

# PCI-1202/1602/180x 軟體說明手冊

---

## (for Windows 98/NT/2K/XP)

### **Warranty**

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

### **Warning**

ICP DAS assumes no liability for damage consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

### **Copyright**

Copyright 1998 by ICP DAS. All rights are reserved.

### **Trademark**

The names used for identification only may be registered trademarks of their respective companies.

## 目錄

<b>1. 導讀.....</b>	<b>4</b>
1.1 函式定義(FUNCTION DEFINE).....	5
1.2 錯誤碼定義(ERROR CODE DEFINE) .....	8
1.3 配置碼(CONFIGURATION CODE) .....	9
1.4 優先權(PRIORITY) .....	10
<b>2. 函式介紹.....</b>	<b>11</b>
2.1 測試函式集 .....	12
2.1.0 P1202_FloatSub2 .....	12
2.1.1 P1202_ShortSub2 .....	12
2.1.2 P1202_GetDllVersion .....	13
2.1.3 P1202_GetDriverVersion.....	13
2.2 驅動函式集 .....	14
2.2.0 P1202_DriverInit .....	14
2.2.1 P1202_DriverClose.....	15
2.2.2 P1202_GetConfigAddressSpace .....	16
2.2.3 P1202_WhichBoardActive .....	17
2.2.4 P1202_ActiveBoard .....	18
2.3 DIGITAL I/O 函式集 .....	19
2.3.0 P1202_Di .....	19
2.3.1 P1202_Do .....	19
2.4 DA 函式集 .....	20
P1202_Da .....	20
2.5 AD 函式集 .....	21
導讀 .....	21
2.5.0 P1202_SetChannelConfig .....	22
2.5.1 P1202_AdPolling .....	22
2.5.2 P1202_AdsPolling .....	23
2.5.3 P1202_AdsPacer.....	24
2.6 MAGICSCAN 函式集 .....	25
2.6.0 導讀.....	25
2.6.1 P1202_ClearScan .....	26
2.6.2 P1202_StartScan.....	27
2.6.3 P1202_StartScanPostTrg .....	28
2.6.4 P1202_StartScanPreTrg .....	29
2.6.5 P1202_StartScanMiddleTrg.....	30

---

2.6.6	<i>P1202_ReadScanStatus</i> .....	31
2.6.7	<i>P1202_AddToScan</i> .....	33
2.6.8	<i>P1202_SaveScan</i> .....	35
2.6.9	<i>P1202_WaitMagicScanFinish</i> .....	36
2.6.10	<i>P1202_StopMagicScan</i> .....	38
2.7	<b>M_FUNCTIONS 函式集</b> .....	39
2.7.0	導論.....	39
2.7.1	<i>P1202_M_FUN_1</i> .....	40
2.7.2	<i>P1202_M_FUN_2</i> .....	42
2.7.3	<i>P1202_M_FUN_3</i> .....	43
2.8	<b>THE SINGLE BOARD BATCH CAPTURE</b> .....	45
2.8.0	導讀.....	45
2.8.1	<i>P1202_FunB_Start</i> .....	46
2.8.2	<i>P1202_FunB_ReadStatus</i> .....	47
2.8.3	<i>P1202_FunB_Stop</i> .....	48
2.8.4	<i>P1202_FunB_Get</i> .....	48
2.9	<b>MULTI-BOARD BATCH CAPTURE</b> .....	49
2.9.0	<i>P1202_FunA_Start</i> .....	49
2.9.1	<i>P1202_FunA_ReadStatus</i> .....	50
2.9.2	<i>P1202_FunA_Stop</i> .....	51
2.9.3	<i>P1202_FunA_Get</i> .....	51
2.10	<b>THE CONTINUOUS CAPTURE 函式集</b> .....	52
2.10.0	導覽.....	52
2.10.1	<i>P1202_Card0_StartScan</i> .....	53
2.10.2	<i>P1202_Card0_ReadStatus</i> .....	54
2.10.3	<i>P1202_Card0_Stop</i> .....	55
2.10.4	<i>P1202_Card1_StartScan</i> .....	56
2.10.5	<i>P1202_Card1_ReadStatus</i> .....	57
2.10.6	<i>P1202_Card1_Stop</i> .....	58
2.11	<b>THE OTHER FUNCTIONS</b> .....	59
2.11.0	<i>P1202_DelayUs</i> .....	59
3.	<b>附錄</b> .....	60
4.	<b>範例程式</b> .....	61

# 1. 導讀

PCI-1202/1602/180x 的發展套件提供動態函式庫(Dynamic Link Library) , 以及簡明易懂的範例程式。參考這些範例程式, 您可以快速的呼叫 DLL 的函式完成控制程序, 不再需要面對繁瑣的硬體暫存器規劃步驟。

本文件為 PCI-1202 系列、PCI-1602 系列及 PCI-180x 系列的 DLL 函式介紹。您可以使用它來了解 DLL 提供了哪些函式, 使用特用功能(連續擷取, 多通道數入)要如何規劃程式; 也可以在編寫控制程式時查閱函式的參數與傳回值等資訊。

DLL 提供的函式分為測試函式集(Test)、驅動函式集(Driver)、類比輸出函式集(D/A)、類比輸入函式集(A/D)、數位輸出入函式集(DI/O)、多通道類比輸入函式集(Magic Scan)、M\_Function 函式集、連續擷取函式集(Continuous Capture)及批次擷取函式集(Batch Capture)。

1.1 節將這些函式集所屬函數簡單列表分類。1.2 節為函式傳回值錯誤碼定義表。1.3 節表列了 PCI-1202/1602/180x 在量測不同的類比輸入範圍時, 調整訊號放大倍率的 Configuration Code。1.4 節簡單列出執行緒優先權的設定值與定義, 您可以參考設定批次或連續擷取資料時執行緒的優先權。

## 1.1 函式定義(Function Define)

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

下表函式名稱與第二章之後的函式使用介紹，是以 PCI-1202 函式庫定義的函式名稱為主。PCI-1602 或 PCI-180x 系列板卡 DLL 函式名稱與 PCI-1202 函數名稱相似，只需更換紅色前置字串即可。

例如：

**P1202**\_ClearScan(void)→**P1602**\_ClearScan(void) 或 **P180x**\_ClearScan(void)

區分	函式定義
Test	float <b>P1202</b> _FloatSub2(float fA, float fB);
	short <b>P1202</b> _ShortSub2(short nA, short nB);
	WORD <b>P1202</b> _GetDllVersion(void);
	WORD <b>P1202</b> _GetDriverVersion(WORD *wVxdVersion);
Driver	WORD <b>P1202</b> _DriverInit(WORD *wTotalBoards);
	void <b>P1202</b> _DriverClose(void);
	WORD <b>P1202</b> _GetConfigAddressSpace(WORD wBoardNo, WORD *wAddrTimer,WORD *wAddrCtrl, WORD *wAddrDio, WORD *wAddrAdda);
	WORD <b>P1202</b> _ActiveBoard( WORD wBoardNo )
	WORD <b>P1202</b> _WhichBoardActive(void);
Digital I/O	WORD <b>P1202</b> _Di(WORD *wDi);
	WORD <b>P1202</b> _Do(WORD wDo);
D/A	WORD <b>P1202</b> _Da(WORD wDaChannel, WORD wDaVal);
A/D	WORD <b>P1202</b> _SetChannelConfig(WORD wAdChannel, WORD wConfig);
	WORD <b>P1202</b> _AdPolling(float *fAdVal);
	WORD <b>P1202</b> _AdsPolling(float fAdVal[], WORD wNum);
	WORD <b>P1202</b> _AdsPacer(float fAdVal[], WORD wNum, WORD wSampleDiv);

區分	函式定義
Magic	<b>WORD</b> P1202_ClearScan(void);
	<b>WORD</b> P1202_StartScan(WORD wSampleRateDiv, DWORD dwNum, SHORT nPriority);
	<b>WORD</b> P1202_StartScanPostTrg(WORD wSampleRateDiv, DWORD dwNum, SHORT nPriority);
	<b>WORD</b> P1202_StartScanPreTrg(WORD wSampleRateDiv, DWORD dwNum, SHORT nPriority);
	<b>WORD</b> P1202_StartScanMiddleTrg(WORD wSampleRateDiv, DWORD dwNum, DWORD dwN2, SHORT nPriority);
	<b>WORD</b> P1202_StartScanPreTrgVerC(WORD wSampleRateDiv, DWORD dwNum, SHORT nPriority);
	<b>WORD</b> P1202_StartScanMiddleTrgVerC(WORD wSampleRateDiv, DWORD dwNum, DWORD dwN2, SHORT nPriority);
	<b>void</b> P1202_ReadScanStatus(WORD *wStatus, DWORD *dwLowAlarm, DWORD *dwHighAlarm);
	<b>WORD</b> P1202_AddToScan(WORD wAdChannel, WORD wConfig, WORD wAverage, WORD wLowAlarm, WORD wHighAlarm, WORD wAlarmType);
	<b>WORD</b> P1202_SaveScan(WORD wAdChannel, WORD wBuf[]);
	<b>void</b> P1202_WaitMagicScanFinish(WORD *wStatus, DWORD *dwLowAlarm, DWORD *dwHighAlarm);
	<b>WORD</b> P1202_StopMagicScan();
M_Function	<b>WORD</b> P1202_M_FUN_1(WORD wDaNumber, WORD wDaWave, float fDaAmplitude, WORD wAdSampleRateDiv, WORD wAdNumber, WORD wAdConfig, float fAdBuf[], float fLowAlarm, float HighAlarm);
	<b>WORD</b> P1202_M_FUN_2(WORD wDaNumber, WORD wDaWave, WORD wDaBuf[], WORD wAdSampleRateDiv, WORD wAdNumber, WORD wAdConfig, WORD wAdBuf[]);
	<b>WORD</b> P1202_M_FUN_3(WORD wDaNumber, WORD wDaWave, float fDaAmplitude, WORD wAdSampleRateDiv, WORD wAdNumber, WORD wChannelStatus[], WORD wAdConfig[], float fAdBuf[], float fLowAlarm, float fHighAlarm);
	<b>WORD</b> P1202_M_FUN_4(WORD wType, WORD wDaNumber, WORD wDaWave, float fDaAmplitude, WORD wAdSampleRateDiv, WORD wAdNumber, WORD wChannelStatus[], WORD wAdConfig[], float fAdBuf[], float fLowAlarm, float fHighAlarm);

區分	函式定義
Continuous Capture	<b>WORD</b> P1202_Card0_StartScan(WORD wSampleRate, WORD wChannelStatus[], WORD wChannelConfig[],WORD wCount);
	<b>WORD</b> P1202_Card0_ReadStatus(WORD wBuf[], WORD wBuf2[], DWORD *dwP1, DWORD *dwP2, WORD *wStatus);
	void P1202_Card0_Stop(void);
	<b>WORD</b> P1202_Card1_StartScan(WORD wSampleRate, WORD wChannelStatus[],WORD wChannelConfig[],WORD wCount);
	<b>WORD</b> P1202_Card1_ReadStatus(WORD wBuf[], WORD wBuf2[], DWORD *dwP1, DWORD *dwP2,WORD *wStatus);
	void P1202_Card1_Stop(void);
	<b>WORD</b> P1202_FunA_Start(WORD wClock0Div, WORD wChannel0[], WORD wConfig0[], WORD Buffer0, DWORD dwMaxCount0, WORD wClock1Div, WORD wChannel1[], WORD wConfig1[], WORD *Buffer1, DWORD dwMaxCount1, SHORT nPriority);
Batch Capture	<b>WORD</b> P1202_FunA_ReadStatus(void);
	<b>WORD</b> P1202_FunA_Stop(void);
	<b>WORD</b> P1202_FunA_Get(DWORD *P0, DWORD *P1);
	<b>WORD</b> P1202_FunB_Start(WORD wClock0Div, WORD wChannel0[], WORD wConfig0[], WORD *Buffer0, DWORD dwMaxCount0, SHORT nPriority);
	<b>WORD</b> P1202_FunB_ReadStatus(void);
	<b>WORD</b> P1202_FunB_Stop(void);
	<b>WORD</b> P1202_FunB_Get(DWORD *P0);

## 1.2 錯誤碼定義(Error Code Define)

函式回傳值定義

回傳值	定義	敘述
0	NoError	正常
1	DriverHandleError	無效的 <b>VxD/SYS</b> 處理。註 1
2	DriverCallError	<b>VxD/SYS</b> 函式呼叫錯誤。註 1
3	AdControllerError	嵌入的控制器發生處理錯誤。可能是硬體發生損壞請連絡代理商作進一步的處理或送修。
4	M_FunExecError	M_Functions 傳回錯誤碼
5	ConfigCodeError	wAdConfig 參數設定碼錯誤
7	HighAlarm	fAdBuf[?]>fHighAlarm
8	LowAlarm	fAdBuf[?]< fLowAlarm
9	AdPollingTimeOut	硬體計時超過時間發生錯誤。註 1
10	AlarmTypeError	僅有 0/1/2/3/4 是有效的驗證值
11	FindBoardError	找不到板卡。請確認板卡是否正確插在 PCI 槽上。
12	AdChannelError	無效的 A/D 頻道
13	DaChannelError	D/A 頻道必須為頻道 0 或是頻道 1
14	InvalidateDelay	dwDelayUs > 8191
15	DelayTimeOut	延遲時間發生超過時間。
16	InvalidateData	無效的資料。
17	FifoOverflow	FIFO 溢出
18	TimeOut	超過時間。
19	ExceedBoardNumber	無效的板卡順序號碼。(第一張版卡號碼為 0)
20	NotFoundBoard	偵測不到板卡。註 1
22	FindTwoBoardError	尋找不到第二張板卡。
23	ThreadCreateError	無法產生執行緒。
24	StopError	停止錯誤
25	AllocateMemoryError	記憶體分配失敗。也許系統缺少可以使用的記憶體。

註 1:請更新你的驅動程式，並且重新啟動電腦。

請到此處下載最新的驅動程式 <http://www.icpdas.com/download/pci/index.htm>



## 1.3 配置碼(Configuration Code)

PCI-1202L/1800L/1802L 配置碼

Bipolar/Unipolar	類比輸入訊號範圍	放大倍率	Settling Time	Configuration Code
Bipolar	+/- 5V	1	3 us	0x00
Bipolar	+/- 2.5V	2	3 us	0x01
Bipolar	+/- 1.25V	4	3 us	0x02
Bipolar	+/- 0.625V	8	3 us	0x03
Bipolar	+/- 10V	0.5	3 us	0x04
Bipolar	+/- 5V	1	3 us	0x05
Bipolar	+/- 2.5V	2	3 us	0x06
Bipolar	+/- 1.25V	4	3 us	0x07
Unipolar	0V ~ 10V	1	3 us	0x08
Unipolar	0V ~ 5V	2	3 us	0x09
Unipolar	0V ~ 2.5V	4	3 us	0x0A
Unipolar	0V ~ 1.25V	8	3 us	0x0B

PCI-1202H/1800H/1802H 配置碼

Bipolar/Unipolar	類比輸入訊號範圍	放大倍率	Settling Time	Configuration Code
Bipolar	+/- 5V	1	23 us	0x10
Bipolar	+/- 0.5V	10	28 us	0x11
Bipolar	+/- 0.05V	100	140 us	0x12
Bipolar	+/- 0.005V	1000	1300 us	0x13
Bipolar	+/- 10V	0.5	23 us	0x14
Bipolar	+/- 1V	5	28 us	0x15
Bipolar	+/- 0.1V	50	140 us	0x16
Bipolar	+/- 0.01V	500	1300 us	0x17
Unipolar	0V ~ 10V	1	23 us	0x18
Unipolar	0V ~ 1V	10	28 us	0x19
Unipolar	0V ~ 0.1V	100	140 us	0x1A
Unipolar	0V ~ 0.01V	1000	1300 us	0x1B

**PCI-1602 配置碼**

Bipolar/Unipolar	類比輸入訊號範圍	放大倍率	Settling Time	Configuration Code
Bipolar	+/- 10V	1	3 us	0
Bipolar	+/- 5V	2	3 us	1
Bipolar	+/- 2.5V	4	3 us	2
Bipolar	+/- 1.25V	8	3 us	3

## 1.4 優先權(Priority)

執行緒的優先權設定

設定值	定義
-2	THREAD_PRIORITY_LOWEST
-1	THREAD_PRIORITY_BELOW_NORMAL
<b>0(Default)</b>	<b>THREAD_PRIORITY_NORMAL</b>
1	THREAD_PRIORITY_ABOVE_NORMAL
2	THREAD_PRIORITY_HIGHEST
15	THREAD_PRIORITY_TIME_CRITICAL
Others	THREAD_PRIORITY_NORMAL

## 2. 函式介紹

PCI-1202 函式庫提供的副程式函數主要可區分為以下 9 個子集:

- 測試 (Test) 函式集
- 驅動 (Driver)函式集
- Digital I/O 函式集
- D/A 函式集
- A/D 函數
- A/D MagicScan mode 函式集
- M\_Function 函式集
- Continuous-Capture 函式集
- Batch-Continuous-Capture 函式集

章節 1.1 已詳細列出每個子集下包含的函式清單。以下將依序說明這些函式的參數, 功能與可參考的範例程式。

## 2.1 測試函式集

---

### 2.1.0 P1202\_FloatSub2

計算  $C=A-B$ (浮點數格式), **float=4**個位元組的浮點數. 這個函式目的在測試動態連結函式庫(P1202.DLL)是否可以連結, 正確的傳回運算的結果.

- **語法:**

float P1202\_FloatSub2(float fA, float fB);

- **參數:**

**fA** : [Input] 4 個位元組的浮點數

**fB** : [Input] 4 個位元組的浮點數

- **回傳值:**

傳回  $fA-fB$  之值。

- **範例程式:**

DEMO1.C

---

### 2.1.1 P1202\_ShortSub2

計算  $C=A-B$  (短整數格式), **SHORT=16** 個位元有號數. 這個函式目的在測試動態連結函式庫(P1202.DLL)是否可以連結, 正確的傳回運算的結果.

- **語法:**

short **P1202**\_ShortSub2(Short nA, Short nB);

- **參數:**

**nA** : [Input] 16 位元整數

**nB** : [Input] 16 位元整數

- **Return:**

傳回  $nA-nb$  之值。

- **範例程式:**

DEMO1.C

---

## 2.1.2 P1202\_GetDllVersion

讀取 P1202.DLL 的 DLL 版本號碼。

- **語法：**

WORD **P1202\_GetDllVersion**(void);

- **參數：**

無參數

- **回傳值：**

直接傳回 DLL 版本，如果回傳值等於 0x200，DLL 版本為 2.0

- **範例程式：**

DEMO1.C

---

## 2.1.3 P1202\_GetDriverVersion

讀取 NAPPCI.VxD 或是 NAPPCI.SYS 的版本。

- **語法：**

WORD **P1202\_GetDriverVersion**(WORD \*wDriverVersion);

- **參數：**

**\*wDriverVersion** : **[output]** 傳址參數。傳回的驅動程式版本號碼。

※wDriverVersion=0x200 → 驅動程式的版本號碼為 2.0

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：**

DEMO1.C

## 2.2 驅動函式集

---

### 2.2.0 P1202\_DriverInit

這個函式會向系統要求分配資源，偵測板卡是否已安裝成功，並傳回安裝板卡的數量。您必須在程式起始處，使用其他函式之前呼叫這個函式。

- **語法：**

WORD P1202\_DriverInit(WORD \*wTotalBoard);

- **參數：**

**\*wTotalBoard** :[Output] 傳址參數，用來儲存及傳回所尋找到的板卡數量。

※wTotalBoard=1 → 系統安裝有一張 PCI-1202 板卡

wTotalBoard=n → 系統安裝有 N 張 PCI-1202 板卡

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

所有的範例程式。

## 2.2.1 P1202\_DriverClose

釋放板卡佔用的資源。

這個函式必需在程式結束前呼叫，將佔用的資源釋放歸還給系統。

- **語法：**

`void P1202_DriverClose(void);`

- **參數：**

無參數

- **回傳值：**

無回傳值

- **範例程式：**

所有範例程式。

## 2.2.2 P1202\_GetConfigAddressSpace

取得第 N 張卡的 I/O 位址。這個函式沒有呼叫不會影響到板卡其他程序的使用。

- **語法：**

```
WORD P1202_GetConfigAddressSpace(WORD wBoardNo, WORD  
*wAddrTimer, WORD *wAddrCtrl, WORD *wAddrDio, WORD *wAddrAdda);
```

- **參數：**

wBoardNo: [Input] 用戶指定的板卡編號,由 0 算起.

\*wAddrTimer: [Output] 取得 8254 計時器的 I/O address map.

\*wAddrCtrl: [Output] 取得 Controller 的 I/O address map.

\*wAddrDio: [Output] 取得 Digital I/O 埠的 I/O address map.

\*wAddrAdda: [Output] 取得 A/D 及 D/A 的 I/O address map.

※請參考“PCI-1202/1602/1800/1802 Hardware manual”第三章 會有詳細的說明.

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO1.C



## 2.2.3 P1202\_WhichBoardActive

傳回正在啟用(Active)狀態中的板卡號碼。P1202.dll

- **語法:**

WORD **P1202\_WhichBoardActive**(void);

- **參數:**

無參數

- **回傳值:**

將正在啟用狀態中的板卡號碼回傳出來。如果您使用第二張板卡。將會回傳 1  
(第一張板卡會回傳 0 而不是 1，板卡排序是從零開始)

- **範例程式:**

DEMO1.C

## 2.2.4 P1202\_ActiveBoard

這個函式會啟動(Active)您選用的板卡，就像打開電視的電源開關一樣。當系統安裝兩張(含)以上的板卡時，同時只能設定一張板卡為啟動狀態。要對非啟動狀態的板卡進行 Digital I/O、A/D、D/A、M\_Function、MagicScan 及 Continuous capture 等程序之前必須先呼叫這個函式啟動 wBoardNo 對應的板卡。

- **語法：**

WORD P1202\_ActiveBoard(WORD wBoardNo);

- **參數：**

wBoardNo: 要啟動的板卡編號 (從 0 開始，第一張版卡號碼為 0)。

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：**

所有的範例程式。

- **參考資料：**

不需要呼叫 P1202\_ActiveBoard()即可運作的函式如下:

P1202_FloatSub2
P1202_ShortSub2
P1202_GetDriverVersion
P1202_DriveInit
P1202_DriveClose
P1202_GetConfigAddressSpace
P1202_Card0_StartScan
P1202_Card0_ReadData
P1202_Card0_Stop
P1202_Card1_StartScan
P1202_Card1_ReadData
P1202_Card1_Stop

## 2.3 Digital I/O 函式集

### 2.3.0 P1202\_Di

這個函式會從 Digital Input 埠讀取 16 位元的數位資料回來。它只能作用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard (2.2.4 節) 啟動您要使用的板卡。

- **語法：**

WORD P1202\_Di(WORD \*wDi);

- **參數：**

**\*wDi** : [Output] 傳址參數。用來儲存並傳回一個 16 位元的 DI(輸入)值

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：**

DEMO1.C

### 2.3.1 P1202\_Do

這個函式用來傳送一 16 位元的值到 Digital Output 埠。它只能作用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

- **語法：**

WORD P1202\_Do(WORD wDo);

- **參數：**

**wDo** : [Input] 一個 16 位元的輸出值。

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：**

DEMO1.C

**CN1 與 CN2 對接時：**

P1202\_Do(0x10) ;//輸出 0x10

P1202\_Di(\*wDi) ;//\*wDi=0x10 輸入

\*wDi 值會依接線或外部信號之不同而有所改變。

## 2.4 DA 函式集

---

### 2.4.0 P1202\_Da

這個函式會傳送一 12 位元的值至類比輸出埠。輸出電壓的範圍由板卡的 JP1 跳線決定是 10V 至 -10V 之間或 5 ~ -5 之間。它只能作用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節)函式啟動您要使用的板卡。

- **語法：**

WORD P1202\_Da(WORD wChannel, WORD wDaVal);

- **參數：**

**wChannel** : [Input] 類比輸出通道。

**wChannel=0** (設定輸出到 analog output Channel 0)

**wChannel=1** (設定輸出到 analog output Channel 1)

**wDaVal** : [Input] 一 12 位元值將會傳送到 DA 埠。此值就是控制類比輸出的電壓大小。最小值為 0 最大值為 4095。類比輸出的範圍有二種選擇分別為 +/- 5V 或 +/- 10V 可以藉由設定版卡上的 JP1 來調整。但是函式無法偵測板卡上 JP1 的設定。所以當 wDaVal 等於 4095 時可能是+10V 也可能是+5V, 是由 JP1 的設定決定。

- **回傳值：**

0 : 無錯誤

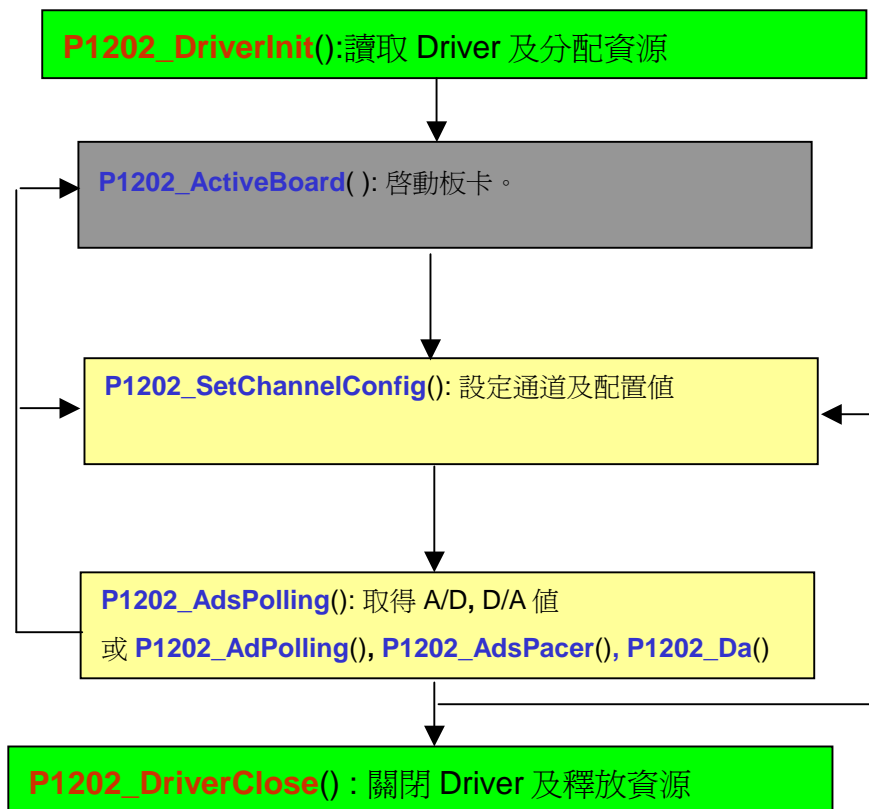
其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：** DEMO1.C

## 2.5 AD 函式集

### 2.5.0 導讀

#### AD/DA 輪詢 函式使用程序



## 2.5.1 P1202\_SetChannelConfig

用來設定各通道的配置值(請參考 1.3 節)。呼叫 **P1202\_AdPolling**, **P1202\_AdsPolling** 及 **P1202\_AdsPacer** 之前, 須先使用此函式設定要從那一個通道擷取資料與合適的配置值(sec.1.3). 它只能使用在啟動狀態中的板卡, 請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

- **語法：**

WORD P1202\_SetChannelConfig(WORD wChannel, WORD wConfig);

- **參數：**

**wChannel** : [Input] 選擇擷取資料的通道。

**wConfig** : [Input] 設定資料擷取範圍配置碼。請參考 1.3 節。

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：**

DEMO1.C

---

## 2.5.2 P1202\_AdPolling

這個函式以軟體輪詢的方式作一次 AD 轉換, 並傳回此轉換值。呼叫這個函式之前需先使用 **P1202\_SetChannelConfig()** 設定(或改變)通道號碼與配置值。P1202\_AdPolling 會根據您的設定令指定的通道完成一次 AD 轉換, 並回傳此轉換值。它只能使用在啟動狀態中的板卡, 請使用 P1202\_ActiveBoard (2.2.4 節) 啟動您要使用的板卡。

- **語法：**

WORD P1202\_AdPolling(float \*fAdVal);

- **參數：**

**\*fAdVal** : [Output] 傳址參數。用來儲存一個 A/D 資料。

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：**

DEMO1.C

## 2.5.3 P1202\_AdsPolling

如果想要取得一個以上的值，使用上一個函式P1202\_AdPolling，需要重複多次的呼叫，將會降低連續擷取的效率。所以如果想要連續性的取得AD值的話可以使用這個函式來操作。它會一次將所有轉換的AD值通通存入到buffer裡面，這樣用起來就方便很多了。**P1202\_AdsPolling** 函式利用軟體輪詢的方式對ADC trigger（觸發），使 ADC 開始作轉換。呼叫這個函式之前需先使用**P1202\_SetChannelConfig()**設定(或改變)頻道及配置值。P1202\_AdsPolling 將會根據您的設定來操作。它只能使用在啟動狀態中的板卡，請使用P1202\_ActiveBoard(2.2.4節) 啟動您要使用的板卡。

- **語法：**

WORD P1202\_AdsPolling(float fAdVal[], WORD wNum);

- **參數：**

**fAdVal[]:[Output]**是一個浮點數陣列。用來儲存 ADC 轉換完成的數據。每一筆數據與 **P1202\_SetChannelConfig()**的設定會有關聯性。

**wNum:[Input]** 由用戶指定要取得的數據筆數。

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO1.C

## 2.5.4 P1202\_AdsPacer

這個函式與P1202\_AdsPolling()相同，用來取得同一通道上的多筆連續資料，呼叫的程序也與P1202\_AdsPolling()相似。唯一不同的是這個函式透過硬體8254計時器在固定的時間產生觸發信號通知AD轉換器將類比信號轉換為數位值。

- **語法：**

WORD P1202\_AdsPacer(float fAdVal[], WORD wNum, WORD wSampleDiv);

- **參數：**

fAdVal[]: [Output] 是一個浮點數陣列。用來儲存 ADC 轉換完成後得到的數據。每一筆數據與 P1202\_SetChannelConfig()的設定會有關聯性

wNum: [Input] 一次轉換幾筆 AD 值。

wSampleDiv: [Input] **AD 取樣頻率 = 8M/wSampleDiv.**

Example:

wSampleDiv =80 → 取樣頻率=8M/80=100K

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：** DEMO1.C

P1202\_SetChannelConfig  
P1202\_AdPolling  
P1202\_AdsPolling  
P1202\_AdsPacer

Fix channel AD conversion mode





## 2.6 MagicScan 函式集

### 2.6.0 導讀

- 使用 MagicScan 的程序:

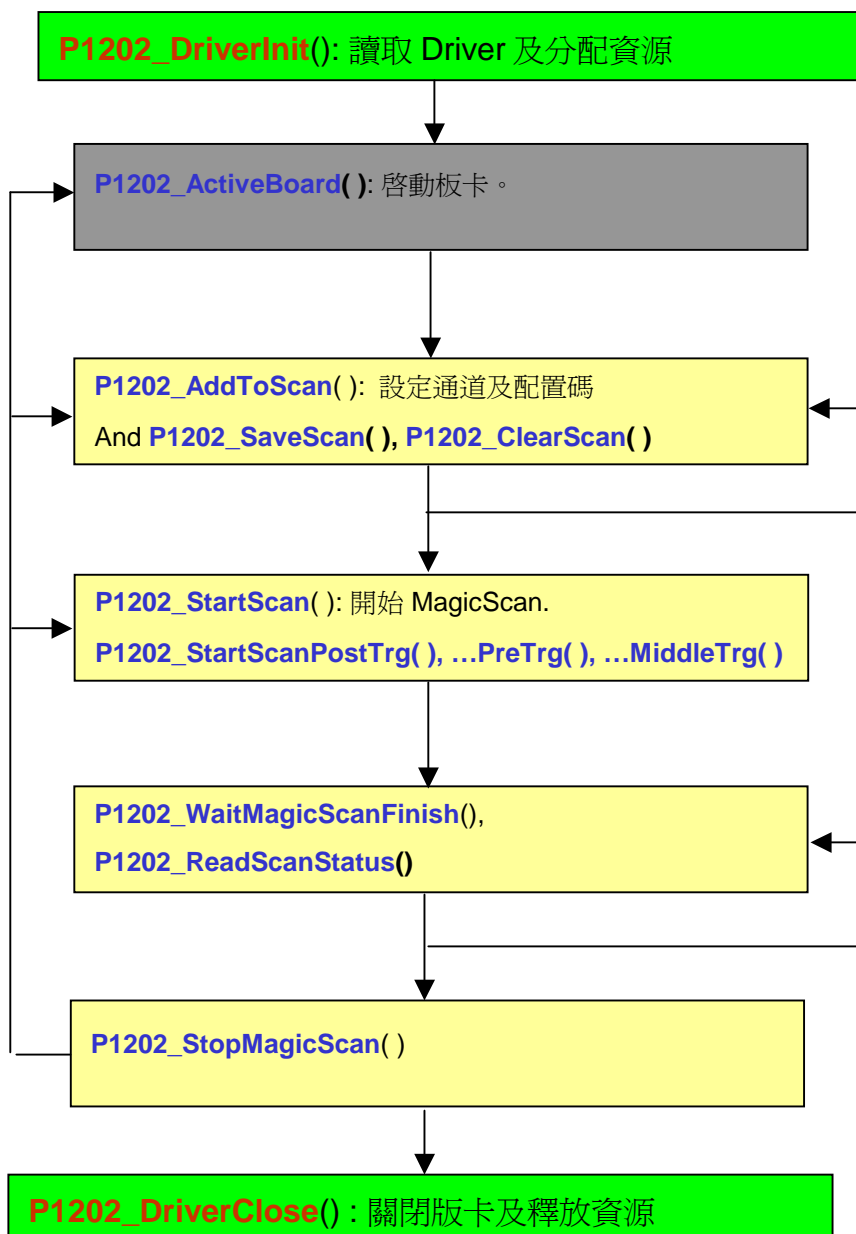
步驟 1. 設定使用 MagicScan 的通道掃描順序與其配置碼

步驟 2. 開始 MagicScan

步驟 3. 取得 MagicScan 的狀態

步驟 4. 取得資料

下圖為 MagicScan 的函式呼叫流程圖(以 PCI-1202 為例):



## 2.6.1 P1202\_ClearScan

這個函式程序用來重新設定 **MagicScan controller**※到初始狀態。它只能使用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

※：請參考第三章的註 1。

- **語法：**

WORD P1202\_ClearScan();

- **參數：**

無參數

- **函數回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO11.C

## 2.6.2 P1202\_StartScan

這個程序用來啟動 MagicScan 的運作。您可以使用 **P1202\_WaitMagicScanFinish()** 或是 **P1202\_ReadScanStatus()** 檢查 MagicScan 的運作狀態。它只能使用在啟動狀態中的板卡，請使用 **P1202\_ActiveBoard(2.2.4 節)** 啟動您要使用的板卡。

- **語法：**

WORD **P1202\_StartScan**(WORD wSampleRateDiv, WORD wNumCycles, ,  
SHORT nPriority);

- **參數：**

**wSampleRateDiv** :[Input] **A/D 取樣頻率 = 8M/wSampleRateDiv.**

wSampleRateDiv=80 → 取樣頻率=8M/80=100K

**wNumCycles** :[Input] 每一次掃描命令要完成的掃描週數。換句話說，也就是每一次掃描命令完成時每一個通道要取得的資料數

**nPriority** :[Input] 設定執行緒的優先權(Priority)。請參考 1.4 節。

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

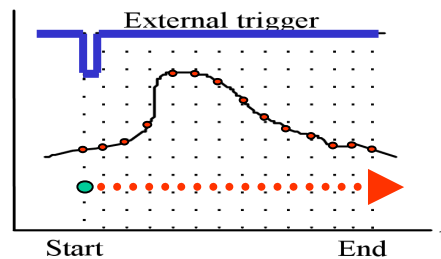
- **範例程式：**

DEMO11.C

## 2.6.3 P1202\_StartScanPostTrg

這個函式使用在當外部觸發信號發生時，**Magic Controller※** 會命令 Timer1 定時觸發 AD 轉換，簡而言之就是先接收到外部信號才會開始抓資料。使用者可以使用 **P1202\_ReadScanStatus()** 這個函式檢查運作狀態。使用 **P1202\_StopMagicScan(...)** 函式停止 MagicScan 運作(強制結束)

※：請參考第三章的註 1。



- **語法：**

WORD **P1202\_StartScanPostTrg**(WORD wSampleRateDiv, WORD wNumCycles, , SHORT nPriority);

- **參數：**

**wSampleRateDiv** :[Input] **A/D 取樣頻率 = 8M/wSampleRateDiv.**

wSampleRateDiv=80 → 取樣頻率=8M/80=100K

**wNumCycles** :[Input] 每一次掃描命令要完成的掃描週數。換句話說，也就是每一次掃描命令完成時每一個通道要取得的資料數。

**nPriority** :[Input] 設定執行緒的優先權(Priority)。請參考 1.4 節。

- **回傳值：**

0：無錯誤

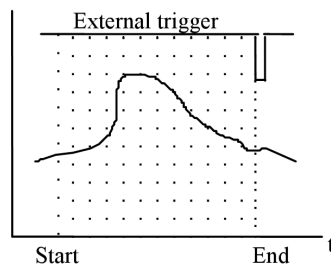
其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO23.C

## 2.6.4 P1202\_StartScanPreTrg

一開始先以指定的頻率將資料取回並存放在 FIFO。此函式會先設定好 Timer 並開始觸發擷取 AD 轉換，直到收到外部觸發信號才會結束 MagicScan 的動作，然後傳回觸發之前的 **wNumCycles** 筆資料。簡而言之就是先抓取資料再接收外部觸發信號。使用者可以使用 **P1202\_ReadScanStatus()** 函式檢查運作狀態；使用 **P1202\_StopMagicScan(...)** 函式停止 MagicScan 運作(強制結束)。



- **語法：**

```
WORD P1202_StartScanPreTrg(WORD wSampleRateDiv, WORD
wNumCycles, , SHORT nPriority);
```

- **參數：**

**wSampleRateDiv** :[Input] **AD 取樣頻率 = 8M/wSampleRateDiv.**

wSampleRateDiv=80 → 取樣頻率=8M/80=100K

**wNumCycles** :[Input] 每一次掃描命令要完成的掃描週數。換句話說，也就是每一次掃描命令完成時每一個通道要取得的資料數。

**nPriority** :[Input] 設定執行緒的優先權(Priority)。請參考 1.4 節。

- **回傳值：**

0：無錯誤

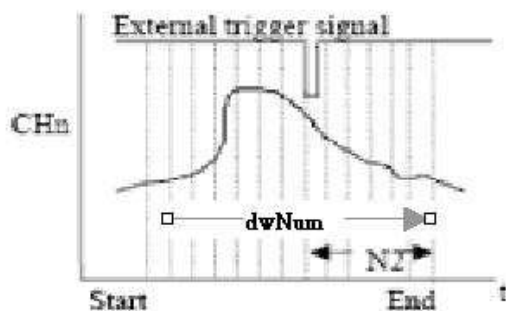
其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO24.C

## 2.6.5 P1202\_StartScanMiddleTrg

這個函數大致上與上一個函數作法相同，唯一不同之處就是在外部信號觸發時會先擷取觸發後的 **N2** 筆資料再停止 MagicScan 的動作。使用者可以使用 **P1202\_ReadScanStatus()** 這個函式去檢查運作狀態。使用 **P1202\_StopMagicScan(...)** 函式來停止 MagicScan 運作(強制結束)。



- 語法：

WORD P1202\_StartScanMiddleTrg(WORD wSampleRateDiv, DWORD dwNum, WORD wN2, SHORT nPriority);

- 參數：

**wSampleRateDiv** :[Input] AD 取樣頻率 =  $8M/wSampleRateDiv$ .

wSampleRateDiv=80 → 取樣頻率 =  $8M/80=100K$

**dwNum** :[Input] 設定要取得外部觸發訊號觸發之前的資料筆數(per channel)。

**WN2** :[Input] 設定要取得外部觸發訊號觸發之後的資料筆數(per channel)。

**nPriority** :[Input] 設定執行緒的優先權(Priority)。請參考 1.4 節。

- 回傳值：

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- 範例程式：

DEMO25.C

## 2.6.6 P1202\_ReadScanStatus

這個函式目的在取得 **MagicScan**、**Pre-trigger**、**Post-trigger**、**Mid-trigger** 的運作狀態。它只能用來檢查狀態但並不會阻斷程式的運作。所以建議您必須隨時檢查狀態，當狀態值 0X80 時為正常，若為 0X01 時則代表失敗。

它只能使用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

- **語法：**

```
void P1202_ReadScanStatus(WORD *wStatus, DWORD *dwLowAlarm,
DWORD *dwHighAlarm);
```

- **參數：**

**\*wStatus** :**[Output]** 傳址函數。用來儲存 MagicScan 的狀態。

**Table 2.8.6.1 wStatus 值狀態表**

wStatus Value(HEX)	狀態	敘述
0x00	initial	MagicScan 初始狀態 (發呆中)
0x01	start	MagicScan 工作中
0x02	time out1	MagicScan stage 1 controller 超出時間
0x04	time out2	MagicScan stage 2 controller 超出時間
0x08	overflow	MagicScan FIFO overflow※註
0x80	OK	MagicScan 完成

※註：請參考第三章的註 2

**\*dwLowAlarm** :**[Output]** 傳址函數。用來設定 MagicScan 每個通道是否使用 Low alarm 狀態。

**\*dwHighAlarm** :**[Output]** 傳址函數。用來設定 MagicScan 每個通道是否使用 High alarm 狀態。

**dwLowAlarm(bit 0~15)**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
channel	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

**dwLowAlarm(bit 16~31)**

Bit	1F	1E	1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10
channel	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

**dwLowAlarm** 是 32 個 bit 的參數直接對應到 32 個頻道。

Ch ? = 0 → 無 low alarm

Ch ? = 1 → 有 low alarm

Example:

dwLowAlarm=0 → 所有頻道正常，皆無 low alarm

dwLowAlarm=1 → 頻道 0 是 low alarm, 其他皆正常。

dwLowAlarm=3 → 頻道 0、1 是 low alarm, 其他皆正常。

### dwHighAlarm(bit 0~15)

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
channel	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

### dwHighAlarm(bit 16~31)

Bit	1F	1E	1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10
channel	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

**dwHighAlarm** 是 32 個 bit 的參數直接對應到 32 個通道。

Ch ? = 0 → 無 high alarm

Ch ? = 1 → 有 high alarm

Example:

dwHighAlarm=0 → 所有頻道正常，皆無 high alarm

dwHighAlarm=4 → 頻道 2 是 high alarm, 其他皆正常。

dwHighAlarm=7 → 頻道 0、1、2 是 high alarm, 其他皆正常。

- 回傳值:

無回傳值

- 範例程式:

DEMO11.C



## 2.6.7 P1202\_AddToScan

這個函式用來新增一個通道到 **MagicScan** 循環掃描佇列。**MagicScan** 循環掃描佇列最多可以增加到 48 個通道。呼叫這個函式增加通道的順序極為掃描的順序，例如：

```
P1202_AddToScan(1, 0, 1, 0, 0, 0);  
P1202_AddToScan(1, 0, 1, 0, 0, 0);  
P1202_AddToScan(0, 0, 1, 0, 0, 0);  
則循環掃描順序為 ch1 -> ch1 -> ch0 -> ch1 -> ch1 -> ch0 -> ch1
```

它只能使用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

- **語法：**

word **P1202\_AddToScan**(WORD wAdChannel, WORD wConfig, WORD wAverage, WORD wLowAlarm, WORD wHighAlarm, WORD wAlarmType);

- **參數：**

**wAdChannel** :[Input] A/D 通道號碼。

**wConfig** :[Input] 配置碼。請參考 1.3 節。

**wAverage** :[Input] 數位平均篩選值。在一個平均數量裡作出平均值。例如：擷取十個 AD 值作平均計算出來的平均值。最小值為 1。

(數位平均篩選值)如果將 **wAverage** 設為 5。你將會得到一個值，這一個值是由五個擷取值所平均計算出來的。所以取樣頻率也會只剩下五分之一。

**wLowAlarm** :[Input] 12 bits low alarm 資料

**wHighAlarm** :[Input] 12 bits high alarm 資料

**wAlarmType** :[Input] Alarm 型態.

Table 2.8.7.1 Alarm Type Define Table

AlarmType Value	Alarm Type	Comment
0	No alarm	關閉
1	High alarm	範圍以上
2	Low alarm	範圍以下
3	In alarm	介於之間
4	Out alarm	在範圍之外

Example:

**P1202\_AddToScan(0, 0, 1, 0, 0, 0)**

**wAdChannel** =0→ 開啓頻道 0 加入掃描

**wConfig** = 0→設定 A/D 範圍爲 5V~ -5V

**wAverage**=1→不作平均

**wLowAlarm**=0 →無 LowAlarm

**wHighAlarm**=0 →無 HighAlarm

**wAlarmType**=0→關閉 Alarm

- 回傳值：

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- 範例程式：DEMO11.C

## 2.6.8 P1202\_SaveScan

這個函式用來指定循環掃描取得的資料儲存的位置。須先為每一個加入循環掃描的通道宣告適當大小的記憶體空間，再用這個函式指定循環掃描佇列中每個通道的資料要放在哪一個記憶體中。

- **語法：**

```
void P1202_SaveScan(WORD wAdChannel, WORD wBuf[]);
```

- **參數：**

**wAdChannel** :[Input] 循環掃描順序號碼，從 0 開始。(注意：非通道號碼)。

例如：第一個被掃描的通道，可能是任一個實體通道，但在這個函式中 wAdChannel 都是 0，因為這個參數是掃描的順序號碼。

**wBuf** :[Output] 儲存 wAdChannel 取回資料的記憶體。

Example:

```
WORD wV0[100000];      // AD ch:0 buffer
WORD wV2[100000];      // AD ch:4 buffer
:
wRetVal=P1202_ClearScan();
//**** For PCI-1202L
wRetVal += P1202_AddToScan(0,0,1,0,0,0); // add CH:0 to scan
wRetVal += P1202_SaveScan(0,wV0);
/*CH0 為第一個加入掃描的通道, 掃描順序號碼為 0, 將掃描順序號碼為 0 這個通道的資料存入 wV0[ ] buffer 裡*/

wRetVal += P1202_AddToScan(4,0,1,0,0,0); // 將通道 4 加入循環掃描佇列
wRetVal += P1202_SaveScan(1,wV2);
//注意：1 => 第二個被掃描的通道，而非實體通道號碼 4
/* CH4 是第二個加入掃描順序的通道, 掃描順序號碼為 1, 將掃描順序號碼為 1 的這個通道的資料存入 wV2[ ] buffer 裡*/

wSampleRateDiv=80;      // 取樣頻率=8M/80 = 100k
P1202_StartScan(wSampleRateDiv,DATALENGTH,nPriority); // 開始掃描
```

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式** : DEMO11.C

## 2.6.9 P1202\_WaitMagicScanFinish

這個程序用作擱置模式 (blocking mode) 處理及等待持續到 MagicScan 運作停止為止。

它只能使用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

- **語法：**

```
void P1202_WaitMagicScanFinish(WORD *wStatus, DWORD  
*dwLowAlarm, DWORD *dwHighAlarm);
```

- **參數：**

**wStatus** :[Output] 傳址函數。用來檢視 MagicScan 的狀態。

**Table 2.8.6.1 wStatus 值狀態表**

wStatus Value(HEX)	狀態	敘述
0x00	initial	MagicScan 初始狀態 (發呆中)
0x01	start	MagicScan 開始運作
0x02	time out1	MagicScan stage 1 controller 超出時間
0x04	time out2	MagicScan stage 2 controller 超出時間
0x08	overflow	MagicScan FIFO overflow※註
0x80	OK	MagicScan 完成

※註：請參考第三章的註 2

**\*dwLowAlarm** :[Output] 傳址函數。用來檢視 MagicScan 每個通道是否處於 Low alarm 狀態。

**\*dwHighAlarm** :[Output] 傳址函數。用來檢視 MagicScan 每個通道是否處於 High alarm 狀態。

**dwLowAlarm(bit 0~15)**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
channel	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

**dwLowAlarm(bit 16~31)**

Bit	1F	1E	1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10
channel	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

**wLowAlarm** 是 32 個 bit 的參數直接對應到 32 個通道。

Ch ? = 0 → 無 low alarm

Ch ? = 1 → 有 low alarm

Example:

dwLowAlarm=0 → 所有頻道正常，皆無 low alarm

dwLowAlarm=1 → 通道 0 是 low alarm, 其他皆正常。

dwLowAlarm=3 → 通道 0、1 是 low alarm, 其他皆正常。

**dwHighAlarm(bit 0~15)**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
channel	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

**dwHighAlarm(bit 16~31)**

Bit	1F	1E	1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10
channel	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

**dwHighAlarm** 是 32 個 bit 的參數直接對應到 32 個通道。

Ch ? = 0 → 無 high alarm

Ch ? = 1 → 有 high alarm

Example:

dwHighAlarm=0 → 所有通道正常，皆無 high alarm

dwHighAlarm=4 → 通道 2 是 high alarm, 其他皆正常。

dwHighAlarm=7 → 通道 0、1、2 是 high alarm, 其他皆正常。

- **回傳值:**

無回傳值

- **範例程式:**

DEMO11.C

## 2.6.10 P1202\_StopMagicScan

停止 MagicScan 的程序

- **語法：**

`void P1202_StopMagicScan(void);`

- **參數：**

無參數

- **回傳值：**

無回傳值

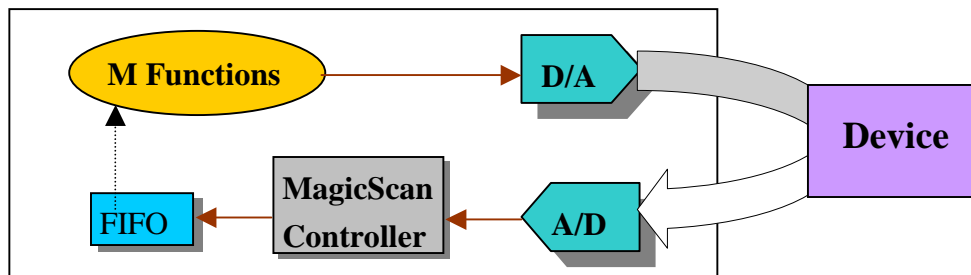
- **範例程式：**

DEMO14.C

## 2.7 M\_Functions 函式集

### 2.7.0 導論

**M\_Function** 應用程式運作原理：



※MagicScan Control：請參考第三章註 1。

Table 2.9.0.1 M\_Function 特性一覽表

Function Name	D/A	A/D
M_FUN_1	D/A 通道 0(Analog Output 0) 輸出正弦波	A/D 通道 0(Analog Input 0) +10V ~ -10V
M_FUN_2	D/A 通道 0(Analog Output 0) 輸出值可自行設定	A/D 通道 0(Analog Input 0) +10V ~ -10V
M_FUN_3	D/A 通道 0(Analog Output 0) 輸出正弦波	多個 A/D 通道 使用者決定(最多 32 個通道)
M_FUN_4	D/A 通道 0(Analog Output 0) 輸出正弦波、方波、半方波	多個 A/D 通道 使用者決定(最多 32 個通道)

## 2.7.1 P1202\_M\_FUN\_1

此函式是根據輸入的點數與振幅，計算出描述一個正弦波的每個單點的數值，從板卡的 DA Channel 0 連續輸出。使用者也可以設定要將這組正弦波數值連續多次輸出，輸出幾組由使用者自行設定。輸出的同時間從 AD Channel 0 擷取資料，使用者可以設定要擷取的速度與資料數。

參考「PCI-1202/1602/1800/1802 Hardware manual」第五章有更多詳細的內容。

※ 此函式輸出波形到 analog output 0 的同時，會從 analog input 0 擷取資料。

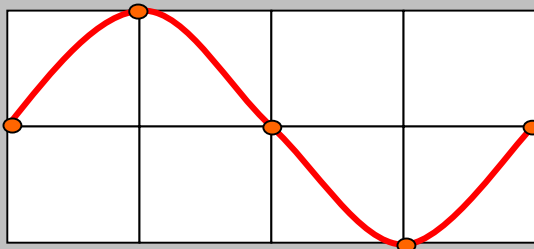
### ● 語法：

```
WORD P1202_M_FUN_1(WORD wDaNumber, WORD wDaWave, float
    fDaAmplitude, WORD wAdSampleRateDiv, WORD wAdNumber,
    WORD wAdConfig, float fAdBuf[], float fLowAlarm, float fHighAlarm)
```

### ● 參數：

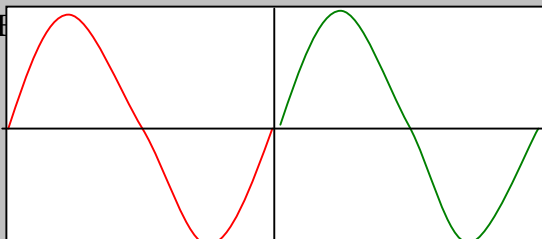
**wDaNumber:** [Input] 描述一組輸出的正弦波要用的點數。點數越多，波形會越像真正的類比輸出；但因輸出的點數多，輸出一個完整波的速度會變慢。

Example1:



左圖是將 wDaNumber 設為 5，所以他將會輸出一組有五個 DA 值的正弦波，而這五個值由程序自動計算而生成一個可以模擬波型的五個點。要送出一組同樣點數的波的時間通常取決於電腦的效能，效能愈好時間愈短。

**wDaWave:** [Input] 輸出多少組的正弦波。



當 wDaWave 設定為 2 時。將會送出二組正弦波。



**fDaAmplitude** : [Input] 正弦波的振幅.

※fDaAmplitude 範圍為 -10~10

注意：板卡上的 J1 一定要設定成 +/-10V

**wAdSampleRateDiv**: [Input] AD 取樣頻率 = 8M/ wAdSampleRateDiv (取樣/秒)

**wAdNumber**: [Input] 讀取 AD 資料的筆數。

**wAdConfig**: [Input] 配置碼，設定錯誤會量測到不正確的值請小心。

請參考 1.3 節

**fAdBuf[]** : [Output] 為一浮點數陣列。用來儲存 A/D 資料的 buffer

**fLowAlarm** : [Input] 最低警戒值。當 **fAdBuf[?]** 小於 **fLowAlarm** → LowAlarm

**fHighAlarm** : [Input] 最高警戒值。當 **fAdBuf[?]** 大於 **fHighAlarm** → HighAlarm

- 回傳值：

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- 範例程式：DEMO5.C

## 2.7.2 P1202\_M\_FUN\_2

其基本上跟上一個函式的用法類似，只是本函式輸出的波形是由使用者自由設定每一個 DA 值，而不像 P1202\_M\_FUN\_1 是由函式自動產生。

參考「PCI-1202/1602/1800/1802 Hardware manual」第五章有更詳細的內容。

※此函式輸出波形到 **analog output 0** 的同時，會從 **analog input 0** 擷取資料。

- **語法：**

```
WORD P1202_M_FUN_2(WORD wDaNumber, WORD wDaWave, WORD  
wDaBuf[], WORD wAdSampleRateDiv, WORD wAdNumber, WORD  
wAdConfig, WORD wAdBuf[]);
```

- **參數：**

**wDaNumber**: [Input] 描述使用者自由定義的波形使用的點數。

請參考 P1202\_M\_FUN\_1 的 Example 1

**wDaWave**: [Input] 輸出多少組完整波形。

請參考 P1202\_M\_FUN\_1 的 Example 2

**wDaBuf[]**: [Output] 儲存使用者自由定義的波形的數值陣列。

**wAdSampleRateDiv**: [Input] **AD 取樣頻率 = 8M/ wAdSampleRateDiv** (取樣數/秒)

**wAdNumber**: [Input] 讀取 AD 資料的筆數

**wAdConfig**: [Input] 配置碼，設定錯誤會量測到不正確的值請小心。

請參考 1.3 節

**wAdBuf[]**: [Output] 為一浮點數陣列。用來儲存 A/D 資料的記憶體空間。

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO7.C

## 2.7.3 P1202\_M\_FUN\_3

此函式是根據輸入的點數與振幅，計算出符合的正弦波的每個單點的數值，從板卡的 DA Channel 0 連續輸出。使用者也可以設定要將這組正弦波數值連續多次輸出，輸出幾組由使用者自行設定。輸出的時間從使用者選擇的一個或多個輸入通道擷取資料，使用者可以設定要擷取的速度與資料數。

參考「PCI-1202/1602/1800/1802 Hardware manual」第五章有更多詳細的內容。

※此函式輸出波形到 **analog output 0** 的同時，會從指定的通道擷取資料。

### ● 語法：

```
WORD P1202_M_FUN_3(WORD wDaNumber, WORD wDaWave, float
    fDaAmplitude, WORD wAdSampleRateDiv, WORD wAdNumber, WORD
    wChannelStatus[], WORD wAdConfig[], float fAdBuf[], float fLowAlarm,
    float fHighAlarm)
```

### ● 參數：

**wDaNumber**: [Input] 描述一組輸出的正弦波要用的數值點數。點數越多，波形會越像真正的類比輸出；但因輸出的點數多，輸出一個完整波的速度會變慢。

請參考 P1202\_M\_FUN\_1 的 Example 1

**wDaWave**: [Input] 輸出多少組的正弦波。

請參考 P1202\_M\_FUN\_1 的 Example 2

**fDaAmplitude**: [Input] 正弦波的振幅。

※fDaAmplitude value range = -10~10  
NOTE : the hardware J1 must select +/-10V

**wAdSampleRateDiv**: [Input] AD 取樣頻率 = 8M/ wAdSampleRateDiv (取樣數/秒)

**wAdNumber**: [Input] 讀取 AD 資料的筆數

**wAdChannel[]**: [Input] 陣列的索引值對應到通道號碼, 1=要擷取此通道資料,  
0=不擷取此通道資料

Example3:

wAdChannel[0] = 1 → 要擷取 Channel 0 的資料

wAdChannel[0]=0 → 不擷取 Channel 0 的資料

wAdChannel[2]=1 → 要擷取 Channel 2 的資料

**wAdConfig[]**:**[Input]** 配置碼，設定錯誤會量測到不正確的值請小心。

請參考 1.3 節

**fAdBuf[]**: **[Output]** 為一浮點數陣列。用來儲存 A/D 的資料

**fLowAlarm**: **[Input]**最低警戒值. 當 **fAdBuf[?]**小於 **fLowAlarm** → LowAlarm

**fHighAlarm**: **[Input]**最高警戒值. 當 **fAdBuf[?]**大於 **fHighAlarm** → HighAlarm

- 回傳值：

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

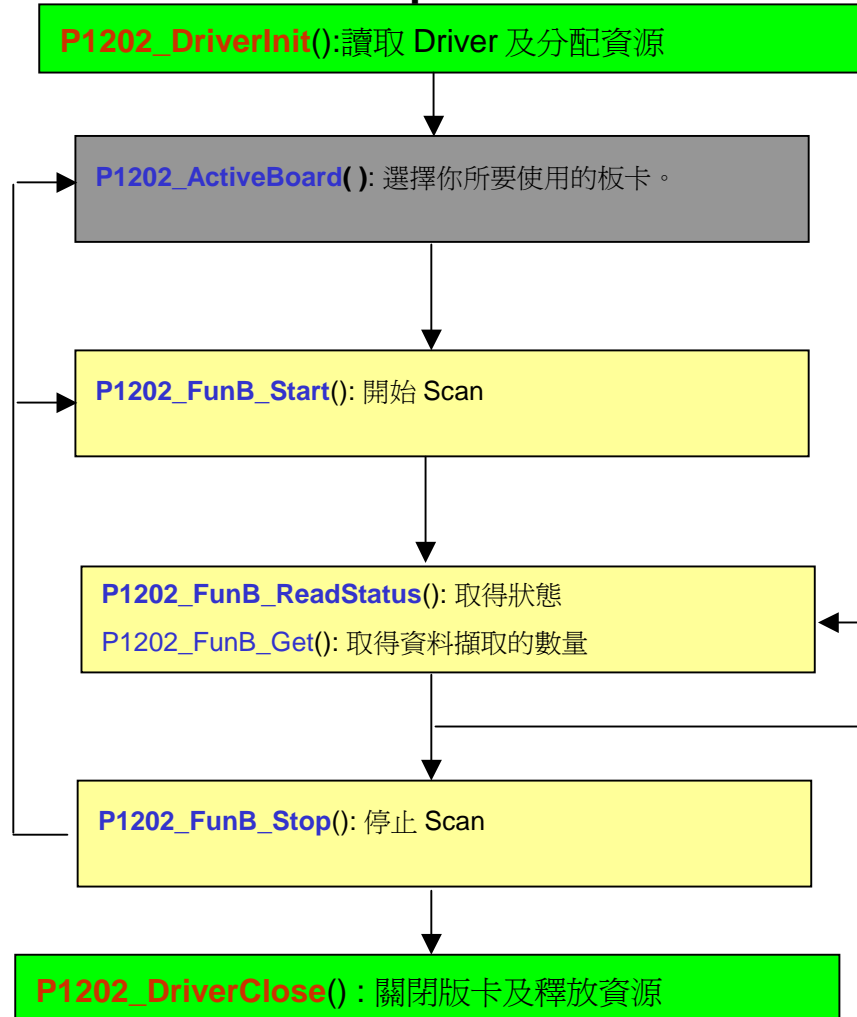
- 範例程式：

DEMO9.C

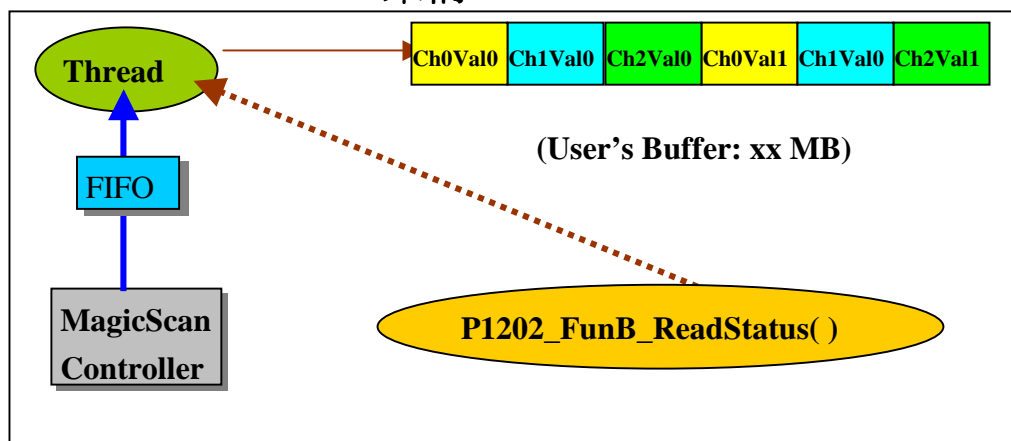
## 2.8 The Single Board Batch Capture

### 2.8.0 導讀

#### FunB Batch Capture Procedures



#### 架構



## 2.8.1 P1202\_FunB\_Start

電腦上只安裝一張板卡時，使用這個函式以連續的方式取值並將值儲存在自行分配的記憶體裡。詳細的運作方式可以參考 2.9.0 節的說明。

- **語法：**

```
WORD P1202_FunB_Start(WORD wClockDiv0, WORD wChannel0[],  
                      WORD wConfig0[], WORD *Buffer0, DWORD dwMaxCount0,  
                      SHORT nPriority);
```

- **參數：**

wClockDiv0: **[Input]** 取樣頻率。取樣頻率等於 8M/wClockDiv0。

wChannel0[]: **[Input]** (0=no scan, 1=scan) 設定索引值對應的通道是否加入掃描取值。

Example1:

wChannel0[0] = 1 → Scan Channel 0 for first board

wChannel0[0]=0 → No scan Channel 0 for first board

wChannel0[2]=1 → Scan Channel 2 for first board

wConfig0[]: **[Input]** 配置碼。請參考 1.3 節

\*Buffer0: **[Output]** 儲存擷取資料的記憶體

dwMaxCount0: **[Input]** 該張板卡 buffer 的資料長度

nPriority : **[Input]** 設定執行緒的優先權(Priority)。請參考 1.4 節。

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**

DEMO21.C

## 2.8.2 P1202\_FunB\_ReadStatus

讀取 batch capture process 的狀態.

- **語法：**

WORD P1202\_FunB\_ReadStatus( void );

- **參數：**

無參數

- **回傳值：**

0: 資料已準備好, 可以取回儲存或分析使用

1: 資料未準備好

- **範例程式：**

DEMO21.C

## 2.8.3 P1202\_FunB\_Stop

停止 batch capture process.

- **語法:**

word P1202\_FunB\_Stop(void);

- **參數:**

無參數

- **傳回值:**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式 :** DEMO21.C

---

## 2.8.4 P1202\_FunB\_Get

取得 FunB A/D 資料的擷取資料筆數。

- **語法:**

word P1202\_FunB\_Get(DWORD \*P0);

- **參數:**

\*P0: [Output] A/D 資料的擷取資料筆數。

- **回傳值:**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式 :** DEMO21.C



## 2.9 Multi-board Batch Capture

### (two boards operating simultaneously)

---

#### 2.9.0 P1202\_FunA\_Start

---

**電腦上安裝兩張板卡時**，使用這個函式可以一次規劃兩張板卡連續擷取資料並將值儲存在自行分配的記憶體裡。詳細的運作方式可以參考 2.9.0 節的說明。

- **語法：**

```
WORD P1202_FunA_Start(WORD wClockDiv0, WORD wChannel0[],  
                      WORD wConfig0[], WORD *Buffer0, DWORD dwMaxCount0,  
                      WORD wClockDiv1, WORD wChannel1[],  
                      WORD wConfig1[], WORD *Buffer1, DWORD dwMaxCount1,  
                      Short nPriority);
```

- **參數：**

wClockDiv0: **[Input]** 第一張板卡的取樣頻率。取樣頻率是  $8M/wClockDiv0$ 。

wChannel0[:**[Input]**](0=no scan, 1=scan) 設定第一張板卡的通道是否加入掃描取值：陣列的索引值對應到同號碼的通道，1 為加入掃描取值，0 則不對此通道取值。

Example1:

wChannel0[0] = 1 → **Scan** Channel 0 for first board

wChannel0[0]=0 → **No scan** Channel 0 for first board

wChannel0[2]=1 → **Scan** Channel 2 for first board

wConfig0[:**[Input]**] 設定第一張板卡的配置值。

請參考 1.3 節

\*Buffer0: **[Output]** 儲存第一張板卡的 A/D 資料的 buffer。

dwMaxCount0:**[Input]** 第一張板卡 buffer 的資料長度

wClockDiv1: **[Input]** 第二張板卡的取樣頻率。取樣頻率是  $8M/wClockDiv0$ 。

wChannel1[]:[Input] (0=no scan, 1=scan) 設定第一張板卡的通道是否加入掃描取值：陣列的索引值對應到同號碼的通道，1 為加入掃描取值，0 則不對此通道取值。

Example1:

wChannel1[0] = 1 → Scan Channel 0 for second board

wChannel1[0]=0 → No scan Channel 0 for second board

wChannel1[2]=1 → Scan Channel 2 for second board

wConfig1[]:設定第二張板卡的配置值。

請參考 1.3 節

\*Buffer1:[Output] 儲存第二張板卡的 A/D 資料的 buffer。

dwMaxCount1: [Input] 第二張板卡 buffer 的資料長度

nPriority:[Input] 設定執行緒的優先權(Priority)。請參考 1.4 節。

- 回傳值：

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- 範例程式：DEMO20.C

## 2.9.1 P1202\_FunA\_ReadStatus

用來取得 batch capture process 的狀態。

- 語法：

WORD P1202\_FunA\_ReadStatus( void );

- 參數：

無參數

- 回傳值：

0: 資料已準備好，可以取回儲存或分析使用

1: 資料未準備好。

- 範例程式：DEMO20.C

## 2.9.2 P1202\_FunA\_Stop

停止 batch capture 函式。

- **語法：**

word P1202\_FunA\_Stop(void);

- **參數：**

無參數

- **回傳值：**

無回傳值

- **範例程式：**

DEMO20.C

---

## 2.9.3 P1202\_FunA\_Get

取得 FunA A/D 資料的擷取筆數。

- **語法：**

word P1202\_FunA\_Get(DWORD \*P0, DWORD \*P1);

- **參數：**

\*P0:[Output] 第一張板卡有多少筆 A/D 資料被擷取。

\*P1:[Output] 第二張板卡有多少筆 A/D 資料被擷取。

- **回傳值：**

0：無錯誤

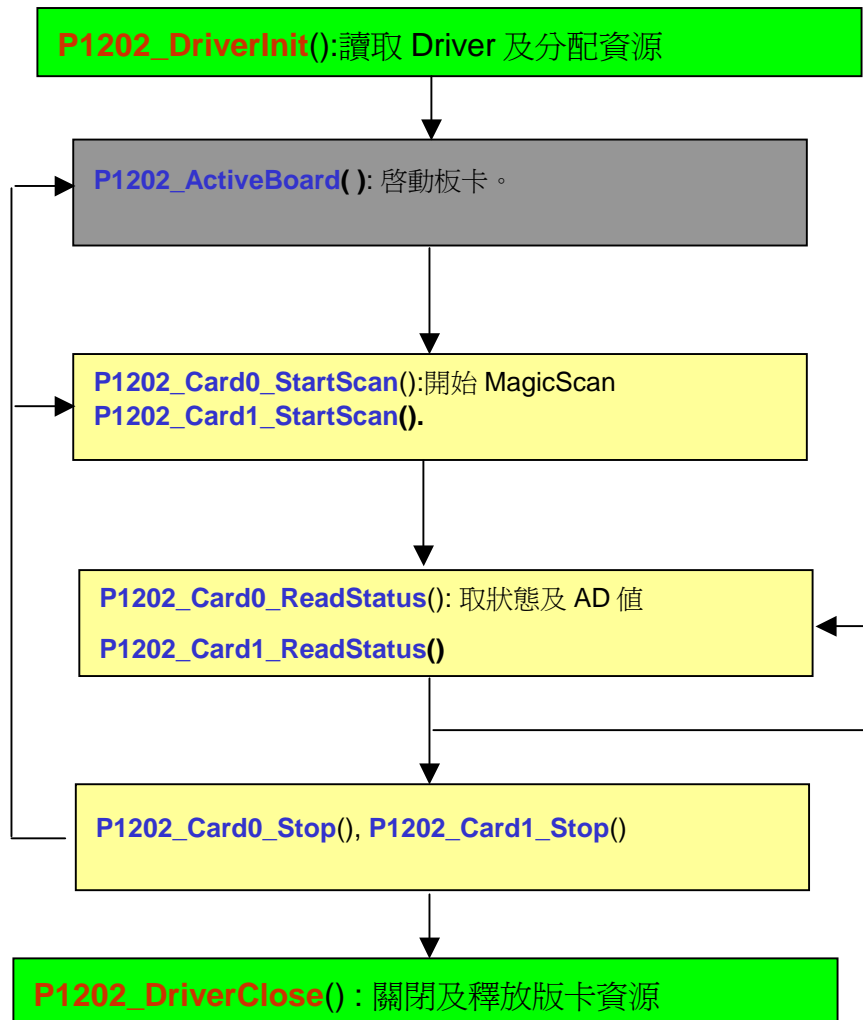
其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：** DEMO20.C

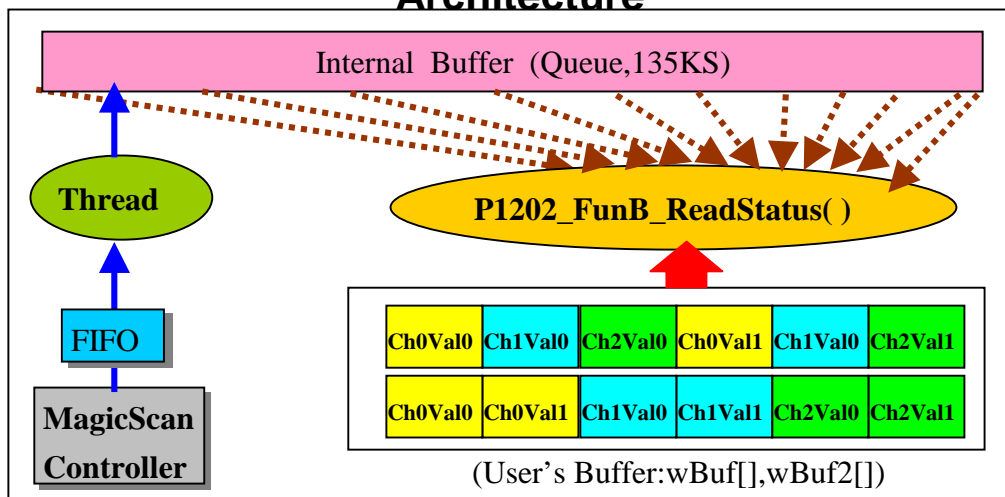
## 2.10 The Continuous Capture 函式集

### 2.10.0 導覽

#### Continuous Capture 流程



#### Architecture



## 2.10.1 P1202\_Card0\_StartScan

這個函式會使用第一張卡以掃描通道，連續取值的方式擷取 AD 值。建議使用在較低的速度及較長時間來使用。否則 Internal buffer 溢滿的機會將會增高。依各人的電腦狀況不同，建議取樣頻率最好在 40kHz 以下。請參考「PCI-1202/1602/180x Hardware User Manual」可以得到其他的參考資訊。

- **語法：**

WORD P1202\_Card0\_StartScan(WORD wSampleRateDiv, WORD wChannelStatus[], WORD wChanelConfig[], WORD wCycles);

- **參數：**

**wSampleRateDiv** :[Input] **AD 取樣頻率 = 8M/wSampleRateDiv.**

wSampleRateDiv=80 → 取樣頻率=8M/80=100K Hz

**wChannelStatus[]** :[Input] (0=no scan, 1=scan) 設定每一個通道是否加入掃描取值：陣列的索引值對應到同號碼的通道，1 為加入掃描取值，0 則不對此通道取值。

Example1:

wChannelStatus [0] = 1 → Scan Channel 0

wChannelStatus [0]=0 → No scan Channel 0

wChannelStatus [2]=1 → Scan Channel 2

**wChannelConfig[]** :[Input] 設定每個通道的配置值，陣列的索引值對應到同號碼的通道。例如 wChannelConfig[1]=0 ← 設定通道 1 的配置值為 0。

**wCycles**:[Input] 因為此函式是連續不停的擷取資料，由函式分次將資料讀入，所以需使用此參數設定一次資料讀入時要取得每一個通道上的資料數。這個次數沒有限制。一次取回的資料量=wCycles \* 加入掃描的通道數

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：** DEMO13.C

## 2.10.2 P1202\_Card0\_ReadStatus

這個函式用來取得連續擷取程序的狀態及檢查資料蒐集是否完成。在連續取值的過程中，不會阻止程式的運作。您必須使用這個回傳值，若為 0 表示資料是準備好的，你可以取回資料並繼續檢查程序的狀態，再取回下一次蒐集完的資料，直到長時間連續取值工作完成為止。

- **語法：**

```
P1202_Card0_ReadStatus(WORD wBuf[], WORD wBuf2[], DWORD *dwP1,
DWORD *dwP2, WORD *wStatus);
```

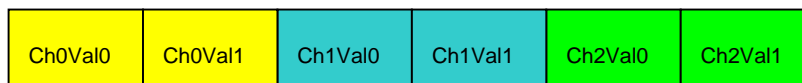
- **參數：**[output]

**wBuf[]** :[Output]儲存資料陣列，依照掃描順序儲存資料。

掃描順序為(012...N012...N.....012...N)的話，擷取資料儲存順序如



**wBuf2[]** :[Output] 儲存資料陣列。依照通道別將資料儲存如  
(0000.....11111.....22222....NNNNN....)



**\*dwP1** :[Input/Output] 保留給內部使用。

**\*dwP2** :[Input/Output] 保留給內部使用。

**\*wStatus** :[Output] 1=開始執行緒, 2=超過時間, 8=FIFO 溢滿, 0x80=結束執行緒,  
5=Buffer 已滿

- **回傳值：**

0: 資料已準備好

1: 無資料

2: FIFO 溢滿

3: 執行緒超過時間

4: Buffer 已滿

- **範例程式：**DEMO13.C

## 2.10.3 P1202\_Card0\_Stop

停止連續擷取資料的程序。

- **語法：**

`void P1202_Card0_Stop(void);`

- **參數：**

(None)

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：** DEMO13.C

## 2.10.4 P1202\_Card1\_StartScan

這個函式會使用第二張卡以連續取值的方式擷取 AD 值。建議使用在較低的速度及較長時間來使用。否則 Internal buffer 溢滿的機會將會增高。依各人的電腦狀況不同，建議取樣頻率最好在 40kHz 以下。請參考「PCI-1202/1602/180x Hardware User Manual」可以得到其他的參考資訊。

- **語法：**

WORD P1202\_Card1\_StartScan(WORD wSampleRateDiv, WORD wChannelStatus[], WORD wChanelConfig[], WORD wCount);

- **參數：**

**wSampleRateDiv** :[Input] **AD 取樣頻率 = 8M/wSampleRateDiv.**

wSampleRateDiv=80 → 取樣頻率=8M/80=100K Hz

**wChannelStatus[]** :[Input] (0=no scan, 1=scan) (0=no scan, 1=scan) 設定每一個通道是否加入掃描取值：陣列的索引值對應到同號碼的通道，1 為加入掃描取值，0 則不對此通道取值。

Example1:

wChannelStatus[0] = 1 → Scan Channel 0

wChannelStatus[0]=0 → No scan Channel 0

wChannelStatus[2]=1 → Scan Channel 2

**wChannelConfig[]** :[Input] 設定每個通道的配置值(1.3 節)，陣列的索引值對應到同號碼的通道。例如 wChannelConfig[1]=0 ← 設定通道 1 的配置值為 0。

**wCycles**:[Input] 因為此函式是連續不停的擷取資料，由函式分次將資料讀入，所以需使用此參數設定一次資料讀入時要取得每一個通道上的資料數。這個次數沒有限制。一次取回的資料量=wCycles \* 加入掃描的通道數

- **回傳值：**

0：無錯誤

其他：請參考 1.2 節：錯誤碼定義。

- **範例程式：**DEMO14.C



## 2.10.5 P1202\_Card1\_ReadStatus

這個函式用來取得連續擷取程序的狀態及檢查資料蒐集是否完成。在連續取值的過程中，不會阻止程式的運作。您必須使用這個回傳值，若為 0 表示資料是準備好的，你可以取回資料並繼續檢查程序的狀態，再取回下一次蒐集完的資料，直到長時間連續取值工作完成為止。

- **語法：**

```
P1202_Card1_ReadStatus(WORD wBuf[], WORD wBuf2[], DWORD *dwP1,
DWORD *dwP2, WORD *wStatus);
```

- **參數** :[output]

**wBuf[]** :[Output]陣列。用來存入掃描資料，順序為(012...N012...N.....012...N)

Ch0Val0	Ch1Val0	Ch2Val0	Ch0Val1	Ch1Val1	Ch2Val1
Ch0Val0	Ch0Val1	Ch1Val0	Ch1Val1	Ch2Val0	Ch2Val1

**wBuf2[]** :[Output]陣列。用來存入掃描資料，順序為  
(0000.....11111.....22222....NNNNN.....)

**\*dwP1** :[Input/Output] 保留給內部使用。

**\*dwP2** :[Input/Output] 保留給內部使用。

**\*wStatus** :[Output] 1=開始執行緒, 2=超過時間, 8=FIFO 溢滿, 0x80=結束執行緒,  
5=Buffer 已滿

- **回傳值：**

0: 資料已準備好

1: 無資料

2.:FIFO 溢滿

3: 執行緒超過時間

4: Buffer 已滿

- **範例程式：** DEMO14.C

## 2.10.6 P1202\_Card1\_Stop

停止連續擷取的程序

- **語法：**

`void P1202_Card1_Stop(void);`

- **參數：**

無參數

- **回傳值：**

無回傳值

- **範例程式：**

DEMO14.C

## 2.11 The Other Functions

---

### 2.11.0 P1202\_DelayUs

這個函式驅動板卡上的一個獨立的計時器，用來計算一小段時間或延遲一小段時間。它只能使用在啟動狀態中的板卡，請使用 P1202\_ActiveBoard(2.2.4 節) 啟動您要使用的板卡。

- **語法：**

word P1202\_DelayUs(WORD wDelayUs);

- **參數：**

wDelayUs : 延遲時間( $\mu$ s), 最大值為 8191  $\mu$ s

wDelayUs=1      → 延遲 1 us

wDelayUs=1000   → 延遲 1000 us = 1 ms

wDelayUs=8191   → 延遲 8191 us = 8.191 ms (最大的延遲時間)

wDelayUs=8192   → 無效 delay (將會回傳錯誤碼)

- **回傳值：**

0 : 無錯誤

其他 : 請參考 1.2 節 : 錯誤碼定義。

- **範例程式：** DEMO1.C

- **Long Time Delay：**

```
WORD DelayMs(WORD wDelayMs) // maximum delay=4294967.295 sec
{
    WORD wDelay,wRetVal

    wRetVal=0;
    for (wDelay=0; wDelay<wDelayMs; wDelay++)
        wRetVal+=P1202_DelayUs(1000);
    return(wRetVal);
}
```

## 3. 附錄

### ※ 註 1:

#### [MagicScan controller]

這是一種硬體掃描的機制，程式設定掃描的通道順序與數入範圍的配置值，之後啟動硬體掃描程序。一個硬體的 **MagicScan Controller** 就會依照指定的頻率，順序與輸入訊號範圍配置值，在指定通道間切換，取值，並將取得的 **AD** 值放到 **FIFO**，**driver** 同時將 **FIFO** 的資料搬到 **Buffer** 中，使用者的程式只須使用函式將 **Buffer** 內的資料取出。這樣的處理方式可以減少 **CPU** 的處理時間。

### ※ 註 2:

#### [FIFO Overflow]

當 **FIFO** 空間已經全滿時，下一筆進入的資料將覆蓋第一筆資料而造成資料遺失稱為 **FIFO Overflow**。

要減少 **FIFO Overflow** 的發生，建議停止使用其他的程式像是 IE 或是防毒軟體；或是選擇 **FIFO** 容量更大的板卡像是 **8K** 的版本，通常這樣可以有效降低 **FIFO Overflow** 發生的機率

## 4. 範例程式

PCI-1202 提供 20 個以上的範例程式, 每個範例功能條列如下:

- demo1: one board, D/I/O test, D/A test, A/D polling & pacer trigger test, general test
- demo2: two board, same as demo1
- demo3: one board, all 32 channels of A/D by software trigger(by polling)
- demo4: two board, same as demo3
- demo5: one board, M\_function\_1 demo
- demo6: two board, same as demo5
- demo7: one board, M\_function\_2 demo
- demo8: two board, same as demo7
- demo9: one board, M\_function\_3 demo
- demo10: two board, same as demo9
- demo11: one board, MagicScan demo
- demo12: two board, same as demo11
- demo13: one board, continuous capture demo
- demo14: two board, continuous capture demo (Windows 95/NT only)
- demo15: all installed board, D/I/O test for board number identification
- demo16: one board, performance evaluation demo
- demo17: one board, MagicScan demo, scan sequence: 4→3→5
- demo18: one board, MagicScan demo, scan 32 channel, show channel 0/1/15/16/17
- demo19: one board, A/D calibration.
- demo20: two board, P1202\_FUNA, continuous capture demo
- demo21: single board, P1202\_FUNB, continuous capture demo
- demo23: single board, post-trigger demo
- demo24: single board, pre-trigger demo
- demo25: single board, middle-trigger demo

Refer to the company floppy disk for details.