

Decision USB Series Products in LabVIEW

The basic idea is used the dll file “USBDII.dll” which are written in Visual C++ and provides lots of functions to perform a variety of Analog input/output, Digital input/output and RS-232/RS-485 Communication operations with the hardware of the Decision USB Device. You can find user manual about USBDII in Product CD package or our website. The DLL files can be easily used by higher-level computer language. For example, it provides a large variety of demo programs that are written in Visual C++, Delphi, Visual Basic and even in LabVIEW.

In this document, it contains two concepts and four LabView VI files.

1. Call functions in USBDII.dll in LabVIEW
2. Directly use the Decision USB Device component in LabVIEW

VIs

1. USBDII.vi

It contains “**Call Library Function Node**” for USBDII.dll. This VI can’t be executed.

2. Decision_USBDevice.vi

It has already been packed as an universal object for Decision USB Device.

User can directly use this VI as a SubVI in their own project.

3. ex_1Device.vi

It demonstrates one device (USB16PR) built by SubVI “**Decision_USBDevice.vi**”.

This VI can be executed directly.

4. ex_2Device.vi

It demonstrates two devices in the same project and use clusters to simplify.

This VI can be executed directly.

Stand-alone exe file

LabVIEW can let users export their VI into a stand-alone exe file to execute their project without develop environment. But still needs to install “LabView Runtime Engine”. You can download it on LabVIEW official website for free. And then execute the exe file in “exe_file_for_ex_2Device” folder to experience Decision USB devices in LabVIEW.

Call functions in USBDI.dll in LabVIEW

We use “**Call Library Function Node**” to call subroutines in DLL directly in LabVIEW.



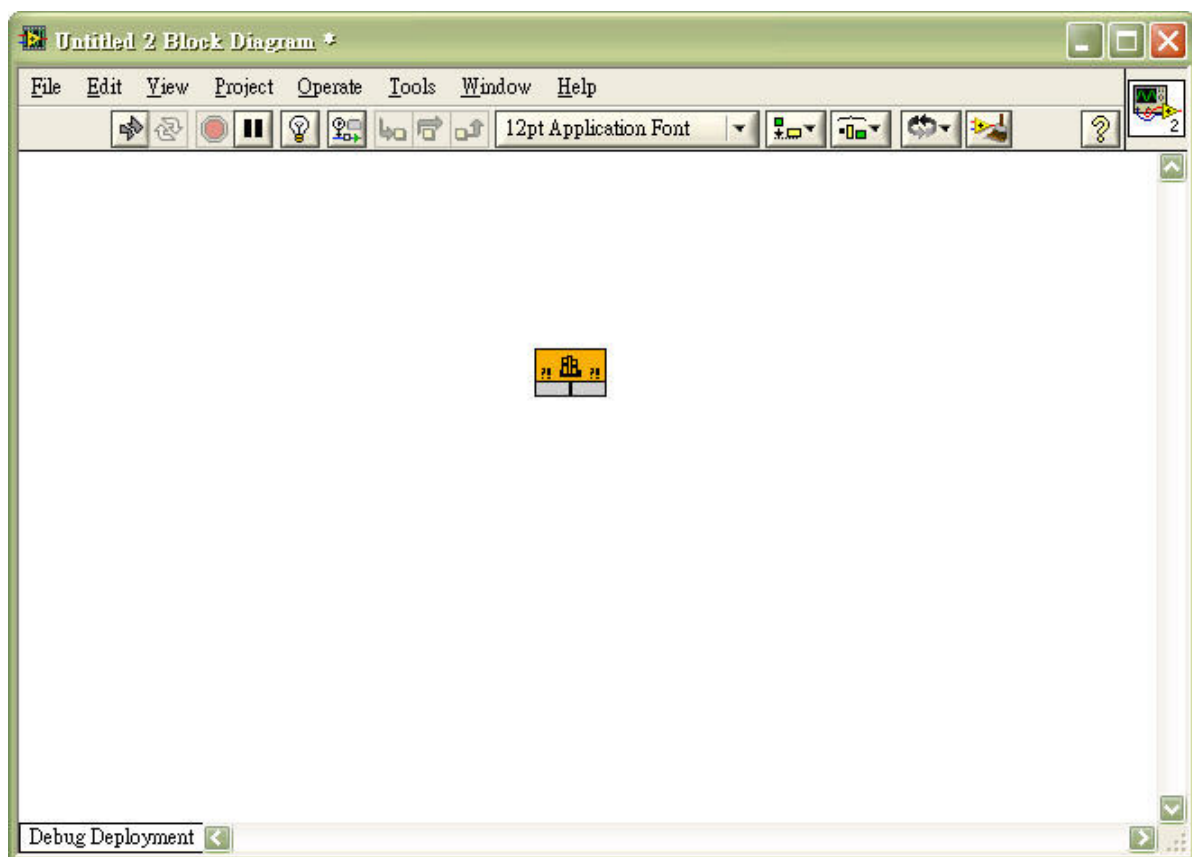
Step 1:

Execute LabVIEW and press “New VI”.

Step 2:

In Block Diagram, select Functions -> Connectivity

-> Libraries & Executables -> **Call Library Function Node**.



Step 3:

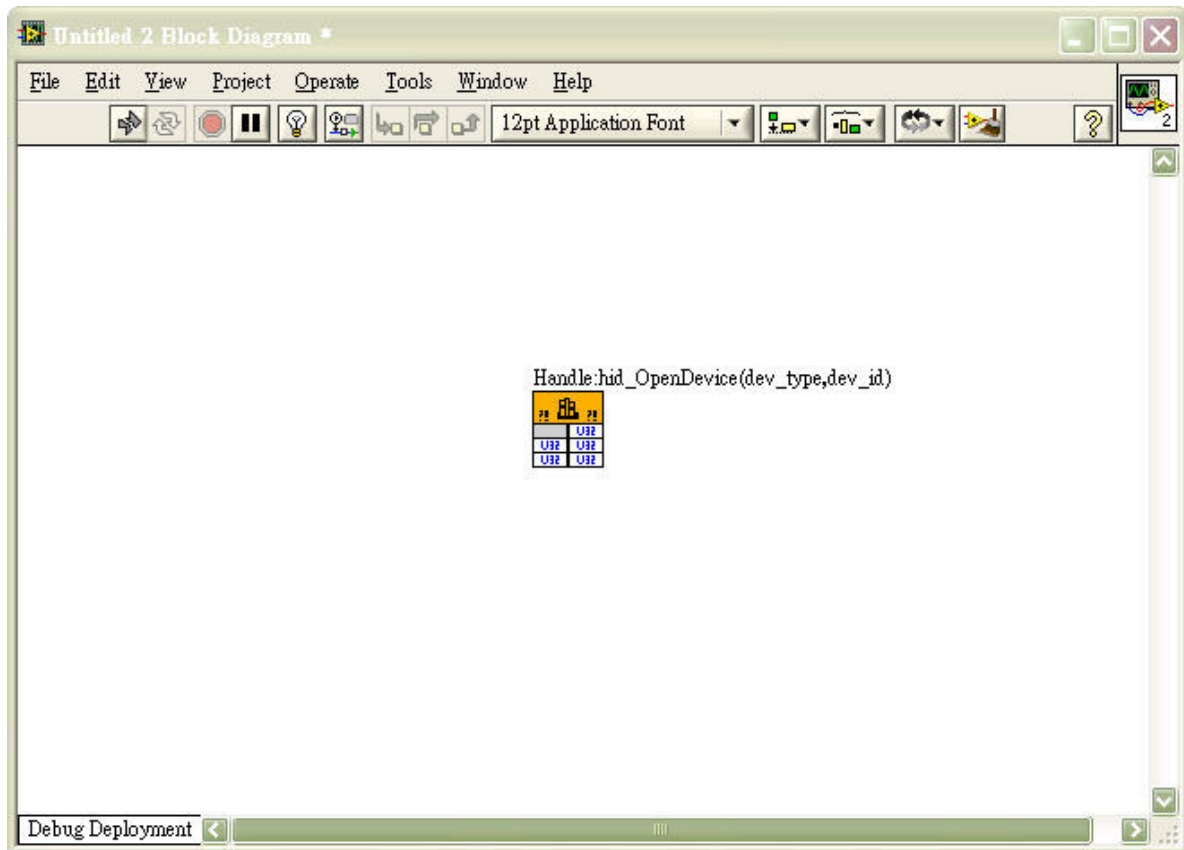
Press right button into its configure setting.

In “Function” page, you can select the path for the dll file and its function name.

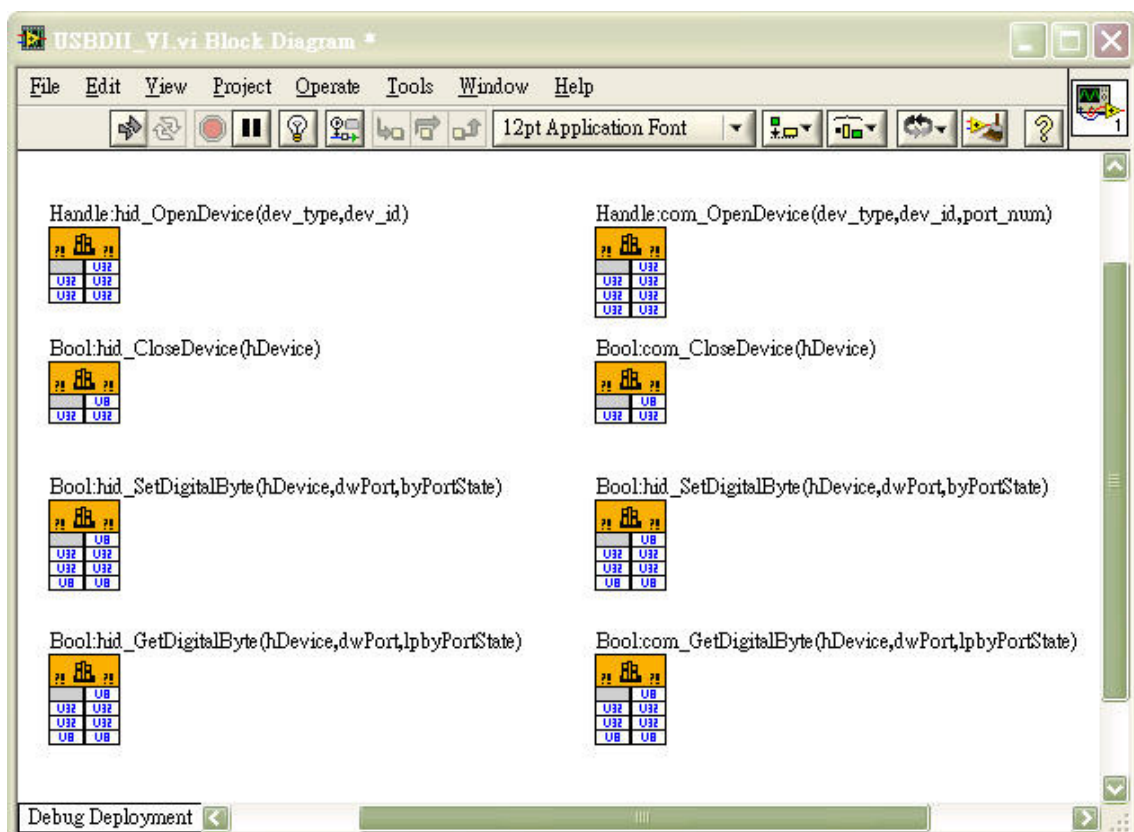
In “Parameters” page, you can set the parameters for the function and its return value.

We use “HANDLE hid_OpenDevice (DWORD device_type, DWORD device_id)” in USBDI for example.

About functions in USBDI, user can find more detail in “USBDI_Manual.pdf”.

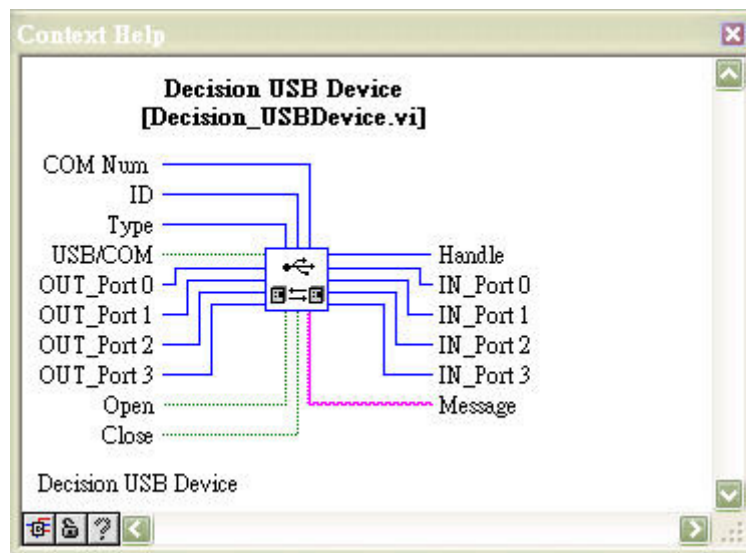


User can use components we already built directly which contains functions in USBDI.dll in “USBDI.vi”.



Directly use the Decision USB Device component in LabVIEW

To connect Decision USB Device with LabVIEW, you can use “**Call Library Function Node**” to call subroutines in DLL to fit your own project or simply use a SubVI “**Decision_USBDevice.vi**” we provided, which has already been packed as an universal object for Decision USB Device.



Inputs

Name	Type	Description
Open	Boolean	Trigger True to Open Device
Close	Boolean	Trigger True to Close Device
USB/COM	Boolean	Device Connect by USB or COM, true for USB / false for COM
Type	Unsigned Int32	Select Device Type (More detail in “ USBDII_Manual.pdf ”)
ID	Unsigned Int32	Select Device ID when multi-device connecting
COM Num	Unsigned Int32	Select COM Port Number when Device Connect by COM
OUT_Port0	Unsigned Int8	Set Write Value to Port0 (only works on devices Port0 as output)
OUT_Port1	Unsigned Int8	Set Write Value to Port1 (only works on devices Port0 as output)
OUT_Port2	Unsigned Int8	Set Write Value to Port2 (only works on devices Port0 as output)
OUT_Port3	Unsigned Int8	Set Write Value to Port3 (only works on devices Port0 as output)

Outputs

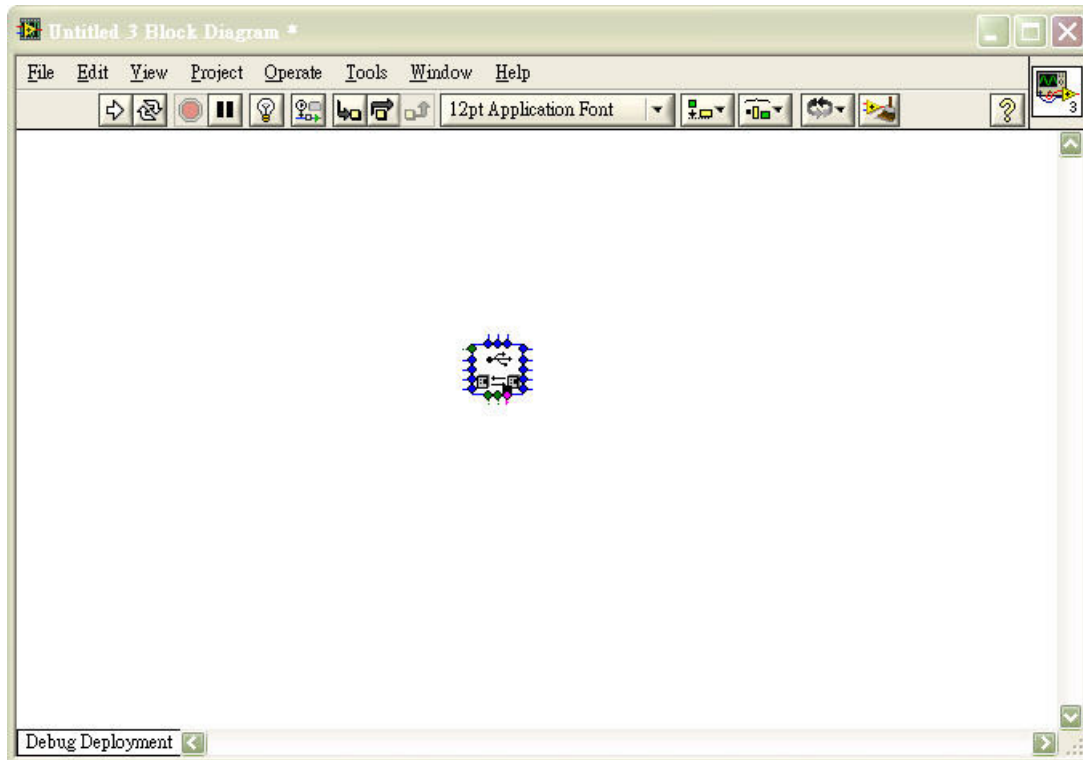
Name	Type	Description
Handle	Unsigned Int32	Return handle for the device (4294967295 means open error)
Message	String	Return some info for the device
IN_Port0	Unsigned Int8	Return Read Value from Port0 (only works on devices Port0 as input)
IN_Port1	Unsigned Int8	Return Read Value from Port1 (only works on devices Port1 as input)
IN_Port2	Unsigned Int8	Return Read Value from Port2 (only works on devices Port2 as input)
IN_Port3	Unsigned Int8	Return Read Value from Port3 (only works on devices Port3 as input)

We use USB16PR as example to build a VI in LabVIEW by Subvi “**Decision_USBDevice.vi**”

Step 1: Execute LabVIEW and press “New VI”.

Step 2:

In Block Diagram, select “Select a VI” and direct the path to “**Decision_USBDevice.vi**”



Step3:

Connect USB16PR by USB and ID on the board is default 0.

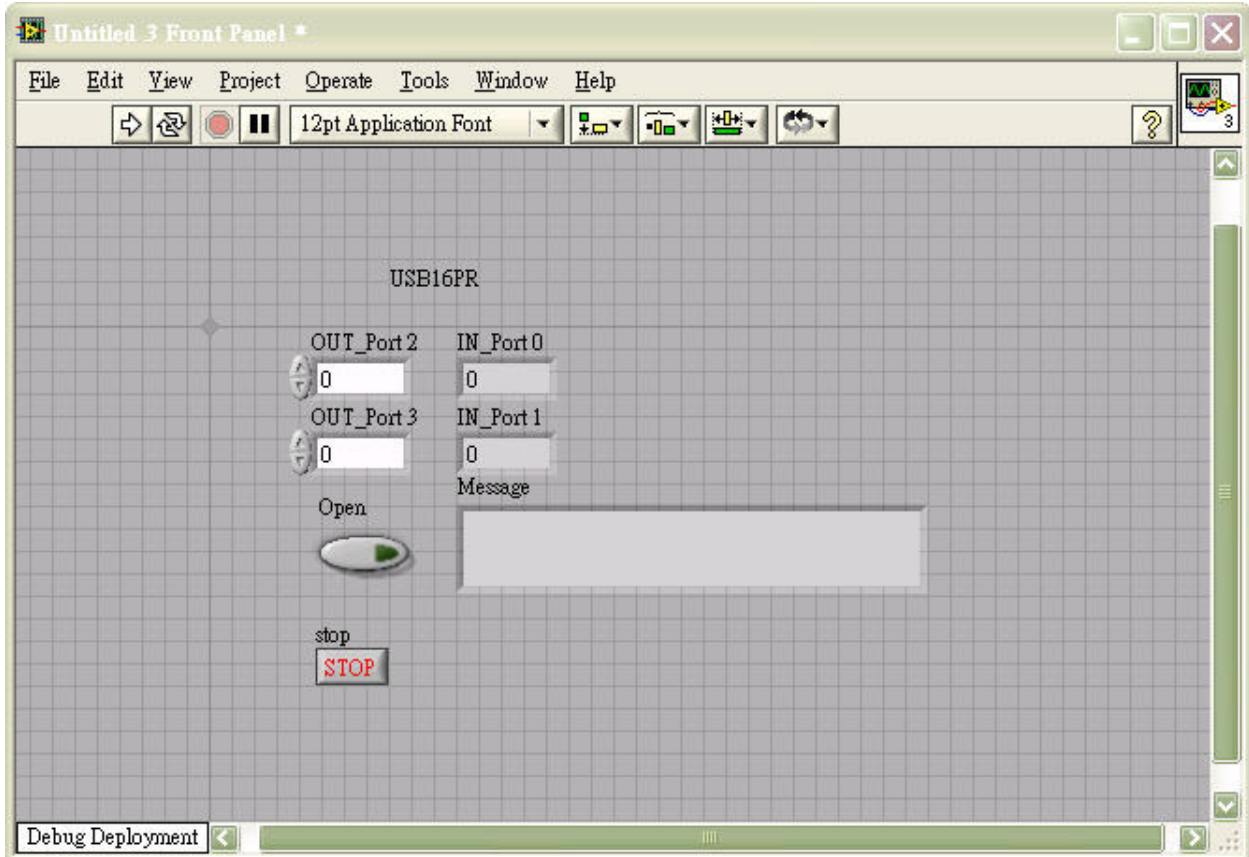
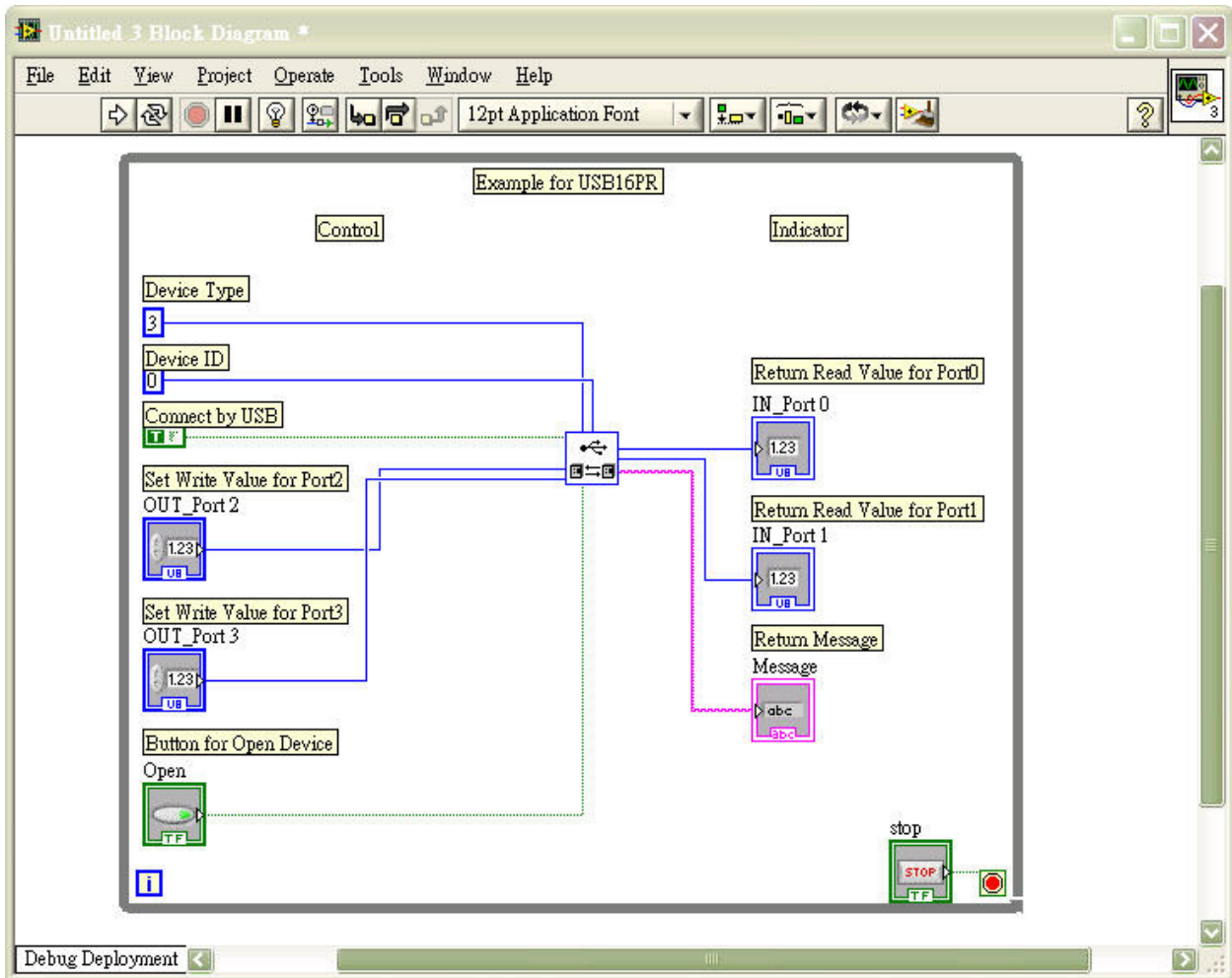
In this device, Port0 and Port1 are Input and Port2 and Port3 are Output.

So connect the control and indicator as following below.

You can find the example in “**ex_1Device.vi**”.

Notice:

1. You can only connect pins the device needs and ignore others.
2. You need a while loop to continue running and get result.
3. Control for Open only needs one time and if Open pins set always true in loop, the function `hid_OpenDevice()` will always be called and returns different handle value. It will affect the system efficiency and the result. The better way is change the open control “button behavior” as “Switch until released”, and Control for Close as well.



You even can combine input and output as “cluster” to simplify the diagram.

In “ex_2Device.vi”, it demonstrates two devices in the same project and use clusters to simplify.

