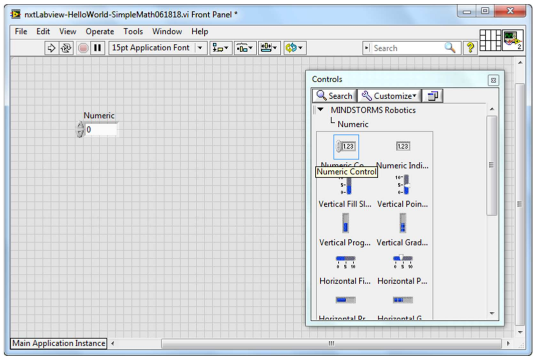


Figure 1B: In the Control Palette, clicking Numeric reveals specific components'

Step 2: Create your Front Panel

Figure 1B shows a `numeric controller` that has already been dragged to the Front Panel. Numeric controllers feature up/down rocker switches. These switches allow one to control the value. This numeric controller was renamed "A". Create a second one. From the Control Palette, click-and-drag a numeric controller into the grey area of your front panel as shown in **Figure 1C**. Double-click the "Numeric" and rename it as "B"

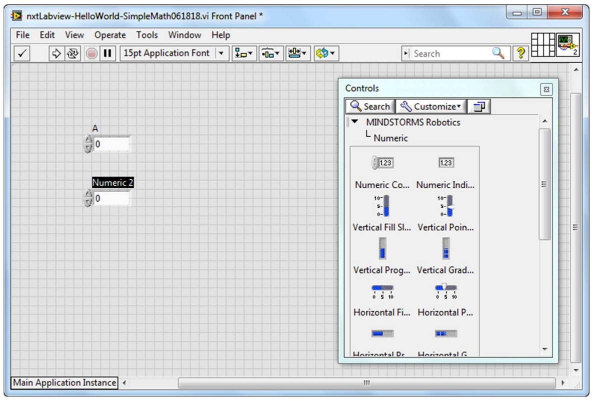


Figure 1C: Two numeric controllers (labeled as A and B) were created by clicking-and-dragging "`numeric controller`" from the Controls palette

Complete your front panel by adding numeric indicators; click-and-drag a `numeric indicator` from the Controls Palette as seen in **Figure 1D**.

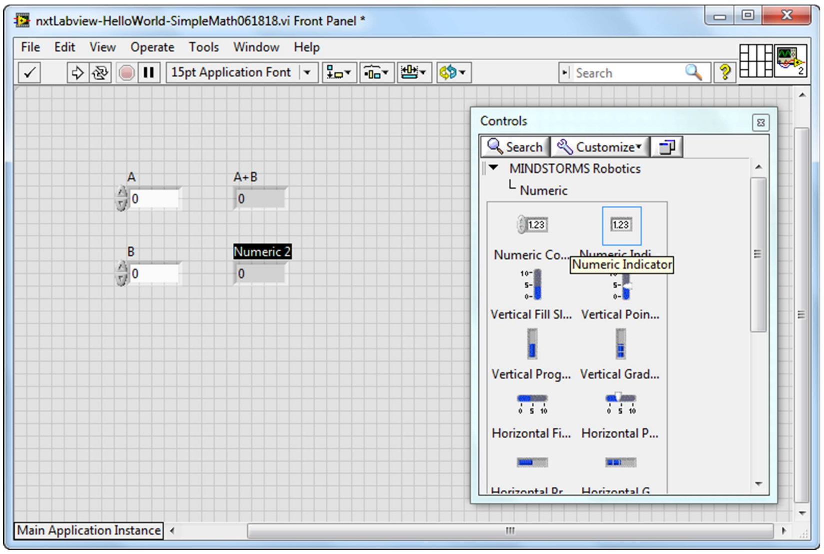
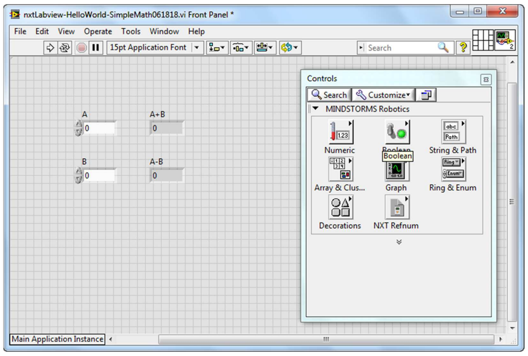


Figure 1D: Two numeric indicators, labeled “A+B” and “AxB” were dragged from the Control Palette.

Figure 1D also the 2 numeric indicators. Double-click their labels and rename as “`A+B`” (for the top one) and “`AxB`” (for the bottom one).

Next, add a round LED. From the Controls Palette, select Boolean (**Figure 1E**). Click-and-drag a “`Round LED`” into the front panel (**Figure 1F**).


nxtLabview-HelloWorld-SimpleMath061818.vi Front Panel *
File Edit View Operate Tools Window Help
Controls
MINDSTORMS Robotics
Numeric
Boolean
String & Path
Array & Clus...
Graph
Ring & Enum
Decorations
NXT Refnum
A
A+B
B
A-B
Main Application Instance

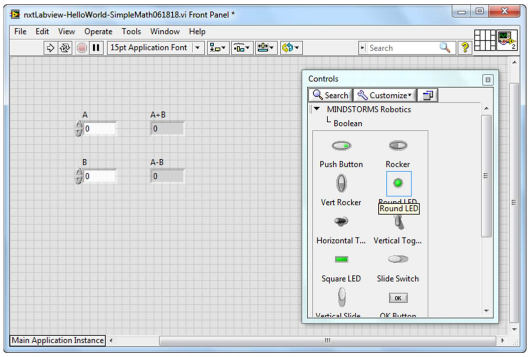


Figure 1E: Select `Boolean` from Controls Palette

Figure 1F: Select `Round LED` from the Boolean category

Re-label the round LED as “`Equal`” and verify that your completed front panel looks like **Figure 1G**. Click File-Save All to ensure your program is saved.

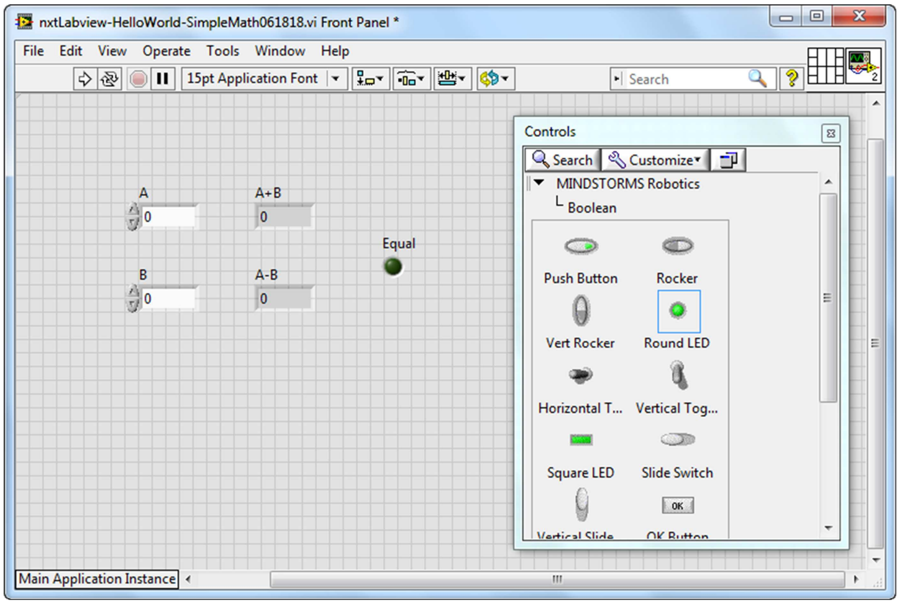
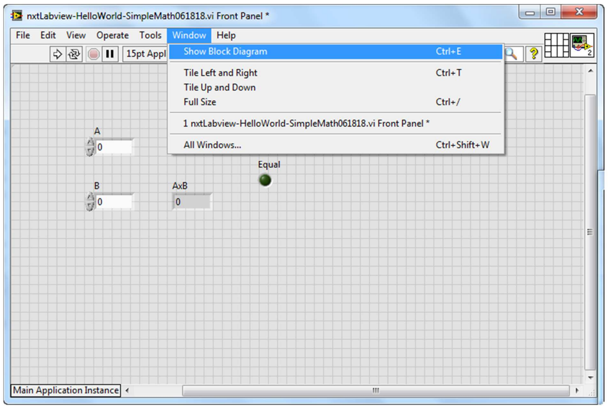


Figure 1G: Completed front panel with 2 numeric controls, 2 numeric indicators, and 1 round LED

Step 3: Create the Block Diagram

With your front panel completed, it's time to program the functions of all the controls. From your front panel, click `Window – Show Block Diagram` (**Figure 1H**). This will pop up the Block Diagram. Click-and-drag components so that the block diagram looks like **Figure 1I**.


nxtLabview-HelloWorld-SimpleMath061818.vi Front Panel *
File Edit View Operate Tools Window Help
15pt Appl
Show Block Diagram Ctrl+E
Tile Left and Right Ctrl+T
Tile Up and Down
Full Size Ctrl+/
1 nxtLabview-HelloWorld-SimpleMath061818.vi Front Panel *
All Windows... Ctrl+Shift+W
A
0
Equal
B
0
AxB
0
Main Application Instance

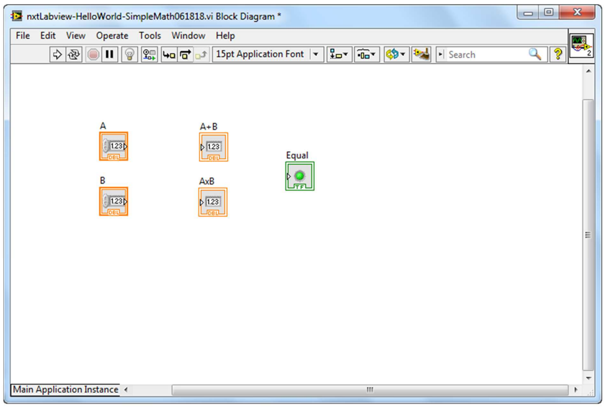
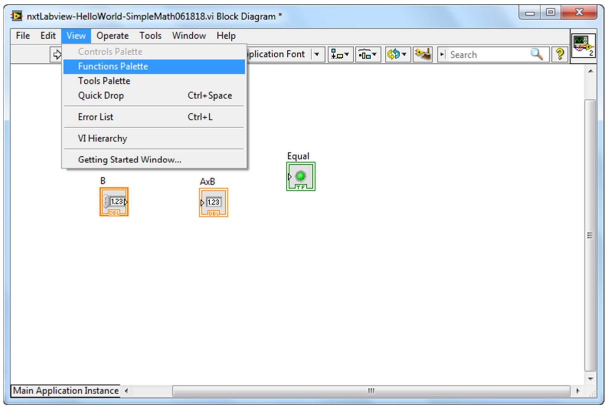

nxtLabview-HelloWorld-SimpleMath061818.vi Block Diagram *
File Edit View Operate Tools Window Help
15pt Application Font
Search
A
123
A+B
123
Equal
B
123
AxB
123
Main Application Instance

Figure 1H: Click Window-Show Block Diagram

Figure 1I: Rearrange components. This will make wiring easier in the next steps to see

Now one must program the relationship between components used in your front panel. Click `View-Functions Palette` (**Figure 1J**) will reveal sub-categories like Programming, I/O and Behaviors (**Figure 1K**).


nxtLabview-HelloWorld-SimpleMath061818.vi Block Diagram *
File Edit View Operate Tools Window Help
Controls Palette
Functions Palette
Tools Palette
Quick Drop Ctrl+Space
Error List Ctrl+L
VI Hierarchy
Getting Started Window...
Equal
B
AxB
Main Application Instance

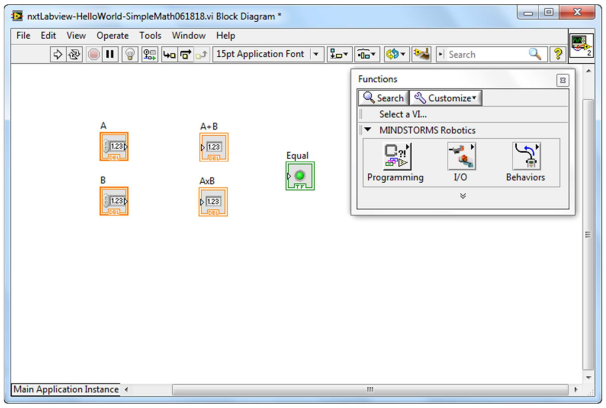
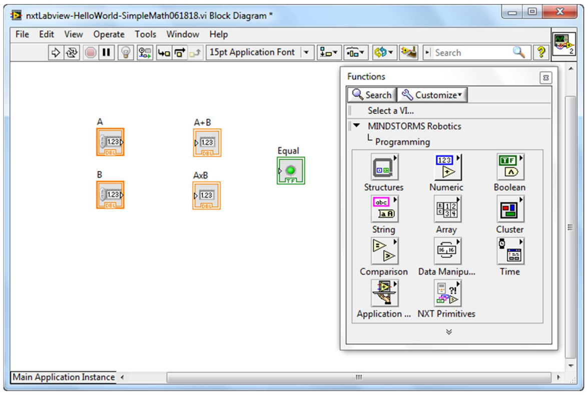

nxtLabview-HelloWorld-SimpleMath061818.vi Block Diagram *
File Edit View Operate Tools Window Help
15pt Application Font
Search
A
A+B
B
AxB
Equal
Functions
Search
Customize
Select a VI...
MINDSTORMS Robotics
Programming
I/O
Behaviors
Main Application Instance

Figure 1J: Click View-Functions Palette

Figure 1K: Functions palette shows 3 options.

Click `Programming` reveals sub-categories like Structures, Numeric, and Boolean (**Figure 1L**). Click `Numeric` reveals functions like Add, Subtract and Multiply (**Figure 1M**).



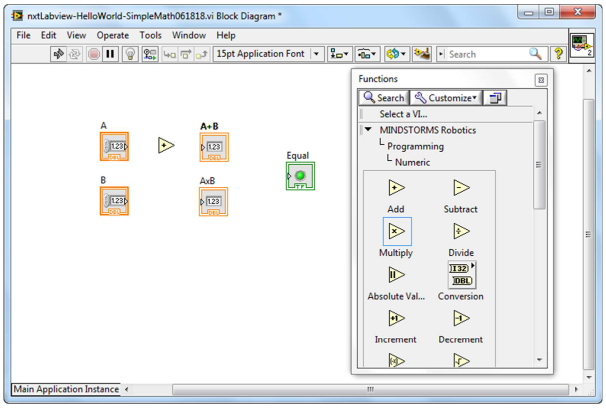


Figure 1L: Programming categories reveal components like Structure and Numeric

Figure 1M: Numeric categories show functions like Add and Subtract

Figure 1M shows that an "`Add`" function already dragged between the numeric control (A) and numeric indicator (A+B). Select the "Multiply" function so that your block diagram looks like **Figure 1N**. During run-time, when the two values in the numeric controls are equal, the round LED will glow. Thus, under "Comparison" select the "Equal" function and drag into your block diagram (**Figure 1N**).

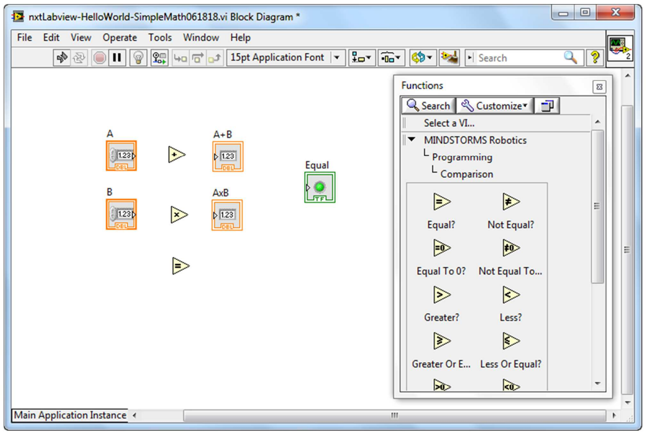


Figure 1N: Under the Programming category, select Comparison. Find the "Equal" function and click-and-drag to the block diagram.

Step 4: Wiring up the block diagram

To final step to complete the block diagram, is to wire connections between the functions and controls. Click `View – Tools Palette` and select the `wiring tool` (spool icon) as seen in **Figure 1O**.

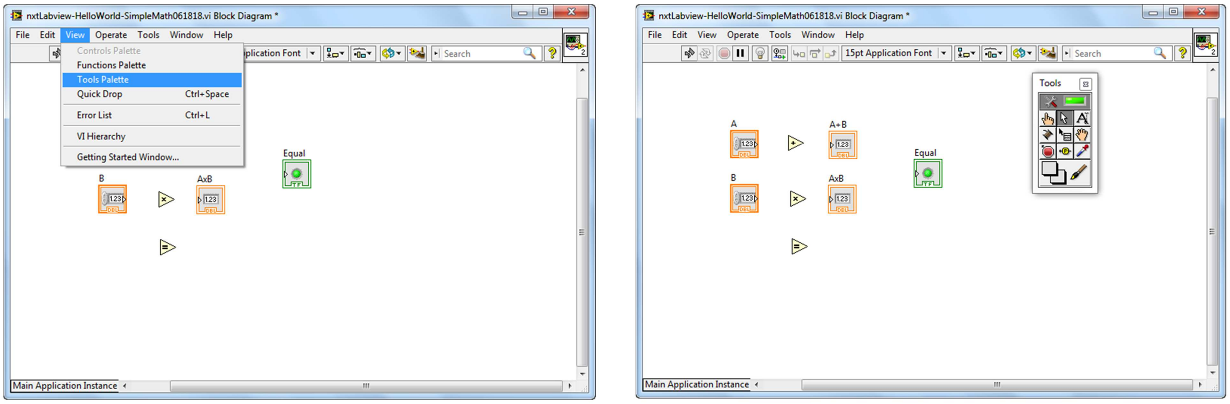

nxtLabview-HelloWorld-SimpleMath061818.vi Block Diagram *
File Edit View Operate Tools Window Help
Controls Palette
Functions Palette
Tools Palette
Quick Drop Ctrl+Space
Error List Ctrl+L
VI Hierarchy
Getting Started Window...
Search
A
B
A+B
AxB
Equal
Tools
Main Application Instance

Figure 1O: Selecting `View – Tools Palette` (left) brings up Tools (right). One can select the `wiring tool` (which is the spool icon).

With the wiring tool, one mouse-clicks nodes on the controls. **Figure 1P**, shows wiring of the two numeric controls (A and B), into the left side of the Add function. The output of the Add function is then wired to the input of the A+B numeric indicator.

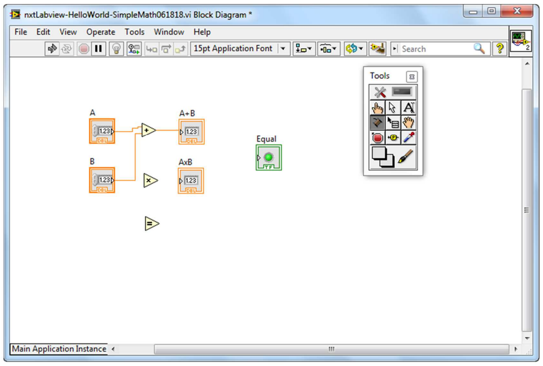

nxtLabview-HelloWorld-SimpleMath061818.vi Block Diagram *
File Edit View Operate Tools Window Help
15pt Application Font
Search
A
B
A+B
AxB
Equal
Tools
Main Application Instance

Figure 1P: Wiring of the Add function

In a similar fashion, wire up the Multiply and Equal functions as shown in **Figure 1Q**.

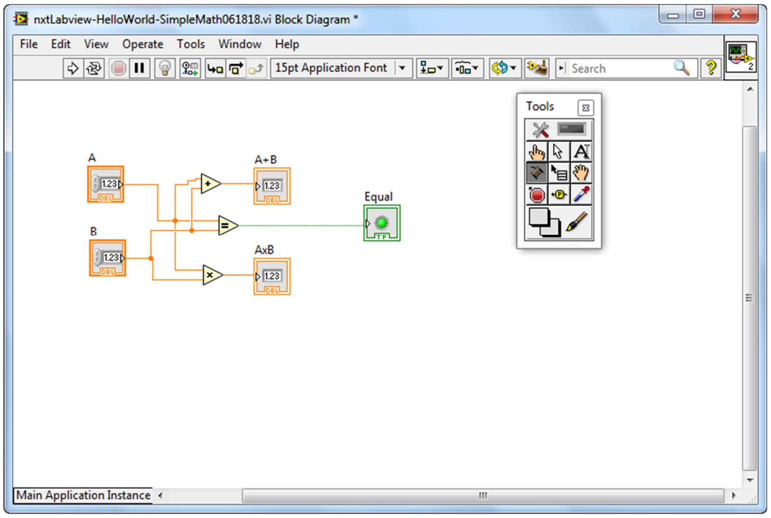

nxtLabview-HelloWorld-SimpleMath061818.vi Block Diagram *
File Edit View Operate Tools Window Help
15pt Application Font
Search
Tools
A
B
A+B
AxB
Equal
123
DBL
Main Application Instance

Figure 1Q: One can click a function or controls and move it in the block diagram. This helps make "wiring" easier and cleaner to view and use.

Step 5: Execute Program

Return back to your front panel. Click the `Run Continuously` icon (**Figure 1R**). One can now increment or decrement values in the numeric controls. The numeric indicators will show their sum and product. When the two values of the numeric indicators are equal, the LED will glow green.

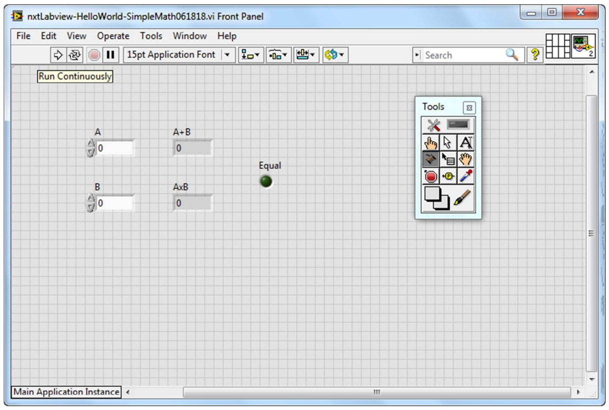

nxtLabview-HelloWorld-SimpleMath061818.vi Front Panel
File Edit View Operate Tools Window Help
15pt Application Font
Search
Run Continuously
A
0
B
0
A+B
0
AxB
0
Equal
Tools
Main Application Instance

Figure 1R: The `Run Continuously` button (double arrows in a circle) is clicked to execute the program.

Congratulations! You've written your first LEGO Labview Program!

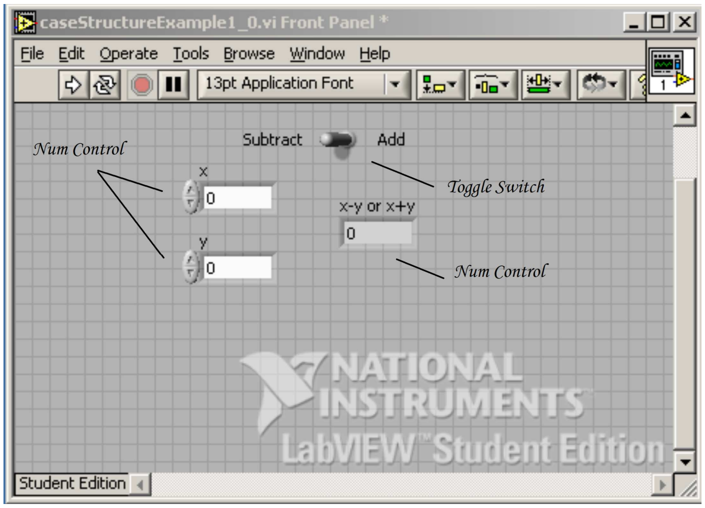
Exercise 1: In LabVIEW create programs for the following:

1-1. Accepts temperatures in Celsius and displays its Fahrenheit equivalent (try pointer slides)
1-2. Accepts 3 numbers and displays the average
1-3. Accepts 2 numbers and displays which is the maximum

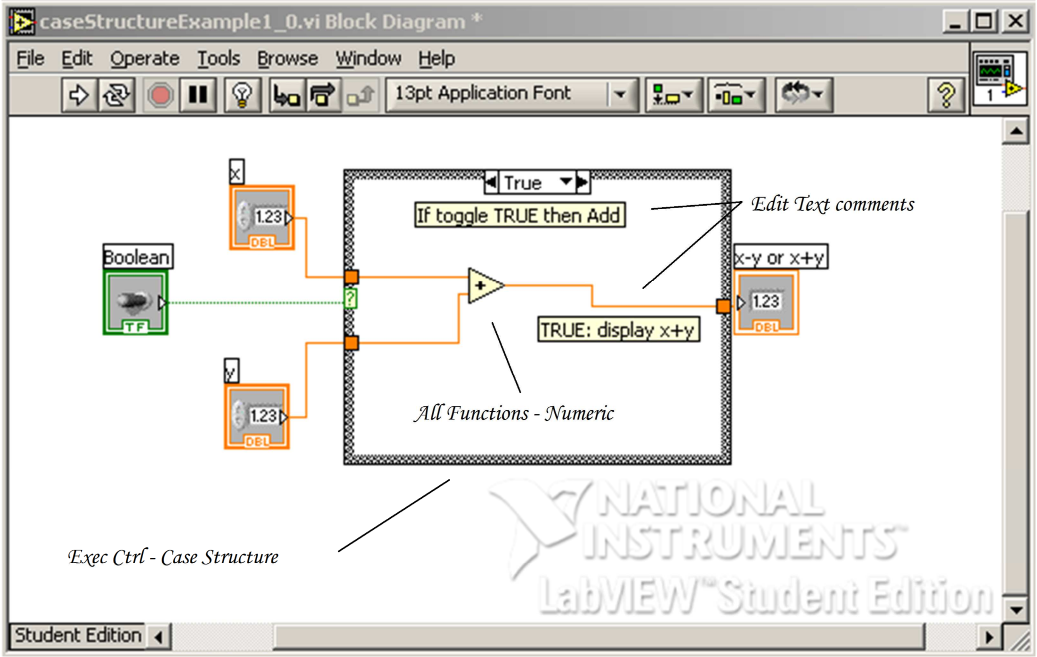
Concept 2: Case Structures

Branch statements are those that test a condition (true or false) and execute commands based on those conditions. LabVIEW has several controls for branching. In the block diagram programming, the `Case Structure` control is invoked under the `Executive Control` category.

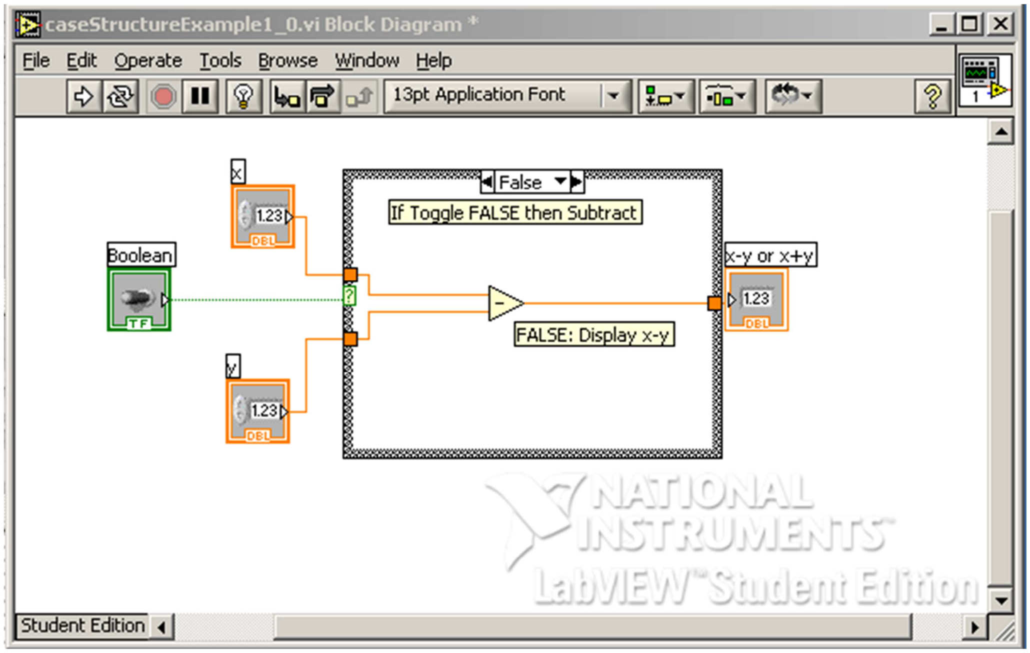
Step 1: Create the following LabVIEW front end. Save your VI as `nxtLabviewCaseStatement.vi` – Note that toggle switch by default is `FALSE` by with a "`Switch When Pressed`" mechanical action.



Step 2: Create the Block Diagram (`True Case`: adds the two numbers). Note: clicking the down arrow in the `Condition Terminal` sets the `True` condition.



Step 3: Create the `False Case` condition (False case: subtracts the two numbers)



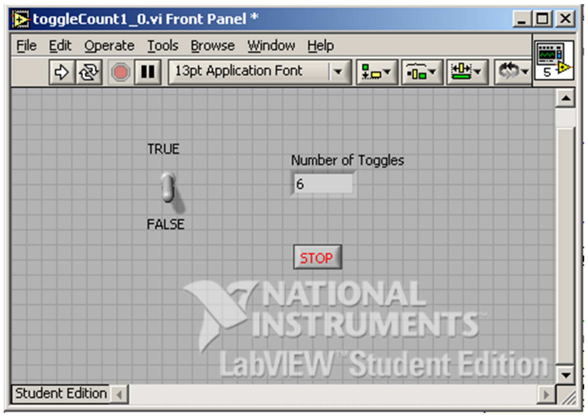
Exercise 2: In LabVIEW create programs for the following:

2-1. Given an input in inches, the equivalent in centimeters is displayed (by default) and the equivalent in feet is displayed (if the switch is toggled)
2-2. The maximum or minimum of two numbers is displayed based on toggle position

Concept 3: While Loops

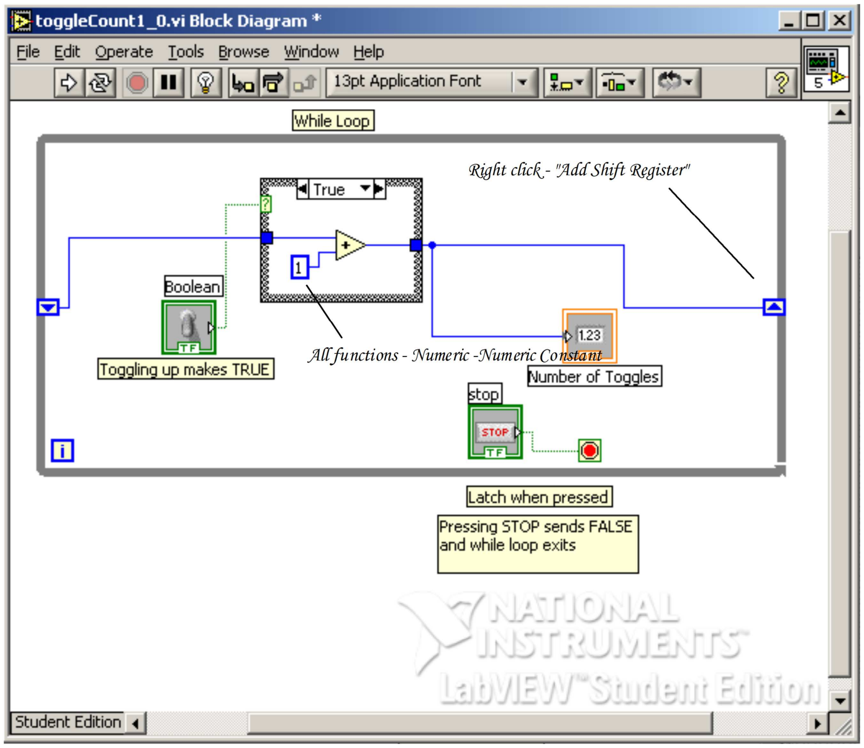
Loops that repeat until a condition is met are found in every programming language and LabVIEW is no exception. The `While-loop` repeats commands in its block control while the condition is `TRUE`. The `While-loop` control also has an iteration terminal that increases each time the block is executed.

Step 1: This LabVIEW VI counts the number of times the switch is toggled. Create the following front panel. Save it as `nxtLabviewWhileStatement.vi`.



Note: the toggle switch should have its mechanical action set to "`Latch when released`".

Step 2: Create the following block diagram. A `case structure` control is inside a `while-loop` control. When the case is `TRUE`, a variable (shift register) increases, thereby displaying the number of toggles. The while-loop control stops when the user hits the `Stop` button.

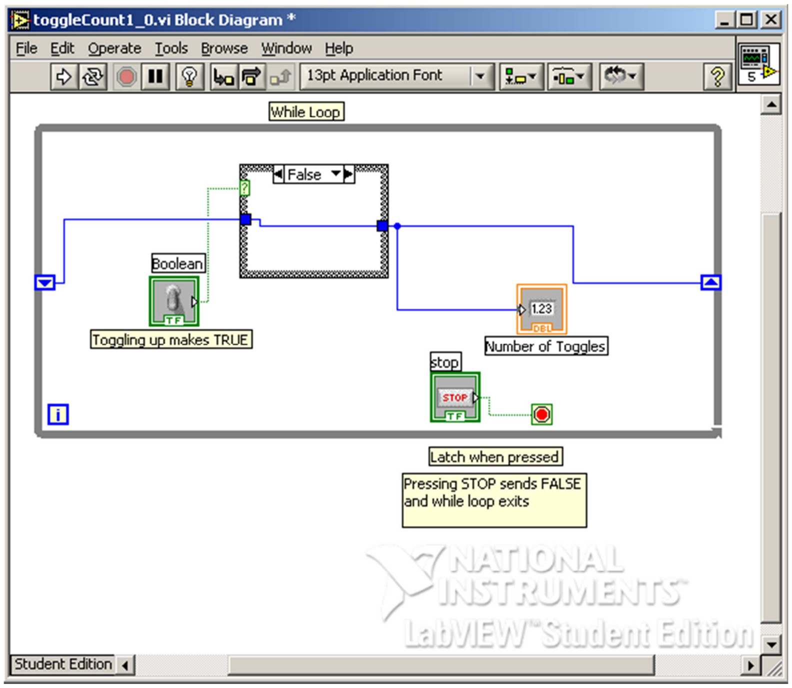


Right click - "Add Shift Register"

All functions - Numeric -Numeric Constant

Pressing STOP sends FALSE and while loop exits

Step 3: Add the `False` condition in the conditional terminal. Note that there are no contents (i.e. do nothing) in the case structure



Pressing STOP sends FALSE and while loop exits

Step 4: Run your program and toggle the switch. Also, try clicking on the `Highlight Execution` (light bulb icon) button from the Block Diagram menu bar and notice what happens when you execute your program.

Exercise 3: In LabVIEW create programs for the following:

3-1. The display increases by 2 each time the switch is toggled
3-2. Increments or decrements depending on the users choice (i.e. using a 2nd toggle switch)
3-3. Create a for-loop example.