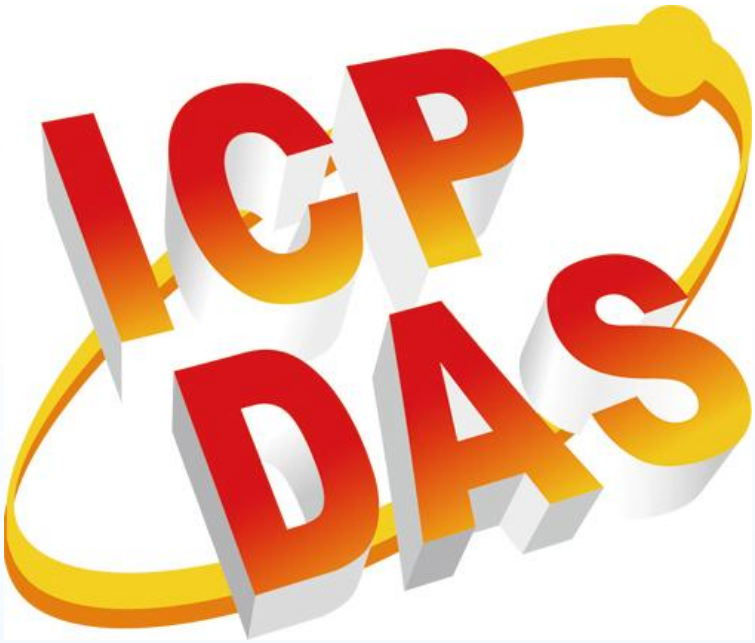


泓格

Driver & SDK

UNiDAQ

# 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 8

支援大多數 PCI I/O 板卡

## ➤ 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



## ➤ 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 UniDAQ 驅動函式庫在 Windows 下對泓格數據採集板卡作 I/O 操作。本手冊提供了使用泓格板卡的相關訊息。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 UniDAQ 函式庫驅動在 Windows 系統下使用 VB、VC、BCB、Delphi、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 UniDAQ 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咨詢泓格技術工程師。

Email: service@icpdas.com

# Table of Contents


Table of Contents ..... 3

1. 導讀 ..... 8
   1.1. 關於泓格 UniDAQ 驅動程式 ..... 9
   1.2. 支援的泓格產品 ..... 10
   1.3. 系統需求 ..... 11

2. 開始安裝使用 ..... 12
   2.1. 取得 UniDAQ 驅動函式庫安裝程式 ..... 13
   2.2. 安裝 UniDAQ 驅動程式函式庫 ..... 14
   2.3. 移除 UniDAQ 驅動函式庫 ..... 19

3. 開發指南 ..... 20
   3.1. 應用程式架構 ..... 21
   3.2. 在 Win32 Console ..... 22
   3.3. 在 Visual Basic 6.0 ..... 25
   3.4. 在 Borland Delphi ..... 28
   3.5. 在 Borland C++ Builder ..... 31
   3.6. 在 Visual C++.NET ..... 34
   3.7. 在 Visual Basic.NET ..... 40
   3.8. 在 Visual C#.NET ..... 46
   3.9. 範例程式及文件 ..... 52

4. 函式應用 ..... 53
   4.1. 導讀 ..... 54
   4.2. 驅動函式庫 ..... 56
   4.3. 數位輸出輸入 ..... 58
      4.3.1. 數位輸入 ..... 59

Table Of Contents

Industrial Communication Products

4.3.2. 數位輸出 ..... 60
4.4. 類比輸入 ..... 61
4.5. 類比輸出 ..... 72
4.6. 計時計數器 ..... 74
4.7. 記憶體存取 ..... 75

**5. 函式參考 ..... 76**

5.1. 函式支援列表 ..... 77
5.2. 函式介紹 ..... 87
5.2.1. 驅動函式集 ..... 88
Ixud_GetDllVersion ..... 88
Ixud_DriverInit ..... 88
Ixud_DriverClose ..... 89
Ixud_SearchCard ..... 89
Ixud_GetBoardNoByCardID ..... 90
Ixud_GetCardInfo ..... 91
Ixud_ReadPort ..... 92
Ixud_WritePort ..... 93
Ixud_ReadPort32 ..... 94
Ixud_WritePort32 ..... 95
Ixud_ReadPhyMemory ..... 96
Ixud_WritePhyMemory ..... 97
5.2.2. 數位輸出輸入函式集 ..... 98
Ixud_SetDIOModes32 ..... 98
Ixud_SetDIOMode ..... 99
Ixud_ReadDI ..... 100
Ixud_WriteDO ..... 101
Ixud_ReadDIBit ..... 102
Ixud_WriteDOBit ..... 103
Ixud_ReadDI32 ..... 104

Table Of Contents

Ixud_WriteDO32 ..... 105
Ixud_SoftwareReadbackDO ..... 106
Ixud_StartDI ..... 107
Ixud_StartDO ..... 108
Ixud_GetDIBufferH ..... 110
Ixud_StopDI ..... 111
Ixud_StopDO ..... 112
5.2.3. 中斷事件函式集 ..... 113
Ixud_SetEventCallback ..... 113
Ixud_RemoveEventCallback ..... 116
Ixud_InstallIrq ..... 116
Ixud_RemoveIrq ..... 118
5.2.4. 類比輸入函式集 ..... 119
Ixud_ConfigAI ..... 119
Ixud_ConfigAIEx ..... 121
Ixud_ClearAIBuffer ..... 122
Ixud_ClearAIBuffer ..... 124
Ixud_GetBufferStatus ..... 124
Ixud_ReadAI ..... 126
Ixud_ReadAIH ..... 127
Ixud_PollingAI ..... 128
Ixud_PollingAIH ..... 129
Ixud_PollingAIScan ..... 130
Ixud_PollingAIScanH ..... 132
Ixud_StartAI ..... 134
Ixud_StartAIScan ..... 136
Ixud_StartExtAI ..... 138
Ixud_StartExtAnalogTrigger ..... 140
Ixud_StartExtAIScan ..... 142
Ixud_GetAIBuffer ..... 144

Ixud_GetAIBufferH ..... 145
Ixud_StopAI ..... 146
5.2.5. 類比輸出函式集 ..... 147
Ixud_ConfigAO ..... 147
Ixud_WriteAOVoltage ..... 148
Ixud_WriteAOVoltageH ..... 148
Ixud_WriteAOCurrent ..... 149
Ixud_WriteAOCurrentH ..... 150
Ixud_StartAOVoltage ..... 151
Ixud_StartAOVoltageH ..... 152
Ixud_StopAO ..... 154
5.2.6. 計時計數函式集 ..... 155
Ixud_DisableCounter ..... 155
Ixud_ReadCounter ..... 155
Ixud_ReadFrequency ..... 156
Ixud_SetCounter ..... 157
Ixud_SetFCChannelMode ..... 158
5.2.7. 記憶體輸出輸入函式集 ..... 160
Ixud_ReadMemory ..... 160
Ixud_WriteMemory ..... 161
Ixud_ReadMemory32 ..... 162
Ixud_WriteMemory32 ..... 162
5.3. 資料型態 ..... 164
PIXUD_DEVICE_INFO ..... 164
PIXUD_CARD_INFO ..... 166

**附錄 A. 函式回傳值與配置碼 ..... 169**

A.1. 函數回傳值定義 ..... 170
A.2. 模組識別號碼 ..... 172
A.3. 配置碼定義 ..... 174

A.3.1. 類比輸入配置碼 ..... 175
A.3.2. 類比輸出配置碼(電壓)..... 178
A.3.3. 類比輸出配置碼(電流)..... 179
A.3.4. 中斷事件配置碼 ..... 180
A.4. 數位輸入埠定義號碼 ..... 181
A.5. 數位輸出埠定義號碼 ..... 183

**附錄 B. 其他..... 185**

B.1. 常見問題集..... 186
B.2. 版本修改資訊 ..... 189

# 1. 導讀

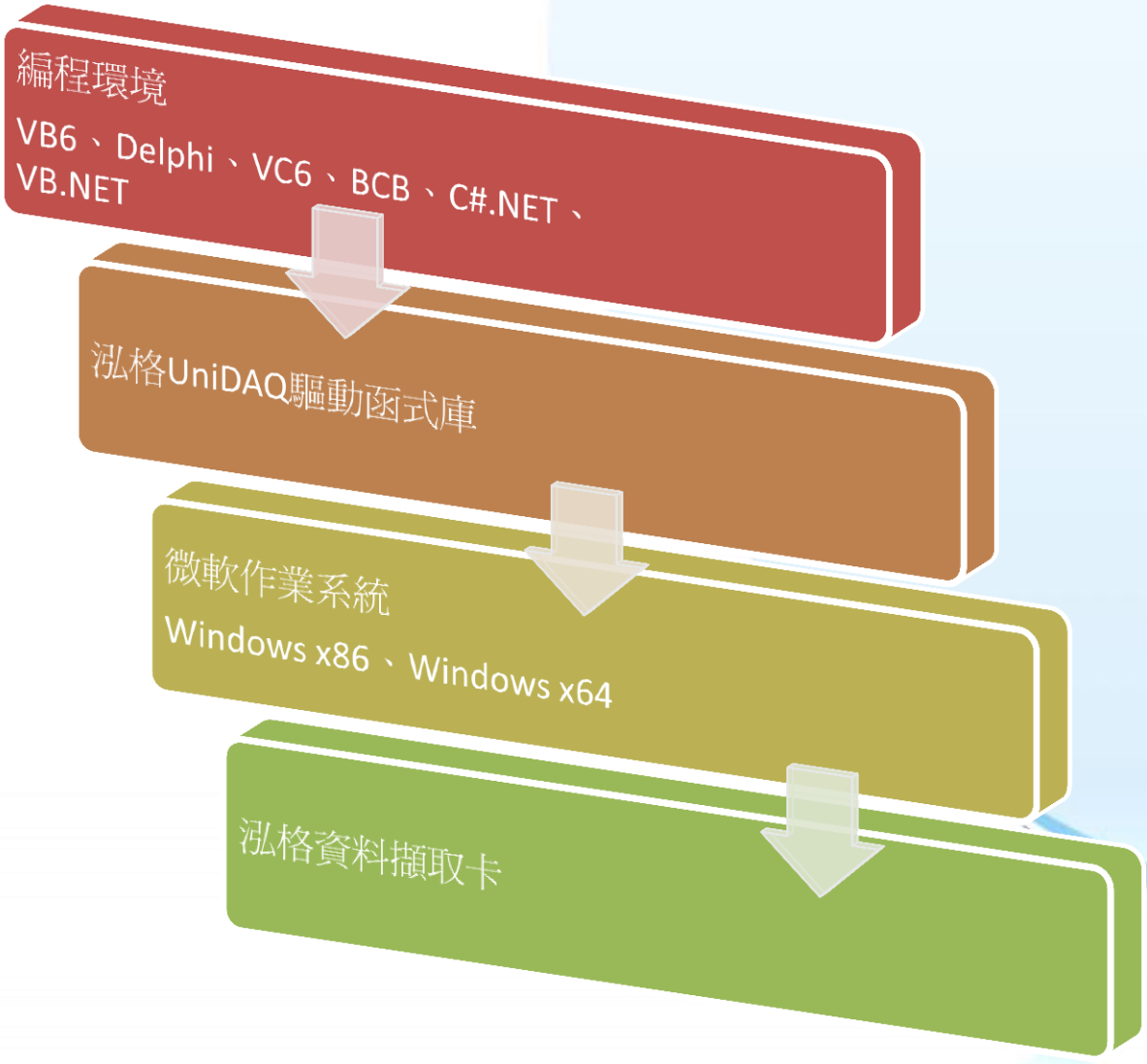
本章節將會簡單介紹泓格 UniDAQ 驅動程式庫的功能及系統需求

# 1.1. 關於泓格 UniDAQ 驅動程式

泓格 UniDAQ 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 UniDAQ 驅動函式庫裡，不需使用特定的硬體暫存器命令，UniDAQ 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 UniDAQ 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，另外支援中斷及事件通知功能，當硬體中斷事件發生時，會透過應用程式通知使用者來採取必要的行動，無需手動檢查硬體狀態，有效的減少程式的複雜度及大幅提昇設備的即時可靠性。

泓格 UniDAQ 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：

| 產品型號 | 產品型號 |
|---|---|
| PIO-D24/D56/D24U/D56U、PEX-D24/D56 | PIO-D48/D48U/D48SU、PEX-D48 |
| PIO-D64/D64U | PIO-D96/D96U/D96SU |
| PIO-D144/D144U | PIO-D168/D168U |
| PIO-DA4/DA8/DA16/DA4U/DA8U/DA16U | PISO-DA4U/DA8U/DA16U |
| PEX-DA4/DA8/DA16 | PIO-821L/821H/821LU/821HU |
| PISO-C64/C64U/P64/P64U | PEX-C64/P64 |
| PISO-A64/P32A32/P32A32U | PISO-P32C32/P32C32U/P32S32WU |
| PEX-P32C32 | PISO-P8R8/P8R8U |
| PISO-P8R8AC/P8R8DC | PISO-P16R16U、PEX-P16R16i/P8R8i |
| PISO-730/730A/730U/730AU | PISO-725 |
| PISO-DA2/DA2U | PISO-813/813U |
| PCI-TMC12/PCI-TMC12A | PCI-M512/M256/M128/M512U |
| PCI-P16R16/P16C16/P16POR16/P8R8 | PEX-P16ROR16i/P8POR8i |
| PCI-1002L/1002H/1002LU/1002HU | PCI-1202L/1202H/1202LU/1202HU |
| PEX-1002L/1002H | PEX-1202L/1202H |
| PCI-1602/1602U,PCI-1602F/1602FU | PCI-1800L/1800H/1800LU/1800HU |
| PCI-1802L/1802H/1802LU/1802HU | PCI-822LU/826LU |
| PCI-FC16U | PCI-2602U |

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：

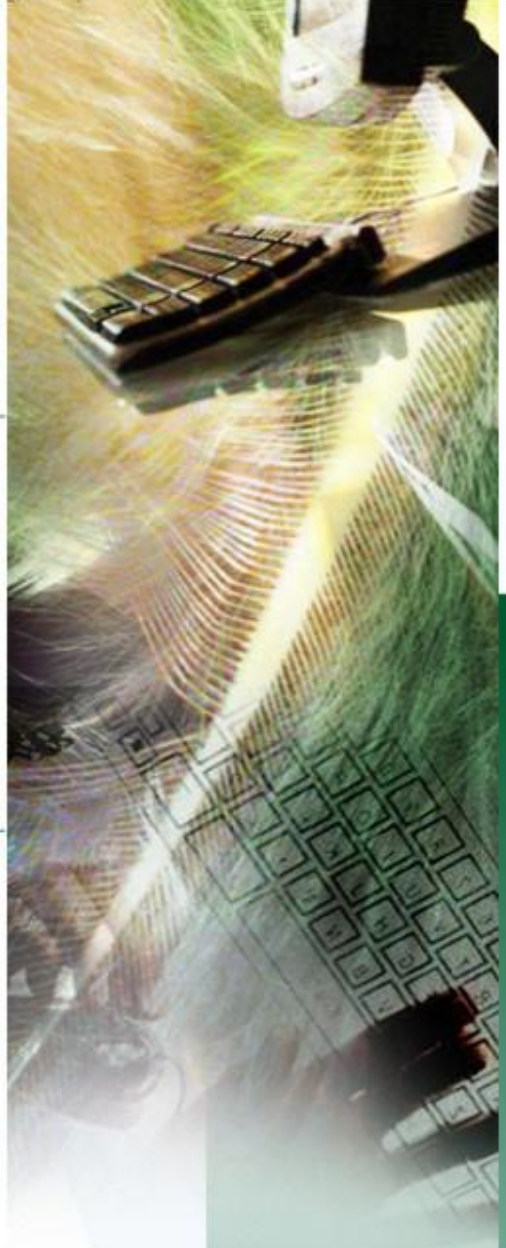
- 使用 266MHz 或更快的 32 位元(x86)或 64 位元(x64)處理器
- 至少 64 MB 的記憶體空間
- 相容於 VGA 的圖型顯示卡
- 至少 20 MB 磁碟空間
- 一台相容 DVD/CD-ROM
- Microsoft Windows 2000 以上的 32 位元或 64 位元作業系統

支援以下的 32 及 64 位元的 Windows 作業系統

| 32 位元(x86) | 64 位元(x64) |
|---|---|
| **Microsoft Windows 2000** | **-** |
| **Microsoft Windows XP 32-bit** | **Microsoft Windows XP 64-bit** |
| **Microsoft Windows 2003 32-bit** | **Microsoft Windows 2003 64-bit** |
| **Microsoft Windows Vista 32-bit** | **Microsoft Windows Vista 64-bit** |
| **Microsoft Windows 7 32-bit** | **Microsoft Windows 7 64-bit** |
| **Microsoft Windows 2008 32-bit** | **Microsoft Windows 2008 64-bit** |
| **Microsoft Windows 8 32-bit** | **Microsoft Windows 8 64-bit** |
| **-** | **Microsoft Windows 2012 64-bit** |

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 UniDAQ 驅動函式庫安裝程式

取得 UniDAQ 驅動函式庫安裝程式的方法可以從板卡上附的 CD 上取得或是從網路上下載，取得路徑請參考下表：

| | |
|---|---|
|  | CD:\\ NAPDOS\PCI\UniDAQ\DLL\Driver |
|  | http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/driver/ |
|  | ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/driver/ |

# 2.2. 安裝 UniDAQ 驅動程式函式庫

## 步驟一安裝資料擷取板卡

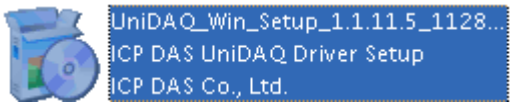
**請依照下列步驟安裝板卡:**

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

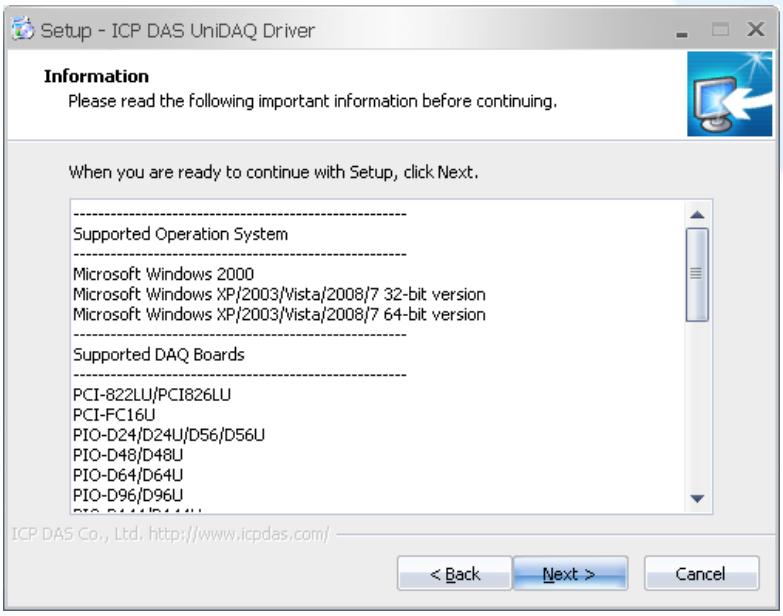
## 請依照下列步驟執行安裝

1.雙擊 UniDAQ_Win_Setup…安裝驅動函式庫。

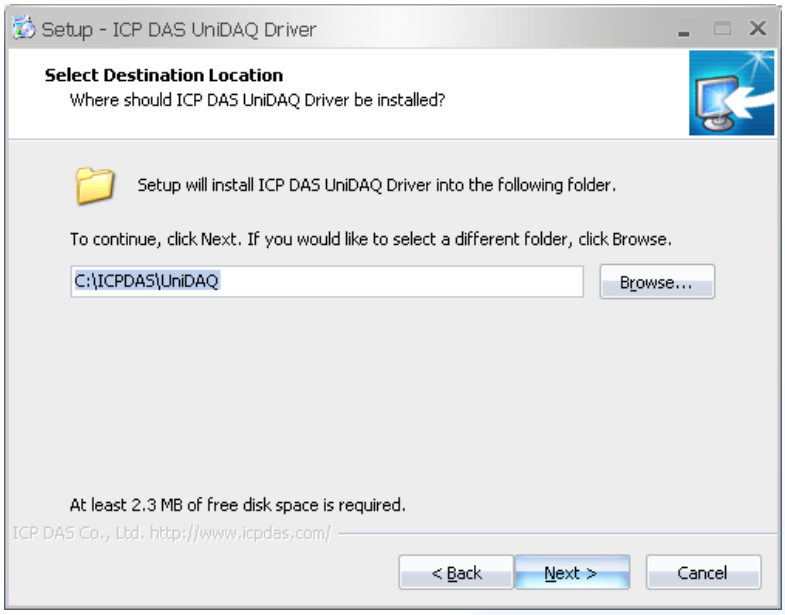



2.按 Next>到下一個畫面。

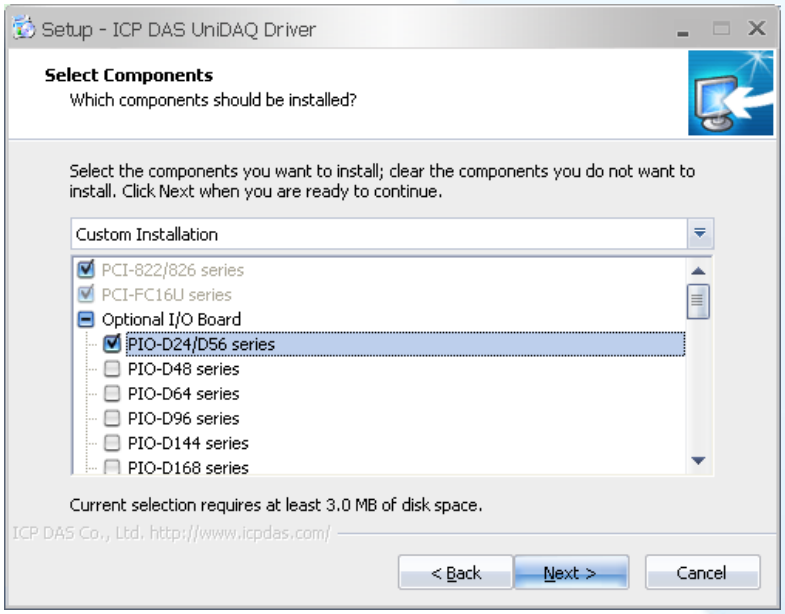

3.檢查您的板卡及系統是否在支援內，按 Next>到下一個畫面。



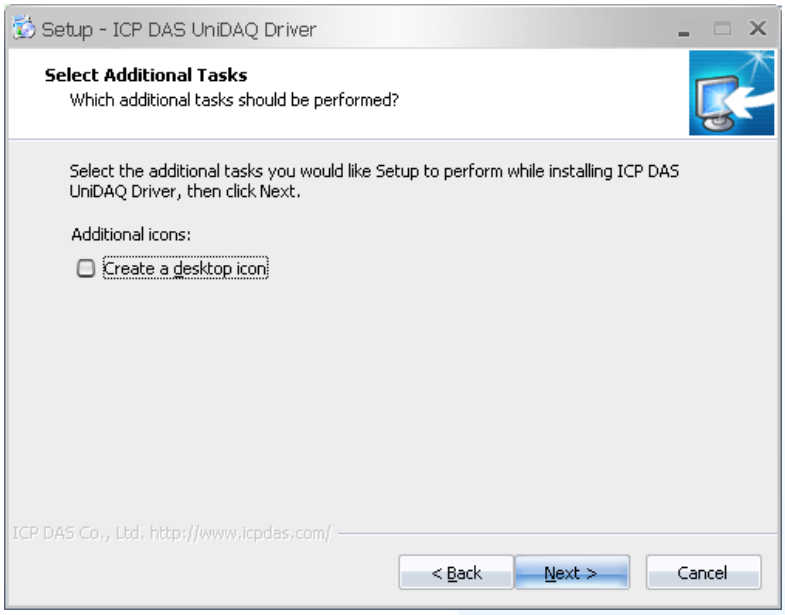
4.選擇安裝目錄，預設爲 C:\ICPDAS\UniDAQ，確認後按 Next>到下一個畫面。



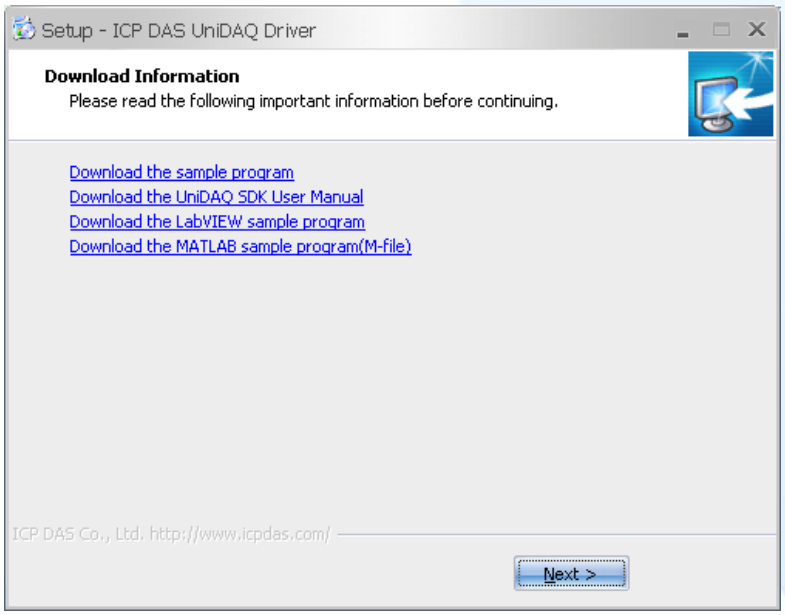

5.在列表內勾選您所需要安裝驅動程式的板卡，勾選完後按 Next>到下一個畫面。



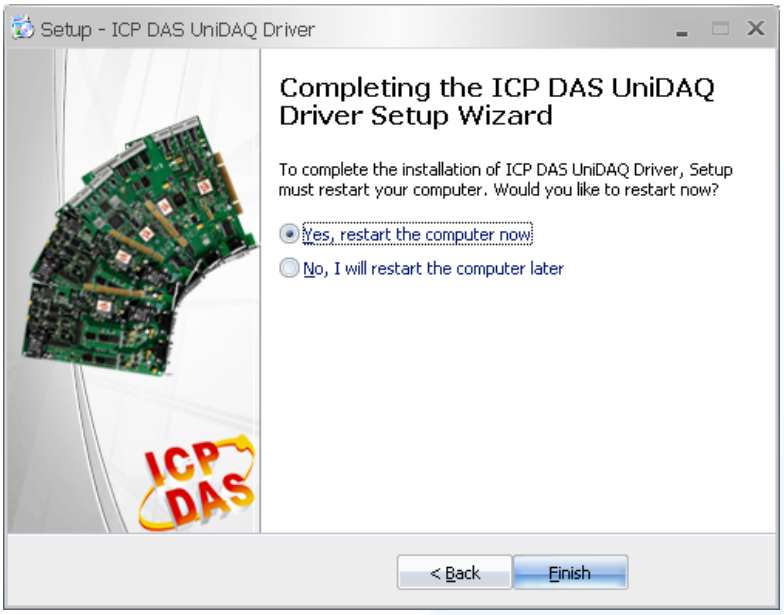
6. 按 Next>到下一個畫面。




7. 按 Next>到下一個畫面。



8.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 UniDAQ 驅動函式庫安裝完成。

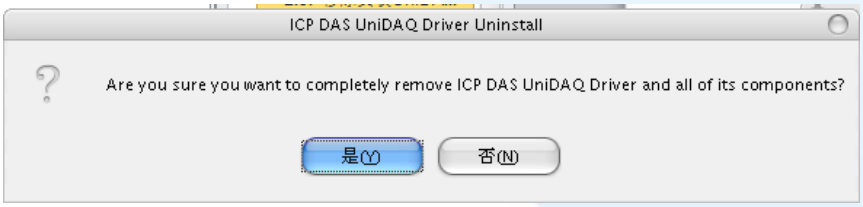


# 2.3. 移除 UniDAQ 驅動函式庫

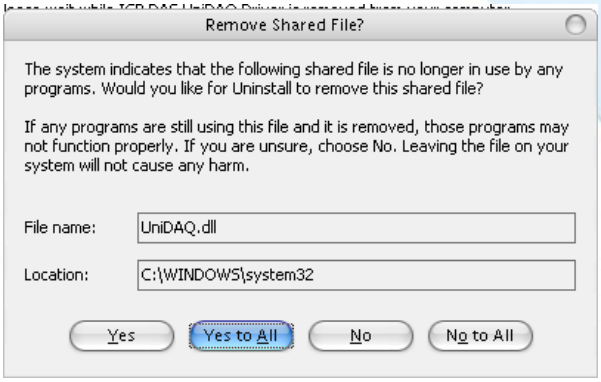
泓格驅動函式庫包括反安裝工具來協助您從電腦上移除軟體，如果您想要移除軟體請完成下列的流程來執行反安裝工具。

1. 至設定|控制台|新增或移除程式下。
2. 在選單列表上選擇 ICP DAS UniDAQ Windows 項目，並點擊最右方的移除按鍵。


3. 將會跳出一個對話框，並選擇是(Y)開始執行反安裝。



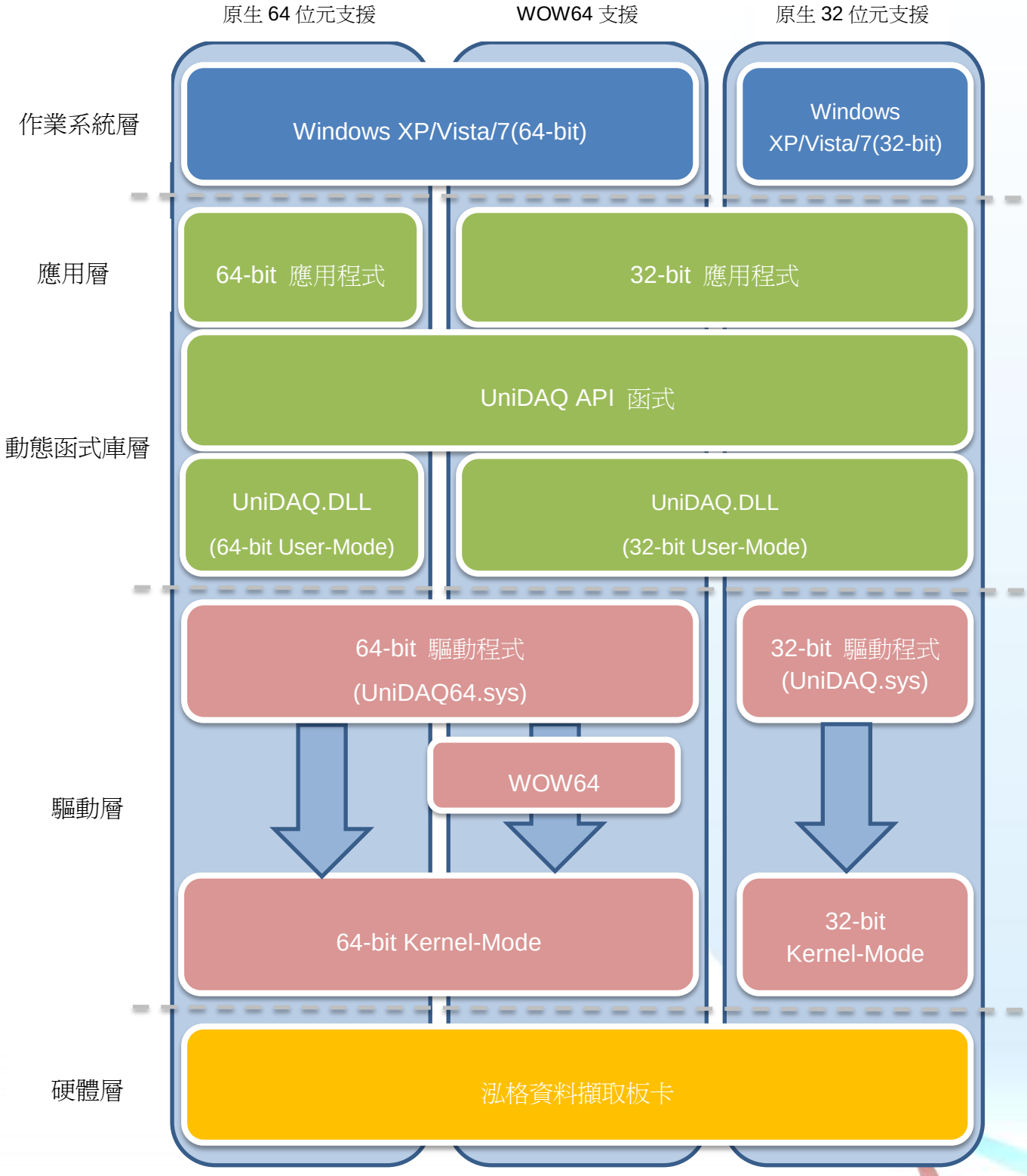

4. 點擊 Yes to All 完全移除 UniDAQ.dll 檔案，之後將會完成移除軟件的動作。



# 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，Delphi， BCB，Visual Studio.NET 及 Visual Studio.NET x64 環境下逐步編寫程式的範例。

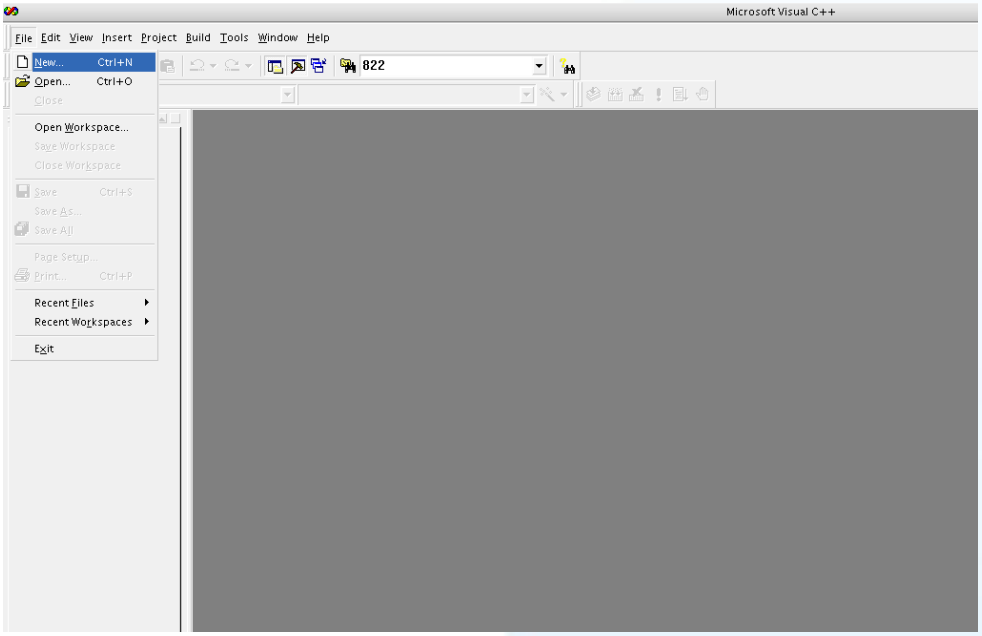
# 3.1. 應用程式架構



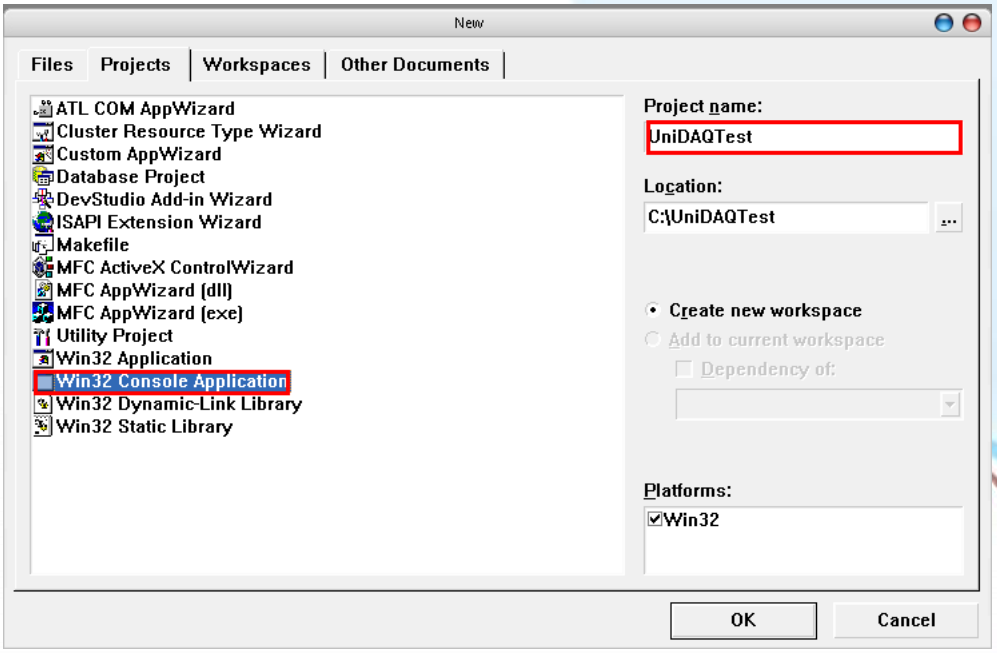
# 3.2. 在 Win32 Console

步驟 1:撰寫應用程式

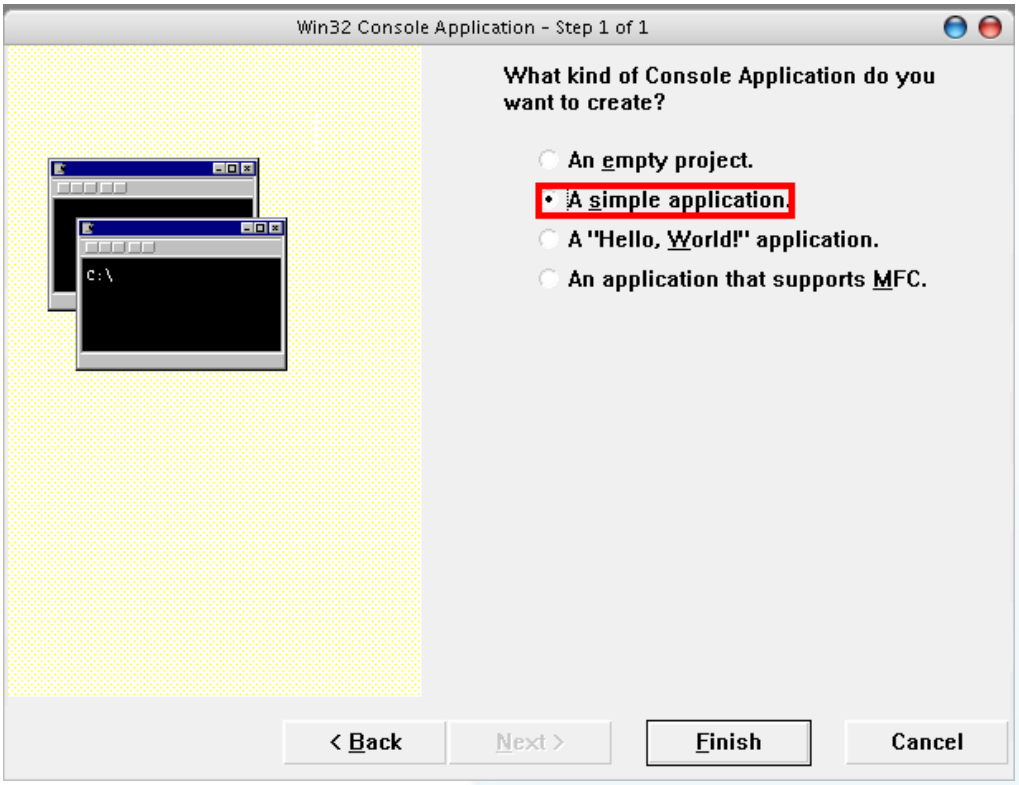
1. 至程式集開啓 Microsoft Visual C++ 6.0
2. 從主要選單內選擇 File|New…



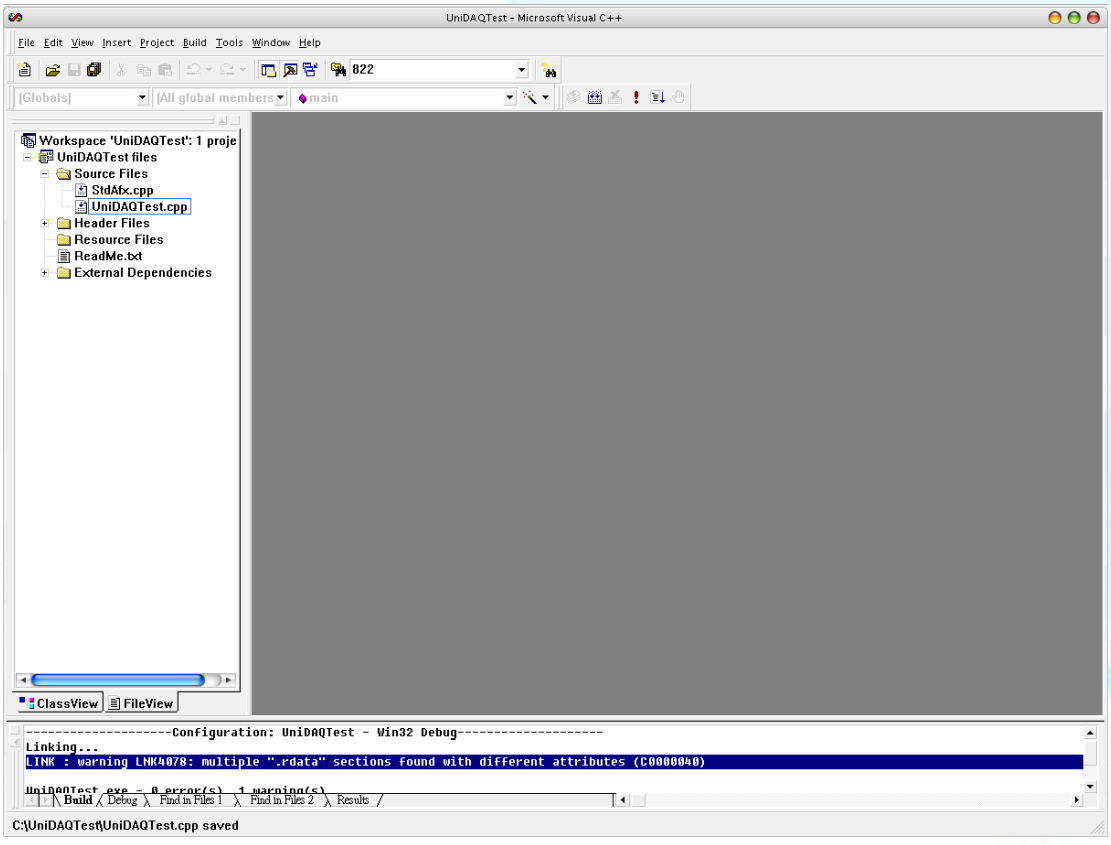

3. 在 dialog box 下的列表點選項目 Win32 Console Application ，並在 Project name 欄位輸入 UniDAQTest，然後按下 OK 按鍵。



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




5. 雙擊 UniDAQTest.cpp 開啓程式碼寫入視窗。



6. 在 UniDAQTest.cpp 填寫程式碼如下：

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

WORD wRtn;
WORD wBoardNo;
WORD wTotalBoards;

int main(int argc, char* argv[])
{
      WORD wOutPortNo;

      //Initial the resource and get total board number form  Driver
      wRtn=Ixud_DriverInit(&wTotalBoards);
      if (wRtn!=Ixud_NoErr)
      {
            printf("\nDriver Init Error(%d)",wRtn);
            return wRtn;
      }
      printf("Write DO Value 0xFF");
      wBoardNo=0;
      wOutPortNo=0;
      //Write DO
      wRtn = Ixud_WriteDO(wBoardNo,wOutPortNo,0xFF);
      //Release the resource from driver
      wRtn = Ixud_DriverClose();
      return 0;
}
```

步驟 2:測試應用程式

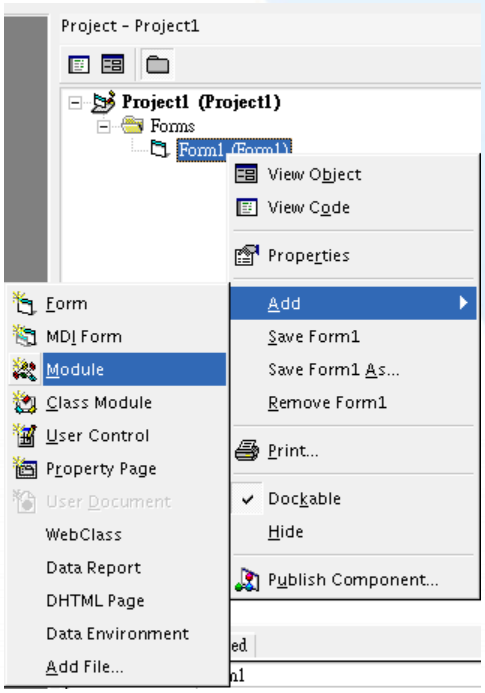
1. 在 Build 選單點擊 Compiler 來編譯程式碼。
2. 立即在 DOS Box 下執行程式。

# 3.3. 在 Visual Basic 6.0

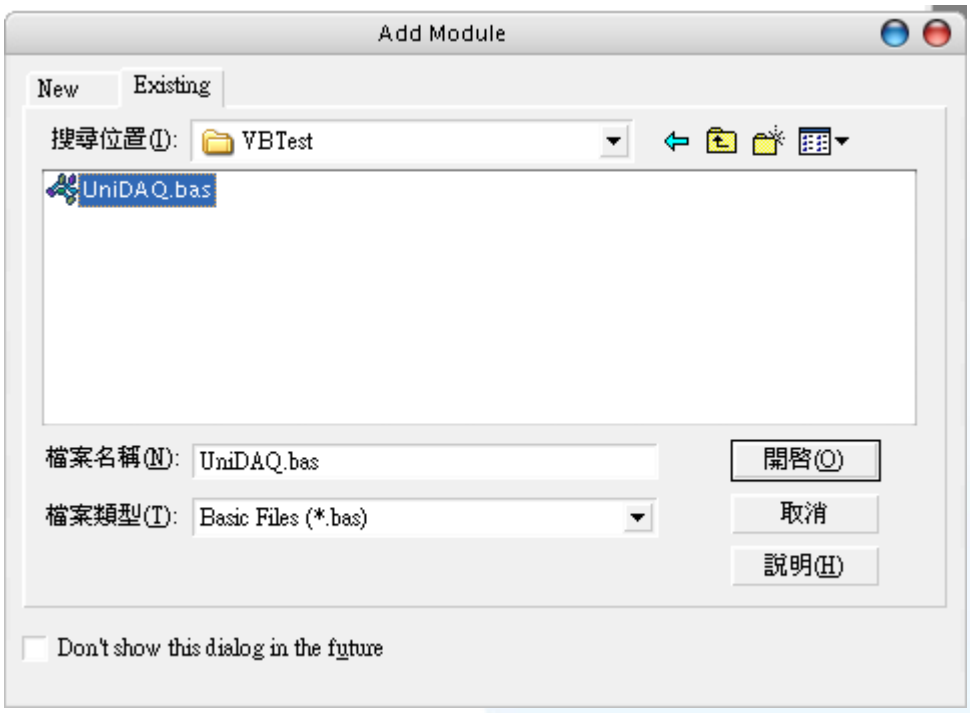
步驟 1:撰寫應用程式

1. 至程式集開啓 Microsoft Visual Basic 6.0
2. 選擇 Standard EXE 圖示並按下開啓按鍵後將建立一個新的專案。

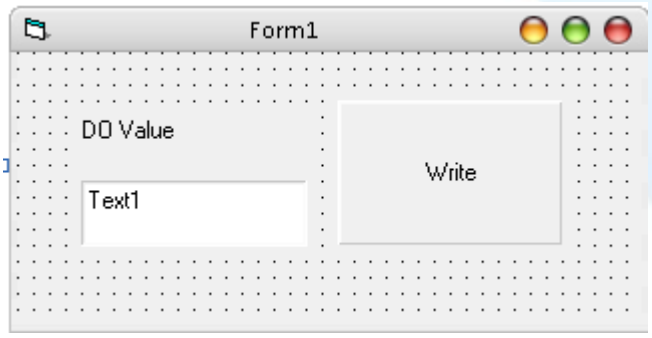

3. 在 Project explorer 開啓 Add Module 視窗。



4. 在 Add Module 選擇 Existing 後添加宣告檔 UniDAQ.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 屬性爲 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 wOutPortNo 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 = Ixud_DriverInit(wTotalBoards)
If (wRtn) Then
    MsgBox ("Driver Initial Error!!Error Code:" + Str(wRtn))
    End
End If

wBoardNo:=0;
wOutportNo =0;

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

‘//Release the resource form Driver
wRtn = Ixud_DriverClose()
End Sub
```

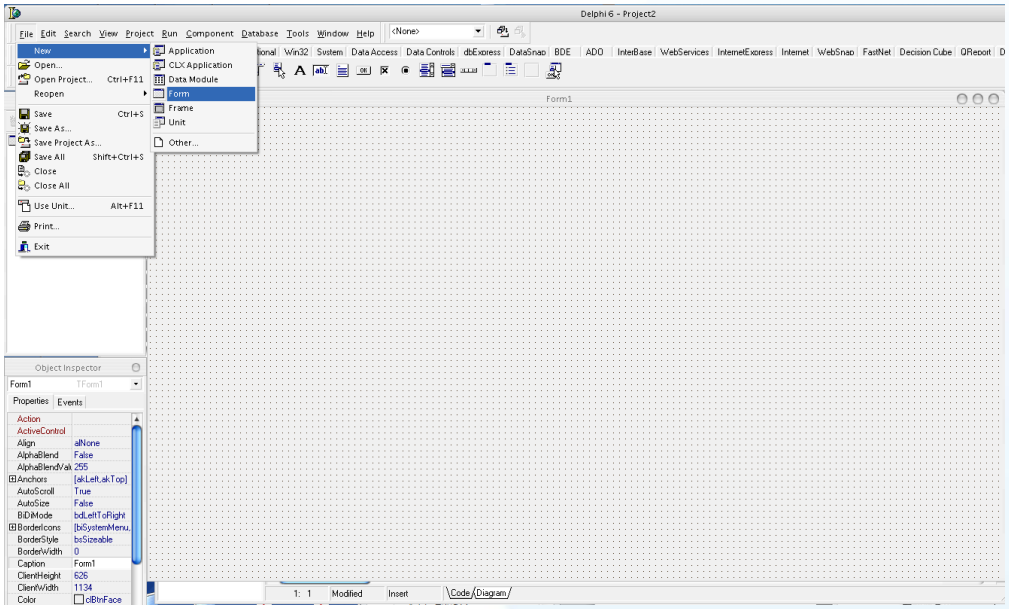
步驟 2:測試應用程式

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。

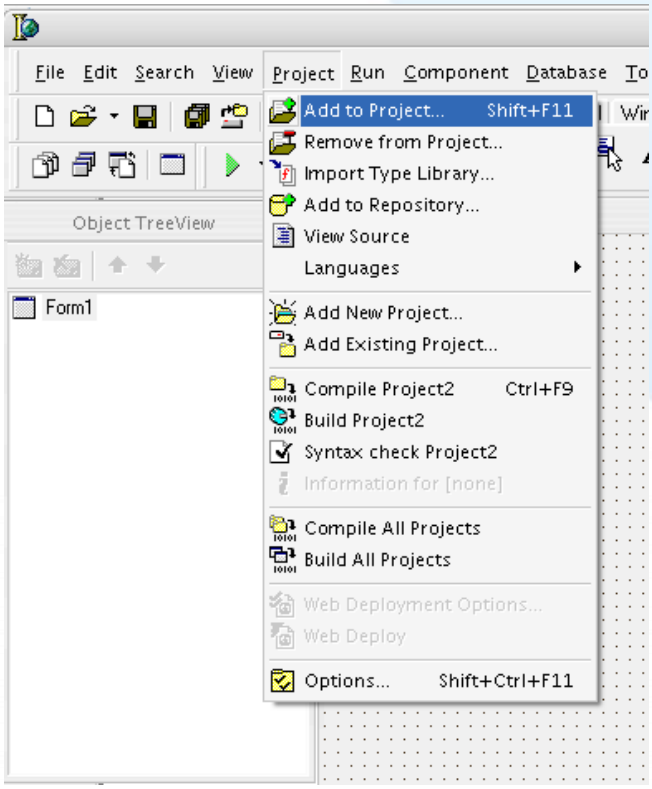
# 3.4. 在 Borland Delphi

步驟 1:撰寫應用程式

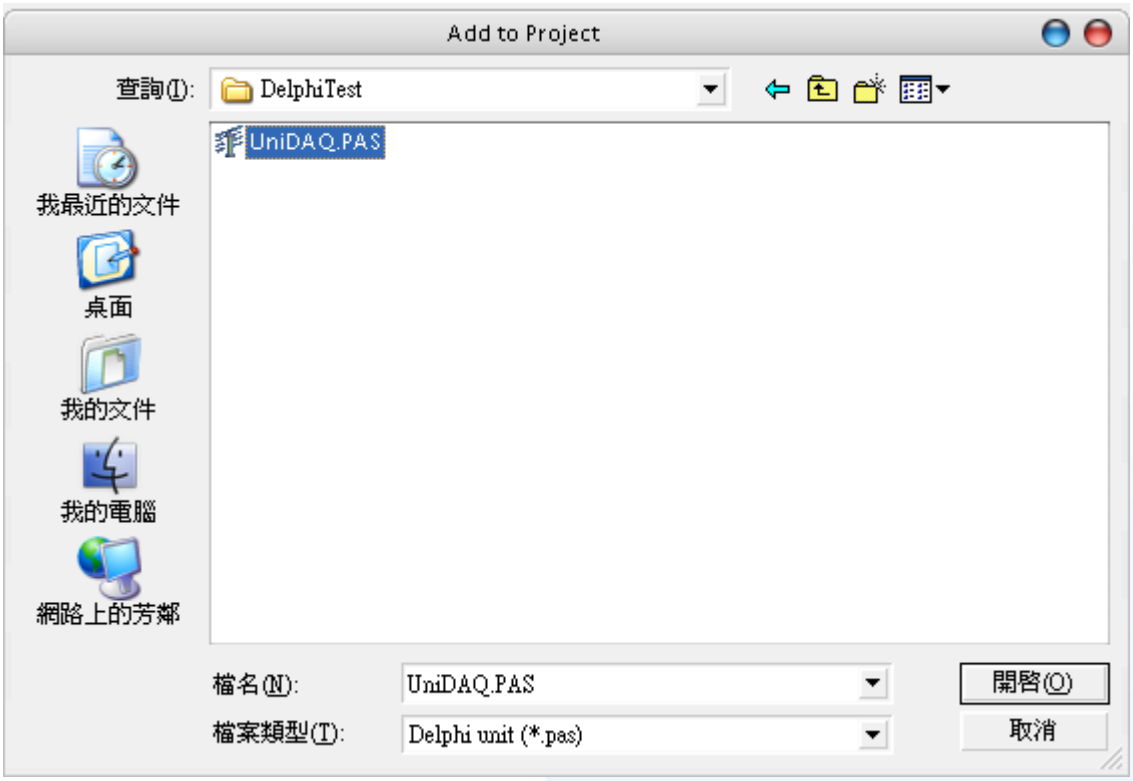
1. 至程式集開啓 Delphi 6.0
2. 從主要選單內選擇 New|Form 將建立一個新的專案。



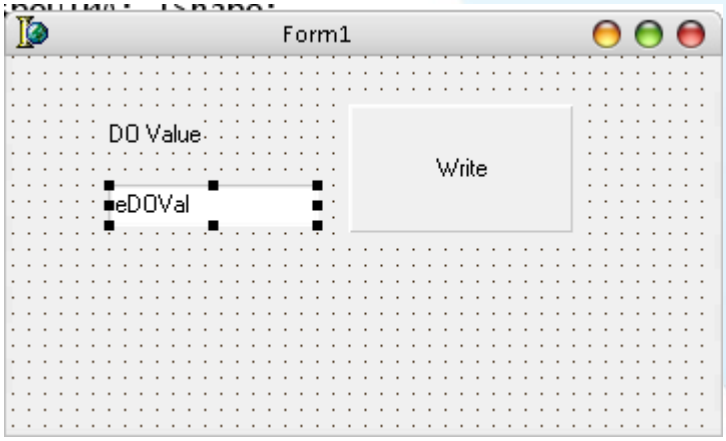

3. 在主要選單內選擇 Project|Add to Project 開啓一視窗。



4. 在 Add to Projcet 視窗移至 UniDAQ.pas 的路徑下選擇 UniDAQ.pas 按下開啓(O) 添加宣告檔至專案裡。




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 Edit 控制項，並切換至屬性視窗上至 Name 屬性輸入 eDOVal，最後放置一個 Button 控制項，並修改 Name 屬性爲 btnWrite 及在 Caption 屬性上輸入 Write。



6. 雙擊 Form1 上的 btnWrite 按鍵控制項，進入程式碼編輯視窗，在 implementation 下加入程式碼如下。

```
implementation
uses UniDAQ;

{$R *.dfm}

procedure TForm1.btnWriteClick(Sender: TObject);
var
   wTotalBoards,wRtn,wBoardNo,wOutportNo:Word;
   dwDOValue : LongInt;
begin
     //Initital resource and get total board number from driver
     wRtn := Ixud_DriverInit(wTotalBoards);
     If wRtn <> Ixud_NoErr Then
     begin
        Application.MessageBox('*** DriverInit Error! ***', 'Error' , IDOK);
        Exit;
     End;
     wBoardNo :=0;
     wOutportNo :=0;

     //Write DO
     wRtn:=Ixud_WriteDO(wBoardNo,wOutportNo,StrToInt(eDOVal.Text));

     //Release the resource from driver
     wRtn := Ixud_DriverClose;

end;

end.
```

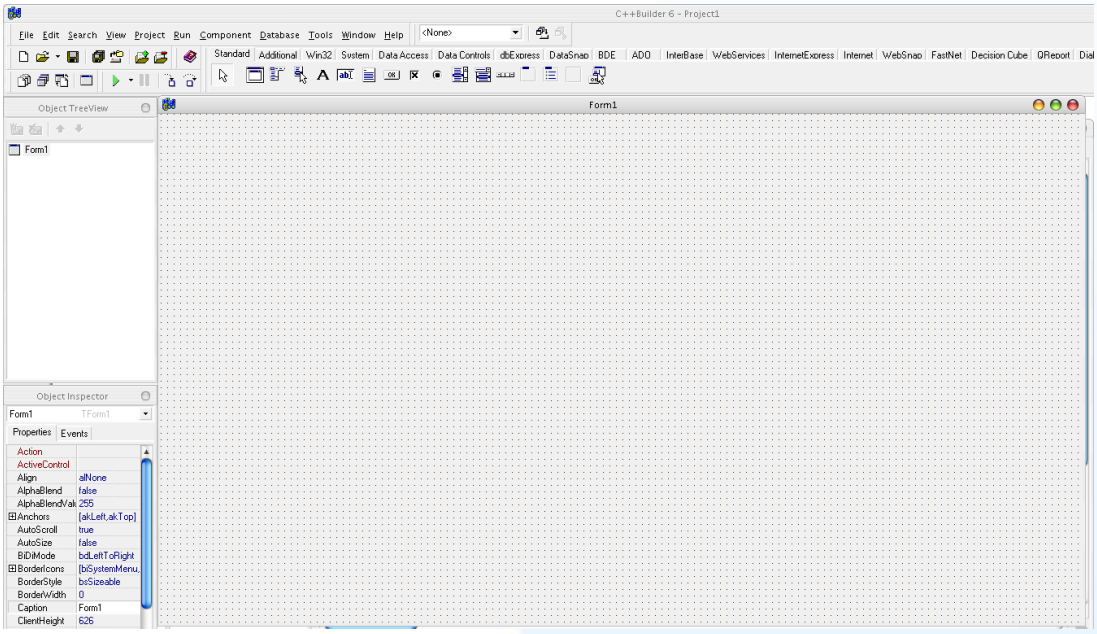
步驟 2:測試應用程式

1. 按下 F9 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。

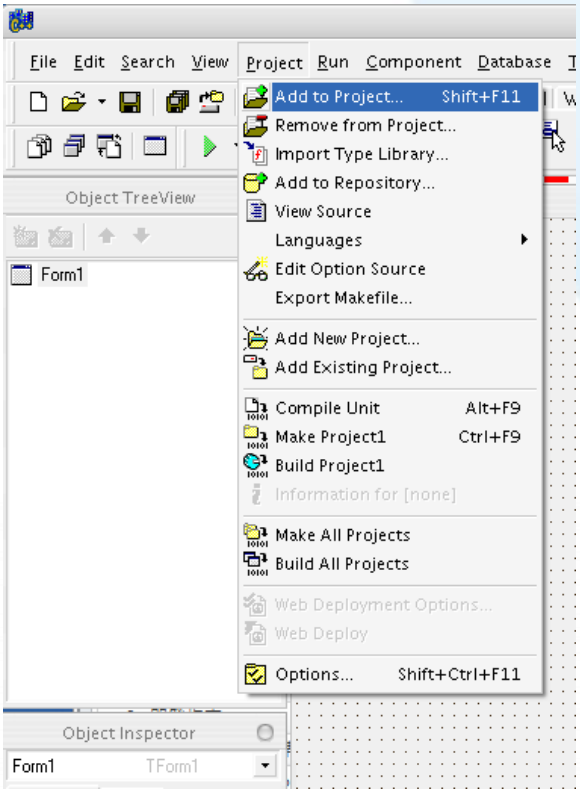
# 3.5. 在 Borland C++ Builder

步驟 1:撰寫應用程式

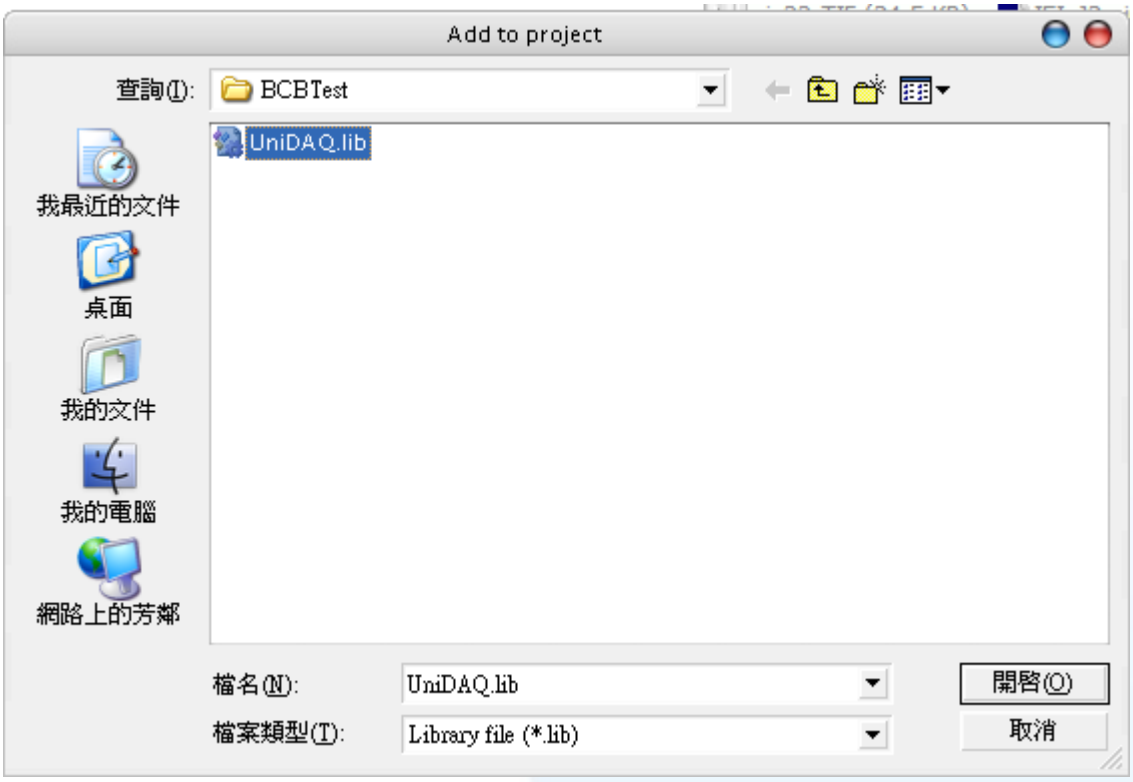
1. 至程式集開啓 C++ Builder 6
2. 從主要選單內選擇 New|Form 將建立一個新的專案。



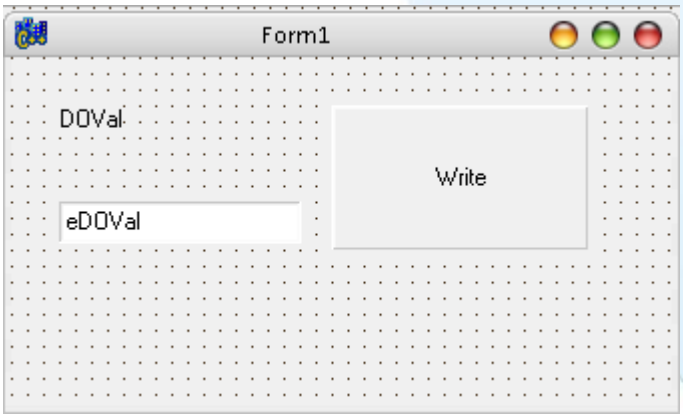
3. 在主要選單內選擇 Project|Add to Project 開啓一視窗。



4. 在 Add to Projcet 視窗移至 UniDAQ.lib 的路徑下選擇 UniDAQ.lib 按下開啓(O)添加宣告檔至專案裡。




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 Edit 控制項，並切換至屬性視窗上至 Name 屬性輸入 eDOVal，最後放置一個 Button 控制項，並修改 Name 屬性爲 btnWrite 及在 Caption 屬性上輸入 Write。



6. 雙擊 Form1 上的 btnWrite 按鍵控制項，進入程式碼編輯視窗，加入程式碼如下。

```
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "UniDAQ.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
void __fastcall TForm1::btnWriteClick(TObject *Sender)
{
Word wTotalBoard, wRtn ;
Word wOutPortNo;
Word wBoardNo;
//Initial the resource and get the total board number from driver
wRtn = Ixud_DriverInit(&wTotalBoard);
if ( wRtn != Ixud_NoErr )
{
    ShowMessage( "Driver Initial Err!!Error Code:" + IntToStr(wRtn)) ;
}
wOutPortNo=0;
wBoardNo=0;
wRtn=Ixud_WriteDO(wBoardNo,wOutPortNo,StrToInt(eDOVal->Text));

//Release the resource from driver
wRtn= Ixud_DriverClose();
}
```

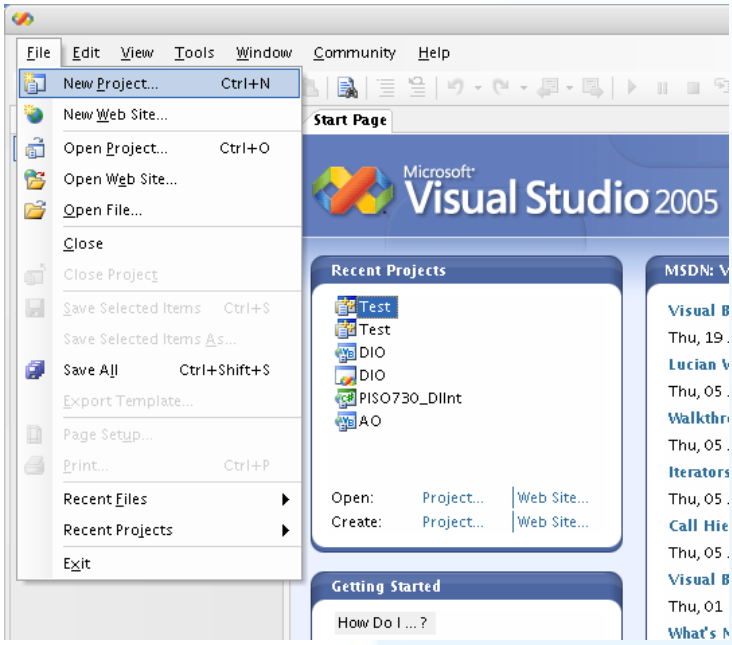
步驟 2:測試應用程式

1. 按下 F9 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。

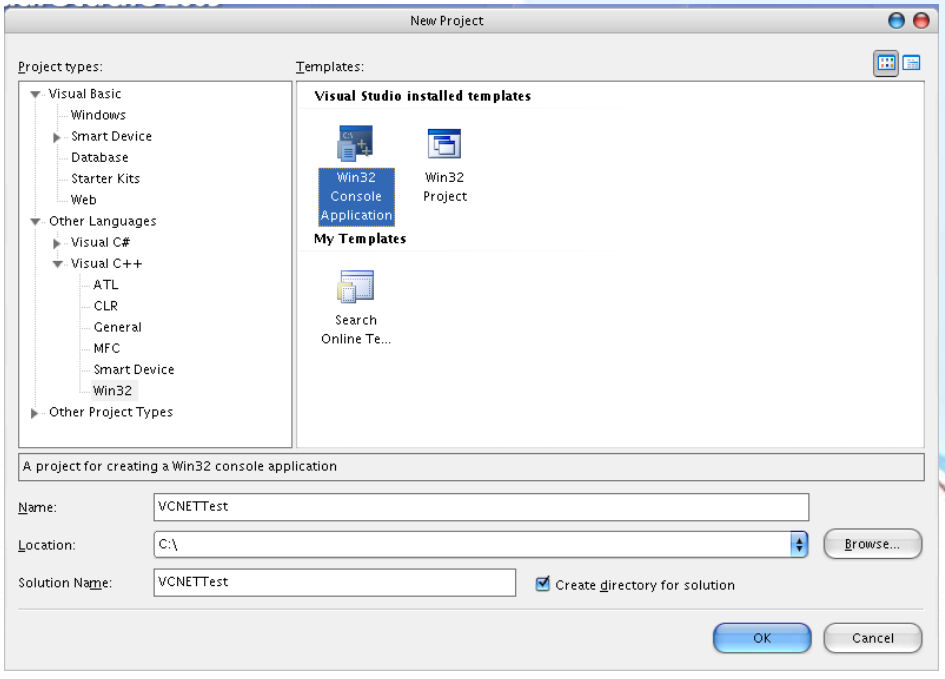
# 3.6. 在 Visual C++.NET

步驟 1:撰寫應用程式

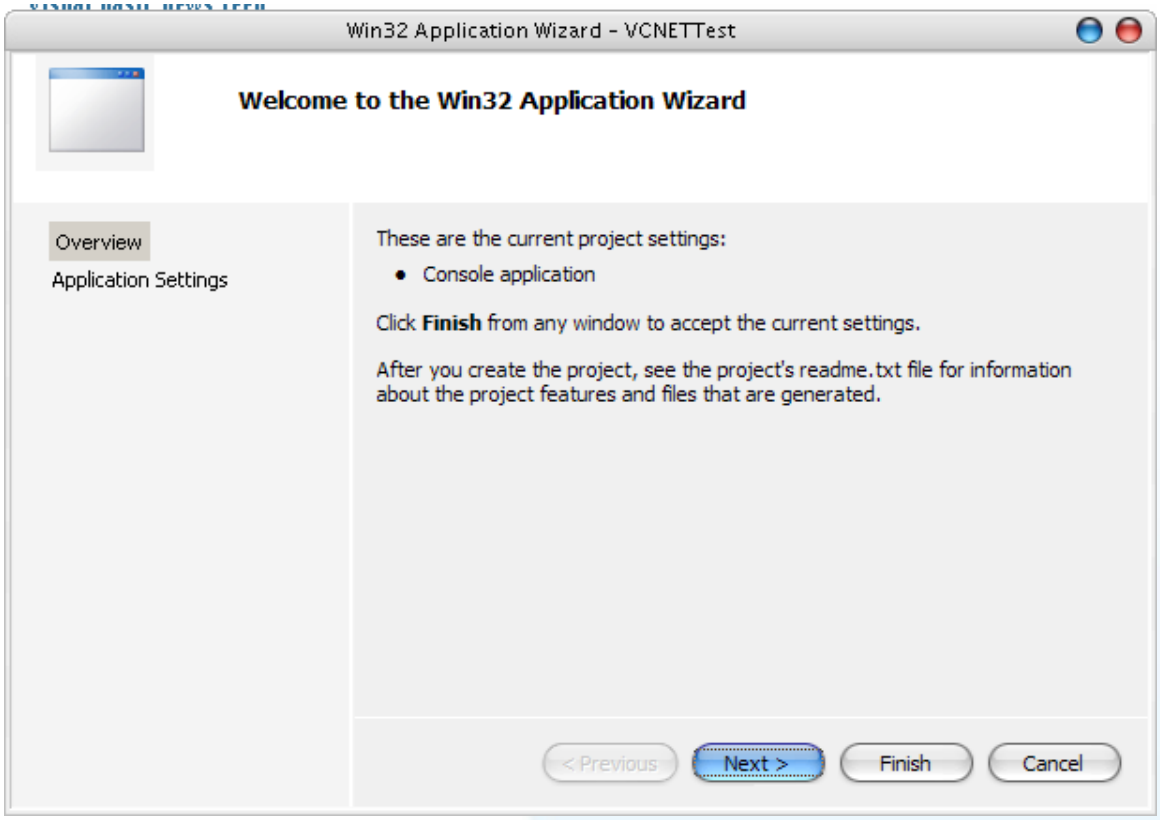
1. 至程式集開啓 Microsoft Visual Studio 2005
2. 從主要選單內選擇 File|New Project…



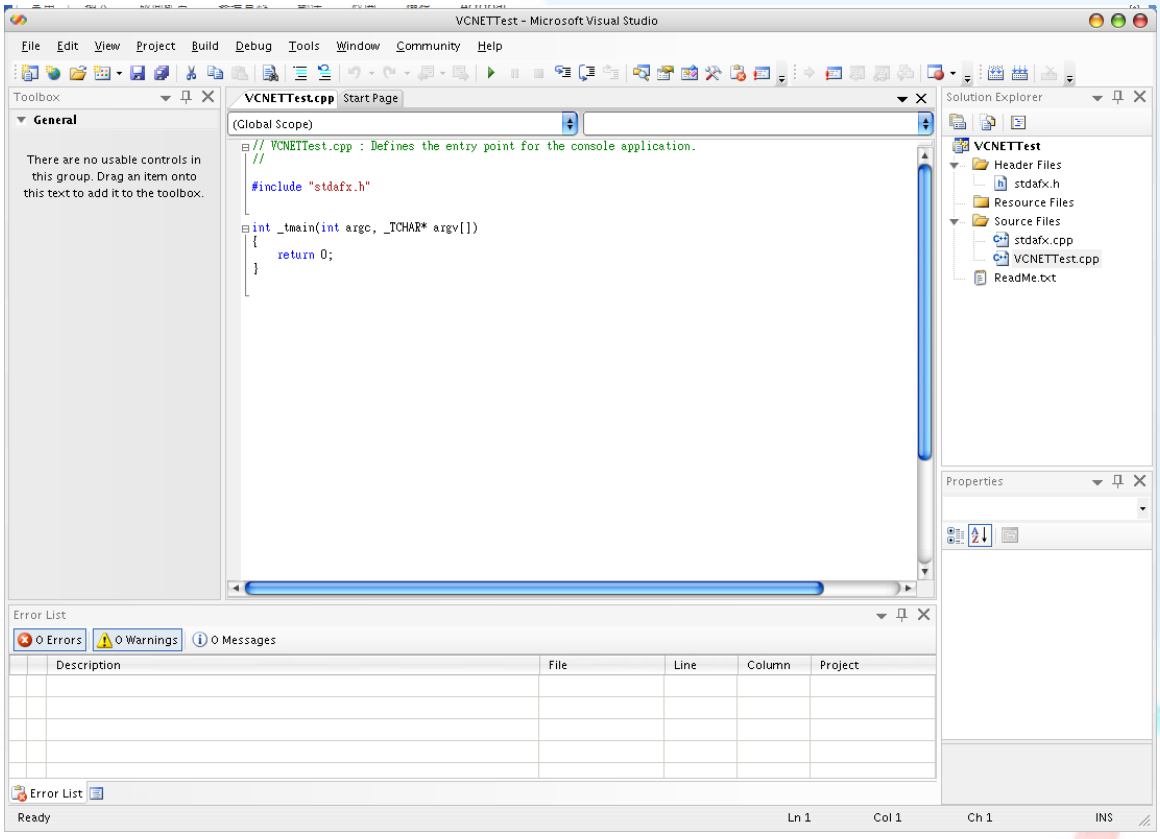

3. 在 Project type 下的列表點選項目 Visual C++ 並在展開選單內選擇 Win32，然後再右方 Templates 表框內選擇 Win32 Console Application，接下來至下方的 Name 鍵入專案名稱 VCNETTest 然後按下 OK 按鍵。



4. 按下 Finish 鍵後，將會產生爲使用者產生最基本的程式碼。




5. 雙擊 VCNETTest.cpp 開啓程式碼寫入視窗。

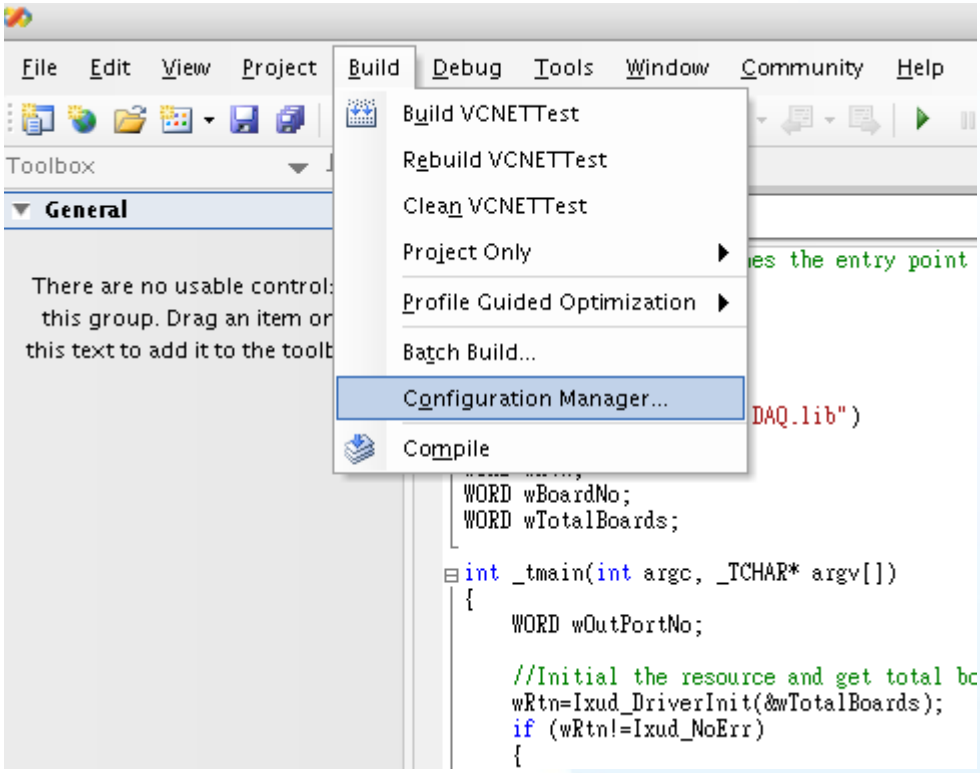


6. 在 VCNETTest.cpp 填寫程式碼如下：

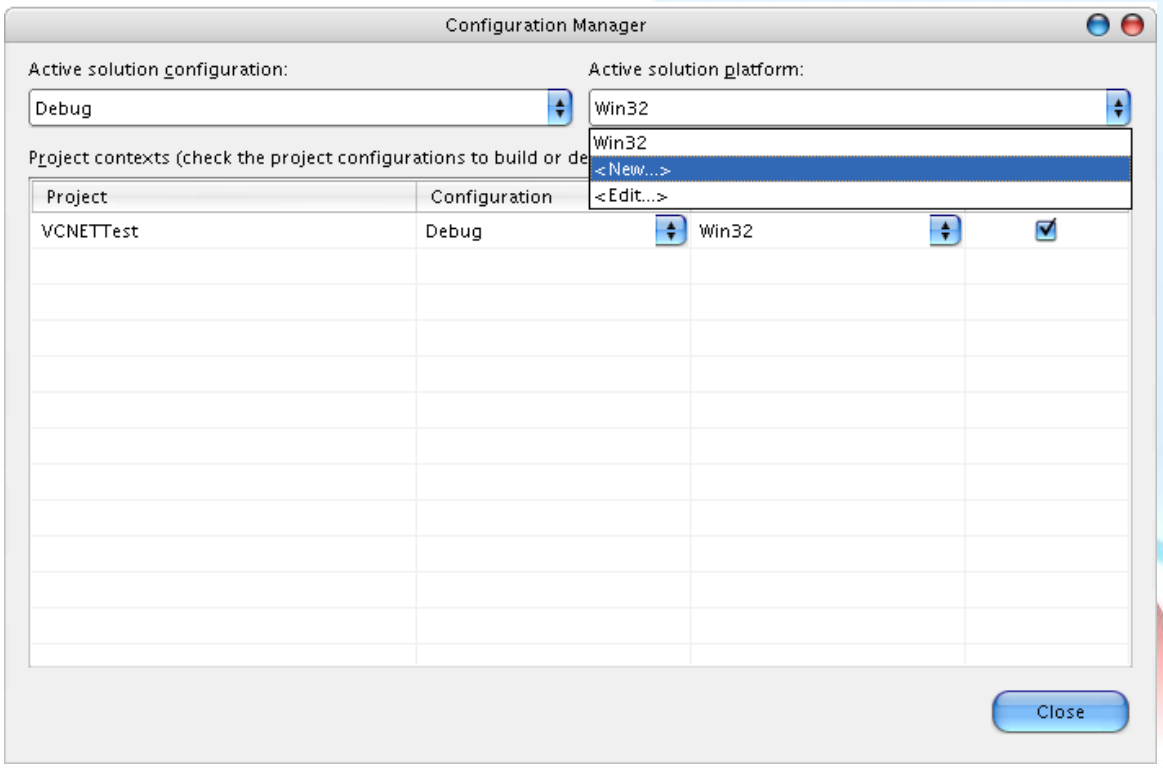
```
// VCNETTest.cpp : Defines the entry point for the console application.
//

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

WORD wRtn;
WORD wBoardNo;
WORD wTotalBoards;

int _tmain(int argc, _TCHAR* argv[])
{
	WORD wOutPortNo;

	//Initial the resource and get total board number form  Driver
	wRtn=Ixud_DriverInit(&wTotalBoards);
	if (wRtn!=Ixud_NoErr)
	{
		printf("\nDriver Init Error(%d)",wRtn);
		return wRtn;
	}
	printf("Write DO Value 0xFF");
	wBoardNo=0;
	wOutPortNo=0;
	//Write DO
	wRtn = Ixud_WriteDO(wBoardNo,wOutPortNo,0xFF);
	//Release the resource from driver
	wRtn = Ixud_DriverClose();
	return 0;
}
```

步驟 2:編譯應用程式

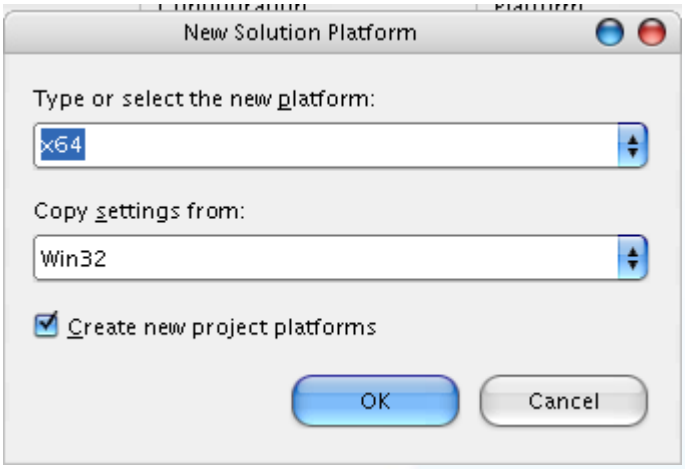
1. 在 Build 選單點擊 Configuration Manager。



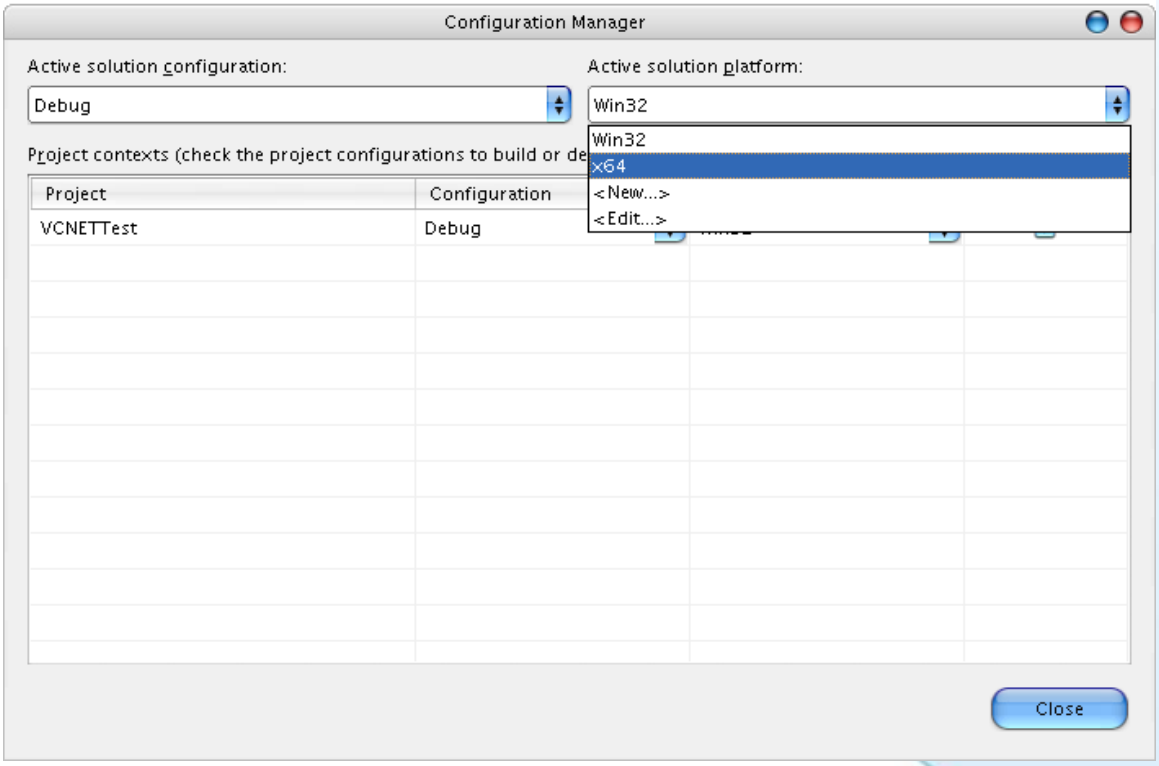

2. 在 Configuration Manager 點擊 Active solution Platform 下拉選單選擇<New…>。



3. 點擊 Type or select the new platform 下拉選單選擇 x64，並按下 OK，創建 x64 編譯平台。



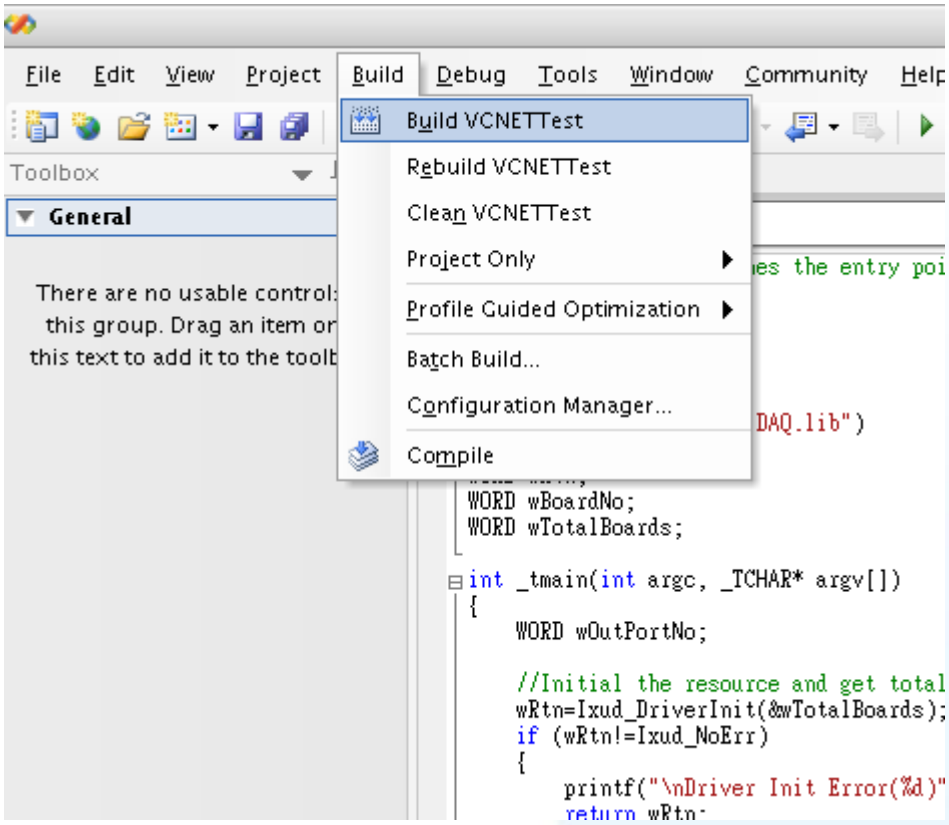

4. 選擇編譯平台，如果您想產生 64 位元的應用程式請在下拉選單選擇 x64，如果您想產生 32 位元程式請在下拉選單選擇 Win32 選取完畢後按下 Close。




若使用者編譯的是 64 位元應用程式應使用 64-bit UniDAQ.lib

若使用者編譯的是 32 位元應用程式應使用 32-bit UniDAQ.lib

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



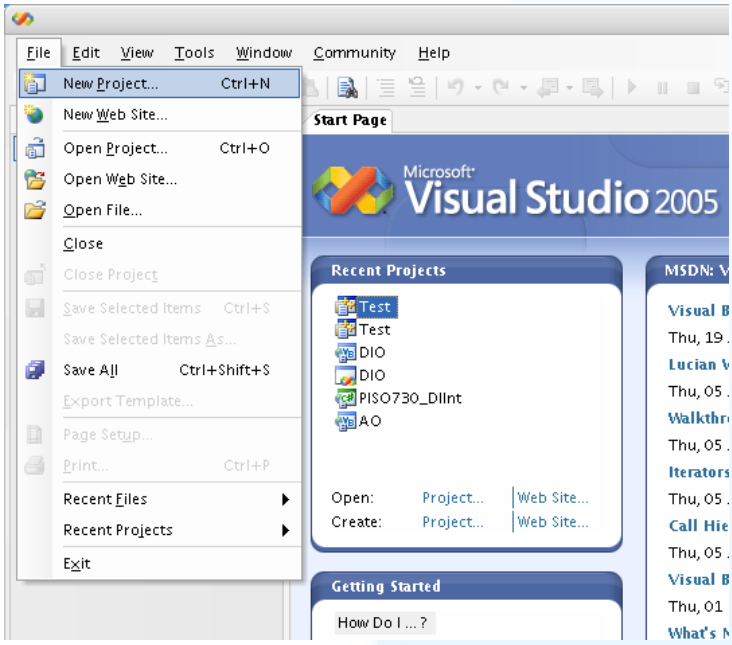

步驟 3:測試應用程式

立即在 DOS Box 下執行程式。

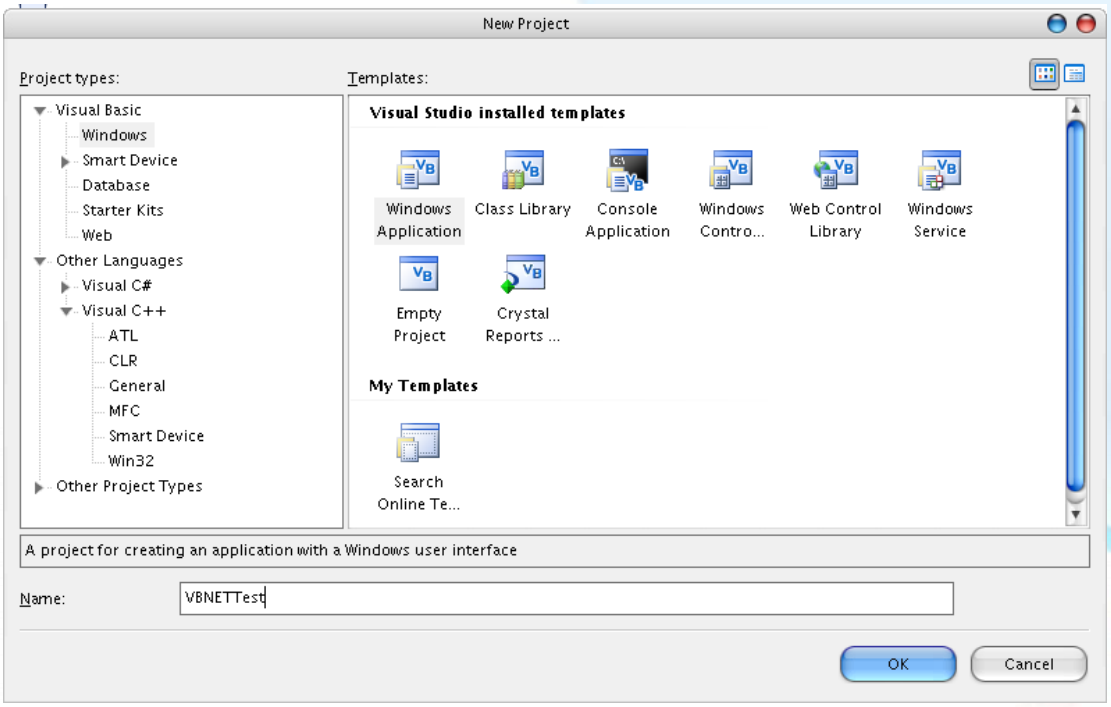
# 3.7. 在 Visual Basic.NET

步驟 1:撰寫應用程式

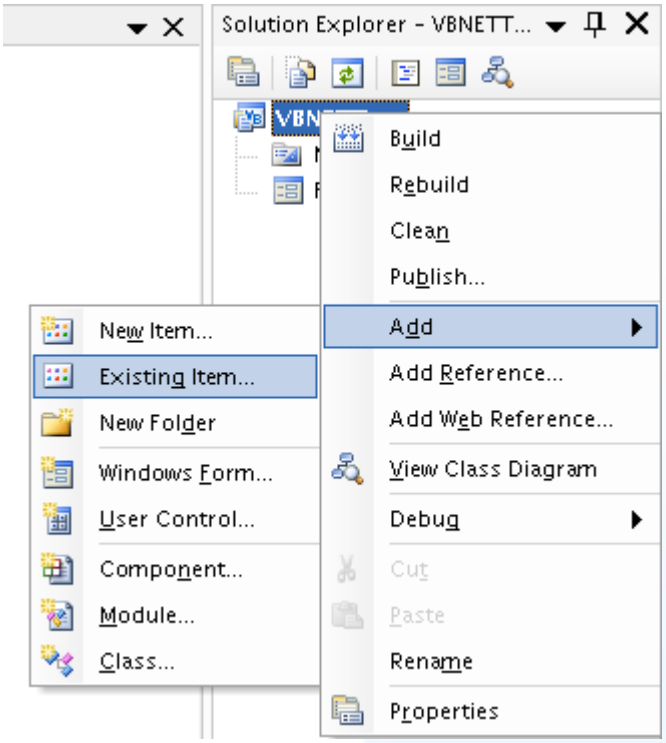
1. 至程式集開啓 Microsoft Visual Studio 2005
2. 從主要選單內選擇 File|New Project…



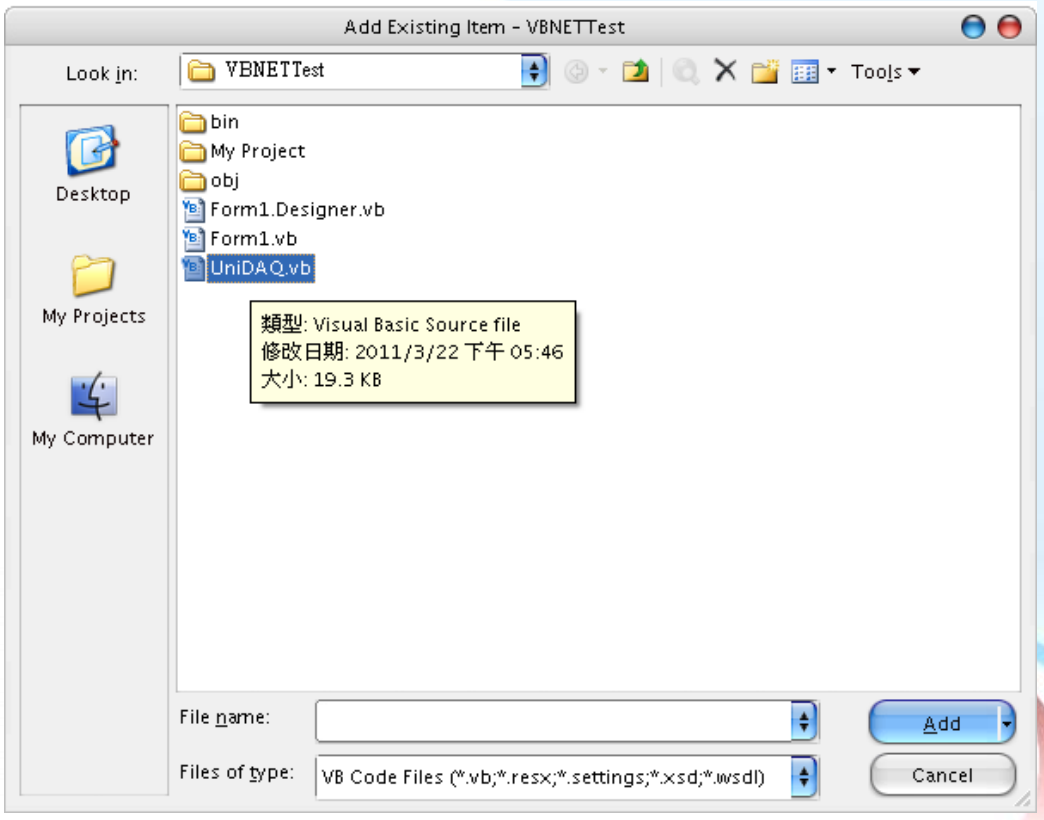

3. 在 Project type 下的列表點選項目 Visual Basic 並在展開選單內選擇 Windows，然後再右方 Templates 表框內選擇 Windows Application，接下來至下方的 Name 鍵入專案名稱 VBNETTest 然後按下 OK 按鍵。



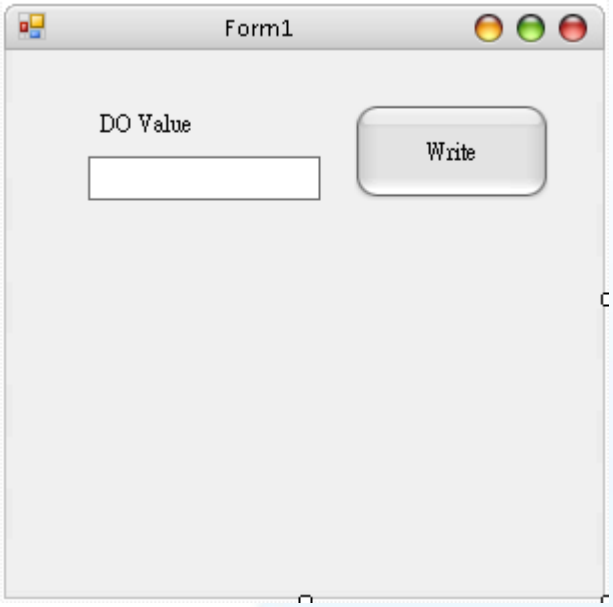
4. 在 Solution Explorer 開啓 Add Existion item 視窗。




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



6. 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 屬性爲 btnWrite 及在 Text 屬性上輸入 Write。

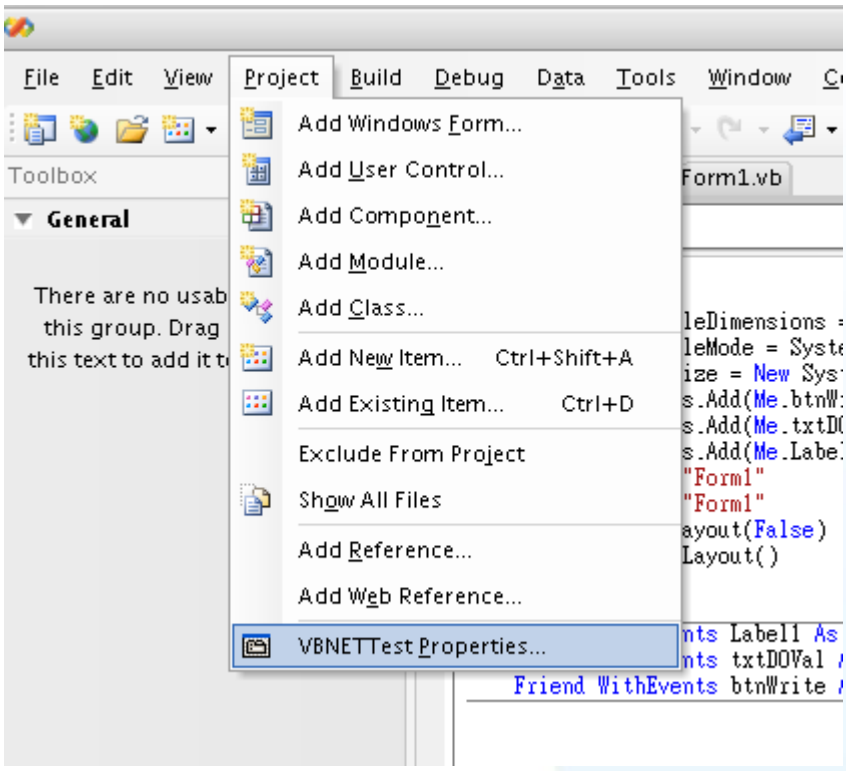



7. 在 btnWrite 填寫程式碼如下：

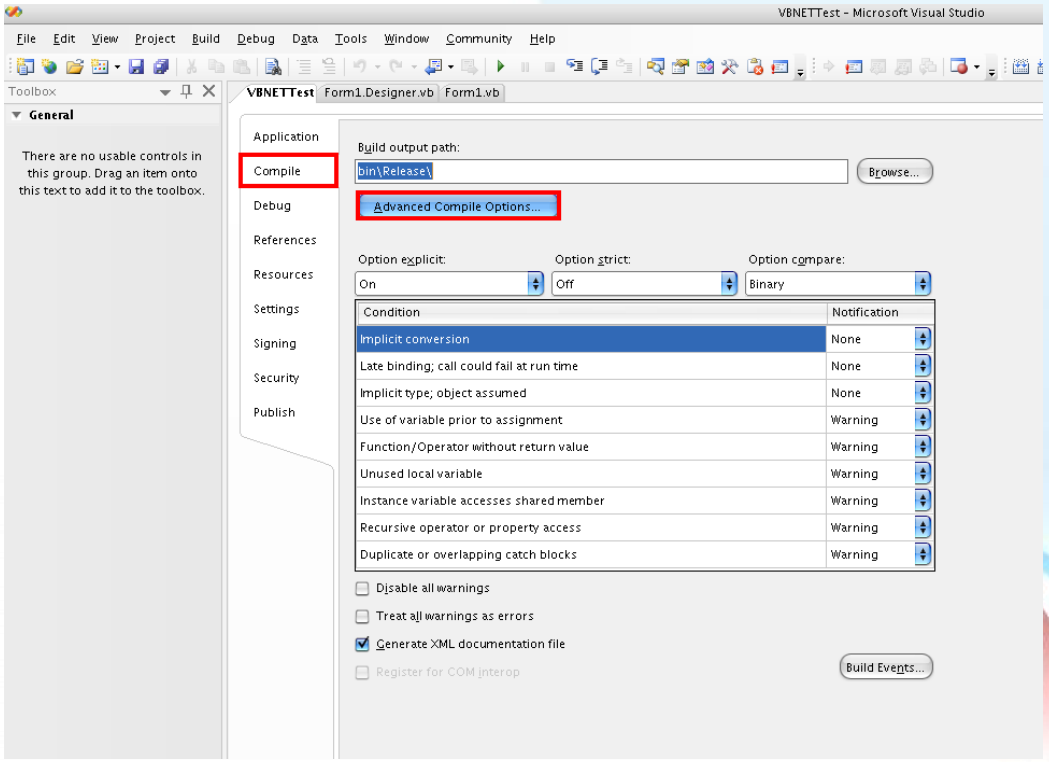
```
Private Sub btnWrite_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnWrite.Click
        Dim wTotalBoards As UInteger
        Dim wBoardNo As UInteger
        Dim wOutPortNo As UInteger
        Dim wRtn As UInteger

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

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

        wRtn = Ixud_DriverClose()
    End Sub
```

步驟 2:編譯應用程式

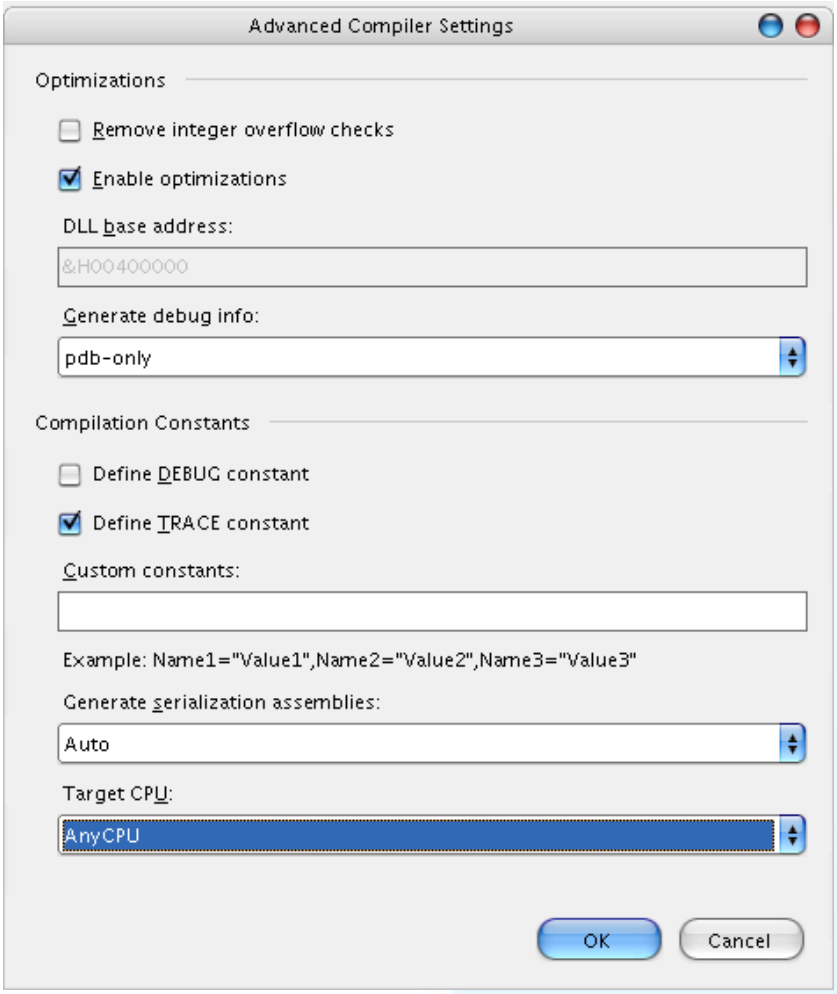
1. 在 Project 選單點擊 VBNETTest Properties。




2. 點擊 Compiler 後點擊 Advanced Compiler Option 按鍵，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。

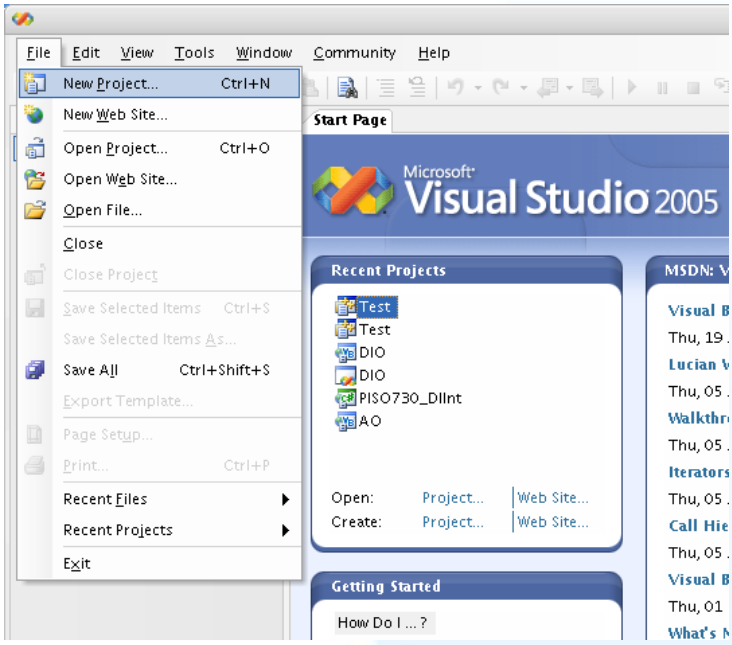
步驟 3:測試應用程式

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。

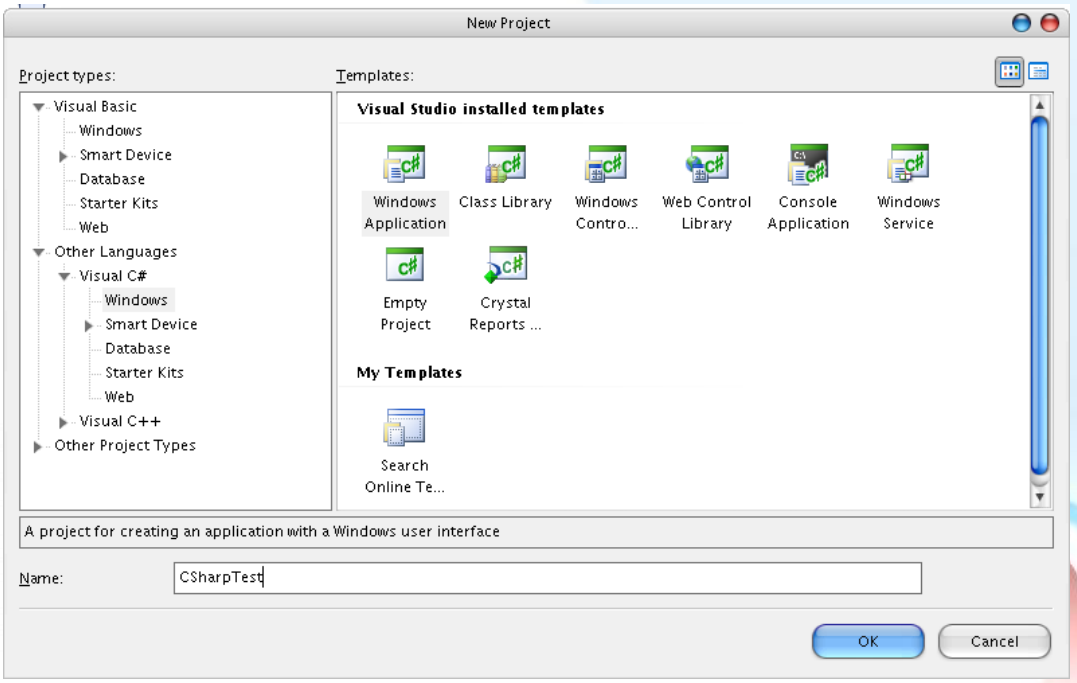
# 3.8. 在 Visual C#.NET

步驟 1:撰寫應用程式

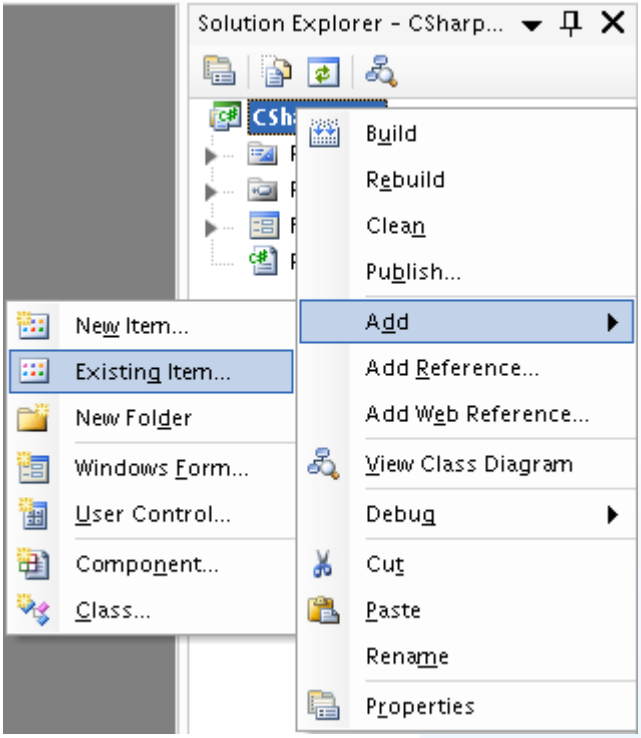
1. 至程式集開啓 Microsoft Visual Studio 2005
2. 從主要選單內選擇 File|New Project…



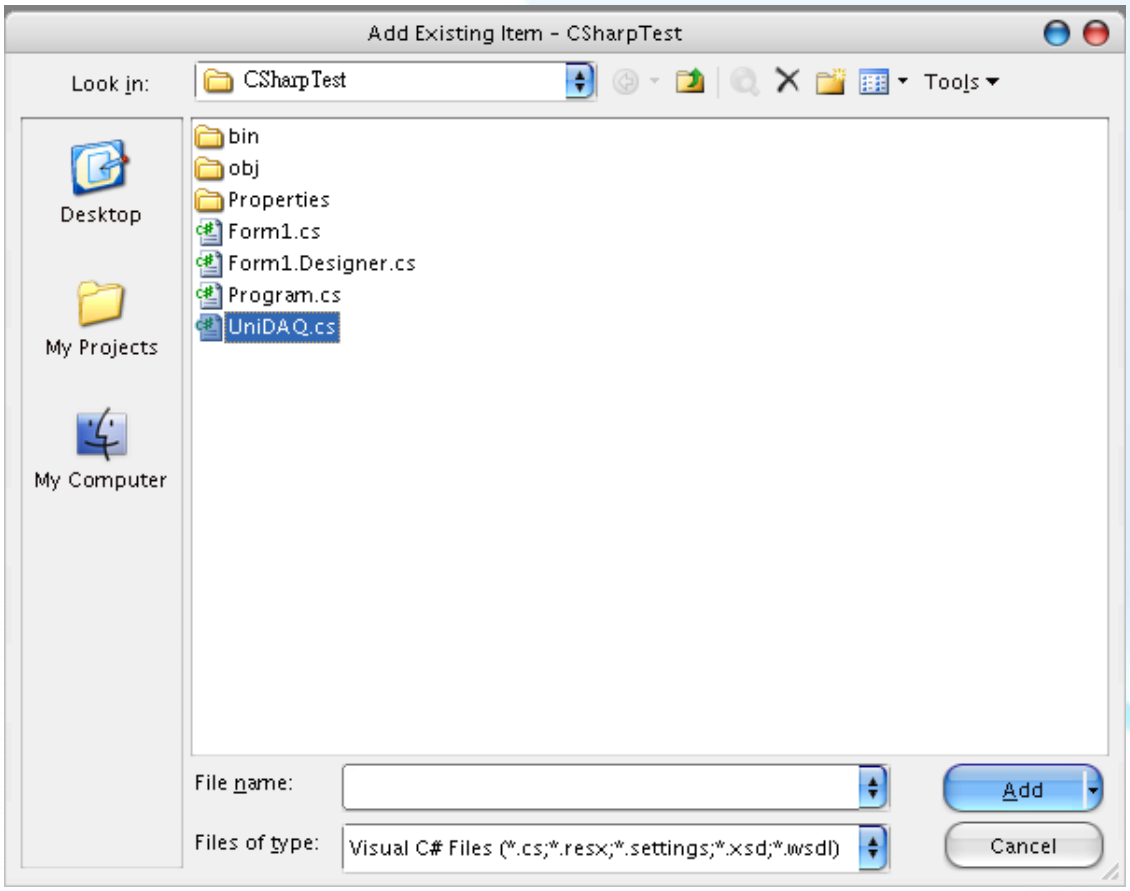

3. 在 Project type 下的列表點選項目 Visual C# 並在展開選單內選擇 Windows，然後再右方 Templates 表框內選擇 Windows Application，接下來至下方的 Name 鍵入專案名稱 CSharpTest 然後按下 OK 按鍵。



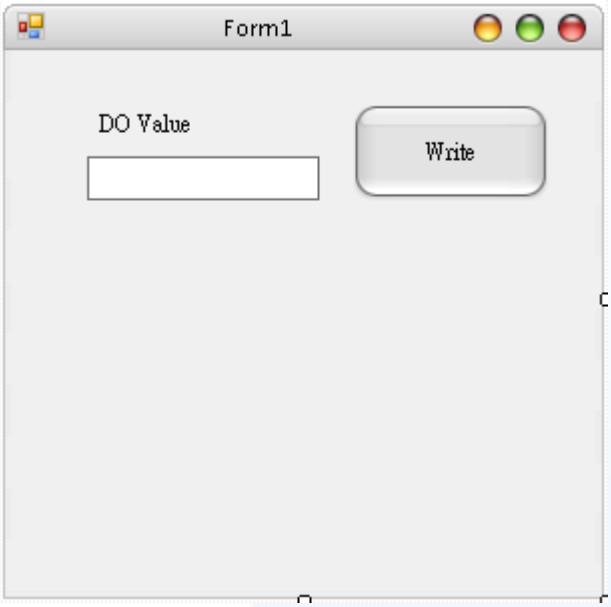
4. 在 Solution Explorer 開啓 Add Existion item 視窗。




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



6. 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 屬性爲 btnWrite 及在 Text 屬性上輸入 Write。

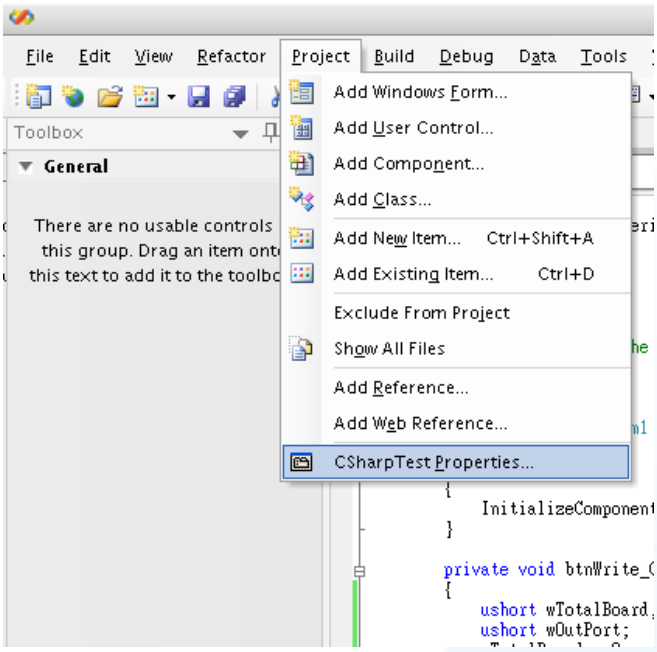


7. 在 Form.cs 填寫程式碼如下：

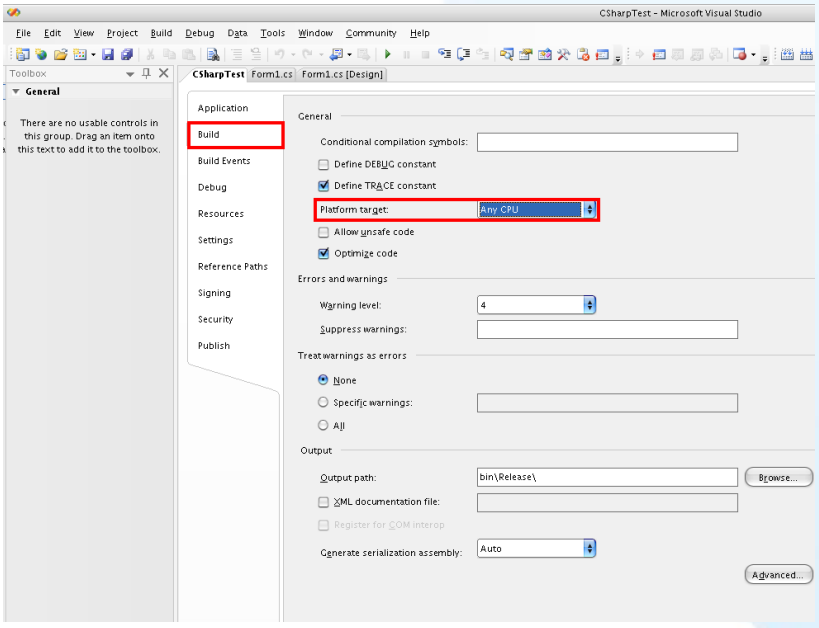
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using UniDAQ_Ns; //Include the UniDAQ namespace

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

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

            wBoardNo = 0;
            wOutPort = 0;
            //Write DO
            wRtn = UniDAQ.Ixud_WriteDO(wBoardNo, wOutPort, Convert.ToUInt32(txtDOVal.Text));
            //Release the resource from the driver
            wRtn = UniDAQ.Ixud_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 targer 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。

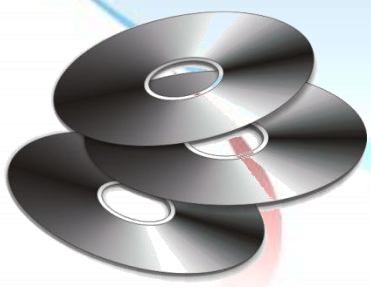
# 3.9. 範例程式及文件

放置 UniDAQ 相關資料的網址與位置：

| | |
|---|---|
|  | CD:\\ NAPDOS\PCI\UniDAQ\ |
|  | http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/ |
|  | ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/ |

UniDAQ 資料夾內檔案文件結構

- UniDAQ
  - DLL
    - 驅動程式、函式庫、各種語言的標頭檔案及範例程式
  - Manual
    - 泓格UniDAQ驅動函式庫的使用手冊
  - LabView
    - LabView範例程式(需安裝UniDAQ驅動程式函式庫)
  - Matlab
    - MATLAB範例程式(需安裝UniDAQ驅動程式函式庫)



# 4. 函式應用

提供泓格 UniDAQ 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

UniDAQ 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，Borland Delphi，Borland C Builder++，Microsoft Visual C++.NET，Microsoft Visual C#.NET，Microsoft Visual VB.NET。

**UniDAQ 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. 中斷事件函式集：支援具有中斷功能的裝置，當類比輸入及數位輸入操作完成時產生通知事件。
4. 類比輸出函式集：操作裝置透過 DAC 輸送出電壓、電流
5. 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…
6. 計時計數函式集：設定及讀取計時計數器
7. 記憶體輸出入函式集：存取記憶體內的數值。

支援程式語言：

- Microsoft Visual C++ 4.0 or higher
- Microsoft Visual Basic 4.0 or higher
- Borland Delphi 2.0 or higher
- Borland C++ Builder 1.0 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 | 中斷事件函式集 | 類比輸入函式集 |
|---|---|---|---|
| Ixud_GetDllVersion | Ixud_SetDIOModes32 | Ixud_SetEventCallback | Ixud_ConfigAI |
| Ixud_OptionMode | Ixud_SetDIOMode | Ixud_RemoveEventCallback | Ixud_ConfigAIEx |
| Ixud_DriverInit | Ixud_ReadDI | Ixud_InstallIrq | Ixud_ClearAIBuffer |
| Ixud_DriverClose | Ixud_WriteDO | Ixud_RemoveIrq | Ixud_GetBufferStatus |
| Ixud_SearchCard | Ixud_ReadDIBit | | Ixud_ReadAI |
| Ixud_GetBoardNoByCardID | Ixud_WriteDIBit | | Ixud_ReadAIH |
| Ixud_GetCardInfo | Ixud_ReadDI32 | | Ixud_PollingAI |
| Ixud_ReadPort | Ixud_WriteDO32 | | Ixud_PollingAIH |
| Ixud_WritePort | Ixud_SoftwareReadbackDO | | Ixud_PollingAIScan |
| Ixud_ReadPort32 | Ixud_StartDI | | Ixud_PollingAIScanH |
| Ixud_WritePort32 | Ixud_StopDI | | Ixud_StartAI |
| Ixud_ReadPhyMemory | Ixud_GetDIBufferH | | Ixud_StartAIScan |
| Ixud_WritePhyMemory | Ixud_StartDO | | Ixud_StartExtAI |
| | Ixud_StopDO | | Ixud_StartExtAnalogTrigger |
| | | | Ixud_StartExtAIScan |
| | | | Ixud_GetAIBuffer |
| | | | Ixud_GetAIBufferH |
| | | | Ixud_StopAI |

| 類比輸出函式集 | 計時計數函式集 | 記憶體輸出入函式集 |
|---|---|---|
| Ixud_ConfigAO | Ixud_ReadCounter | Ixud_ReadMemory |
| Ixud_WriteAOVoltage | Ixud_SetCounter | Ixud_WriteMemory |
| Ixud_WriteAOVoltageH | Ixud_DisableCounter | Ixud_ReadMemory32 |
| Ixud_WriteAOCurrent | Ixud_SetFCChannelMode | Ixud_WriteMemory32 |
| Ixud_WriteAOCurrentH | Ixud_ReadFrequency | |
| Ixud_StartAOVoltage | | |
| Ixud_StartAOVoltageH | | |
| Ixud_StopAO | | |

# 4.2. 驅動函式庫

當使用者使用泓格 **UniDAQ** 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啓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。


在設備項目”0 PCI-826”，Board Num 即爲 0，您可以透過 Board Num 直接指定來操作裝置。

### Ixud_DriverInit 及 Ixud_DriverClose 函式

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

### Ixud_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 Ixud_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爲埠，每一個埠依板卡的設計會有 8、16 或 32 線。

某一些資料擷取板卡(例如：PIO-D24U/D56U/D48U/D96U/144U/168U)的數位輸出輸入埠，可以被設定爲輸入或輸出。您可以使用 Ixud_SetDIOModes32 或 Ixud_SetDIOMode 函式來配置指定的埠爲輸出或輸入。

### 4.3.1. 數位輸入

泓格 UniDAQ 驅動函式庫的數位輸出入函式集可執行數位輸入等功能，它支援軟體觸發數位輸入及數位輸入中斷。

- **軟體觸發**

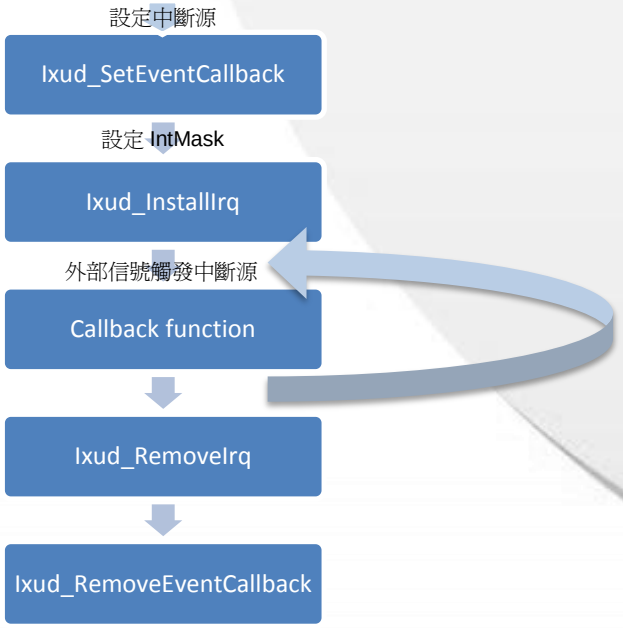
使用者可以呼叫 Ixud_ReadDI 函式讀取指定數位輸入埠的資料。

**呼叫流程**


- **中斷觸發**

使用者可透過中斷觸發功能監控數位輸入通道的狀態。當輸入狀態產生變化，輸入準位由低變高或由高變低時，它將會透過硬體中斷通知驅動程式，所以使用者就不需定時的去輪詢數位輸入通道。

**呼叫流程**



### 4.3.2. 數位輸出

泓格 UniDAQ 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 Ixud_WriteDO 函式簡易的設定數位輸出埠的資料，透過 Ixud_SoftwareReadbackDO 可取得目前數位輸出埠的狀態。

**呼叫流程**

# 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。它可以擷取單筆資料、多通道資料或波型資料。類比輸入根據不同的觸發模式及資料傳輸方式提供各種操作方式。

- **軟體觸發**

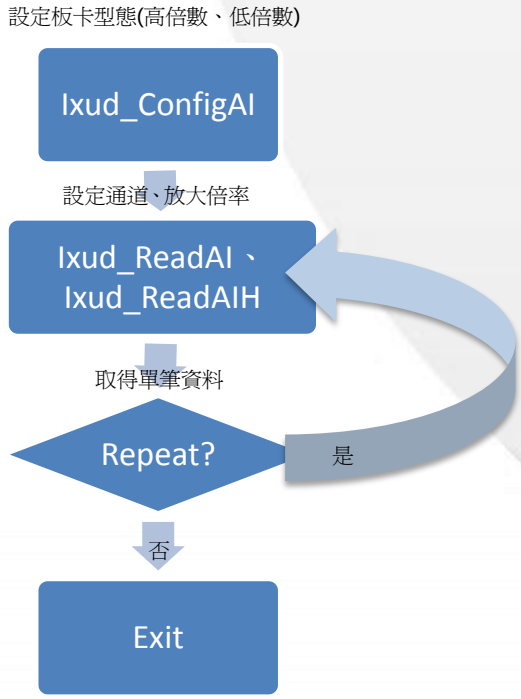
  透過軟體觸發資料轉換來取得類比資料，函式庫提供三種應用方案。第一種是單通道單筆資料讀取，第二種是單通道多筆資料讀取，最後一種是多通道掃描。

  > 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)

  - **單通道單筆資料取樣**

    如果使用者需要定時取樣多筆類比資料的功能，使用者可以創建一個軟體定時器(事件)，定時地呼叫函式 Ixud_ReadAI 或 Ixud_ReadAIH。

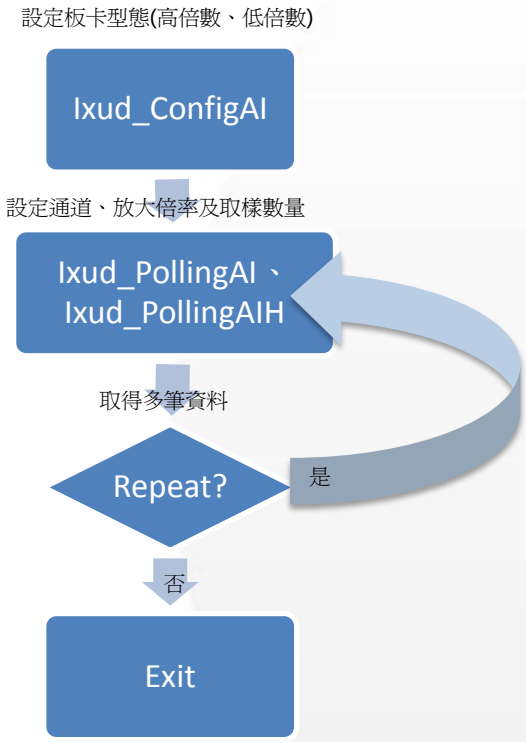
**呼叫流程**



## ■ 單通道多筆資料取樣

單通道採樣的功能與單通道單筆資料取樣相似，可以在單一的通道上連續採集一個以上的類比資料。

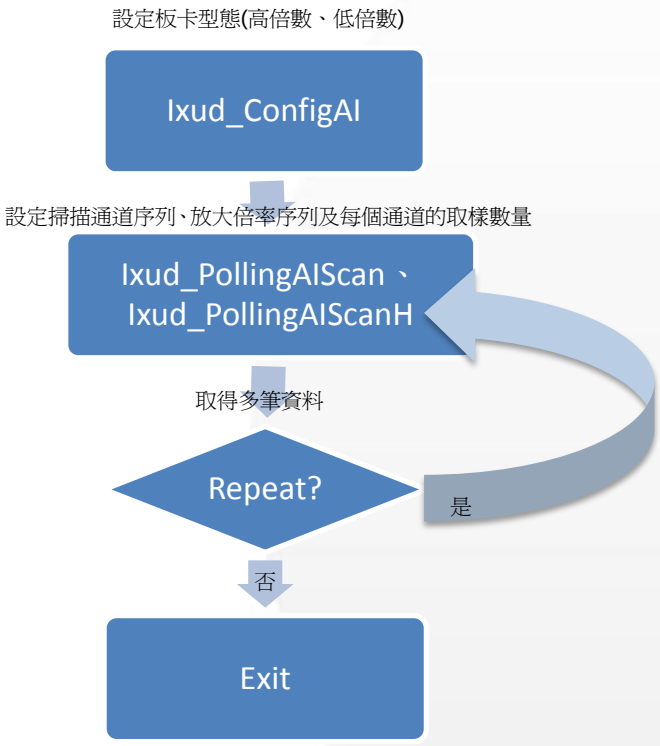
### 呼叫流程



## ■ 多通道採樣

多通道採樣的功能與單通道單筆資料取樣相似，除了可以採樣多個通道之外，還可以採集一個以上的類比資料。

### 呼叫流程



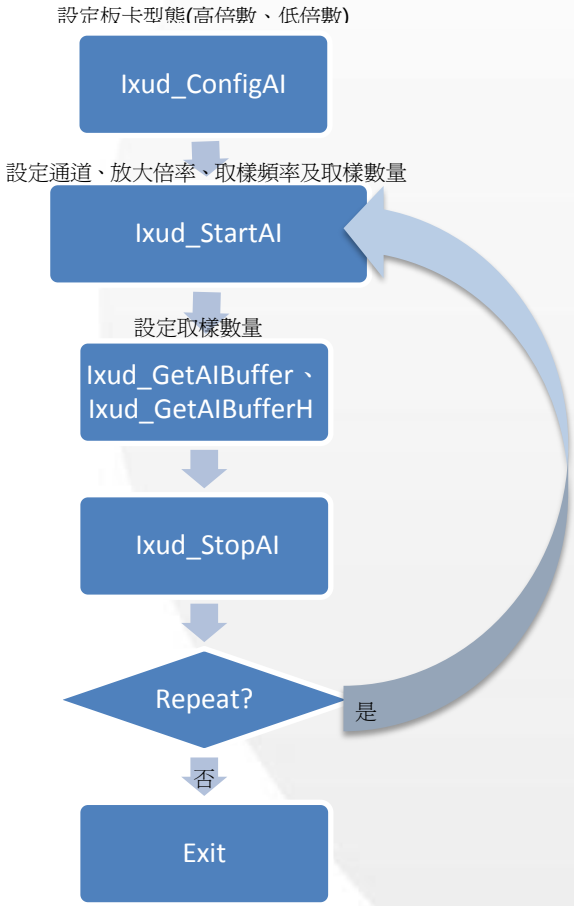
- **讀取波型資料**

類比輸入功能提供各種方式的應用讓使用者取得最精確的波型資料，一般採集波型的觸發模式有內部定時觸發、中斷觸發和外部觸發。

- **單通道內部定時觸發(single-channel Internal Pacer trigger)**

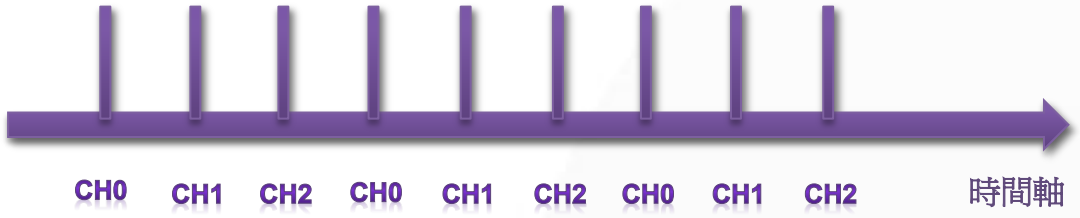
使用者設定取樣頻率後透過板卡上內建的定時器以固定的頻率觸發 ADC 去採集單個類比通道的波型資料。

**呼叫流程**

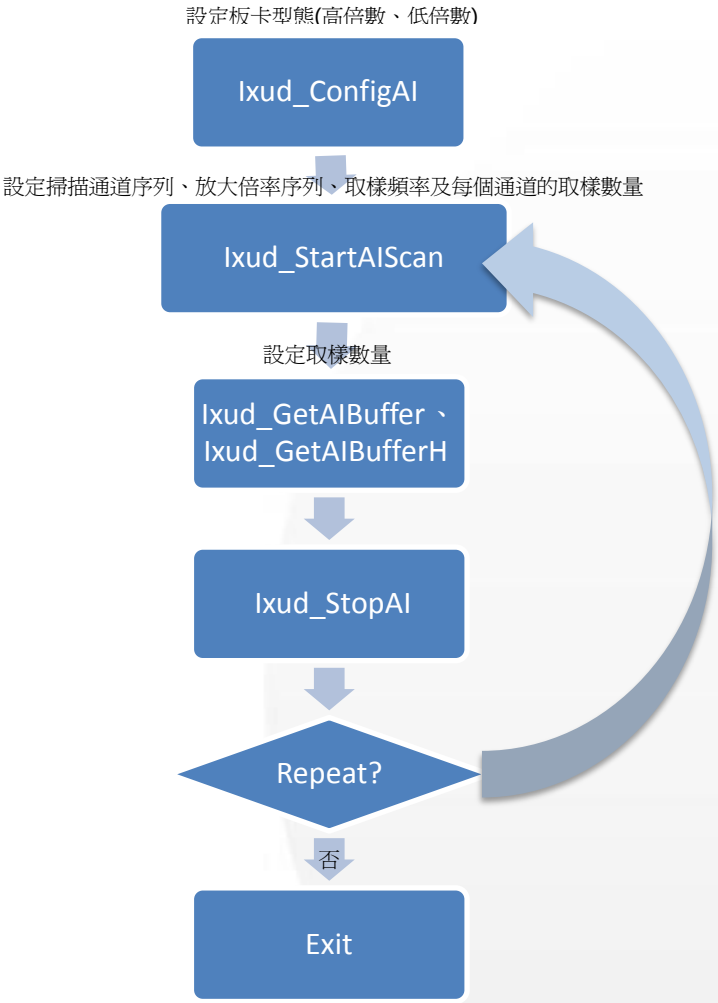


- **多通道內部定時觸發(multi-channel Internal Pacer trigger)**

  使用者設定取樣頻率後透過板卡上內建的定時器以固定的頻率觸發 ADC 去採集多個類比通道的波型資料。



## 呼叫流程



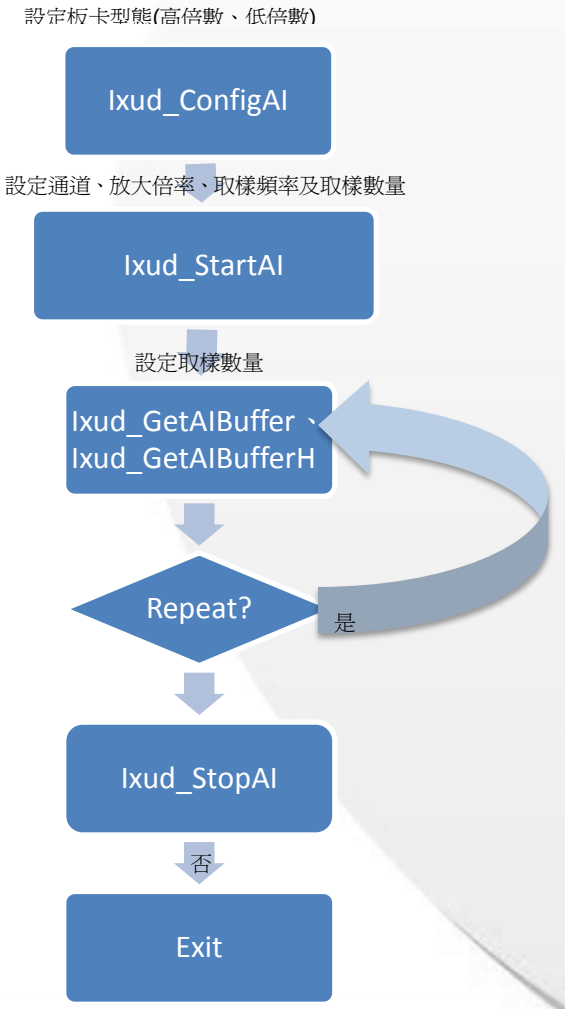
- **類比訊號長時間監控**

  採集類比資料前將會利用系統記憶體配置一塊緩衝區(預設值 2MB)，用來暫存所採集的類比資料，來達到連續採集的功能，使用者可以再採集的過程中從系統緩衝區內提取類比數值。

  ■ 單通道連續採集(single channel continuous capture)

  使用單一通道來連續採集類比資料並儲存至系統緩衝區等待使用者提取。

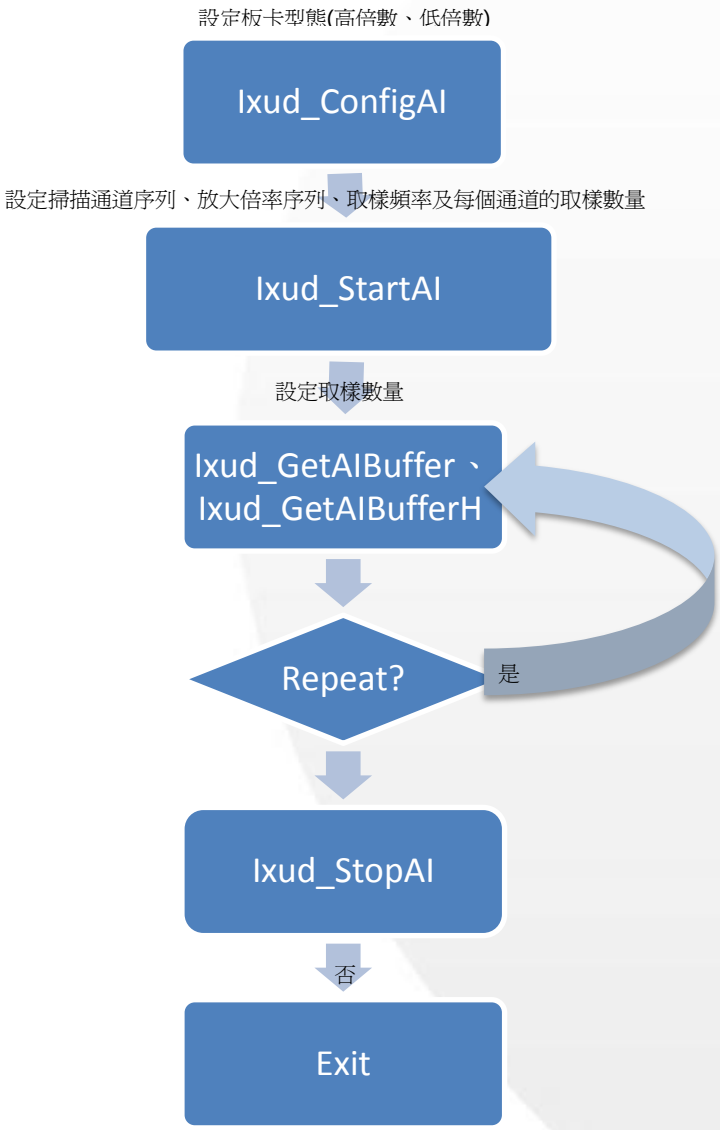
  **呼叫流程**



■ 多通道連續採集(multi-channel continuous capture)

使用多通道來連續採集類比資料並儲存至系統緩衝區等待使用者提取。

呼叫流程



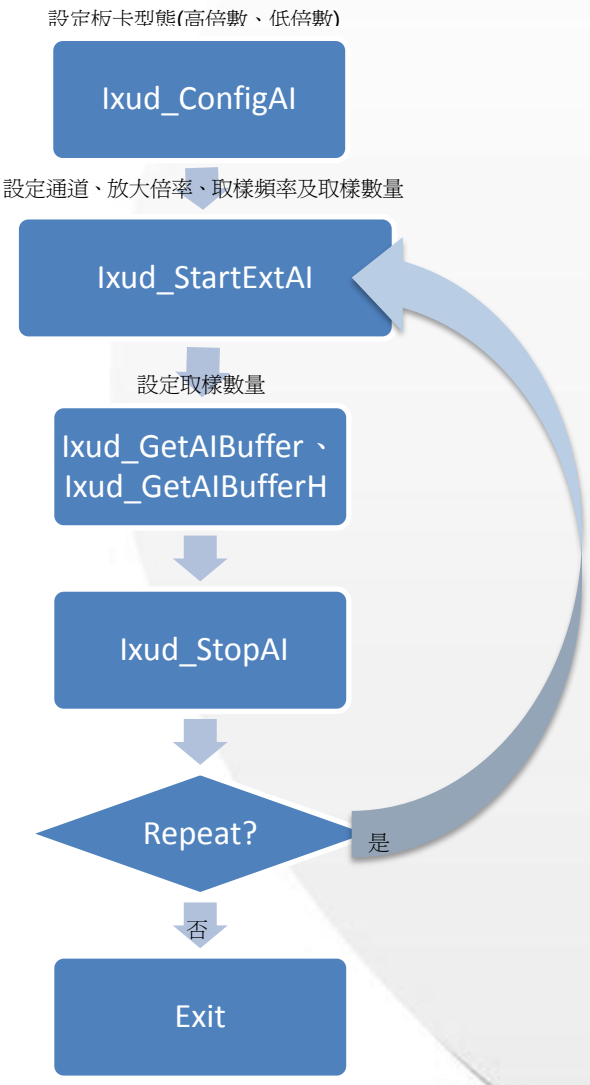
- 外部觸發

  透過外部信號來決定採樣的起始時間，函式集提供三種方式外部觸發的功能分別是 post-trigger、pre-trigger 及 mid-trigger。

  - 單通道外部觸發

    透過外部觸發信號，來決定單一通道資料採集的時機。

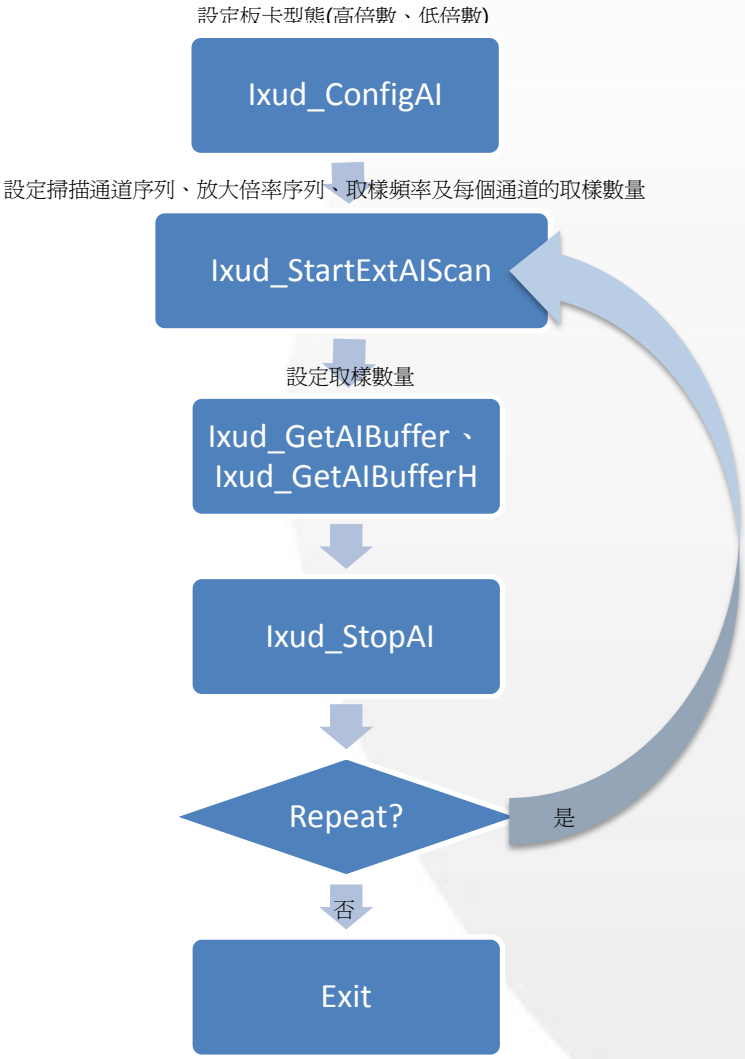
    呼叫流程



■ 多通道外部觸發波型採集

透過外部觸發信號，來決定多通道資料採集的時機。

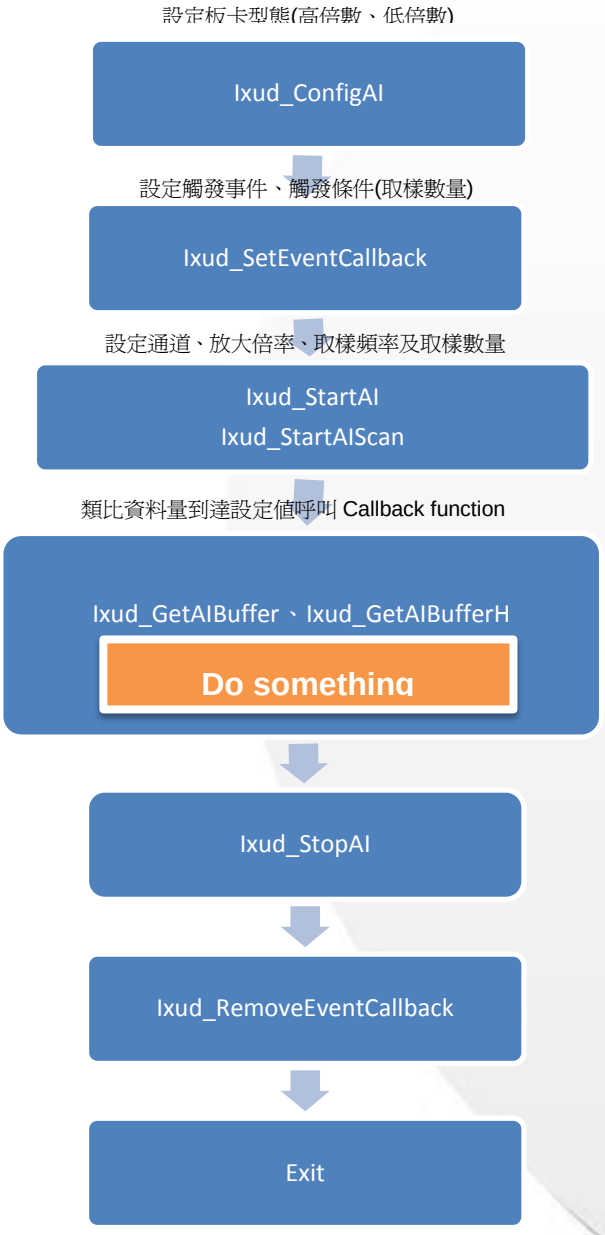
呼叫流程



- 類比資料事件觸發

  當類比資料每到達一定的數量時，驅動函式庫會產生出一個事件通知使用者。

呼叫流程



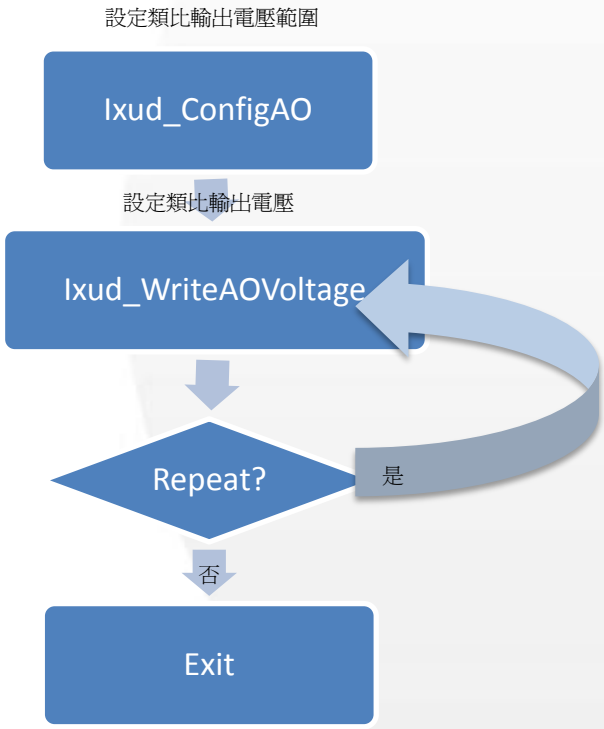
# 4.5. 類比輸出

泓格 UniDAQ 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。

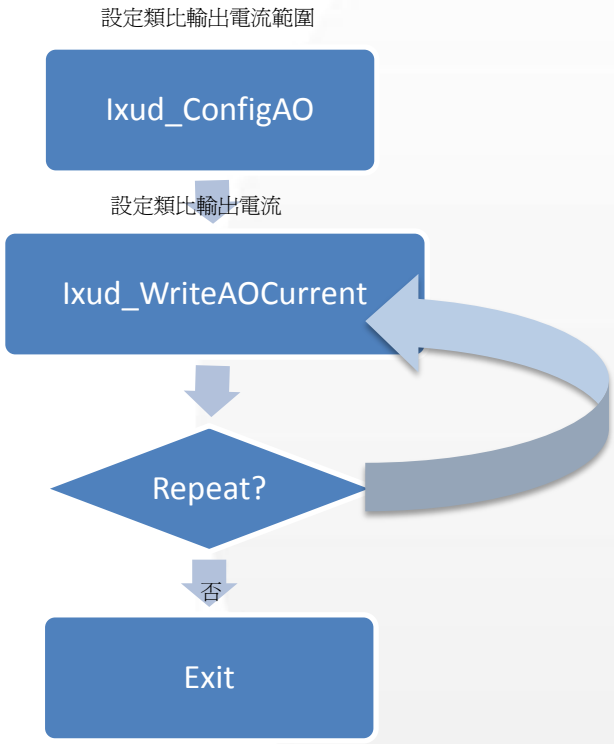
  呼叫流程



- 靜態電流輸出

  設定類比輸出通道輸出一固定的電流。

呼叫流程



# 4.6. 計時計數器

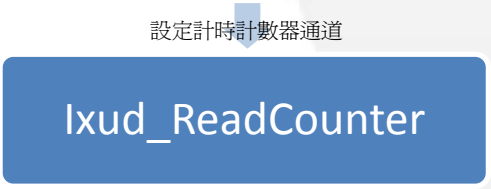
泓格 UniDAQ 驅動函式庫的計時計數函式集可執行 8254 計時計數器等功能。

- 寫入設定計時計數器

**呼叫流程**


- 讀取計時計數器

**呼叫流程**



# 4.7. 記憶體存取

泓格 UniDAQ 驅動函式庫的記憶體輸出入函式集可執行對記憶體位址存取等功能。

- 寫入記憶體位址

**呼叫流程**


- 讀取記憶體位址

**呼叫流程**

# 5. 函式參考

提供泓格 UniDAQ 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. 函式支援列表

| 函式名稱 | Ixud_DriverInit | Ixud_DriverClose | Ixud_SearchCard | Ixud_GetCardInfo | Ixud_GetBoardNoByCardID |
|---|---|---|---|---|---|
| PIO-D24/D24U/D56/D56U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D48/D48U/D48SU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D64/D64U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D96/D96U/D96SU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D144/D144U/D144LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D168/D168U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-D24/D56 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-D48 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-DA4/DA4U, PEX-DA4 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-DA8/DA8U, PEX-DA8 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-DA16/DA16U,PEX-DA16 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-821 Series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P64/P64U, PEX-P64 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-A64/C64/C64U, PEX-C64 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P32C32/P32C32U/P32S32WU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-P32C32 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P32A32/P32A32U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-725/730/730A | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P16R16U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-P16R16i/P8R8i | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P8R8/P8R8U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P8R8UDC/AC | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-DA4U/DA8U/DA16U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-DA2/DA2U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-813/813U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-TMC12A | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-M512/M512U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-P16R16/P16POR16 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-P16C16/P8R8 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-P16POR16i/P8POR8i | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1002 series, PEX-1002 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1202 series, PEX-1202 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1602 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1800/1802 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-822LU/826LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-FC16U | ✔ | ✔ | ✔ | ✔ | ✔ |

| 函式名稱 | Ixud_ReadPort | Ixud_WritePort | Ixud_ReadPort32 | Ixud_WritePort32 | Ixud_SetDIOModes32 |
|---|---|---|---|---|---|
| PIO-D24/D24U/D56/D56U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D48/D48U/D48SU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D64/D64U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D96/D96U/D96SU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D144/D144U/D144LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D168/D168U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-D24/D56 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-D48 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-DA4/DA4U, PEX-DA4 | ✔ | ✔ | ✔ | ✔ | |
| PIO-DA8/DA8U, PEX-DA8 | ✔ | ✔ | ✔ | ✔ | |
| PIO-DA16/DA16U,PEX-DA16 | ✔ | ✔ | ✔ | ✔ | |
| PIO-821 Series | ✔ | ✔ | ✔ | ✔ | |
| PISO-P64/P64U, PEX-P64 | ✔ | ✔ | ✔ | ✔ | |
| PISO-A64/C64/C64U, PEX-C64 | ✔ | ✔ | ✔ | ✔ | |
| PISO-P32C32/P32C32U/P32S32WU | ✔ | ✔ | ✔ | ✔ | |
| PEX-P32C32 | ✔ | ✔ | ✔ | ✔ | |
| PISO-P32A32/P32A32U | ✔ | ✔ | ✔ | ✔ | |
| PISO-725/730/730A | ✔ | ✔ | ✔ | ✔ | |
| PISO-P16R16U | ✔ | ✔ | ✔ | ✔ | |
| PEX-P16R16i/P8R8i | ✔ | ✔ | ✔ | ✔ | |
| PISO-P8R8/P8R8U | ✔ | ✔ | ✔ | ✔ | |
| PISO-P8R8UDC/AC | ✔ | ✔ | ✔ | ✔ | |
| PISO-DA4U/DA8U/DA16U | ✔ | ✔ | ✔ | ✔ | |
| PISO-DA2/DA2U | ✔ | ✔ | ✔ | ✔ | |
| PISO-813/813U | ✔ | ✔ | ✔ | ✔ | |
| PCI-TMC12A | ✔ | ✔ | ✔ | ✔ | |
| PCI-M512/M512U | ✔ | ✔ | ✔ | ✔ | |
| PCI-P16R16/P16POR16 | ✔ | ✔ | ✔ | ✔ | |
| PCI-P16C16/P8R8 | ✔ | ✔ | ✔ | ✔ | |
| PEX-P16POR16i/P8POR8i | ✔ | ✔ | ✔ | ✔ | |
| PCI-1002 series, PEX-1002 series | ✔ | ✔ | ✔ | ✔ | |
| PCI-1202 series, PEX-1202 series | ✔ | ✔ | ✔ | ✔ | |
| PCI-1602 series | ✔ | ✔ | ✔ | ✔ | |
| PCI-1800/1802 series | ✔ | ✔ | ✔ | ✔ | |
| PCI-822LU/826LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-FC16U | ✔ | ✔ | ✔ | ✔ | ✔ |

| 函式名稱 | Ixud_SetDIOMode | Ixud_ReadDI | Ixud_WriteDO | Ixud_ReadDIBit<br>Ixud_ReadDI32 | Ixud_WriteDOBit<br>Ixud_WriteDO32 |
|---|---|---|---|---|---|
| PIO-D24/D24U/D56/D56U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D48/D48U/D48SU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D64/D64U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D96/D96U/D96SU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D144/D144U/D144LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-D168/D168U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-D24/D56 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PEX-D48 | ✔ | ✔ | ✔ | ✔ | ✔ |
| PIO-DA4/DA4U, PEX-DA4 | | ✔ | ✔ | ✔ | ✔ |
| PIO-DA8/DA8U, PEX-DA8 | | ✔ | ✔ | ✔ | ✔ |
| PIO-DA16/DA16U,PEX-DA16 | | ✔ | ✔ | ✔ | ✔ |
| PIO-821 Series | | ✔ | ✔ | ✔ | ✔ |
| PISO-P64/P64U, PEX-P64 | | ✔ | | ✔ | |
| PISO-A64/C64/C64U, PEX-C64 | | | ✔ | | ✔ |
| PISO-P32C32/P32C32U/P32S32WU | | ✔ | ✔ | ✔ | ✔ |
| PEX-P32C32 | | ✔ | ✔ | ✔ | ✔ |
| PISO-P32A32/P32A32U | | ✔ | ✔ | ✔ | ✔ |
| PISO-725/730/730A | | ✔ | ✔ | ✔ | ✔ |
| PISO-P16R16U | | ✔ | ✔ | ✔ | ✔ |
| PEX-P16R16i/P8R8i | | ✔ | ✔ | ✔ | ✔ |
| PISO-P8R8/P8R8U | | ✔ | ✔ | ✔ | ✔ |
| PISO-P8R8UDC/AC | | ✔ | ✔ | ✔ | ✔ |
| PISO-DA4U/DA8U/DA16U | | ✔ | ✔ | ✔ | ✔ |
| PISO-DA2/DA2U | | | | | |
| PISO-813/813U | | | | | |
| PCI-TMC12A | | ✔ | ✔ | ✔ | ✔ |
| PCI-M512/M512U | | ✔ | ✔ | ✔ | ✔ |
| PCI-P16R16/P16POR16 | | ✔ | ✔ | ✔ | ✔ |
| PCI-P16C16/P8R8 | | ✔ | ✔ | ✔ | ✔ |
| PEX-P16POR16i/P8POR8i | | ✔ | ✔ | ✔ | ✔ |
| PCI-1002 series, PEX-1002 series | | ✔ | ✔ | ✔ | ✔ |
| PCI-1202 series, PEX-1202 series | | ✔ | ✔ | ✔ | ✔ |
| PCI-1602 series | | ✔ | ✔ | ✔ | ✔ |
| PCI-1800/1802 series | | ✔ | ✔ | ✔ | ✔ |
| PCI-822LU/826LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-FC16U | ✔ | ✔ | ✔ | ✔ | ✔ |

| 函式名稱 | lxud_SoftwareReadbackDO | lxud_SetEventCallback<br>lxud_RemoveEventCallback | lxud_InstallIrq<br>lxud_RemoveIrq |
|---|---|---|---|
| PIO-D24/D24U/D56/D56U | ✔ | ✔ | ✔ |
| PIO-D48/D48U/D48SU | ✔ | ✔ | ✔ |
| PIO-D64/D64U | ✔ | ✔ | ✔ |
| PIO-D96/D96U/D96SU | ✔ | ✔ | ✔ |
| PIO-D144/D144U/D144LU | ✔ | ✔ | ✔ |
| PIO-D168/D168U | ✔ | ✔ | ✔ |
| PEX-D24/D56 | ✔ | ✔ | ✔ |
| PEX-D48 | ✔ | ✔ | ✔ |
| PIO-DA4/DA4U, PEX-DA4 | ✔ | ✔ | ✔ |
| PIO-DA8/DA8U, PEX-DA8 | ✔ | ✔ | ✔ |
| PIO-DA16/DA16U,PEX-DA16 | ✔ | ✔ | ✔ |
| PIO-821 Series | ✔ | ✔ | |
| PISO-P64/P64U, PEX-P64 | ✔ | | |
| PISO-A64/C64/C64U, PEX-C64 | ✔ | | |
| PISO-P32C32/P32C32U/P32S32WU | ✔ | | |
| PEX-P32C32 | ✔ | | |
| PISO-P32A32/P32A32U | ✔ | | |
| PISO-725/730/730A | ✔ | ✔ | ✔ |
| PISO-P16R16U | ✔ | | |
| PEX-P16R16i/P8R8i | ✔ | | |
| PISO-P8R8/P8R8U | ✔ | | |
| PISO-P8R8UDC/AC | ✔ | | |
| PISO-DA4U/DA8U/DA16U | ✔ | ✔ | ✔ |
| PISO-DA2/DA2U | | ✔ | ✔ |
| PISO-813/813U | | | |
| PCI-TMC12A | ✔ | ✔ | ✔ |
| PCI-M512/M512U | ✔ | | |
| PCI-P16R16/P16POR16 | ✔ | | |
| PCI-P16C16/P8R8 | ✔ | | |
| PEX-P16POR16i/P8POR8i | ✔ | | |
| PCI-1002 series, PEX-1002 series | ✔ | ✔ | |
| PCI-1202 series, PEX-1202 series | ✔ | | |
| PCI-1602 series | ✔ | | |
| PCI-1800/1802 series | ✔ | | |
| PCI-822LU/826LU | ✔ | ✔ | |
| PCI-2602U | ✔ | | |
| PCI-FC16U | ✔ | | |

| 函式名稱 | Ixud_ConfigAI<br>Ixud_ConfigAIEx | Ixud_ClearAIBuffer | Ixud_GetBufferStatus | Ixud_ReadAI | Ixud_ReadAIH |
|---|---|---|---|---|---|
| PIO-D24/D24U/D56/D56U | | | | | |
| PIO-D48/D48U/D48SU | | | | | |
| PIO-D64/D64U | | | | | |
| PIO-D96/D96U/D96SU | | | | | |
| PIO-D144/D144U/D144LU | | | | | |
| PIO-D168/D168U | | | | | |
| PEX-D24/D56 | | | | | |
| PEX-D48 | | | | | |
| PIO-DA4/DA4U, PEX-DA4 | | | | | |
| PIO-DA8/DA8U, PEX-DA8 | | | | | |
| PIO-DA16/DA16U,PEX-DA16 | | | | | |
| PIO-821 Series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PISO-P64/P64U, PEX-P64 | | | | | |
| PISO-A64/C64/C64U, PEX-C64 | | | | | |
| PISO-P32C32/P32C32U/P32S32WU | | | | | |
| PEX-P32C32 | | | | | |
| PISO-P32A32/P32A32U | | | | | |
| PISO-725/730/730A | | | | | |
| PISO-P16R16U | | | | | |
| PEX-P16R16i/P8R8i | | | | | |
| PISO-P8R8/P8R8U | | | | | |
| PISO-P8R8UDC/AC | | | | | |
| PISO-DA4U/DA8U/DA16U | | | | | |
| PISO-DA2/DA2U | | | | | |
| PISO-813/813U | ✔ | | | ✔ | ✔ |
| PCI-TMC12A | | | | | |
| PCI-M512/M512U | | | | | |
| PCI-P16R16/P16POR16 | | | | | |
| PCI-P16C16/P8R8 | | | | | |
| PEX-P16POR16i/P8POR8i | | | | | |
| PCI-1002 series, PEX-1002 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1202 series, PEX-1202 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1602 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1800/1802 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-822LU/826LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-FC16U | | | | | |

| 函式名稱 | Ixud_PollingAI | Ixud_PollingAIH | Ixud_PollingAIScan | Ixud_PollingAIScanH |
|---|---|---|---|---|
| PIO-D24/D24U/D56/D56U | | | | |
| PIO-D48/D48U/D48SU | | | | |
| PIO-D64/D64U | | | | |
| PIO-D96/D96U/D96SU | | | | |
| PIO-D144/D144U/D144LU | | | | |
| PIO-D168/D168U | | | | |
| PEX-D24/D56 | | | | |
| PEX-D48 | | | | |
| PIO-DA4/DA4U, PEX-DA4 | | | | |
| PIO-DA8/DA8U, PEX-DA8 | | | | |
| PIO-DA16/DA16U,PEX-DA16 | | | | |
| PIO-821 Series | ✔ | ✔ | ✔ | ✔ |
| PISO-P64/P64U, PEX-P64 | | | | |
| PISO-A64/C64/C64U, PEX-C64 | | | | |
| PISO-P32C32/P32C32U/P32S32WU | | | | |
| PEX-P32C32 | | | | |
| PISO-P32A32/P32A32U | | | | |
| PISO-725/730/730A | | | | |
| PISO-P16R16U | | | | |
| PEX-P16R16i/P8R8i | | | | |
| PISO-P8R8/P8R8U | | | | |
| PISO-P8R8UDC/AC | | | | |
| PISO-DA4U/DA8U/DA16U | | | | |
| PISO-DA2/DA2U | | | | |
| PISO-813/813U | ✔ | ✔ | ✔ | ✔ |
| PCI-TMC12A | | | | |
| PCI-M512/M512U | | | | |
| PCI-P16R16/P16POR16 | | | | |
| PCI-P16C16/P8R8 | | | | |
| PEX-P16POR16i/P8POR8i | | | | |
| PCI-1002 series, PEX-1002 series | ✔ | ✔ | ✔ | ✔ |
| PCI-1202 series, PEX-1202 series | ✔ | ✔ | ✔ | ✔ |
| PCI-1602 series | ✔ | ✔ | ✔ | ✔ |
| PCI-1800/1802 series | ✔ | ✔ | ✔ | ✔ |
| PCI-822LU/826LU | ✔ | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ | ✔ |
| PCI-FC16U | | | | |

| 函式名稱 | lxud_StartAI<br>lxud_StopAI | lxud_StartAIScan | lxud_StartExtAI | lxud_StartExtAIScan | lxud_GetAIBuffer<br>lxud_GetAIBufferH |
|---|---|---|---|---|---|
| PIO-D24/D24U/D56/D56U | | | | | |
| PIO-D48/D48U/D48SU | | | | | |
| PIO-D64/D64U | | | | | |
| PIO-D96/D96U/D96SU | | | | | |
| PIO-D144/D144U/D144LU | | | | | |
| PIO-D168/D168U | | | | | |
| PEX-D24/D56 | | | | | |
| PEX-D48 | | | | | |
| PIO-DA4/DA4U, PEX-DA4 | | | | | |
| PIO-DA8/DA8U, PEX-DA8 | | | | | |
| PIO-DA16/DA16U,PEX-DA16 | | | | | |
| PIO-821 Series | ✔ | | | | ✔ |
| PISO-P64/P64U, PEX-P64 | | | | | |
| PISO-A64/C64/C64U, PEX-C64 | | | | | |
| PISO-P32C32/P32C32U/P32S32WU | | | | | |
| PEX-P32C32 | | | | | |
| PISO-P32A32/P32A32U | | | | | |
| PISO-725/730/730A | | | | | |
| PISO-P16R16U | | | | | |
| PEX-P16R16i/P8R8i | | | | | |
| PISO-P8R8/P8R8U | | | | | |
| PISO-P8R8UDC/AC | | | | | |
| PISO-DA4U/DA8U/DA16U | | | | | |
| PISO-DA2/DA2U | | | | | |
| PISO-813/813U | | | | | |
| PCI-TMC12A | | | | | |
| PCI-M512/M512U | | | | | |
| PCI-P16R16/P16POR16 | | | | | |
| PCI-P16C16/P8R8 | | | | | |
| PEX-P16POR16i/P8POR8i | | | | | |
| PCI-1002 series, PEX-1002 series | ✔ | ✔ | | | ✔ |
| PCI-1202 series, PEX-1202 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1602 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-1800/1802 series | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-822LU/826LU | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ | ✔ | ✔ |
| PCI-FC16U | | | | | |

| 函式名稱 | lxud_ConfigAO | lxud_WriteAOVoltage<br>lxud_WriteAOVoltageH | lxud_WriteAOCurrent<br>lxud_WriteAOCurrentH |
|---|---|---|---|
| PIO-D24/D24U/D56/D56U | | | |
| PIO-D48/D48U/D48SU | | | |
| PIO-D64/D64U | | | |
| PIO-D96/D96U/D96SU | | | |
| PIO-D144/D144U/D144LU | | | |
| PIO-D168/D168U | | | |
| PEX-D24/D56 | | | |
| PEX-D48 | | | |
| PIO-DA4/DA4U, PEX-DA4 | ✔ | ✔ | ✔ |
| PIO-DA8/DA8U, PEX-DA8 | ✔ | ✔ | ✔ |
| PIO-DA16/DA16U,PEX-DA16 | ✔ | ✔ | ✔ |
| PIO-821 Series | ✔ | ✔ | ✔ |
| PISO-P64/P64U, PEX-P64 | | | |
| PISO-A64/C64/C64U, PEX-C64 | | | |
| PISO-P32C32/P32C32U/P32S32WU | | | |
| PEX-P32C32 | | | |
| PISO-P32A32/P32A32U | | | |
| PISO-725/730/730A | | | |
| PISO-P16R16U | | | |
| PEX-P16R16i/P8R8i | | | |
| PISO-P8R8/P8R8U | | | |
| PISO-P8R8UDC/AC | | | |
| PISO-DA4U/DA8U/DA16U | ✔ | ✔ | ✔ |
| PISO-DA2/DA2U | ✔ | ✔ | ✔ |
| PISO-813/813U | | | |
| PCI-TMC12A | | | |
| PCI-M512/M512U | | | |
| PCI-P16R16/P16POR16 | | | |
| PCI-P16C16/P8R8 | | | |
| PEX-P16POR16i/P8POR8i | | | |
| PCI-1002 series, PEX-1002 series | | | |
| PCI-1202 series, PEX-1202 series | ✔ | ✔ | ✔ |
| PCI-1602 series | ✔ | ✔ | ✔ |
| PCI-1800/1802 series | ✔ | ✔ | ✔ |
| PCI-822LU/826LU | ✔ | ✔ | ✔ |
| PCI-2602U | ✔ | ✔ | ✔ |
| PCI-FC16U | | | |

| 函式名稱 | Ixud_ReadCounter | Ixud_SetCounter | Ixud_DisableCounter |
|---|---|---|---|
| PIO-D24/D24U/D56/D56U | | | |
| PIO-D48/D48U/D48SU | ✔ | ✔ | ✔ |
| PIO-D64/D64U | ✔ | ✔ | ✔ |
| PIO-D96/D96U/D96SU | | | |
| PIO-D144/D144U/D144LU | | | |
| PIO-D168/D168U | | | |
| PEX-D24/D56 | | | |
| PEX-D48 | ✔ | ✔ | ✔ |
| PIO-DA4/DA4U, PEX-DA4 | ✔ | ✔ | ✔ |
| PIO-DA8/DA8U, PEX-DA8 | ✔ | ✔ | ✔ |
| PIO-DA16/DA16U,PEX-DA16 | ✔ | ✔ | ✔ |
| PIO-821 Series | ✔ | ✔ | ✔ |
| PISO-P64/P64U, PEX-P64 | | | |
| PISO-A64/C64/C64U, PEX-C64 | | | |
| PISO-P32C32/P32C32U/P32S32WU | | | |
| PEX-P32C32 | | | |
| PISO-P32A32/P32A32U | | | |
| PISO-725/730/730A | | | |
| PISO-P16R16U | | | |
| PEX-P16R16i/P8R8i | | | |
| PISO-P8R8/P8R8U | | | |
| PISO-P8R8UDC/AC | | | |
| PISO-DA4U/DA8U/DA16U | ✔ | ✔ | ✔ |
| PISO-DA2/DA2U | ✔ | ✔ | ✔ |
| PISO-813/813U | | | |
| PCI-TMC12A | ✔ | ✔ | ✔ |
| PCI-M512/M512U | | | |
| PCI-P16R16/P16POR16 | | | |
| PCI-P16C16/P8R8 | | | |
| PEX-P16POR16i/P8POR8i | | | |
| PCI-1002 series, PEX-1002 series | | | |
| PCI-1202 series, PEX-1202 series | | | |
| PCI-1602 series | | | |
| PCI-1800/1802 series | | | |
| PCI-822LU/826LU | | | |
| PCI-2602U | ✔ | ✔ | ✔ |
| PCI-FC16U | | | |

| Function name | PCI-M512U |
|---|---|
| Ixud_ReadMemory<br>Ixud_WriteMemory | ✔ |
| Ixud_ReadMemory32<br>Ixud_WriteMemory32 | ✔ |

| 函式名稱 | PCI-2602U |
|---|---|
| Ixud_StartDO<br>Ixud_StopDO | ✔ |
| Ixud_StartDI<br>Ixud_StopDI<br>Ixud_GetDIBufferH | ✔ |
| Ixud_ConfiAIEx<br>Ixud_StartExtAnalogTrigger | ✔ |
| Ixud_StartAOVoltage<br>Ixud_StartAOVoltageH<br>Ixud_StopAO | ✔ |

# 5.2. 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 UniDAQ 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 的格式爲一 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

## 5.2.1. 驅動函式集

# *Ixud_GetDllVersion*

取得 UniDAQ SDK 函式庫的版本編號。

- **語法**

```
WORD Ixud_GetDllVersion(
    DWORD *dwDLLVer
);
```

- **參數**

*dwDLLVer*

[Output] 取得 UniDAQ SDK 函式庫的版本值。

- **回傳值**

請參考 A.1. 函數回傳值定義。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_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 UniDAQ 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 Ixud_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。

# Ixud_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 Ixud_DriverClose(
        void
);
```

➢ **參數**

無任何參數。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_SearchCard

此函式提供使用者取得特定的版卡的數量，使用此函式後，此後其餘需透過板卡編號來 I/O 的函式將會以此板卡的排序爲基準。

➢ **語法**

```
WORD Ixud_SearchCard(
   WORD *wTotalBoards,
   DWORD wModelNo
);
```

➢ **參數**

*wTotalBoards*

[Output] 由使用者設定的板卡模組識別號碼，取得該板卡的數量。

*wModelNo*

[Input] 使用者設定板卡模組識跑號碼，此號碼可參考 A.2. 模組識別號碼。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_GetBoardNoByCardID*

此函式提供給使用者，可透過模組識別號碼或卡片識別號碼二種參數來取得該板卡排序的編號。

➢ **語法**

```
WORD Ixud_GetBoardNoByCardID(
        WORD *wBoardNo,
        DWORD dwModelNumber,
        WORD wCardID
);
```

➢ **參數**

*wBoardNo*

[Output] 取得板卡排序的編號。

*dwModelNumber*

[Input] 由使用者設定板卡模組號碼，此序號可參考 A.2. 模組識別號碼。此值設定爲 0 時，wBoardNo 會以 CardID 設定爲主。

*wCardID*

[Input] 由使用者設定板卡的識別號碼。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_GetCardInfo

取得板卡的硬體資料、軟體資料及板卡名稱。

➢ **語法**

```
WORD Ixud_GetCardInfo(
        WORD wBoardNo,
        PIXUD_DEVICE_INFO sDevInfo,
        PIXUD_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 爲 0，第二張板卡的 wBoardNo 爲 1，依此類推。

*sDevInfo*

[Output] 取得板卡在電腦上的系統資訊。變數型態爲 PIXUD_DEVICE_INFO 結構。

*sCardInfo*

[Output] 取得板卡的硬體規格資訊。變數型態爲一個 PIXUD_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。

# Ixud_ReadPort

從 I/O 埠讀取一個 8/16/32bit 值。

- **語法**

```
WORD Ixud_ReadPort(
        DWORD dwAddress,
        WORD wSize,
        DWORD* dwVal
);
```

- **參數**

*dwAddress*

[Input] 設定 I/O 埠的位址。

*wSize*

[Input] 設定讀取的 bit 長度。

| *wSize* | 長度 (bit) |
|---|---|
| 8 | 8 (Byte) |
| 16 | 16 (WORD) |
| 32 | 32 (DWORD) |

表格 5-1 wSize 參數設定

*dwVal*

[Output] 取得 I/O 埠的值。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WritePort

從 I/O 埠寫入一個 8/16/32bit 的值。

➢ **語法**

**WORD Ixud_WritePort(**
**DWORD** dwAddress**,**
**WORD** wSize**,**
**DWORD** dwVal
**);**

➢ **參數**

*dwAddress*

[Input] 設定 I/O 埠的位址。

*wSize*

[Input] 設定寫入的 bit 長度。

| *wSize* | 長度 (bit) |
|---|---|
| 8 | 8 (Byte) |
| 16 | 16 (WORD) |
| 32 | 32 (DWORD) |

表格 5-2 wSize 參數設定

*dwVal*

[Input] 寫入 I/O 埠的值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ReadPort32

從 I/O 埠讀取一個 32bit 的值。Visual Basic 6.0 使用者建議使用此函式讀取 32 位元數值。

➢ **語法**

```
WORD Ixud_ReadPort32(
    DWORD dwAddress,
    DWORD* dwLow,
    DWORD* dwHigh
);
```

➢ **參數**

*dwAddress*

[Input] 設定 I/O 埠的位址。

*dwLow*

[Output] 取得 I/O 埠 32bit 低位元部分的值。

*dwHigh*

[Output] 取得 I/O 埠 32bit 高位元部分的值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WritePort32

從 I/O 埠寫入一個 32bit 的值。Visual Basic 6.0 的使用者建議使用此函式寫入 32 位元數值。

➢ **語法**

```
WORD Ixud_WriteReadPort32(
    DWORD dwAddress,
    DWORD dwLow,
    DWORD dwHigh
);
```

➢ **參數**

*dwAddress*

[Input] 設定 I/O 埠的位址。

*dwLow*

[Input] 寫入 I/O 埠 32bit 低位元部份的值。

*dwHigh*

[Input] 寫入 I/O 埠 32bit 高位元部份的值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ReadPhyMemory

從 memory mapping I/O 埠讀取一個 8/16/32bit 值。

➢ **語法**

**WORD Ixud_ReadPhyMemory(**
**DWORD** dwAddress**,**
**WORD** wSize**,**
**DWORD*** dwValue
**);**

➢ **參數**

*dwAddress*

[Input] 設定 I/O 埠的位址。

*wSize*

[Input] 設定讀取的 bit 長度。

| *wSize* | 長度 (bit) |
|---|---|
| 8 | 8 (Byte) |
| 16 | 16 (WORD) |
| 32 | 32 (DWORD) |

表格 5-3 wSize 參數設定

*dwValue*

[Output] 取得 memory mapping I/O 埠的值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WritePhyMemory

從 memory mapping I/O 埠寫入一個 8/16/32bit 的值。

➢ **語法**

**WORD Ixud_WritePort(**
**DWORD** dwAddress**,**
**WORD** wSize**,**
**DWORD** dwValue
**);**

➢ **參數**

*dwAddress*

[Input] 設定 memory mapping I/O 埠的位址。

*wSize*

[Input] 設定寫入的 bit 長度。

| *wSize* | 長度 (bit) |
|---|---|
| 8 | 8 (Byte) |
| 16 | 16 (WORD) |
| 32 | 32 (DWORD) |

表格 5-4 wSize 參數設定

*dwValue*

[Input] 寫入 memory mapping I/O 埠的值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

## 5.2.2. 數位輸出輸入函式集

# *Ixud_SetDIOModes32*

同時設定多個埠的輸入/出模式，僅支援含有雙向 Digital I/O 功能的埠。

➢ **語法**

```
WORD Ixud_SetDIOModes32(
        WORD wBoardNo,
        DWORD dwDioModeMask
);
```

➢ **參數**

*wBoardNo*

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

*dwDioModeMask*

[Input] 設定雙向 Digital I/O 埠爲輸出模式或輸入模式，每個 bit 代表一個埠，最高可設定到 32 個埠，bit0 代表第 0 個埠，依此類推。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼及 A.5. 數位輸出埠定義號碼。

| 設定值 | 埠模式 |
|---|---|
| 0 | 輸入模式 |
| 1 | 輸出模式 |

表格 5-5 埠模式參數設定

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

➢ **範例**

```
wBoardNo = 0;//設定第一張板卡
dwDioModeMask = 5;//設定埠 0 及 2 爲輸出模式，埠 1 爲輸入模式
wRtn = Ixud_SetDIOModes32(wBoardNo, dwDioModeMask);
```

# Ixud_SetDIOMode

由使用者所選擇的埠號，設定該埠的輸入/出模式，僅支援含有雙向 Digital I/O 功能的埠。

➢ **語法**

```
WORD Ixud_SetDIOMode(
        WORD wBoardNo,
        WORD wPortNo,
        WORD wDioMode
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。

*wDioMode*

[Input] 設定雙向 Digital I/O 埠爲輸出模式或輸入模式。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼及 A.5. 數位輸出埠定義號碼。

| 設定值 | 埠模式 |
|---|---|
| 0 | 輸入模式 |
| 1 | 輸出模式 |

表格 5-6 埠模式參數設定

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

➢ **範例**

```
wRtn = Ixud_SetDIOMode(0, 1, 1);//設定埠 1 爲輸出埠
```

# Ixud_ReadDI

讀取使用者所指定的輸入埠的資料。

- **語法**

```
WORD Ixud_ReadDI(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD *dwDIVal
);
```

- **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼。

*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。

# Ixud_WriteDO

寫入資料到所用者所指定的輸出埠。

- **語法**

```
WORD Ixud_WriteDO(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD dwDOVal
);
```

- **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.5. 數位輸出埠定義號碼。

*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。

# Ixud_ReadDIBit

讀取使用者所指定的輸入通道內的資料。如果使用者需要同步使用多個通道來操作數位輸入功能，建議使用 Ixud_ReadDI 函式能得到較高的效能。

➢ **語法**

```
WORD Ixud_ReadDIBit(
        WORD wBoardNo,
        WORD wPortNo,
        WORD wBitNo,
        WORD *wDIVal
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼。

*wBitNo*

[Input] 由使用者指定的通道號。

*wDIVal*

[Output] 取得輸入通道的資料。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WriteDOBit

寫入資料到所用者所指定的輸出通道。如果使用者需要同步使用多個通道來操作數位輸出功能，建議使用 Ixud_WriteDO 函式能得到較高的效能。

➢ **語法**

```
WORD Ixud_WriteDO(
        WORD wBoardNo,
        WORD wPortNo,
        WORD wBitNo,
        WORD wDOVal
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.5. 數位輸出埠定義號碼。

*wBitNo*

[Input] 由使用者指定的通道號。

*wDOVal*

[Input]寫入資料至輸出通道。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ReadDI32

從使用者所指定的輸出埠讀取出 32bit 值。不帶有無號整數型態的編程語言的使用者，如 Visual Basic 6.0，建議使用此函式。

➢ **語法**

```
WORD Ixud_ReadDI32(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD* dwLow,
        DWORD* dwHigh,
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼。

*dwLow*

[Output] 讀取輸入埠裡 bit 0 ~ 15 的資料。

*dwHigh*

[Output] 讀取輸入埠裡 bit 16 ~ 31 的資料。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WriteDO32

寫入 32-Bit 資料到所用者所指定的輸出埠，不帶有無號整數型態的編程語言的使用者，如 Visual Basic 6.0 建議使用此函式。

➢ **語法**

```
WORD Ixud_WriteDO32(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD dwLow,
        DWORD dwHigh,
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.5. 數位輸出埠定義號碼。

*dwLow*

[Input]寫入 bit 0 ~ 15 的資料至輸入埠。

*dwHigh*

[Input] 寫入 bit16 ~ 31 的資料至輸出埠。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_SoftwareReadbackDO

在使用函式庫的期間軟體將會自動記錄數位輸出值，讓使用者可以透過函式庫的協助讀取輸出埠的值(非暫存器狀態)。

➢ **語法**

```
WORD Ixud_SoftwareReadbackDO(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD *dwDOVal
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

**[Input]** 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.5. 數位輸出埠定義號碼。

*dwDOVal*

[Output] 回讀輸出埠的寫入值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StartDI

透過此函式來初始化數位輸入埠、設定取樣頻率、採樣資料量，並開始啓動數位資料擷取，啓動後將以等速度擷取單一埠的數位資料並將資料暫存至系統記憶體內，可透過 Ixud_GetDIBufferH 函式取出類比輸入資料，欲停止高速類比輸入採集功能時，需呼叫 Ixud_StopDI 函式。

本函式僅支援 PCI-2602U

爲了確保資料採集的準確度，使用此函式在資料採集的過程中時將會佔用 CPU 一段短暫的時間至採集數據完成，此時間將取決於使用者設定的採集資料量及取樣頻率。

➢ **語法**

```
WORD Ixud_StartDI(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD dwReserved,
        float fSamplingRate,
        DWORD dwDataCount
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼。

*dwReserved*

[Input] 保留。

*fSamplingRate*

[Input] 爲一浮點數值，由使用者設定數位輸入埠的取樣頻率(次/秒)。

*dwDataCount*

[Input] 由使用者設定需要採集的資料筆數。使用 Ixud_StopDI 函式停止採集。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_StartDO*

依使用者指定的數據大小、數據緩衝區及循環模式啓動高速數位輸出模式。

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_StartDO(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD dwReserved,
        float fFrequency,
        DWORD dwDataCount,
        DWORD dwCycleNum,
        DWORD dwDOBuf[ ]
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

**[Input]** 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.5. 數位輸出埠定義號碼。

*dwReserved*

[Input] 保留。

*fFrequency*

[Input] 爲一浮點數值，由使用者設定數位輸出埠的輸出每筆數位資料的頻率(次/秒)。

*dwDataCount*

[Input] 由使用者設定需要送出一個波型的資料筆數。使用 Ixud_StopDO 函式停止採集。

*dwCycleNum*

[Input] 0:連續模式，如果要終止高速數位輸出功能，請使用**錯誤! 找不到參照來源。** 函式。

*dwDOBuf[]*

[Input] 儲存類比輸出資料至數位輸出緩衝區。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_GetDIBufferH

此函式將會從記憶體裡取得數位輸入值，並儲存在使用者所指令的陣列裡，其值爲十六進位值。呼叫此函式前需先呼叫 Ixud_StartDI 函式。

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_GetAIBufferH(
        WORD wBoardNo,
        WORD wPortNo,
        DWORD dwDataCount,
        DWORD hValue[ ]
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼。

*dwDataCount*

[Input] 由使用者設定需要的資料筆數。

*hValue[ ]*

[Output] 請宣告一個 DWORD 陣列。用來取得數位輸入值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StopDI

呼叫此函式，用來停止高速數位輸入擷取的運作。

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_StopDI(
        WORD wBoardNo,
        WORD wPortNo
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

[Input] 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.4. 數位輸入埠定義號碼。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StopDO

呼叫此函式，用來停止高速數位輸出的運作。

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_StopDO(
        WORD wBoardNo,
        WORD wPortNo
);
```

➢ **參數**

*wBoardNo*

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

*wPortNo*

**[Input]** 由使用者指定的埠號。詳細埠的對映資訊可參考附錄 A.5. 數位輸出埠定義號碼。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

## 5.2.3. 中斷事件函式集

# *Ixud_SetEventCallback*

此函式用來設定中斷事件發生時的 Callback function，當不再使用 Callback 功能時請呼叫 Ixud_RemoveEventCallback 函式來關閉此功能。

➢ **語法**

```
WORD Ixud_SetEventCallback(
        WORD wBoardNo,
        WORD wEventTypeMask,
        WORD wInterruptSource,
        HANDLE *hEvent,
        PVOID CallbackFun,
        DWORD dwCallBackParameter
);
```

➢ **參數**

*wBoardNo*

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

*wEventType*

[Input] 由使用者設定事件發生的條件，每個 bit 代表一種模式，可同時開啓。設定值請參考 A.3.4. 中斷事件配置碼

*wInterruptSource*

[Input] 由使用者設定事件發生時所啓用的中斷源。各板卡使用的中斷源設定值請參考下面表格。

| wInterrupt Source | PIO-D24U<br>PEX-D24<br>PIO-D56U<br>PEX-D56 | PIO-D48U<br>PIO-D48SU<br>PEX-D48 | PIO-D64U | PIO-D96U<br>PIO-D96SU | PIO-D144U<br>PIO-D144LU<br>PIO-D168U |
|---|---|---|---|---|---|
| 1 | P2C0 | P2C3/P2C7 | EXTIRQ | P2C0 | P2C0 |
| 2 | P2C1 | P5C3/P5C7 | EVTIRQ | P5C0 | P2C1 |
| 3 | P2C2 | COUT0 | TMRIRQ | P8C0 | P2C2 |
| 4 | P2C3 | COUT2 | - | P11C0 | P2C3 |

| wInterrupt Source | PIO-DA4U<br>PIO-DA8U<br>PIO-DA16U<br>PISO-DA4U<br>PISO-DA8U<br>PISO-DA16U | PEX-DA4<br>PEX-DA8<br>PEX-DA16 | PISO-730<br>PISO-730A<br>PISO-730U<br>PISO-730AU | PISO-725 | PCI-TMC12A | PIO-821<br>PCI-1002<br>PEX-1002 | PCI-822LU<br>PCI-826LU<br>PCI-2602U |
|---|---|---|---|---|---|---|---|
| 1 | COUT0 | COUT0 | DI0 | IDI0 | COUT3/6/9/12/Ext | AD Data | AD Data |
| 2 | COUT2 | COUT2 | DI1 | IDI1 | - | - | - |
| 3 | - | | - | IDI2 | - | - | - |
| 4 | - | | - | IDI3 | - | - | - |
| 5 | - | | - | IDI4 | - | - | - |
| 6 | - | | - | IDI5 | - | - | - |
| 7 | - | | - | IDI6 | - | - | - |
| 8 | - | | - | IDI7 | - | - | - |

*Digital Input*
*Timer/Counter*
*Analog Input*

*hEvent*

[Input] 由使用者傳入事件的指標，此事件請使用 Windows API CreateEvent(..) 函式創造，或設定爲 0，系統會自動生成一個事件。

*CallbackFun*

[Input] 由使用者設定事件所使用的 Callback Function。

*dwCallBackParameter*

[Input] 由使用者設定事件所使用 Callback Function 的參數，wEventType 設置爲 ADC Ready 模式時此參數可用來設定 ADCReady 的資料量。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

➢ **範例**

DI Callback

```
//Set DI Callback function
// Use Source = 1 Event Type = Active High +Hardware Interrupt
wRtn = Ixud_SetEventCallback(wBoardNo, IXUD_HARDWARE_INT|IXUD_ACTIVE_HIGH , 1, hEvent0,
Callbackfun0, 0);

//Use Source = 3 Event Type = Active Low +Hardware Interrupt
wRtn = Ixud_SetEventCallback(wBoardNo, IXUD_HARDWARE_INT|IXUD_ACTIVE_LOW, 3, hEvent2,
Callbackfun2, 0);
```

AI Callback

```
// Set AI Callback function
// Use Source = 1 Event Type = APC Ready+Hardware Interrupt  每筆AD資料產生一次Callback Event
DataNum=1000;
wRtn = Ixud_SetEventCallback(wBoardNo, IXUD_HARDWARE_INT|IXUD_APC_READY_INT  , 1, hEvent0,
Callbackfun0,DataNum);
```

# Ixud_RemoveEventCallback

此函式用來關閉或移除中斷事件所發生的 Callback function。使用此函式需在使用 Ixud_SetEventCallback 函式之後，終結使用 Callback function 前呼叫。

- **語法**

```
WORD Ixud_RemoveEventCallback(
        WORD wBoardNo,
        WORD wInterruptSource
);
```

- **參數**

*wBoardNo*

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

*wInterruptSource*

[Input] 由使用者設定事件發生時所啓用的中斷源，第一個中斷源的 wInterruptSource 爲 1，依此類推。如果設置爲 0 將會一次移除該板卡所使用的事件。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_InstallIrq

此函式用來安裝設定中斷常式，並支援同時啓動多個中斷源。

註：類比輸入中斷功能的使用者請勿呼叫此函式。

- **語法**

```
WORD Ixud_InstallIrq(
        WORD wBoardNo,
        DWORD dwInterruptMask
);
```

- **參數**

*wBoardNo*

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

*dwInterruptMask*

[Input] 由使用者設定安裝中斷源，每個 bit 代表一個中斷源，bit0 爲第 0 個中斷源，依此類推。

| Bit | PIO-D24U<br>PEX-D24<br>PIO-D56U<br>PEX-D56 | PIO-D48U<br>PEX-D48 | PIO-D64U | PIO-D96U | PIO-D144U<br>PIO-D144LU<br>PIO-D168U | PIO-DA4U<br>PIO-DA8U<br>PIO-DA16U<br>PISO-DA4U<br>PISO-DA8U<br>PISO-DA16U | PEX-DA4<br>PEX-DA8<br>PEX-DA16 |
|---|---|---|---|---|---|---|---|
| 0 | P2C0 | P2C3/P2C7 | EXTIRQ | P2C0 | P2C0 | COUT0 | COUT0 |
| 1 | P2C1 | P5C3/P5C7 | EVTIRQ | P5C0 | P2C1 | COUT2 | COUT2 |
| 2 | P2C2 | COUT0 | TMRIRQ | P8C0 | P2C2 | - | |
| 3 | P2C3 | COUT2 | - | P11C0 | P2C3 | - | |

| Bit | PISO-730<br>PISO-730A<br>PISO-730U<br>PISO-730AU | PISO-725 | PCI-TMC12A |
|---|---|---|---|
| 0 | DI0 | IDI0 | COUT3/6/9/12/<br>EXT |
| 1 | DI1 | IDI1 | - |
| 2 | - | IDI2 | - |
| 3 | - | IDI3 | - |
| 4 | - | IDI4 | - |
| 5 | - | IDI5 | - |
| 6 | - | IDI6 | - |
| 7 | - | IDI7 | - |

*Digital Input*

*Timer/Counter*

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

➢ **範例**

```
dwInterruptMask = 0xF //(開啓INT_0,INT_1,INT_2及INT_3)
wRtn=Ixud_InstallIrq(wBoardNo,dwInterruptMask);
```

# Ixud_RemoveIrq

此函式用來關閉中斷常式。

- **語法**

**WORD Ixud_RemoveIrq(**

**WORD** wBoardNo

**);**

- **參數**

*wBoardNo*

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

- **回傳值**

請參考 A.1. 函數回傳值定義。

## 5.2.4. 類比輸入函式集

# *Ixud_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 Ixud_ConfigAI(
        WORD wBoardNo,
        WORD wFIFOSizeKB,
        DWORD BufferSizeCount,
        WORD wCardType,
        WORD wDelaySettlingTime
);
```

➢ **參數**

*wBoardNo*

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

*wFIFOSizeKB*

[Input] 由使用者設定板卡內建 FIFO 的大小，單位爲 KByte。 建議設定值爲 0 時，會自動依板卡最適合的 FIFO 大小來設定。

*wBufferSizeCount*

[Input] 由使用者設定類比輸入緩衝區資料筆數，此緩衝區將於配置於系統記憶體上之上作爲暫存類比資料值使用，設定值爲 0 時，直接使用預設值 524288 筆 DWORD 大小的資料量約佔用 2MB 的系統記憶體。

*wCardType*

[Input] 由使用者依板卡規格作設定。如果您的板卡爲低倍數板卡請設定爲 0，若爲高倍數板卡請設定爲 1。此設定值將會影響將會影響擷取資料的精度及量測範圍，詳細設定請參考下列表格。

| *wCardType* | PISO-813 | PIO-821 | PCI-1002<br>PEX-1002 | PCI-1202<br>PEX-1202 | PCI-1602 | PCI-1802<br>PCI-1800 | PCI-822<br>PCI-826 |
|---|---|---|---|---|---|---|---|
| 0 | JP1 = 20 V | PIO-821L | PCI-1002LU<br>PEX-1002L | PCI-1202LU | PCI-1602U | PCI-1802LU<br>PCI-1800LU | PCI-822LU<br>PCI-826LU |
| 1 | JP1 = 10V | PIO-821H | PCI-1002HU<br>PEX-1002H | PCI-1202HU | PCI-1602FU | PCI-1802HU<br>PCI-1800HU | - |

| *wCardType* | PCI-2602U |
|---|---|
| 0 | PCI-2602U |
| 1 | - |

表格 5-7 wCardType 參數設定

*wDelaySettingTime*

[Input] 板卡的類比數位轉換器的設定加權時間，單位爲微秒(μs)。此時間的設定會影響到類比輸入的效能，建議設定值爲 0。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ConfigAIEx

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 Ixud_ConfigAIEx(
        WORD wBoardNo,
        WORD wFIFOSizeKB,
        DWORD BufferSizeCount,
        WORD wCardType,
        WORD wDelaySettlingTime,
        DWORD dwMode
);
```

➢ **參數**

*wBoardNo*

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

*wFIFOSizeKB*

[Input] 由使用者設定板卡內建 FIFO 的大小，單位爲 KByte。 建議設定值爲 0 時，會自動依板卡最適合的 FIFO 大小來設定。

*wBufferSizeCount*

[Input] 由使用者設定類比輸入緩衝區資料筆數，此緩衝區將於配置於系統記憶體上之上作爲暫存類比資料值使用，設定值爲 0 時，直接使用預設值 524288 筆 DWORD 大小的資料量約佔用 2MB 的系統記憶體。

*wCardType*

[Input] 由使用者依板卡規格作設定。如果您的板卡爲低倍數板卡請設定爲 0，若爲高倍數板卡請設定爲 1。此設定值將會影響將會影響擷取資料的精度及量測範圍，詳細設定請參考下列表格。

| *wCardType* | PISO-813 | PIO-821 | PCI-1002<br>PEX-1002 | PCI-1202<br>PEX-1202 | PCI-1602 | PCI-1802<br>PCI-1800 | PCI-822<br>PCI-826 |
|---|---|---|---|---|---|---|---|
| 0 | JP1 = 20 V | PIO-821L | PCI-1002LU<br>PEX-1002L | PCI-1202LU | PCI-1602U | PCI-1802LU<br>PCI-1800LU | PCI-822LU<br>PCI-826LU |
| 1 | JP1 = 10V | PIO-821H | PCI-1002HU<br>PEX-1002H | PCI-1202HU | PCI-1602FU | PCI-1802HU<br>PCI-1800HU | - |

| *wCardType* | PCI-2602U |
|---|---|
| 0 | PCI-2602U |
| 1 | - |

表格 5-8 wCardType 參數設定

*wDelaySettingTime*

[Input] 板卡的類比數位轉換器的設定加權時間，單位爲微秒( $\mu$ s)。此時間的設定會影響到類比輸入的效能，建議設定值爲 0。

*dwMode*

[Input] The analog input data transfer mode.

| *dwMode* | PCI-2602U |
|---|---|
| ENABLEDMAAI | 使用 DMA 傳輸 |

Table 5-9 dwMode Parameters setting

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_ClearAIBuffer*

清除系統記憶體裡類比輸入緩衝區的資料。

➢ **語法**

```
WORD Ixud_ClearAIBuffer(
        WORD wBoardNo
);
```

➢ **參數**

*wBoardNo*

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

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ClearAIBuffer

清除類比輸入緩衝區的資料。

- **語法**

```
WORD Ixud_ClearAIBuffer(
        WORD wBoardNo,
);
```

- **參數**

*wBoardNo*

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

- **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_GetBufferStatus

取得類比輸入緩衝區的狀態代碼及資料量。

- **語法**

```
WORD Ixud_GetBufferStatus(
        WORD wBoardNo,
        WORD *wBufferStatus,
        DWORD *dwDataCount
);
```

- **參數**

*wBoardNo*

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

*wBufferStatus*

[Output] 取得類比緩衝區的使用狀態。狀態代碼請參考下圖。

| *wBufferStatus* | 狀態簡述 |
|---|---|
| 0 | 無任何資料。 |
| 1 | 正常。已有資料且未溢滿。 |
| 2 | 緩衝區溢滿。 |
| 3 | 系統未分配緩衝區。 |
| 4 | FIFO 溢滿。 |
| 5 | 其他異常狀態。 |

表格 5-10 類比緩衝區狀態代碼說明

*dwDataCount*

[Output] 取得類比緩衝區的資料量(筆數)。

- **回傳值**

  請參考 A.1. 函數回傳值定義。

# Ixud_ReadAI

讀取一筆的類比輸入通道內的電壓值，單位爲 volts。

➢ **語法**

```
WORD Ixud_ReadAI(
        WORD wBoardNo,
        WORD wChannel,
        WORD wConfig,
        float *fValue
);
```

➢ **參數**

*wBoardNo*

[Input] 由使用者指定的板卡編號，第一張板卡的 wBoardNo 爲 0，第二張板卡的 wBoardNo 爲 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。

*fValue*

[Output] 讀取類比輸入通道內的資料。單位爲 volts。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ReadAIH

讀取一筆的類比輸入通道內的數值，此值爲十六進制。

➢ **語法**

```
WORD Ixud_ReadAIH(
        WORD wBoardNo,
        WORD wChannel,
        WORD wConfig,
        DWORD *dwValue
);
```

➢ **參數**

*wBoardNo*

[Input] 由使用者指定的板卡編號，第一張板卡的 wBoardNo 爲 0，第二張板卡的 wBoardNo 爲 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*

[Output] 讀取類比輸入通道內的資料。此值爲十六進制值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_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 Ixud_PollingAI(
        WORD wBoardNo,
        WORD wChannel,
        WORD wConfig,
        DWORD dwDataCount,
        float fValue[ ]
);
```

➢ **參數**

*wBoardNo*

[Input] 由使用者指定的板卡編號，第一張板卡的 wBoardNo 爲 0，第二張板卡的 wBoardNo 爲 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。

*dwDataCount*

[Input] 由使用者設定需要擷取的類比輸入資料筆數。

*fValue[ ]*

[Output] 請宣告一個浮點數陣列(陣列大小爲 dwDataCount)，將輪詢後得到的每筆類比輸入資料儲存在此陣列裡，單位爲 volts。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_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 Ixud_PollingAIH(
        WORD wBoardNo,
        WORD wChannel,
        WORD wConfig,
        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 爲 0，第二張板卡的 wBoardNo 爲 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。

*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(陣列大小爲 dwDataCount)，將輪詢後得到的每筆類比輸入資料儲存在此陣列裡，此數值爲十六進制。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_PollingAIScan*

此函式以軟體輪詢的方式取得多個通道裡複數筆數的類比輸入資料，單位爲 volts。

➢ **語法**

```
WORD Ixud_PollingAIScan(
        WORD wBoardNo,
        WORD wChannels,
        WORD wChannelList[ ],
        WORD wConfigList[ ],
        DWORD dwDataCountPerChannel,
        float fValue[ ]
);
```

➢ **參數**

*wBoardNo*

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

*wChannels*

[Input] 由使用者設定將擷取 wChannels 個通道擷取資料的配置值。

*wChannelList[ ]*

[Input] 一個 WORD 陣列(陣列大小等於 wChannles)。由使用者設定欲使用來做擷取資料的類比輸入通道。請從陣列第 0 個元素開始設定第一個類比輸入通道號碼。

*wConfigList[ ]*

[Input] 請宣告一個 WORD 陣列(陣列大小至少大於 wChannels)。由使用者設定每個通道的配置值，從陣列第 0 個元素開始設定第一個類比輸入通道的配置碼。配置值代碼請參考 A.3.1. 類比輸入配置碼。

*dwDataCountPerChannel*

[Input] 設定每一個通道的取樣數量。

*fValue[ ]*

[Output] 請宣告一個浮點數陣列，陣列大小爲 wChannels 乘以 dwDataCountPerChannel。將輪詢使用者設定的每個通道後所得到的每筆類比輸入資料儲存在此陣列裡，此資料爲浮點數值，此陣列儲存排列方式請參考表格 5-11。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

➢ **使用範例**

```
DWORD dwDataCountPerChannel = 2 //每個通道擷取2個類比資料
wChannels = 3                   //總共使用3個通道
float fValue[dwDataCounterPerChannel*wChannels]; //宣告一個2 x3大小的類比資料陣列
wChannelList[0]= 5       //第1次擷取類比輸入通道5的資料
wChannelList[1]= 3       //第2次擷取類比輸入通道3的資料
wChannelList[2]= 6       //第3次擷取類比輸入通道6的資料
wConfigList[0]= IXUD_BI_10V   //第1次擷取類比輸入通道5的資料量測範圍爲+/-10V
wConfigList[1]= IXUD_BI_5V    //第2次擷取類比輸入通道3的資料量測範圍爲+/-5V
wConfigList[2]= IXUD_BI_2V5   //第3次擷取類比輸入通道6的資料量測範圍爲+/-2.5V
wRtn = lxud_PollingAIScan(wBoardNo, wChannels, wChannelList,wConfigList, dwDataCountPerChannel,
fValue)
```

呼叫函式後各個通道的輸入值儲存至陣列(fValue[])裡，而儲存方式如下圖：

| | |
|---|---|
| 0 | Channel 5 Value 0 |
| 1 | Channel 3 Value 0 |
| 2 | Channel 6 Value 0 |
| 3 | Channel 5 Value 1 |
| 4 | Channel 3 Value 1 |
| 5 | Channel 6 Value 1 |

表格 5-11 陣列儲存方式

# Ixud_PollingAIScanH

此函式以軟體輪詢的方式一次取得多個通道裡複數筆數的類比輸入值，此值爲十六進值。

➢ **語法**

```
WORD Ixud_PollingAIScanH(
        WORD wBoardNo,
        WORD wChannels,
        WORD wChannelList[ ],
        WORD wConfigList[ ],
        DWORD dwDataCountPerChannel,
        DWORD dwValue[ ]
);
```

➢ **參數**

*wBoardNo*

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

*wChannels*

[Input] 由使用者設定將擷取 wChannels 個通道擷取資料的配置值。

*wChannelList[ ]*

[Input] 一個 WORD 陣列(陣列大小等於 wChannles)。由使用者設定欲使用來做擷取資料的類比輸入通道。請從陣列第 0 個元素開始設定第一個類比輸入通道號碼。

*wConfigList[ ]*

[Input] 請宣告一個 WORD 陣列(陣列大小至少大於 wChannels)。由使用者設定每個通道的配置值，從陣列第 0 個元素開始設定第一個類比輸入通道的配置碼。

*dwDataCountPerChannel*

[Input] 設定每一個通道的取樣數量。

*dwValue[ ]*

[Output] 請宣告一個雙倍無號整數陣列，陣列大小至少大於 wChannels 乘以 dwDataCountPerChannel。將輪詢使用者設定的每個通道後所得到的每筆類比輸入資料儲存在此陣列裡，此資料爲十六進值，此陣列儲存排列方式請參考表格 5-12。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

➢ **使用範例**

```
DWORD dwDataCountPerChannel = 2 //每個通道擷取2個類比資料
wChannels = 3                   //總共使用3個通道
DWORD dwValue[dwDataCounterPerChannel*wChannels]; //宣告一個2 x3大小的類比資料陣列
wChannelList[0] = 5             //第1次擷取類比輸入通道5的資料
wChannelList[1] = 3             //第2次擷取類比輸入通道3的資料
wChannelList[2] = 6             //第3次擷取類比輸入通道6的資料
wConfigList[0] = IXUD_BI_10V //第1次擷取類比輸入通道5的資料量測範圍爲+/-10V
wConfigList[1] = IXUD_BI_5V  //第2次擷取類比輸入通道3的資料量測範圍爲+/-5V
wConfigList[2] = IXUD_BI_2V5 //第3次擷取類比輸入通道6的資料量測範圍爲+/-2.5V
```

呼叫函式後各個通道的輸入值儲存至陣列(dwValue[])裡，而儲存方式如下表：

| | |
|---|---|
| 0 | Channel 5 Val0 |
| 1 | Channel 3 Val0 |
| 2 | Channel 6 Val0 |
| 3 | Channel 5 Val1 |
| 4 | Channel 3 Val1 |
| 5 | Channel 6 Val1 |

表格 5-12 陣列儲存方式

# Ixud_StartAI

透過此函式來初始化 ADC、設定取樣頻率、採樣資料量，並開始啓動類比資料擷取，啓動後將以等速度擷取單一通道的類比資料並將資料暫存至系統記憶體內，可透過 Ixud_GetAIBuffer 函式或 Ixud_GetAIBufferH 函式取出類比輸入資料，欲停止類比輸入採集功能時，需呼叫 Ixud_StopAI 函式。

爲了確保資料採集的準確度，使用此函式在資料採集的過程中時將會佔用 CPU 一段短暫的時間至採集數據完成，此時間將取決於使用者設定的採集資料量及取樣頻率。

➢ **語法**

```
WORD Ixud_StartAI(
        WORD wBoardNo,
        WORD wChannel,
        WORD wConfig,
        float fSamplingRate,
        DWORD dwDataCount
);
```

➢ **參數**

*wBoardNo*

[Input] 由使用者指定的板卡編號，第一張板卡的 wBoardNo 爲 0，第二張板卡的 wBoardNo 爲 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。

*fSamplingRate*

[Input] 爲一浮點數值，由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwDataCount*

[Input] 由使用者設定需要採集的資料筆數。當 dwDataCount 設定爲 0 時，此時將會啓動連續擷取模式，除非使用者使用 Ixud_StopAI 函式停止採集。

使用連續擷取模式時的注意事項

1. 需注意頻率的大小，如果取樣頻率設定過大，由於採集資料量變大，將會很容易造成 FIFO 的溢滿。
2. 連續擷取模式中會將資料暫存在記憶體之中，使用者在適當的時間應將資料取出，否則隨時時間的增長記憶體緩衝區將會溢滿。
3. 採集過程中請盡量縮短對系統負荷過重的動作，例如檔案處理等等，否則將會提高憶體緩衝區溢滿的機率。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StartAIScan

透過此函式來初始化 ADC、設定取樣頻率、採樣資料量，並開始啓動類比資料擷取，啓動後將以等速度擷取多個通道的類比資料並將資料暫存至系統記憶體內，可透過 Ixud_GetAIBuffer 函式或 Ixud_GetAIBufferH 函式取出類比輸入資料，欲停止類比輸入採集功能時，需呼叫 Ixud_StopAI 函式。

爲了確保資料採集的準確度，使用此函式在資料採集的過程中時將會佔用 CPU 一段短暫的時間至採集數據完成，此時間將取決於使用者設定的採集資料量及取樣頻率。

➢ **語法**

```
WORD Ixud_StartAIScan(
        WORD wBoardNo,
        WORD wChannels,
        WORD wChannelList[ ],
        WORD wConfigList[ ],
        float fSamplingRate,
        DWORD dwDataCountPerChannel
);
```

➢ **參數**

*wBoardNo*

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

*wChannels*

[Input] 由使用者設定將擷取 wChannels 個通道擷取資料的配置值。

*wChannelList[ ]*

[Input] 一個 WORD 陣列(陣列大小等於 wChannles)。由使用者設定欲使用來做擷取資料的類比輸入通道。請從陣列第 0 個元素開始設定第一個類比輸入通道號碼。

*wConfigList[ ]*

[Input] 請宣告一個 WORD 陣列(陣列大小至少大於 wChannels)。由使用者設定每個通道的配置值，從陣列第 0 個元素開始設定第一個類比輸入通道的配置碼。

*fSamplingRate*

[Input] 爲一浮點數值，由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwDataCountPerChannel*

[Input] 設定每一個通道的取樣數量。設定值爲 0，將會啓動連續擷取模式，除非使用者使用 Ixud_StopAI 函式停止採集。

使用連續擷取模式時的注意事項

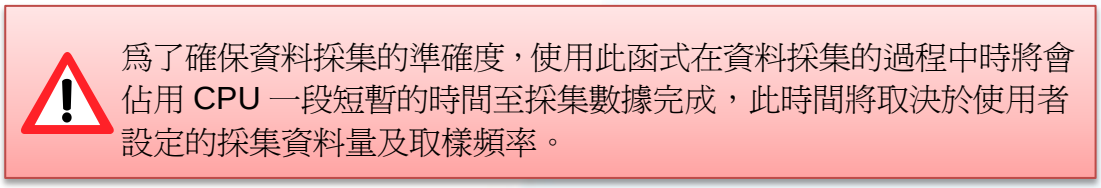
1. 需注意頻率的大小，如果取樣頻率設定過大，由於採集資料量變大，將會很容易造成 FIFO 的溢滿。
2. 連續擷取模式中會將資料暫存在記憶體之中，使用者在適當的時間應將資料取出，否則隨時時間的增長記憶體緩衝區將會溢滿。
3. 採集過程中請盡量縮短對系統負荷過重的動作，例如檔案處理等等，否則將會提高憶體緩衝區溢滿的機率。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StartExtAI

透過此函式來初始化 ADC、設定取樣頻率、採樣資料量，並開始啓動外部觸發(TTL level)類比資料擷取，支援三種觸發模式：post-trigger、middle-trigger 及 pre-trigger 啓動後將以等速度擷取單一通道的類比資料並將資料暫存至系統記憶體內，可透過 Ixud_GetAIBuffer 函式或 Ixud_GetAIBufferH 函式取出類比輸入資料，欲停止類比輸入採集功能時，需呼叫 Ixud_StopAI 函式。




➢ **語法**

```
WORD Ixud_StartExtAI(
        WORD wBoardNo,
        WORD wActive,
        WORD wChannel,
        WORD wConfig,
        float fSamplingRate,
        DWORD dwPostDataCount
        DWORD dwPerDataCount
);
```

➢ **參數**

*wBoardNo*

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

*wActive*

[Input] 設定外部觸發條件。

| *dwActive* | PCI-822LU<br>PCI-826LU | PCI-2602U |
|---|---|---|
| 0 | 下降緣觸發 | 任意 |
| 1 | 上昇緣觸發 | 任意 |

*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。

*fSamplingRate*

[Input] 爲一浮點數值，由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwPostDataCount*

[Input] 由使用者設定外部觸發後需要的資料筆數。

*dwPreDataCount*

[Input] 由使用者設定外部觸發前需要的資料筆數。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StartExtAnalogTrigger

透過此函式來初始化 ADC、設定取樣頻率、採樣資料量，並開始啓動外部觸發(類比信號)類比資料擷取，啓動後將以等速度擷取單一通道的類比資料並將資料暫存至系統記憶體內，可透過 Ixud_GetAIBuffer 函式或 Ixud_GetAIBufferH 函式取出類比輸入資料，欲停止類比輸入採集功能時，需呼叫 Ixud_StopAI 函式。

本函式僅支援 PCI-2602U

爲了確保資料採集的準確度，使用此函式在資料採集的過程中時將會佔用 CPU 一段短暫的時間至採集數據完成，此時間將取決於使用者設定的採集資料量及取樣頻率。

➢ **語法**

```
WORD Ixud_StartExtAnalogTrigger(
        WORD wBoardNo,
        WORD wActive,
        WORD wChannel,
        WORD wConfig,
        float fSamplingRate,
        DWORD dwReserved,
        float fAboveTrgVoltage,
        float fBelowTrgVoltage
);
```

➢ **參數**

*wBoardNo*

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

*wActive*

[Input] 設定外部觸發條件。

| *dwActive* | PCI-2602U |
|---|---|
| IXUD_ANALOGTRIGGER_ABOVE | Above High |
| IXUD_ANALOGTRIGGER_BELOW | Below Low |

| | |
|---|---|
| IXUD_ANALOGTRIGGER_LEAVE | Leave Region |
| IXUD_ANALOGTRIGGER_ENTRY | Entry Region |

*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。

*fSamplingRate*

[Input] 爲一浮點數值，由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwDataCount*

[Input] 由使用者設定外部觸發後需要的資料筆數。

*dwReserved*

[Input] 保留。

*fAboveTrgVoltage*

[Input] 設定 Above trigger 電壓範圍。

*fBelowTrgVoltage*

[Input] 設定 Below trigger 電壓範圍。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StartExtAIScan

透過此函式來初始化 ADC、設定取樣頻率、採樣資料量，並開始啓動外部觸發(類比信號)類比資料擷取，啓動後將以等速度擷取多個通道的類比資料並將資料暫存至系統記憶體內，可透過 Ixud_GetAIBuffer 函式或 Ixud_GetAIBufferH 函式取出類比輸入資料，欲停止類比輸入採集功能時，需呼叫 Ixud_StopAI 函式。

爲了確保資料採集的準確度，使用此函式在資料採集的過程中時將會佔用 CPU 一段短暫的時間至採集數據完成，此時間將取決於使用者設定的採集資料量及取樣頻率。

➢ **語法**

```
WORD Ixud_StartExtAIScan(
        WORD wBoardNo,
        WORD wActive,
        WORD wChannels,
        WORD wChannelList[ ],
        WORD wConfigList[ ],
        float fSamplingRate,
        DWORD dwDataCountPostChannel,
        DWORD dwDataCountPerChannel
);
```

➢ **參數**

*wBoardNo*

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

*wActive*

[Input] 設定外部觸發條件。

| *dwActive* | PCI-822LU<br>PCI-826LU | PCI-2602U |
|---|---|---|
| 0 | 下降緣觸發 | 任意 |
| 1 | 上昇緣觸發 | 任意 |

*wChannels*

[Input] 由使用者設定將擷取 wChannels 個通道擷取資料的配置值。

*wChannelList[ ]*

[Input] 一個 WORD 陣列(陣列大小等於 wChannles)。由使用者設定欲使用來做擷取資料的類比輸入通道。請從陣列第 0 個元素開始設定第一個類比輸入通道號碼。

*wConfigList[ ]*

[Input] 請宣告一個 WORD 陣列(陣列大小至少大於 wChannels)。由使用者設定每個通道的配置值，從陣列第 0 個元素開始設定第一個類比輸入通道的配置碼。

*fSamplingRate*

[Input] 爲一浮點數值，由使用者設定類比輸入通道的取樣頻率(次/秒)。

*dwDataPostCountPerChannel*

[Input] 由使用者設定每一個通道外部觸發後需要的資料筆數。

*dwDataPreCountPerChannel*

[Input] 由使用者設定每一個通道外部觸發前需要的資料筆數。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_GetAIBuffer

此函式將會從記憶體裡取得類比輸入值，並儲存在使用者所指令的陣列裡，其值爲浮點數。呼叫此函式前需先呼叫 Ixud_StartAI、Ixud_StartAIScan、Ixud_StartExtAI 及 Ixud_StartExtAIScan 函式。

➢ **語法**

**WORD Ixud_GetAIBuffer(**
**WORD** wBoardNo**,**
**DWORD** dwDataCount**,**
**float** fValue[ ]
**);**

➢ **參數**

*wBoardNo*

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

*dwDataCount*

[Input] 由使用者設定需要的資料筆數。

*fValue[ ]*

[Output] 請宣告一個浮點數陣列。用來取得從類比輸入資料值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_GetAIBufferH

此函式將會從記憶體裡取得類比輸入值，並儲存在使用者所指令的陣列裡，其值爲十六進位值。呼叫此函式前需先呼叫 Ixud_StartAI、Ixud_StartAIScan、Ixud_StartExtAI 及 Ixud_StartExtAIScan 函式。

➢ **語法**

```
WORD Ixud_GetAIBufferH(
        WORD wBoardNo,
        DWORD dwDataCount,
        DWORD hValue[ ]
);
```

➢ **參數**

*wBoardNo*

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

*dwDataCount*

[Input] 由使用者設定需要的資料筆數。

*hValue[ ]*

[Output] 請宣告一個 DWORD 陣列。用來取得從類比輸入值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StopAI

呼叫此函式，用來停止類比輸入擷取的運作。

- **語法**

```
WORD Ixud_StopAI(
        WORD wBoardNo
);
```

- **參數**

*wBoardNo*

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

- **回傳值**

請參考 A.1. 函數回傳值定義。

## 5.2.5. 類比輸出函式集

# *Ixud_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 Ixud_ConfigAO(
        WORD wBoardNo,
        WORD wChannel,
        WORD wCfgCode
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*wCfgCode*

[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)及 A.3.3. 類比輸出配置碼(電流)。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。

# Ixud_WriteAOVoltage

呼叫此函式控制類比輸出通道輸出一固定的電壓，此電壓設定值爲浮點數值。

- **語法**

**WORD Ixud_WriteAOVoltage(**
**WORD** wBoardNo,
**WORD** wChannel,
**float** fValue
**);**

- **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*fValue*

[Input] 由使用者設定類比輸出電壓值，此值爲浮點數值，單位爲伏特(V)。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WriteAOVoltageH

呼叫此函式控制類比輸出通道輸出一固定的電壓，此電壓設定值爲十六進位值。

- **語法**

**WORD Ixud_WriteAOVoltageH(**
**WORD** wBoardNo,
**WORD** wChannel,

**DWORD** hValue

**);**

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*hValue*

[Input] 由使用者設定類比輸出電壓值，此值爲十六進制值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_WriteAOCurrent*

呼叫此函式控制類比輸出通道輸出一固定的電流，此電流設定值爲浮點數值。

➢ **語法**

**WORD Ixud_WriteAOCurrent(**

**WORD** wBoardNo**,**

**WORD** wChannel**,**

**float** fValue

**);**

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*fValue*

[Input] 由使用者設定類比輸出電流值，此值爲浮點數值，單位爲 mA。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_WriteAOCurrentH*

呼叫此函式控制類比輸出通道輸出一固定的電流，此電流設定值爲十六進位值。

➢ **語法**

```
WORD Ixud_WriteAOCurrentH(
        WORD wBoardNo,
        WORD wChannel,
        DWORD hValue
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*hValue*

[Input] 由使用者設定類比輸出電流值，此值爲十六進制值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_StartAOVoltage

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

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_StartAOVoltage(
        WORD wBoardNo,
        WORD wChannel,
        DWORD wCfgCode,
        float fFrequency,
        DWORD dwDataCount,
        DWORD dwCycleNum,
        float fAOBuf[ ]
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*wCfgCode*

[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)及 A.3.3. 類比輸出配置碼(電流)。此配置碼將會影響類比輸出的精度及輸出範圍。

*fFrequencye*

[Input] 爲一浮點數值，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。使用函式停止採集。

*dwCycleNum*

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

*fAOBuf[]*

[Input] 儲存類比輸出資料(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。

# *Ixud_StartAOVoltageH*

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

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_StartAOVoltage(
        WORD wBoardNo,
        WORD wChannel,
        DWORD wCfgCode,
        float fFrequency,
        DWORD dwDataCount,
        DWORD dwCycleNum,
        DWORD dwAOBuf[ ]
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

*wCfgCode*

[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)及 A.3.3. 類比輸出配置碼(電流)。此配置碼將會影響類比輸出的精度及輸出範圍。

*fFrequencye*

[Input] 爲一浮點數值，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。使用函式停止採集。

*dwCycleNum*

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

*dwAOBuf[]*

[Input] 儲存類比輸出資料(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。

# Ixud_StopAO

呼叫此函式，用來停止高速類比輸出的運作。

本函式僅支援 PCI-2602U

➢ **語法**

```
WORD Ixud_StopDO(
        WORD wBoardNo,
        WORD wChannel
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定類比輸出通道的號碼。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

## 5.2.6. 計時計數函式集

# *Ixud_DisableCounter*

關閉(停止)計數器通道。

- **語法**

**WORD Ixud_DisableCounter(**
**WORD** wBoardNo**,**
**WORD** wChannel
**);**

- **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定計數器通道的編號，第一個通道 wChannel 爲 0，第二個通道 wChannel 爲 1，依此類推。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_ReadCounter*

呼叫此函式可讀取計數器通道的計數值。

- **語法**

**WORD Ixud_ReadCounter(**
**WORD** wBoardNo**,**
**WORD** wChannel**,**
**DWORD ***dwValue

);

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定計數器通道的編號，第一個通道 wChannel 爲 0，第二個通道 wChannel 爲 1，依此類推。

*dwValue*

[Output] 取得計數器的計數值。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_ReadFrequency*

讀取信號源的頻率。

本函式僅支援 PCI-FC16U

➢ **語法**

```
WORD Ixud_ReadFrequency(
        WORD wBoardNo,
        WORD wChannel,
        float *fFrequency,
        DWORD dwTimeOutMs,
        WORD *wStatus
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定計數器通道的編號，第一個通道 wChannel 爲 0，第二個通道 wChannel 爲 1，依此類推。

*fFrequency*

[Output] 儲存從信號源讀取出來的頻率，單位爲 Hz。

*wTimeOutMs*

[Input] 由使用者設定計數器的逾時時間，單位爲 ms。

*wStatusy*

[Output] 取得計數器通道的狀態。

| *wStatus* | 狀態描述 |
|---|---|
| 0 | 取得頻率中 |
| 1 | 逾時 |
| 2 | 閂鎖住頻率 |

表格 5-13 wStatus 參數設定

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_SetCounter*

設定計數器通道的計數值及計數模式。

➢ **語法**

**WORD Ixud_SetCounter(**
**WORD** wBoardNo,
**WORD** wChannel,
**WORD** wMode,
**DWORD** dwValue

);

- **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定計數器通道的編號，第一個通道 wChannel 爲 0，第二個通道 wChannel 爲 1，依此類推。

*wMode*

[Input] 由使用者設定計數器通道的計數模式。計數模式的詳細說明請參考 Intel 8254 Datasheet。

| *wMode* | 模式名稱 |
|---|---|
| 0 | 數完時中斷 |
| 1 | 可程式單擊閘觸發 |
| 2 | 比率產生器 |
| 3 | 方波產生器 |
| 4 | 軟體觸發擷取 |
| 5 | 硬體觸發擷取 |

表格 5-14 wMode 參數設定

*dwValue*

[Input] 由使用者設定計數器通道的計數值。

- **回傳值**

請參考 A.1. 函數回傳值定義。

# *Ixud_SetFCChannelMode*

設定每個通道的計數模式。

本函式僅支援 PCI-FC16U

- **語法**

**WORD Ixud_SetFCChannelMode(**

```
    WORD wBoardNo,
    WORD wChannel,
    WORD wMode,
    WORD wDelayMs
);
```

➢ **參數**

*wBoardNo*

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

*wChannel*

[Input] 由使用者設定計數器通道的編號，第一個通道 wChannel 爲 0，第二個通道 wChannel 爲 1，依此類推。

*wMode*

[Input] 由使用者設定計數器通道的計數模式。

| *wMode* | 模式名稱 |
|---|---|
| 0 | - |
| 1 | - |
| 2 | 倒數模式(down count) |
| 3 | - |
| 4 | - |
| 5 | - |

表格 5-15 wMode 參數設定

*wDelayMs*

[Input] 由使用者設定計數器的延遲時間，此時間的長短取決於使用者欲量的頻率的範圍，單位爲 ms。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# 5.2.7. 記憶體輸出輸入函式集

## *Ixud_ReadMemory*

從使用者所指定的記憶體位址裡讀取出 8/16/32 bit 的數值。

➢ **語法**

```
WORD Ixud_ReadMemory(
        WORD wBoardNo,
        DWORD dwOffsetByte,
        WORD wSize,
        DWORD *dwValue
);
```

➢ **參數**

*wBoardNo*

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

*dwOffsetByte*

[Input] 由使用者指定欲讀取記憶體的位移位址。

*wSize*

[Input] 由使用者設定讀取資料的長度。

| *wSize* | 長度 |
|---|---|
| 8 | 8-bit |
| 16 | 16-bit |
| 32 | 32-bit |

表格 5-16 wSize 參數設定

*dwValue*

[Output]讀取記憶體內的資料。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WriteMemory

寫入 8/16/32 bit 的數值至使用者所指定的記憶體位址裡。

➢ **語法**

```
WORD Ixud_WriteMemory(
        WORD wBoardNo,
        DWORD dwOffsetByte,
        WORD wSize,
        DWORD dwValue
);
```

➢ **參數**

*wBoardNo*

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

*dwOffsetByte*

[Input] 由使用者指定欲寫入記憶體的位移位址。

*wSize*

[Input] 由使用者設定寫入資料的長度。

| *wSize* | 長度 |
|---|---|
| 8 | 8-bit |
| 16 | 16-bit |
| 32 | 32-bit |

表格 5-17 wSize 參數設定

*dwValue*

[Input] 寫入資料至記憶體內。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_ReadMemory32

從使用者所指定的記憶體位址裡讀取出 32bit 值，此函式支援不帶有無號整數型態的編輯器，如 Visual Basic 6.0。

➢ **語法**

```
WORD Ixud_ReadMemory32(
        WORD wBoardNo,
        DWORD dwOffsetByte,
        WORD *dwLow,
        DWORD *dwHigh
);
```

➢ **參數**

*wBoardNo*

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

*dwOffset*

[Input] 設定讀取記憶體的位移位址。

*dwLow*

[Output] 讀取記憶體內 Bit 0~15 的資料。

*dwHigh*

[Output] 讀取記憶體內 Bit 16~31 的資料。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# Ixud_WriteMemory32

從記憶體位址裡寫入一個 32bit 的值，此函式支援不帶有無號整數型態的編輯器，如 Visual Basic 6.0。

➢ **語法**

```
WORD Ixud_WriteMemory32(
        WORD wBoardNo,
        DWORD dwOffsetByte,
        WORD dwLow,
        DWORD dwHigh
);
```

➢ **參數**

*wBoardNo*

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

*dwOffsetByte*

[Input] 設定寫入記憶體的位移位址。

*dwLow*

[Input] 設定寫入記憶體的 bit 0~15 的資料。

*dwHigh*

[Input] 設定寫入記憶體的 bit16~31 的資料。

➢ **回傳值**

請參考 A.1. 函數回傳值定義。

# 5.3. 資料型態

## *PIXUD_DEVICE_INFO*

- 語法

```
typedef struct _IXUD_DEVICE_INFO_
{
    DWORD dwSize;
    WORD wVendorID;
    WORD wDeviceID;
    WORD wSubVendorID;
    WORD wSubDeviceID;
    DWORD dwBAR[6];
    UCHAR BusNo;
    UCHAR DevNo;
    UCHAR IRQ;
    UCHAR Aux;
    DWORD dwBarVirtualAddress [6];
}IXUD_DEVICE_INFO,*PIXUD_DEVICE_INFO;
```

- 成員

*dwSize*

[Output] 取得此資料結構的大小，單位爲 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。

*AuxNo*

[Output] 取得板卡的 Aux 號碼。

*dwBarVirtualAddress []*

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

| Virual 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] |

# PIXUD_CARD_INFO

➢ 語法

```
typedef struct _IXUD_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];
}IXUD_CARD_INFO,*PIXUD_CARD_INFO;
```

➢ 成員

*dwSize*

[Output] 取得此資料結構的大小，單位爲 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，若取得數值爲 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，設定值請參考下面表格。

| 數值 | 數值(Hex) | 接線設定 |
|---|---|---|
| 1 | 1 | 單端式(SE) |
| 2 | 2 | 差動式(DIFF) |
| 255 | FF | 未支援此設定 |

*wAIOResolution*

[Output] 取得板卡類比輸出入的解析度，高位元爲類比輸入通道解析度 ((wAIOResolution>>8)&0xFF)，低位元爲類比輸出通道解析度 (wAIOResolution&0xFF)。

| 數值 | 數值(Hex) | 解析度 |
|---|---|---|
| 12 | C | 12-bit |
| 14 | E | 14-bit |
| 16 | 10 | 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。

| 數值 | 寬度 |
|---|---|
| 8 | 8-bit |
| 16 | 16-bit |
| 32 | 32-bit |

*wCounterChannels*

[Output] 取得板卡計時計數的通道數量。

*wCounterChannels*

[Output] 取得板卡的內建記憶體大小，單位爲 kByte。

*dwReserved1[]*

[Output] 取得保留資訊。


# 附錄 A. 函式回傳值與配置碼

本章節列出了泓格 UniDAQ 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 | Ixud_NoErr | 正確 |
| 1 | Ixud_OpenDriverErr | 開啓驅動程式發生錯誤 |
| 2 | Ixud_PnPDriverErr | Plug&Play 時發生錯誤 |
| 3 | Ixud_DriverNoOpen | 驅動程式未開啓 |
| 4 | Ixud_GetDriverVersionErr | 取得驅動程式版本錯誤 |
| 5 | Ixud_ExceedBoardNumber | 板卡號碼錯誤 |
| 6 | Ixud_FindBoardErr | 找不到任何的板卡 |
| 7 | Ixud_BoardMappingErr | 板卡物件索引(Board Mapping)錯誤 |
| 8 | Ixud_DIOModesErr | 數位輸出輸入模式設定錯誤 |
| 9 | Ixud_InvalidAddress | 不合法的位址 |
| 10 | Ixud_InvalidSize | 不合法的大小 |
| 11 | Ixud_InvalidPortNumber | 不合法的埠號 |
| 12 | Ixud_UnSupportedModel | 未支援此板卡 |
| 13 | Ixud_UnSupportedFun | 未支援此函式 |
| 14 | Ixud_InvalidChannelNumber | 不合法的通道號碼 |
| 15 | Ixud_InvalidValue | 不合法的值 |
| 16 | Ixud_InvalidMode | 不合法的模式 |
| 17 | Ixud_GetAIStatusTimeOut | 取得類比輸入狀態逾時 |
| 18 | Ixud_TimeOutErr | 超過時間發生異常 |
| 19 | Ixud_CfgCodeIndexErr | 找不到適合的配置碼表格索引 |
| 20 | Ixud_ADCCTLTimeoutErr | ADC 控制器愈期 |
| 21 | Ixud_FindPCIIndexErr | 找不到適合的 PCI 表格索引值 |
| 22 | Ixud_InvalidSetting | 不合法的設定值 |
| 23 | Ixud_AllocateMemErr | 分配記憶體空間失敗 |
| 24 | Ixud_InstallEventErr | 安裝中斷事件失敗 |
| 25 | Ixud_InstallIrqErr | 安裝中斷失敗 |
| 26 | Ixud_RemoveIrqErr | 移除中斷失敗 |
| 27 | Ixud_ClearIntCountErr | 清除中斷計數量失敗 |
| 28 | Ixud_GetSysBufferErr | 取得系統緩衝區失敗 |
| 29 | Ixud_CreateEventErr | CreateEvent 失敗 |
| 30 | Ixud_UnSupportedResolution | 未支援此解析度 |
| 31 | Ixud_CreateThreadErr | CreateThread 失敗 |

| | | |
|---|---|---|
| 32 | Ixud_ThreadTimeOutErr | 執行緒逾時 |
| 33 | Ixud_FIFOOverFlowErr | FIFO 溢滿 |
| 34 | Ixud_FIFOTimeOutErr | FIFO 逾時 |
| 35 | Ixud_GetIntInstStatus | 取得中斷安裝狀態 |
| 36 | Ixud_GetBufStatus | 取得 SYS 緩衝區狀態 |
| 37 | Ixud_SetBufCountErr | 設定緩衝區大小錯誤 |
| 38 | Ixud_SetBufInfoErr | 設定緩衝區資料錯誤 |
| 39 | Ixud_FindCardIDErr | 找不到卡片識別碼 |
| 40 | Ixud_EventThreadErr | 事件執行緒錯誤 |
| 41 | Ixud_AutoCreateEventErr | 自動創建事件失敗 |
| 42 | Ixud_RegThreadErr | 註冊執行緒失敗 |
| 43 | Ixud_SearchEventErr | 尋找事件失敗 |
| 44 | Ixud_FifoResetErr | FIFO 清除失敗 |
| 45 | Ixud_InvalidBlock | 不合法的 EEPROM 區塊 |
| 46 | Ixud_InvalidAddr | 不合法的 EEPROM 位址 |
| 47 | Ixud_AcqireSpinLock | 獲得旋轉鎖失敗 |
| 48 | Ixud_ReleaseSpinLock | 釋放旋轉鎖失敗 |
| 49 | Ixud_SetControlErr | 類比輸入設定錯誤 |
| 50 | Ixud_InvalidChannels | 不合法的通道數 |
| 51 | Ixud_SearchCardErr | 不合法的板卡號碼 |
| 52 | Ixud_SetMapAddressErr | 設定映像位址失敗 |
| 53 | Ixud_ReleaseMapAddressErr | 釋放映像位址失敗 |
| 54 | Ixud_InvalidOffset | 不合法的位移 |
| 55 | Ixud_ShareHandleErr | 開啓 Share Memory 失敗 |
| 56 | Ixud_InvalidDataCount | 不合法的資料量 |
| 57 | Ixud_WriteEEPErr | 寫入 EEPROM 失敗 |
| 58 | Ixud_CardIOErr | 使用 CardIO 失敗 |
| 59 | Ixud_IOErr | 使用 MemoryIO 失敗 |
| 60 | Ixud_SetScanChannelErr | 設置 channel scan number 失敗 |
| 61 | Ixud_SetScanConfigErr | 設置 channel scan configuration 失敗 |
| 62 | Ixud_GetMMIOMapStatus | 取得 Memory Mapping IO 狀態失敗 |

# 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 |
|---|---|---|
| PIOD56 | 800140 | PIO-D24/D56/D24U/D56U |
| PEXD56 | 800140 | PEX-D24/D56 |
| PIOD48 | 800130 | PIO-D48/D48U/D48SU |
| PEXD48 | 800130 | PEX-D48 |
| PIOD64 | 800120 | PIO-D64/D64U |
| PIOD96 | 800110 | PIO-D96/D96SU |
| PIOD144 | 800100 | PIO-D144 |
| PIOD168 | 800150 | PIO-D168 |
| PIODA | 800400 | PIO-DA4/DA8/DA16/DA4U/DA8U/DA16U/PISO-DA4U/DA8U/DA16U |
| PEXDA | 800400 | PEX-DA4/DA8/DA16 |
| PIO821 | 800310 | PIO-821 L/H/LU/HU |
| PISOP16R16U | 1800FF | PISO-P16R16U/P16R16E |
| PEXP16R16 | 1800FF | PEX-P16R16i |
| PEXP8R8 | 1800FF | PEX-P8R8i |
| PISOC64 | 800800 | PISO-C64 |
| PEXC64 | 800800 | PEX-C64 |
| PISOP64 | 800810 | PISO-P64 |
| PEXP64 | 800810 | PEX-P64 |
| PISOA64 | 800850 | PISO-A64 |
| PISOP32C32 | 800820 | PISO-P32C32/P32C32U/P32S32WU |
| PEXP32C32 | 800820 | PEX-P32C32 |
| PISO1730 | 800820 | PISO-1730U |
| PISOP32A32 | 800870 | PISO-P32A32 |
| PISOP8R8 | 800830 | PISO-P8R8/PISO-P8R8AC/PISO-P8R8DC |
| PISO730 | 800840 | PISO-730 |
| PISO730A | 800880 | PISO-730A |
| PISO725 | 8008FF | PISO-725 |
| PISODA2 | 800B00 | PISO-DA2 |
| PISO813 | 800A00 | PISO-813/813U |
| PCITMC12 | DF2962 | PCI-TMC12/PCI-TMC12A |
| PCIM512 | DE9562 | PCI-M512 |
| PCIM256 | DE92A6 | PCI-M256 |
| PCIM128 | DE9178 | PCI-M128 |
| PCIFC16 | B13017 | PCI-FC16U |
| PCID64 | DE3513 | PCI-D64 |
| PCI822 | DE3823 | PCI-822 LU |
| PCI826 | DE3827 | PCI-826 LU |
| PCI2602 | 2CB656 | PCI-2602U |

| | | |
|---|---|---|
| **PCI100X** | 341002 | PCI-1002 LU/HU |
| **PEX100X** | 341002 | PEX-1002 |
| **PCI1202** | 345672 | PCI-1202 L/H ,PCI-1202U L/H |
| **PEX1202** | 345672 | PEX-1202 L/H |
| **PCI1602** | 345676 | PCI-1602/1602U,PCI-1602 F |
| **PCI180X** | 345678 | PCI-1800 L/H, PCI-1802 L/H |
| **PCIP8R8** | D6102B | PCI-P8R8 |
| **PEXP8POR8** | D6102B | PEX-P8POR8i |
| **PCIP16R16** | D61E39 | PCI-P16R16/P16C16/P16POR16 |
| **PEXP16POR16** | D61E39 | PEX-P16POR16i |

# 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，更詳細的資訊可以參考板卡的硬體手冊或參考本節**泓格板卡支援類比輸入配置碼**表格。

| 配置碼 | 定義 | 極性 | 電壓範圍 |
|---|---|---|---|
| 0 | IXUD_BI_10V | Bipolar | +/- 10V |
| 1 | IXUD_BI_5V | Bipolar | +/- 5V |
| 2 | IXUD_BI_2V5 | Bipolar | +/- 2.5V |
| 3 | IXUD_BI_1V25 | Bipolar | +/- 1.25V |
| 4 | IXUD_BI_0V625 | Bipolar | +/- 0.625V |
| 5 | IXUD_BI_0V3125 | Bipolar | +/- 0.3125V |
| 6 | IXUD_BI_0V5 | Bipolar | +/- 0.5V |
| 7 | IXUD_BI_0V05 | Bipolar | +/- 0.05V |
| 8 | IXUD_BI_0V005 | Bipolar | +/- 0.005 |
| 9 | IXUD_BI_1V | Bipolar | +/- 1V |
| 10 | IXUD_BI_0V1 | Bipolar | +/- 0.1V |
| 11 | IXUD_BI_0V01 | Bipolar | +/- 0.01V |
| 12 | IXUD_BI_0V001 | Bipolar | +/- 0.001V |
| 13 | IXUD_UNI_20V | Unipolar | 0 ~ 20V |
| 14 | IXUD_UNI_10V | Unipolar | 0 ~ 10V |
| 15 | IXUD_UNI_5V | Unipolar | 0 ~ 5V |
| 16 | IXUD_UNI_2V5 | Unipolar | 0 ~ 2.5V |
| 17 | IXUD_UNI_1V25 | Unipolar | 0 ~ 1.25V |
| 18 | IXUD_UNI_0V625 | Unipolar | 0 ~ 0.625V |
| 19 | IXUD_UNI_1V | Unipolar | 0 ~ 1V |
| 20 | IXUD_UNI_0V1 | Unipolar | 0 ~ 0.1V |
| 21 | IXUD_UNI_0V01 | Unipolar | 0 ~ 0.01V |
| 22 | IXUD_UNI_0V001 | Unipolar | 0 ~ 0.001V |
| 23 | IXUD_BI_20V | Bipolar | +/- 20V |

泓格板卡支援類比輸入配置碼表

| 電壓範圍 | PIO-821L<br>PIO-821LU | PIO-821H<br>PIO-821HU | PISO-813<br>PIO-813U<br>(JP1=10V) | PISO-813<br>PIO-813U<br>(JP1=20V) | PCI-1002LU<br>PEX-1002L | PCI-1002HU<br>PEX-1002H |
|---|---|---|---|---|---|---|
| +/- 10V | | | | ✔ | ✔ | ✔ |
| +/- 5V | ✔ | ✔ | ✔ | ✔ | ✔ | |
| +/- 2.5V | ✔ | | ✔ | ✔ | ✔ | |
| +/- 1.25V | ✔ | | ✔ | ✔ | ✔ | |
| +/- 0.625V | ✔ | | ✔ | ✔ | | |
| +/- 0.3125V | | | | | | |
| +/- 0.5V | | ✔ | | | | |
| +/- 0.05V | | ✔ | | | | |
| +/- 0.005 | | ✔ | | | | |
| +/- 1V | | | | | | ✔ |
| +/- 0.1V | | | | | | ✔ |
| +/- 0.01V | | | | | | ✔ |
| +/- 0.001V | | | | | | |
| 0 ~ 20V | | | | | | |
| 0 ~ 10V | | | ✔ | | | |
| 0 ~ 5V | | | ✔ | | | |
| 0 ~ 2.5V | | | ✔ | | | |
| 0 ~ 1.25V | | | ✔ | | | |
| 0 ~ 0.625V | | | ✔ | | | |
| 0 ~ 1V | | | | | | |
| 0 ~ 0.1V | | | | | | |
| 0 ~ 0.01V | | | | | | |
| 0 ~ 0.001V | | | | | | |

泓格板卡支援類比輸入配置碼表

| 電壓範圍 | PCI-1202LU<br>PCI-1800LU<br>PCI-1802LU<br>PEX-1202L | PCI-1202HU<br>PCI-1800HU<br>PCI-1802HU<br>PEX-1202H | PCI-1602<br>PCI-1602U<br>PCI-1602F<br>PCI-1602FU | PCI-822LU<br>PCI-826LU | PCI-2602U |
|---|---|---|---|---|---|
| +/- 10V | ✔ | ✔ | ✔ | ✔ | ✔ |
| +/- 5V | ✔ | ✔ | ✔ | ✔ | ✔ |
| +/- 2.5V | ✔ | | ✔ | ✔ | ✔ |
| +/- 1.25V | ✔ | | ✔ | ✔ | ✔ |
| +/- 0.625V | ✔ | | | | ✔ |
| +/- 0.3125V | | | | | |
| +/- 0.5V | | ✔ | | | |
| +/- 0.05V | | ✔ | | | |
| +/- 0.005 | | ✔ | | | |
| +/- 1V | | ✔ | | | |
| +/- 0.1V | | ✔ | | | |
| +/- 0.01V | | ✔ | | | |
| +/- 0.001V | | | | | |
| 0 ~ 20V | | | | | |
| 0 ~ 10V | ✔ | ✔ | | | |
| 0 ~ 5V | ✔ | | | | |
| 0 ~ 2.5V | ✔ | | | | |
| 0 ~ 1.25V | ✔ | | | | |
| 0 ~ 0.625V | | | | | |
| 0 ~ 1V | | ✔ | | | |
| 0 ~ 0.1V | | ✔ | | | |
| 0 ~ 0.01V | | ✔ | | | |
| 0 ~ 0.001V | | | | | |

PCI-2602U 類比輸入配置電壓表

| 電壓設定值 | 實際電壓 |
|---|---|
| +/- 10V | +/- 10.24V |
| +/- 5V | +/- 5.12V |
| +/- 2.5V | +/- 2.56V |
| +/- 1.25V | +/- 1.28V |
| +/- 0.625V | +/- 0.64V |

# 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，更詳細的資訊可以參考板卡的硬體手冊或參考本節**泓格板卡支援類比輸出配置碼**表格。

| 配置碼 | 定義 | 電壓範圍 |
|---|---|---|
| 0 | IXUD_AO_UNI_5V | 0 ~ 5V |
| 1 | IXUD_AO_BI_5V | +/- 5V |
| 2 | IXUD_AO_UNI_10V | 0 ~ 10V |
| 3 | IXUD_AO_BI_10V | +/- 10V |
| 4 | IXUD_AO_UNI_20V | 0 ~ 20V |
| 5 | IXUD_AO_BI_20V | +/- 20V |

泓格板卡支援類比輸出配置碼表(電壓)

| 配置碼 | 電壓範圍 | PIO-DA4U<br>PIO-DA8U<br>PIO-DA16U | PISO-DA4U<br>PISO-DA8U<br>PISO-DA16U | PIO-821L<br>PIO-821H<br>PIO-821LU<br>PIO-821HU | PISO-DA2U | CI-1202<br>PCI-1602<br>PCI-1800<br>PCI-1802<br>PEX-1202 | PCI-822<br>PCI-826<br>PCI-2602U |
|---|---|---|---|---|---|---|---|
| 0 | 0 ~ 5V | - | - | ✔ | ✔ | - | ✔ |
| 1 | +/- 5V | - | - | ✔ | ✔ | ✔ | ✔ |
| 2 | 0 ~ 10V | - | - | - | ✔ | - | ✔ |
| 3 | +/- 10V | ✔ | ✔ | - | ✔ | ✔ | ✔ |

# A.3.3. 類比輸出配置碼(電流)

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流範圍及極性，每張板卡支援的類比輸出電流範圍及極性皆有所不同，更詳細的資訊可以參考板卡的硬體手冊或參考本節**泓格板卡支援類比輸出配置碼(電流)**表格。

| 配置碼 | 定義 | 電流範圍 |
|---|---|---|
| **16** | IXUD_AO_I_0_20_MA | 0 ~ 20 mA |
| **17** | IXUD_AO_I_4_20_MA | 4 ~ 20 mA |

泓格板卡支援類比輸出配置碼表(電流)

| 配置碼 | 電流範圍 | PIO-DA4U<br>PIO-DA8U<br>PIO-DA16U | PISO-DA4U<br>PISO-DA8U<br>PISO-DA16U | PEX-DA4<br>PEX-DA8<br>PEX-DA16 | PISO-DA2U |
|---|---|---|---|---|---|
| **16** | 0 ~ 20 | ✔ | ✔ | ✔ | ✔ |
| **17** | 4~20 | - | - | | ✔ |

# A.3.4. 中斷事件配置碼

下表配置碼定義中斷事件

| 配置碼 | 定義 | 敘述 |
|---|---|---|
| **1** | IXUD_HARDWARE_INT | 硬體中斷 |
| **2** | IXUD_APC_READY_INT | 類比資料準備完成產生中斷 |
| **4** | IXUD_ACTIVE_LOW | 下降緣觸發產生事件 |
| **8** | IXUD_ACTIVE_HIGH | 上昇緣觸發產生事件 |

# A.4. 數位輸入埠定義號碼

| DI 埠號 | PIO-D24U<br>PEX-D24 | PIO-D56U<br>PEX-D56 | PIO-D48U<br>PIO-D48SU<br>PEX-D48 | PIO-D64U | PIO-D96U<br>PIO-D96SU | PIO-D144U | PIO-D168U | PISO-P64<br>PISO-P64U<br>PEX-P64 |
|---|---|---|---|---|---|---|---|---|
| 0 | CN3 Port0 | CN3 Port0 | CN1 Port0 | CN2 DI 0 ~ 7 | CN1 Port0 | CN1 Port0 | CN1 Port0 | IDI 0 ~ 7 |
| 1 | CN3 Port1 | CN3 Port1 | CN1 Port1 | CN2 DI 8 ~ 15 | CN1 Port1 | CN1 Port1 | CN1 Port1 | IDI 8 ~ 15 |
| 2 | CN3 Port2 | CN3 Port2 | CN1 Port2 | CN4 DI 0 ~ 7 | CN1 Port2 | CN1 Port2 | CN1 Port2 | IDI 16 ~ 23 |
| 3 | - | CN2 DI 0 ~ 7 | CN2 Port3 | CN4 DI 8 ~ 15 | CN2 Port3 | CN2 Port3 | CN2 Port3 | IDI 24 ~ 31 |
| 4 | - | CN2 DI 8 ~ 15 | CN2 Port4 | - | CN2 Port4 | CN2 Port4 | CN2 Port4 | IDI 32 ~ 39 |
| 5 | - | - | CN2 Port5 | - | CN2 Port5 | CN2 Port5 | CN2 Port5 | IDI 40 ~ 47 |
| 6 | - | - | - | - | CN3 Port6 | CN3 Port6 | CN3 Port6 | IDI 48 ~ 55 |
| 7 | - | - | - | - | CN3 Port7 | CN3 Port7 | CN3 Port7 | IDI 56 ~ 63 |
| 8 | - | - | - | - | CN3 Port8 | CN3 Port8 | CN3 Port8 | |
| 9 | - | - | - | - | CN4 Port9 | CN4 Port9 | CN4 Port9 | |
| 10 | - | - | - | - | CN4 Port10 | CN4 Port10 | CN4 Port10 | |
| 11 | - | - | - | - | CN4 Port11 | CN4 Port11 | CN4 Port11 | |
| 12 | - | - | - | - | - | CN5 Port12 | CN5 Port12 | |
| 13 | - | - | - | - | - | CN5 Port13 | CN5 Port13 | |
| 14 | - | - | - | - | - | CN5 Port14 | CN5 Port14 | |
| 15 | - | - | - | - | - | CN6 Port15 | CN6 Port15 | |
| 16 | - | - | - | - | - | CN6 Port16 | CN6 Port16 | |
| 17 | - | - | - | - | - | CN6 Port17 | CN6 Port17 | |
| 18 | - | - | - | - | - | - | CN6 Port18 | |
| 19 | - | - | - | - | - | - | CN6 Port19 | |
| 20 | - | - | - | - | - | - | CN6 Port20 | |

| DI 埠號 | PISO-P32A32U<br>PISO-P32C32U<br>PISO-P32S32WU<br>PISO-1730U<br>PEX-P32C32 | PISO-P16R16U<br>PEX-P16R16i | PISO-P8R8U<br>PEX-P8R8i<br>PISO-725 | PISO-730<br>PISO-730U<br>PISO-730A<br>PISO-730AU<br>PEX-730 | PCI-P8R8<br>PEX-P8POR8i | PCI-P16R16<br>PCI-P16C16<br>PEX-P16POR16i |
|---|---|---|---|---|---|---|
| 0 | CN1 IDI 0 ~ 7 | CN1 IDI 0 ~ 7 | CN1 IDI 0 ~ 7 | CN1 IDI 0 ~ 7 | IDI 0 ~ 7 | IDI 0 ~ 15 |
| 1 | CN1 IDI 8 ~ 15 | CN2 IDI 8 ~ 15 | - | CN1 IDI 8 ~ 15 | - | - |
| 2 | CN2 IDI 16 ~ 23 | - | - | CN2 DI 0 ~ 7 | - | - |
| 3 | CN2 IDI 24 ~ 31 | - | - | CN2 DI 8 ~ 15 | - | - |

| DI 埠號 | PCI-822LU<br>PCI-826LU<br>PCI-FC16U | PIO-821L<br>PIO-821H<br>PIO-821LU<br>PIO-821HU | PIO-DA4U<br>PIO-DA8U<br>PIO-DA16U | PISO-DA4U<br>PISO-DA8U<br>PISO-DA16U | PEX-DA4<br>PEX-DA8<br>PEX-DA16 | PCI-1002<br>PEX-1002 | PCI-1202<br>PEX-1202 | PCI-1602<br>PCI-1800<br>PCI-1802 |
|---|---|---|---|---|---|---|---|---|
| 0 | **PA 0 ~ 15** | DI 0~7 | DI 0 ~ 7 | DI 0 ~ 7 | DI 0 ~ 7 | DI 0 ~ 15 | DI 0 ~ 15 | DI 0 ~ 15 |
| 1 | **PB 0 ~ 15** | DI 8~15 | DI 8 ~ 15 | DI 8 ~ 15 | DI 8 ~ 15 | - | - | - |

| DI 埠號 | PCI-M512<br>PCI-M512U | PCI-TMC12A | PCI-2602U |
|---|---|---|---|
| 0 | DI 0 ~ 11 | DI 0 ~ 15 | **PA0~7**<br>**PB0~7**<br>**PC0~7**<br>**PD0~7** |

**雙向數位輸出入埠** **數位輸入埠**

雙向數位輸出入埠，在使用前需使用 Ixud_SetDIOModes32 及 Ixud_SetDIOMode 函式設定 I/O 模式爲輸入模式

# A.5. 數位輸出埠定義號碼

| DO 埠號 | PIO-D24U<br>PEX-D24 | PIO-D56U<br>PEX-D56 | PIO-D48U<br>PIO-D48SU<br>PEX-D48 | PIO-D64U | PIO-D96U<br>PIO-D96SU | PIO-D144U<br>PIO-D144LU | PIO-D168U | PISO-A64<br>PISO-C64U<br>PEX-C64 |
|---|---|---|---|---|---|---|---|---|
| 0 | CN3 Port0 | CN3 Port0 | CN1 Port0 | CN1 DO 0 ~ 7 | CN1 Port0 | CN1 Port0 | CN1 Port0 | IDO 0 ~ 7 |
| 1 | CN3 Port1 | CN3 Port1 | CN1 Port1 | CN1 DO 8 ~ 15 | CN1 Port1 | CN1 Port1 | CN1 Port1 | IDO 8 ~ 15 |
| 2 | CN3 Port2 | CN3 Port2 | CN1 Port2 | CN3 DO 0 ~ 7 | CN1 Port2 | CN1 Port2 | CN1 Port2 | IDO 16 ~ 23 |
| 3 | - | CN1 DO 0 ~ 7 | CN2 Port3 | CN3 DO 8 ~ 15 | CN2 Port3 | CN2 Port3 | CN2 Port3 | IDO 24 ~ 31 |
| 4 | - | CN1 DO 8 ~ 15 | CN2 Port4 | - | CN2 Port4 | CN2 Port4 | CN2 Port4 | IDO 32 ~ 39 |
| 5 | - | - | CN2 Port5 | - | CN2 Port5 | CN2 Port5 | CN2 Port5 | IDO 40 ~ 47 |
| 6 | - | - | - | - | CN3 Port6 | CN3 Port6 | CN3 Port6 | IDO 48 ~ 55 |
| 7 | - | - | - | - | CN3 Port7 | CN3 Port7 | CN3 Port7 | IDO 56 ~ 63 |
| 8 | - | - | - | - | CN3 Port8 | CN3 Port8 | CN3 Port8 | |
| 9 | - | - | - | - | CN4 Port9 | CN4 Port9 | CN4 Port9 | |
| 10 | - | - | - | - | CN4 Port10 | CN4 Port10 | CN4 Port10 | |
| 11 | - | - | - | - | CN4 Port11 | CN4 Port11 | CN4 Port11 | |
| 12 | - | - | - | - | - | CN5 Port12 | CN5 Port12 | |
| 13 | - | - | - | - | - | CN5 Port13 | CN5 Port13 | |
| 14 | - | - | - | - | - | CN5 Port14 | CN5 Port14 | |
| 15 | - | - | - | - | - | CN6 Port15 | CN6 Port15 | |
| 16 | - | - | - | - | - | CN6 Port16 | CN6 Port16 | |
| 17 | - | - | - | - | - | CN6 Port17 | CN6 Port17 | |
| 18 | - | - | - | - | - | - | CN6 Port18 | |
| 19 | - | - | - | - | - | - | CN6 Port19 | |
| 20 | - | - | - | - | - | - | CN6 Port20 | |

| DO 埠號 | PISO-P32A32U<br>PISO-P32C32U<br>PISO-P32S32WU<br>PISO-1730U<br>PEX-P32C32 | PISO-P16R16U<br>PEX-P16R16i | PISO-P8R8U<br>PEX-P8R8i<br>PISO-725 | PISO-730<br>PISO-730A<br>PISO-730U<br>PISO-730AU | PCI-P8R8<br>PEX-P8POR8i | PCI-P16R16<br>PCI-P16C16<br>PEX-P16POR16i |
|---|---|---|---|---|---|---|
| 0 | CN1 IDO 0 ~ 7 | CN1 IDO 0 ~ 7 | CN1 IDO 0 ~ 7 | CN1 IDO 0 ~ 7 | IDO 0 ~ 7 | IDO 0 ~ 15 |
| 1 | CN1 IDO 8 ~ 15 | CN2 IDO 8 ~ 15 | - | CN1 IDO 8 ~ 15 | - | - |
| 2 | CN2 IDO 16 ~ 23 | - | - | CN2 DO 0 ~ 7 | - | - |
| 3 | CN2 IDO 24 ~ 31 | - | - | CN2 DO 8 ~ 15 | - | - |

| DO 埠號 | PCI-822LU<br>PCI-826LU<br>PCI-FC16U | PIO-821L<br>PIO-821H<br>PIO-821LU<br>PIO-821HU | PIO-DA4U<br>PIO-DA8U<br>PIO-DA16U | PISO-DA4U<br>PISO-DA8U<br>PISO-DA16U | PEX-DA4<br>PEX-DA8<br>PEX-DA16 | PCI-1002<br>PEX-1002 | PCI-1202<br>PEX-1202 | PCI-1602<br>PCI-1802 |
|---|---|---|---|---|---|---|---|---|
| 0 | **PA 0 ~ 15** | DO 0~7 | DO 0 ~ 7 | DO 0 ~ 7 | | DO 0 ~ 15 | DO 0 ~ 15 | |
| 1 | **PB 0 ~ 15** | DO 8~15 | DO 8 ~ 15 | DO 8 ~ 15 | | - | - | |

| DO 埠號 | PCI-M512 | PCI-TMC12A | PCI-2602U |
|---|---|---|---|
| 0 | DO 0 ~ 15 | DO 0 ~ 15 | **PA 0 ~ 7**<br>**PB 0 ~ 7**<br>**PC 0 ~ 7**<br>**PD 0 ~ 7** |

**雙向數位輸出入埠** **數位輸出埠**

雙向數位輸出埠，在使用前需使用 Ixud_SetDIOModes32 及 Ixud_SetDIOMode 函式設定 I/O 模式爲輸出模式

# 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. UniDAQ 支援 64-bit 的作業系統嗎？

A.支援，由於 UniDAQ 驅動程式函式庫支援 64 位元作業系統。

---

Q. 如果我原來使用 Classic 版本的驅動程式，換成使用 UniDAQ 驅動程式，我需要修改軟體嗎?

A. 需要，爲了加速使用者開發的速度及整合泓格所有 I/O 板卡，所以 UniDAQ 驅動程式的 API 介面設計已與 Classic 版本的驅動程式完全不相容。

---

Q. 我不知道我安裝的驅動程式是 Classic 系列的或是 UniDAQ，請問該如何判別？

A. 請您至裝置管理員，檢查板卡裝置名稱，如果您安裝的是 UniDAQ 驅動程式，裝置名稱前方會有[UniDAQ]字樣，如果沒有即代表安裝的是 Classic 系列驅動程式。

---

Q. 如果系統需要再增加第二張不同的泓格板卡來開發新的專案，一張因爲原先就是使用 Classic 驅動程式，因爲我不想變更軟體，我可以讓第二張板卡使用 UniDAQ 的函式庫來作開發嗎？

A. 可以，請您第一張保持使用 Classic 驅動程式，第二張使用 UniDAQ 驅動程式即可。

---

Q. UniDAQ 支援 ISA 總線的板卡嗎？

A. UniDAQ 目前尚未支援任何 ISA 總線的板卡。

---

數位輸出入

Q. 使用 PIO-D24U/D56U/D48U/D96U/D144U/D168U 時，輸出埠無法輸出，輸入埠無法輸入?

A. 因爲 PIO-D24U/D56U/D48U/D96U/D144U/D168U 的輸出輸入埠爲雙向埠，需先設定規畫埠，請先使用 Ixud_SetDIOModes32 或 Ixud_SetDIOMode 函式設定埠的模式，再對埠作輸出或輸入的動作。

類比輸出

Q. 使用 PIO-DA4U/8U/16U 或 PISO-DA4U/8U/16U 板卡，爲什麼我設定類比輸出範圍 ±5V，0 ~ 10V，0 ~ 5V 及 4~20 mA 時後使用類比輸出函式卻輸出不正確的電壓或電流。

A. 因爲 PIO-DA4U/8U/16U 或 PISO-DA4U/8U/16U 的硬體設計僅支援±10V 的電壓輸出及 0~20 mA 的電流輸出，如果您設定成其他未支援的電壓或電流範圍，將會輸出不正確的電壓及電流。

Q. 使用類比輸出函式輸出一電壓或電流值爲何輸出不正確的電壓或電流。

A. 請您檢查您的電壓範圍設定是否正確，並利用 Ixud_ConfigAO 設定正確輸出範圍，再使用 Ixud_WriteAOVoltage 或 Ixud_WriteAOCurrent 函式輸出電壓或電流。

Q. 錯誤碼 1.

A. 請重新安裝泓格 UniDAQ 驅動函式庫或重新開機。

Q. 錯誤碼 2.

A. (1)請在使用 UniDAQ 函式前使用 Ixud_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 | Sep. 2009 | 初版 |
| 1.3 | Sep. 2011 | 增加函式 |
| 2.0 | June. 2012 | 新增安裝教學、開發指南及函式應用說明 |
| 2.1 | Dec. 2012 | 修正中斷事件配置碼<br>修正中斷支援列表 |
| 2.2 | May. 2013 | 修正 PISO-813 的 CardType 參數設定<br>新增 PCI-1002 支援 Channel Scan |
| 2.3 | Feb. 2014 | 新增新產品<br>新增新的 API 函式 |