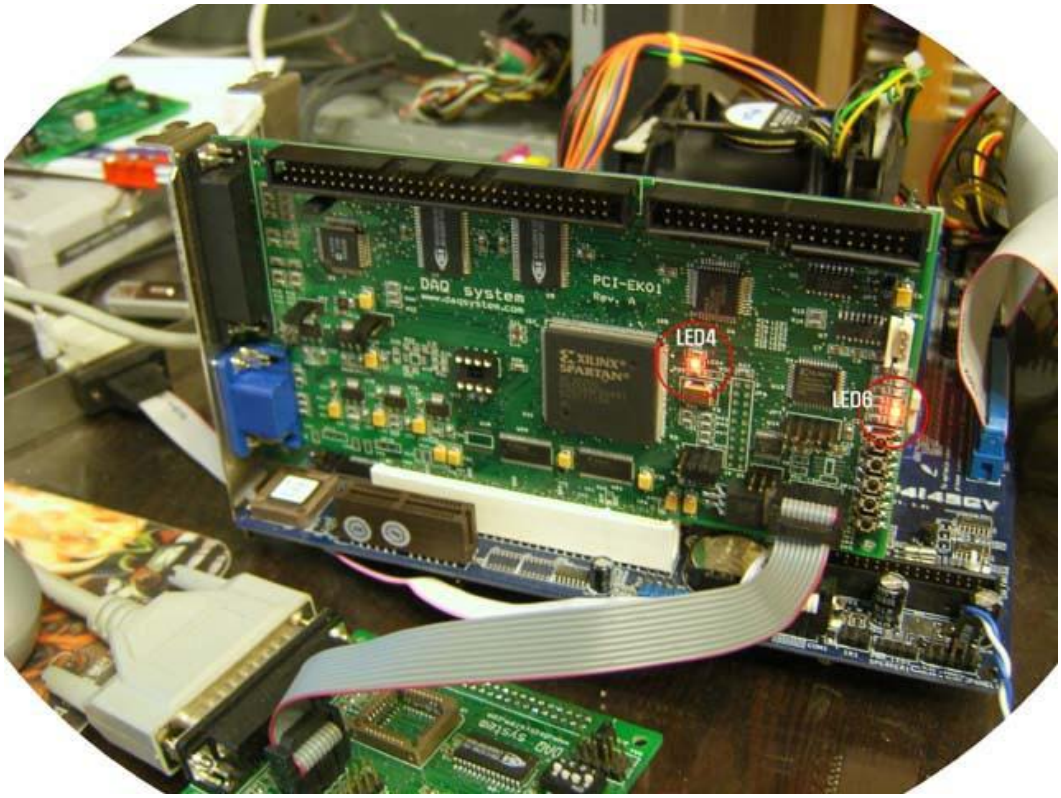
How to build application using API



Benefits of using client DLL for building application

When the user program is used by DAQ boards, you can control a register of the board to program. It is called at "Register Level Programming". You use a client DLL to provide in board manufacture companies in other way. It is called at "API(Application Programming Interfaces) Programming.

The advantage of the "Register Level Programming" can control the board functions in detail. But, you have to know about each register using, so this method spends a lot of times and increases the probability to fail when program coding. And, it can cause a fatal issue like system down in case of wrong writing in a program.

The advantage of the "API Programming" can made a user's program easy and fast without knowing each register. The disadvantage can't control a function in detail and the debugging is very difficult for solving it. However, a programmer prefers its method.

There is that Client DLL(API) to provide in board manufacture company provides freely it, but shall buy it according to each enterprises. DAQ System is providing a Client DLL basically free, and a manual is providing for register level's programming.

How to link the client DLL to the program project

There are two ways in order to let a user program link client DLL. The method of suggestive connection is the first; the second is the method of explicit connection. We will explain it with Windows Operating System and Visual C++ basically.

1. Suggestive Connection

(1) First, make a project of a basic program.

(2) To copy from "xxxx.dll", "xxxx.lib", "xxxx.h" which it provided at DAQ to a project.(xxxx is a name of each board.)

xxxx.dll : Client Dynamic Link Library

xxxx.lib : Import Library

xxxx.h : Function define Header

(3) A way to each file is as follow.

Suggestive Link Library can be linked when a program is performed as it make copy to a folder of an execution file or a system folder. An order to find "DLL" is as follows.

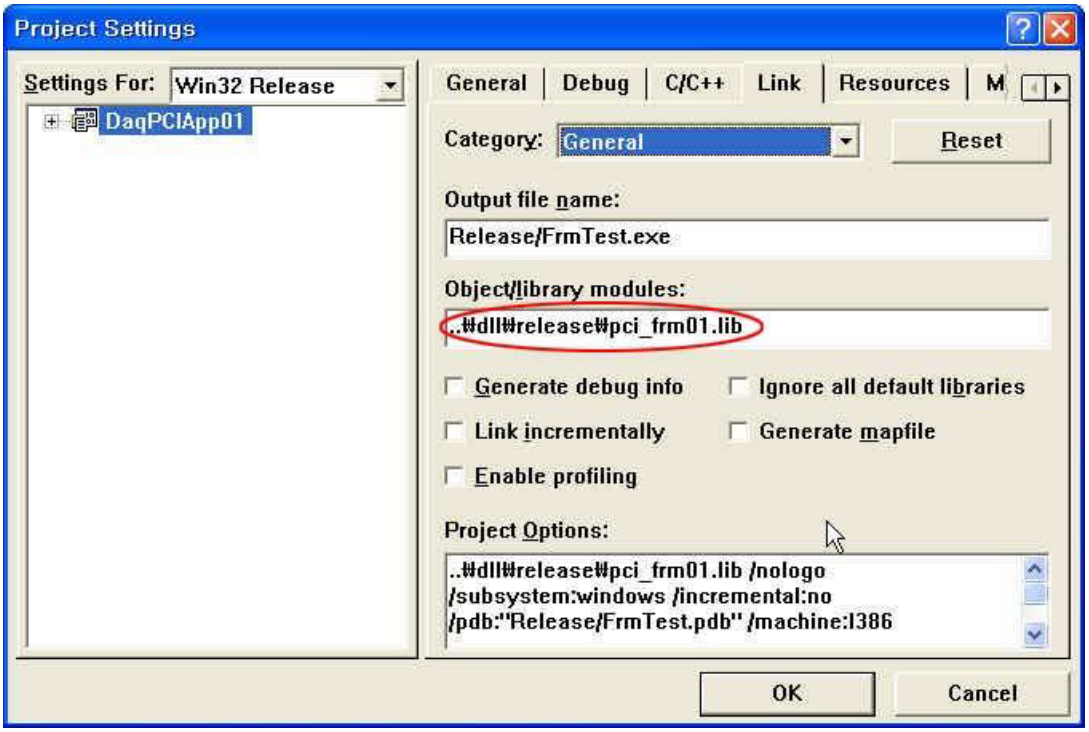
<1> A folder with user's program

<2> Current folder at process

<3> Windows System folder

<4> Folder with PATH Environment variable

An Import Library sets it up in project settings in case of a program link. It set up like the red circle content at the following picture. “pci_frm01.lib” was used at picture, but sets up library file name to get to each board.



Finally, an Import Header File include file of “#include” command to each program source calling API as below.

```
#include "frm01_import.h"
```

Notice : An Import Library isn’t thing with actual execution code image unlike static library. The Import Library has the contents regarding necessary function definition in case of dynamic link.

2. Explicit Connection

The suggestive method can easily connect necessary DLL in project settings because the DLL file name is set up. An explicit connection method designates a DLL file at program source without project setting.

(1) For link DLL, use “LoadLibrary” function that is Windows API as below.

```
Handle = LoadLibrary(“pci_frm01.dll”);
```

If the LoadLibrary function connect DLL file, it returns “handle value”. If the function fails, it returns “NULL”.

(2) DLL use is over, and it uses a “FreeLibray” of Windows API in order to release a link. A factor uses the handle which acquired at LoadLibrary.

```
FreeLibrary(Handle);
```

(3) For use a function in DLL, you can use GetProcAddress for get a pointer. A factor uses the handle which acquired at LoadLibrary.

```
DWORD count;
unsigned char buffer[0x600000];
BOOL returnVal;
typedef BOOL (myfunction)(DWORD *nCnt, unsigned char* buf);
myfunction* pfunc;
pfunc = (myfunction*)GetProcAddress(Handle, “LVDS_GetFrame”);

returnVal = (*pfunc)(count, buffer);
```

Notice : It is easy to use the Suggestive Connection more than the Explicit Connection. But, the Explicit Connection use at programs if it make dynamic link library link/cancellation at any time.