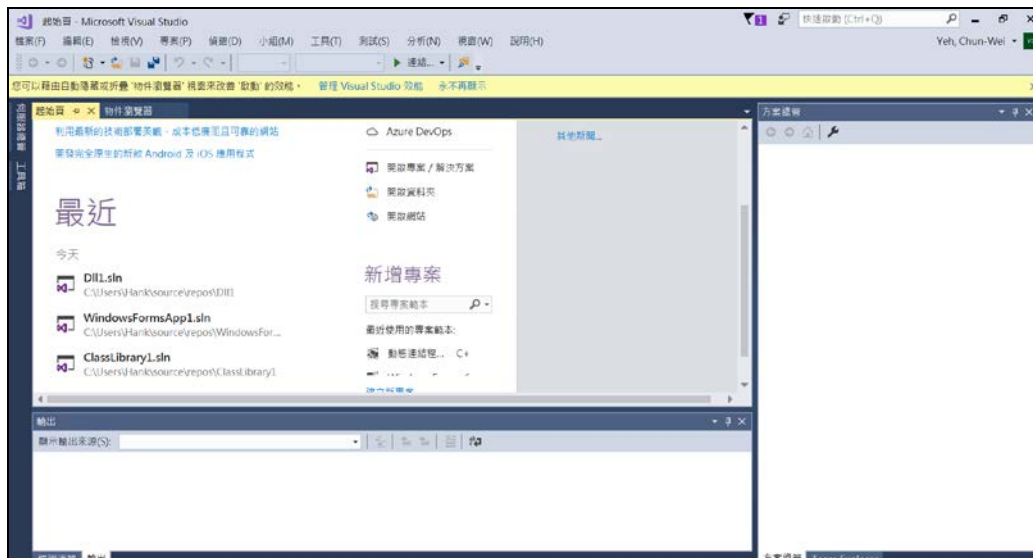# 國立臺北科技大學自動化所 嵌入式工業機器視覺

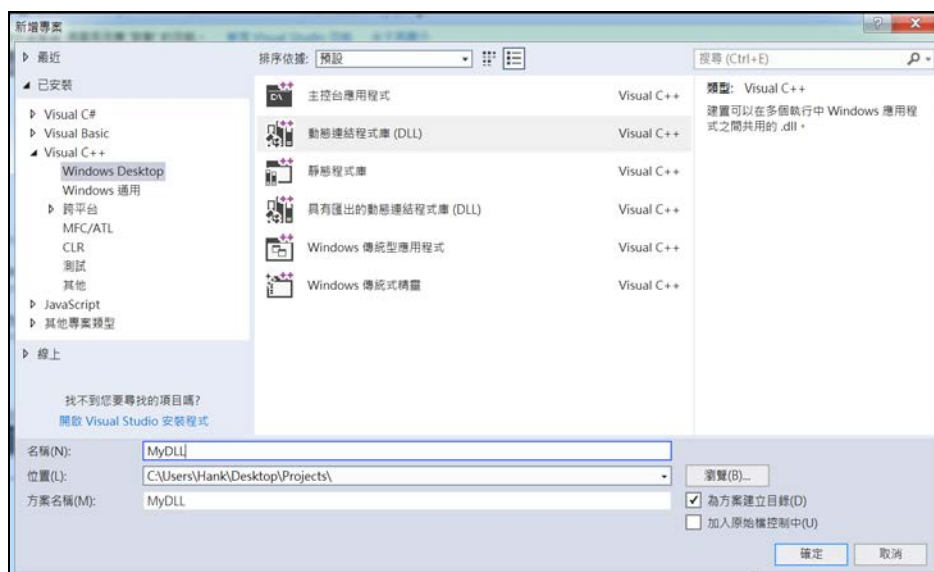## *Creating Dynamic Libraries (*.dll) on Windows*

[Using P/Invoke for C Functions]

The following steps describe how to create a dynamic library on Windows. These steps are for Microsoft Visual Studio 2017, although the steps are similar for other versions of Visual Studio.

1. Select the menu File > New > Project

2. Select the Visual C++ -> Windows Desktop and the DLL option, and then put the project name and path.
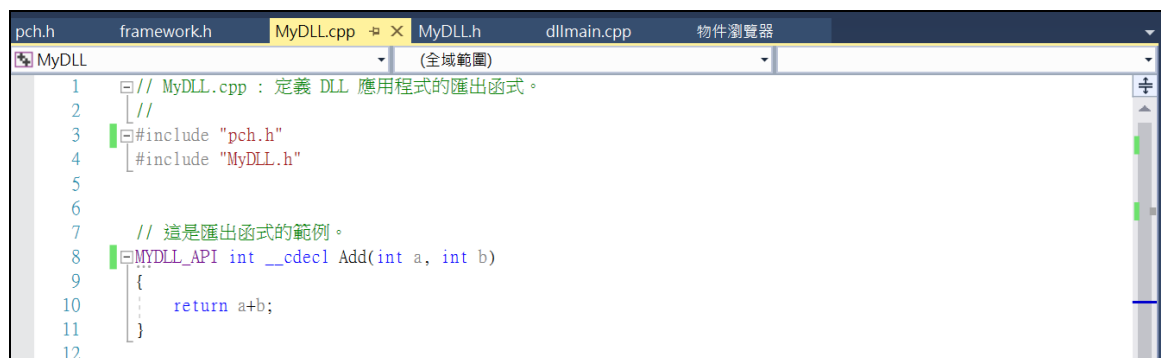
3. If you want a function to be callable from a DLL on Windows, you must explicitly mark its declaration. The following code (MyDLL.h) provides a simple demonstration of this.

```
pch.h        framework.h        MyDLL.cpp        MyDLL.h  ⊟ ×  dllmain.cpp        物件瀏覽器
MyDLL                                    (全域範圍)
    1    // 下列 ifdef 區塊是建立巨集以協助從 DLL 匯出的標準方式。
    2    // 這個 DLL 中的所有檔案都是使用命令列中所定義 MYDLL_EXPORTS 符號編譯的。
    3    // 在命令列定義的符號。任何專案都不應定義這個符號
    4    // 這樣一來，原始程式檔中包含這檔案的任何其他專案
    5    // 會將 MYDLL_API 函式視為從 DLL 匯入的，而這個 DLL 則會將這些符號視為
    6    // 匯出的。
    7    #ifdef MYDLL_EXPORTS
    8    #define MYDLL_API __declspec(dllexport)
    9    #else
   10    #define MYDLL_API __declspec(dllimport)
   11    #endif
   12
   13    #ifdef __cplusplus
   14    extern "C" {
   15    #endif
   16
   17    MYDLL_API int __cdecl Add(int a, int b);
   18
   19    #ifdef __cplusplus
   20    }
   21    #endif
110 %
```
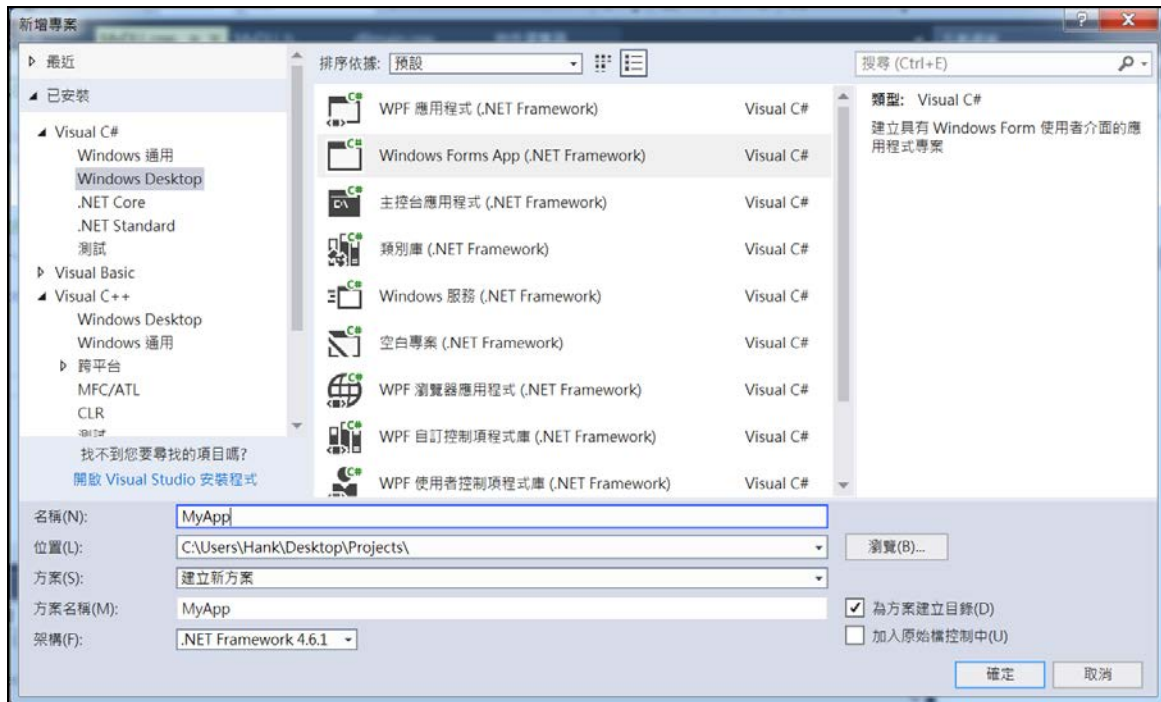
4. You can then add new or existing source files (MyDLL.cpp) to your project under the Source Files folder in the left-hand pane.

```
pch.h        framework.h        MyDLL.cpp  ⊟ ×  MyDLL.h        dllmain.cpp        物件瀏覽器
MyDLL                                    (全域範圍)
    1    // MyDLL.cpp : 定義 DLL 應用程式的匯出函式。
    2    //
    3    #include "pch.h"
    4    #include "MyDLL.h"
    5
    6
    7    // 這是匯出函式的範例。
    8    MYDLL_API int __cdecl Add(int a, int b)
    9    {
   10        return a+b;
   11    }
   12
```
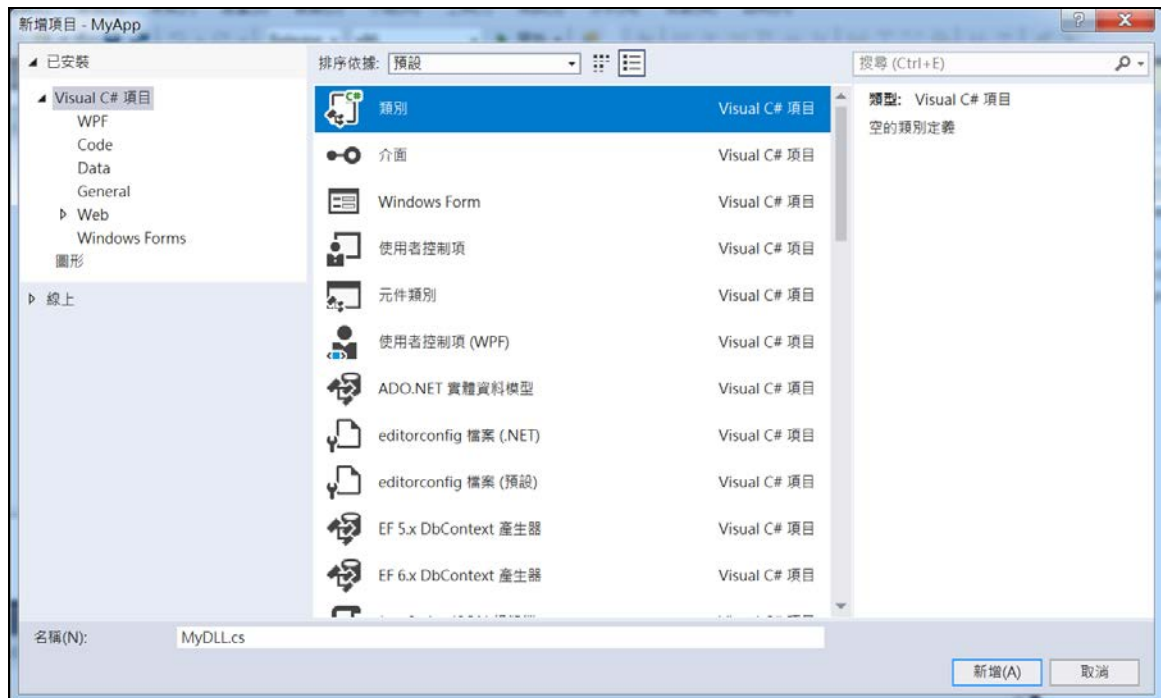
5. Build → Build Project (or Build MyDLL), then Visual Studio will generate a .dll file and an associated .lib import file.
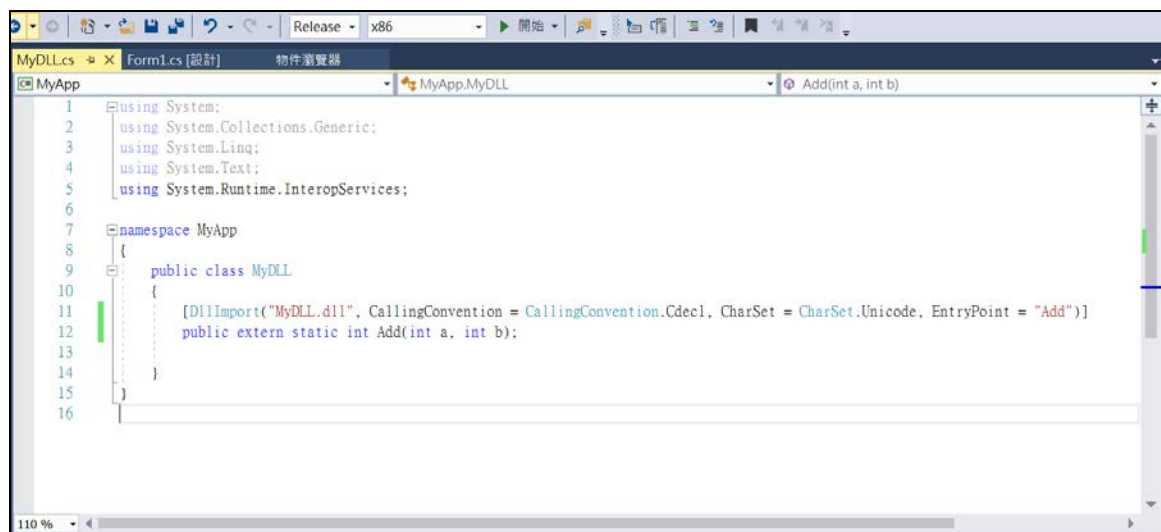
6. The man-machine interface, named MyApp, will be created by using C# to new a button and a textbox as follows.
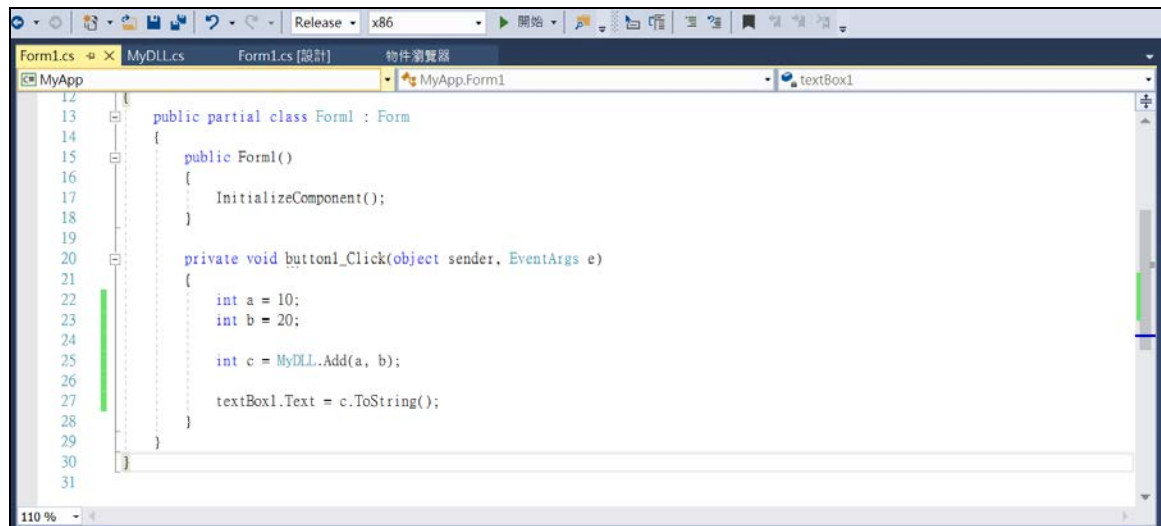
7. Add MyDLL.cs to your C# project.
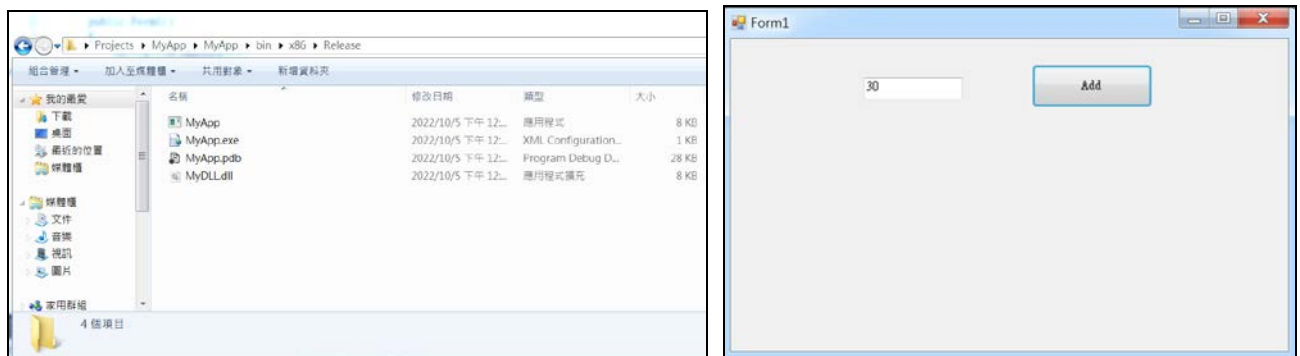

8. The details of MyDLL.cs are as follows.

9. Call the function in the event of button1_Click:

10. Build → Build MyApp, and copy the MyDLL.dll file built in step 5 to the folder of MyApp.exe. Now your MyApp.exe is executable.

Exercises:

1. Please add the functions of subtract, multiply and divide to MyDLL.dll.
2. Please add the entry point of the subtract, multiply and divide functions to MyDLL.cs.
3. Please add the button and textbox to the GUI.
4. Run the demo to verify your work.