

Frequently Asked Questions

Title: Upgrade from PCI/Px to M4K1553Px

Description: To move an application from EXC-1553PCI/Px to the M4K1553Px module on the EXC-4000PCI board

Date: 02 Mar 15

Card/Board/Module: PCI/P1 and EXC4000PCI[e]/F1

Operating System: all Windows (32bit, 64bit)

Question:

How can I move my application from EXC-1553PCI/P1 to run on EXC-4000PCI/F1 ?

Answer:

1. It is necessary to install the software tools for the specific module/board, in this case, M4K1553Px module on EXC-4000PCI board. This installs two DLLs - one for the 4000PCI base board, and one for the M4K1553Px module.

The DLL name is different:

M4K1553Px module: Exc1553Px.dll (for Borland compiler) or Exc1553PxMs.dll (for Microsoft compiler). px.dll or pxms.dll are also available.

PCI/Px board: pcipx.dll (for Borland compiler) and pcipxms.dll (for Microsoft compiler).

In your project file, replace the use of pcipx.lib/pcipxms.lib with Exc1553Px.lib/Exc1553PxMs.lib, or rename these new DLLs (Exc1553Px[Ms]) with the previously used names (pcipx[ms]).

2. The function names have been preserved, almost in entirety, for backwards compatibility. This assumes that you have only one module on the board. If you have more than one module, it will be necessary to move to the some of the new function names. Please see item #4.

3. Initialize/Release differences:

PCI/Px board : Init_Card initializes all modules on the board. Release_Card releases resources for all modules.

M4K1553Px module: Init_Module_Px initializes only the specific module as per the function parameters. Use Release_Module_Px to release resources for this specific module.

NOTE: If you are porting a project from PCI/P1 to EXC-4000PCI with one M4K1553Px module (in position 0), then you can use the existing application source code as is, including function calls Init_Card and Release_Card. We added these functions to the DLL (in source file initcard.c) to handle PCI/P1 or PCI/EP cards.

NOTE: If you have an EXE that ran on a PCI/P1, and want to run the EXE on a EXC-4000PCI with one M4K1553Px module (in position 0) without recompiling, then install the M4K1553Px software tools, rename the DLL (for the appropriate compiler) to the same name that was used in the application (pcipx.dll or pcipxms.dll), and run the EXE.

4. Init_Module_Px takes two arguments:

device_number - use default value of 25, or as defined using ExcConfig

module_number - the specific module on the board that you wish to initialize (0,1,2,3)

Upon success, the function return value is designated as the "handle" for this module. Use this "handle" as the first (additional) parameter in all function calls.

All function names have been modified to have a suffix of "_Px". If you wish to use the old function names, which do not use the "handle" parameter, please see item #5.

5. To use the old function names (without the "_Px" suffix, and without the additional "handle" parameter), include the file `proto_px_back.h` in your application. This includes the function prototypes for the backwards compatible functions. Then, replace `Init_Card` with `Init_Module_Px`; and `Release_Card` with `Release_Module_Px`.

6. If you have more than one module, you must call the function `Init_Module_Px` for each module. The function returns a "handle" value.

All function calls apply to the currently selected module. If you use the new function names (with the suffix "_Px"), then use the "handle" for this module as the first parameter. If you want to use the old function names, include the file `proto_px_back.h` in your application, and proceed as in item #7.

7. Using old function names: Use `Select_Module` to select a specific module. Then call the mode specific functions that you wish to apply to this module.

Therefore, to switch to another module (i.e., to apply function calls to another module), call `Select_Module(other_module_number)`, and then call the mode specific functions that you wish to apply to this module.

To switch back to the first module, call `Select_Module(first_module)`, and then call the mode specific functions that you wish to apply to this module.