Decision USB Series Products in ProfiLab

ProfiLab Overview

The ProfiLab software is a powerful development tool for PC-based measurement and control applications. It combines features like front panel design and hardware control to one powerful tool, and can be used for circuit simulation, visualisation and presentation in many cases as well. ProfiLab comes along in three different versions and can be upgraded any time. Check out the Internet site at www.abacom-online.de/UK or latest information.

Decision Support for ProfiLab

Users can download a trial version of ProfiLab on the official website to experience. And Decision Computer also provides the example project and demonstration for our USB series products which allow users easy to connect and control our products in ProfiLab. The dll import files of Decision USB series products are in "dll for Decision USB Products" folder which define the device action and can allow user directly to use in the project.

Example Project in "ProfiLab Example for USB16PR" folder

1. USB16PR_0.prj

Basic example for "USB16PR" device in ProfiLab

2. USB16PR_1.prj

Advanced example for two "USB16PR" devices with LED display and hex display

3. USB16PR_wTable.prj

Advanced example for two "USB16PR" devices with LED display and hex display And record input/output channel of first device into table which can be export to an excel file. This following example shows how to load the dll into ProfiLAB as a hardware device component.

- 1. New a project DLLO 2. Choice "Miscellaneous" -> "Miscellaneous" -> "DLL Import" DLL import ProfiLab Expert 4.0 File Edit Front panel Configuration Options Window Register ? abl 🔲 🕕 🍳 🍳 🔍 😤 🍾 🗋 🚔 🖬 🎒 👗 🐚 🖻 ×2 🐐 R ? Miscellaneous * + Control elements ٠ D Logic components ٠ Q ~ Analogue components ٠ ? Miscellaneous Vcc,Ground, Test 😒 Hardware components 🔸 🖪 File access -DDE Macros Hotkeys X E Strings 4 Pint. Miscellaneous Macro pin 2 AL X BABO N? DLL Execute STOP STOP 5100 Simulation
- 3. Press right button and select "Properties"

.L import DLL file	?
No DLL file loaded.	
Import DLL	No functions imported.
	<u>OK</u> <u>C</u> ancel

4. Import DLL and select the destination of "USB16PR.dll".

If import successfully, functions will appear on the table and Configure button will also enable.

DLL import DLL file C:\Documents and Se	ettings\Administrator\点i	重\iconic\dll\delphi\F
[Imported function:	\$
<u>I</u> mport DLL	NumInputs = 16 NumDutputs = 16 InputName OutputName SimStart SimStop Calculate Configure	3
	<u>0</u> K	<u>C</u> ancel

5. Press "Configure" and you can set the ID for the USB16PR and test its connection.

Device Type	USB16PR
Device ID	0 🔹
Interface	© USB € COM COM1 -
Device State	Open Successfully

6. Now you have a USB16PR component to use.

File <u>E</u> dit Fro <u>n</u> t panel <u>C</u> onfigurat	on <u>O</u> ptions <u>W</u> indo	w <u>R</u> egister <u>?</u>
D 🛩 🖬 🎒 👗 🖻 🛙	×2 👘 abl	🔲 🔲 🔍 🖯
Riscellaneous ▼		
+ <u>ex</u> f <u>r</u>	0070	INO
	0071	IN 1
- (Teclerette)	0072	IN2
? counter	0073	IN3
→	0074 0075	IN4 IN5
		1116
	ש פרטס פי דרטס מיטס מיטס	IN7
2 DLL import	0070 8700	1110
	0070 🤍	///0
	00710	IN 10
Pint	00711	IN11
	00712	IN 12
2 Macro pin	00713	IN 13
	00714	IN 14
	OUTIS DUL	IN 15

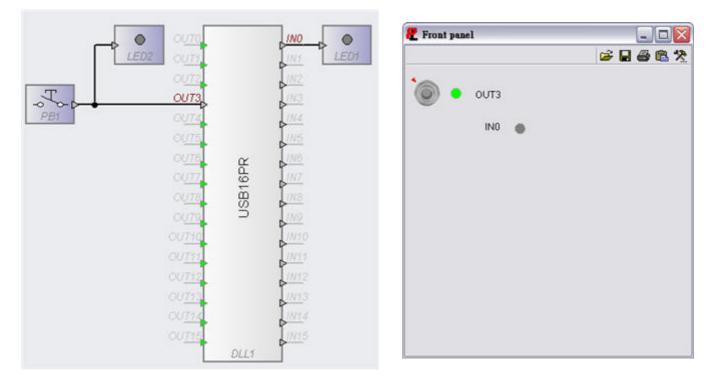
7. USB16PR contains 16 input channels (IN0~IN15) and 16 output channels (OUT0~OUT15).

Pin IN0~IN15 will receive signals from real device.

When it receives HIGH signal from real device, it will output HIGH or 5 in the software. Like the following figure, IN0 connects to a LED display, so when IN0 receives HIGH signal from real device, IN0 will light the LED on the display panel in the software.

Pin OUT0~OUT15 will generate signals to real device.

When it receives HIGH signal from the software, it will generate signal to real device. Like the following figure, OUT3 connects to a digital control (push button) and a LED, so when user press the button on the display panel, the digital control will give HIGH to the OUT3 and light LED on the display panel and OUT3 in the real device will generate high signal as well.



Notice:

- 1. Output pins witch are not connect to any signal will receive High in the software and make the real device output HIGH. To avoid this situation, you can connect them to ground component. You can find the example on the USB16PR_0.prj.
- 2. "USB16PR.dll" is used to define hardware device. "USBDII.dll" is for hardware device control. Please make sure to put them in the same folder with project file.