Lab 2

Smart Sensors and Actuators with myRIO

DCTI-ISCTE-IUL

2023-2024

Octavian Adrian Postolache

Lab 2

Smart Sensors and Actuators with myRIO

Objectives

- Develop smart sensing systems using reconfigurable platforms

- Implement LV programs for the myRIO-1900 FPGA module considering features related to analog or digital I / O channels.

- implement programs in LV for the Real Time module myRIO including the development of GUI and data communication.

- Perform an smart system using force sensors and myRIO module including the analog channels and the LEDs and the on-board button.

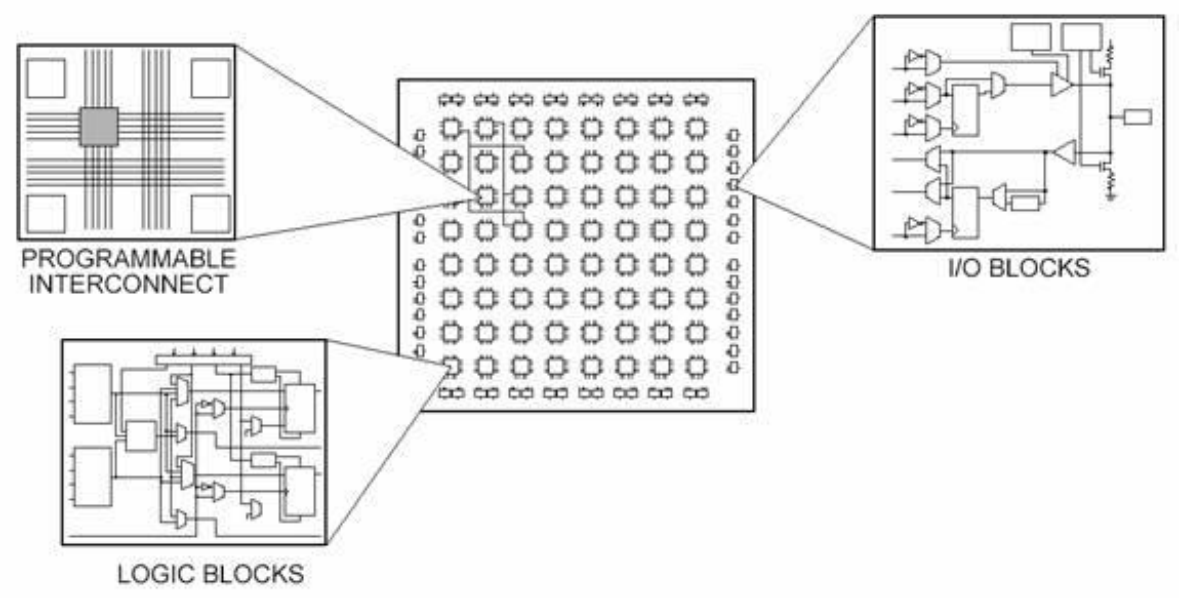
Theory

- **FPGA – definition**

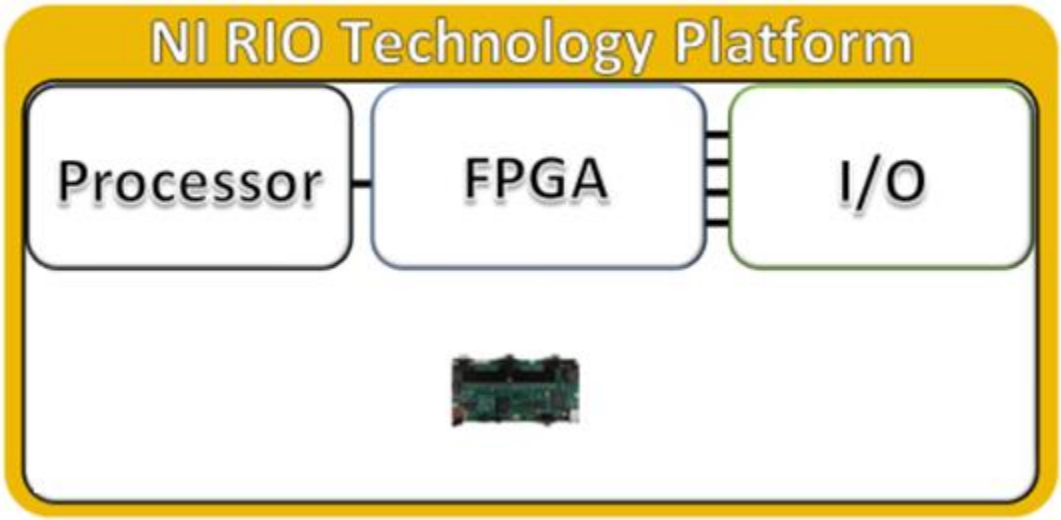
An FPGA is a device that contains a matrix of reconfigurable gate array logic circuitry.

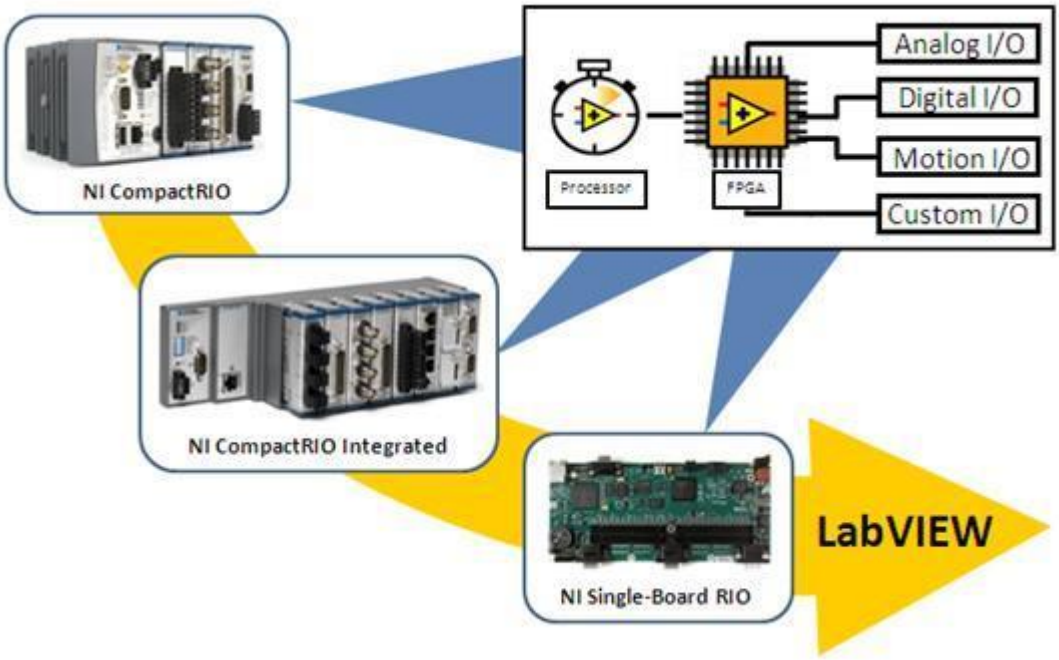
When a FPGA is configured, the internal circuitry is connected in a way that creates a hardware implementation of the software application. Unlike processors, FPGAs use dedicated hardware for processing logic and do not have an operating system. FPGAs are truly parallel in nature so different processing operations do not have to compete for the same resources. As a result, the performance of one part of the application is not

affected when additional processing is added.



FPGA architectures: Single Board RIO





Programação:

Tutorials

1. **LabVIEW MyRIO**
   http://www.ni.com/labview/realtime/pt/
2. **LabVIEW FPGA**
   http://www.ni.com/labview/fpga/pt/

Hardware:

FPGA MyRIO

Characteristics

http://www.ni.com/pdf/manuals/376047a.pdf

Perform an analysis of the characteristics of the myRIO module and present the main features in the report of Lab2

Exercises:

Ex1 - On-board sensors and actuators myRIO

1.1. Run a LabVIEW application for the myRIO platform from reading the x, y, z acceleration values.

1.2. Perform a LabVIEW application whose purpose is to generate an alarm signal using the on-board LED0, LED1, LED2 of the myRIO platform when the acceleration values exceed predefined values for each of the 3 axes, x (LED0-ON) , and (LED1-ON), z (LED2-ON)

1.3. Perform a LabVIEW application for the myRIO platform associated x, y, z acceleration values reading followed by the calculation of the resulting acceleration when the Button 0 is pressed.

Help:

Please follow the tutorial: Creating Your First myRIO Application (Annex 1)

Ex 2 - FPGA myRIO I / O

Perform an application in FPGA that relates the loading of Button 0 (on-board) with the actions on the LEDS of myRIO. Please consider the following conditions:

1. LED0 will be switched on when the boolean control LED0 is charged simultaneously with the Button 0

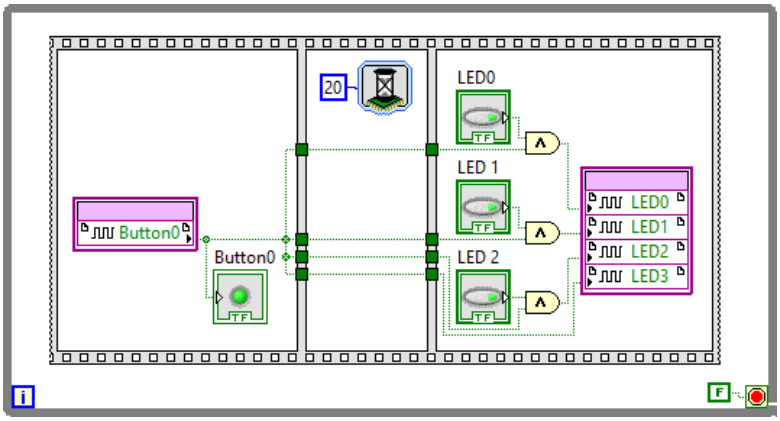
2. LED1 will be turned on when the boolean control LED1 is charged simultaneously with the Button 0

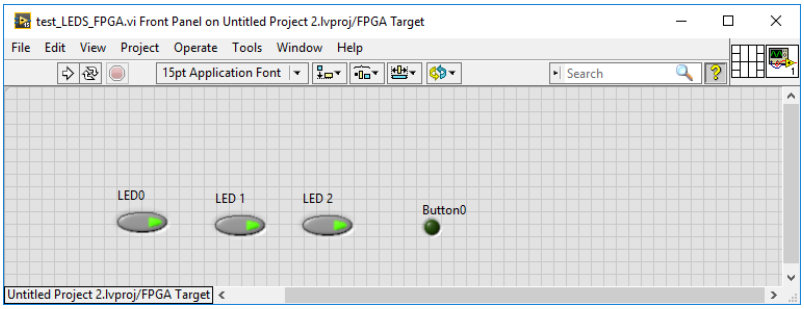
3. LED2 will be turned on when the boolean control LED2 is charged simultaneously with the Button 0

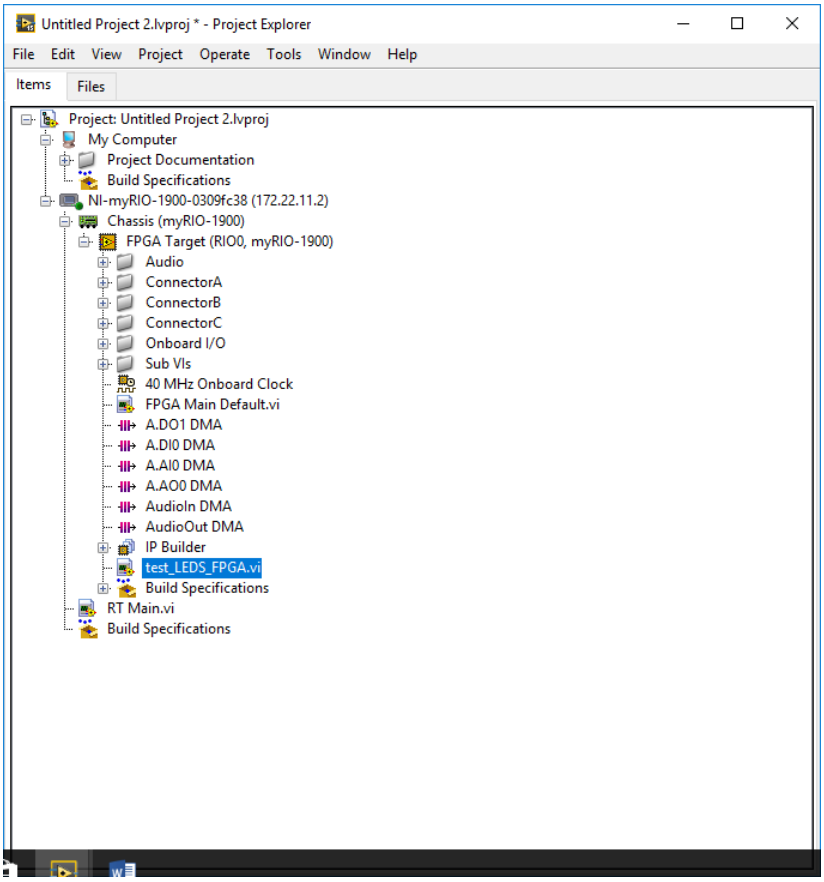
4. LED2 will be turned on when Button 0 is charged

Perform the compilation and indicate in the report the elements that characterize the compilation for myRIO FPGA platform.

Help:


20
LED0
LED 1
LED 2
Button0
Button0
LED0
LED1
LED2
LED3


test_LEDS_FPGA.vi Front Panel on Untitled Project 2.lvproj/FPGA Target
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
LED0
LED 1
LED 2
Button0
Untitled Project 2.lvproj/FPGA Target


Untitled Project 2.lvproj * - Project Explorer
File Edit View Project Operate Tools Window Help
Items
Files
Project: Untitled Project 2.lvproj
My Computer
Project Documentation
Build Specifications
NI-myRIO-1900-0309fc38 (172.22.11.2)
Chassis (myRIO-1900)
FPGA Target (RIO0, myRIO-1900)
Audio
ConnectorA
ConnectorB
ConnectorC
Onboard I/O
Sub VIs
40 MHz Onboard Clock
FPGA Main Default.vi
A.DO1 DMA
A.DI0 DMA
A.AI0 DMA
A.AO0 DMA
AudioIn DMA
AudioOut DMA
IP Builder
test_LEDS_FPGA.vi
Build Specifications
RT Main.vi
Build Specifications

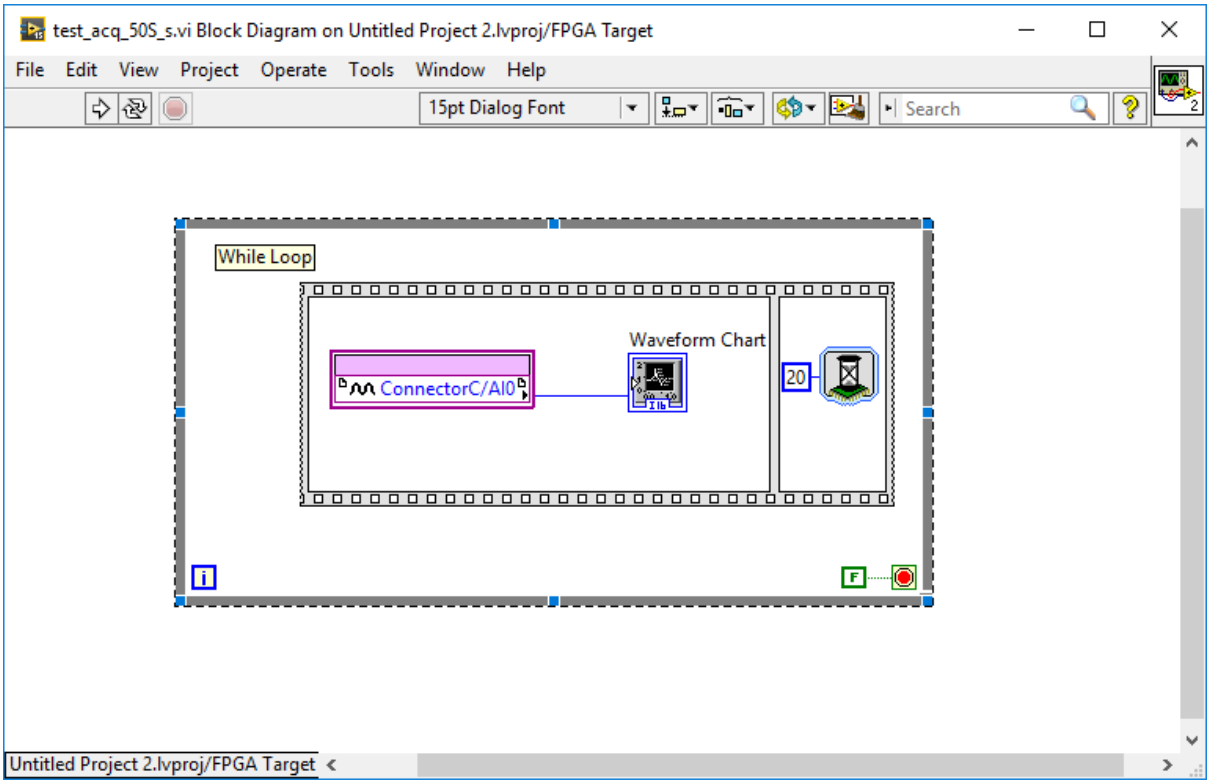
Ex. 3

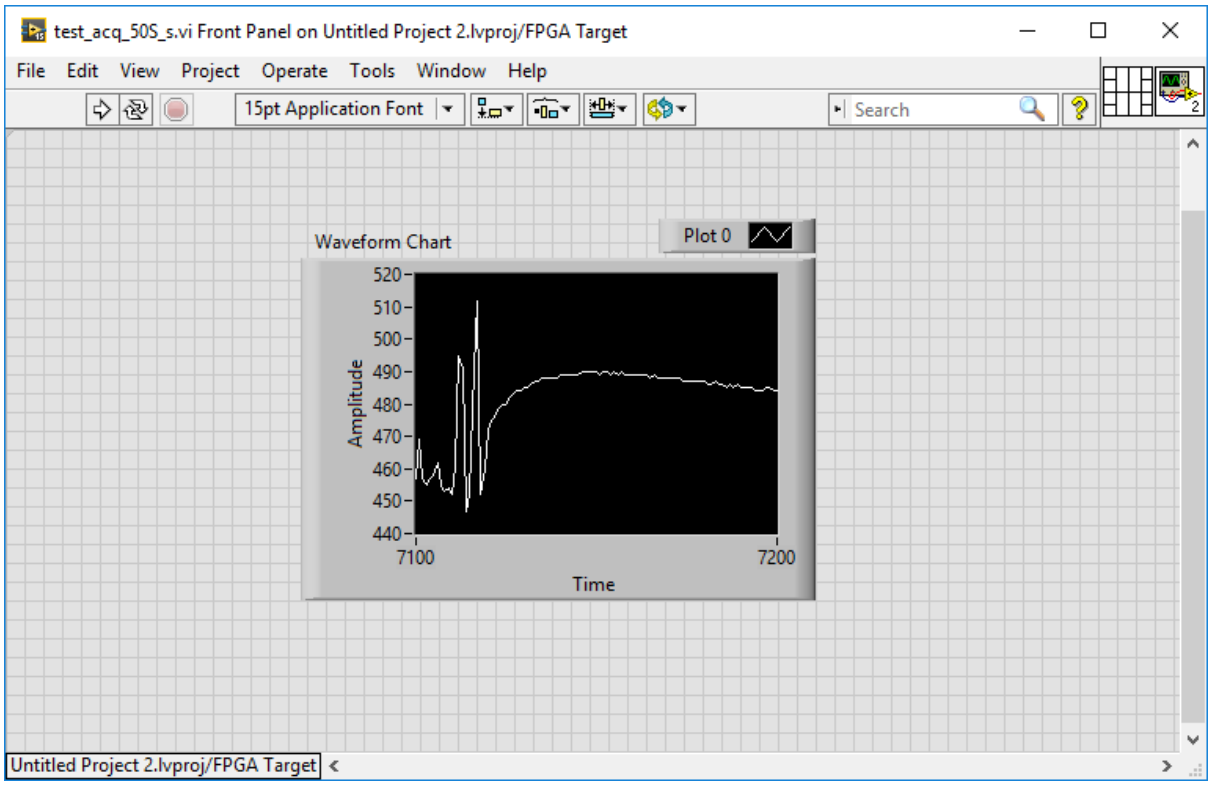
3.1. Perform the acquisition of an analog signal using AI0 from the myRIO platform and an assembly with a potentiometer whose ends are connected to the DGND and 5V and the cursor at the AI0 analog input

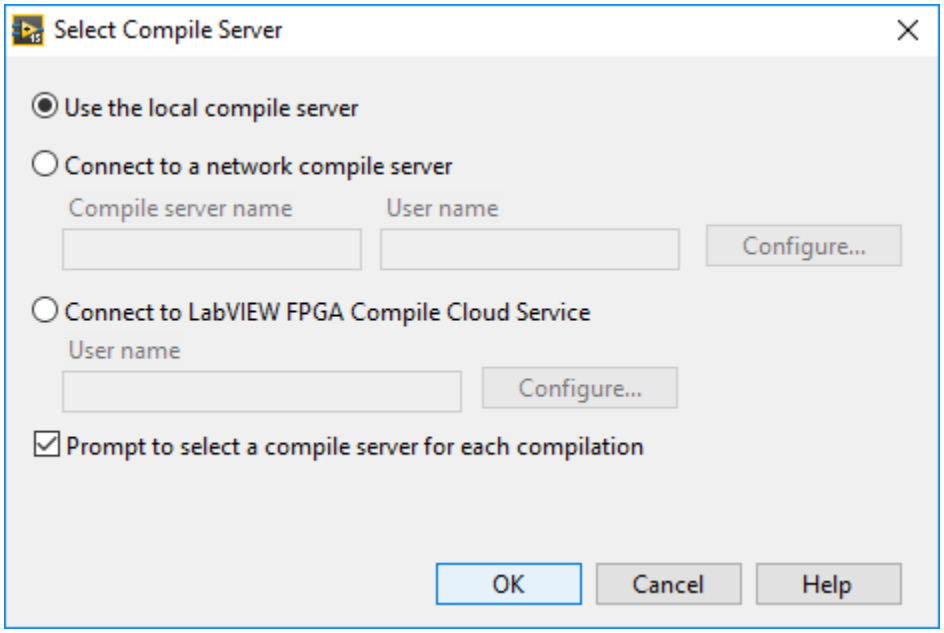
3.2 Use the LED0 ... LED3 to perform a user interface associated with the voltage r values. Thus the 0-5V interval will be divided into 4 ranges of values and exceedance of each of the limit values for each of the ranges will be signaled by using one of the LEDs

3.3. Replace the potentiometer with a force resistive type sensor (piezoresistive) as part of a resistive divider and record the force values on a chart. Use the LEDS to indicate the applied force level.

Ajuda




test_acq_50S_s.vi Front Panel on Untitled Project 2.lvproj/FPGA Target
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Waveform Chart
Plot 0
520
510
500
490
480
470
460
450
440
Amplitude
7100
7200
Time
Untitled Project 2.lvproj/FPGA Target


Select Compile Server
Use the local compile server
Connect to a network compile server
Compile server name
User name
Configure...
Connect to LabVIEW FPGA Compile Cloud Service
User name
Configure...
Prompt to select a compile server for each compilation
OK
Cancel
Help

Annex 1

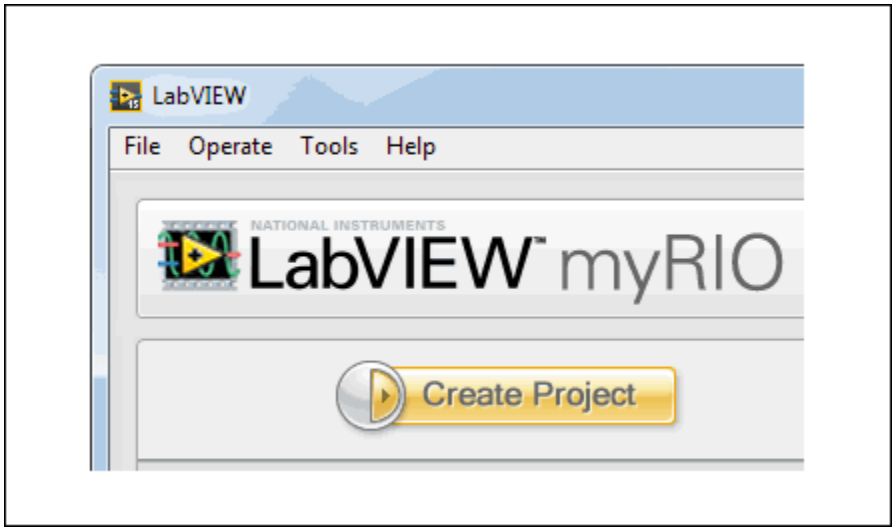
Creating Your First myRIO Application

This tutorial teaches you how to create your own application to control your myRIO onboard devices.

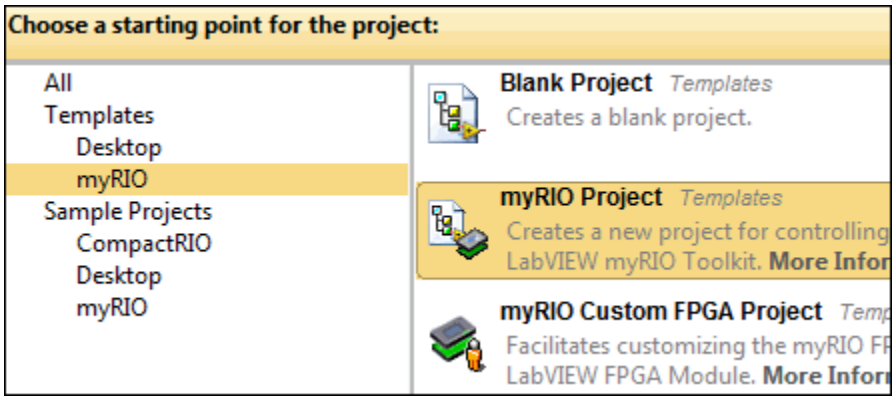
Part 1: Creating a myRIO Project

Before programming with the myRIO, you must first create a myRIO project. With a myRIO project, you can group together all the files relevant to your application and run VIs on the myRIO.

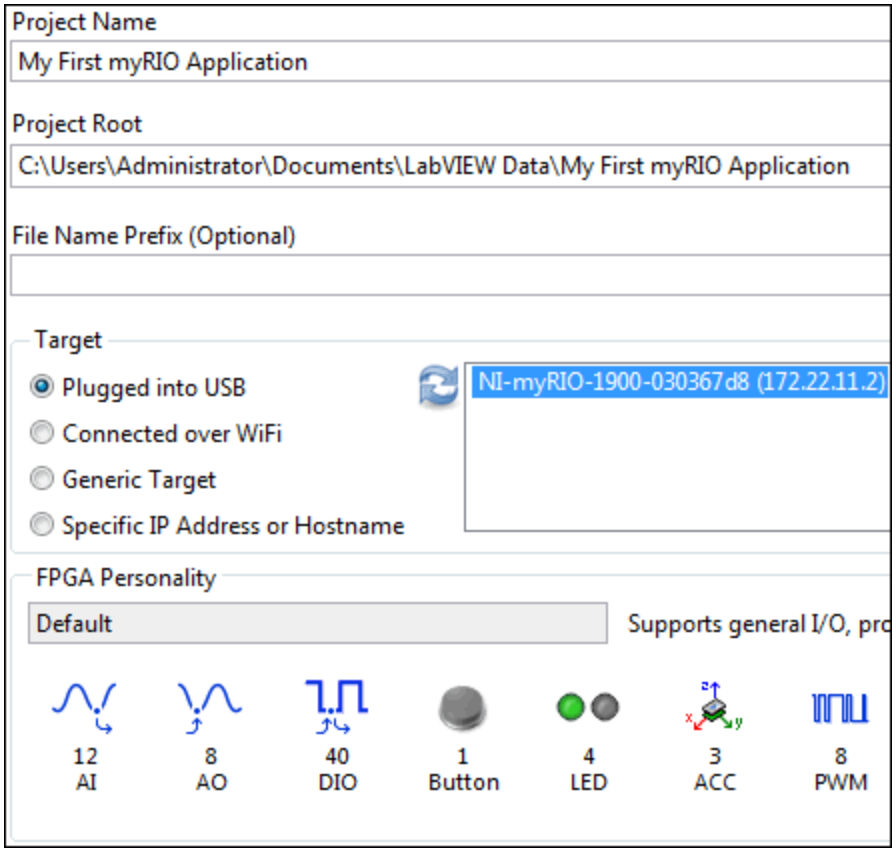
Complete the following steps to create a myRIO project by using the **myRIO Project** template, which is the simplest way to start programming with the myRIO.



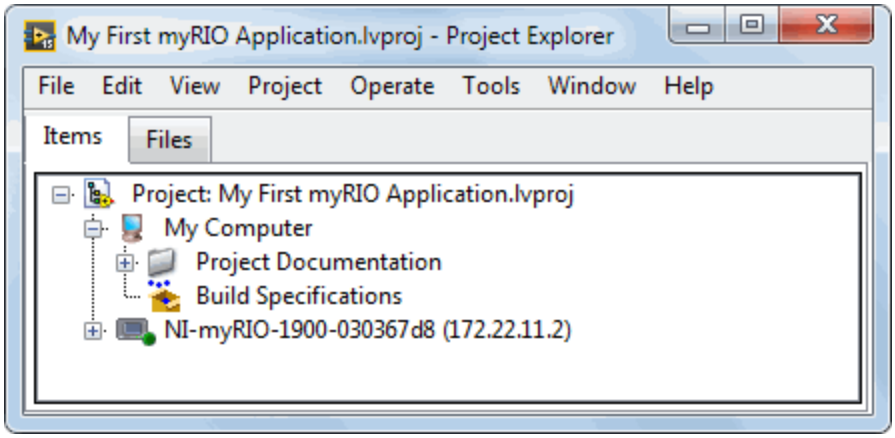
1. Click the **Create Project** button in LabVIEW to display the **Create Project** dialog box.



2. Select **Templates»myRIO** from the project category.
3. Select **myRIO Project** from the project list.
4. Click **Next** to configure details of the project.



5. In the **Project Name** text box, enter My First myRIO Application.
6. In the **Project Root** text box, enter the path to the directory for saving the project.
7. (Optional) In the **File Name Prefix** text box, enter a prefix that distinguishes the copy of the template you create from additional copies of the template you create later.
8. In the **Target** section, select the myRIO on which to run your application.
9. The **FPGA Personality** section indicates that the **myRIO Project** template uses a default FPGA personality. Refer to the *Choosing FPGA Personalities* topic in the *LabVIEW Help* for information about FPGA personalities.
10. Click **Finish**. LabVIEW saves the project and opens the **Project Explorer** window.

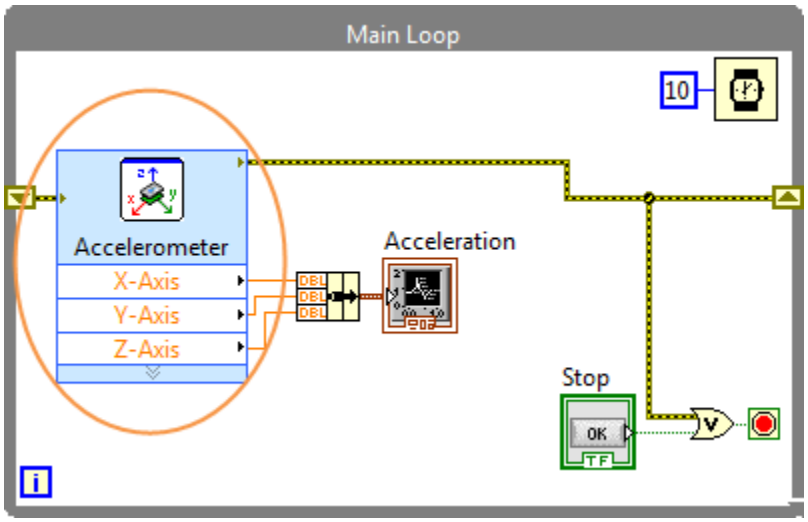


11. Explore the **Project Explorer** window. For example, expand items in the project tree to find `Main.vi`. Refer to the `Project Documentation` folder for detailed information about this myRIO project.

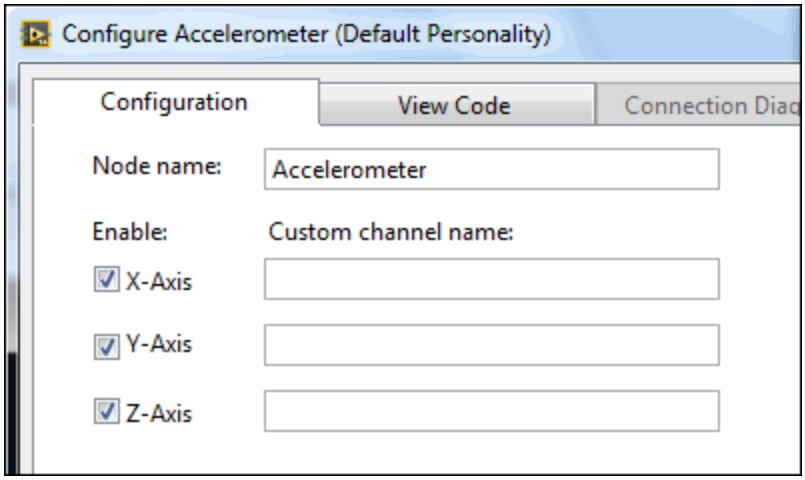
You have successfully created a myRIO project. Now, proceed to learn how to create applications based on the myRIO project.

Part 2: Controlling the Accelerometer

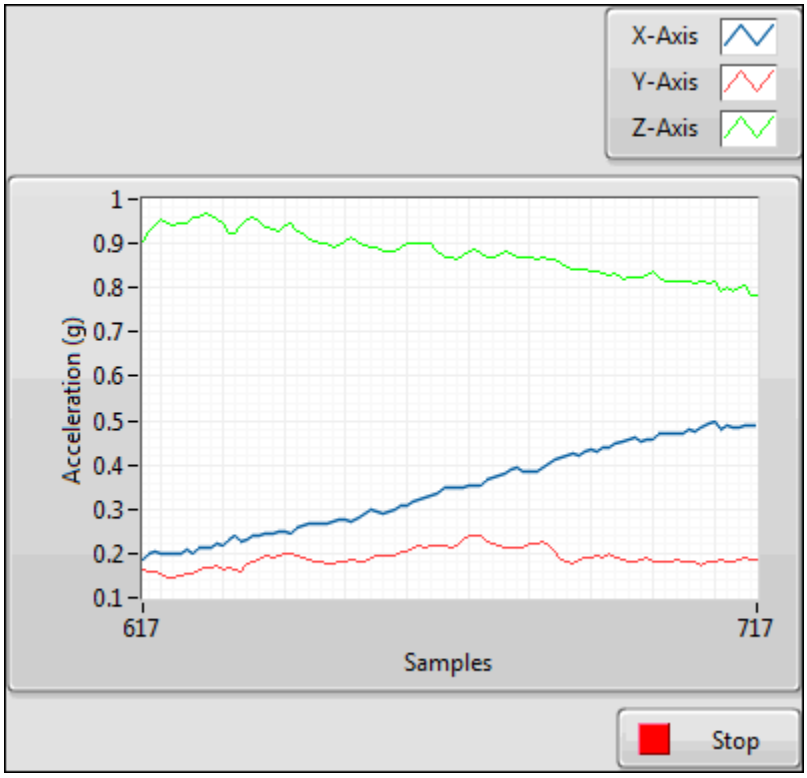
The myRIO contains an onboard accelerometer that measures the magnitude and direction of acceleration. This part of the tutorial teaches you how to create an application to read acceleration values from the onboard accelerometer and to display the acceleration values on a waveform chart. Before starting this part, make sure you have completed the previous part of the tutorial.

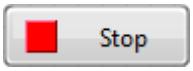


1. In the **Project Explorer** window of your myRIO project, open the Main VI. By default, LabVIEW opens the front panel of the Main VI. The front panel is the user interface of a VI.
2. Press <Ctrl-E> to switch to the block diagram of the Main VI. The block diagram contains the graphical code of a VI. This VI uses the Accelerometer Express VI to read acceleration values from the onboard accelerometer and uses the waveform chart indicator to display the acceleration values.
3. Double-click the Accelerometer Express VI to display the configuration dialog box.



4. Press <Ctrl-H> to display the **Context Help** window. You can move the cursor over options in the configuration dialog box and learn basic information about the options from the **Context Help** window. Most objects in LabVIEW display context help information.
5. Click **OK** to close the configuration dialog box.
6. Press <Ctrl-E> to switch to the front panel of the Main VI.



7. Click **Run**  .
8. Rotate or shake the myRIO and observe the changes of the X, Y, and Z acceleration values on the waveform chart.
9. Click **Stop** 

.
10. Select **File»Save** to save the VI.

Congratulations! You have successfully programmed to control the onboard accelerometer.

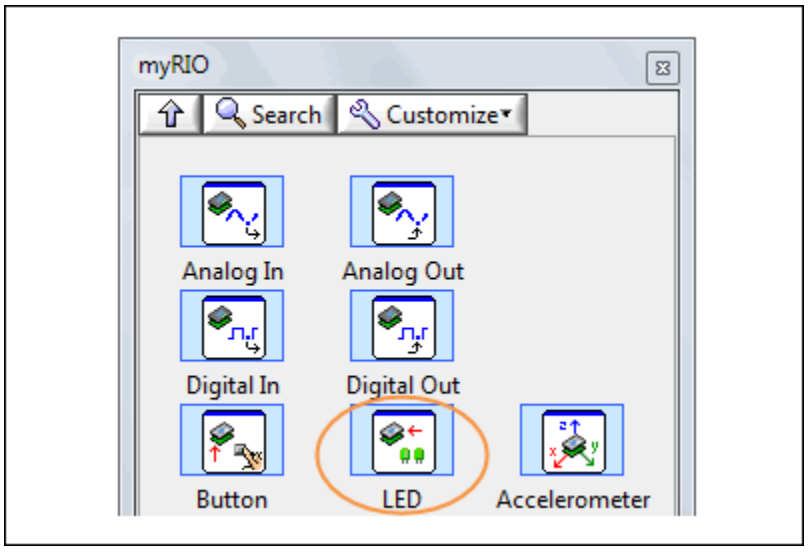
Before you proceed to the next part of the tutorial, spend some time understanding the block diagram of the Main VI. The block diagram uses a Flat Sequence structure that executes the following frames from left to right:

- **Initialize**—Initializes the **error in** cluster to specific values.
- **Acquire and process data**—Acquires acceleration values from the onboard accelerometer by using the Accelerometer Express VI and displays the acceleration values by using a waveform chart. The Main Loop repeats the code until you click **Stop** or an error occurs.
- **Close**—Resets the onboard accelerometer before the application exits.

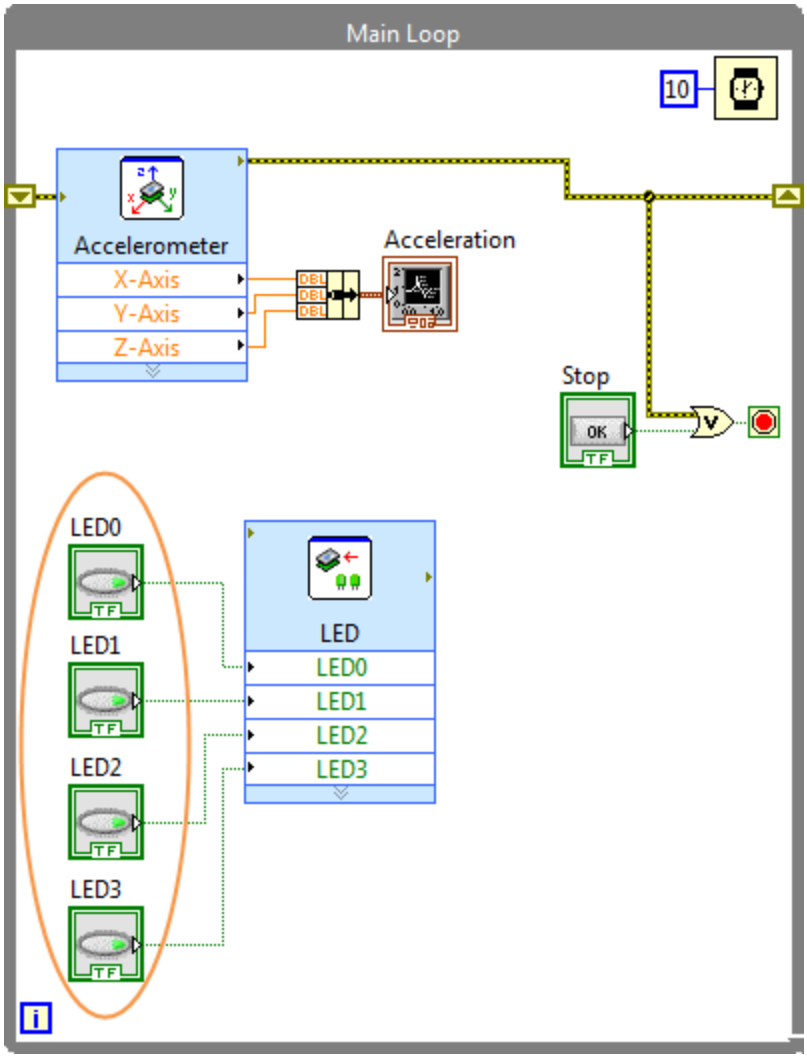
Now, proceed to program the onboard LEDs.

Part 3: Controlling the LEDs

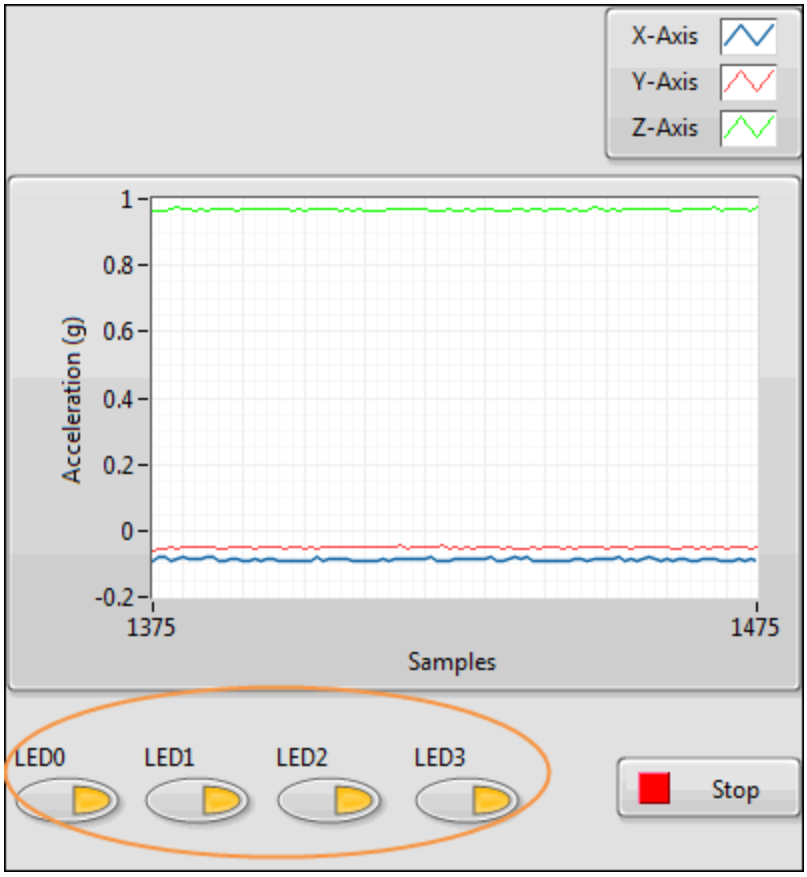
Remember how you toggled buttons in the **Getting Started with myRIO** wizard to change the myRIO onboard LED states? This part of the tutorial teaches you how to create an application to control the onboard LEDs. Before starting this part, make sure you have completed the previous parts of the tutorial.



1. On the block diagram of the Main VI, select **View»Functions Palette** to display the Functions palette. You can find the myRIO VIs on this palette.
2. Navigate to **Functions»myRIO** to locate the LED Express VI.
3. Click the LED Express VI and add this VI to the Main Loop in the Main VI.
4. In the configuration dialog box of the LED Express VI, select the LEDs to control and click **OK** to apply your configuration. This tutorial teaches you to control all the onboard LEDs.



5. Right-click each block diagram input of the LED Express VI and select **Create»Control** to create Boolean controls for the LEDs. You can click a Boolean control to toggle between the TRUE and FALSE states, which determines the ON and OFF states of an onboard LED.
6. Press <Ctrl-E> to switch to the front panel of the Main VI.



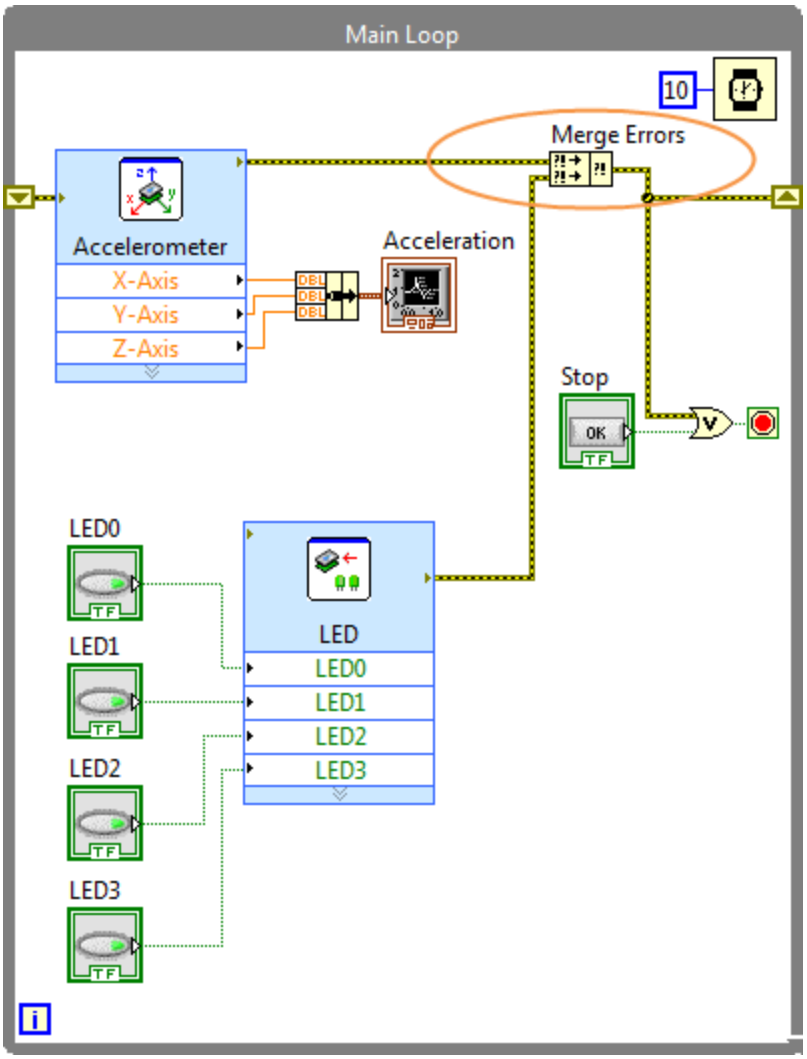
7. Click **Run**.
8. Toggle the Boolean controls and observe the state changes of the onboard LEDs.
9. Click **Stop**.
10. Click **File»Save** to save the VI.

Congratulations! You have successfully programmed to control the onboard LEDs.

Before you proceed to the next part, let's talk about error checking in LabVIEW. Error checking is important because it identifies why and where errors occur in your VI.

The current Main VI checks errors in the Accelerometer Express VI. If the Accelerometer Express VI returns an error, the Main VI stops. To ensure that the Main VI also stops when the LED Express VI returns an error, you must add code to check errors in the LED Express VI. Use the Merge Errors function to merge errors from the Accelerometer and LED Express VIs. You can access the Merge Errors function by navigating to **Functions»Programming»Dialog & User Interface»Merge Errors**.

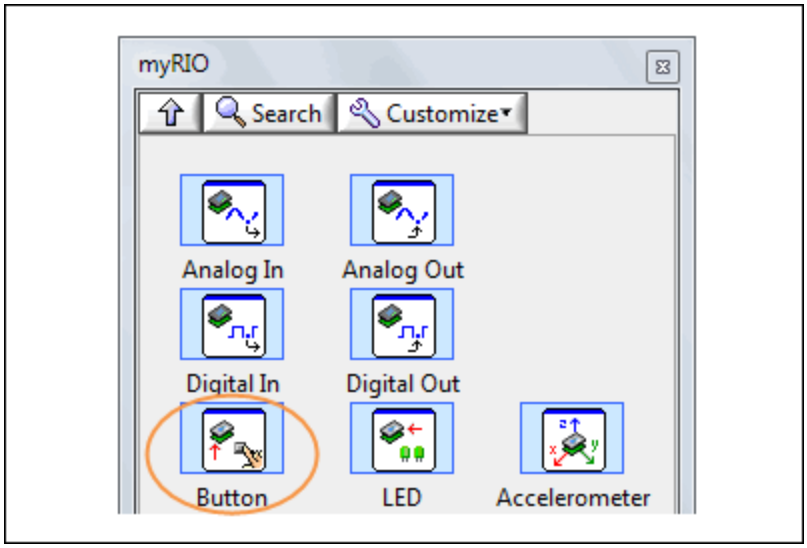
The following figure shows the block diagram after you add the Merge Errors function. The Main VI stops if the Accelerometer Express VI or the LED Express VI returns an error.



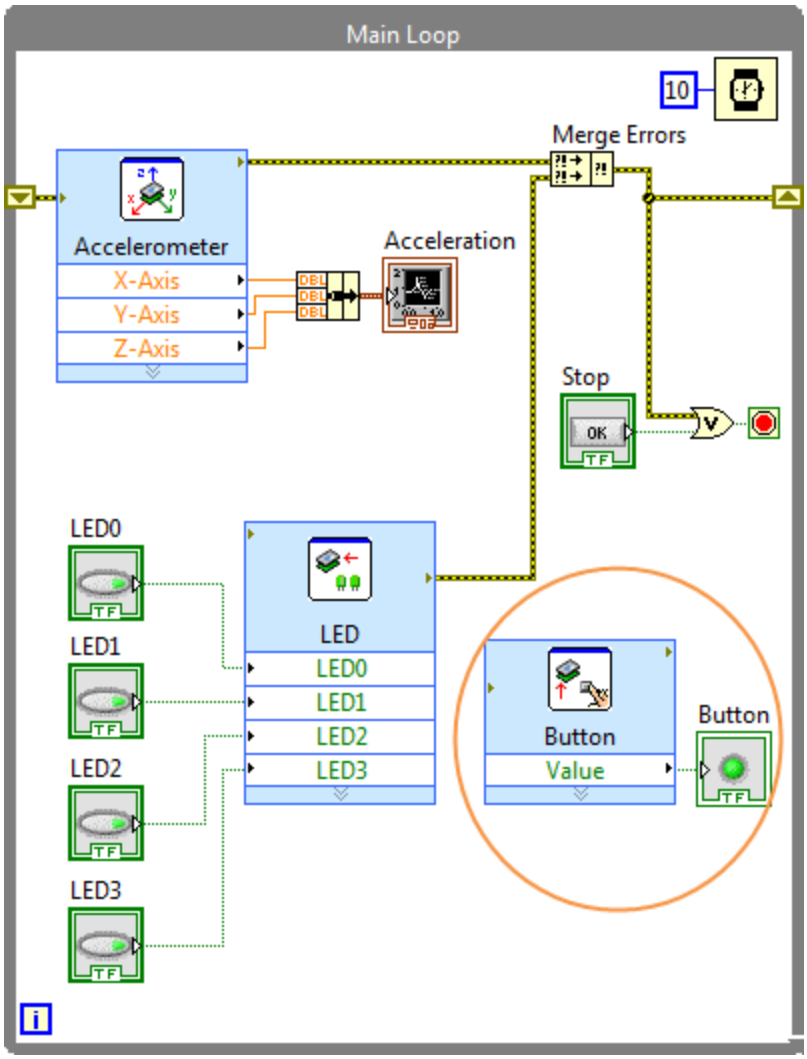
Now, proceed to control the onboard user button.

Part 4: Controlling the User Button

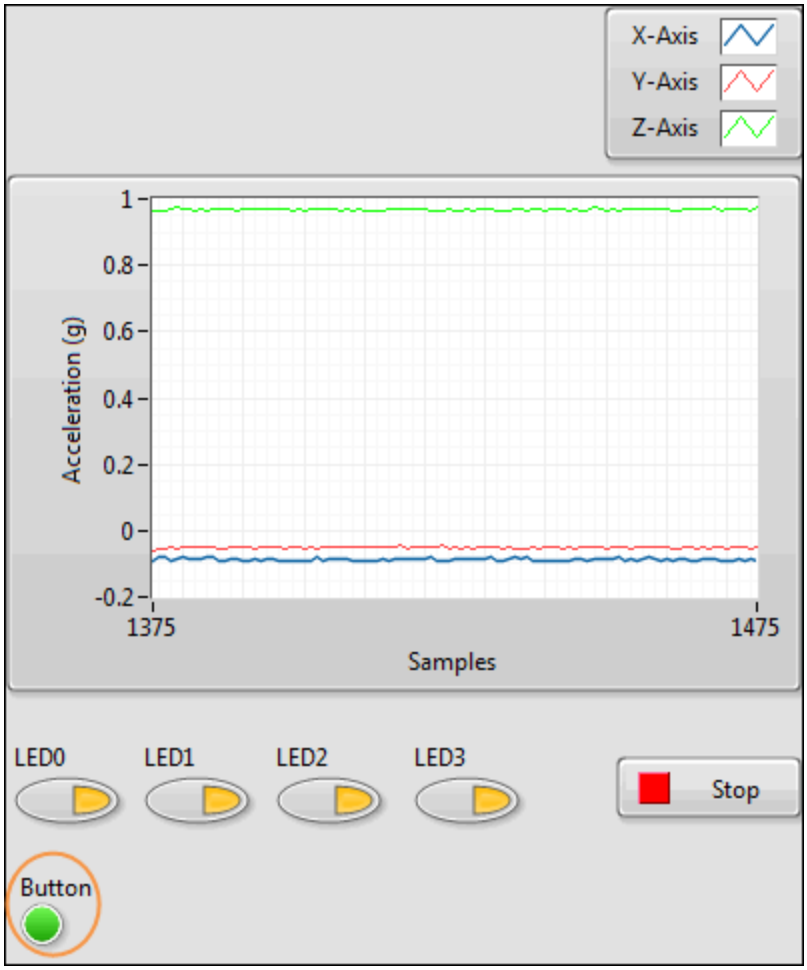
This part of the tutorial teaches you how to control the user button on the myRIO. Before starting this part, make sure you have completed the previous parts of the tutorial.



1. On the block diagram of the Main VI, select **View»Functions Palette** to display the Functions palette.
2. Navigate to **Functions»myRIO** to locate the Button Express VI.
3. Click the Button Express VI and add this VI to the Main Loop in the Main VI.
4. In the configuration dialog box of the Button Express VI, click **OK** to enable controlling the user button.



5. Right-click the **Value** output of the Button Express VI and select **Create»Indicator** to create a Boolean indicator for the user button. The TRUE and FALSE states of the Boolean indicator represent the ON and OFF states of the user button.
6. Rename the **Value** indicator to **Button**.
7. (Optional) Add error checking for the Button Express VI.
8. Press <Ctrl-E> to switch to the front panel of the Main VI.

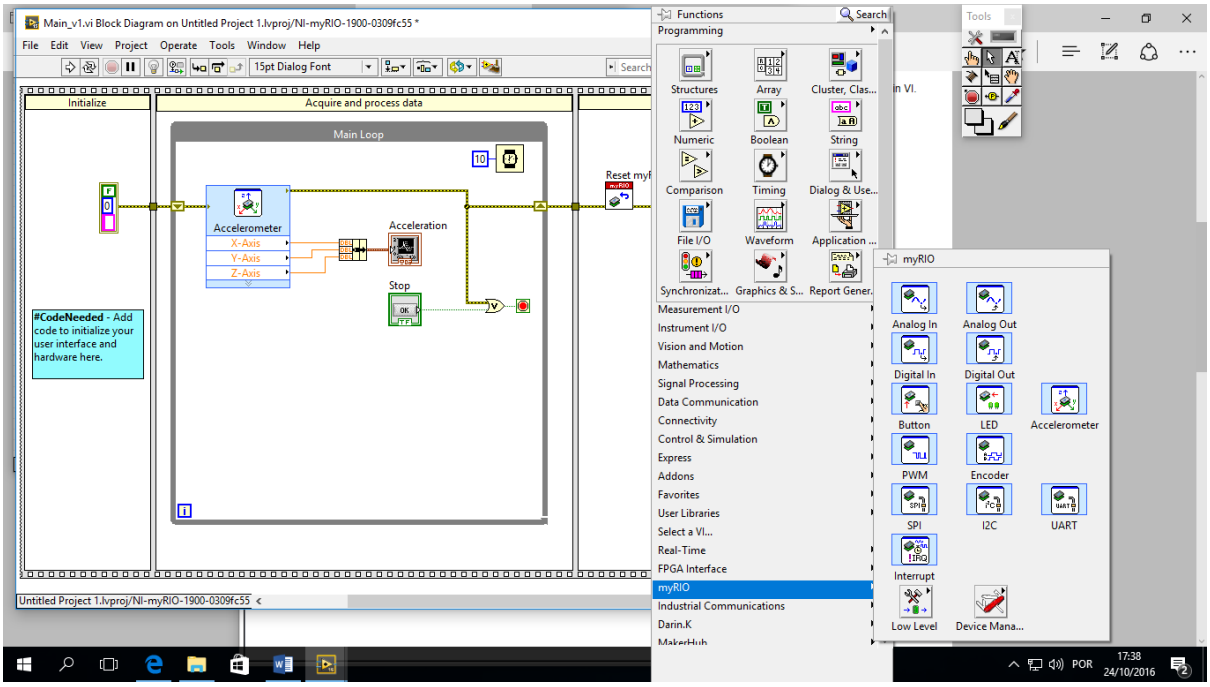


9. Click **Run**.
10. Press the user button on the myRIO and observe the state change of the **Button** indicator.
11. Click **Stop**.
12. Select **File»Save** to save the VI.

Congratulations! You have successfully created a myRIO application that controls all the myRIO onboard devices. You're ready to get started with creating applications of your own.

The following are some resources that you might find helpful as you create your own applications:

- *myRIO Toolkit Help*—Provides conceptual information about myRIO programming in LabVIEW and reference information about the myRIO VIs. To access this help file, select **Help»LabVIEW Help** in LabVIEW and navigate to the **myRIO Toolkit** book.
- Templates and sample projects—Templates provide starting points for useful design patterns. Sample projects demonstrate working applications based on these templates. You can customize templates and sample projects according to your application needs. Select **File»Create Project** from LabVIEW to display the **Create Project** dialog box. Look for the templates and sample projects under the **myRIO** categories.
- Online Resources—Refer to ni.com/learn-myrio for videos and more tutorials about RIO programming.


Main_v1.vi Block Diagram on Untitled Project 1.lvproj/NI-myRIO-1900-0309fc55 *
File Edit View Project Operate Tools Window Help
15pt Dialog Font
Initialize
Acquire and process data
Main Loop
Accelerometer
X-Axis
Y-Axis
Z-Axis
Acceleration
Stop
#CodeNeeded - Add code to initialize your user interface and hardware here.
Untitled Project 1.lvproj/NI-myRIO-1900-0309fc55
Functions
Programming
Structures
Array
Cluster, Clas...
Numeric
Boolean
String
Comparison
Timing
Dialog & Use...
File I/O
Waveform
Application ...
Synchronizat...
Graphics & S...
Report Gener.
Measurement I/O
Instrument I/O
Vision and Motion
Mathematics
Signal Processing
Data Communication
Connectivity
Control & Simulation
Express
Addons
Favorites
User Libraries
Select a VI...
Real-Time
FPGA Interface
myRIO
Industrial Communications
Darin.K
myRIO
Analog In
Analog Out
Digital In
Digital Out
Button
LED
Accelerometer
PWM
Encoder
SPI
I2C
UART
Interrupt
Low Level
Device Mana...
Tools
POR
17:38
24/10/2016

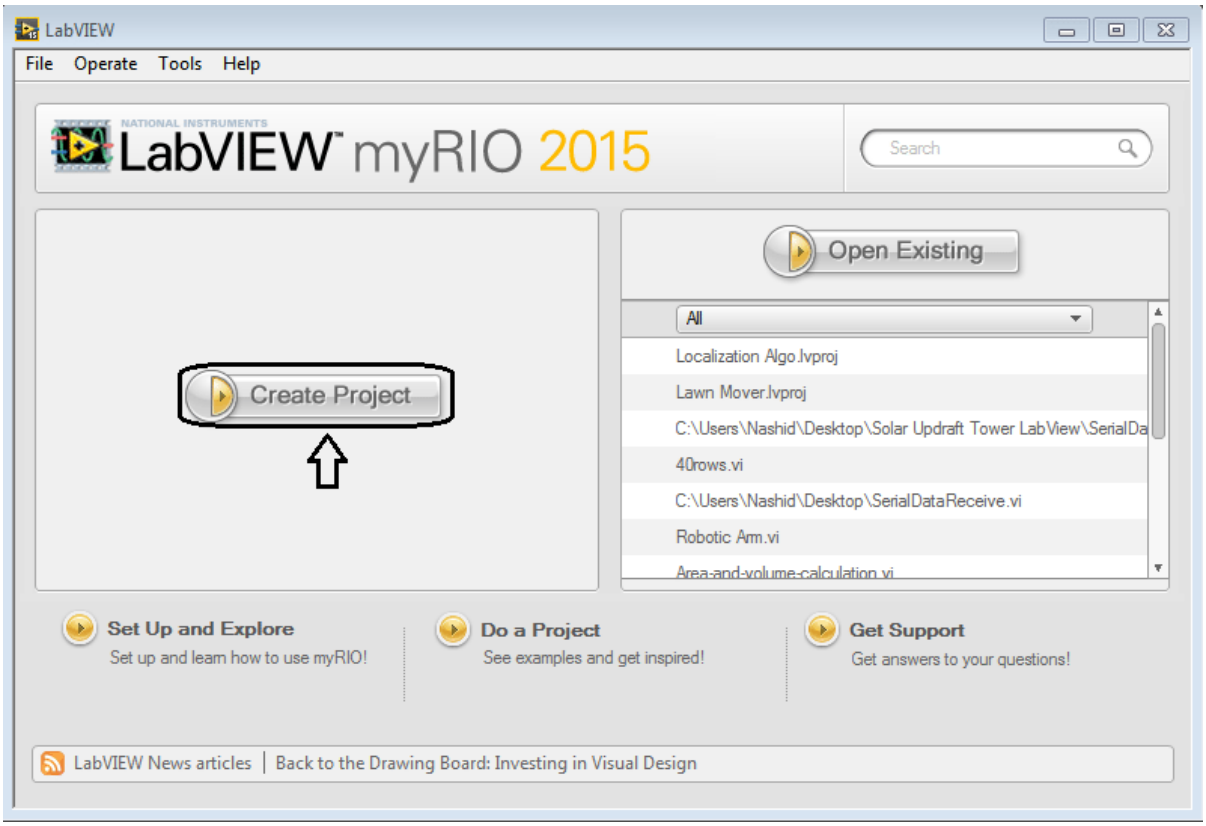
Creating First Program with myRIO

Here, in the tutorial ***Creating First Program with myRIO***, I will tell you the step by step procedure for creating your first program using NI myRIO. The tutorials basically focuses on controlling an LED on Front Panel with the help of the on board button on the device.

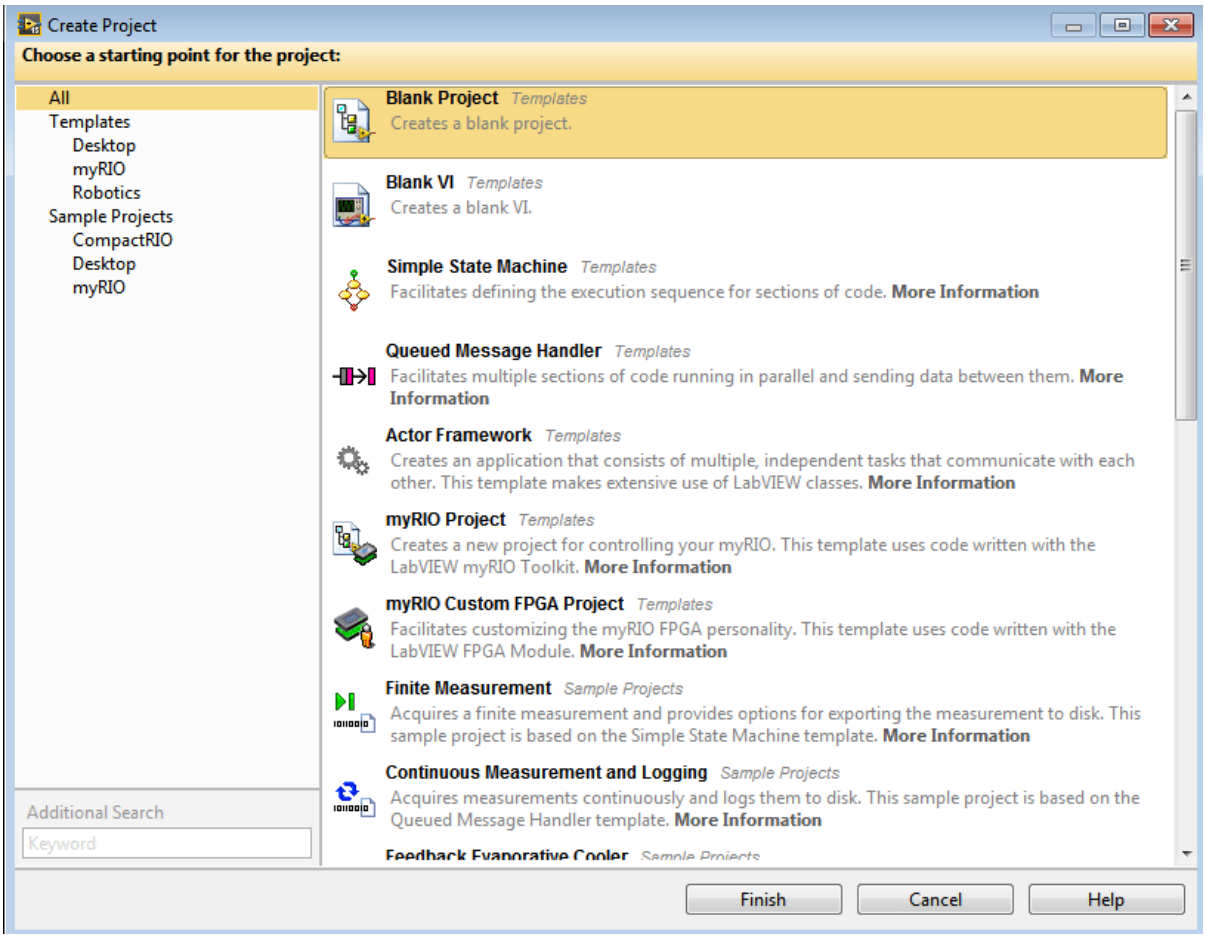
- You can download the complete Virtual Instrument (VI) for controlling LED using on board button here by clicking on the button below.
- Download .rar file, extract it and enjoy the LabVIEW NI myRIO's simulation.

Virtual Instrument (VI) Creation

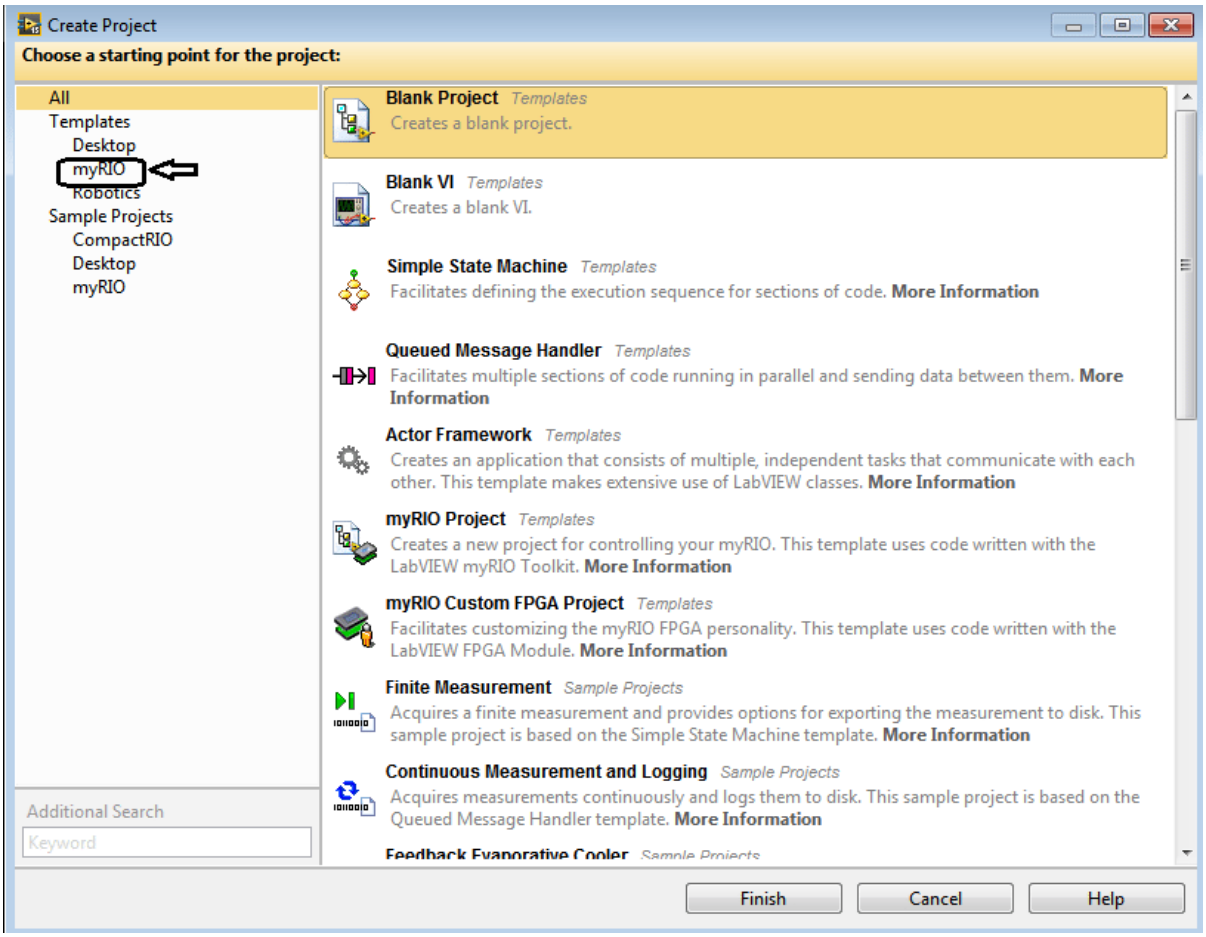
- First of all open the NI LabVIEW software from the softwares list.
- The starting window of the software is shown in the figure below.



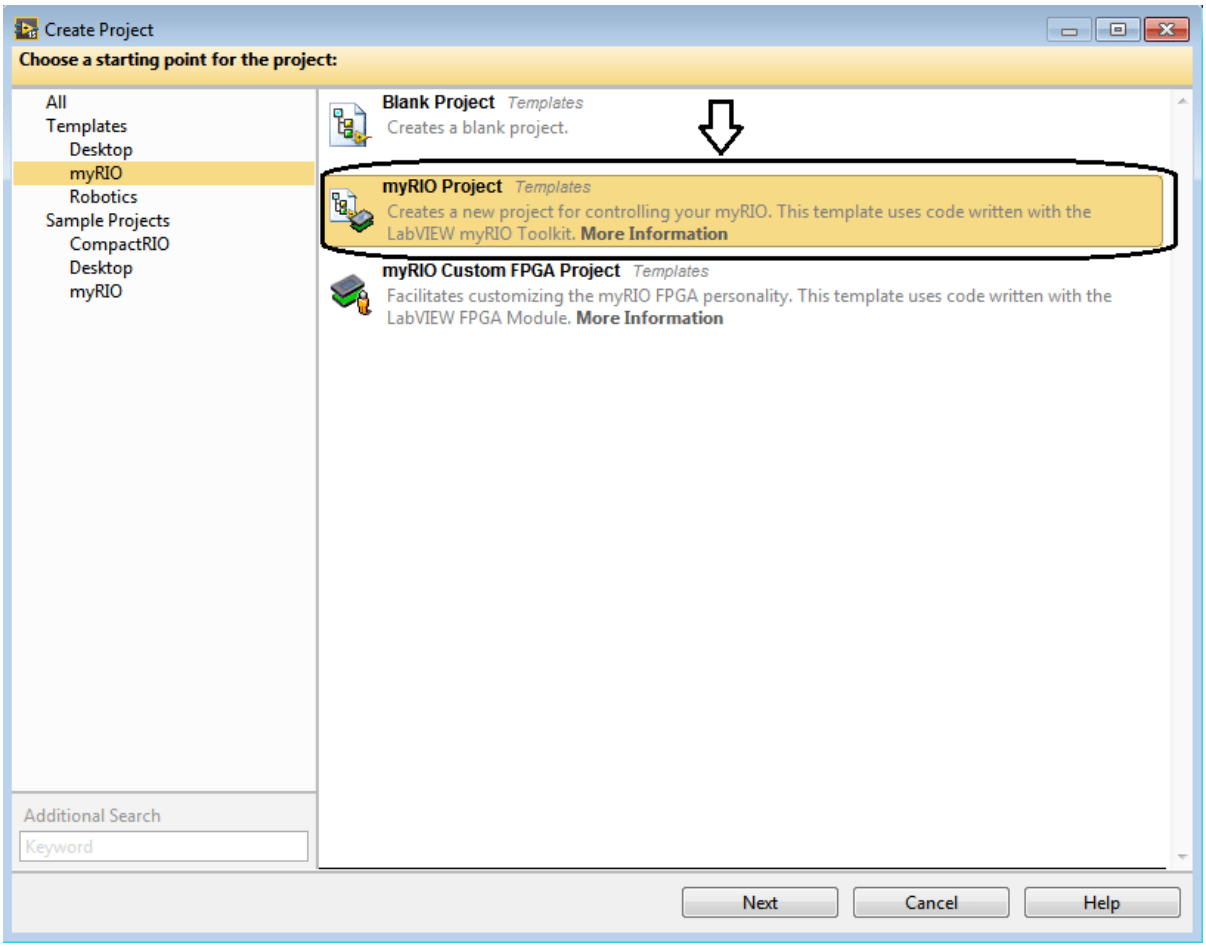
- You can see a lot of options on the starting window having different individual functions.
- Now, click on the encircled button **Create Project.**
- As you press this button a new window will be appeared on your computer screen.
- The new window with multiple options is shown in the figure below.



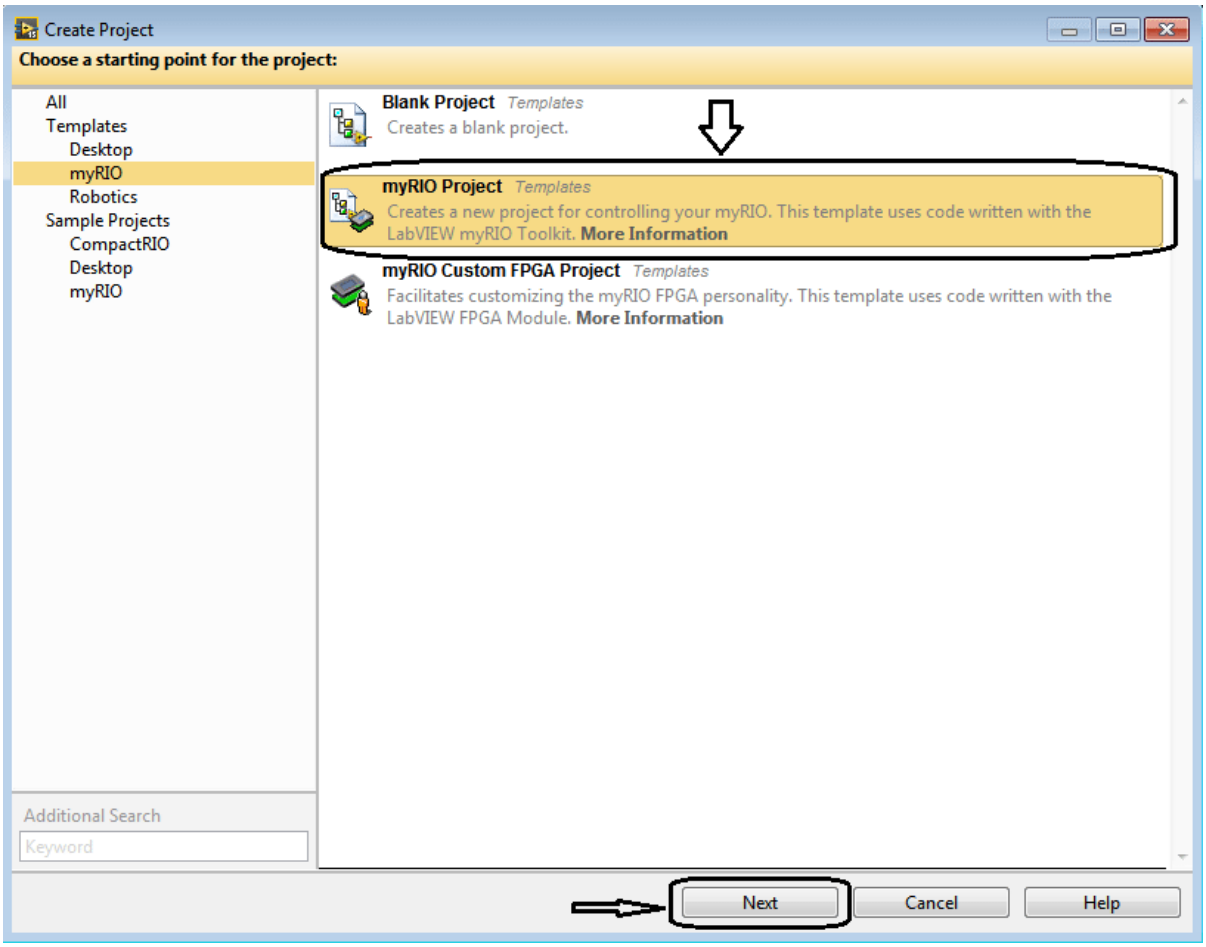
- Go to the **myRIO** option as shown in the figure below.



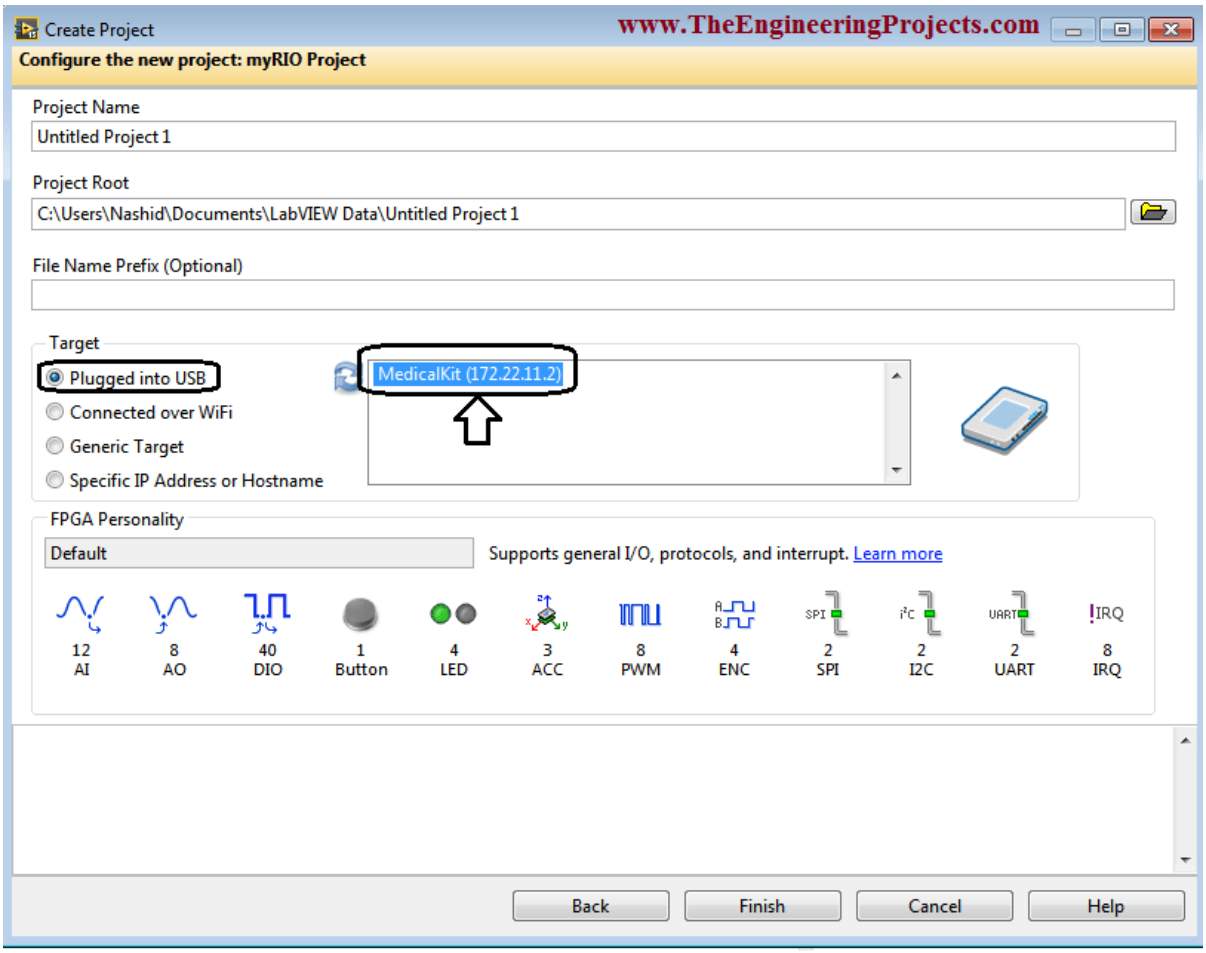
- While clicking on myRIO option, you can see the display on the right side will change by doing so.
- Select the **myRIO Project** from the right portion of the window**.**
- The both of the above steps are shown in the figure below.



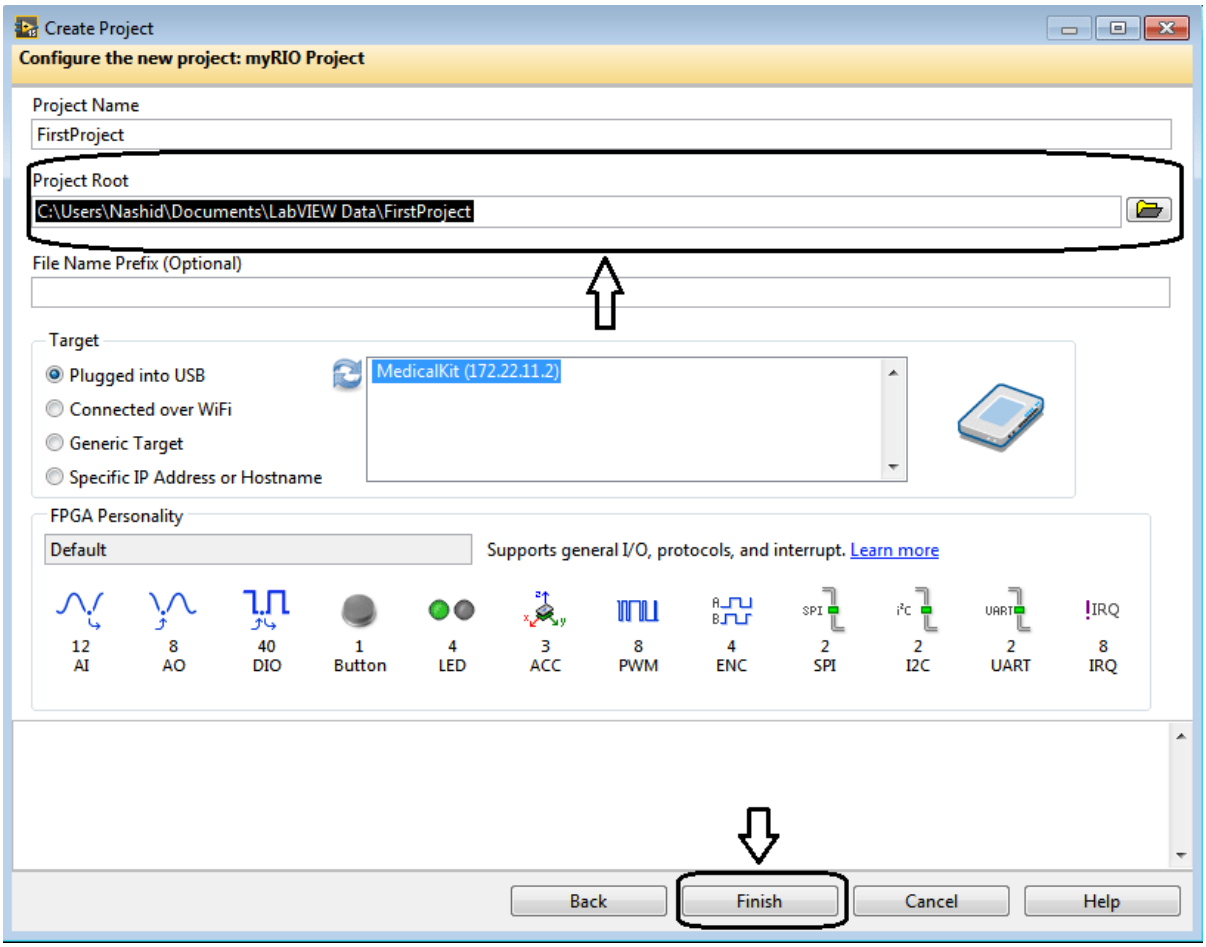
- Now, press the **Next** button as shown in the figure below.



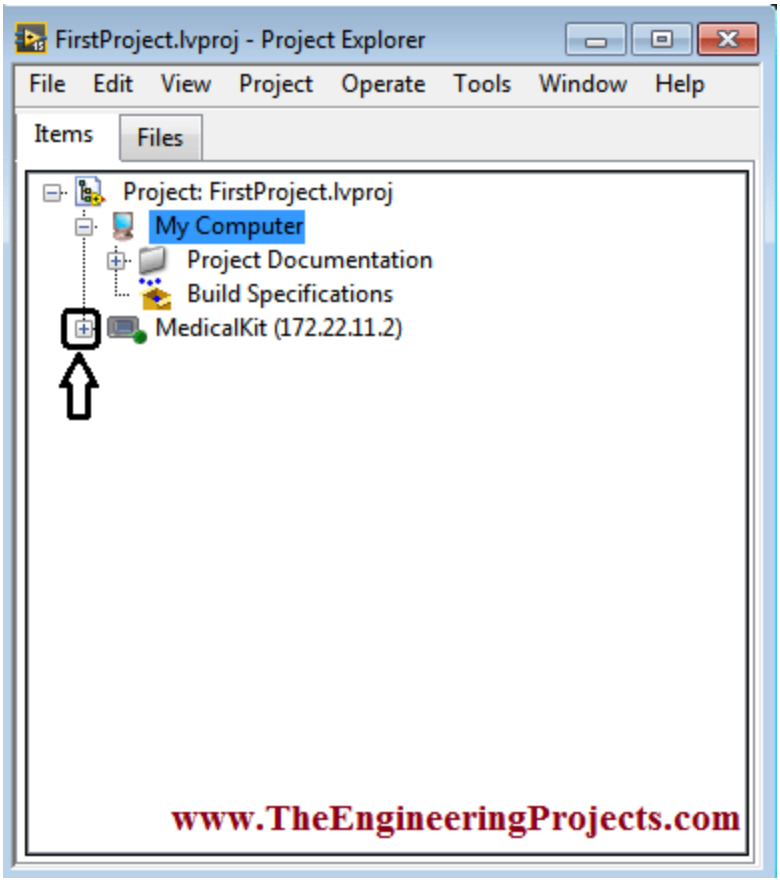
- Just after pressing this button you will be able to see a new window on your computer's screen.
- Select the **Plugged into USB** option and press **Refresh** button, your attached myRIO device will be detected in this way.
- IP address of the used NI myRIO is encircled in the below figure as **22. 11.2.**
- The newly opened window with a bit detail is shown in the figure below.



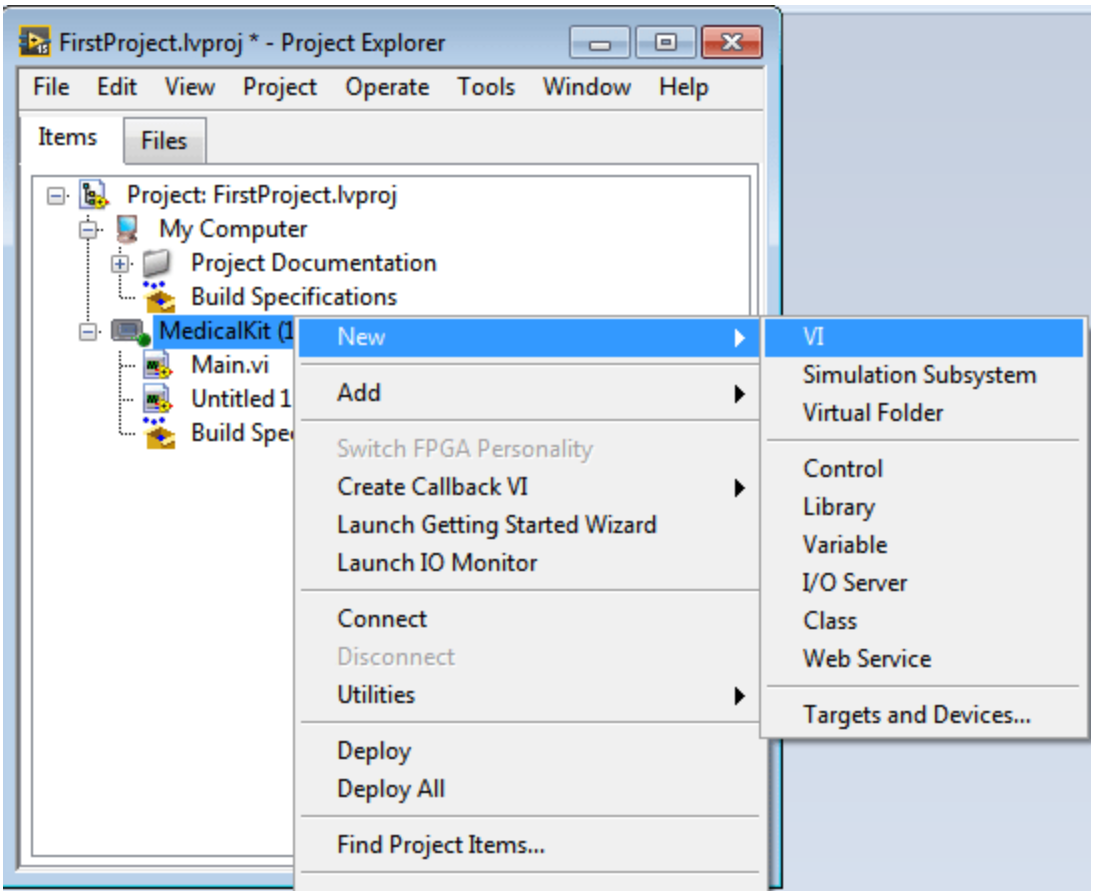
- The destination of the file which is being created is shown in the figure below.
- Now, press the **Finish** button as highlighted in the figure shown below.



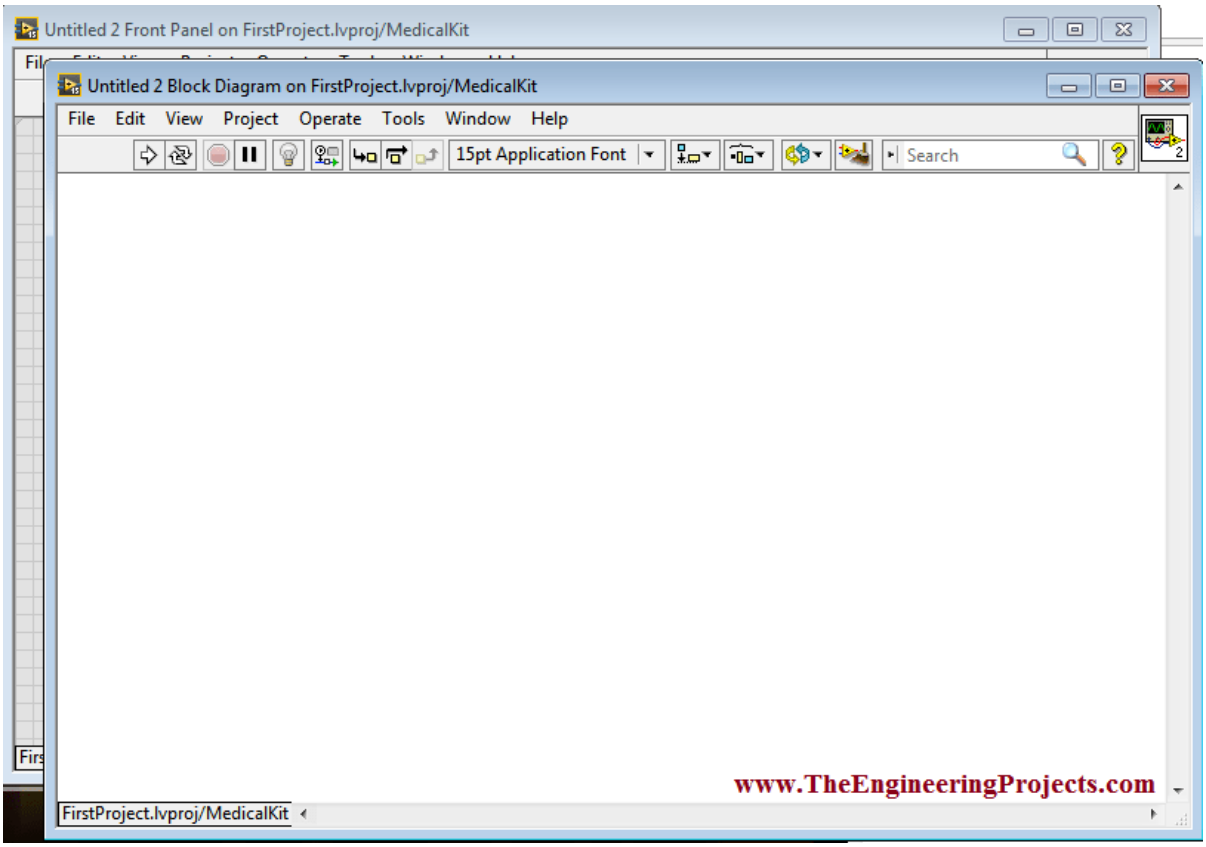
- As you press this button a new window will again appear on your computer's screen.
- Go to the **Medical Kit (172.22.11.2)** as encircled in the figure below**.**
- The new window appeared with all of the above steps is shown in the figure below.



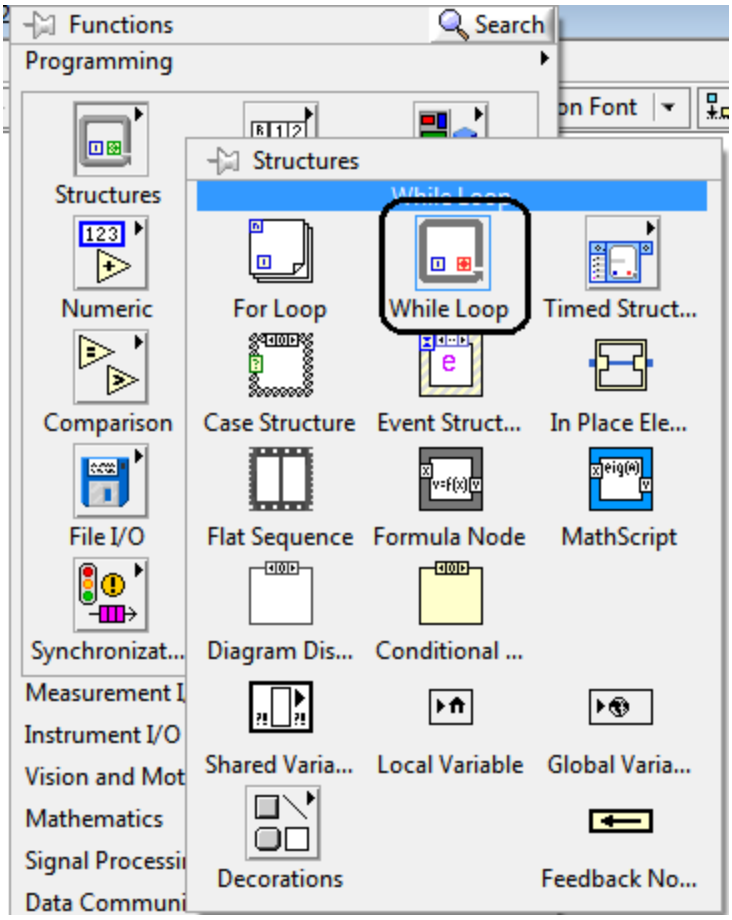
- Now go to the **Medical Kit** option and **Right Click** on it.
- By doing so, you can see a lot of options here in the new mini window, go to **New-> VI.**
- The above steps are shown in the figure given below.



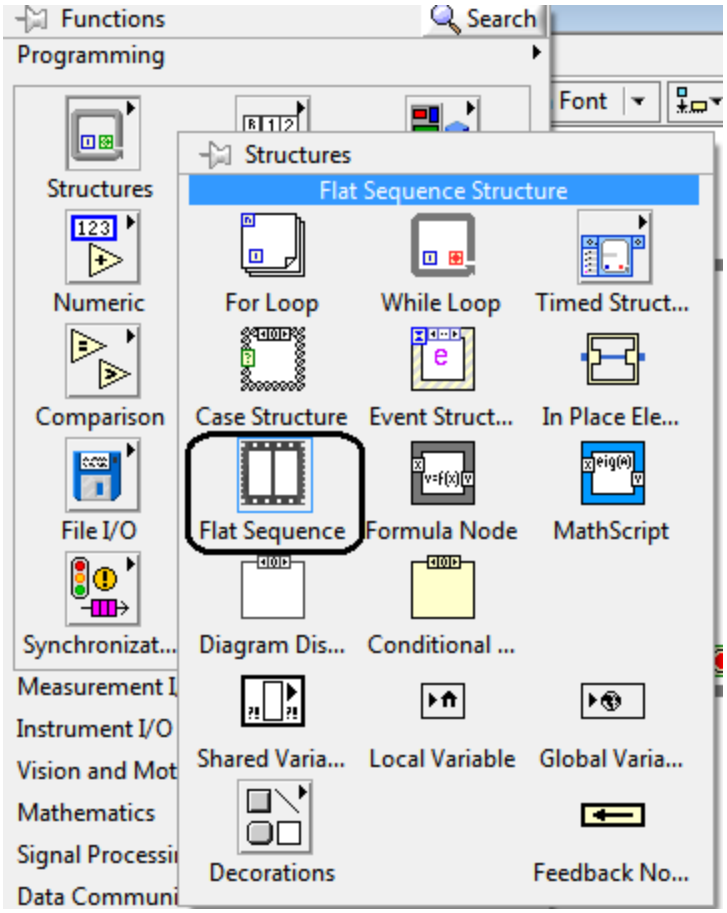
- As u press the VI two windows, Front Panel and Block Diagram will be appeared on your computer's screen.
- Both the windows are shown in the figure below.



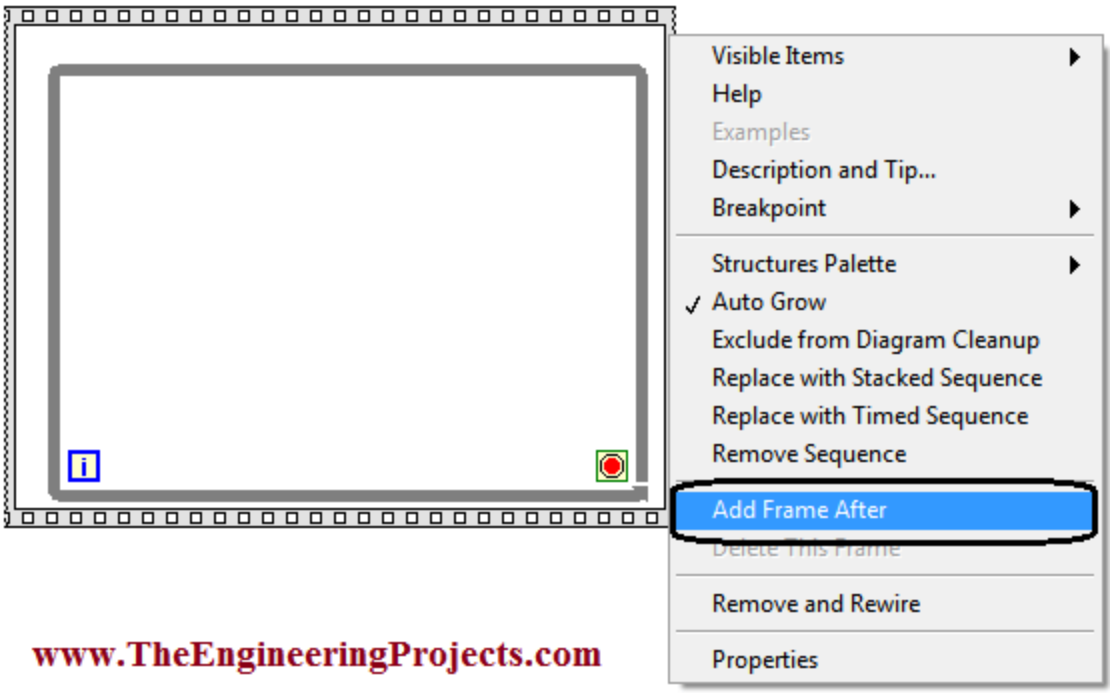
- Now, press **Ctrl+t,** both of the windows will be aligned in this way.
- Now go to the Block Digram window and **Right Click** on it.
- Go to the **Functions-> Programming-> Structures** here you can see different structures blocks.
- Select the **While Loop** as encircled in the figure shown below.



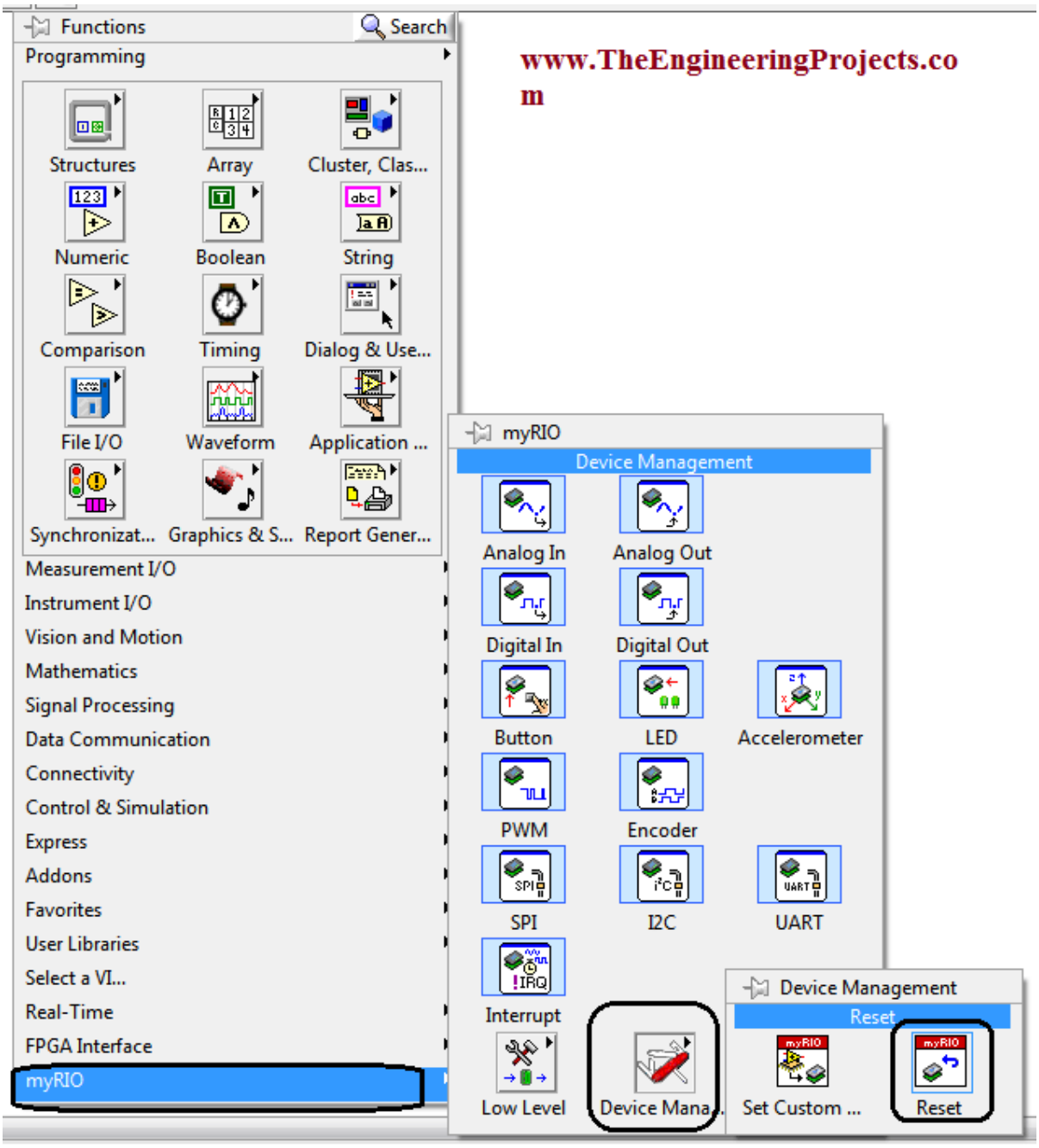
- Select this block and place it on the Block Diagram window as shown I the figure below.
- Now, go to the **Functions-> Programming-> Structures** once again.
- Select the **Flat Sequence** and place it around the **While Loop.**



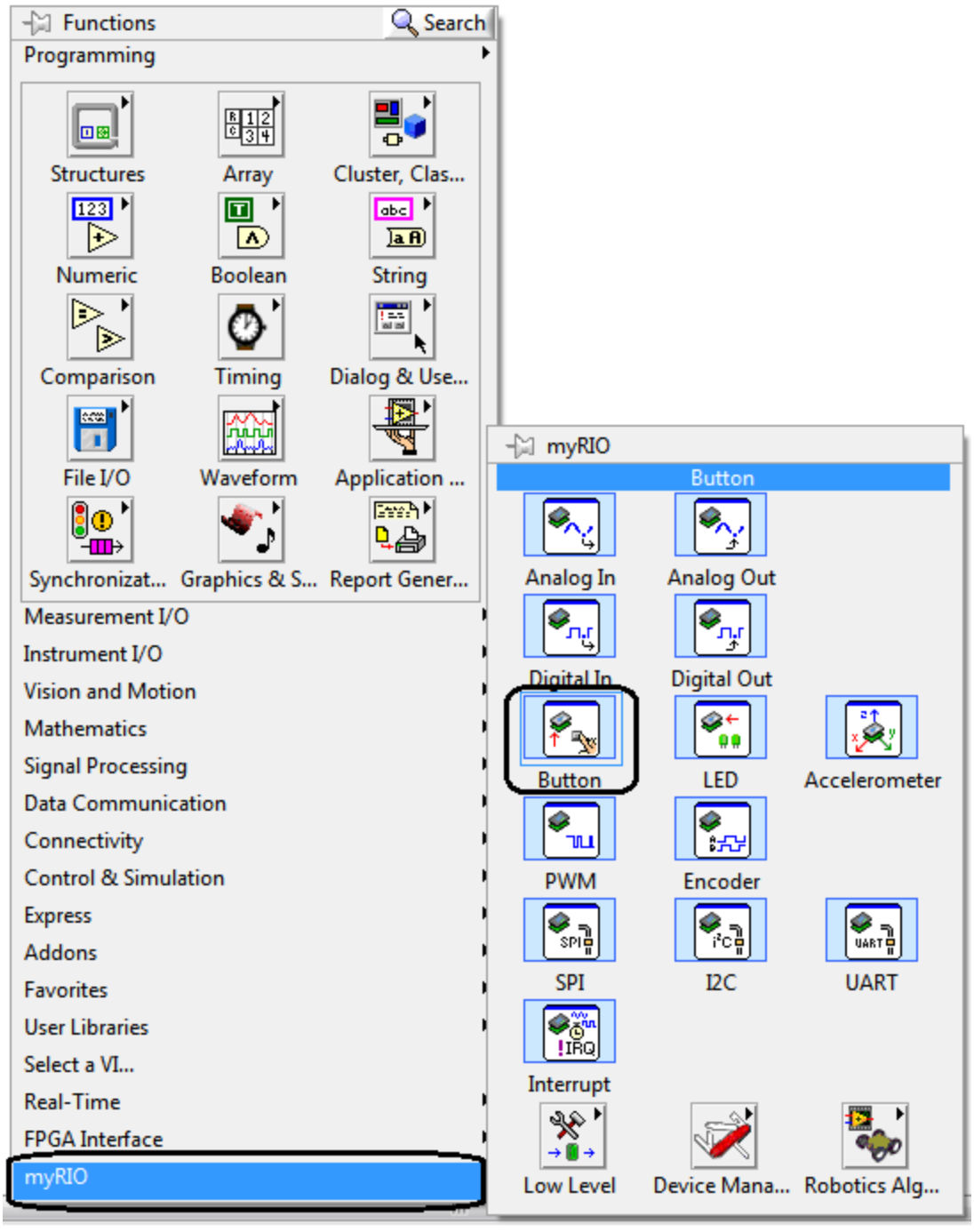
- Now put your cursor on the right side of the Flat Sequence and Right Click on it.
- Go to the **Add Frame After** as shown in the figure below.



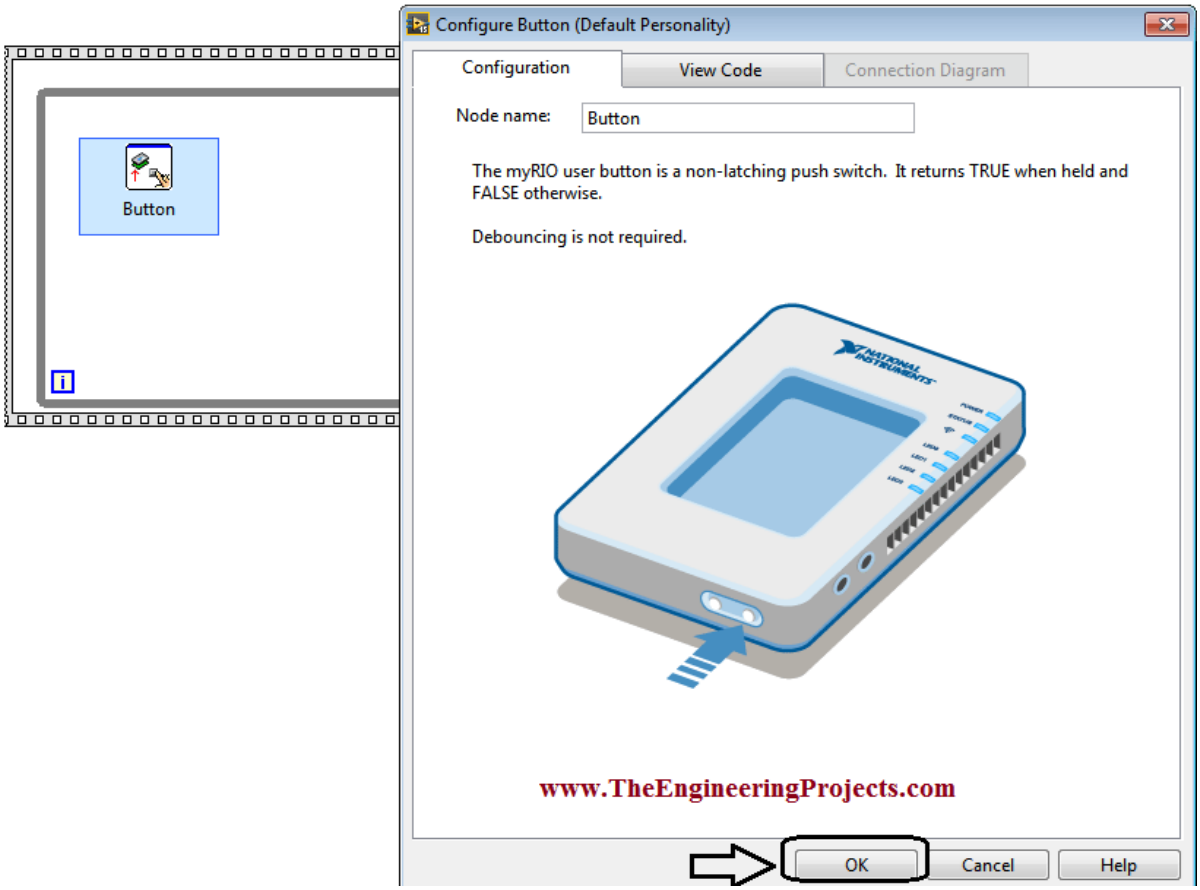
- Now, **Right Click** on the Block Diagram window and go to **Functions-> myRIO-> Device Management,** here you can see the two different management blocks.
- Select the encircled block **Reset myRIO** as shown in the figure below.



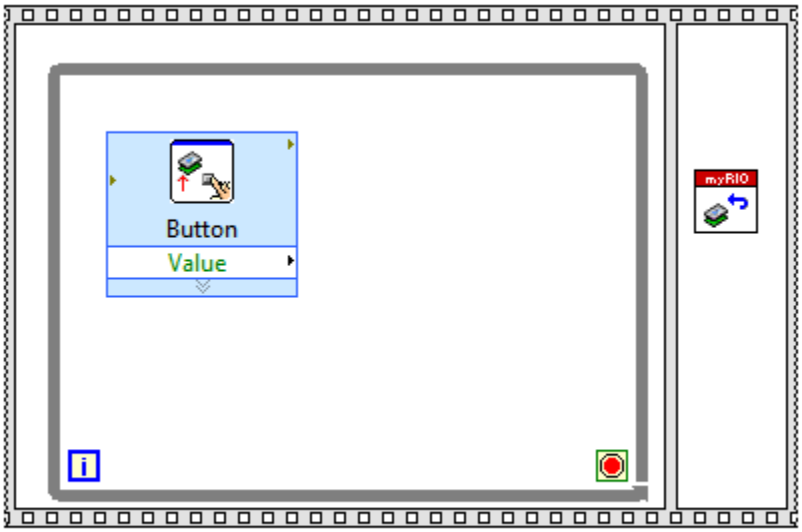
- Place the encircled block **Reset myRIO** inside the second frame of the Flat Sequence.
- Now go to the **Functions-> myRIO** and here you can see the different myRIO blocks.
- Select the encircled block **Button** as shown in the figure below.



- Place **Button** inside the **While Loop.**
- As you place the button a new window will be appeared on your screen.
- New window is shown in the figure below, just press **Ok** as shown below.

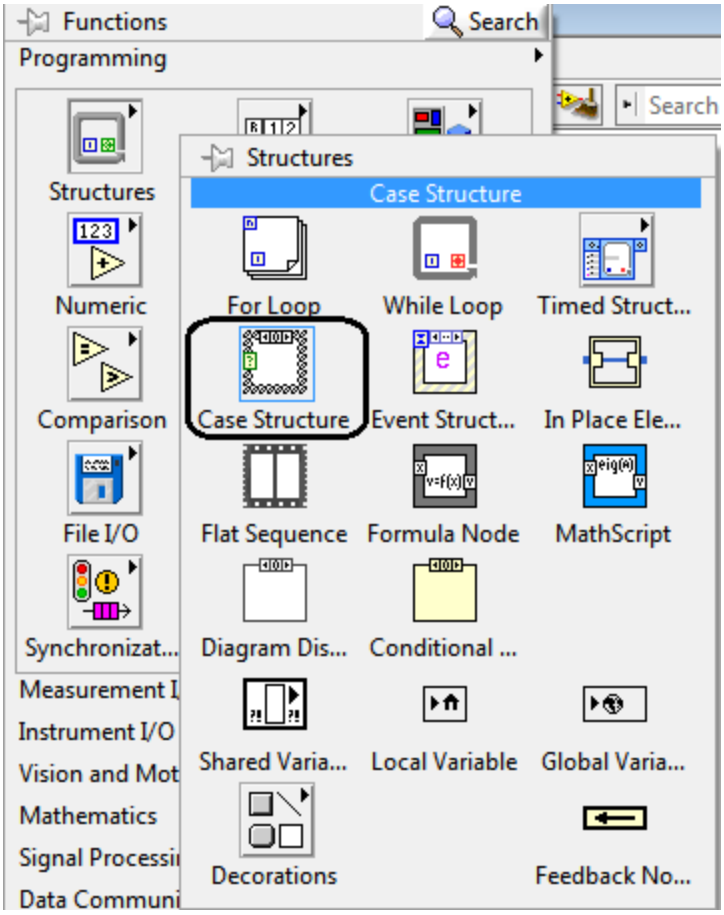


- All of the above steps are shown in the figure below.

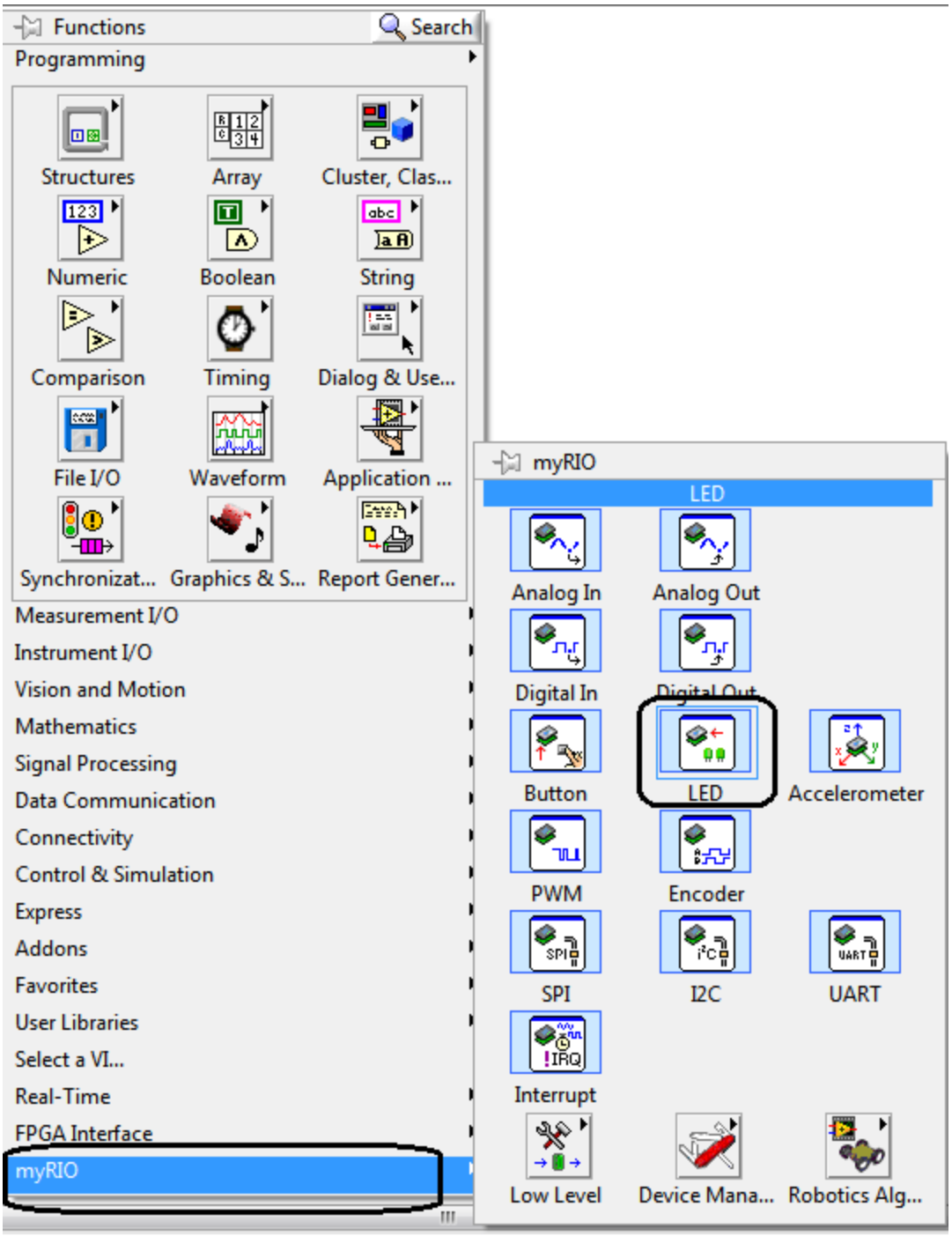


- Now go the **Functions-> Programming-> Structures,** here you can see the different structure blocks.

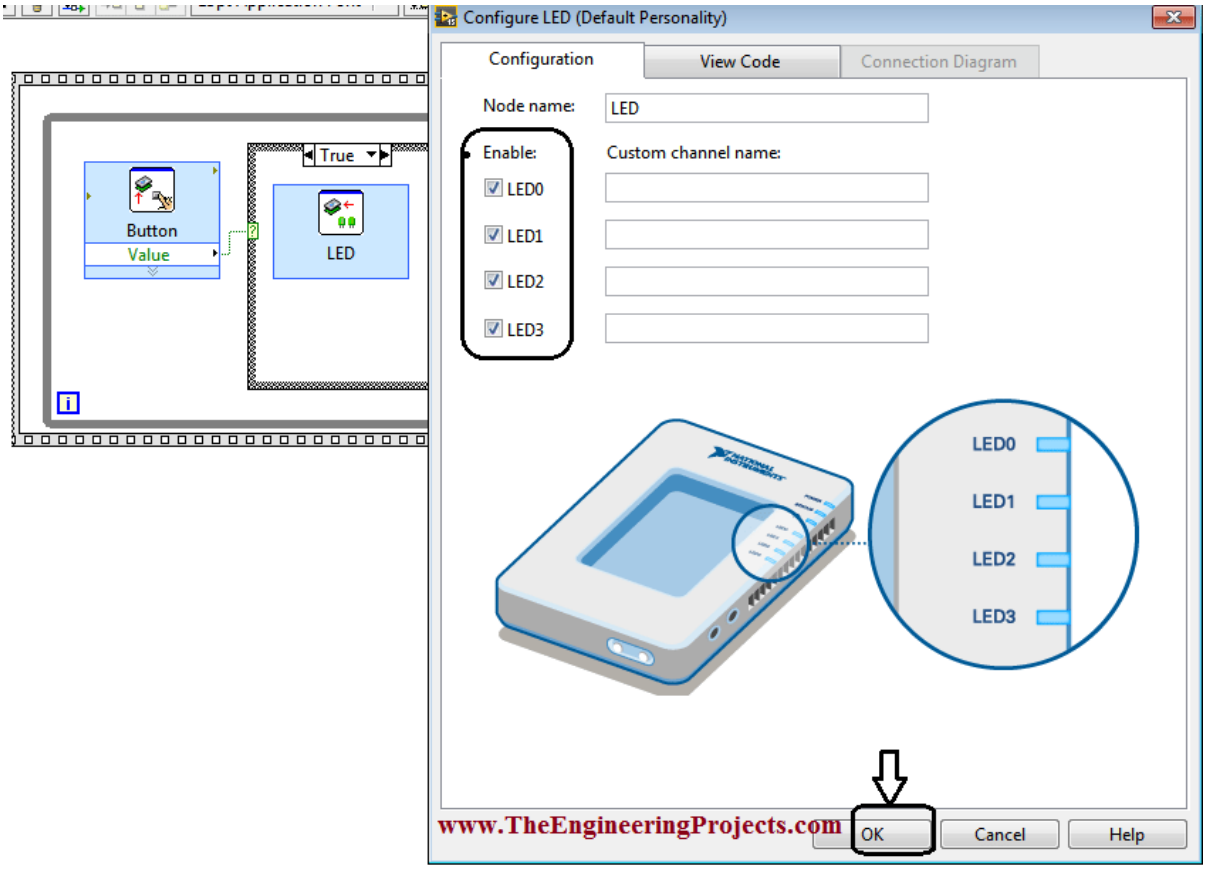
- Select the encircled block **Case Structure** as shown in the figure below.



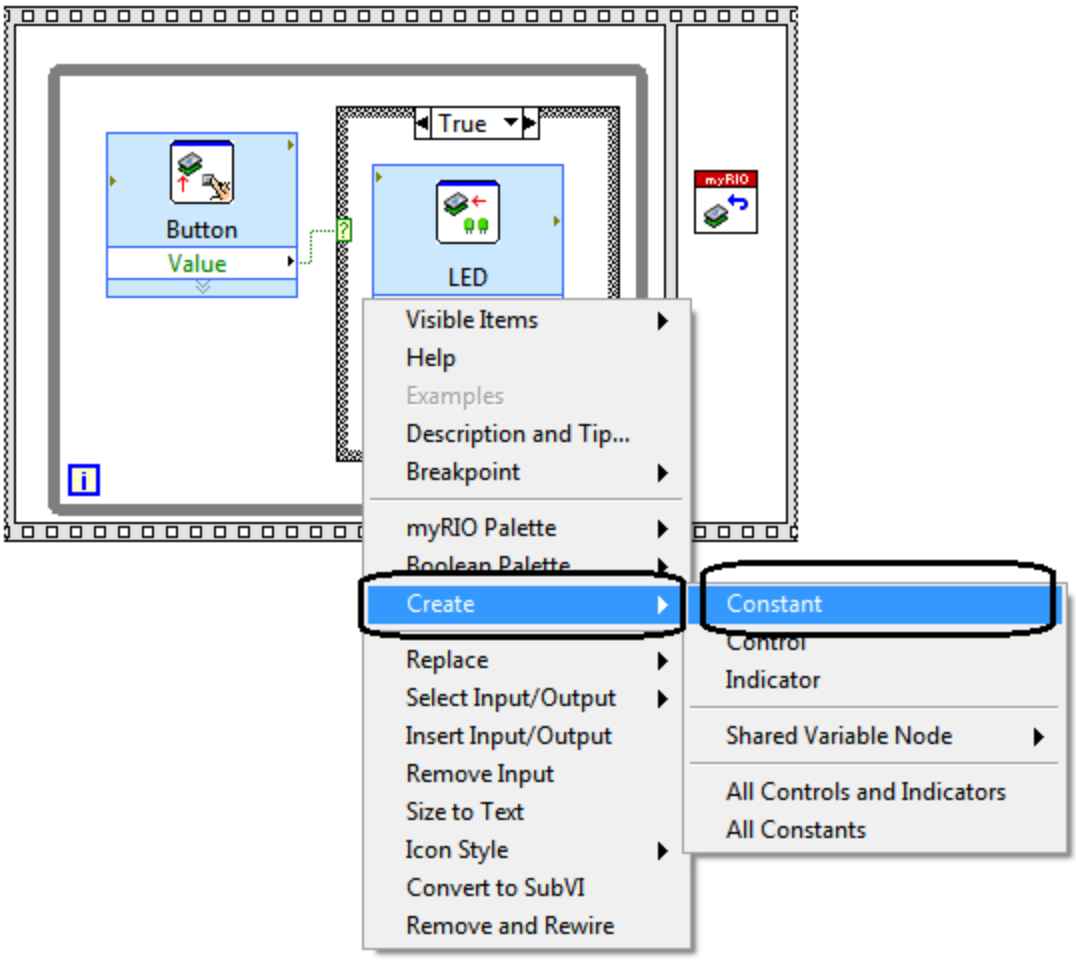
- Now go to the **Functions-> myRIO,** here you will see a lot of myRIO blocks.
- Select the encircled block **LED** as shown in the figure below.



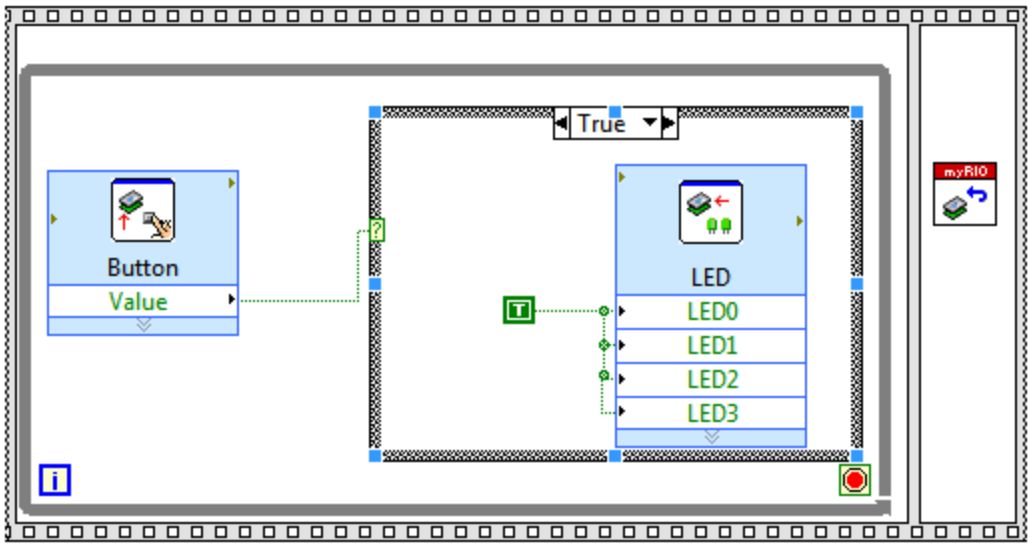
- Select the encircled block and place it inside the **Case Structure.**
- As you place the LED in the Case Structure a new window will be appeared on the screen, just press **Ok**.
- The figure shown below exhibits the above steps.



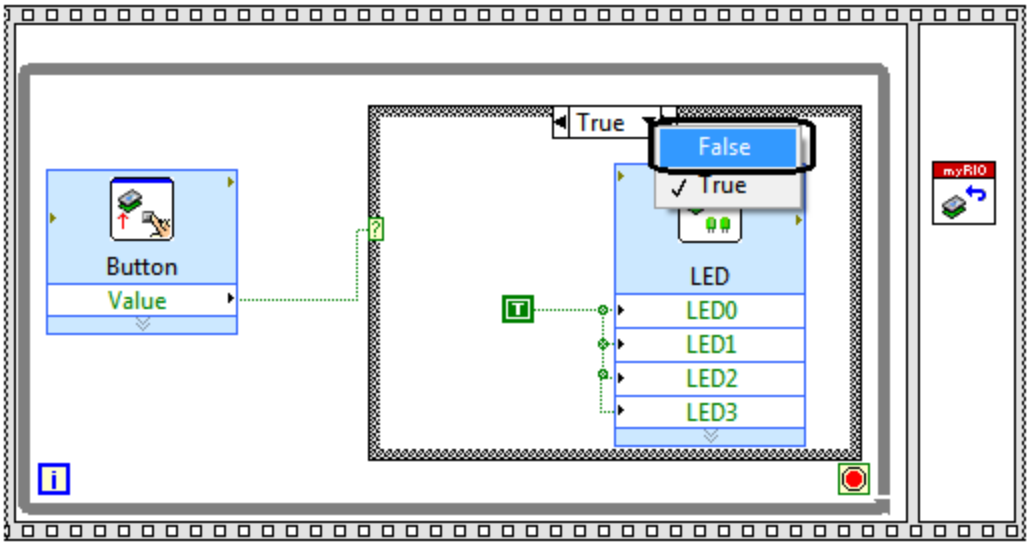
- Now to the First input terminal of the LED and Right Click on it and go to **Create-> Constant.**
- By doing so, four inputs will be added to the LED.
- The above step is shown in the figure below.



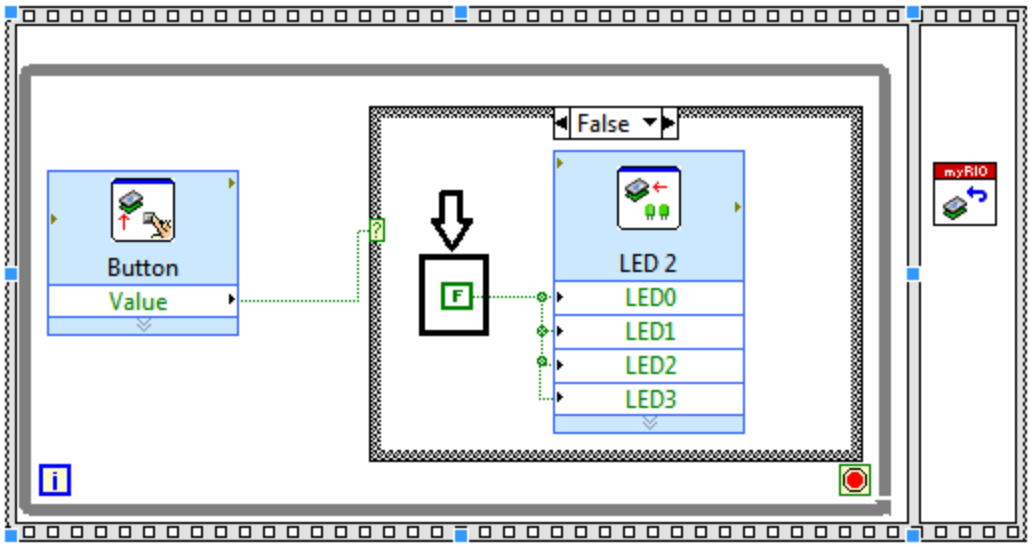
- And change the condition from False to the True e.g. **F** to
- The figure shown below exhibits the above step.



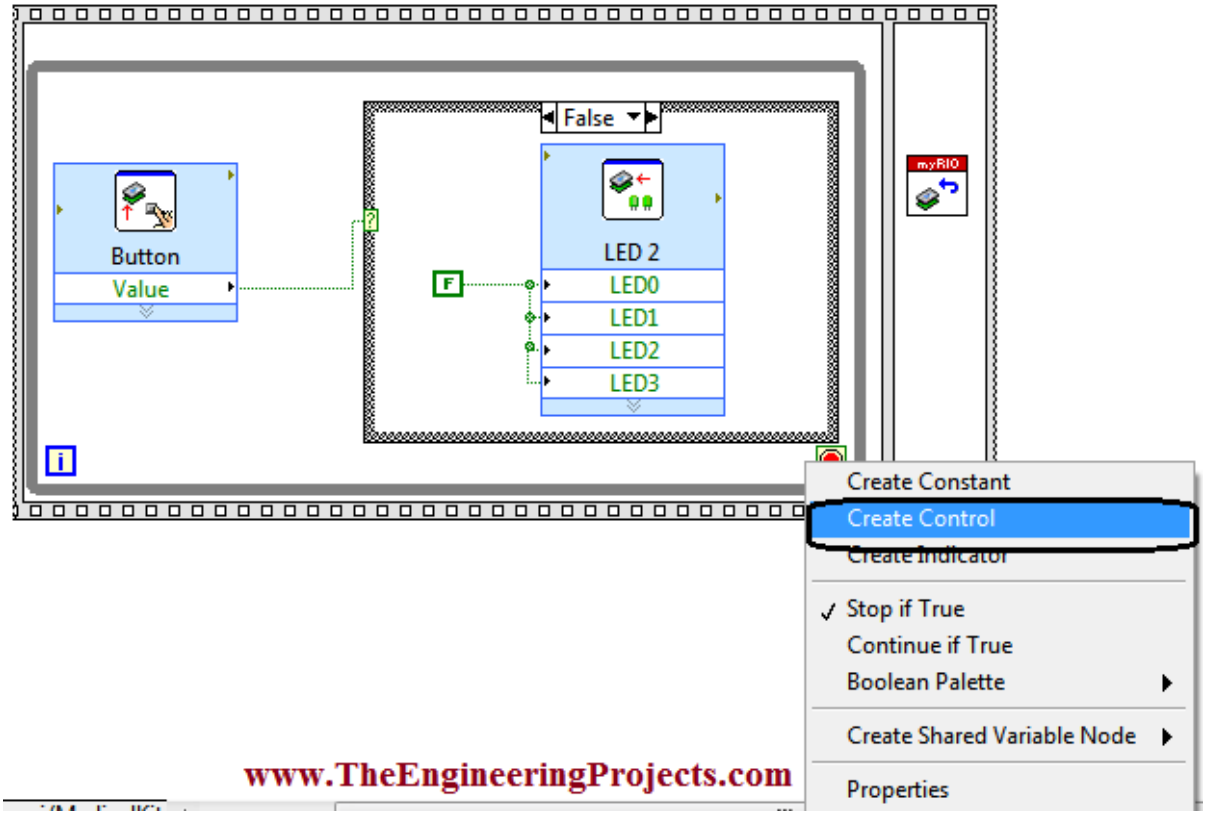
- Now copy the entire code inside the Case Structure and go to the **False Option.**



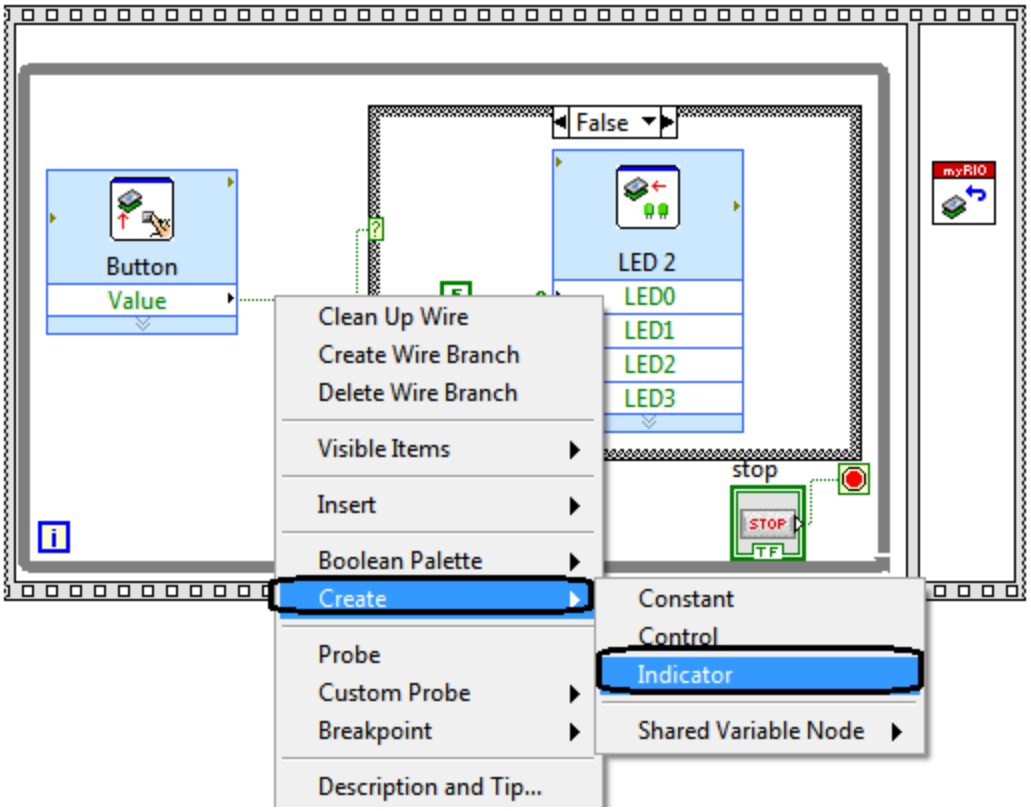
- Paste the copied code inside the False Option and change the condition from True to False i.e. form **T** to
- The above step is shown in the figure below.



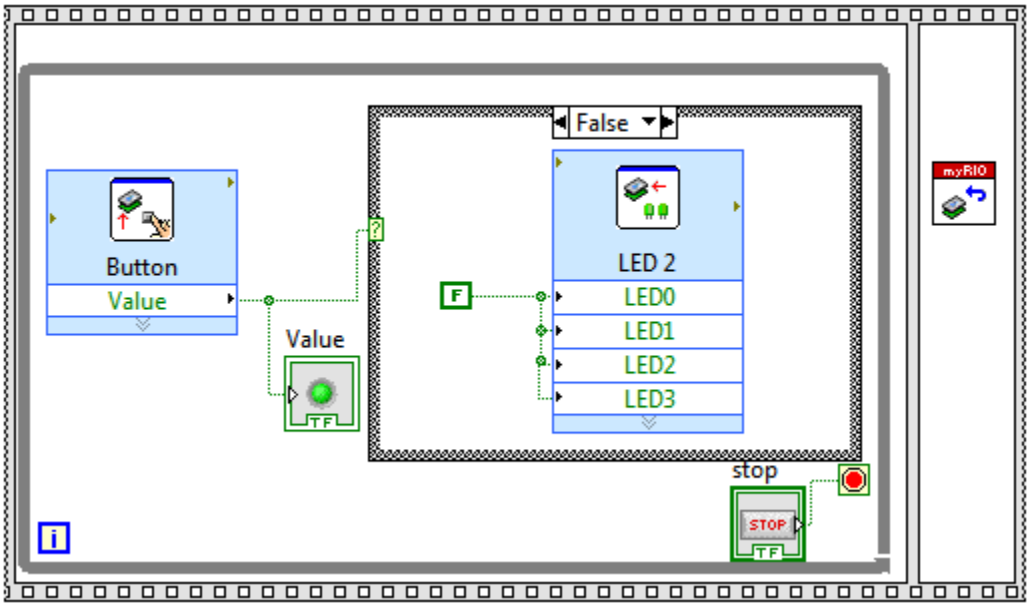
- Now go to the bottom right of the While Loop and go to **Create-> Control** as shown in the figure below**.**



- Now go to output terminal of the **Button** right click on it.
- Go to the **Create-> Indicator** as shown in the figure below**.**

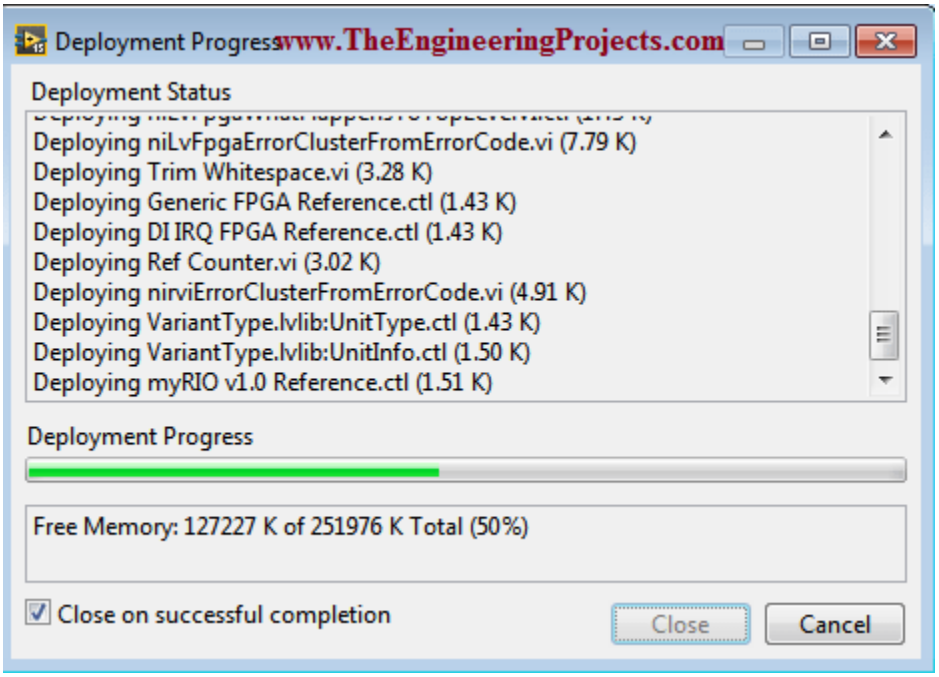


- By doing so, a new indicator will be added on the block diagram window to show the state of the button i.e. whether it is pressed or not.
- The above step is exhibited in the figure below.

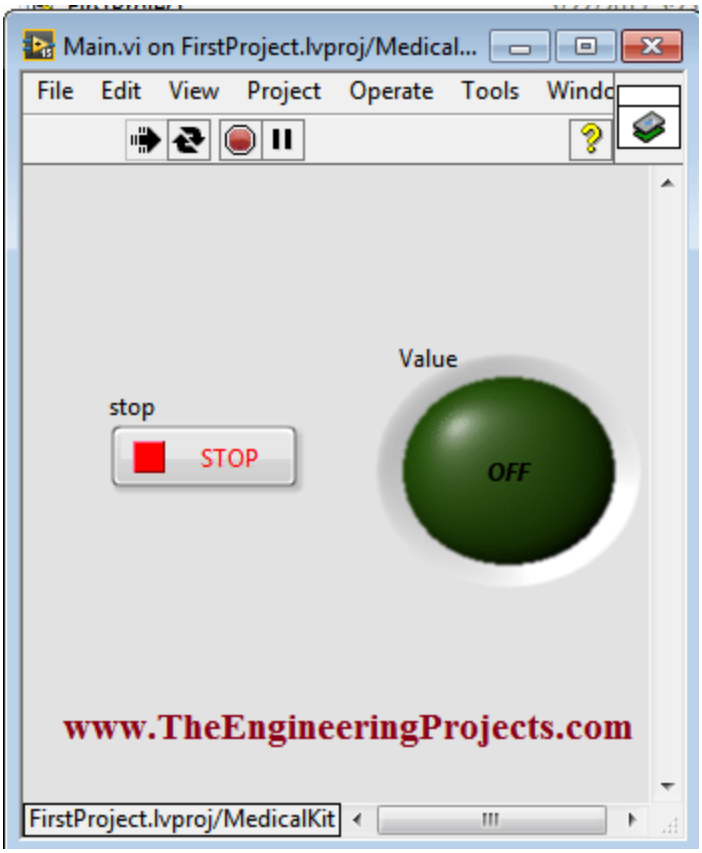


- Now run the program and the code will be started to upload on the NI myRIO.

- The uploading in process is shown in the figure below.



- The Front Panel is shown in the figure below.



- Now if you press the button on the myRIO the Green LED on the Front Panel will be turned on.
- The figure shown below exhibits the above step.

- The **ON** state of the LED shows that the button on the myRIO is pressed.