LEGO® MINDSTORMS® NXT Lab 1

This lab session is to introduce you how to connect your LEGO Mindstorms NXT brick and Wii remote controller (Wiimote) to the laptop or computer via Bluetooth wireless protocol.

Lab Sections

A. Use the LEGO Mindstorms NXT software to connect NXT brick to your computer
B. Connect Wii remote controller to your computer.
C. Run NI LabVIEW sample program and use Wiimote to control NXT Tribot.

Software

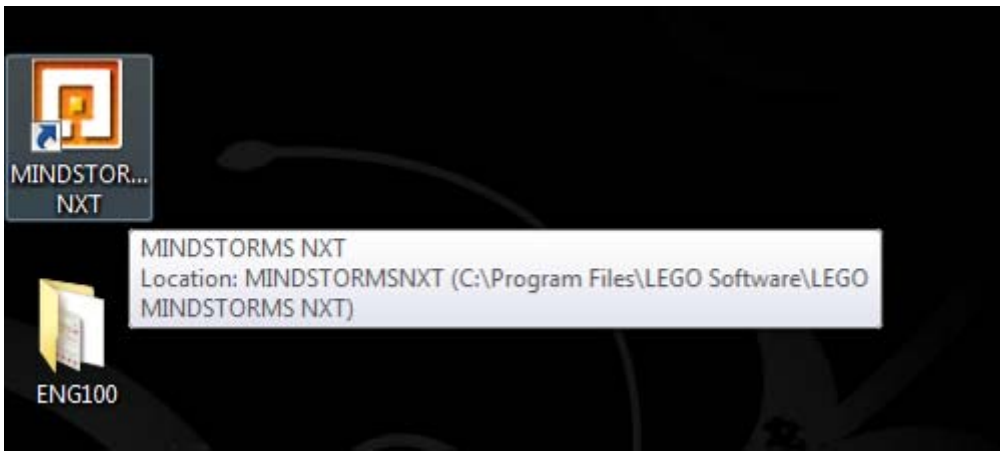
- Lego Mindstorms NXT
- NI LabVIEW with Lego NXT toolkit

Hardware

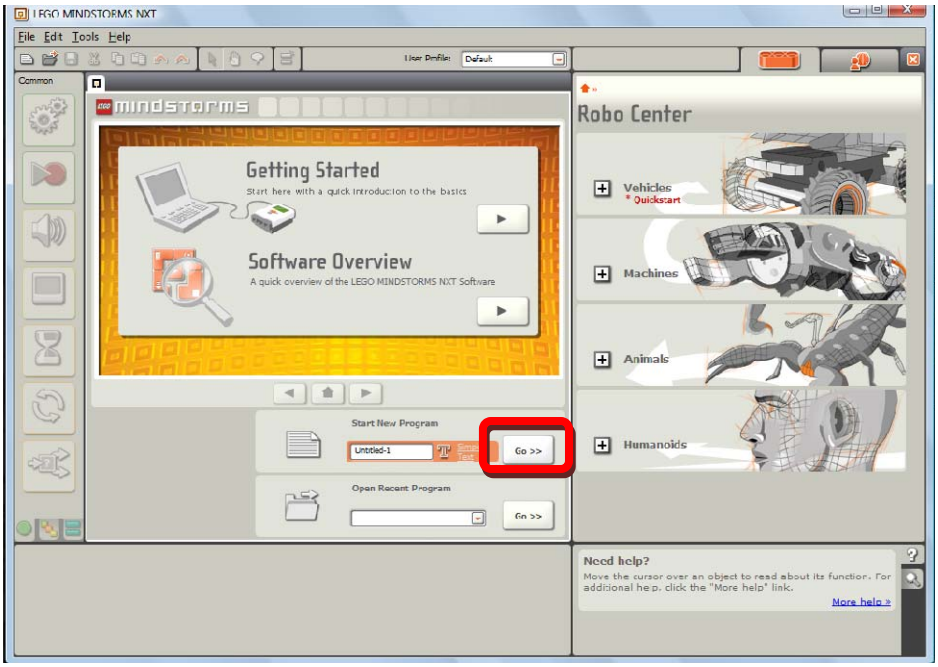
- Lego Mindstorms NXT Tribot
- Wii remote controller
- Laptop or computer with D-Link Bluetooth DBT-120 dongle

Part A: Connect NXT to the Laptop via Bluetooth

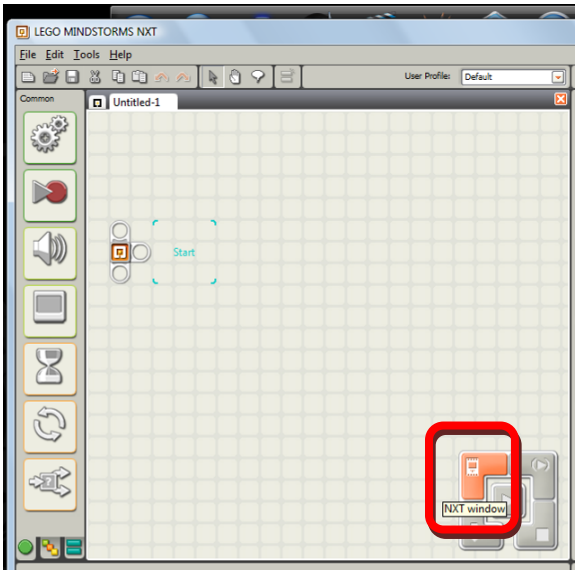
1. Plug in D-Link DBT120 Bluetooth dongle into one of the computer's USB port.
2. Double Click on the Mindstorms NXT icon on your computer's desktop to open the Mindstorms NXT software.



3. Go to **File >> New** to create a new project or just click on **Go>>** button on the screen.

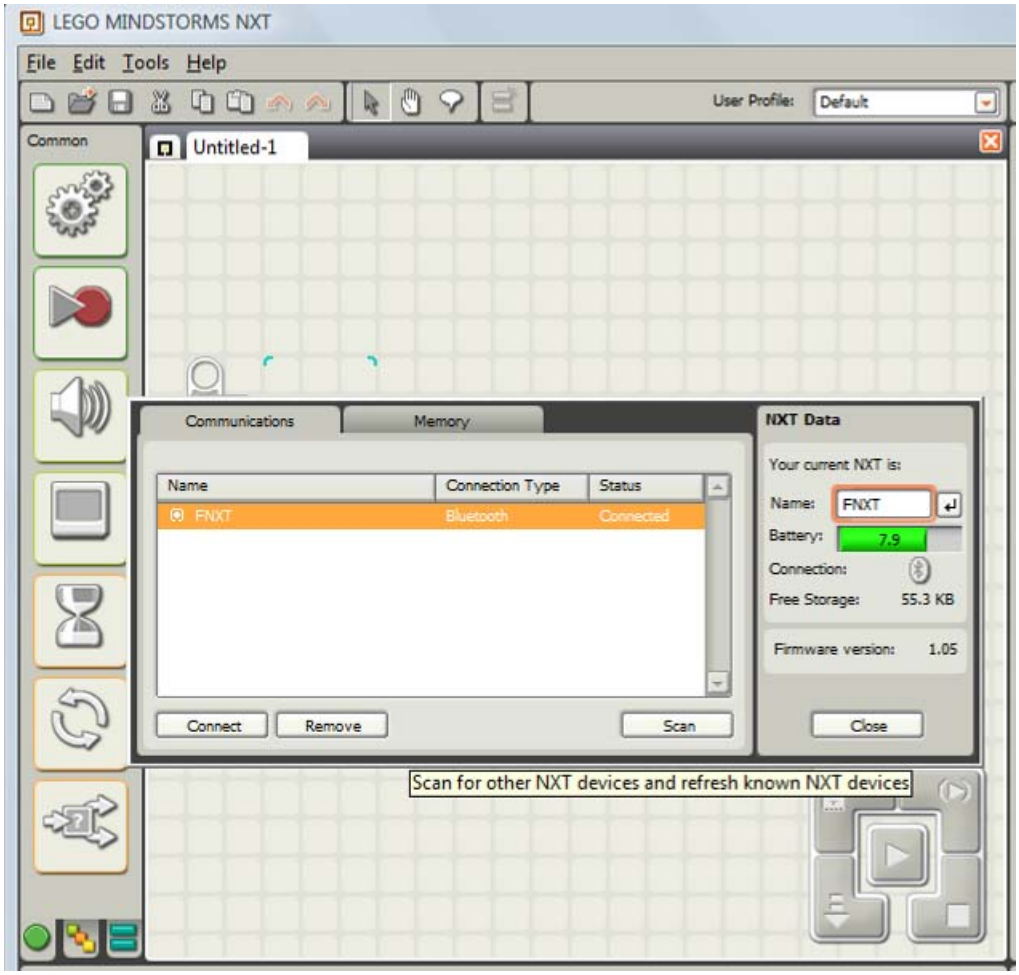


4. Press the orange, square **Enter** button on the Mindstorms NXT brick to turn the robot on.
5. Make sure you have Bluetooth icon and "**<"** shown on the upper left corner of the NXT screen. If not , try to use the arrow button on the NXT brick to choose **Bluetooth** icon and press the orange **Enter** button and choose **ON/Off** icon to turn on Bluetooth function of NXT and also make use NXT brick is visible to other Bluetooth device by choosing **Visibility icon>>Visible**.
6. Click on the **NXT Window** button on the lower right of the screen as shown here:



7. There will be a connection dialog window prompt out. If you have everything setup correctly, you can see the detailed information about your NXT brick including battery life, free storage space on the NXT block, and the version of the firmware installed, which indicates that the connection is established. If you can't have the NXT connected now, click on the **Scan** button to detect the NXT brick that is visible to the PC. The name of

your NXT brick should appear on the list of the dialog window. Click on your robot's name, and then click **Connect.** You may have the query prompt out on your NXT brick windows to ask you to key in passkey, just press Orange button to accept the default value (1234). Once you establish the connection, you can see the icon on upper left corner of your NXT brick windows changed from "**<**" to "**<>**" to indicate you the connection is established.



8. Click on the **Memory** tab at the top of the dialog window to see what programs are currently loaded onto the NXT block.
9. Close the dialog window, and now you can try to program the code in NXT environment or close the whole NXT programming environment and go to the section B to connect Wiimote to your PC. Notice that once you close the Mindstorms NXT program, the icon on your NXT brick windows will change back to "**<**".

Questions:

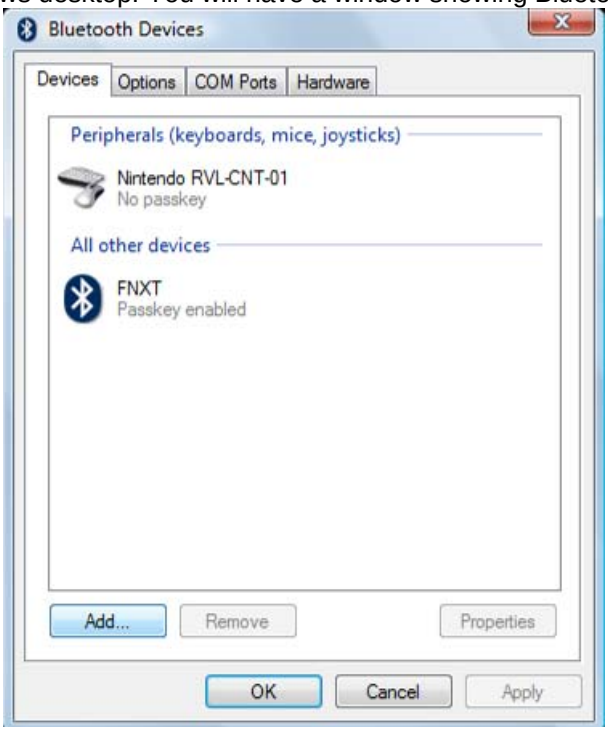
a) What should you do if you want to change the name of your robot?

b) What should you do if you want to download the program to your robot but the memory is full already?

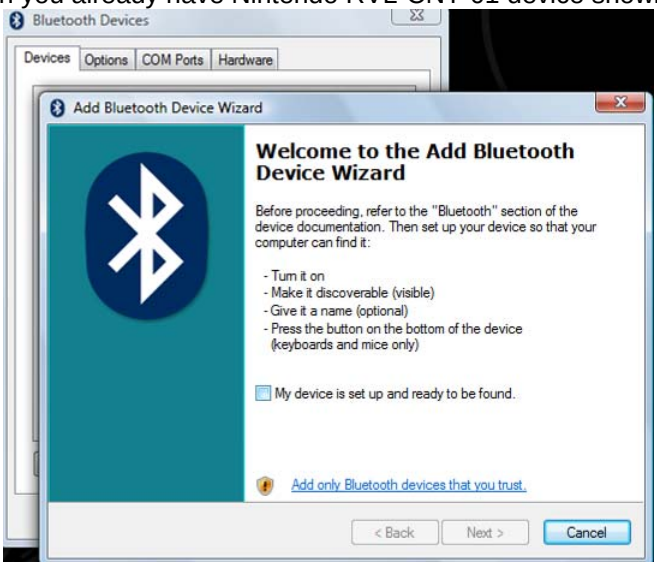
Part B: Connect Wiimote to the Computer

Before this part of exercise, make sure you have plugged in the Bluetooth dongle and finished Part A exercise. Now check that the Wiimote has batteries in it. And follow the procedure.

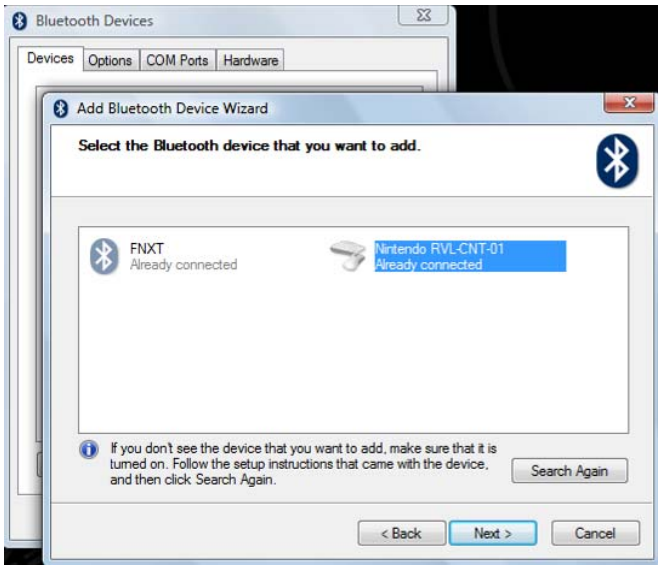
1. Double Click on the Bluetooth icon on the notification area of windows task bar, i.e. lower left side of windows desktop. You will have a window showing Bluetooth Devices.



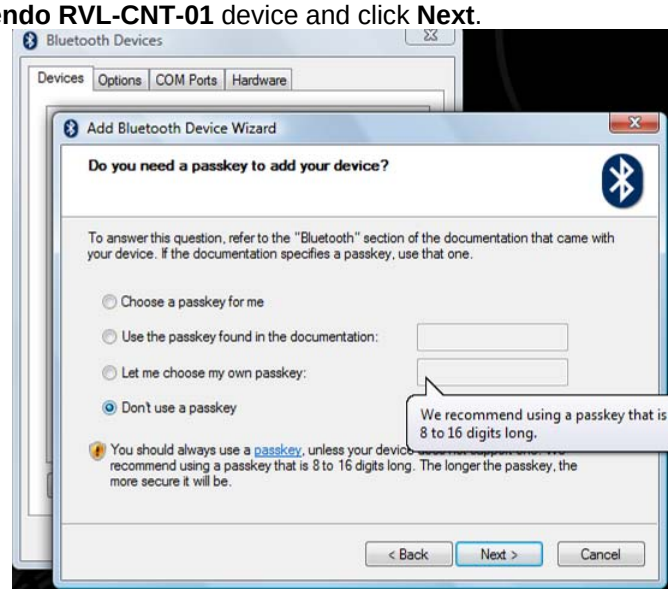
2. Click **Add**, even you already have Nintendo RVL-CNT-01 device shown on your list.



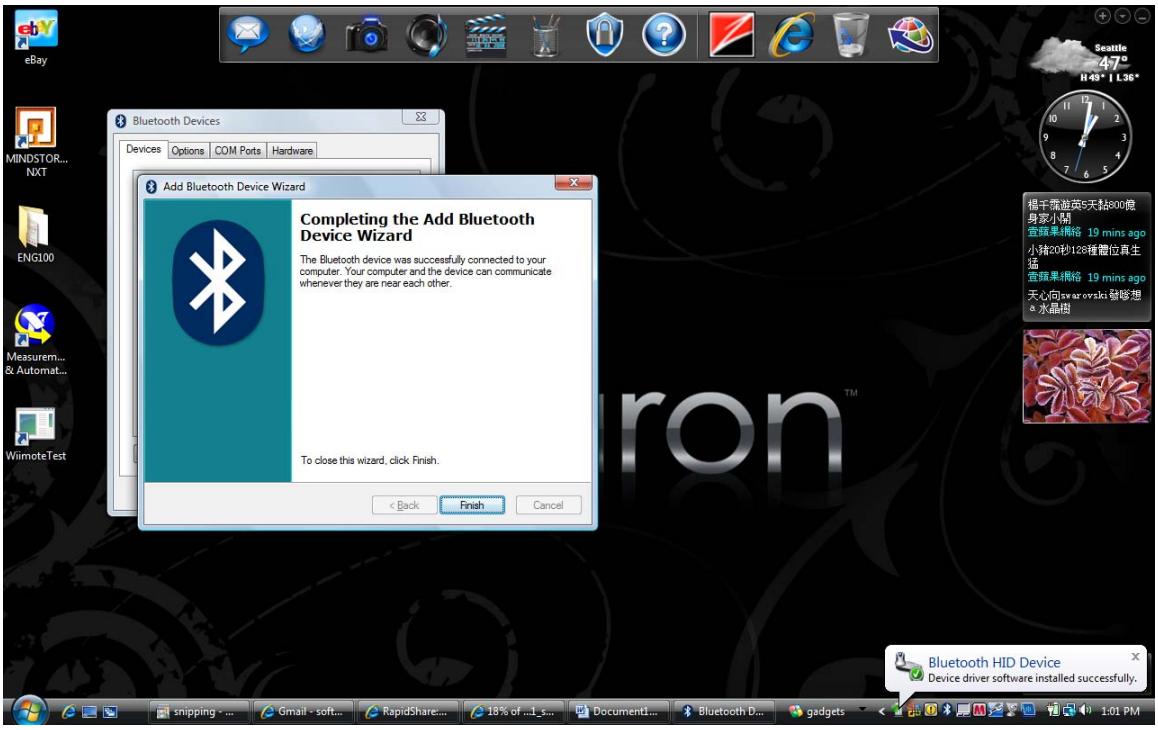
3. Press and hold **1** and **2** buttons of Wiimote at the same time for a while until you see the LEDs flash on and off. Check the **check box** of dialog and click **Next**.
4. Wait for a while for window to search Bluetooth devices. And you will see the following window if you have Wiimote around you.



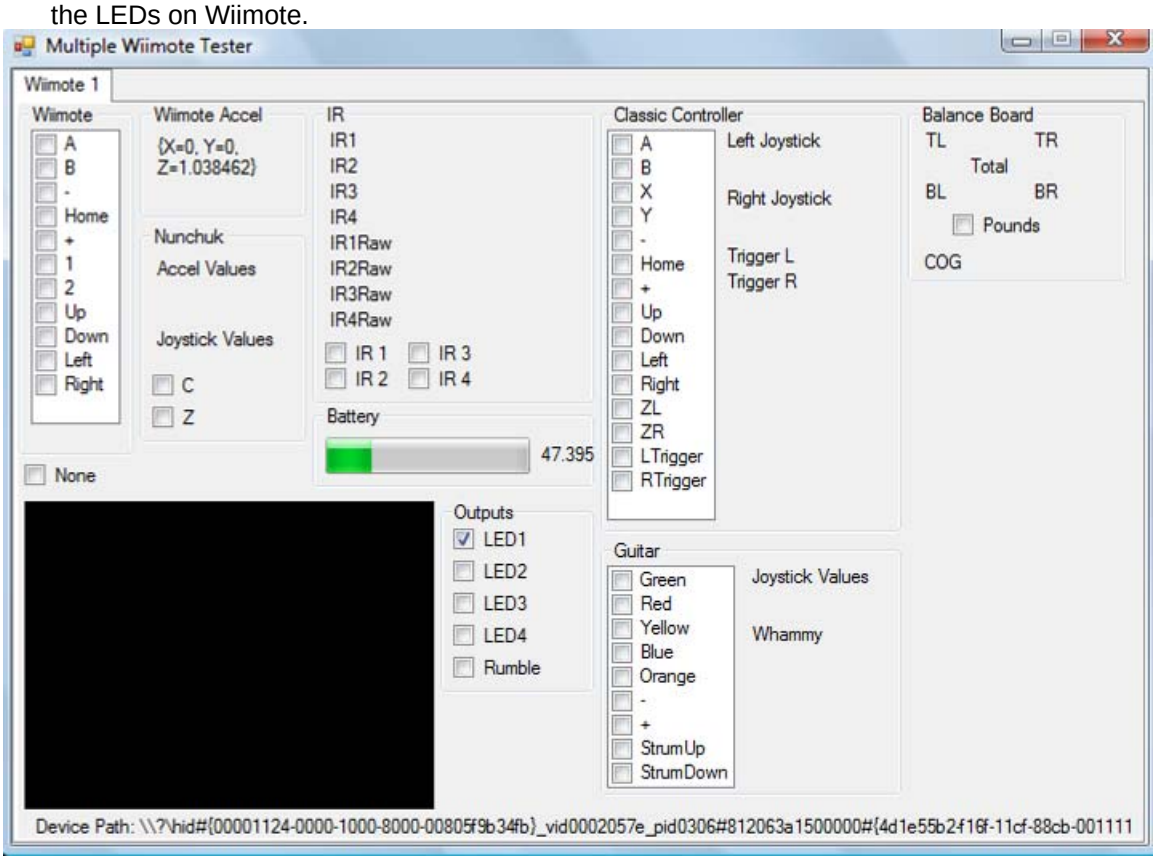
5. Choose **Nintendo RVL-CNT-01** device and click **Next**.



6. Select **Don't use a passkey** for this device. Before you move on, make sure the LED on your Wiimote are still flashing, or you may need to press and hold **1** and **2** buttons of your Wiimote again to make it flash before you click **Next**.
7. Now click **Next**, you will see that the window will try to connect and install the driver for your Wiimote.



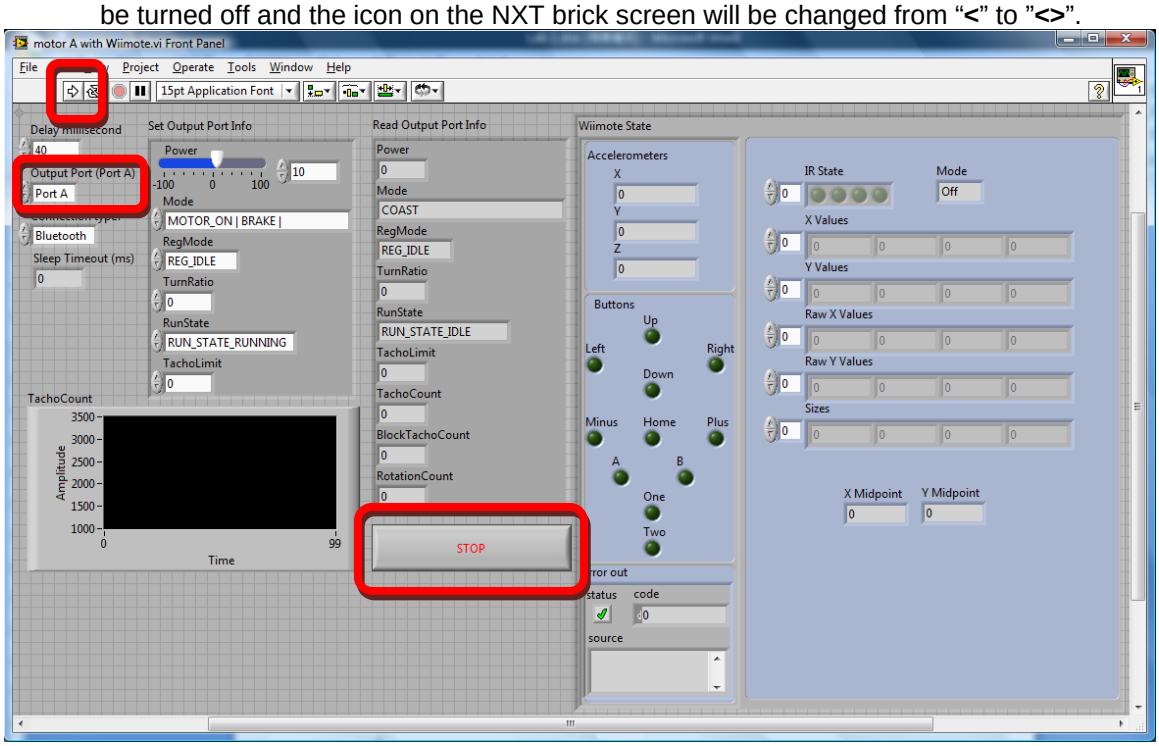
8. Now you have Wiimote connected with your PC. Double Click on **Wiimote Test** icon on your Desktop to check if you can actually read the sensor data from Wiimote and control the LEDs on Wiimote.



Part C: Run LabVIEW Sample Program

For this part of exercise, you will see that we can achieve much more functionality than LEGO Mindstorms NXT program. For example, we can use LabVIEW to accomplish the communication between different kinds of Bluetooth devices and won't be constrained in communication between NXT bricks and PC. Before starting this part of exercise, make sure you already finish the exercise of Part A and B correctly. Follow the procedure to run the sample program. In this demo, you will be able to use Wiimote to control the rotation speed and direction of motor A of LEGO NXT brick by flipping Wiimote forward and backward.

1. Double click on the file, **motor A with Wiimote.vi**, located on the **Desktop>>ENG100>> My LabVIEW example>>run in pc**, to invoke LabVIEW programming environment and prepare to run the demo program.
2. Click on the **Run** button  on the LabVIEW Toolbar to run the demo program. Notice that the system will take a few seconds to connect Wiimote and NXT brick to your PC. Once the connection is established, you will see that the lights of LEDs on Wiimote will be turned off and the icon on the NXT brick screen will be changed from "**<**" to "**<>**".



3. On the left half part of LabVIEW window is the control and information of NXT brick. You can try to control different motor of NXT brick by changing Output Port from Port A to Port B or C. On the right half part of windows is the information about Wiimote Status. You can see the readings of different sensors and tell which button is pressed.
4. Press button **A** on your Wiimote to stop the demo program, or use mouse to click on **STOP** button located on the middle of demo program window.

Questions:

a) What should you do if you have error alert message shown up?

b) How to use Wiimote to control motor B and C at the same time?

LEGO® MINDSTORMS® NXT Lab 2

In this lab session, you will learn the basic knowledge of LabVIEW programming environment and practice the NXT programming using LabVIEW NXT toolkit.

Lab Sections

A. Get familiar with LabVIEW environment
B. Write your code to get readings of sensors on NXT brick

Software

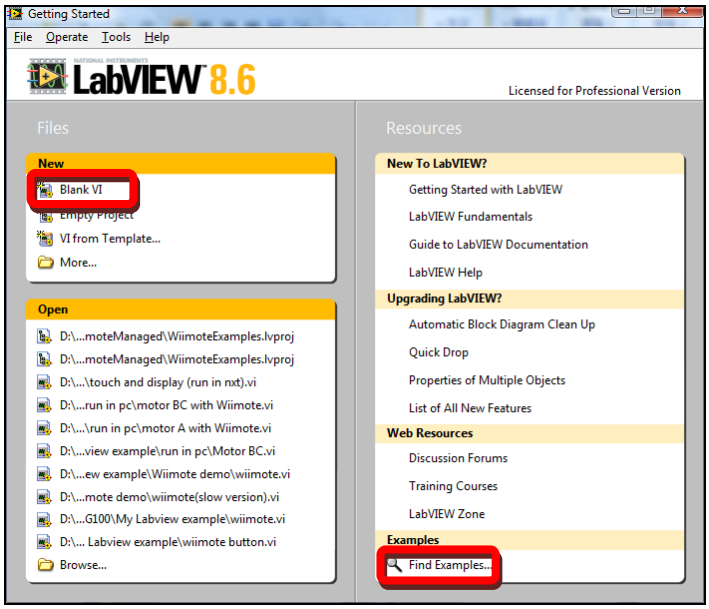
- Lego Mindstorms NXT
- NI LabVIEW with Lego NXT toolkit

Hardware

- Lego Mindstorms NXT Tribot
- Laptop or computer with D-Link Bluetooth DBT-120 dongle

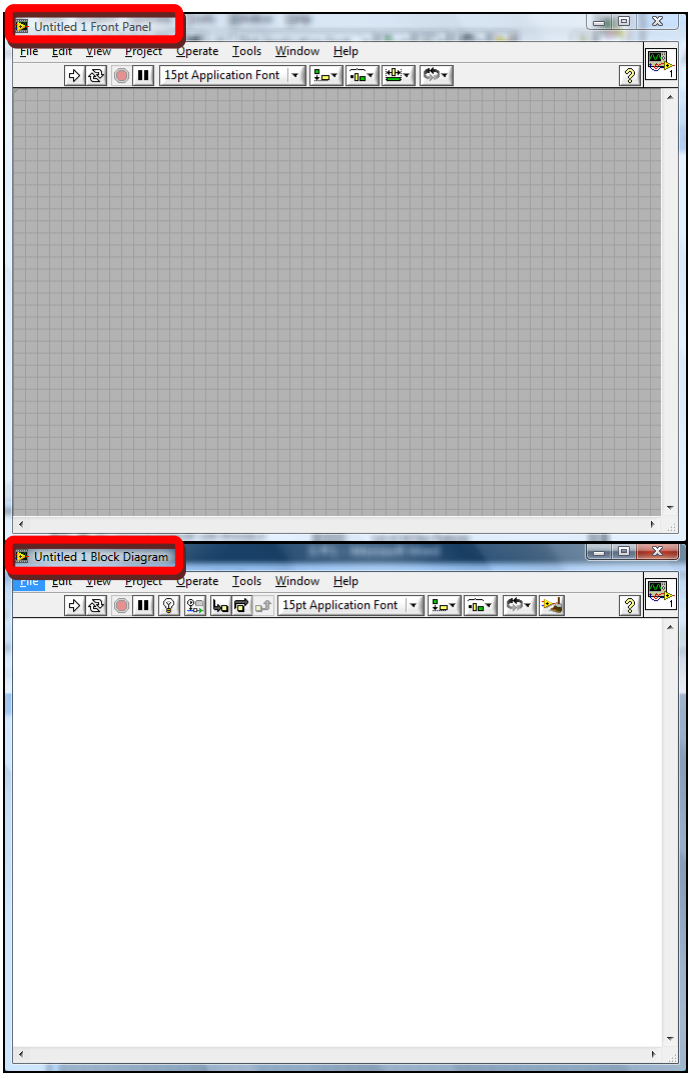
Part A: Get Familiar with LabVIEW Environment

1. Open and run LabVIEW 8.6 by clicking on **Windows Start** icon >> **All Programs** >> **National Instruments LabVIEW 8.6.** Then you can have LabVIEW 8.6 Startup Screen as below.

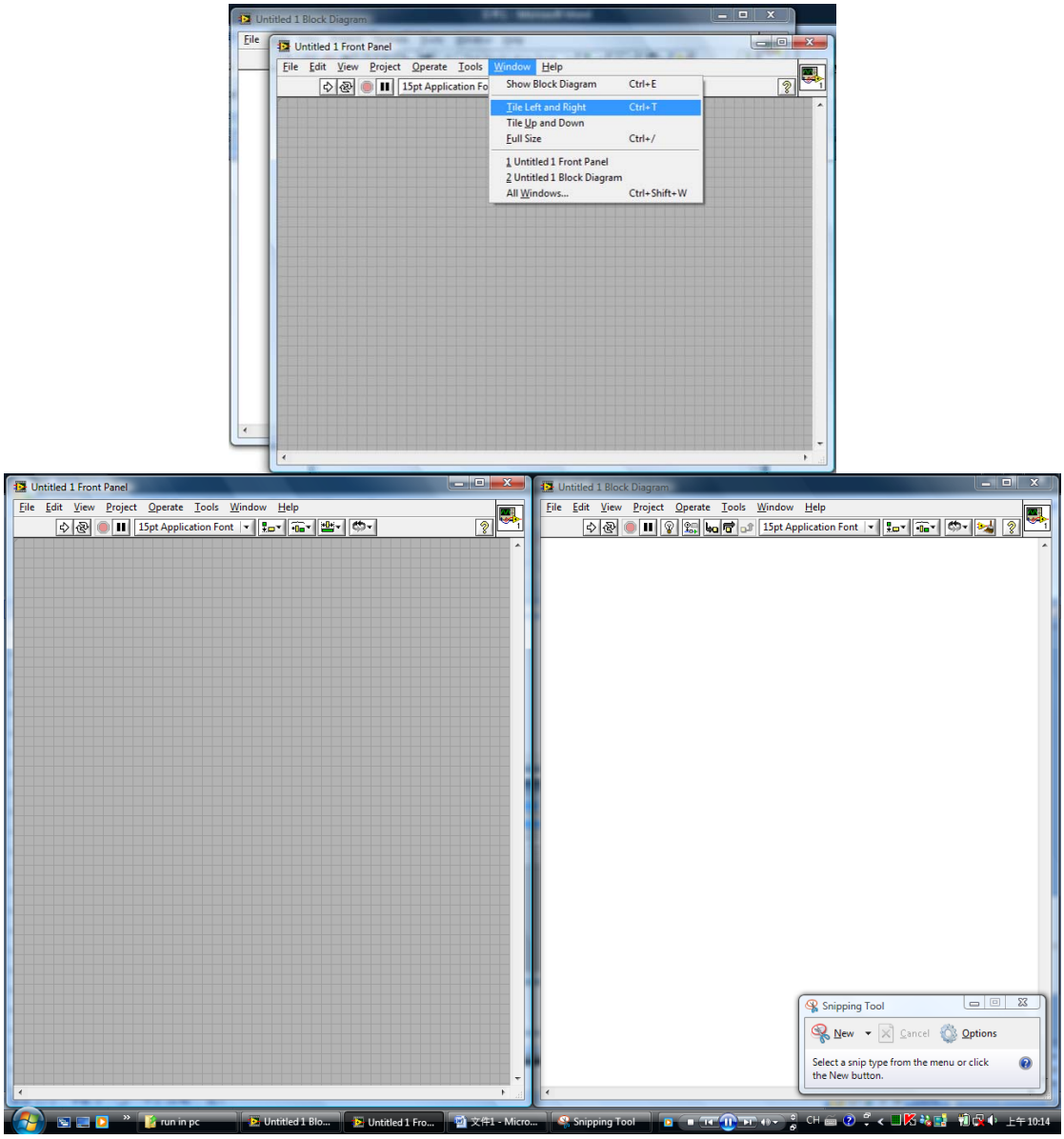


2. You can start from a blank VI by clicking on **New**>> **Blank VI** to program your code from draft. Or you may start from a template/example by clicking on **Examples**>> **Find Examples…**to modify the existing code for your particular application. To begin with, let's start from a blank VI. You will have the following two windows for your new VI. The Front Panel window is the interface for users to interact with your program. You can design the outlook of your program here and the two main objects for front panels are controls and

indicators. You can think of controls as inputs and indicators as outputs. The Block Diagram window is the place for you to write graphical codes that process the user inputs and the logic of your functions.

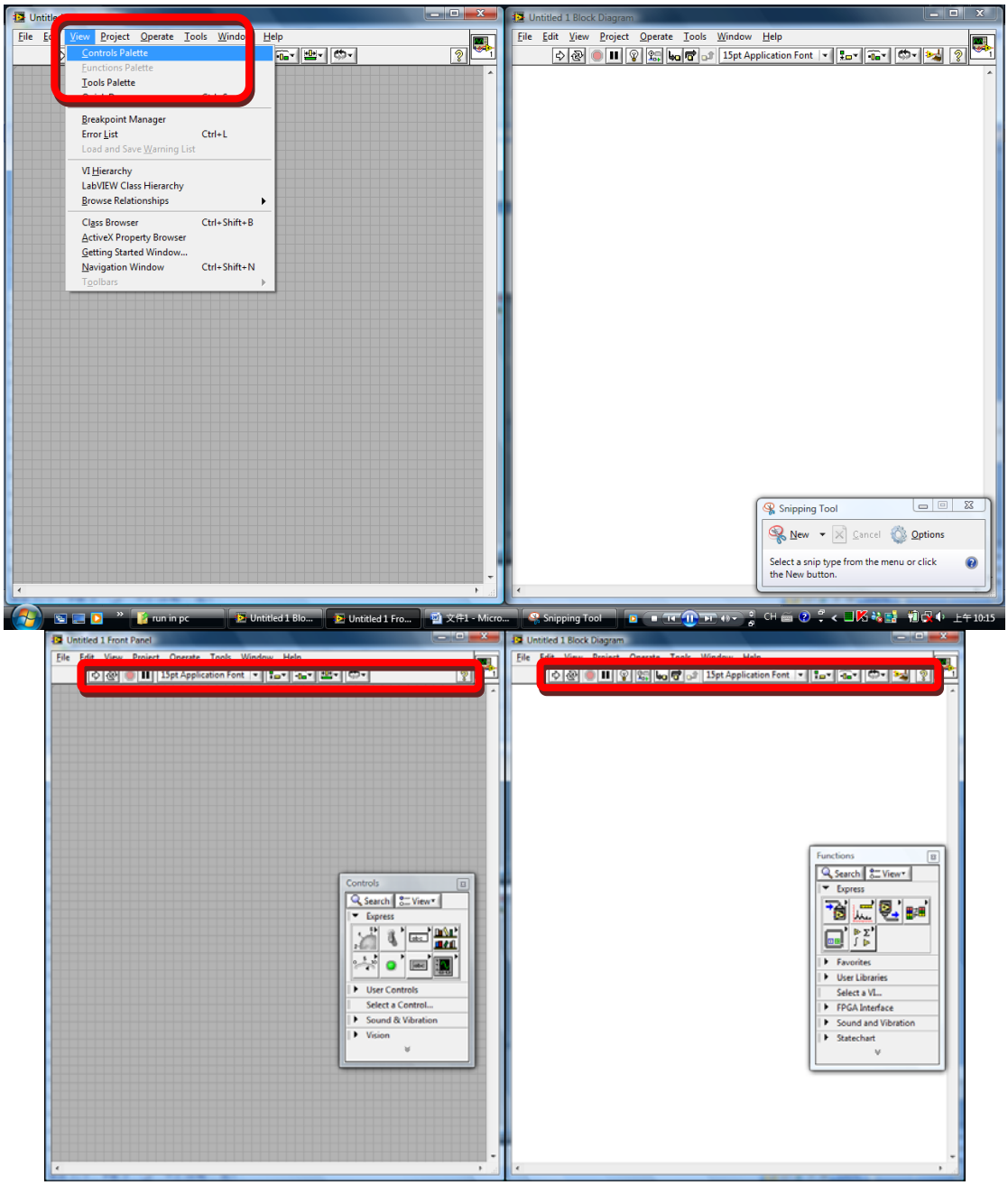


3. You may rearrange the places of windows by choosing **Windows**>>**Tile Left and Right** to have them shown on the screen at the same time.

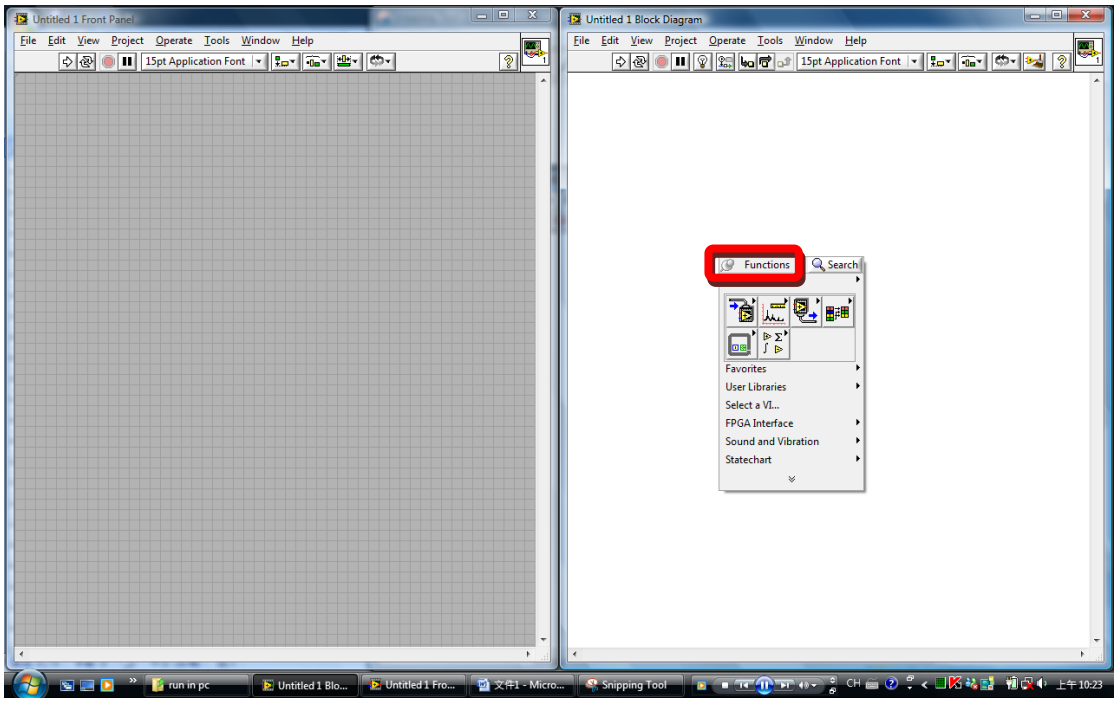


4. To view and see what **Controls** or **Indicators** you can put on the front panel, select **VIEW**>> **Controls Palette**. You can customize the palette view by clicking on **View** icon on the controls palette. To put a control or indicator on the front panel, you just click on the icon on the palette you want first, and put it on the front panel.

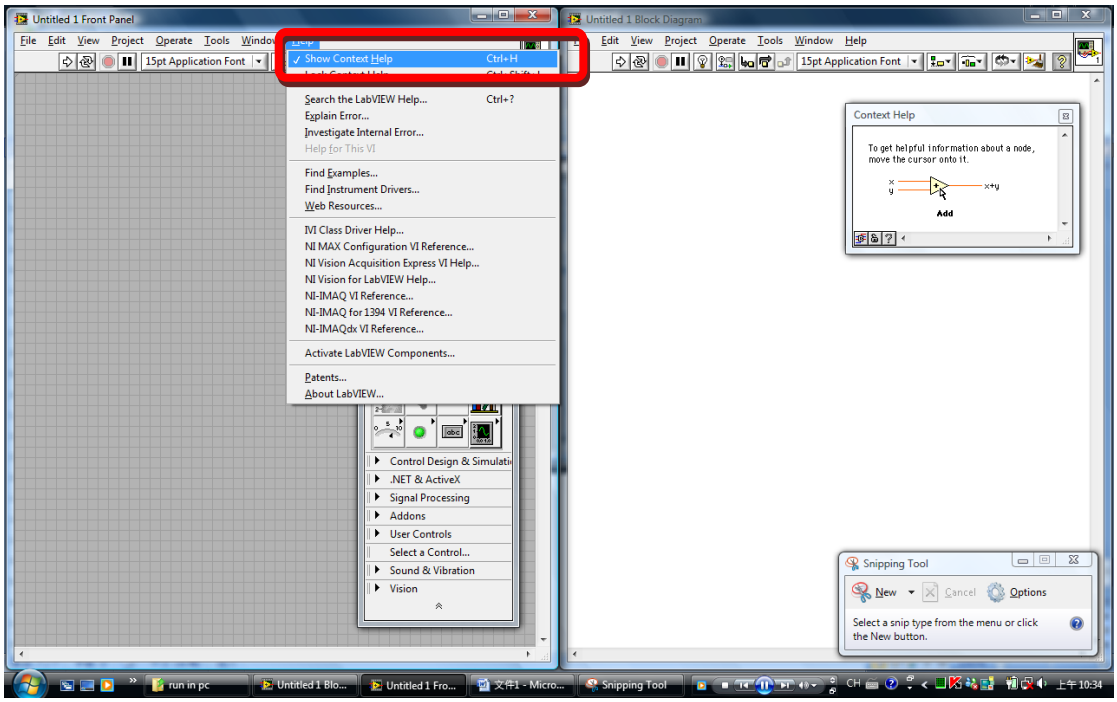
Tips: move your mouse to any empty space on the front panel and right-click your mouse to invoke display of controls palette. And tack down controls palette by clicking the pushpin icon on the top left corner of the palette.



5. To view and see what **Functions** or **Structures** you can use in your graphical code on the block diagram window, select **VIEW**>> **Functions Palette**. Or use the same trick as before, display the functions palette by right-clicking on the empty space of your block diagram window.



6. Once you finish your coding, click the **Run** button on the Status Toolbar. There are other buttons on the toolbar, such as continuous run button, abort button, pause button, etc. Don't forget to get help online by choosing **Help**>>**Show Context Help** or pressing **Ctrl + H** to invoke help window. Please refer to the reference material for more details.

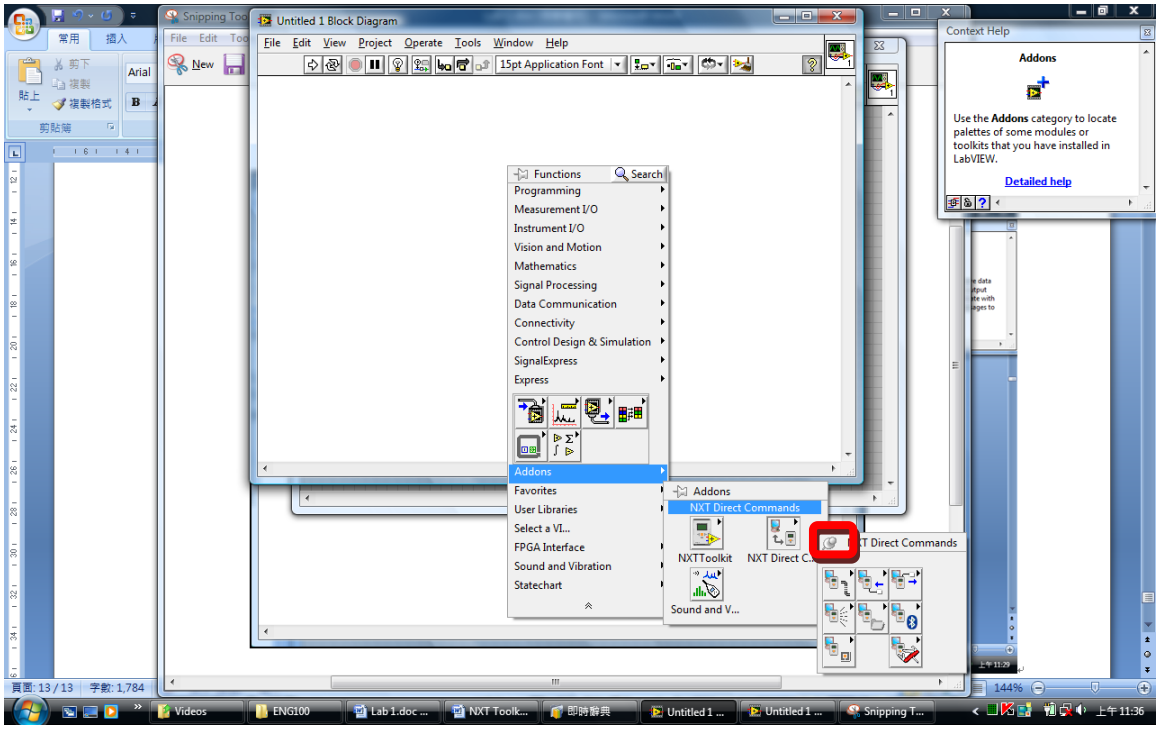


HW: Read Introduction to LabVIEW in 3 hours material.

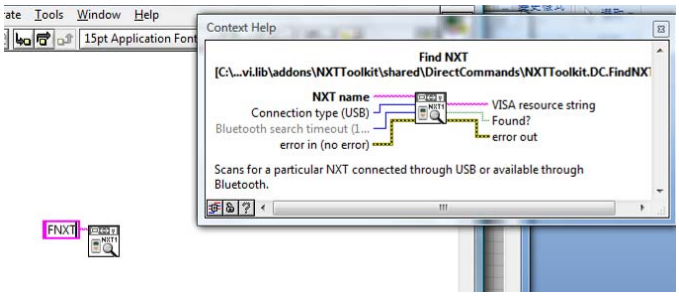
Part B: Write VI Code to Get Readings of NXT Sensors

For this part of exercise, you will learn how to write the codes to read the sensors' values on NXT brick. There are two ways to connect the NXT brick with your PC. One is connected via USB cable, while the other one is connected via Bluetooth. So you can write a code for users to decide which way they would like to use. If you want to use Bluetooth communication, please follow the procedures mentioned in Lab 1 Part A to check if you can connect successfully.

1. Open LabVIEW 8.6 by clicking on **Windows Start** icon **>> All Programs >> National Instruments LabVIEW 8.6.** Then start a blank VI by clicking on **New>> Blank VI**.
2. For the convenience of coding, let's tack down the NXT Toolkit function palette on the block diagram window. Right click on any empty space on block diagram window, click on the » button at the bottom of the palette. Select **Addons>>NXT Direct Commands**. Tack down pushpin icon on the top left corner of the palette.



3. Select **NXT Direct Commands>>Connection>>Find NXT**, and place it on the block diagram.
4. Right-Click on the upper left input corner of the **Find NXT** VI icon, i.e., the **NXT name** input port. You may press **Ctrl + H** to invoke help windows to help you find out the corresponding input port easier. Move your cursor to a certain I/O port of the icon, and you will see the corresponding flashing indicator on the help window. Select **Create >>Constant** from the shortcut menu and key in the name of your NXT brick. In this demo, the name of NXT brick is **FNXT**. So after you complete this step, the block diagram window would look like the figure below.

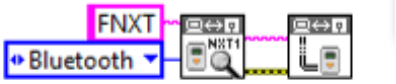


Note: Make sure the capitalization of your NXT brick's name is correct.

5. In order to use Bluetooth communication, we need to have a parameter in the program to tell which connection type we are going to use. Right-Click the Connection type (USB) input port of the Find NXT VI icon, i.e., the second input port on the left side of the icon, and select **Create>>Constant** from the shortcut menu. Change the connection type from USB to Bluetooth by clicking the USB constant and select Bluetooth from the shortcut menu.

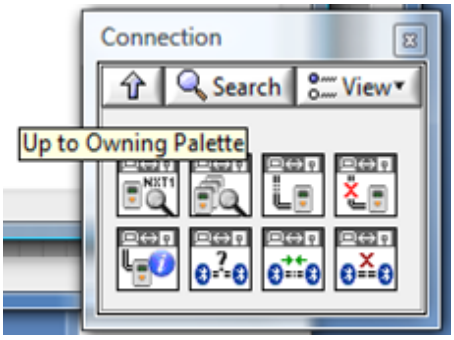
Question: What should you do if you want users to be able to choose the connection type by themselves? (Hint: get something on the interaction window)

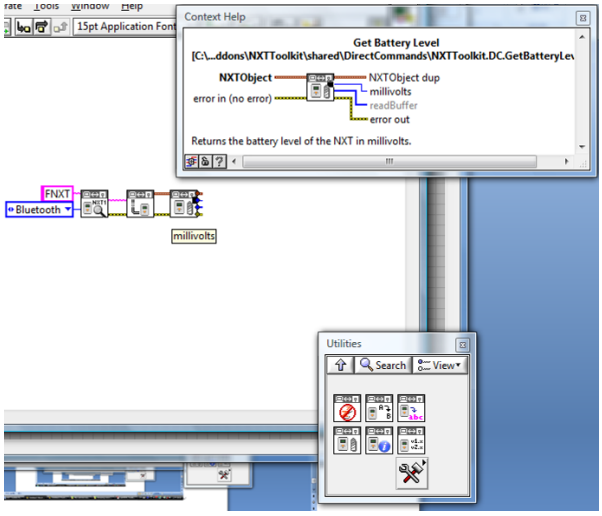
6. From the Function Palette, select **NXT Direct Commands>>Connection>>Create NXTObject**, and place it next to the **Find NXT** VI icon. Wire the **VISA resource string** from the output put of the **Find NXT** icon to the **VISA resource string** input port of the **Create NXTObject** icon. Wire the error information of two icons, from **error out** output of the **Find NXT** icon to the **error in** input of the **Create NXTObject** icon.



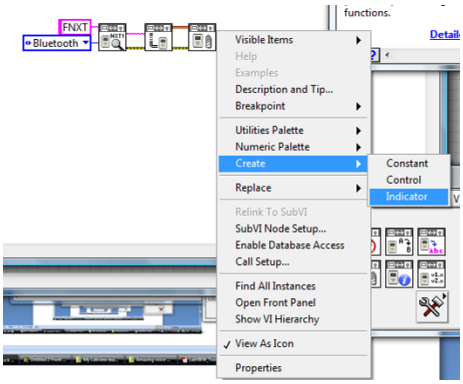
Tip: When you select a new VI such as **Create NXTObject** in this case and place it next to the relating VI icon, try to move it closer to the relating VI icon and you will find that LabVIEW will connect the corresponding ports for you.

7. Click Up to Owning Palette icon on the function palette, and select **NXT Direct Commands>>Utilities>>Get Battery Level**, and place it next to the **Create NXTObject** VI icon. Wire the **NXT Object** information and **error** information of these two VI icons, or use the tip of step 6 to wire these two VI icons.





8. To get the battery level value shown on the front panel, right-click on the second output port of Get Battery Level VI icon, i.e., millivolts output port, select **Create>>Indicator** from the shortcut menu.

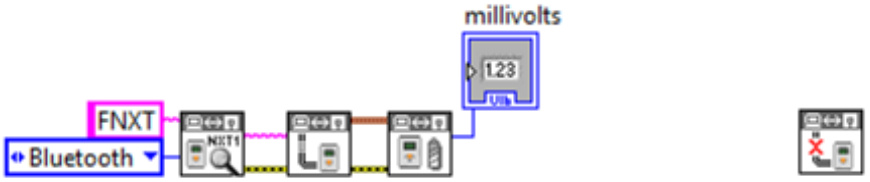


9. The last step is to destroy the NXT object that you create in your previous step. Select **NXT Direct Commands>>Utilities>>Destroy NXTObject**, and place it next to **Get Battery Level** VI icon, and wire the **NXT Object** information and **error** information of these two VI icons, or use the tip of step 6 to wire these two VI icons.

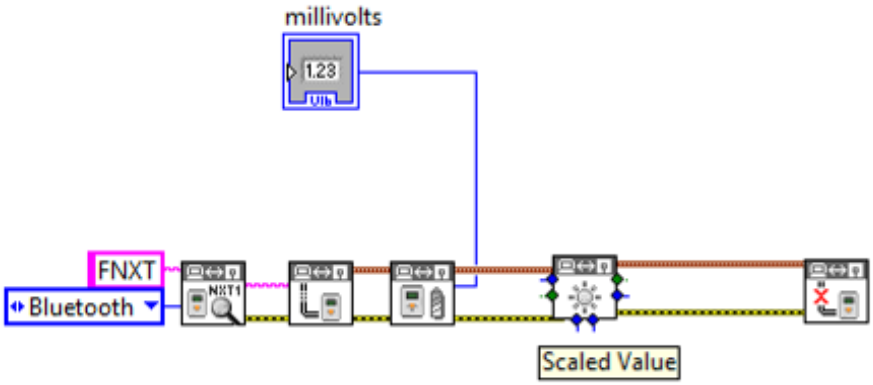
10. Now you finish the coding to read battery level of your NXT brick. Press **Ctrl + E** to switch to the front panel window and click the **Run** button on the toolbar to run your first NXT program.

Exercise: Try to modify this code to get more NXT device information, firmware version, and rename NXT brick by using existing VIs in **NXT Direct Commands>>Utilities.**

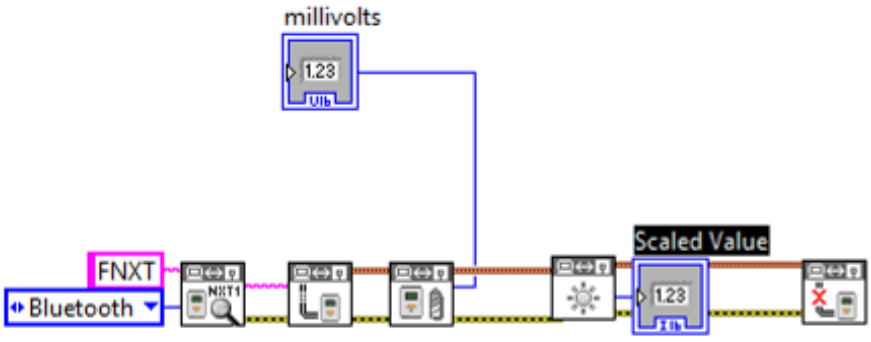
11. There are four different sensors that you can connect to your NXT brick. Make sure you connect with the default setting. Now let's modify the code we wrote to get the reading of the light sensor. First delete the wires between **Destroy NXTObject** and **Get Battery Level** VI icons.



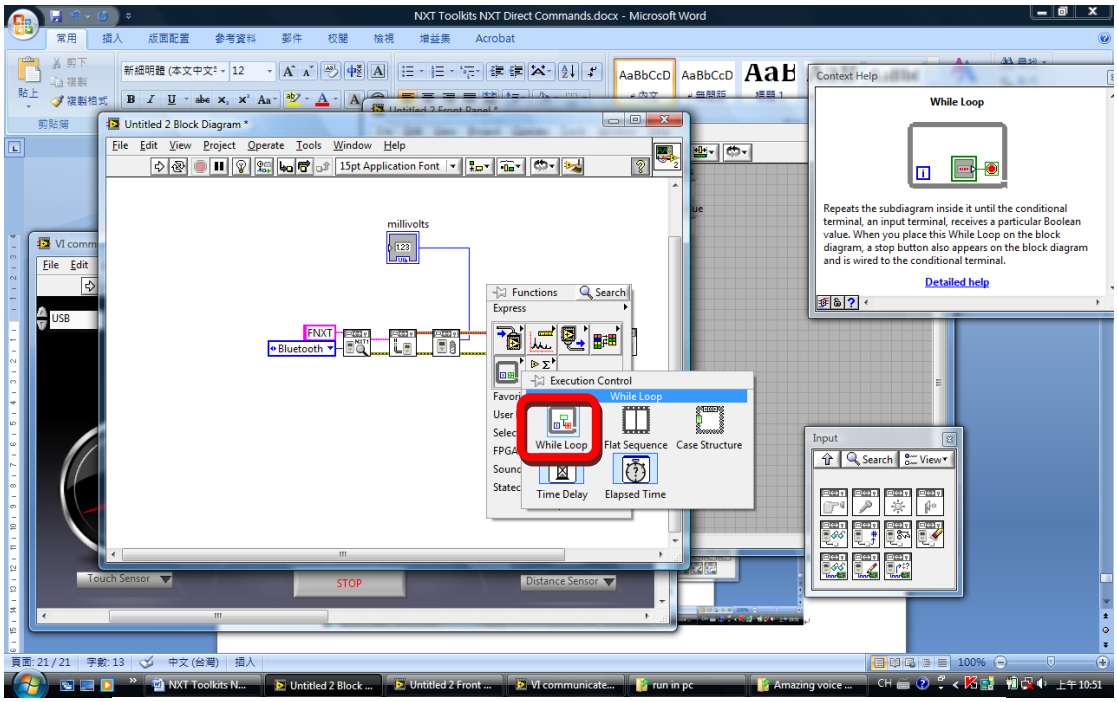
12. Select **NXT Direct Commands>>Input>>Read Light Sensor**, and place it next to **Get Battery Level** VI icon. Wire the **NXT Object** information and **error** information of these VI icons as following.

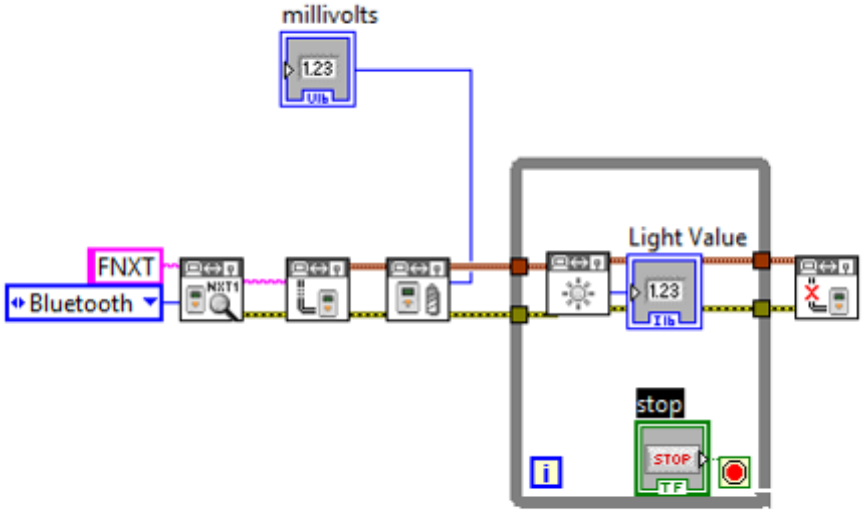


13. Right-Click the **scaled value** output port of the **Read Light Sensor** VI icon, and select **Create>>Indicator** from the shortcut menu. Now you have two indicators on the front panel. Run this program to get the battery level and light sensor readings once.

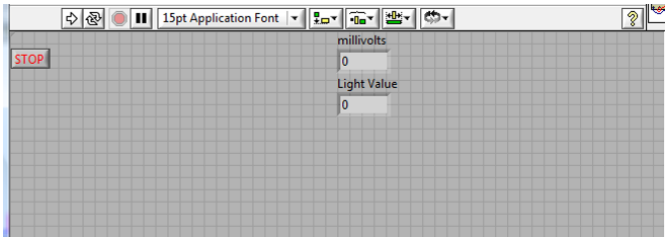


14. Normally, we would like to keep getting the reading until the user presses STOP button. To do this, we need to add a while loop structure in the program. Right-click on any empty space of the block diagram. From the short cut functions palette, select **Express>>Execution Control>>While Loop**, drag a square enclosing **Read Light Sensor** and **Scaled Value** icons





15. LabVIEW will automatically create a control button called STOP on the front panel. Users can press this button to stop the while loop and let the program goes forward to the end. Click the **Run** button on the toolbar and see the values of two indicators. Notice that the value of the battery level won't keep changing, while the value of the light sensor will change with time until the STOP button is pressed.



16. Save the code for future usage.

Exercise: Try to modify this code to get other sensors' readings by using existing VIs in **NXT Direct Commands>>Input**, such as **Read Touch Sensor**, **Read Sound Sensor**, and **Read Ultrasonic Sensor**. Remember to create the corresponding indicator for each sensor.

LEGO® MINDSTORMS® NXT Lab 3

In this lab session, you will learn how to write a code to read the motor status of your NXT robot and make the motors move using LabVIEW NXT toolkit.

Note: Please follow the procedures mentioned in Lab 1 Part A first to make sure that you can connect your NXT robot to PC via Bluetooth successfully.

Lab Sections

A. Read the motor status of your NXT robot
B. Write the code to control motors of your NXT robot

Software

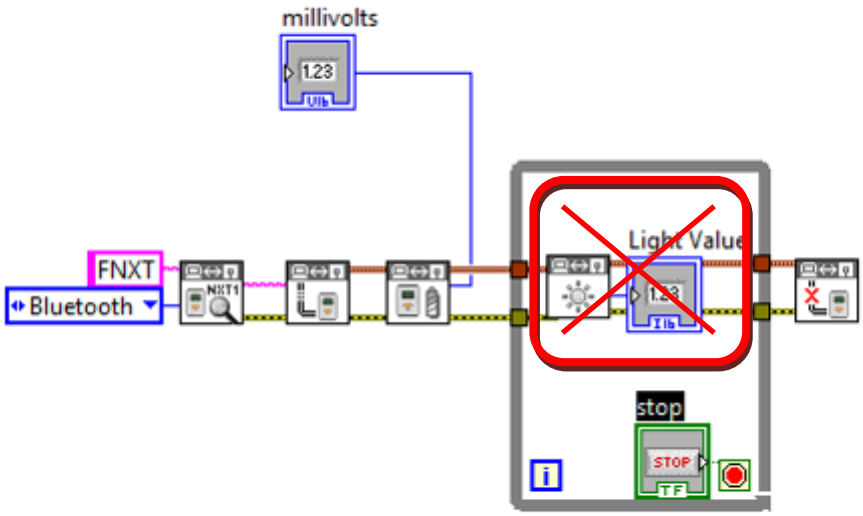
- Lego Mindstorms NXT
- NI LabVIEW with Lego NXT toolkit

Hardware

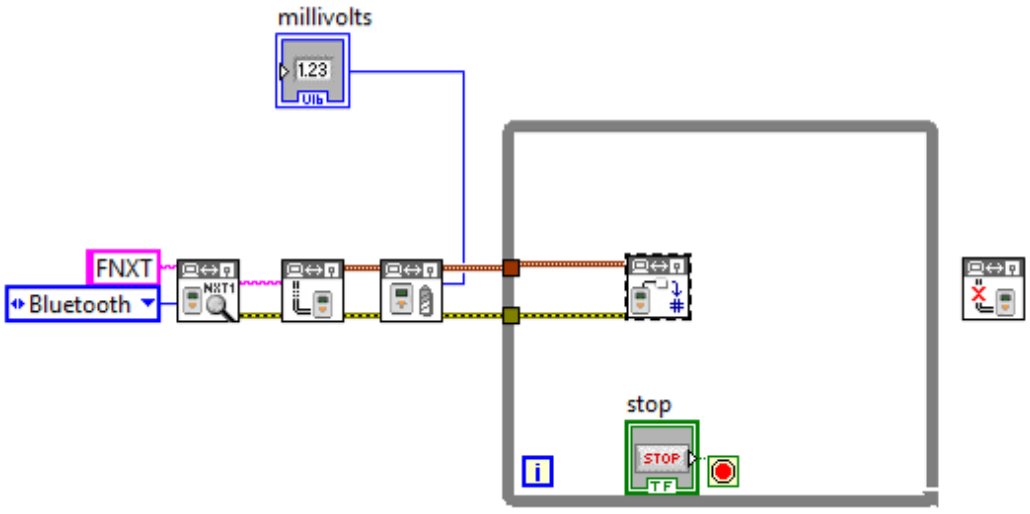
- Lego Mindstorms NXT Tribot
- Laptop or computer with D-Link Bluetooth DBT-120 dongle

Part A: Read the motor status of NXT robot

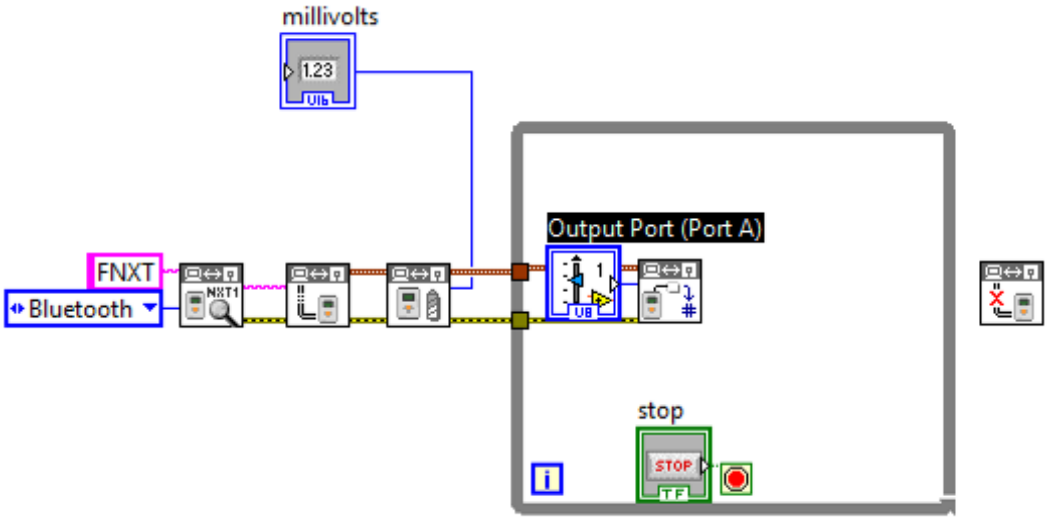
1. To save our time, let's not start from scratch but modify the code we have from previous lab. Open the file you create from Lab2 Part B. Delete the code inside the while loop structure except the STOP button.



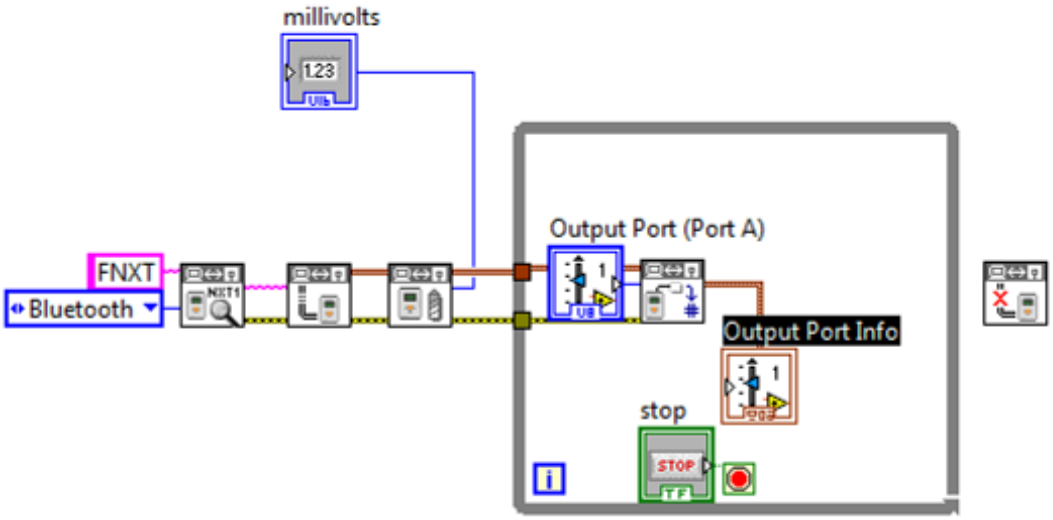
2. Select **NXT Direct Commands>> Output>>Get Output Values**, and place it inside the while loop structure. Wire the **NXT Object** information and **error** information of these VI icons as following.



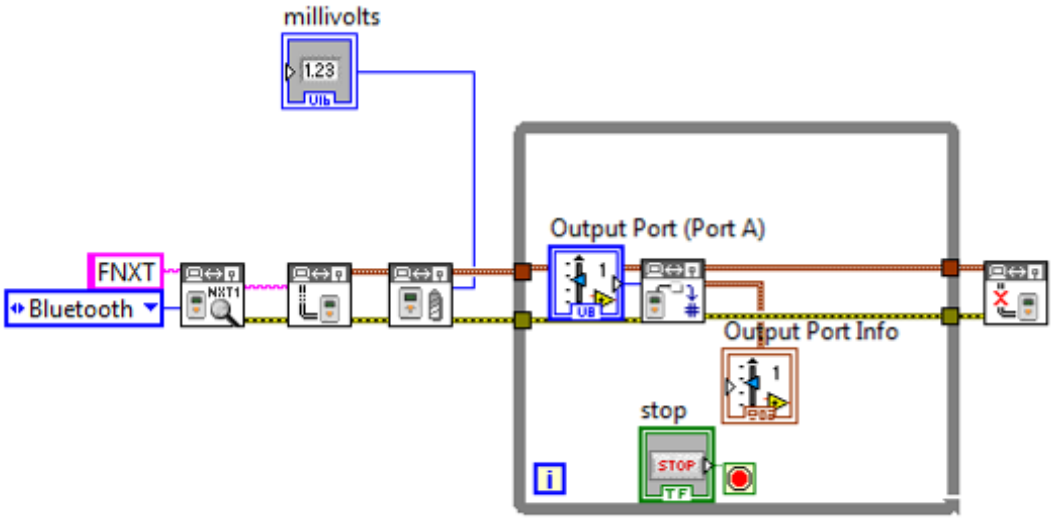
3. Right-Click the second input port of **Get Output Values** VI icon, i.e., **Output Port (Port A)**, and select **Create>>Control** from the short cut menu. With default setting, motor A is connected to output port A of NXT brick. Users can read different motor status by changing the value of this control.



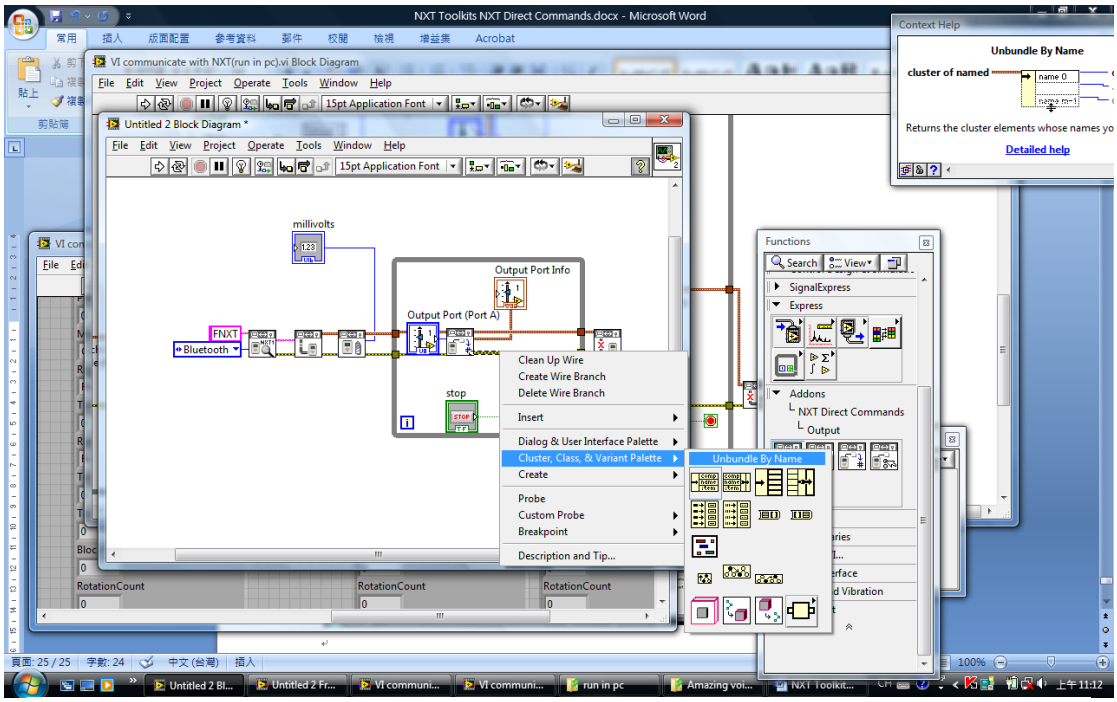
4. Right-Click the second output port of the **Get Output Values** VI icon, i.e., **Output Port Info**, and select **Create>>Indicator** from the shortcut menu.

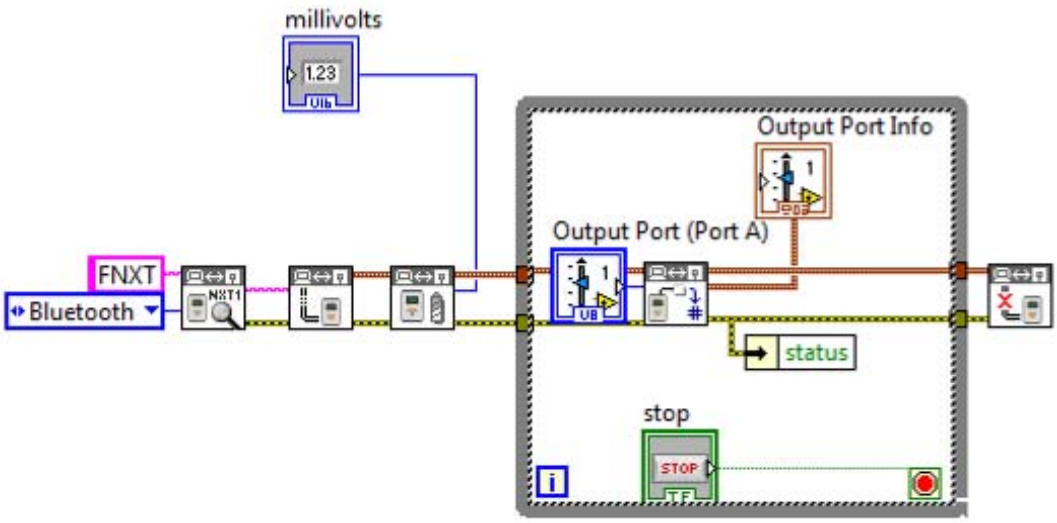


5. Wire the **NXT Object** information and **error** information of **Get Output Values** VI icon and **Destroy NXTObject** VI icon as following.

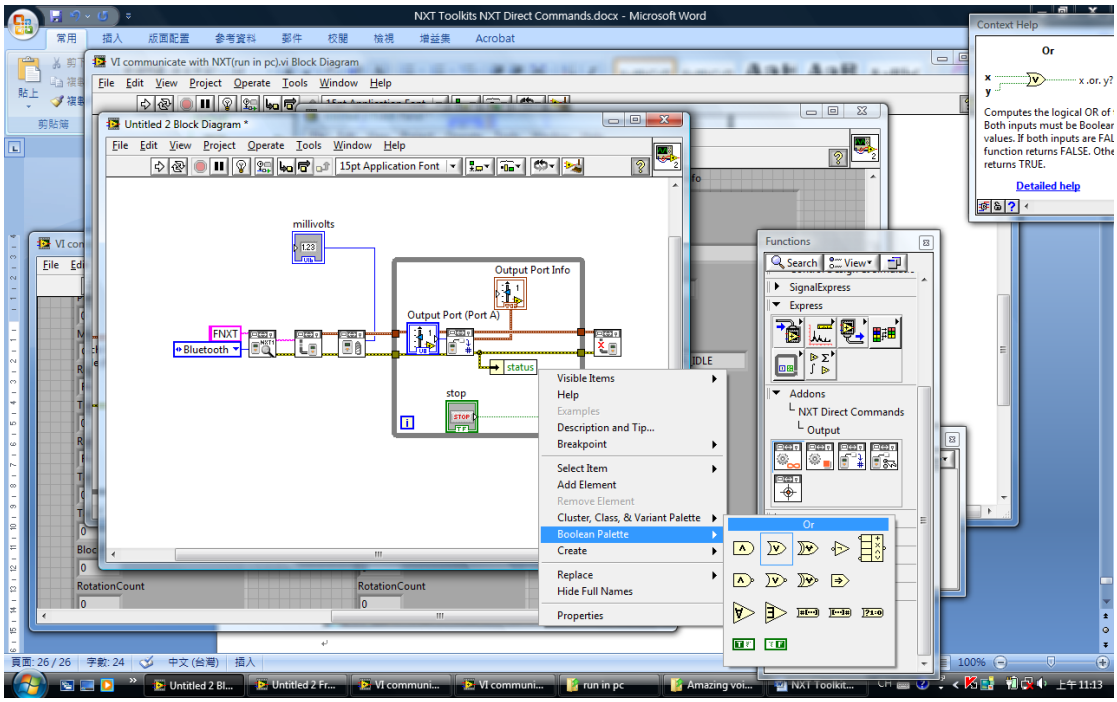


6. Now you can run the program and get one updating motor status. Try to rotate the motor by hands and observe the indicator values on the front panel.
7. To improve the code so that the program would stop the while looping if there is any error on NXT or communication during execution, right-click the error wire in the while loop structure. Select **Cluster, Class, & Variant Palette>>Unbundle By Name** from the short cut menu. Place it inside the while loop structure and connect the error wire to it.

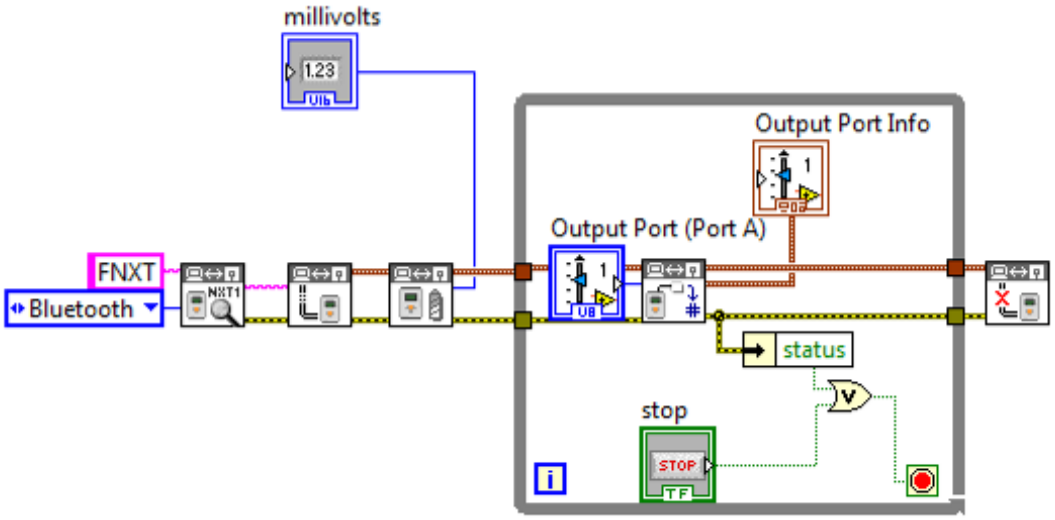




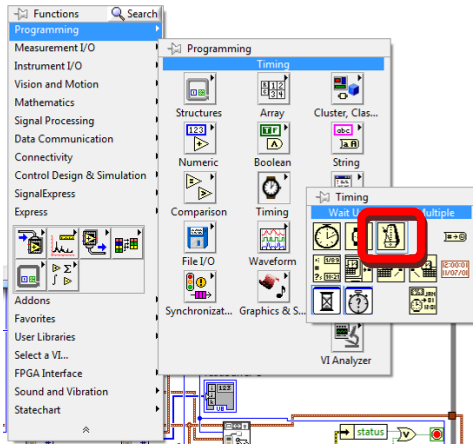
8. Right-Click the green dash wire inside the while loop structure. Select Boolean Palette>>Or from the short cut menu, and place it inside the while loop structure.



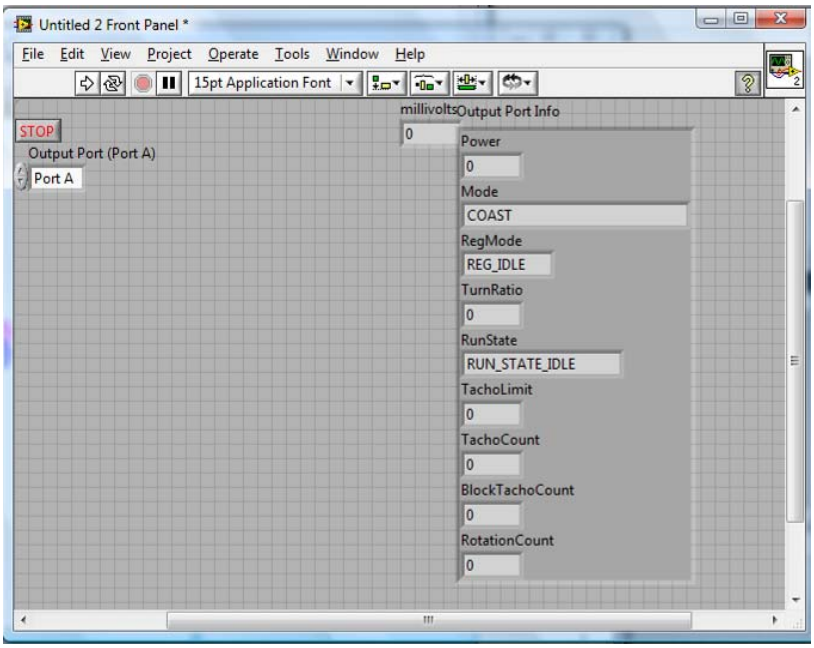
9. Delete the green dash wire and reconnect the wires as below.



10. With this setup, the program would stop running if there is any error occurs or if the stop button is pressed.
11. Add a 40 millisecond time delay in the while loop structure to prevent the code occupying the whole CPU resource. Right-click on any empty space on the block diagram windows, select **Programming>>Timing>>Wait Until Next ms Multiple** from the short cut menu, and place it inside the while loop structure. Right-click the input port of this time delay icon and select **Create>>Constant**. Key in the value **40** for indicating 40 ms time delay.



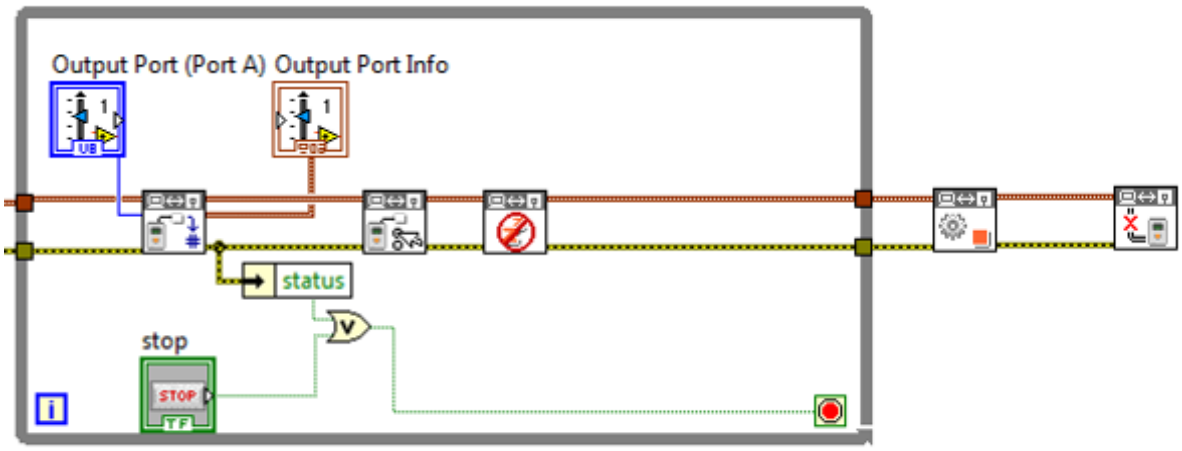
12. Now you finish the coding for reading the motor status. And the front panel should appear similar to the figure below.



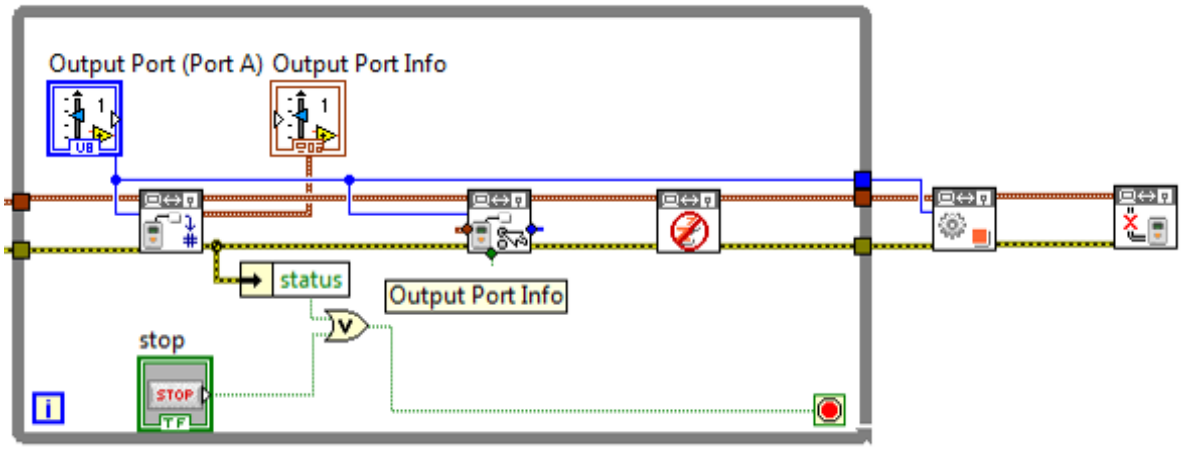
Exercise: Modify the code to enable reading ABC motor status in while loop sequentially.

Part B: Control the motors of NXT robot

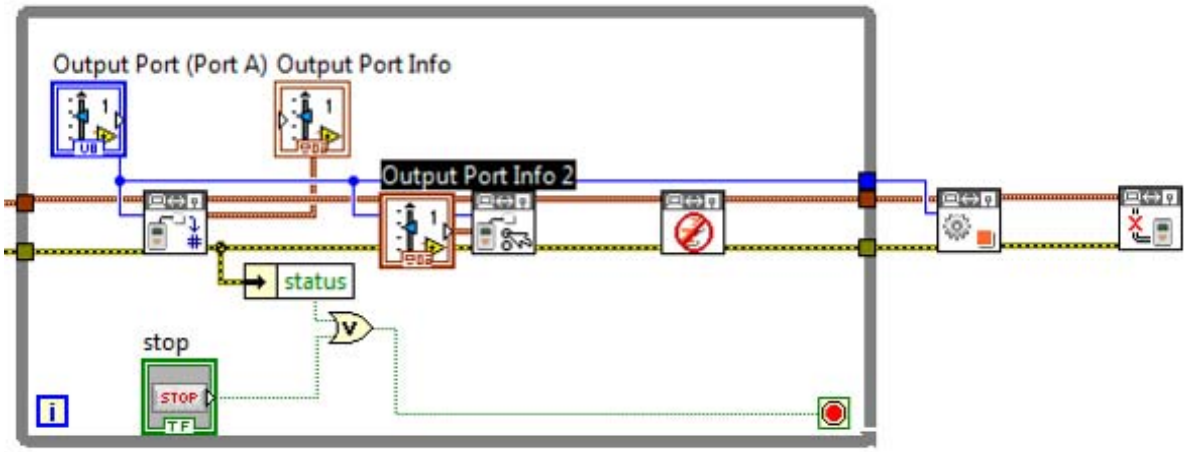
1. Open the file you create from Lab3 Part A. Delete the wires between **Get Output Values** VI icon and **Destroy NXTObject** VI icon. Select and place the following two VI icon inside the while loop structure from functions palette **NXT Direct Commands>>Output>>Set Output Values** and **NXT Direct Commands>>Utilities>>Keep Alive**. Select **NXT Direct Commands>>Output>>Motor Stop**, and place it outside the while loop. Connect these VI icons as following.



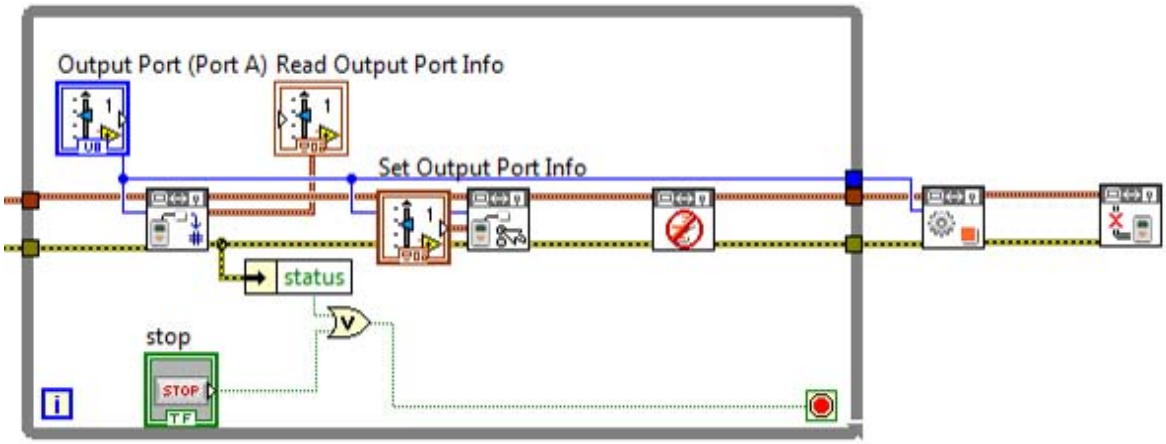
2. Wire the **Output Port (Port A)** to **Set Output Values** VI icon and **Motor Stop** VI icon as following.



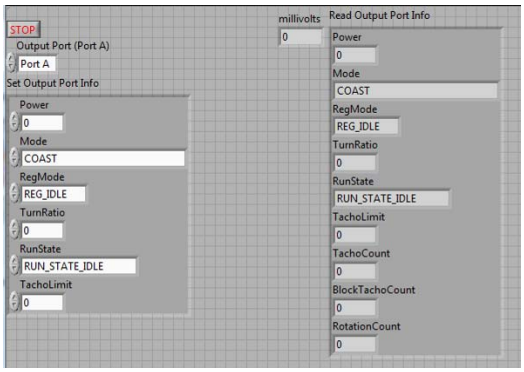
3. Right-Click the third input port of the **Set Output Values** VI icon, i.e., **Output Port Info**, and select **Create>>Control** from the shortcut menu.



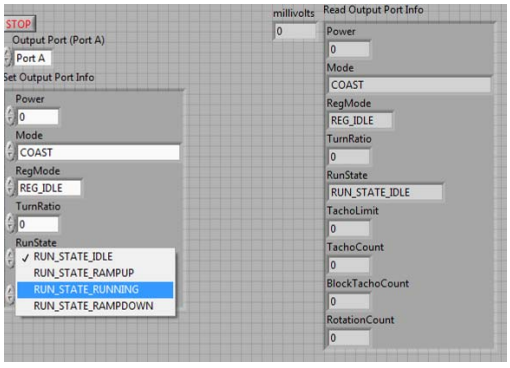
4. Rename the control, **Output Port Info 2**, to **Set Output Port Info**. Rename the indicator, **Output Port Info**, to **Read Output Port Info**.



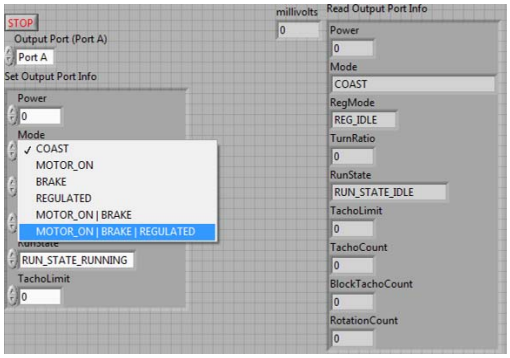
5. Now you should have the front panel similar to the following figure.



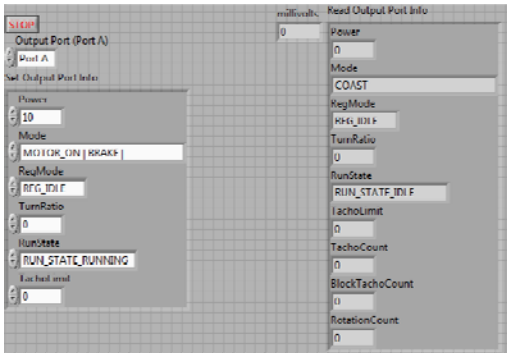
6. Press the run button on the toolbar and you will find that the motor won't move. Because we didn't have the proper setting for the motor from **Set Output Port Info** control.
7. Click the **RunState** control, and change the value from **RUN_STATE_IDLE** to **RUN_STATE_RUNNING**.



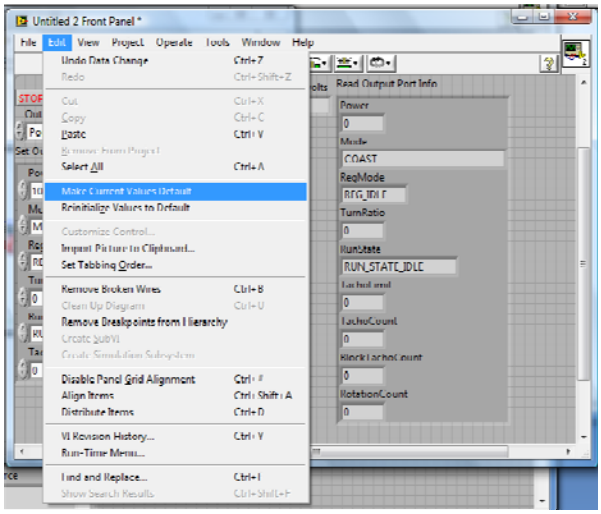
8. Click the **Mode** control and change the value from **COAST** to **MOTOR ON | BRAKE | REGULATED**.



9. Click the Power control, and change the value from 0 to 10.



10. From the windows menu, select **Edit>>Make Current Values Default**.



11. Save the code for future usage. Now you can press the **Run** button from the toolbar and see the motor rotating until you press the **STOP** button.

Exercise: Modify the code to control motor B and C simultaneously.

LEGO® MINDSTORMS® NXT Lab 4

In this lab session, you will learn how to write a code to communicate Wiimote via Bluetooth using LabVIEW.

Note: Please follow the procedures mentioned in Lab 1 Part B first to make sure that you can connect your Wiimote to PC via Bluetooth successfully.

Lab Sections

A. Read XYZ acceleration status of Wiimote
B. Read button status of Wiimote

Software

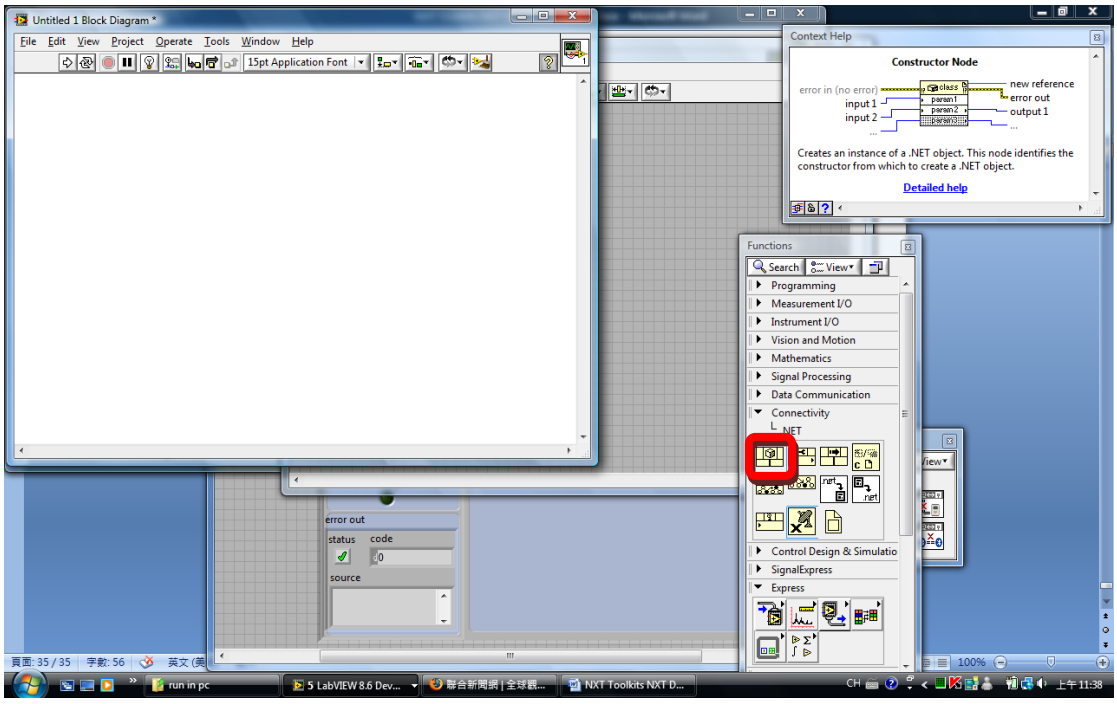
- NI LabVIEW with Nintendo's Wiimote library by Brian Peek

Hardware

- Nintendo's Wiimote
- Laptop or computer with D-Link Bluetooth DBT-120 dongle

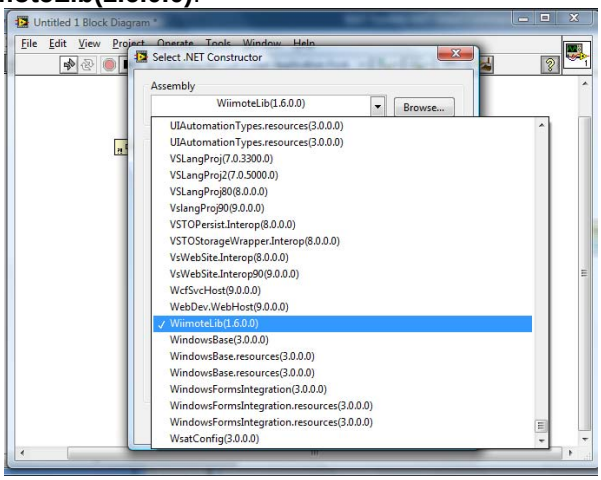
Part A: Read XYZ acceleration status of Wiimote

1. Open LabVIEW 8.6 by clicking on **Windows Start** icon **>> All Programs >> National Instruments LabVIEW 8.6.** Then start a blank VI by clicking on **New>> Blank VI**.

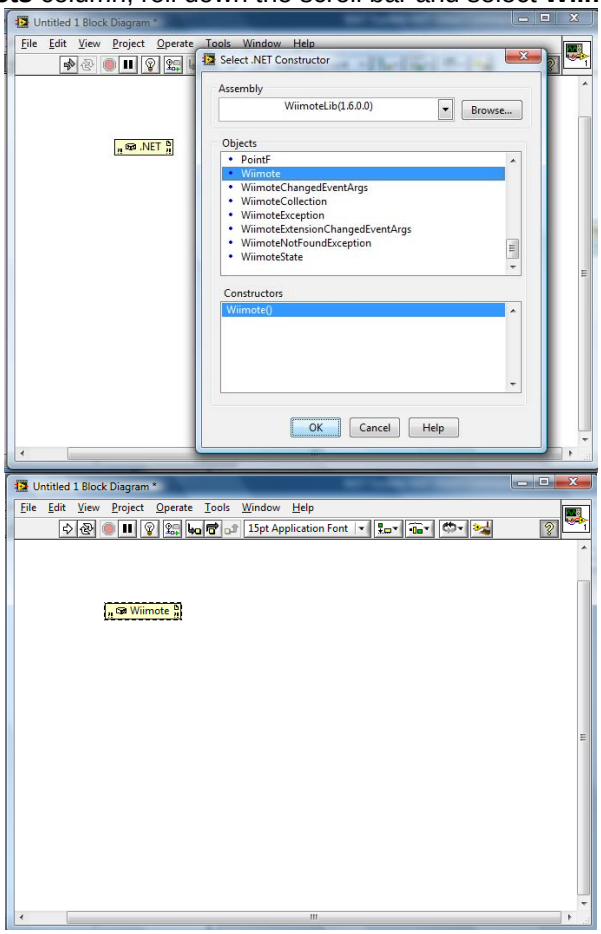


2. From the function palette, select **Connectivity>>.NET>>Constructor Node**, and place it on the block diagram. You will have a popup window asking you to select .NET

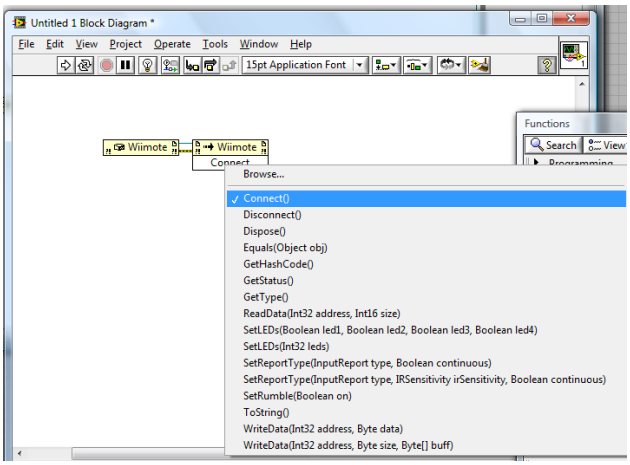
Constructor. Click the **Assembly** column, from the shortcut menu roll down the scroll bar and select **WiimoteLib(1.6.0.0)**.



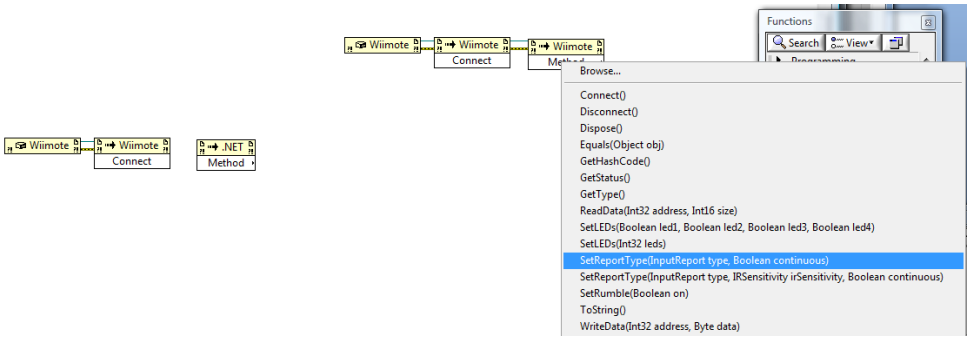
3. From the **Objects** column, roll down the scroll bar and select **Wiimote**. Then click **OK**.



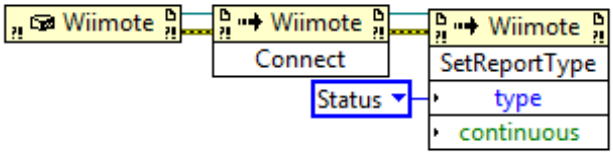
4. From the function palette, select **Connectivity>>.NET>>Invoke Node (.NET)**, and place it next to the **Constructor Node** on the block diagram. Wire the reference and error information between **Constructor Node** and **Invoke Node**. Click on the **method** of the **Invoke Node** and select **Connect()** from the short cut menu.



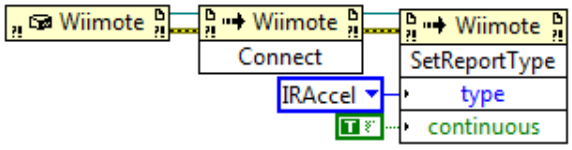
5. From the function palette, select **Connectivity>>.NET>>Invoke Node (.NET)**, and place it next to the previous **Invoke Node** on the block diagram. Wire the reference and error information between these two nodes. Click on the **method** of the **Invoke Node** and select **SetReportType(InputReport type Boolean continuous)** from the short cut menu.



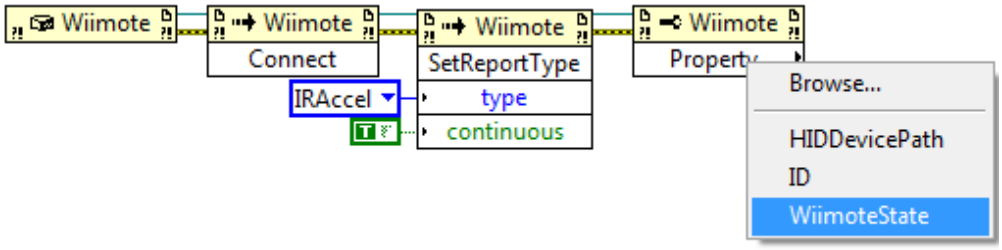
6. Right-Click the first input port of this **Node,** i.e., **type**, and select **Create**>>**Constant** from the short cut menu.



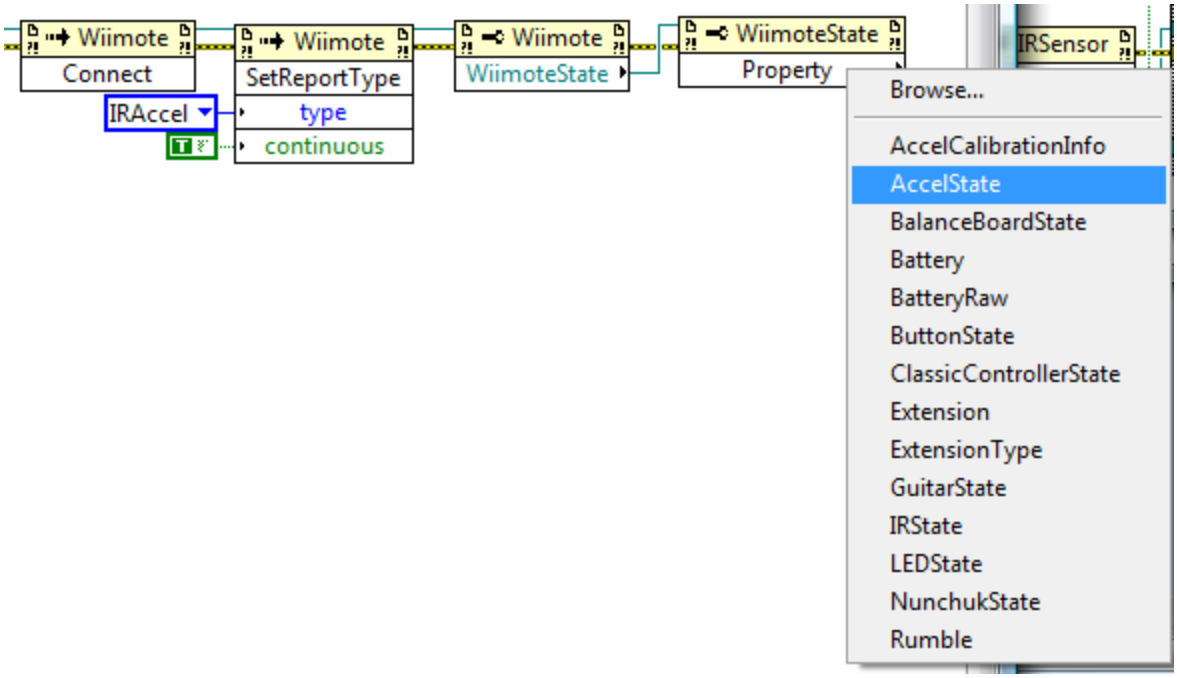
7. Change the constant value from **Status** to **IRAccel**. Right-Click the second input port of the same **Node,** i.e., **continuous**, and select **Create>>Constant** from the short cut menu. Change the constant value from **False** to **True** by clicking on T character of the constant.



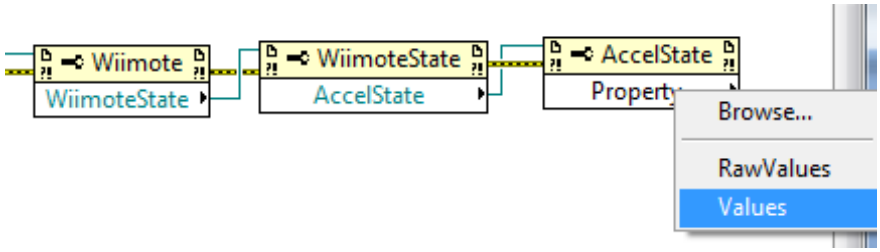
8. From the function palette, select **Connectivity>>.NET>>Property Node (.NET)**, and place it next to the previous **Invoke Node** on the block diagram. Wire the reference and error information between these two nodes. Click on the **Property** of the **Property Node** and select **WiimoteState** from the short cut menu.



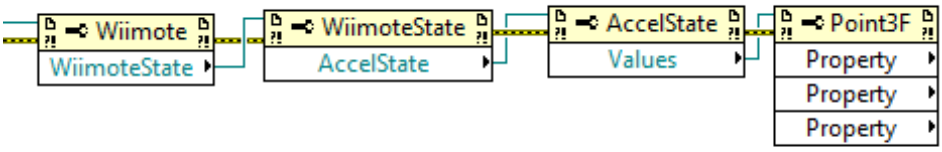
9. From the function palette, select **Connectivity>>.NET>>Property Node (.NET)**, and place it next to the previous **Property Node** on the block diagram. Wire the reference from output port of previous **Property Node**, i.e., **WiimoteState** and error information between these two nodes. Click on the **Property** of the **Property Node** and select **AccelState** from the short cut menu.



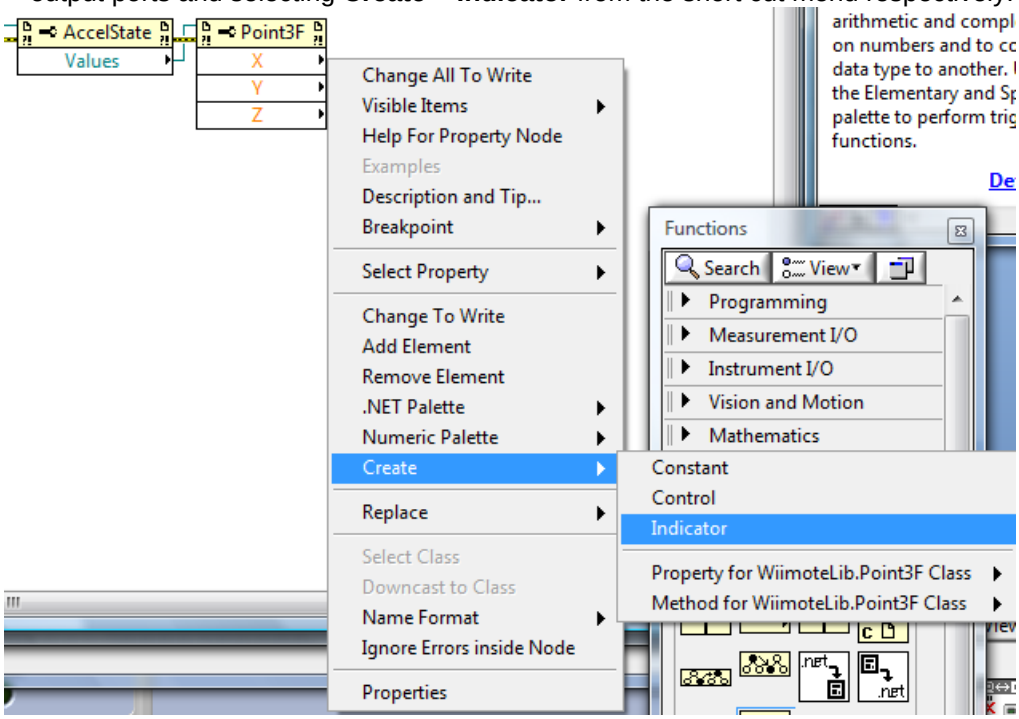
10. From the function palette, select **Connectivity>>.NET>>Property Node (.NET)**, and place it next to the previous **Property Node** on the block diagram. Wire the reference from output port of previous **Property Node**, i.e., **AccelState** and error information between these two nodes. Click on the **Property** of the **Property Node** and select **Values** from the short cut menu.



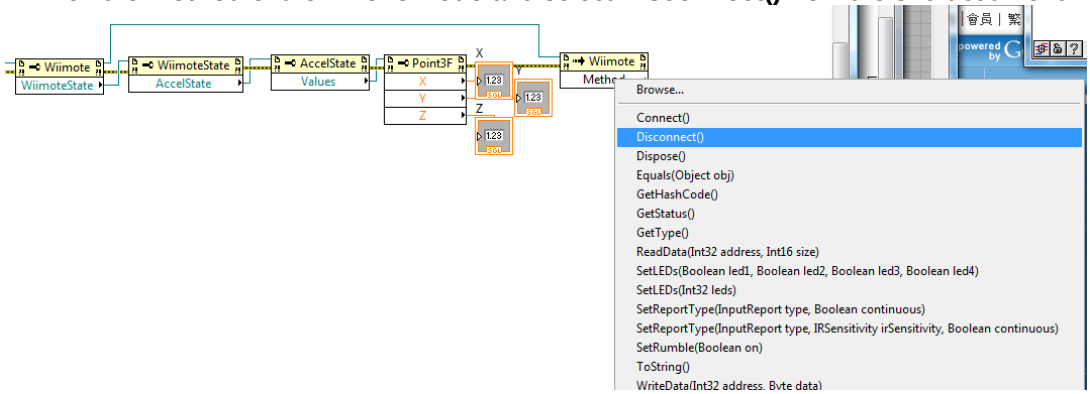
11. From the function palette, select **Connectivity>>.NET>>Property Node (.NET)**, and place it next to the previous **Property Node** on the block diagram. Wire the reference from output port of previous **Property Node**, i.e., **Values** and error information between these two nodes. Move the cursor to the button edge of the Node. You will see a pair of Blue Square shown up at the top and button edges of the icon. Drag the button edge down to have three **Property** sets in the **Node.** Click on each **Property** of the **Node** and select **X, Y,** and **Z** from the short cut menu respectively.



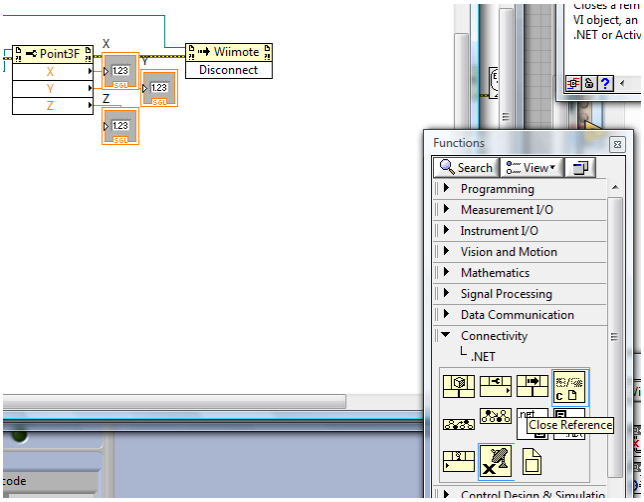
12. Create indicators for the outputs of the **Point3F Property Node** by right-clicking each output ports and selecting **Create>>Indicator** from the short cut menu respectively.



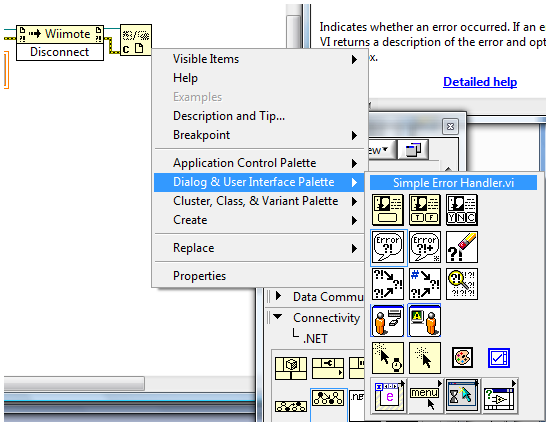
13. From the function palette, select **Connectivity>>.NET>>Invoke Node (.NET)**, and place it next to the previous **Property Node** on the block diagram. Wire the reference from the **Wiimote Property Node** and error information from the **Point3F Property Node**. Click on the **method** of the **Invoke Node** and select **Disconnect()** from the short cut menu.

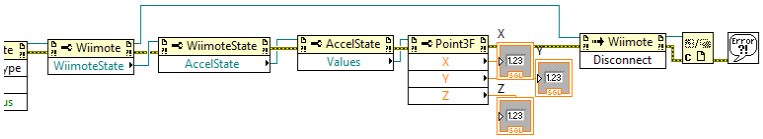


14. From the function palette, select **Connectivity>>.NET>>Close Reference**, and place it next to the previous **Invoke Node** on the block diagram. Wire the reference and error information between these two icons.



15. Right click the output port of the **Close Reference** VI icon, and select **Dialog & User Interface Palette>>Simple Error Handler**, place it next to the **Close Reference** VI icon on the block diagram. Wire the error information between these two icons.



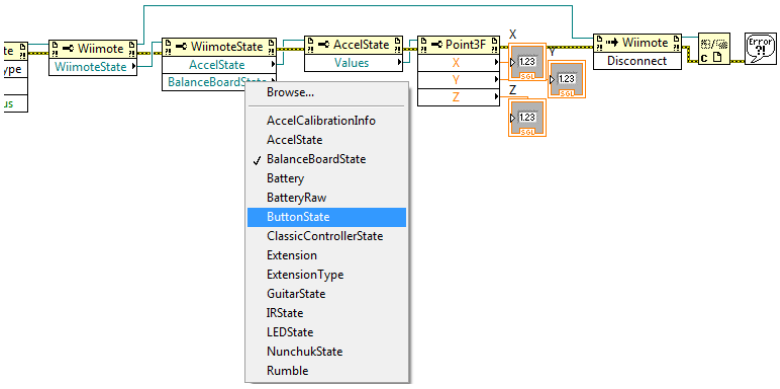


16. Save the code for further usage.

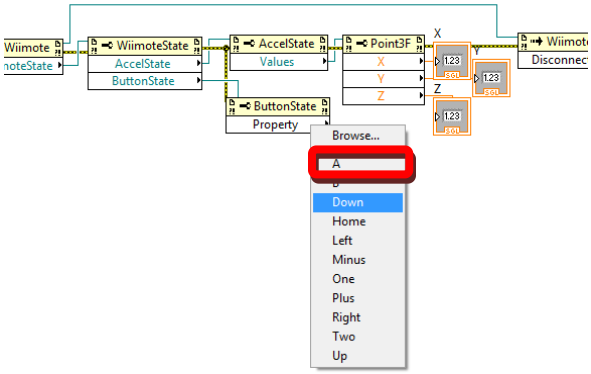
Exercise: Add a while loop structure in the code to keep updating the readings of Wiimote acceleration status.

Part B: Read Button Status of Wiimote

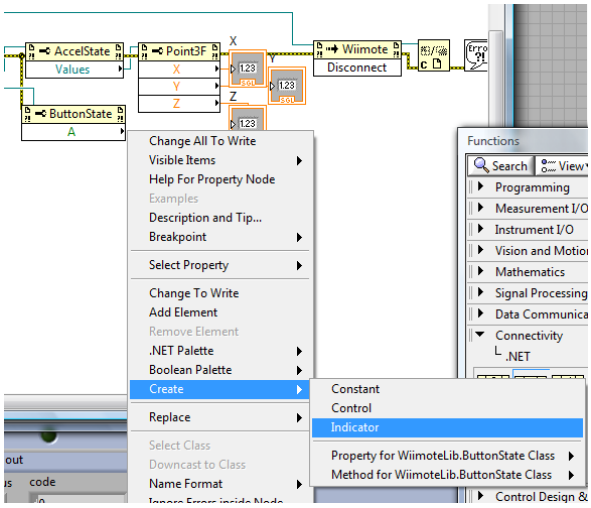
1. Open the file you create from Lab4 Part A. Open the block diagram window. Move the cursor to the button edge of the **WiimoteState Property Node**. You will see a pair of Blue Square shown up at the top and button edges of the icon. Drag the button edge down to have another **Property** set in the **Node.** Click the second **Property** of the **WiimoteState Property Node**, and change the property from **BalanceBoardState** to **ButtonState** from the short cut menu.



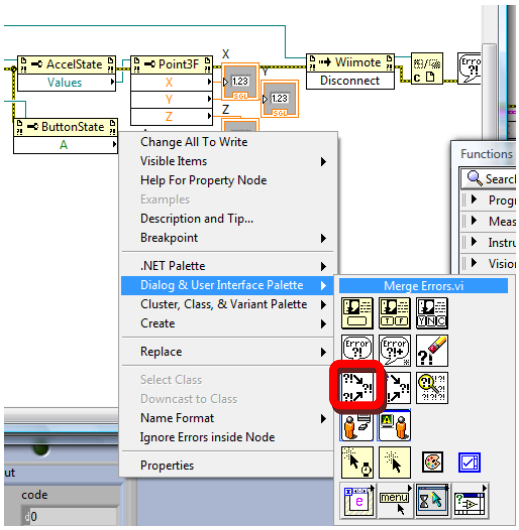
2. From the function palette, select **Connectivity>>.NET>>Property Node (.NET)**, and place it next to the **WiimoteState Property Node** on the block diagram. Wire the reference from the second output port of **WiimoteState Property Node**, i.e., **ButtonState** and error information between these two nodes. Click the **Property** of the **ButtonState Property Node** and select **A** from the short cut menu.



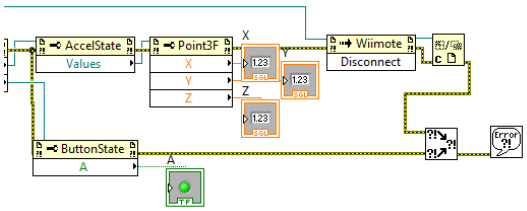
3. Right-click the output port, A, of the **ButtonState Property Node**. Select **Create >> Indicator** from the short cut menu to have an indicator on the front panel showing the status of Button **A** of Wiimote.



4. To merge the error wires in the program, right-click the error output port of **ButtonState Property Node**. Select **Dialog & User Interface Palette>>Merge Errors** from the short cut menu, and place it on the block diagram.



5. Delete the error wire between **Simple Error Handler** and **Close Reference** VI icons. Wire the error outputs of **Simple Error Handler** VI icon and **ButtonState Property Node** VI icon to the inputs of **Merge Errors** VI icon, respectively. And wire the error output of **Merge Errors** VI icon to the input of **Simple Error Handler** VI icon.



6. Now you have the code that can read one button status and XYZ acceleration Status of Wiimote.

Exercise:

a) Add a while loop structure in the code to keep updating the readings of Wiimote button status. (Don't forget to add a 40 ms time delay in your while loop structure)
b) Modify the code so that you can read more than one button status.
c) Combine the codes of Wiimote and NXT motor control you learn from Lab3 and Lab4 to enable users to use Wiimote to control NXT motor. (Hint: use the value of acceleration status of Wiimote as the input value of motor power parameter setting.)