



泓格  
PCIe-LM4

# 驅動函式庫使用手冊

繁體中文版

支援 64 位元作業系統

支援 Windows 10

## ► 免責聲明

---

凡使用泓格產品除產品品質造成的損害，泓格科技股份有限公司不承擔任何法律責任。泓格科技股份有限公司有義務提供正確及詳細的資料，但保留修改權利，且不承擔使用者非法利用資料對第三方所造成侵害構成的法律責任。

## ► 版權

---

版權所有 © 2020 泓格科技股份有限公司,保留所有權利。

## ► 商標

---

手冊中所涉及所有公司商標，商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

## 關於

---

本手冊說明如何透過泓格 **PCIELM4** 驅動函式庫在 **Windows** 下對泓格數據採集板卡作 **I/O** 操作。本手冊提供了使用 **PCIe-LM4** 的相關訊息。包含 **I/O** 操作的流程以及每個 **API** 函數的功能、參數、數據結構的說明。

使用者可以使用泓格 **PCIELM4** 函式庫驅動在 **Windows** 系統下使用 **VC**、**VB**、**VB.NET**、**C#.NET**、**VC.NET**、**Console** 等工具來實作開發。本手冊也提供範例程式，利用開發實例向使用者說明如何使用泓格 **PCIELM4** 函式庫，提供給使用者參考進行應用開發。

如有本手冊未涵蓋之內容請來信諮詢泓格技術工程師。

Email: [service@icpdas.com](mailto:service@icpdas.com)

# Table of Contents

# Table Of Contents

Industrial Communication Products

Table of Contents .....	3
<b>1. 導讀 .....</b>	<b>6</b>
1.1. 關於泓格 PCIELM4 驅動程式 .....	7
1.2. 支援的泓格產品 .....	7
1.3. 系統需求 .....	8
<b>2. 開始安裝使用 .....</b>	<b>9</b>
2.1. 取得 PCIELM4 驅動函式庫安裝程式 .....	10
2.2. 安裝 PCIELM4 驅動程式函式庫 .....	11
2.3. 移除 PCIELM4 驅動函式庫 .....	14
<b>3. 開發指南 .....</b>	<b>15</b>
3.1. 應用程式架構 .....	16
3.2. 在 Win32 Console .....	17
3.3. 在 Visual Basic 6.0 .....	20
3.4. 在 Visual C++.NET .....	23
3.5. 在 Visual Basic.NET .....	29
3.6. 在 Visual C#.NET .....	35
3.7. 範例程式及文件 .....	41
<b>4. 函式應用 .....</b>	<b>42</b>
4.1. 導讀 .....	43
4.2. 驅動函式庫 .....	45
4.3. 數位輸出輸入 .....	47
4.3.1. 數位輸入 .....	47
4.3.2. 數位輸出 .....	48
4.4. 類比輸入 .....	49

# Table Of Contents

Industrial Communication Products

4.5. 類比輸出.....	50
<b>5. 函式參考.....</b>	<b>51</b>
5.1. 函式介紹.....	52
5.1.1. 驅動函式集.....	53
PCIELM4_DriverInit .....	53
PCIELM4_DriverClose .....	53
PCIELM4_GetCardInfo .....	54
5.1.2. 數位輸出輸入函式集.....	55
PCIELM4_ReadDI.....	55
PCIELM4_WriteDO .....	56
5.1.3. 類比輸入函式集.....	57
PCIELM4_ConfigAI .....	57
PCIELM4_PollingAI.....	59
PCIELM4_PollingAIH .....	60
PCIELM4_AIHex2Vol.....	61
PCIELM4_ConfigAutoZero.....	62
PCIELM4_SaveAIAutoZeroVal .....	63
5.1.4. 類比輸出函式集.....	65
PCIELM4_ConfigAO .....	65
PCIELM4_WriteAOVoltage .....	66
PCIELM4_WriteAOVoltageH.....	67
PCIELM4_StartAOVoltageALL.....	68
PCIELM4_StartAOVoltageALLH.....	69
PCIELM4_StopAOALL.....	71
5.2. 資料型態.....	72
PPCIELM4_DEVICE_INFO.....	72
PPCIELM4_CARD_INFO.....	74

附錄 A. 函式回傳值與配置碼 ..... 76

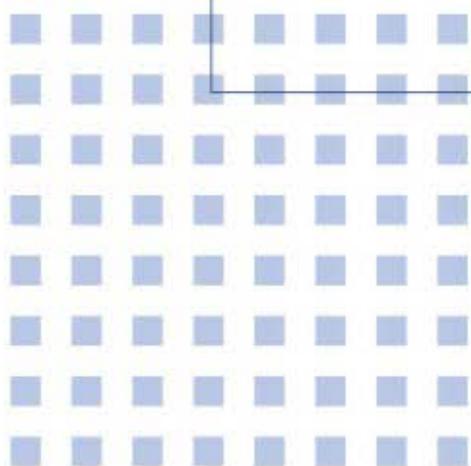
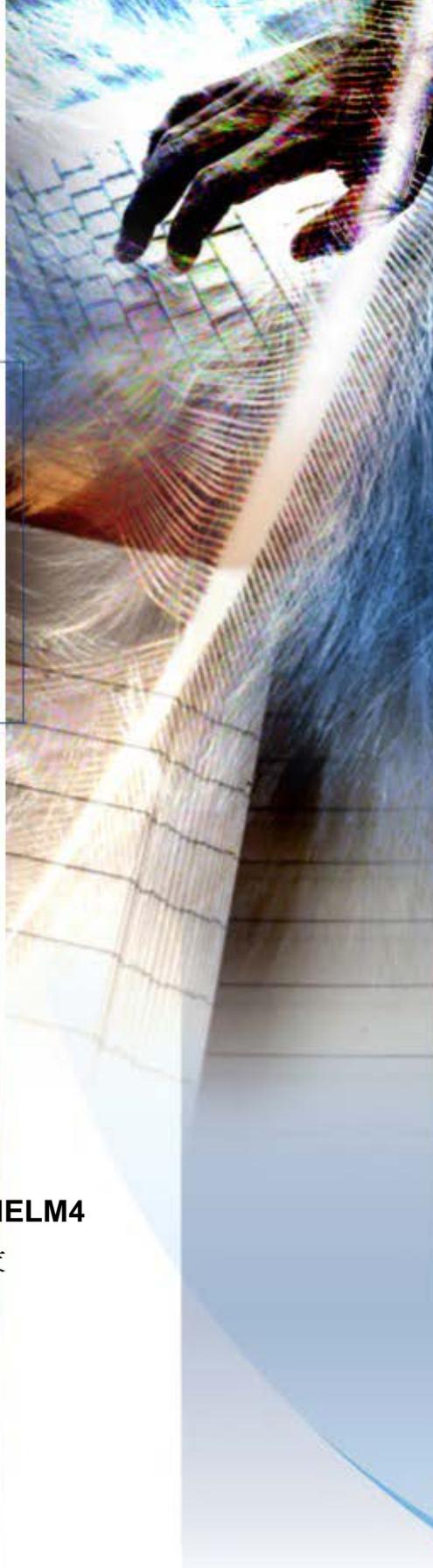
A.1. 函數回傳值定義.....	77
A.2. 模組識別號碼.....	80
A.3. 配置碼定義.....	81
A.3.1. 類比輸入配置碼.....	81
A.3.2. 類比輸出配置碼(電壓).....	81
A.4. 數位輸入埠定義號碼.....	82
A.5. 數位輸出埠定義號碼.....	82

附錄 B. 其他..... 83

B.1. 常見問題集.....	84
B.2. 版本修改資訊.....	86

# Table Of Contents

Industrial Communication Products



# 1. 導讀

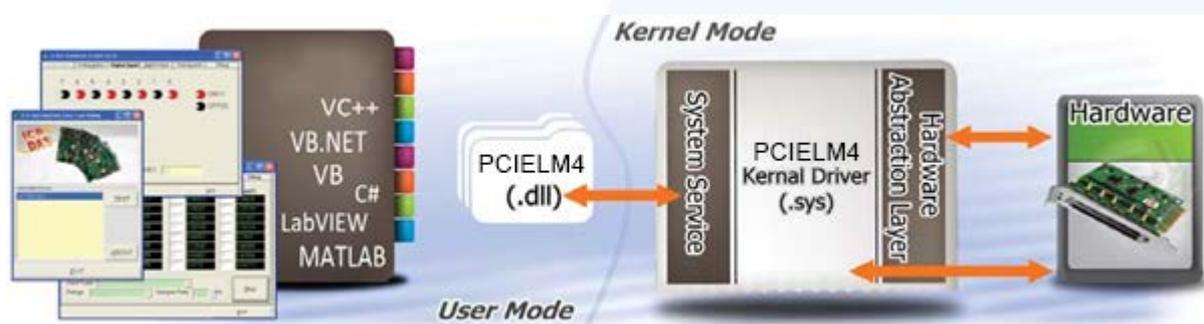
本章節將會簡單介紹泓格 **PCIELM4** 驅動程式庫的功能及系統需求

## 1.1. 關於泓格 PCIELM4 驅動程式

泓格 PCIELM4 驅動函式庫提供完整的硬體函式以及最優良效能。在泓格的 PCIELM4 驅動函式庫裡，不需使用特定的硬體暫存器命令，PCIELM4 提供許多強而有力的函式讓泓格板卡的使用者可以在各種編程語言與環境下開發。

泓格 PCIELM4 驅動函式庫軟體使用者直接 I/O 大幅減低 API 對硬體作 I/O 的時間來達到更好的 I/O 速度。

泓格 PCIELM4 驅動函式庫支援 Windows 2000 之後所有的 32 位元及 64 位元作業系統，使用者就不再需要擔心作業系統的相容性。



## 1.2. 支援的泓格產品

下表泓格驅動函式庫所支援的產品：

型號
PCIe-LM4

表格 1-1 產品支援列表

## 1.3. 系統需求

如果您想在電腦上使用泓格驅動程式庫，以下是一些系統需求：

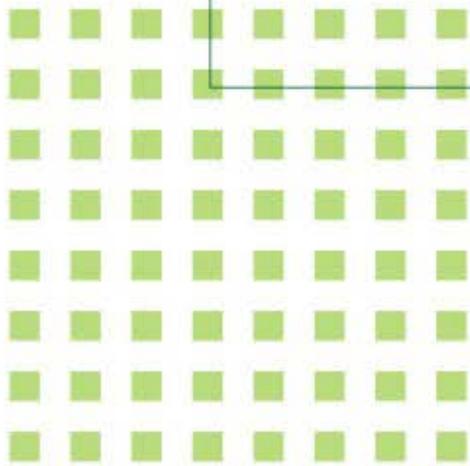
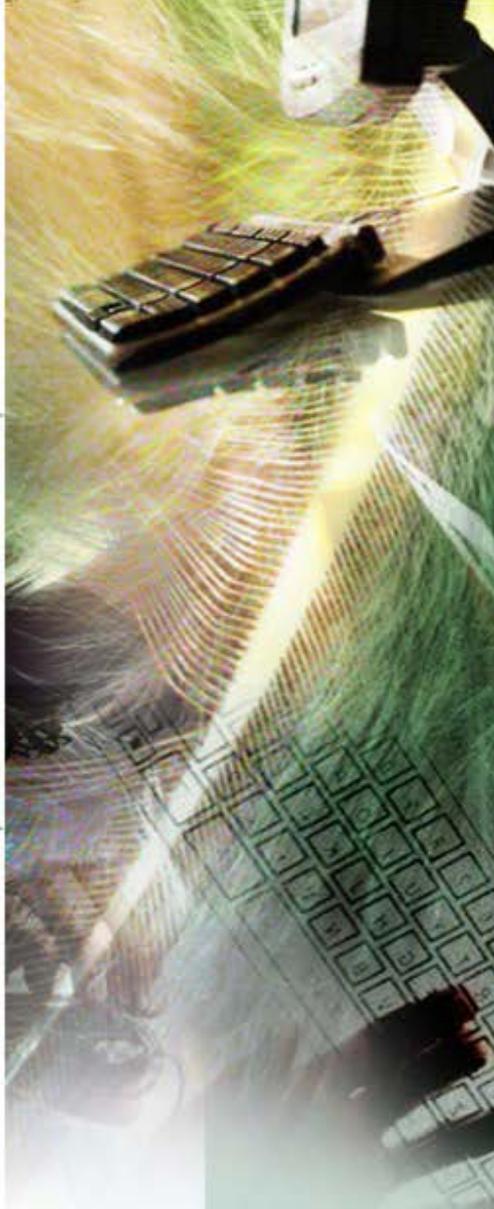
- 使用 266MHz 或更快的 32 位元(x86)或 64 位元(x64)處理器
- 至少 64 MB 的記憶體空間
- 相容於 VGA 的圖型顯示卡
- 至少 20 MB 磁碟空間
- 一台相容 DVD/CD-ROM
- Microsoft Windows 2000 以上的 32 位元或 64 位元作業系統

支援以下的 32 及 64 位元的 Windows 作業系統

32 位元(x86)	64 位元(x64)
Windows 2000	-
Windows XP	Windows XP
Windows Server 2003	Windows Server 2003
Windows Vista	Windows Vista
Windows Server 2008	Windows Server 2008
Windows 7	Windows 7
-	Windows Server 2012
Windows 8/8.1	Windows 8/8.1
Windows 10	Windows 10

表格 1-2 支援作業系統列表

註：不支援 Microsoft Windows 3.1/95/98/ME/NT



## 2. 開始安裝使用

本章節以圖解及簡易的文字引導使用者如何安裝及移除驅動程式

## 2.1. 取得 PCIELM4 驅動函式庫安裝程式

取得 PCIELM4 驅動函式庫安裝程式的方法可以從官方網路上下載，取得路徑請參考下表：



<http://www.icpdas.com/en/download/index.php?model=PCIe-LM4>

## 2.2. 安裝 PCIELM4 驅動程式函式庫

### 步驟一 安裝資料擷取板卡

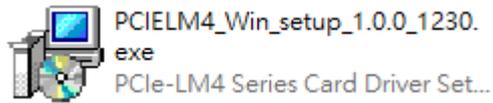
請依照下列步驟安裝板卡：

- 1 關掉電腦電源
- 2 打開電腦機殼
- 3 將 I/O 板卡插入至一個未使用的 PCI 或 PCIe 插槽
- 4 裝上機殼
- 5 重新啟動電源

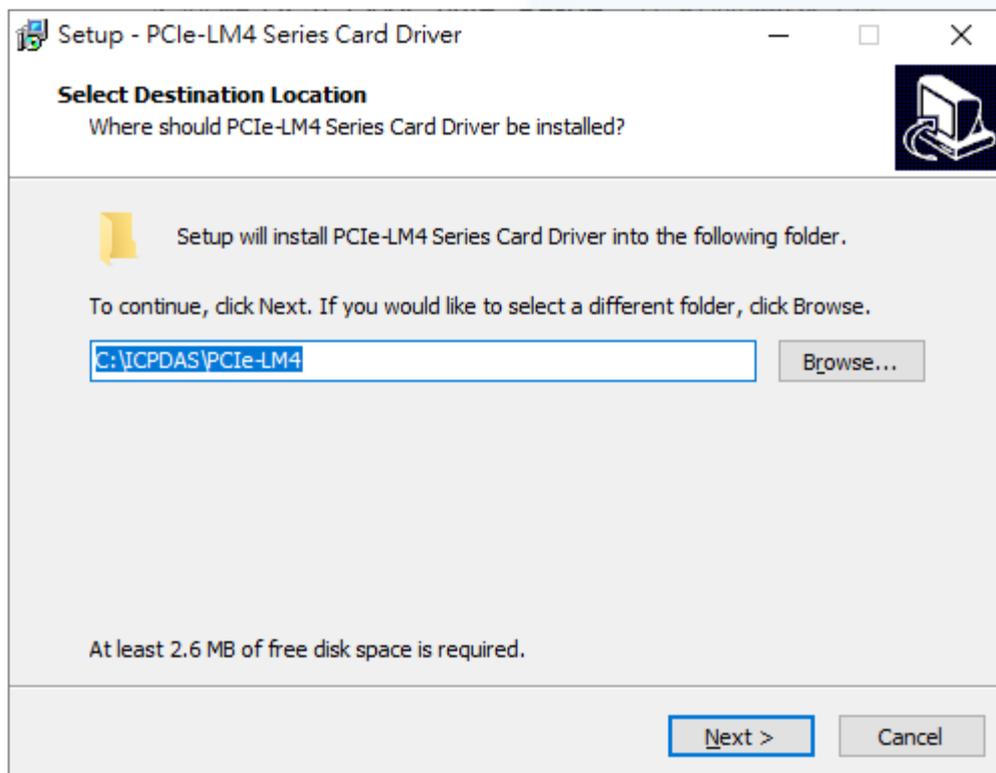
## 步驟二 安裝驅動程式及函式庫

請依照下列步驟執行安裝

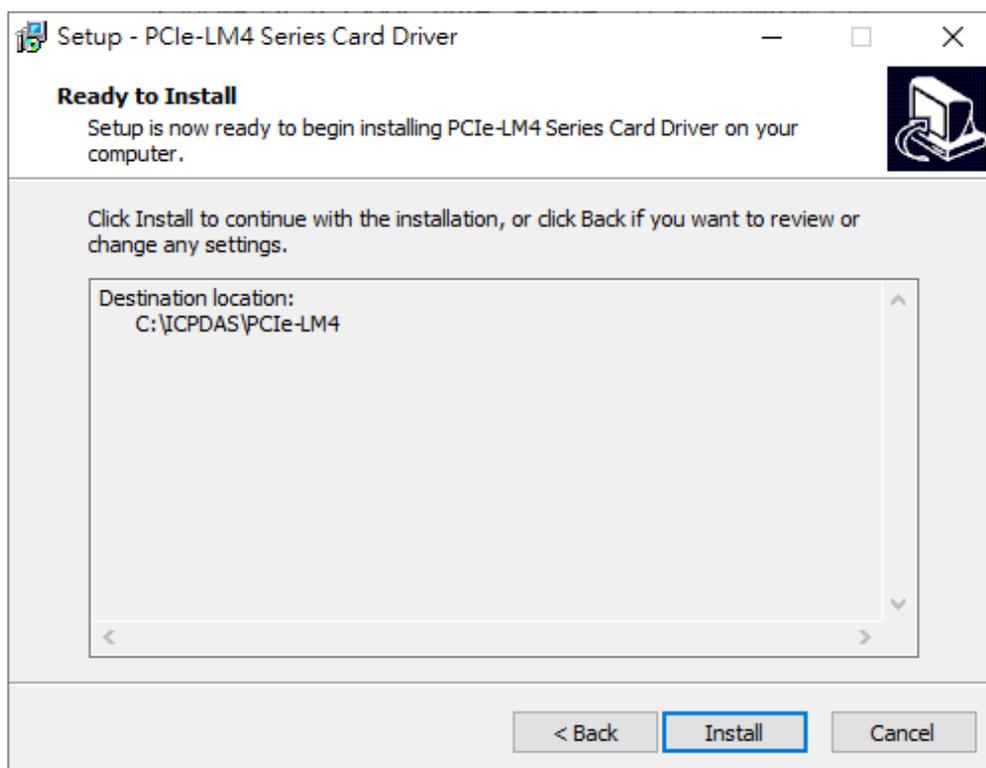
1. 雙擊 PCIELM4\_Win\_setup... 安裝驅動函式庫。



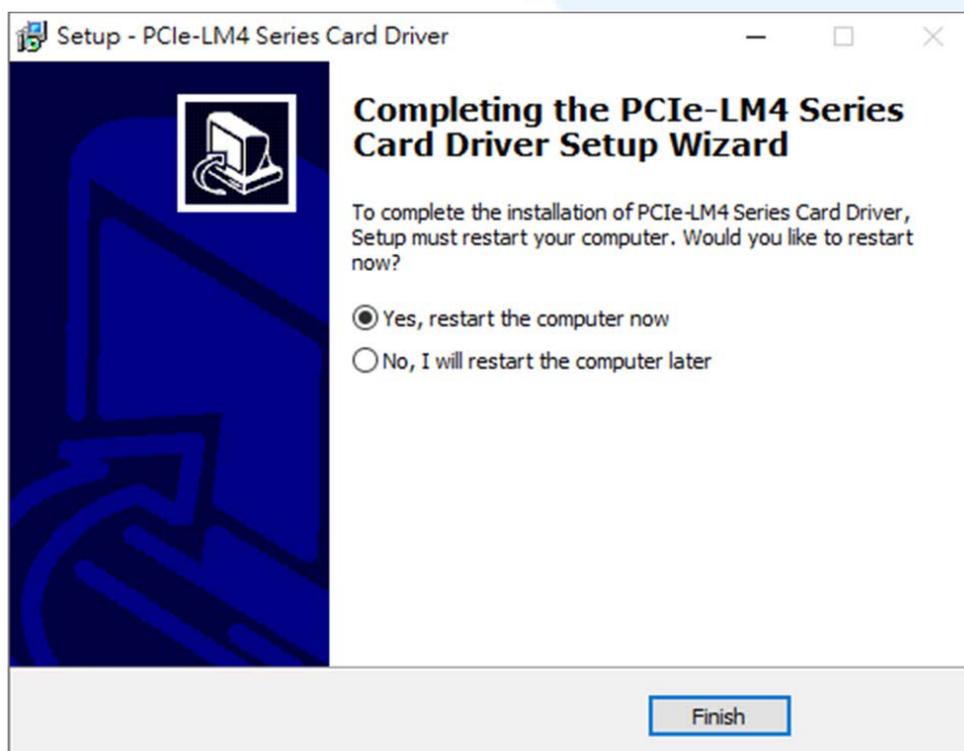
2. 選擇安裝目錄，預設為 C:\ICPDAS\PCIe-LM4，確認後按 **Next>** 到下一個畫面。



### 3.按 Install。



4. 選擇 Yes，restart the computer now 後，按下 Finish 按鍵後，系統會自動重新開機，在重新開機之後，泓格 PCIELM4 驅動函式庫安裝完成。



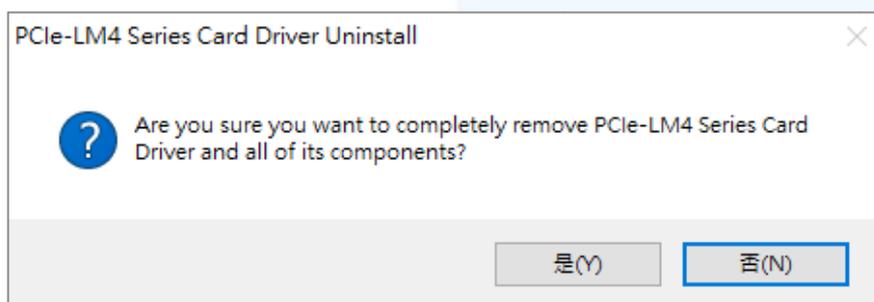
## 2.3. 移除 PCI-ELM4 驅動函式庫

泓格驅動函式庫包括反安裝工具來協助您從電腦上移除軟體，如果您想要移除軟體請完成下列的流程來執行反安裝工具。

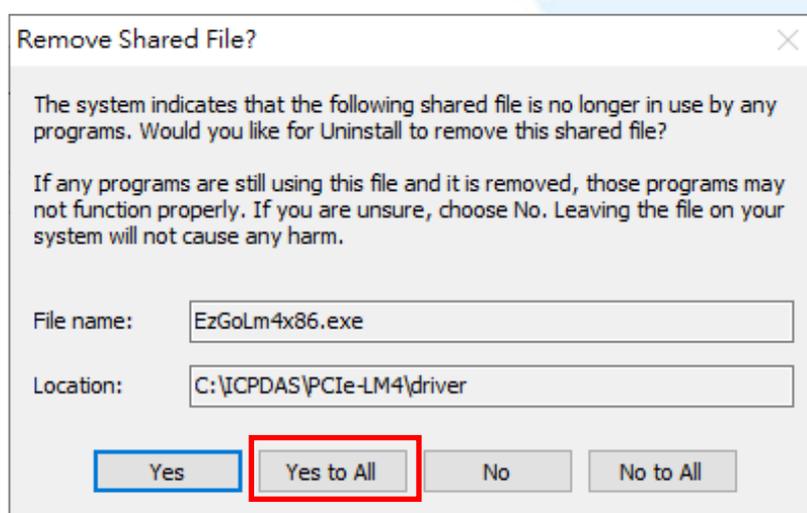
1. 至設定|控制台|新增或移除程式下。
2. 在選單列表上選擇 **PCIe-LM4 Card Windows Driver** 項目，並點擊解除安裝。

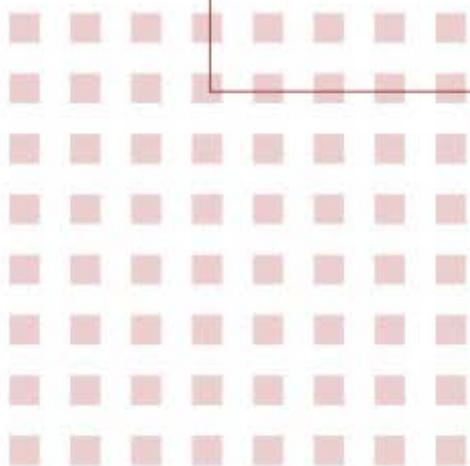


3. 將會跳出一個對話框，並選擇是(Y)開始執行反安裝。



4. 點擊 **Yes to All** 完全移除 **PCI-ELM4.dll** 檔案，之後將會完成移除軟體的動作。

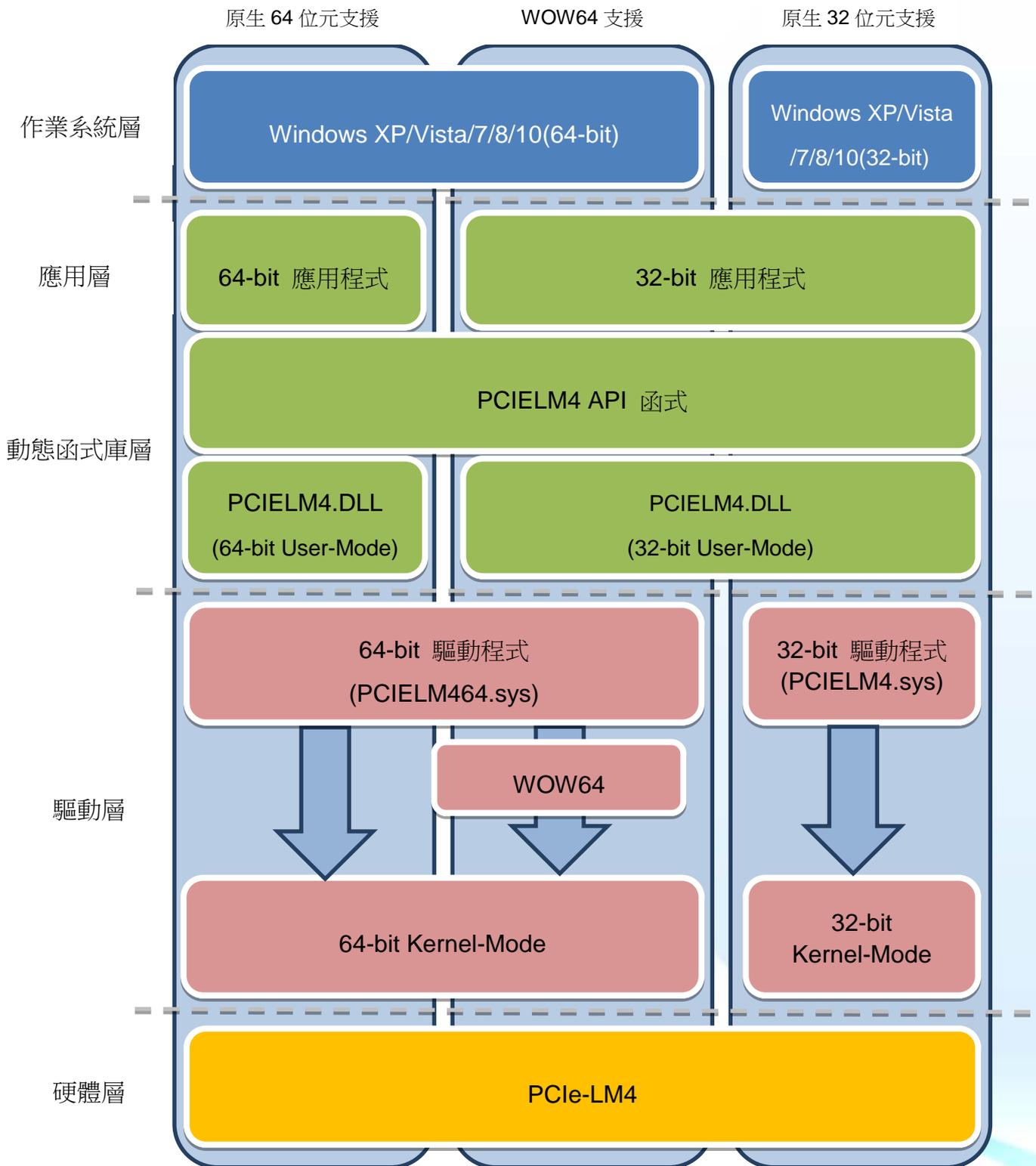




### 3. 開發指南

引導使用者去建構簡單應用程式。並且提供在 Win32 Console， VB6， Visual Studio.NET 及 Visual Studio.NET x64 環境下逐步編寫程式的範例。

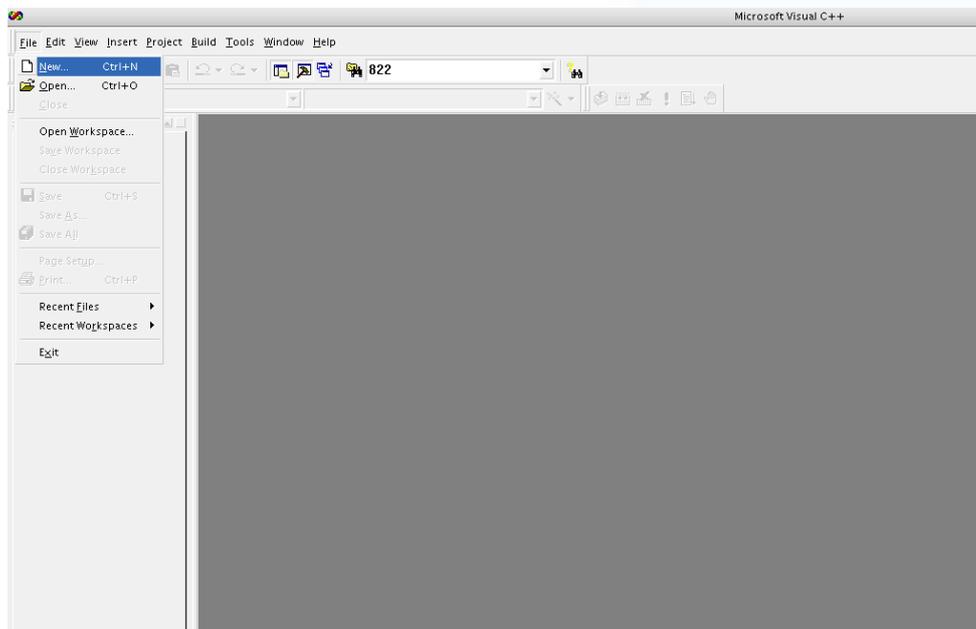
### 3.1. 應用程式架構



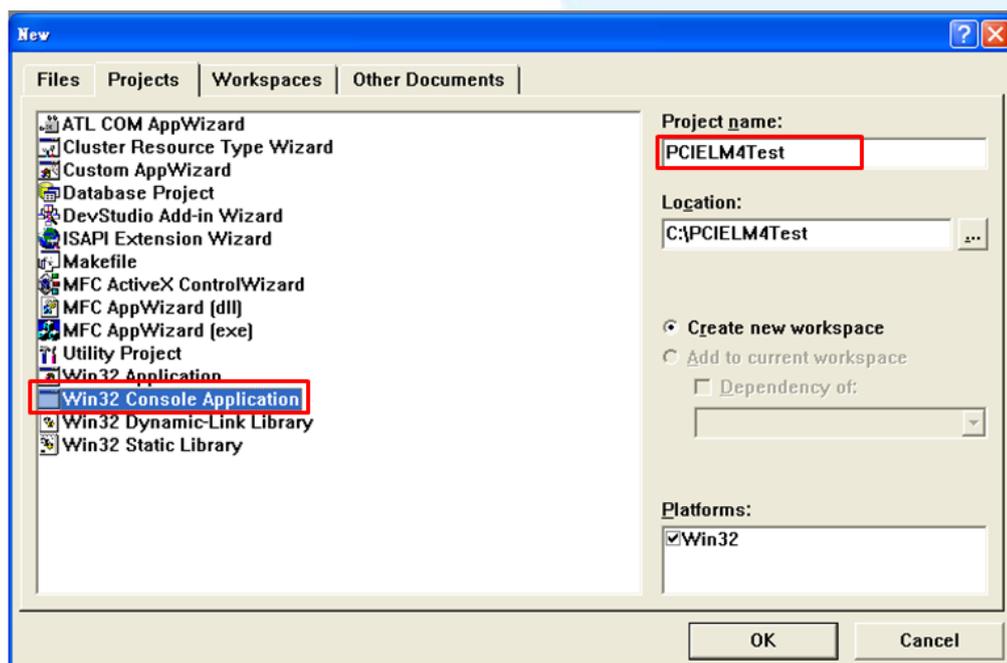
## 3.2. 在 Win32 Console

### 步驟 1: 撰寫應用程式

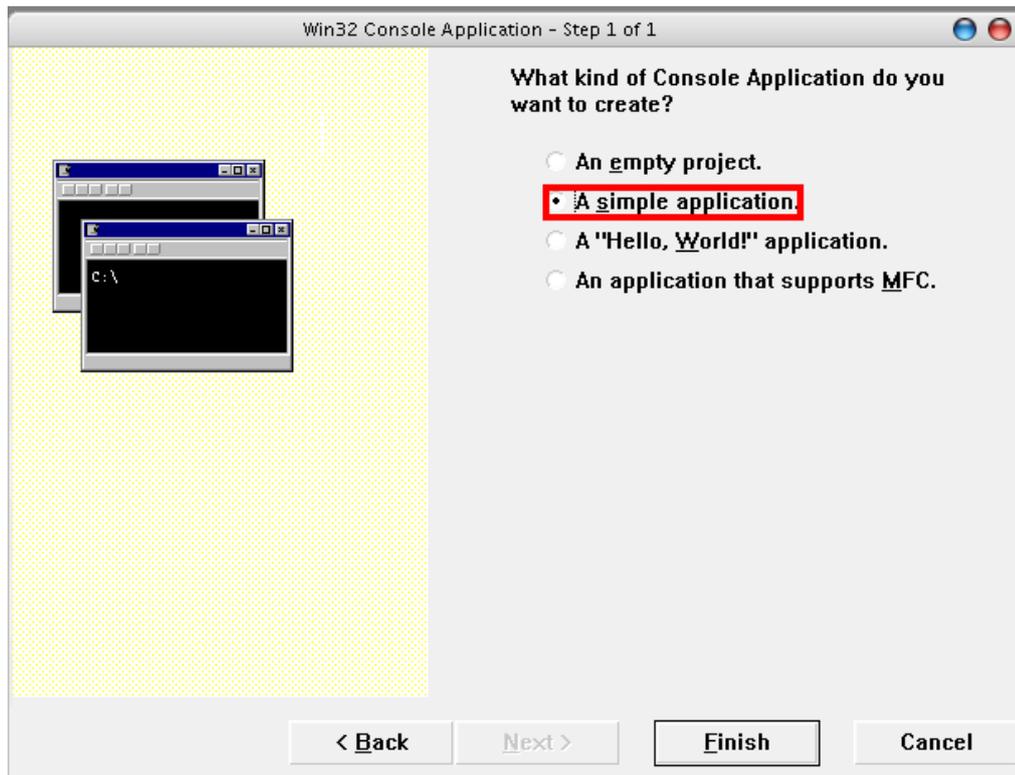
1. 至程式集開啟 Microsoft Visual C++ 6.0
2. 從主要選單內選擇 File|New...



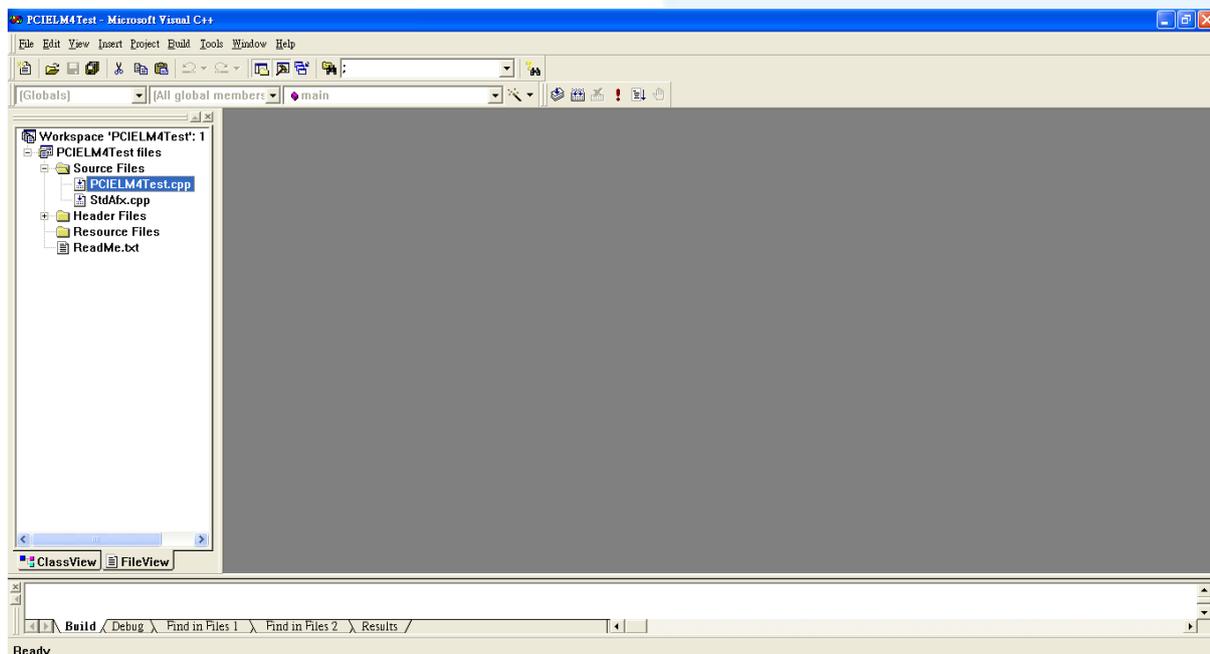
3. 在 dialog box 下的列表點選項目 Win32 Console Application ，並在 Project name 欄位輸入 PCIELM4Test ，然後按下 OK 按鈕。



4. 點選 A simple application 後，按下 Finish 鍵後，將會產生為使用者產生最基本的程式碼。



5. 雙擊 PCIELM4Test.cpp 開啟程式碼寫入視窗。



6. 在 PCIELM4Test.cpp 填寫程式碼如下：

```
#include "stdafx.h"
#include "stdio.h"
#include "PCIELM4.h" //Include the PCIELM4 header file
#pragma comment(lib,"PCIELM4.lib") //Include the PCIELM4 library file

WORD wRtn;
WORD wBoardNo;
WORD wTotalBoards;

int main(int argc, char* argv[])
{
    //Initial the resource and get total board number form Driver
    wRtn=PCIELM4_DriverInit(&wTotalBoards);
    if (wRtn!= PCIELM4_NoErr)
    {
        printf("\nDriver Init Error(%d)",wRtn);
        return wRtn;
    }
    printf("Write DO Value 0xFF");
    wBoardNo=0;
    wOutPortNo=0;
    //Write DO
    wRtn = PCIELM4_WriteDO(wBoardNo,0xFF);
    //Release the resource from driver
    wRtn = PCIELM4_DriverClose();
    return 0;
}
```

## 步驟 2:測試應用程式

1. 在 Build 選單點擊 Compiler 來編譯程式碼。
2. 立即在 DOS Box 下執行程式。

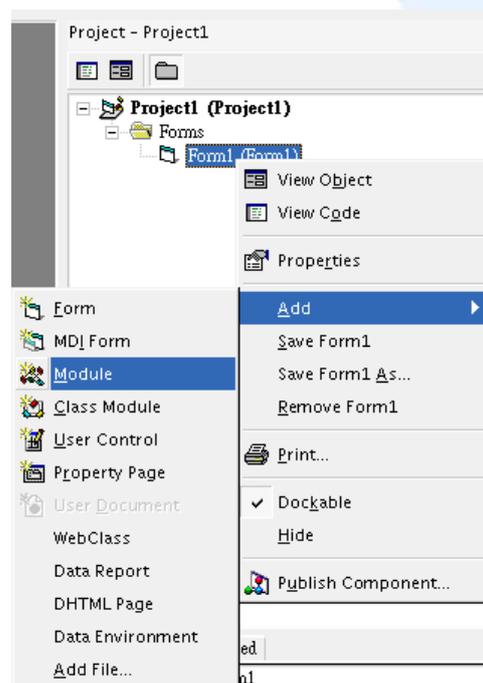
## 3.3. 在 Visual Basic 6.0

### 步驟 1: 撰寫應用程式

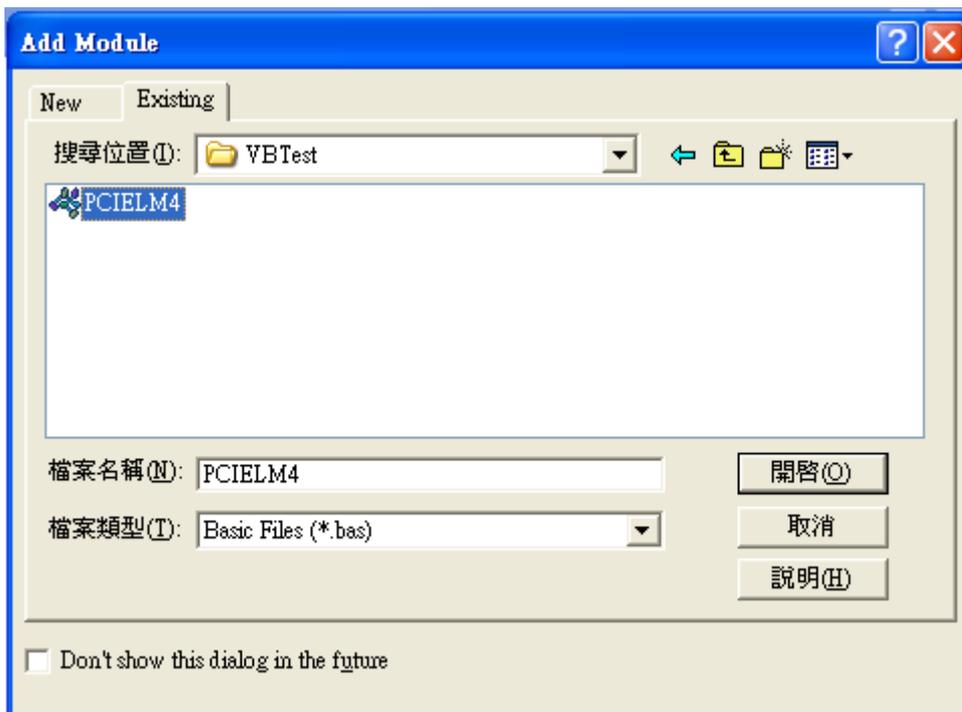
1. 至程式集開啟 Microsoft Visual Basic 6.0
2. 選擇 Standard EXE 圖示並按下開啟按鈕後將建立一個新的專案。



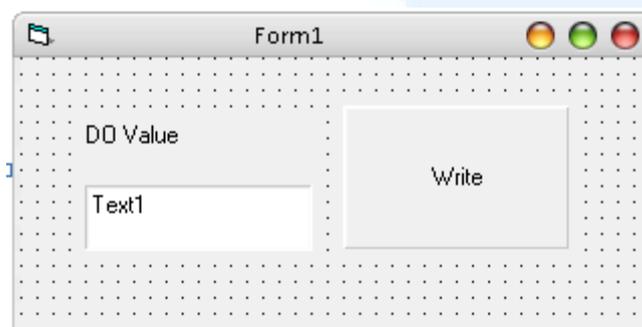
3. 在 Project explorer 開啟 Add Module 視窗。



4. 在 Add Module 選擇 Existing 後添加宣告檔 PCIELM4.bas 至專案裡。



5. 設計視窗，在 form1 放置一個 Label 控制項並在 Caption 屬性上輸入 DO Value。接著放置 TextBox 控制項，並切換至屬性視窗上至 Name 屬性輸入 txtDOVal，最後放置一個 CommandButton 控制項，並修改 Name 屬性為 cmdWrite 及在 Caption 屬性上輸入 Write。



6. 在 cmdWrite 填寫程式碼如下：

```
Option Explicit
Dim wTotalBoards As Integer
Dim wBoardNo As Integer
Dim wRtn As Integer

Private Sub cmdWrite_Click()

Dim wBoardIndex As Integer

'//Initial the resource and get total board number form Driver
wRtn = PCIELM4_DriverInit(wTotalBoards)
If (wRtn) Then
    MsgBox ("Driver Initial Error!!Error Code:" + Str(wRtn))
    End
End If

wBoardNo =0
wOutportNo =0

'//Write DO
wRtn = PCIELM4_WriteDO(wBoardNo, Val(txtDOVal.Text))

'//Release the resource form Driver
wRtn = PCIELM4_DriverClose()
End Sub
```

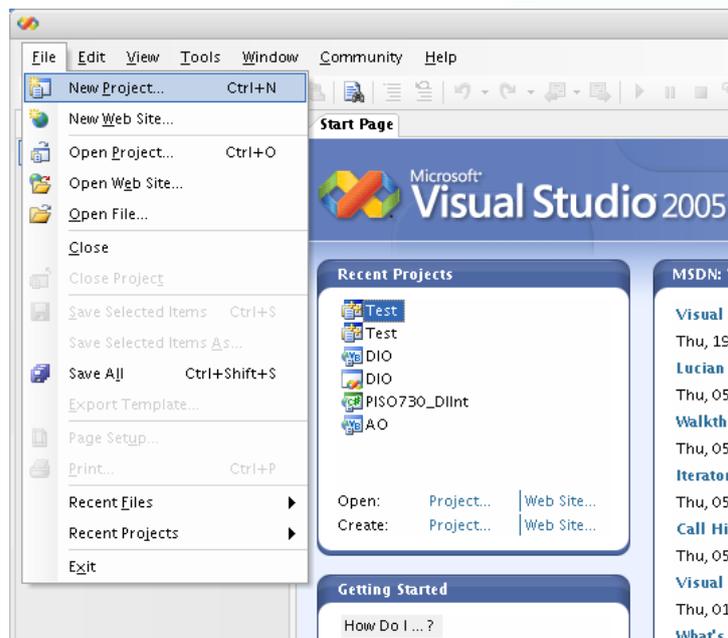
**步驟 2:**測試應用程式

1. 按下 **F5** 來執行程式。
2. 在 **DO Value** 欄位輸入數值 **255**。
3. 並按下 **Write** 按鍵，輸出 **DO** 數值 **255**。

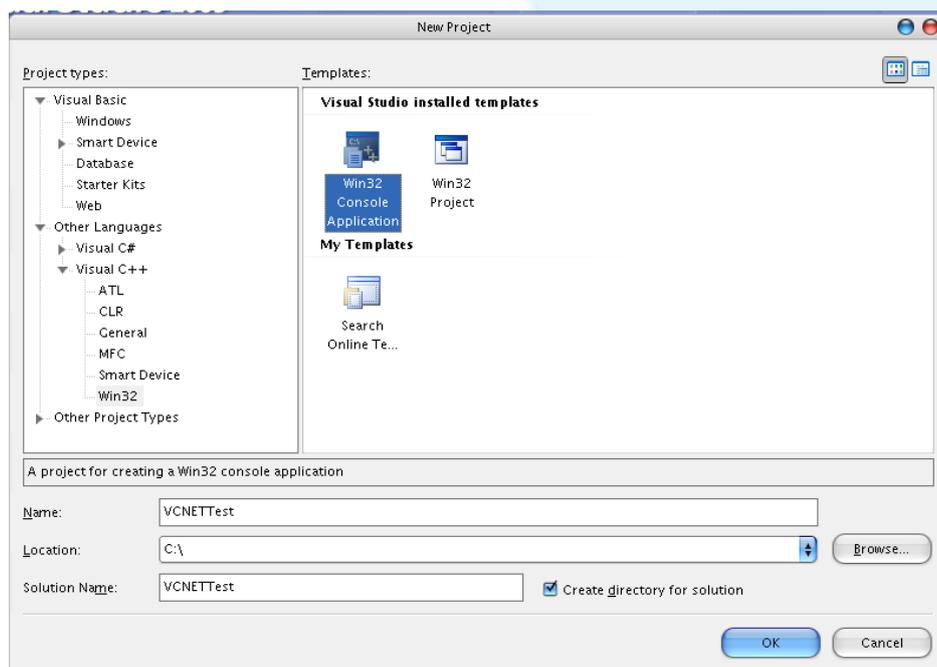
## 3.4. 在 Visual C++.NET

### 步驟 1: 撰寫應用程式

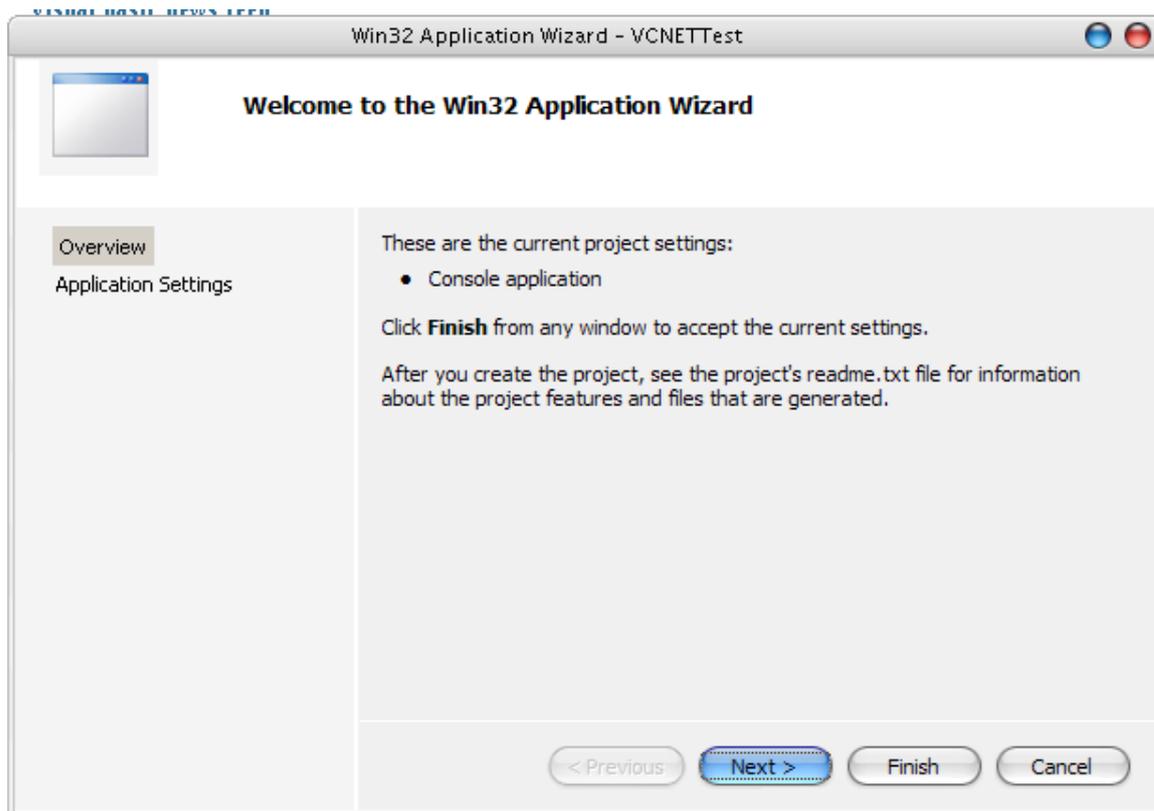
1. 至程式集開啟 Microsoft Visual Studio 2005
2. 從主要選單內選擇 File|New Project...



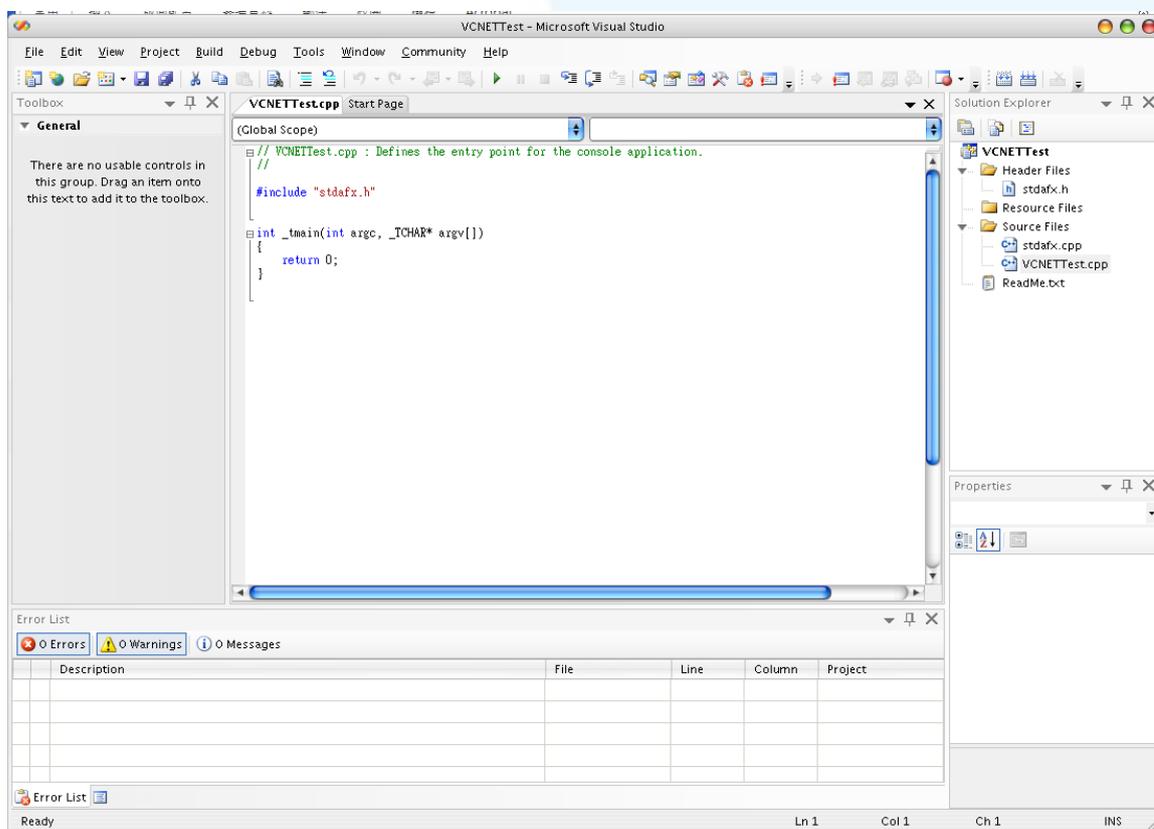
3. 在 Project type 下的列表點選項目 Visual C++ 並在展開選單內選擇 Win32，然後再右方 Templates 表框內選擇 Win32 Console Application，接下來至下方的 Name 鍵入專案名稱 VCNETTest 然後按下 OK 按鍵。



4. 按下 **Finish** 鍵後，將會產生為使用者產生最基本的程式碼。



5. 雙擊 VCNETTest.cpp 開啟程式碼寫入視窗。



6. 在 VCNETTest.cpp 填寫程式碼如下：

```
// VCNETTest.cpp : Defines the entry point for the console application.
//

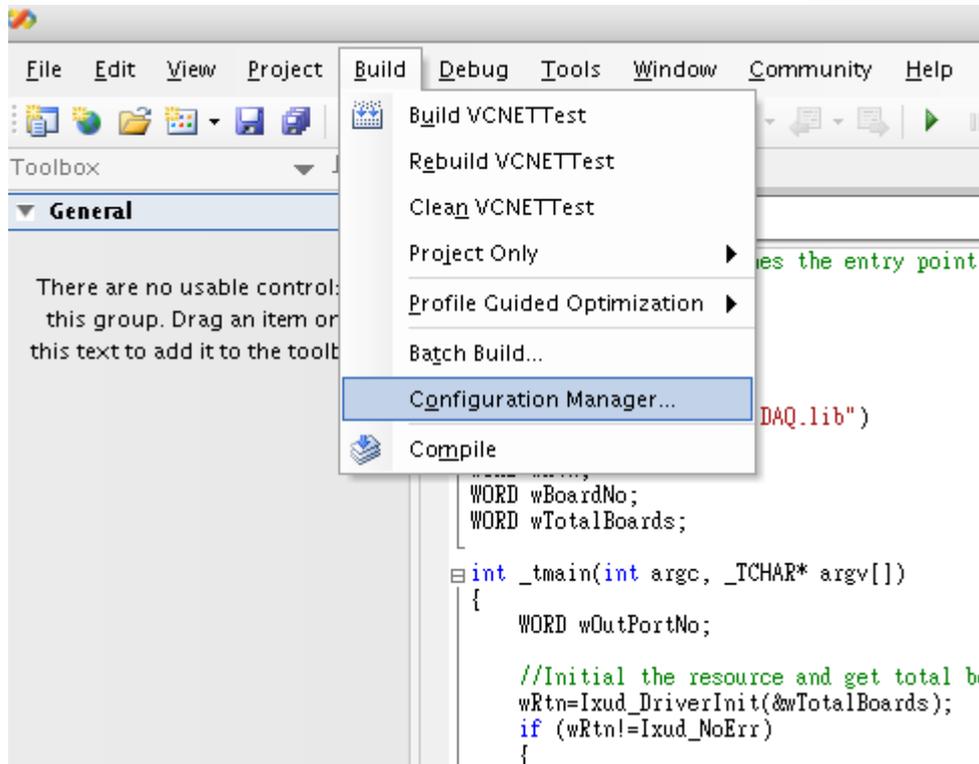
#include "stdafx.h"
#include "stdio.h"
#include "PCIELM4.h"
#pragma comment(lib,"PCIELM4.lib")

WORD wRtn;
WORD wBoardNo;
WORD wTotalBoards;

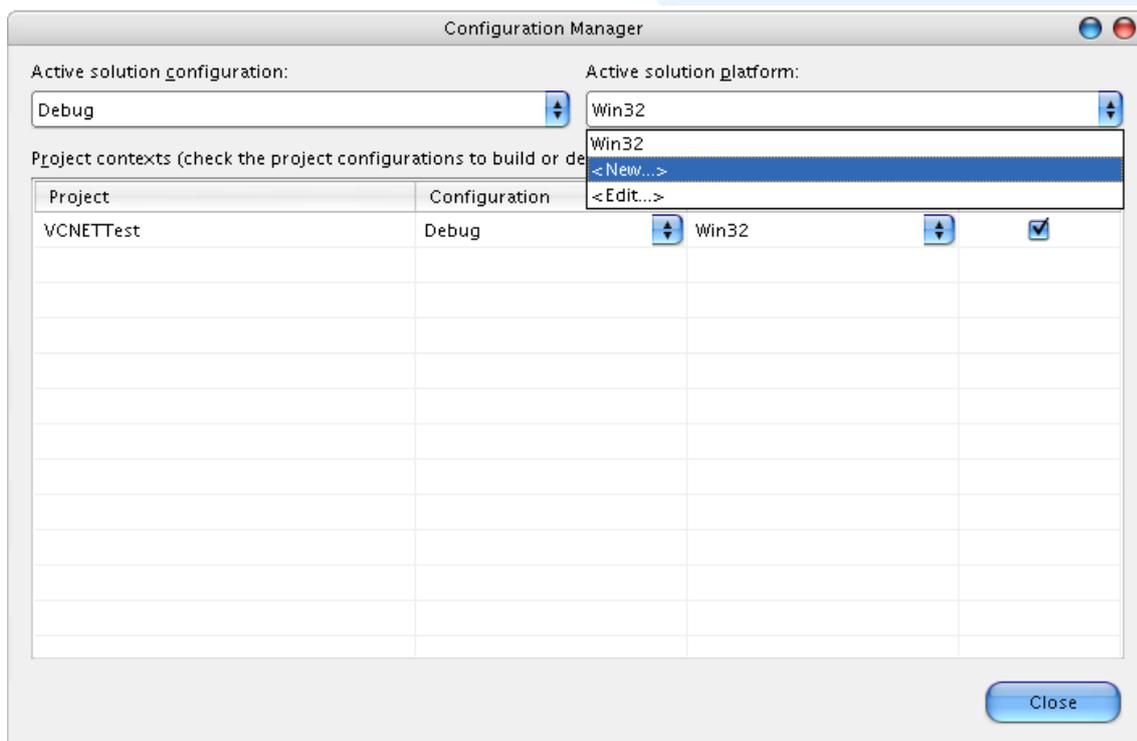
int _tmain(int argc, _TCHAR* argv[])
{
    //Initial the resource and get total board number form Driver
    wRtn=PCIELM4_DriverInit(&wTotalBoards);
    if (wRtn!=PCIELM4_NoErr)
    {
        printf("\nDriver Init Error(%d)",wRtn);
        return wRtn;
    }
    printf("Write DO Value 0xFF");
    wBoardNo=0;
    wOutPortNo=0;
    //Write DO
    wRtn = PCIELM4_WriteDO(wBoardNo,0xFF);
    //Release the resource from driver
    wRtn = PCIELM4_DriverClose();
    return 0;
}
```

## 步驟 2:編譯應用程式

1. 在 Build 選單點擊 Configuration Manager。

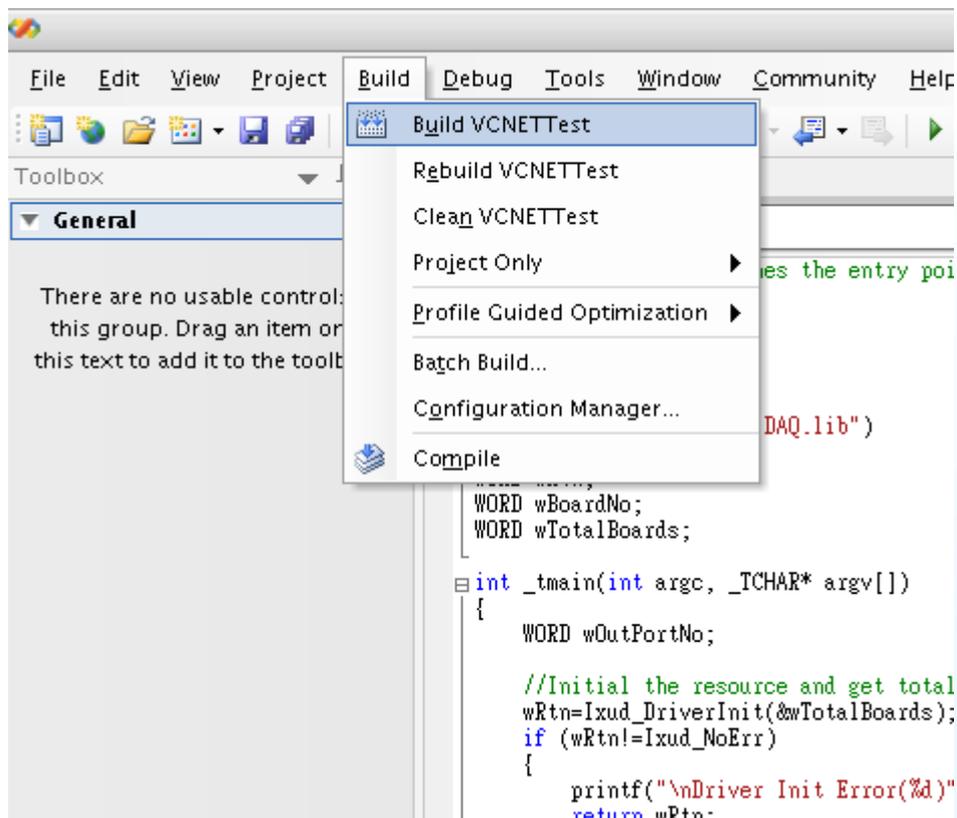


2. 在 Configuration Manager 點擊 Active solution Platform 下拉選單選擇<New...>。





5. 從主要選單內選擇 Build|Build VCNETTest 開始編譯應用程式。



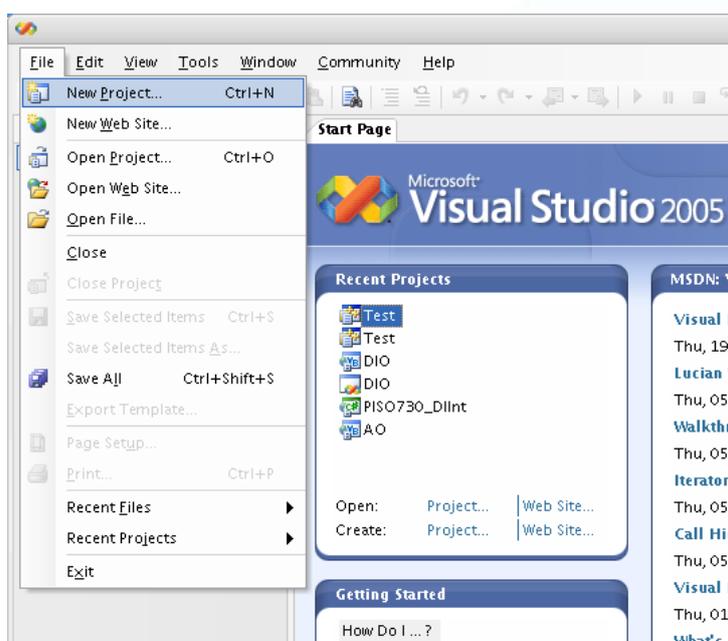
### 步驟 3: 測試應用程式

立即在 DOS Box 下執行程式。

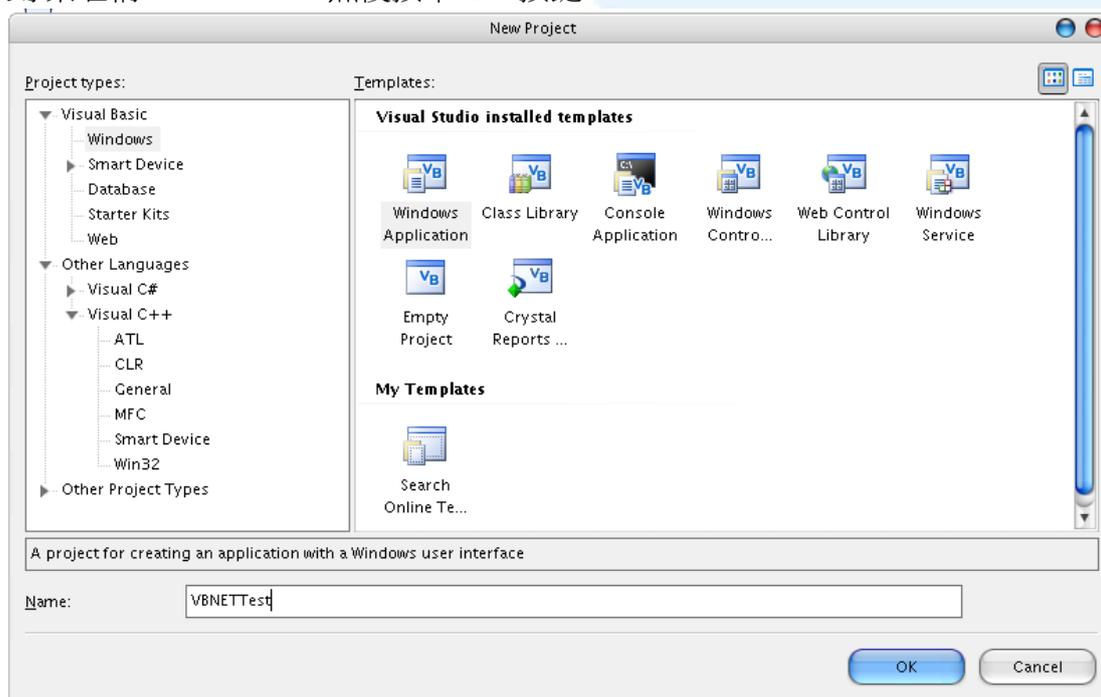
## 3.5. 在 Visual Basic.NET

### 步驟 1: 撰寫應用程式

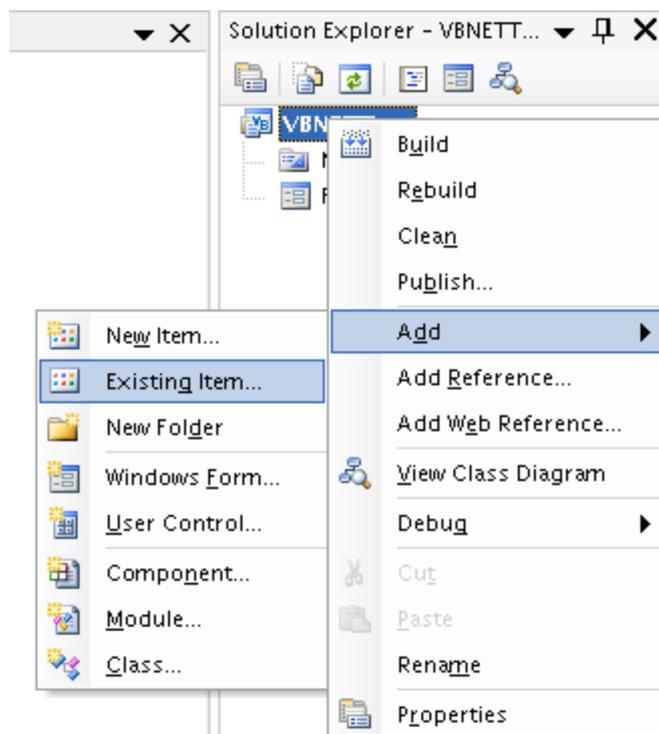
1. 至程式集開啟 Microsoft Visual Studio 2005
2. 從主要選單內選擇 File|New Project...



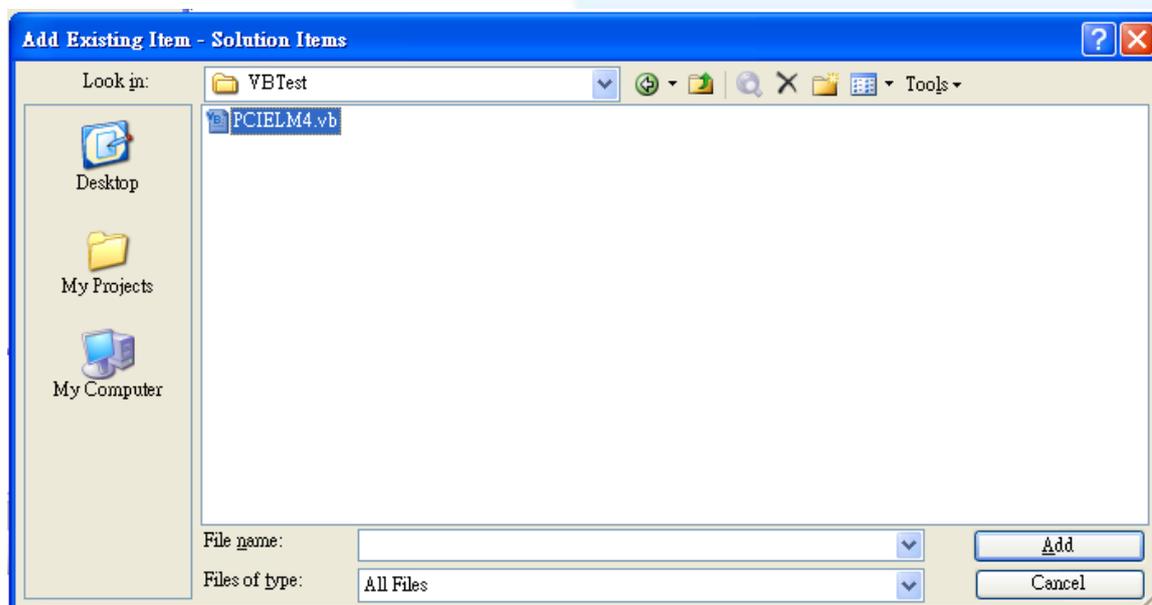
3. 在 Project type 下的列表點選項目 Visual Basic 並在展開選單內選擇 Windows，然後再右方 Templates 表框內選擇 Windows Application，接下來至下方的 Name 鍵入專案名稱 VBNETTest 然後按下 OK 按鍵。



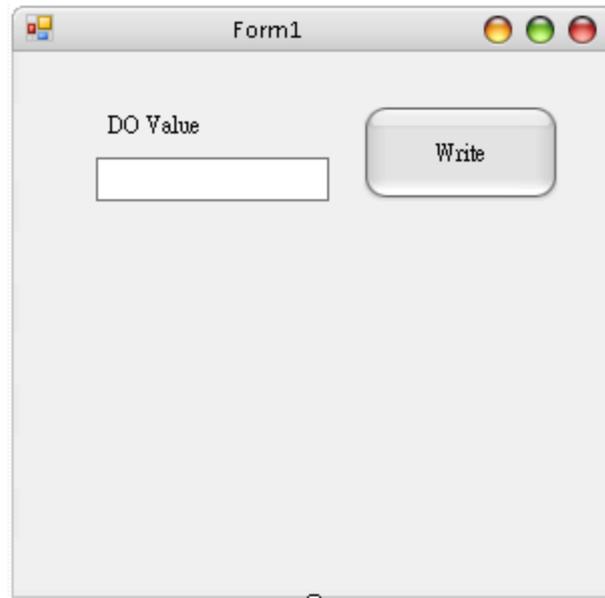
4. 在 Solution Explorer 開啟 Add Existing item 視窗。



5. 在 Add Existing item 選擇 PCIELM4.vb 後按 Add 按鍵添加宣告檔至專案裡。



- 設計視窗，在 Form1 放置一個 Label 控制項並在 Text 屬性上輸入 DO Value。接著放置 TextBox 控制項，並切換至屬性視窗上至 Name 屬性輸入 txtDOVal，最後放置一個 Button 控制項，並修改 Name 屬性為 btnWrite 及在 Text 屬性上輸入 Write。



- 在 btnWrite 填寫程式碼如下：

```
Private Sub btnWrite_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnWrite.Click
    Dim wTotalBoards As UInteger
    Dim wBoardNo As UInteger
    Dim wRtn As UInteger

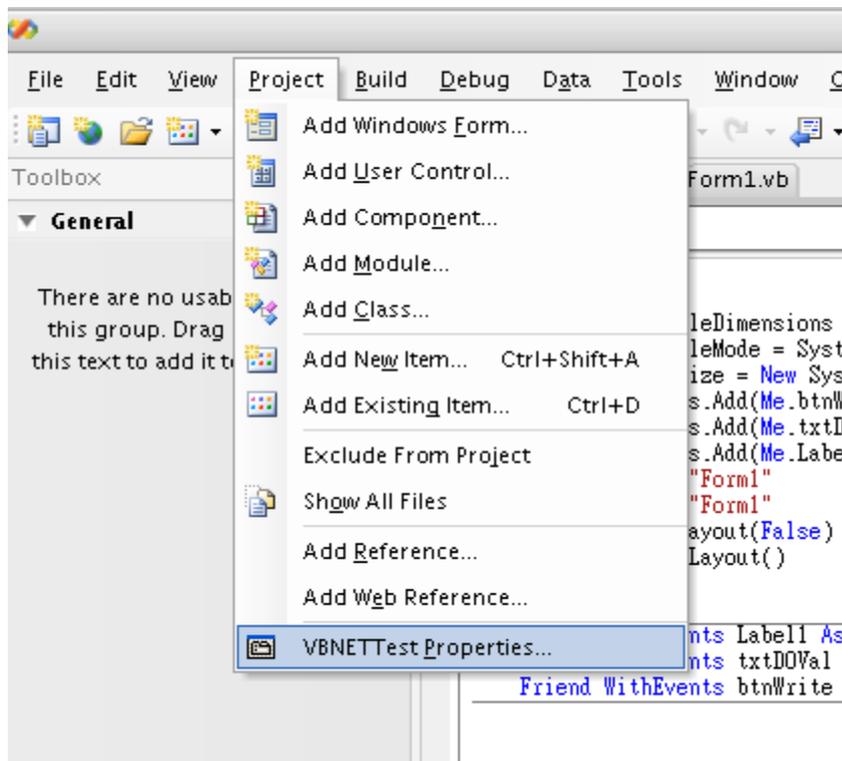
    '//Driver Initial
    wRtn = PCIELM4_DriverInit(wTotalBoards)
    If (wRtn) Then
        MsgBox("Driver Initial Error!!Error Code:" + Str(wRtn))
    End
    End If

    '//Write DO
    wRtn = PCIELM4_WriteDO(wBoardNo, Val(txtDOVal.Text))

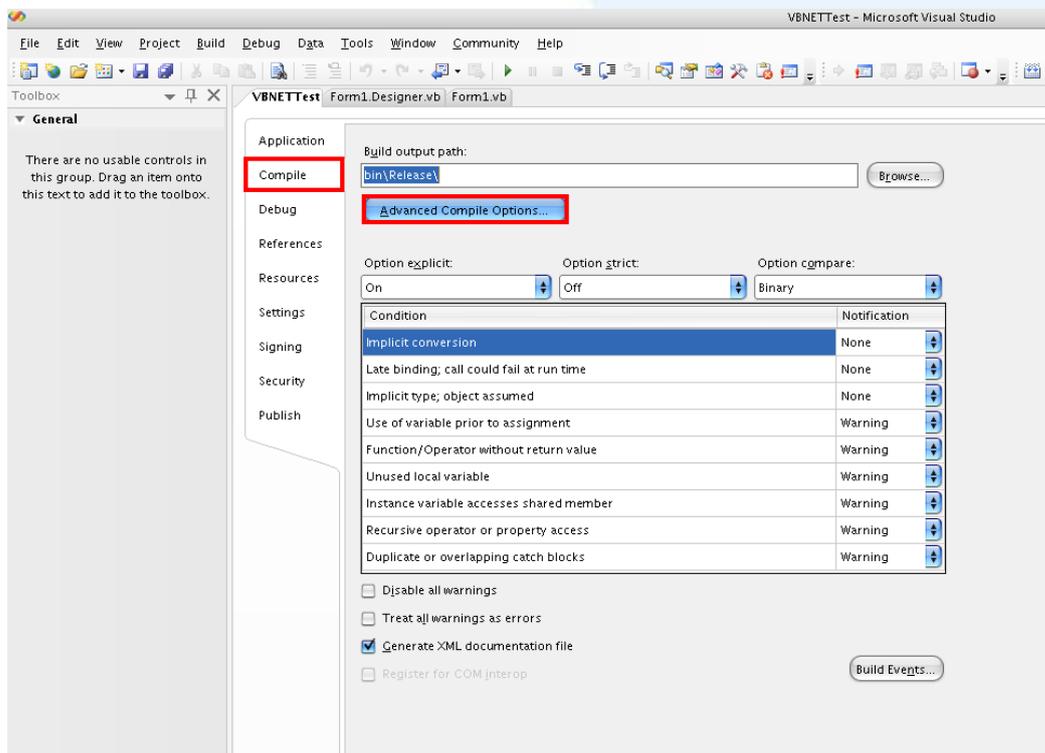
    wRtn = PCIELM4_DriverClose()
End Sub
```

## 步驟 2: 編譯應用程式

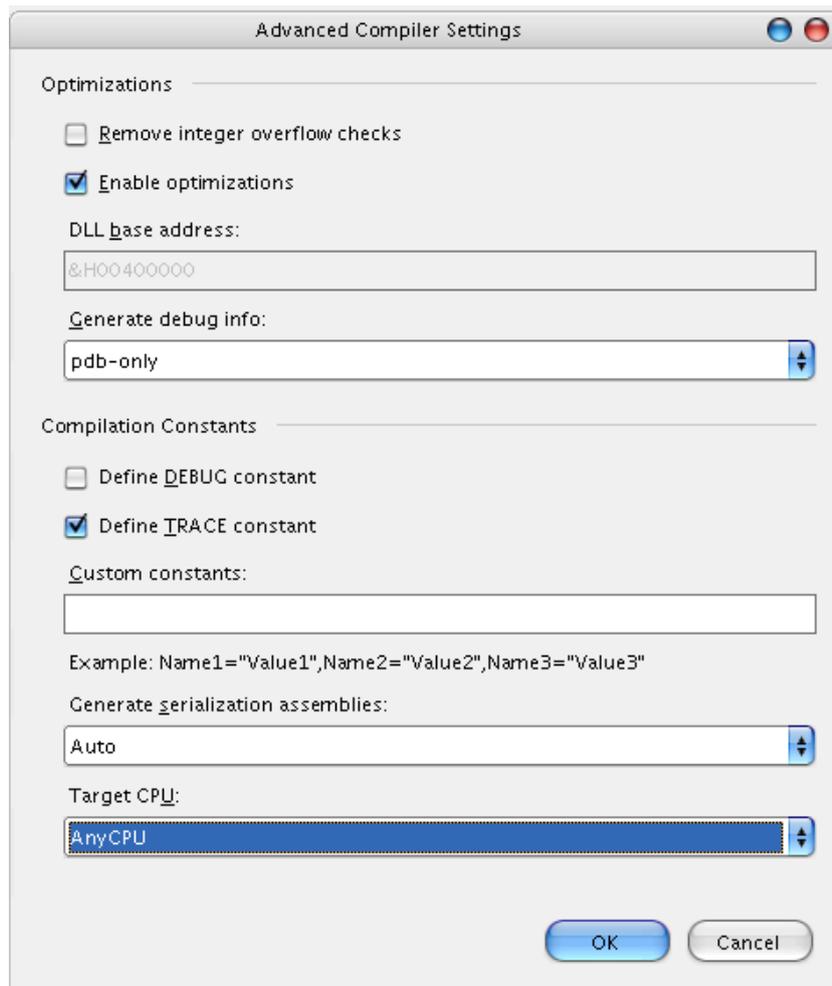
1. 在 Project 選單點擊 VBNETTest Properties。



2. 點擊 Compiler 後點擊 Advanced Compiler Option 按鈕，進入 Advanced Compiler Setting 視窗。



3. 在 Advanced Compiler Settings 下的 Target CPU 下拉選單選擇 AnyCPU。



Any CPU 選項-編譯出來的執行檔，當載入在 64 位元作業系統上的 64 位元版本.NET Framework，程式將會以 64 位元的行程來運作，否則將會以 32 位元的行程來運作。

x86 選項-不論作業系統或.NET Framework 的版本，執行檔永遠以 32 位元來運作。

x64 選項-不論作業系統或.NET Framework 的版本，執行檔永遠以 64 位元來運作。

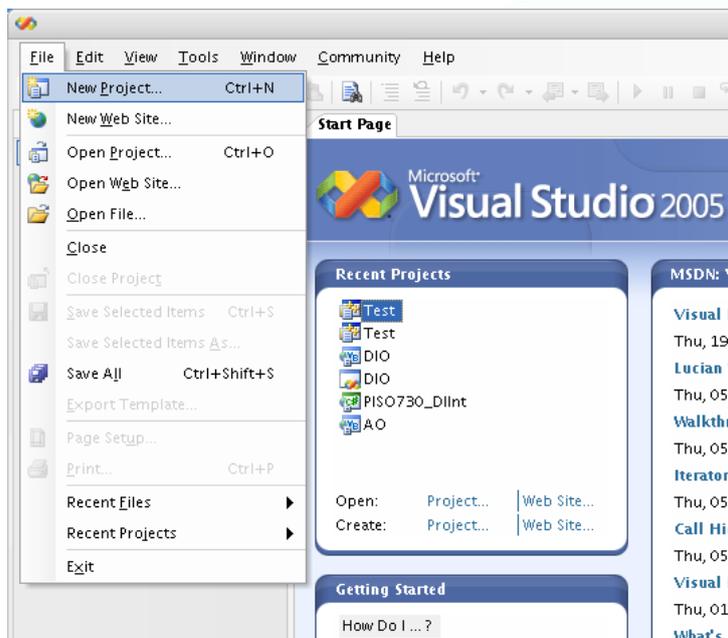
### 步驟 3: 測試應用程式

1. 按下 **F5** 來執行程式。
2. 在 **DO Value** 欄位輸入數值 **255**。
3. 並按下 **Write** 按鍵，輸出 **DO** 數值 **255**。

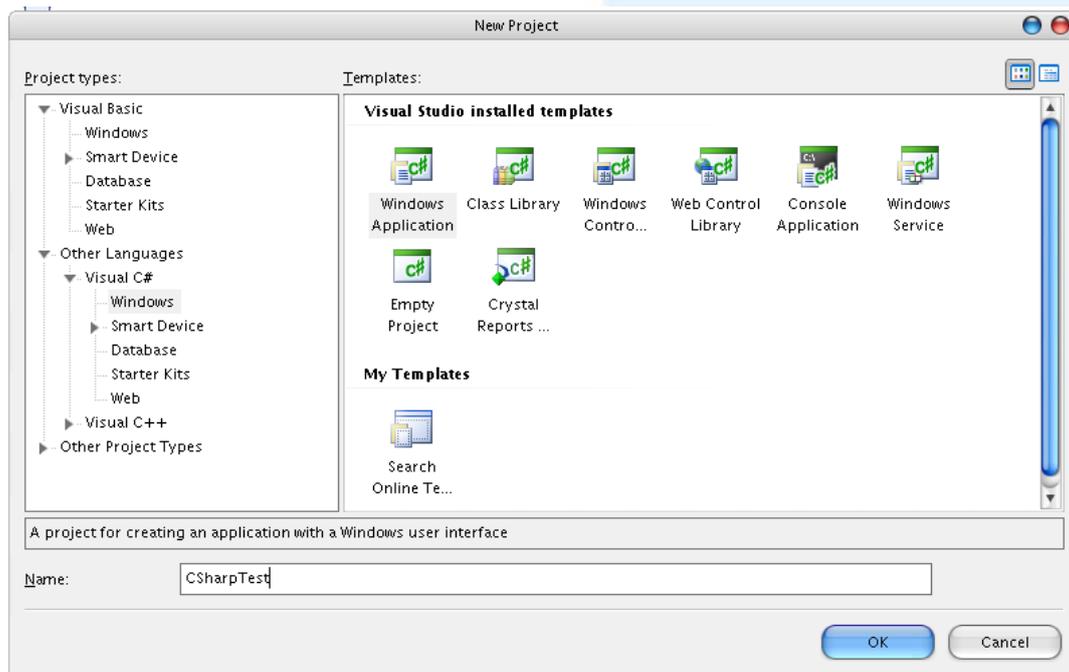
## 3.6. 在 Visual C#.NET

### 步驟 1: 撰寫應用程式

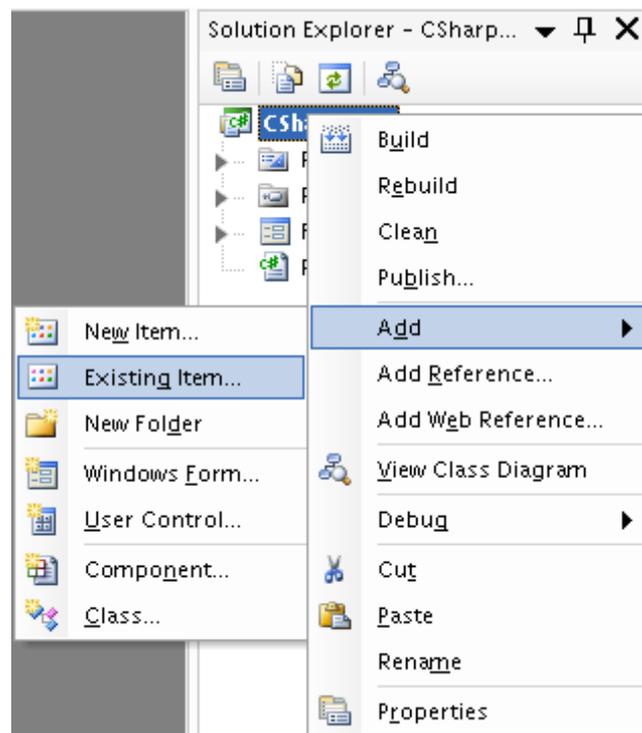
1. 至程式集開啟 Microsoft Visual Studio 2005
2. 從主要選單內選擇 File|New Project...



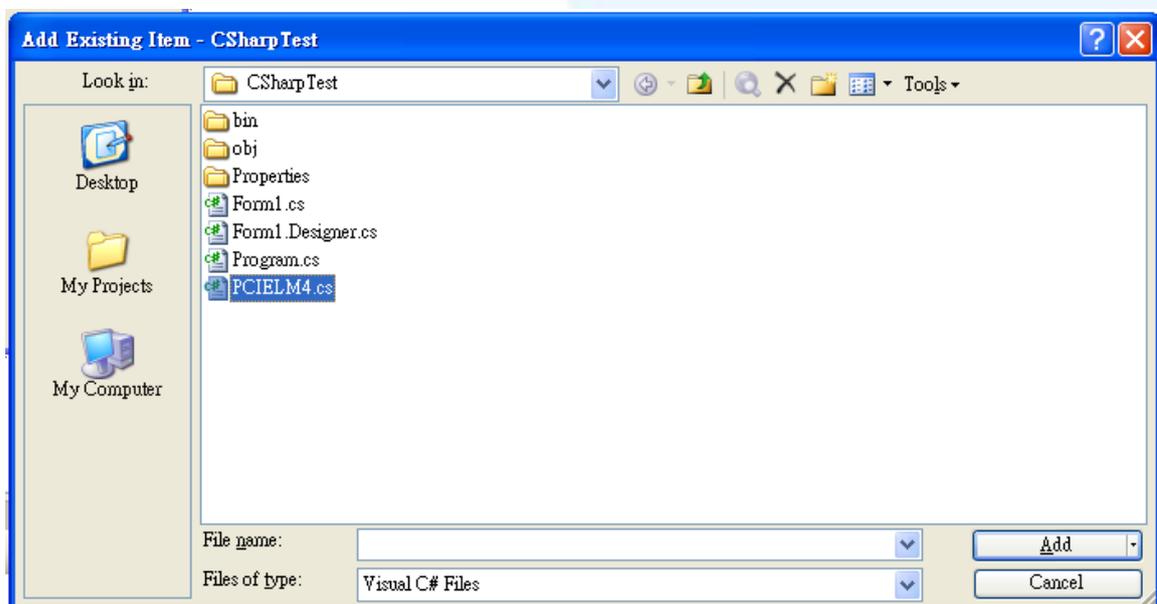
3. 在 Project type 下的列表點選項目 Visual C# 並在展開選單內選擇 Windows，然後再右方 Templates 表框內選擇 Windows Application，接下來至下方的 Name 鍵入專案名稱 CSharpTest 然後按下 OK 按鈕。



4. 在 Solution Explorer 開啟 Add Existing item 視窗。



5. 在 Add Existing item 選擇 PCIELM4.cs 後按 Add 按鍵添加宣告檔至專案裡。



- 設計視窗，在 **Form1** 放置一個 **Label** 控制項並在 **Text** 屬性上輸入 **DO Value**。接著放置 **TextBox** 控制項，並切換至屬性視窗上至 **Name** 屬性輸入 **txtDOVal**，最後放置一個 **Button** 控制項，並修改 **Name** 屬性為 **btnWrite** 及在 **Text** 屬性上輸入 **Write**。



## 7. 在 Form.cs 填寫程式碼如下：

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using PCIELM4_Ns; //Include the PCIELM4 namespace

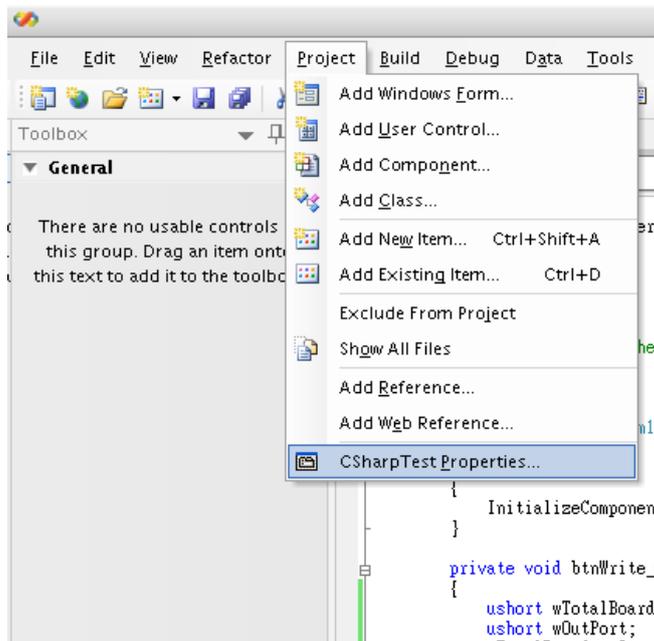
namespace CSharpTest
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnWrite_Click(object sender, EventArgs e)
        {
            ushort wTotalBoard, wRtn, wBoardNo;
            wTotalBoard = 0;
            //Initial the resource and get total board number form Driver
            wRtn = PCIELM4_DLL.PCIELM4_DriverInit(ref wTotalBoard);
            if (wRtn != PCIELM4_DLL.PCIELM4_NoErr)
            {
                MessageBox.Show("Driver Inital Error!!Error Code:" + wRtn.ToString());
                Close();
                return;
            }

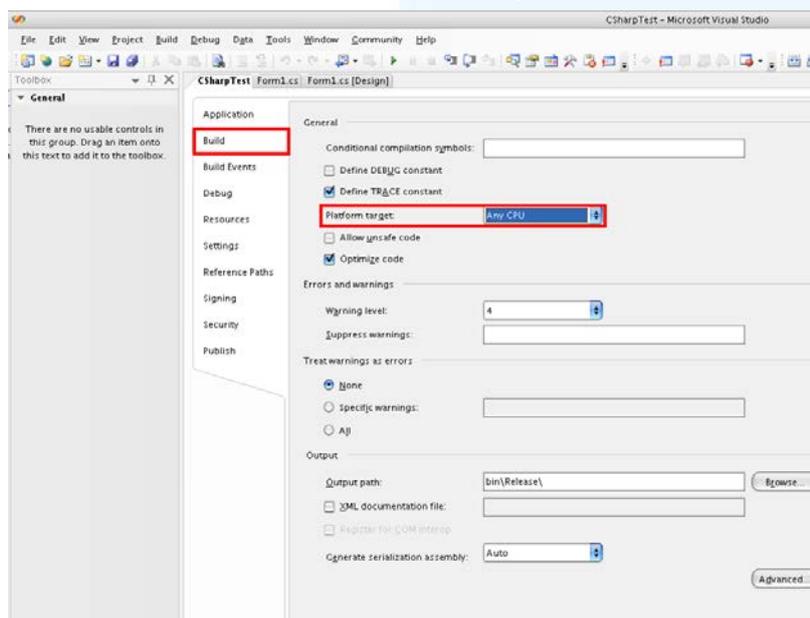
            wBoardNo = 0;
            wOutPort = 0;
            //Write DO
            wRtn = PCIELM4_DLL.PCIELM4_WriteDO(wBoardNo, Convert.ToInt32(txtDOVal.Text));
            //Release the resource from the driver
            wRtn = PCIELM4_DLL.PCIELM4_DriverClose();
        }
    }
}
```

## 步驟 2:編譯應用程式

1. 在 Project 選單點擊 CSharpTest Properties 。



2. 點擊 Build 至 Platform target 下拉選單選擇 Any CPU 。



Any CPU 選項-編譯出來的執行檔，當載入在 64 位元作業系統上的 64 位元版本.NET Framework，程式將會以 64 位元的行程來運作，否則將會以 32 位元的行程來運作。

x86 選項-不論作業系統或.NET Framework 的版本，執行檔永遠以 32 位元來運作。

x64 選項-不論作業系統或.NET Framework 的版本，執行檔永遠以 64 位元來運作。

### 步驟 3: 測試應用程式

1. 按下 **F5** 來執行程式。
2. 在 **DO Value** 欄位輸入數值 **255**。
3. 並按下 **Write** 按鍵，輸出 **DO** 數值 **255**。

## 3.7. 範例程式及文件

放置 PCIELM4 相關資料的網址與位置：



CD:\\ NAPDOS\\PCI\\PCIe-LM4\\

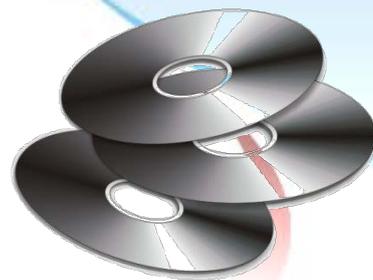


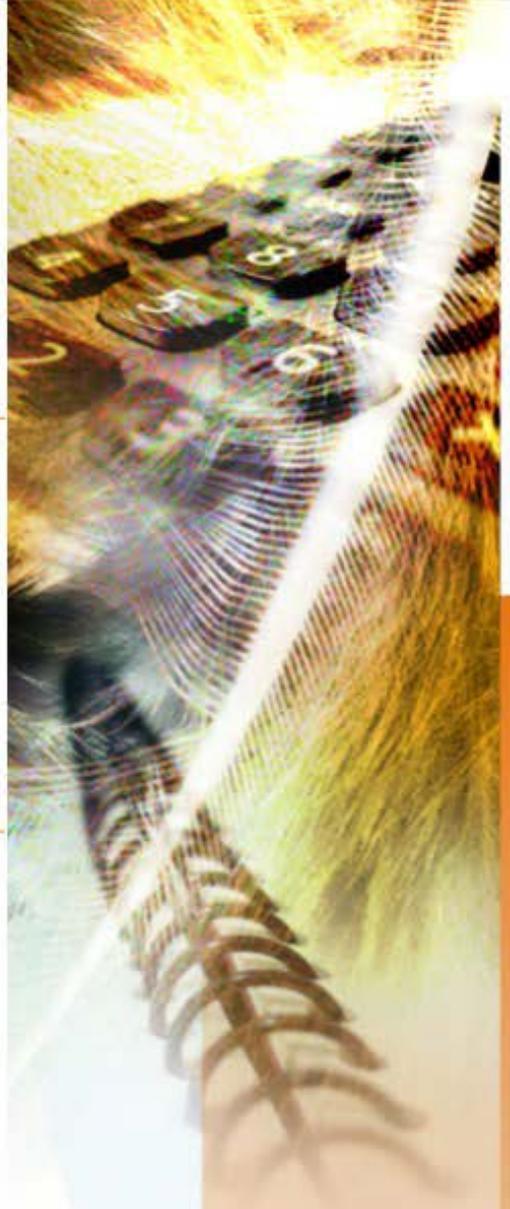
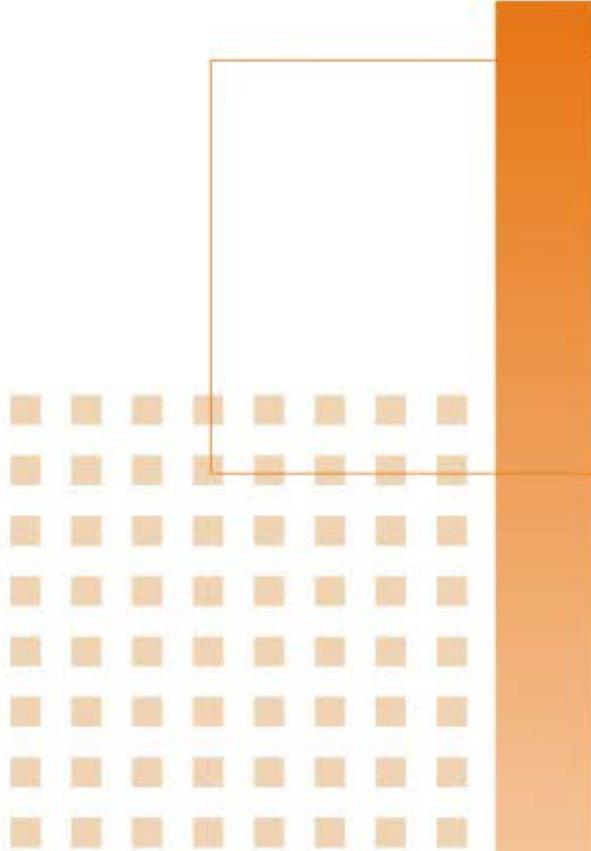
<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pcie-lm4/>



<ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pcie-lm4/>

PCIELM4 資料夾內檔案文件結構





## 4. 函式應用

提供泓格 PCIELM4 驅動函式庫所支援的範例程式列表以及每個範例程式的功能，並且會簡單介紹如何使用函式來產生各種應用方案

## 4.1. 導讀

PCIELM4 驅動函式庫集成了各種函式，使用者可以利用它們來開發各種應用程式在泓格的裝置上。這些 API 函式支援各種開發環境及程式語言，包括了 Microsoft Visual C++，Visual Basic，Microsoft Visual C++.NET，Microsoft Visual C#.NET，Microsoft Visual VB.NET。

**PCIELM4** 提供了幾個大類的函式集如下：

1. 驅動函式集:初始化裝置資源、取得裝置訊息、設定裝置及釋放裝置資源。
2. 數位輸出入函式集：操作控制具有數位輸出入功能的裝置。
3. 類比輸出函式集：操作裝置透過 DAC 輸送出電壓。
4. 類比輸入函式集：操作裝置透過 ADC 擷取電壓、電流、壓力、應變量等等數值…。

支援程式語言：

- Microsoft Visual C++ 4.0 or higher
- Microsoft Visual Basic 4.0 or higher
- Microsoft Visual C++.NET 2003 or higher
- Microsoft Visual C#.NET 2003 or higher
- Microsoft Visual Basic.NET 2003 or higher

泓格驅動函式庫提供的應用函式集總表：

驅動函式集	數位輸出入函式集	類比輸入函式集	類比輸出函式集
PCIELM4_DriverInit PCIELM4_DriverClose PCIELM4_GetCardInfo	PCIELM4_ReadDI PCIELM4_WriteDO	PCIELM4_ConfigAI PCIELM4_PollingAI PCIELM4_PollingAIH PCIELM4_ConfigAIAutoZero PCIELM4_SaveAIAutoZeroVal PCIELM4_AIHex2Vol	PCIELM4_ConfigAO PCIELM4_WriteAOVoltage PCIELM4_WriteAOVoltageH PCIELM4_StartAOVoltageALL PCIELM4_StartAOVoltageALLH PCIELM4_StopAOALL

## 4.2. 驅動函式庫

當使用者使用泓格 **PCIELM4** 驅動函式庫來開發板卡的應用程式時請遵照下列的呼叫流程來初始化及啟動驅動程式及函式庫。

### 呼叫流程

---



## **Board Num (Type: WORD, Size: 2 bytes)**

藉由 Board Num 指定您需要作 I/O 操作的設備。Board Num 由 PCI Configuration space 的 Bus num 及 Device number 決定，如果 Bus number 愈小 Board Number 愈前面，Device Number 愈小 Board Num 愈前面。

## **PCIELM4\_DriverInit 及 PCIELM4\_DriverClose 函式**

PCIELM4\_DriverInit 分配取得所有板卡的資源及數量在應用程式啟動的時候，所以使用者必需呼叫 PCIELM4\_DriverInit 在應用程式起始點，並且在使用其他函式之前。

PCIELM4\_DriverClose 會釋放板卡所佔用的系統資源，當使用者不需要在操作板卡及程式終結前呼叫。

## **PCIELM4\_GetCardInfo 函式**

當使用者有需求知道板卡的名稱或相關硬體資訊時可透過 PCIELM4\_GetCardInfo 函式取得，一般可忽略呼叫此函式。

## 4.3. 數位輸出輸入

數位輸入輸出函式集執行數位輸入及數位輸出等操作。在每一個資料採集板卡上，所有的數位輸入輸出線被分成一個一個的單位稱為埠，每一個埠依板卡的設計會有 8、16 或 32 線。

### 4.3.1. 數位輸入

泓格 PCIELM4 驅動函式庫的數位輸入函式集可執行數位輸入等功能。

- 軟體觸發

使用者可以呼叫 `PCIELM4_ReadDI` 函式讀取指定數位輸入埠的資料。

#### 呼叫流程

---

設定埠號及 DI 值

`PCIELM4_ReadDI`

---

### 4.3.2. 數位輸出

泓格 PCIELM4 驅動函式庫的數位輸出入函式集可執行數位輸出等功能。

使用者可以透過 PCIELM4\_WriteDO 函式簡易的設定數位輸出埠的資料。

#### 呼叫流程

---

設定埠號及 DO 值

PCIELM4\_WriteDO

---

## 4.4. 類比輸入

類比輸入函式集運行資料擷取卡上類比輸入的功能。它可以擷取單通道多筆資料。

- **軟體觸發**

透過軟體觸發資料轉換來取得類比資料，函式庫提供單通道多筆資料讀取。

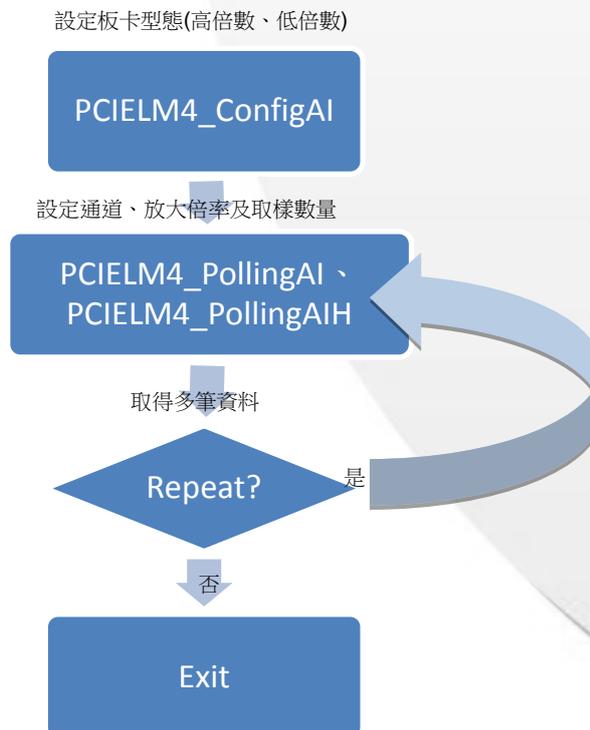


一般在 Windows 下使用軟體觸發，均容易受到多工系統的影響，造成取樣時間點將會受到其他系統任務的影響造成延遲取樣，故不建議使用者應用軟體觸發的方式量測類比訊號波型，除非量測的波型屬於非常低速的波型(低於 500Hz)

- **單通道多筆資料取樣**

可以在單一的通道上連續採集一個以上的類比資料。

### 呼叫流程



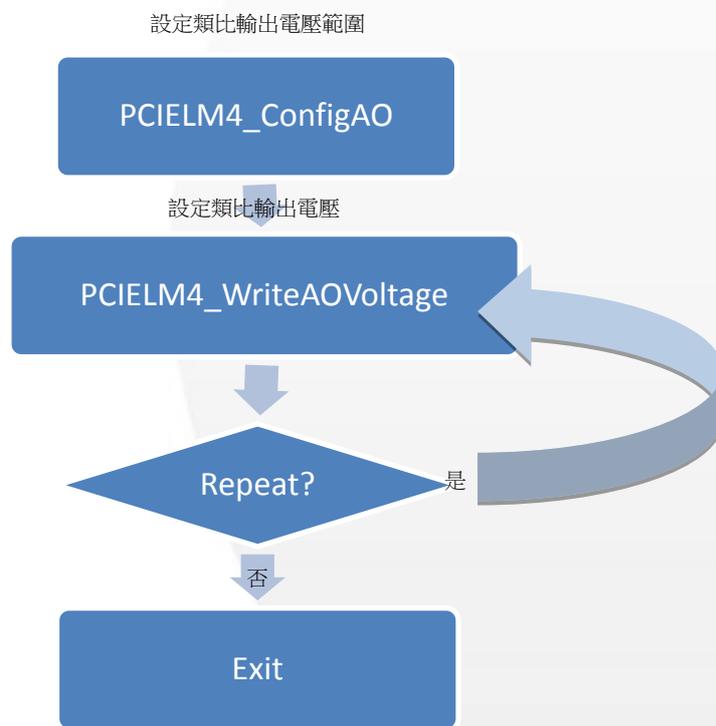
## 4.5. 類比輸出

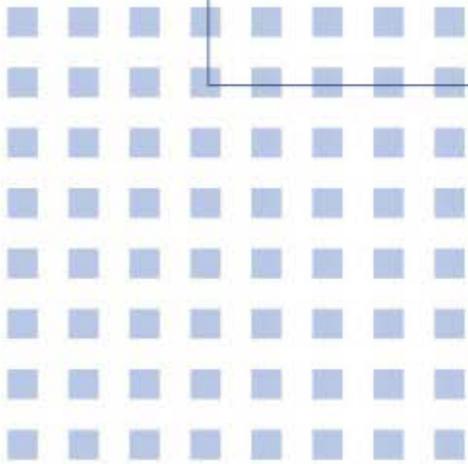
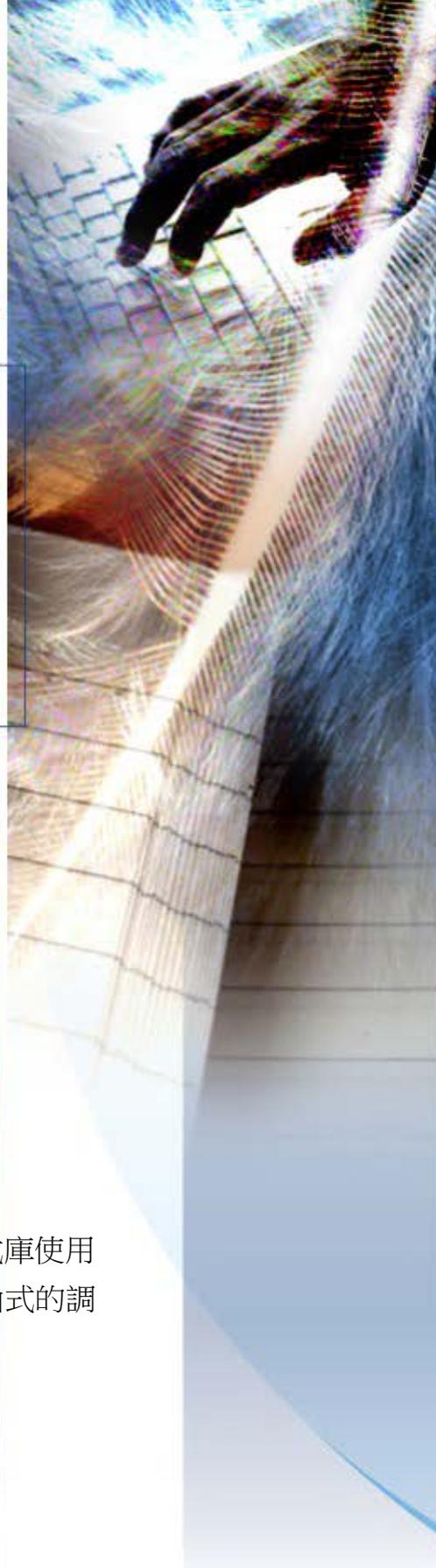
泓格 PCIELM4 驅動函式庫的類比輸出入函式集可執行類比輸出等功能。

- 靜態電壓輸出  
設定類比輸出通道輸出一固定的直流電壓。

呼叫流程

---





## 5. 函式參考

提供泓格 PCIELM4 驅動函式庫使用指南，按照功能分組來說明函式的調用方式。

## 5.1. 函式介紹

使用前請注意下列關鍵字。以方便您的閱讀。

關鍵字	呼叫函式前需由使用者設定該參數	使用者呼叫函式後，會回傳參數值
[Input]	Yes	No
[Output]	No	Yes

每一個泓格 PCIELM4 函式皆是如下的形式:

Status = 函式名稱(參數 1, 參數 2, ...參數 n)

每個函式皆會回傳一個值在 **status** 變數裡，它可以顯示函式的呼叫成功或是失敗。

Status(值)	結果
0	成功
>0	失敗

Status 的格式為一 2 位元無號整數值(WORD)，更多相關錯誤碼 Status 值的定義請參考 A.1. 函數回傳值定義

## 5.1.1. 驅動函式集

---

### ***PCIELM4\_DriverInit***

---

呼叫此函式時會向系統要求分配資源，並且開始尋找所有 **PCIELM4** 有支援的板卡，而對每一張板卡作初使化動作，最後取得板卡的數量。需在程式起始點，使用其他的函式之前呼叫。

➤ 語法

```
WORD PCIELM4_DriverInit(  
    WORD *wTotalBoards  
);
```

➤ 參數

*wTotalBoards*

**[Output]** 取得所有的板卡數量。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

### ***PCIELM4\_DriverClose***

---

呼叫此函式時，會將佔用的資源釋放歸還給系統。需在程式終結前呼叫。

➤ 語法

```
WORD PCIELM4_DriverClose(  
    void  
);
```

➤ 參數

無任何參數。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

## PCIELM4\_GetCardInfo

---

取得板卡的硬體資料、軟體資料及板卡名稱。

➤ 語法

```
WORD PCIELM4_GetCardInfo(  
    WORD wBoardNo,  
    PPCIELM4_DEVICE_INFO sDevInfo,  
    PPCIELM4_CARD_INFO sCardInfo,  
    char *szModelName  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 *wBoardNo* 為 0，第二張板卡的 *wBoardNo* 為 1，依此類推。

*sDevInfo*

**[Output]** 取得板卡在電腦上的系統資訊。變數型態為 `PPCIELM4_DEVICE_INFO` 結構。

*sCardInfo*

**[Output]** 取得板卡的硬體規格資訊。變數型態為一個 `PPCIELM4_CARD_INFO` 結構。

*szModelName[]*

**[Output]** 取得板卡的名稱。請宣告一個 20 個字元大小的字串。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

## 5.1.2. 數位輸出輸入函式集

---

### ***PCIELM4\_ReadDI***

---

讀取使用者所指定的輸入埠的資料。

➤ 語法

```
WORD PCIELM4_ReadDI(  
    WORD wBoardNo,  
    DWORD *dwDIVal  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*dwDIVal*

**[Output]** 讀取輸入埠裡的資料。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

# PCIELM4\_WriteDO

---

寫入資料到所用者所指定的輸出埠。

➤ 語法

```
WORD PCIELM4_WriteDO(  
    WORD wBoardNo,  
    DWORD dwDOVal  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*dwDOVal*

**[Input]** 寫入資料至輸出埠。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

## 5.1.3. 類比輸入函式集

### ***PCIELM4\_ConfigAI***

此函數用來設定板卡類比輸入功能的參數值，使用類比輸入函式集前必需先呼叫此函式。

➤ 語法

```
WORD PCIELM4_ConfigAI(  
    WORD wBoardNo,  
    WORD wFIFOSizeKB,  
    DWORD dwBufferSizeCount,  
    WORD wCardType,  
    WORD wDelaySettingTime,  
    DWORD dwMode  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*wFIFOSizeKB*

**[Input]** 保留參數，尚無作用。

*dwBufferSizeCount*

**[Input]** 保留參數，尚無作用。

*wCardType*

**[Input]** 保留參數，尚無作用。

*wDelaySettingTime*

**[Input]** 保留參數，尚無作用。

*dwMode*

**[Input]** 保留參數，尚無作用。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

# PCIELM4\_PollingAI

---

此函式以軟體輪詢的方式取得類比輸入通道裡複數筆數的資料。

➤ 語法

```
WORD PCIELM4_PollingAI(  
    WORD wBoardNo,  
    WORD wChannel,  
    WORD wConfig,  
    DWORD dwSampleRate,  
    DWORD dwDataCount,  
    double dValue[ ]  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 *wBoardNo* 為 0，第二張板卡的 *wBoardNo* 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定擷取類比輸入資料的通道號碼。

*wConfig*

**[Input]** 由使用者設定資料擷取類比輸入資料的配置碼，不同型號的板卡因設計上的不同也會有不同的類比輸入範圍，配置碼及板卡支援的輸入範圍請查閱附錄 A.3.1. 類比輸入配置碼。此配置碼將會影響擷取資料的精度及量測範圍。

*dwSampleRate*

**[Input]** 由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwDataCount*

**[Input]** 由使用者設定需要擷取的類比輸入資料筆數。

*dValue[ ]*

**[Output]** 請宣告一個雙精度浮點數陣列(陣列大小為 `dwDataCount`)，將輪詢後得到的每筆類比輸入資料儲存在此陣列裡，單位為 `volts`。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

## ***PCIELM4\_PollingAIH***

---

此函式以**軟體輪詢**的方式一次取得同一個通道裡複數筆數的類比輸入值。

➤ 語法

```
WORD PCIELM4_PollingAIH(  
    WORD wBoardNo,  
    WORD wChannel,  
    WORD wConfig,  
    DWORD dwSampleRate,  
    DWORD dwDataCount,  
    DWORD dwValue[ ]  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 `wBoardNo` 為 0，第二張板卡的 `wBoardNo` 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定擷取類比輸入資料的通道號碼。

*wConfig*

**[Input]** 由使用者設定資料擷取類比輸入資料的配置碼，不同型號的板卡因設計上的不同也會有不同的類比輸入範圍，配置碼及板卡支援的輸入範圍請查閱附錄 A.3.1. 類比輸入配置碼。**此配置碼將會影響擷取資料的精度及量測範圍。**

*dwSampleRate*

**[Input]** 由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwDataCount*

**[Input]** 由使用者設定需要擷取的類比輸入資料筆數。

*dwValue[ ]*

**[Output]** 請宣告一個 **DWORD** 陣列(陣列大小為 **dwDataCount**)，將輪詢後得到的每筆類比輸入資料儲存在此陣列裡，此數值為十六進位值。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

## ***PCIELM4\_AIHex2Vol***

---

此函式可將此系列板卡的十六進位值類比輸入值轉換為 **double** 型別。

➤ 語法

```
WORD PCIELM4_AIHex2Vol(  
    WORD wBoardNo,  
    WORD wChannel,  
    WORD wConfig,  
    DWORD dwValue,  
    double *dVal  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定擷取類比輸入資料的通道號碼。

*wConfig*

**[Input]** 由使用者設定資料擷取類比輸入資料的配置碼，不同型號的板卡因設計上的不同也會有不同的類比輸入範圍，配置碼及板卡支援的輸入範圍請查閱附錄 A.3.1. 類比輸入配置碼。此配置碼將會影響擷取資料的精度及量測範圍。

*dwValue*

**[Input]** 請宣告一個 **DWORD**，輸入單筆類比輸入資料的十六進位值。

*dVal*

**[Output]** 依 **dwValue** 數值轉換的 **double** 數值。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

## ***PCIELM4\_ConfigAIAutoZero***

---

此函式僅支援 **Load Cell** 功能的類比輸入通道，由使用者設定啟用或關閉校準偏移量功能。

➤ 語法

```
WORD PCIELM4_ConfigAIAutoZero(  
    WORD wBoardNo,  
    WORD wChannel,  
    WORD wEnableAutoZero  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定擷取類比輸入資料的通道號碼。

*wEnableAutoZero*

**[Input]** 由使用者設定啟用或關閉校準偏移量功能。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

## **PCIELM4\_SaveAIAutoZeroVal**

---

此函式僅支援 **Load Cell** 功能的類比輸入通道，使用者可輸入十六進位值作為指定通道的校準偏移量。

➤ 語法

```
WORD PCIELM4_SaveAIAutoZeroVal(  
    WORD wBoardNo,  
    WORD wChannel,  
    DWORD dwSampleRate,  
    DWORD dwAutoZeroValue  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定擷取類比輸入資料的通道號碼。

*dwSampleRate*

**[Input]** 由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwAutoZeroValue*

**[Input]** 由使用者輸入的十六進位值校準偏移量。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

## 5.1.4. 類比輸出函式集

### **PCIELM4\_ConfigAO**

此函數用來設定類比輸出的參數值，使用類比輸出函式集前必需先呼叫此函式。

➤ 語法

```
WORD PCIELM4_ConfigAO(  
    WORD wBoardNo,  
    WORD wChannel,  
    WORD wMode,  
    WORD wAORange  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 *wBoardNo* 為 0，第二張板卡的 *wBoardNo* 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定類比輸出通道的號碼。

*wMode*

**[Input]** 保留參數，尚無作用。

*wAORange*

**[Input]** 由使用者設定類比輸出通道的配置碼，配置碼的代碼請查閱附錄 A.3.2. 類比輸出配置碼(電壓)。此配置碼將會影響類比輸出的精度及輸出範圍。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

# PCIELM4\_WriteAOVoltage

---

呼叫此函式控制類比輸出通道輸出一固定的電壓，此電壓設定值為浮點數值。

➤ 語法

```
WORD PCIELM4_WriteAOVoltage(  
    WORD wBoardNo,  
    WORD wChannel,  
    float fAOValue  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 *wBoardNo* 為 0，第二張板卡的 *wBoardNo* 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定類比輸出通道的號碼。

*fAOValue*

**[Input]** 由使用者設定類比輸出電壓值，此值為浮點數值，單位為伏特(V)。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

# PCIELM4\_WriteAOVoltageH

---

呼叫此函式控制類比輸出通道輸出一固定的電壓，此電壓設定值為十六進位值。

➤ 語法

```
WORD PCIELM4_WriteAOVoltageH(  
    WORD wBoardNo,  
    WORD wChannel,  
    DWORD dwAOValue  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 *wBoardNo* 為 0，第二張板卡的 *wBoardNo* 為 1，依此類推。

*wChannel*

**[Input]** 由使用者設定類比輸出通道的號碼。

*dwAOValue*

**[Input]** 由使用者設定類比輸出電壓值，此值為十六進位值。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

# PCIELM4\_StartAOVoltageALL

---

依使用者指定的數據大小、數據緩衝區及循環模式啟動高速類比輸出模式，類比輸出緩衝區使用浮點數電壓。

➤ 語法

**WORD** PCIELM4\_StartAOVoltageALL(

**WORD** wBoardNo,

**float** fSamplingRate,

**DWORD** dwDataCount,

**DWORD** dwCycleNum,

**float** fValueCH0[ ],

**float** fValueCH1[ ]

);

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*fSamplingRate*

**[Input]** 為一浮點數值，由使用者設定數位輸出埠的輸出每筆類比資料的頻率(次/秒)。

*dwDataCount*

**[Input]** 由使用者設定需要輸出一個波型的資料筆數。

*dwCycleNum*

**[Input]** 0:連續模式，如果要終止高速類比輸出功能，請使用 **PCIELM4\_StopAOALL** 函式。

*fValueCH0[ ]*

**[Input]** 儲存通道 0 類比輸出資料(浮點數電壓)至數位輸出緩衝區。

*fValueCH1[]*

**[Input]** 儲存通道 1 類比輸出資料(浮點數電壓)至數位輸出緩衝區。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

---

## PCIELM4\_StartAOVoltageALLH

---

依使用者指定的數據大小、數據緩衝區及循環模式啟動高速類比輸出模式，類比輸出緩衝區使用十六進位值。

➤ 語法

**WORD** PCIELM4\_StartAOVoltageALLH(

**WORD** wBoardNo,

**float** fSamplingRate,

**DWORD** dwDataCount,

**DWORD** dwCycleNum,

**DWORD** dwValueCH0[ ],

**DWORD** dwValueCH1[ ]

);

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 **wBoardNo** 為 0，第二張板卡的 **wBoardNo** 為 1，依此類推。

*fSamplingRate*

**[Input]** 為一浮點數值，由使用者設定數位輸出埠的輸出每筆類比資料的頻率(次/秒)。

*dwDataCount*

**[Input]** 由使用者設定需要輸出一個波型的資料筆數。

*dwCycleNum*

**[Input]** 0:連續模式，如果要終止高速類比輸出功能，請使用 `PCIELM4_StopAOALL` 函式。

*dwValueCH0[]*

**[Input]** 儲存通道 0 類比輸出資料(十六進位值)至數位輸出緩衝區。

*dwValueCH1[]*

**[Input]** 儲存通道 1 類比輸出資料(十六進位值)至數位輸出緩衝區。

➤ **回傳值**

請參考 A.1. 函數回傳值定義。

---

# PCIELM4\_StopAOALL

---

呼叫此函式，用來停止高速類比輸出的運作。

➤ 語法

```
WORD PCIELM4_StopAOALL(  
    WORD wBoardNo,  
);
```

➤ 參數

*wBoardNo*

**[Input]** 由使用者指定的板卡編號，第一張板卡的 *wBoardNo* 為 0，第二張板卡的 *wBoardNo* 為 1，依此類推。

➤ 回傳值

請參考 A.1. 函數回傳值定義。

## 5.2. 資料型態

---

### ***PPCIELM4\_DEVICE\_INFO***

---

➤ 語法

```
typedef struct _PCIELM4_DEVICE_INFO_  
{  
    DWORD dwSize;  
    WORD wVendorID;  
    WORD wDeviceID;  
    WORD wSubVendorID;  
    WORD wSubDeviceID;  
    DWORD dwBAR[6];  
    UCHAR BusNo;  
    UCHAR DevNo;  
    UCHAR IRQ;  
    UCHAR Aux;  
    ULONGLONG dwBarVirtualAddress [6];  
}PCIELM4_DEVICE_INFO,*PPCIELM4_DEVICE_INFO;
```

➤ 成員

*dwSize*

**[Output]** 取得此資料結構的大小，單位為 byte。

*wVendorID*

**[Output]** 取得板卡的 Vendor ID。

*wDeviceID*

**[Output]** 取得板卡的 Device ID。

*wSubVendorID*

**[Output]** 取得板卡的 Sub Vendor ID。

*wSubDeviceID*

**[Output]** 取得板卡的 Sub Device ID 。

*dwBAR[]*

**[Output]** 取得板卡的 Base Address 。

Base Address	dwBAR [Index]
Bar 0	dwBAR[0]
Bar 1	dwBAR[1]
Bar 2	dwBAR[2]
Bar 3	dwBAR[3]
Bar 4	dwBAR[4]
Bar 5	dwBAR[5]

*BusNo*

**[Output]** 取得板卡的 Bus 號碼 。

*DevNo*

**[Output]** 取得板卡的 Device 號碼 。

*IRQ*

**[Output]** 取得板卡的 IRQ 號碼 。

*Aux*

**[Output]** 取得板卡的 Aux 號碼 。

*dwBarVirtualAddress []*

**[Output]** 取得 memory mapping I/O 虛擬記憶體位址 。

Virial Memory Address	dwBAR [Index]
Bar 0	dwBarVirtualAddress [0]
Bar 1	dwBarVirtualAddress [1]
Bar 2	dwBarVirtualAddress [2]
Bar 3	dwBarVirtualAddress [3]
Bar 4	dwBarVirtualAddress [4]
Bar 5	dwBarVirtualAddress [5]

---

# PPCIELM4\_CARD\_INFO

---

➤ 語法

```
typedef struct _PCIELM4_CARD_INFO_  
{  
    DWORD dwSize;  
    DWORD dwModelNo;  
    UCHAR CardID;  
    UCHAR wSingleEnded;  
    WORD wAIOResolution;  
    WORD wAIChannels;  
    WORD wAOChannels;  
    WORD wDIPorts;  
    WORD wDOPorts;  
    WORD wDIOPorts;  
    WORD wDIOPortWidth;  
    WORD wCounterChannels;  
    WORD wMemorySize;  
    DWORD dwReserved1[6];  
}PCIELM4_CARD_INFO,*PPCIELM4_CARD_INFO;
```

➤ 成員

*dwSize*

**[Output]** 取得此資料結構的大小，單位為 byte。

*dwModelNo*

**[Output]** 取得板卡的模組識別號碼，模組識別號碼可參考 A.2. 模組識別號碼。

*CardID*

**[Output]** 取得板卡的卡片識別號碼，若取得數值為 255 代表板卡未支援此功能。

*wSingleEnded*

**[Output]** 取得板卡類比輸入接線設定值，PCIe-LM4 為數值為 2，差動式(DIFF)。

*wAIOResolution*

**[Output]** 保留資訊，PCIe-LM4 的 AI 解析度為 24-bit，AO 解析度為 16-bit。

*wAIChannels*

**[Output]** 取得板卡類比輸入的通道數量。

*wAOChannels*

**[Output]** 取得板卡類比輸出的通道數量。

*wDIPorts*

**[Output]** 取得板卡單向數位輸入埠的埠數量。

*wDOPorts*

**[Output]** 取得板卡單向數位輸入埠的埠數量。

*wDIOPorts*

**[Output]** 取得板卡雙向數位輸出入埠的埠數量。

*wDIOPortWidth*

**[Output]** 取得板卡數位輸出入埠的寬度，PCIe-LM4 為 16-bit。

*wCounterChannels*

**[Output]** 取得板卡計時計數的通道數量。

*wMemorySize*

**[Output]** 取得板卡的內建記憶體大小，單位為 kByte。

*dwReserved1[]*

**[Output]** 取得保留資訊。

# A

## 附錄 A. 函式回傳值與配置碼

本章節列出了泓格 PCIELM4 驅動函式庫所返回的狀態回傳值及配置設定代碼。

## A.1. 函數回傳值定義

呼叫所有的函式，都會回傳出一個整數值。透過這個值可以得知該函數運作的狀況，下表提供每個回傳值所代表的定義。

回傳值	定義	說明
0	PCIELM4_NoErr	正確
1	PCIELM4_OpenDriverErr	開啟驅動程式發生錯誤
2	PCIELM4_PnPDriverErr	Plug&Play 時發生錯誤
3	PCIELM4_DriverNoOpen	驅動程式未開啟
4	PCIELM4_GetDriverVersionErr	取得驅動程式版本錯誤
5	PCIELM4_ExceedBoardNumber	板卡號碼錯誤
6	PCIELM4_FindBoardErr	找不到任何的板卡
7	PCIELM4_BoardMappingErr	板卡物件索引(Board Mapping)錯誤
8	PCIELM4_DIOModesErr	數位輸出輸入模式設定錯誤
9	PCIELM4_InvalidAddress	不合法的位址
10	PCIELM4_InvalidSize	不合法的大小
11	PCIELM4_InvalidPortNumber	不合法的埠號
12	PCIELM4_UnSupportedModel	未支援此板卡
13	PCIELM4_UnSupportedFun	未支援此函式
14	PCIELM4_InvalidChannelNumber	不合法的通道號碼
15	PCIELM4_InvalidValue	不合法的值
16	PCIELM4_InvalidMode	不合法的模式
17	PCIELM4_GetAIStatusTimeOut	取得類比輸入狀態逾時
18	PCIELM4_TimeOutErr	超過時間發生異常
19	PCIELM4_CfgCodeIndexErr	找不到適合的配置碼表格索引
20	PCIELM4_ADCCTLTimeoutErr	ADC 控制器愈期
21	PCIELM4_FindPCIIndexErr	找不到適合的 PCI 表格索引值
22	PCIELM4_InvalidSetting	不合法的設定值
23	PCIELM4_AllocateMemErr	分配記憶體空間失敗
24	PCIELM4_InstallEventErr	安裝中斷事件失敗
25	PCIELM4_InstallIrqErr	安裝中斷失敗
26	PCIELM4_RemoveIrqErr	移除中斷失敗
27	PCIELM4_ClearIntCountErr	清除中斷計數量失敗
28	PCIELM4_GetSysBufferErr	取得系統緩衝區失敗
29	PCIELM4_CreateEventErr	CreateEvent 失敗
30	PCIELM4_UnSupportedResolution	未支援此解析度
31	PCIELM4_CreateThreadErr	CreateThread 失敗

32	PCIELM4_ThreadTimeOutErr	執行緒逾時
33	PCIELM4_FIFOOverFlowErr	FIFO 溢滿
34	PCIELM4_FIFOTimeOutErr	FIFO 逾時
35	PCIELM4_GetIntInstStatus	取得中斷安裝狀態
36	PCIELM4_GetBufStatus	取得 SYS 緩衝區狀態
37	PCIELM4_SetBufCountErr	設定緩衝區大小錯誤
38	PCIELM4_SetBufInfoErr	設定緩衝區資料錯誤
39	PCIELM4_FindCardIDErr	找不到卡片識別碼
40	PCIELM4_EventThreadErr	事件執行緒錯誤
41	PCIELM4_AutoCreateEventErr	自動創建事件失敗
42	PCIELM4_RegThreadErr	註冊執行緒失敗
43	PCIELM4_SearchEventErr	尋找事件失敗
44	PCIELM4_FifoResetErr	FIFO 清除失敗
45	PCIELM4_InvalidBlock	不合法的 EEPROM 區塊
46	PCIELM4_InvalidAddr	不合法的 EEPROM 位址
47	PCIELM4_AcqireSpinLock	獲得旋轉鎖失敗
48	PCIELM4_ReleaseSpinLock	釋放旋轉鎖失敗
49	PCIELM4_SetControlErr	類比輸入設定錯誤
50	PCIELM4_InvalidChannels	不合法的通道數
51	PCIELM4_SearchCardErr	不合法的板卡號碼
52	PCIELM4_SetMapAddressErr	設定映像位址失敗
53	PCIELM4_ReleaseMapAddressErr	釋放映像位址失敗
54	PCIELM4_InvalidOffset	不合法的位移
55	PCIELM4_ShareHandleErr	開啟 Share Memory 失敗
56	PCIELM4_InvalidDataCount	不合法的資料量
57	PCIELM4_WriteEEPErr	寫入 EEPROM 失敗
58	PCIELM4_CardIOErr	使用 CardIO 失敗
59	PCIELM4_IOErr	使用 MemoryIO 失敗
60	PCIELM4_SetScanChannelErr	設置 channel scan number 失敗
61	PCIELM4_SetScanConfigErr	設置 channel scan configuration 失敗
62	PCIELM4_GetMMIOMapStatus	取得 Memory Mapping IO 狀態失敗
63	PCIELM4_InvalidEEPCmd	不合法的 EEPROM 命令
64	PCIELM4_CheckEEPCRCErr	檢查 EEPROM CRC 錯誤
65	PCIELM4_CtlEEPFail	控制 EEPROM 失敗
66	PCIELM4_UnknownEEPErr	未知的 EEPROM 錯誤
67	PCIELM4_SetAIMuxErr	設置 AI 多功器 失敗
68	PCIELM4_SetAICONErr	設置 AI CON 失敗
69	PCIELM4_SetAIRATEErr	設置 AI Rate 失敗

70	PCIELM4_WriteAOBufErr	寫入 AO 緩衝區失敗
71	PCIELM4_TimeOut1Err	逾時 1
72	PCIELM4_TimeOut2Err	逾時 2
73	PCIELM4_TimeOut3Err	逾時 3
74	PCIELM4_TimeOut4Err	逾時 4
75	PCIELM4_TimeOut5Err	逾時 5
76	PCIELM4_TimeOut6Err	逾時 6
77	PCIELM4_SetAIUnknowErr	設置 AI 未知錯誤

## A.2. 模組識別號碼

識別碼	值(十六進制)	支援的資料擷取板卡
PCIELM4	357701C4	PCIe-LM4

## A.3. 配置碼定義

透過配置碼可用來設定各種硬體的功能來產生不同的應用。

### A.3.1. 類比輸入配置碼

使用者可查詢下表在

類比輸入函式集設定板卡類比輸入的電壓範圍及極性，每張板卡支援的類比輸入電壓範圍及極性皆有所不同。

配置碼	定義	極性	電壓範圍
0	PCIELM4_AI_BI_227MV	Bipolar	+/- 227 mV
0	PCIELM4_AI_BI_10V	Bipolar	+/- 10V
1	PCIELM4_AI_BI_5V	Bipolar	+/- 5V
2	PCIELM4_AI_BI_2V5	Bipolar	+/- 2.5V
3	PCIELM4_AI_BI_1V25	Bipolar	+/- 1.25V

### A.3.2. 類比輸出配置碼(電壓)

使用者可查詢下表在類比輸出函式集設定板卡類比輸出的電壓範圍及極性，每張板卡支援的類比輸出電壓範圍及極性皆有所不同。

配置碼	定義	電壓範圍
0	PCIELM4_AO_UNI_5V	0 ~ 5V
1	PCIELM4_AO_UNI_10V	0 ~ 10V
2	PCIELM4_AO_BI_5V	+/- 5V
3	PCIELM4_AO_BI_10V	+/- 10V

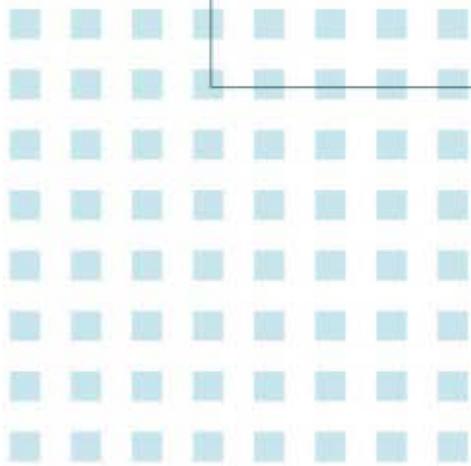
## A.4. 數位輸入埠定義號碼

DI 埠號	PCIe-LM4
0	DI 0 ~ 15

## A.5. 數位輸出埠定義號碼

DO 埠號	PCIe-LM4
0	DO 0 ~ 15

# B



## 附錄 B. 其他

本章節將會提供一些其他的補充資料。

## B.1. 常見問題集

### 系統與安裝

---

Q. PCIELM4 支援 64-bit 的作業系統嗎？

A. 支援，由於 PCIELM4 驅動程式函式庫支援 64 位元作業系統。

---

Q. PCIELM4 支援 ISA 總線的板卡嗎？

A. PCIELM4 目前尚未支援任何 ISA 總線的板卡。

---

### 類比輸出

---

Q. 使用類比輸出函式輸出一電壓為何輸出不正確的電壓。

A. 請您檢查您的電壓範圍設定是否正確，並利用 `PCIELM4_ConfigAO` 設定正確輸出範圍，再使用 `PCIELM4_WriteAOVoltage` 函式輸出電壓。

---

## 函式錯誤碼故障排除

---

Q. 錯誤碼 1.

A. 請重新安裝泓格 PCIELM4 驅動函式庫或重新開機。

---

Q. 錯誤碼 2.

A. (1)請在使用 PCIELM4 函式前使用 PCIELM4\_DriverInit 作初始化的動作。  
(2)wBoardNo 有誤，請重新檢查 wBoardNo。

---

Q. 錯誤碼 5.

A. wBoardNo 有誤，請重新檢查 wBoardNo。

---

Q. 錯誤碼 6.

A. 未找到任何板卡，請您安裝泓格板卡再開始程式。

---

Q. 錯誤碼 13

A. 此板卡不支援此函式功能。

---

Q. 錯誤碼 19

A. 請設定正確的類入輸入範圍。

## B.2. 版本修改資訊

Revision	Date	Description
1.0	Feb. 2020	初版