

# PMDK 之 PC 端基本函式手冊

(Version 1.1)



**ICP DAS CO., LTD.**

泓格科技股份有限公司

---

## Warranty

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

## Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

## Trademark

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

## License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

# Contents

1. 前言 .....	4
2. 系統函式 .....	5
2.1 作業系統相關的設定 .....	5
2.1.1 <i>pmdk_open</i> .....	5
2.1.2 <i>pmdk_close</i> .....	6
2.1.3 <i>pmdk_dsp_reset</i> .....	6
2.1.4 <i>pmdk_dip_switch_get</i> .....	7
2.1.5 <i>pmdk_int_event_get</i> .....	7
2.1.6 <i>pmdk_int_code_get</i> .....	8
2.1.7 <i>pmdk_board_status_get</i> .....	8
2.1.8 <i>pmdk_cpld_version_read</i> .....	9
2.1.9 <i>pmdk_cpld_offset_write</i> .....	10
2.1.10 <i>pmdk_cpld_offset_read</i> .....	10
2.2 DPRAM 讀寫函式 .....	11
2.2.1 <i>pmdk_dpram_offset_write</i> .....	12
2.2.2 <i>pmdk_dpram_offset_read</i> .....	12
2.2.3 <i>pmdk_dpram_offset_write_U32</i> .....	13
2.2.4 <i>pmdk_dpram_offset_read_U32</i> .....	13
2.2.5 <i>pmdk_dpram_entry_write</i> .....	14
2.2.6 <i>pmdk_dpram_entry_read</i> .....	14
回傳值 (錯誤碼) .....	15

## 1. 前言

這是PMDK專屬的軟體，包括有Microsoft® Windows 2000、Windows XP與Windows 7的WDM (Windows Driver Model) 驅動程式、ANSI-C類型的程式庫和相關的範例程式。

在程式庫內對每一張PMDK以一個獨特的編號(Card ID)作為辨識的基礎，而這個編號則透過PMDK板卡上的指撥開關(Dip-Switch)設定。這樣的設計可以用來辨識系統上多張PMDK，讓系統商可以透過Card ID 搭配電控系統，而不需要再擔心作業系統所辨識的板卡順序是否與所規劃的一致。所以Card ID必須唯一且事先設定。

這個程式庫提供了給Microsoft® 的 Visual Studio (Visual C++)的函式供客戶來開發應用。

這份文件對軟體程式庫提供了詳細的說明，包括有函式的宣告、參數與回傳碼的定義。我們將所有的函式分類並分別敘述於第二章。

## 2. 系統函式

### 2.1 作業系統相關的設定

這章的函式提供了與作業系統的介面。系統程式啟動時掃描在作業系統內所有可以使用的PMDK板卡並建立其裝置的Handle，並將用以辨識這些板卡的Card ID與裝置的Handle在系統內建立對應表。透過Handle可以配合程式庫中其他的函式來開啟與使用相對應的PMDK板卡。

**注意：使用PMDK前務必要事先設定好ID，並確他們認沒有重複。**

#### 2.1.1 pmdk\_open

```
short pmdk_open(U16 bCardID, HANDLE *pHandle)
```

**功能：**

這個函式會依據所輸入的Card ID取得相對應的PMDK板卡的Handle。如果這個函式正常地執行，則開啟此板卡的程式會擁有這張板卡的使用權，直到同一個程式呼叫pmdk\_close()。在開始使用其他馬達控制函式之前，相對於Card ID的PMDK板卡需要先用pmdk\_open()開啟。

**參數：**

bCardID：藉由PMDK板卡上的Dip-Switch所設定的Card ID。

pHandle：PMDK會以此Handle值當作其他函式的傳入值。

**回傳值：**

請參考附錄。

## 2.1.2 pmdk\_close

short pmdk\_close(HANDLE hCard)

### 功能：

這個函式會依據所輸入的Handle關閉相對應的PMDK板卡。在呼叫此一函式後，PMDK板卡會重新釋放使用權給其他的程序使用。

### 參數：

hCard：傳入的Handle值。

### 回傳值：

請參考附錄。

## 2.1.3 pmdk\_dsp\_reset

short pmdk\_dsp\_reset(HANDLE hCard)

### 功能：

這個函式會重置PMDK內部的DSP晶片，並將基本的暫存器重新設回預設值。一旦呼叫這個函式，則之前的所有設定都會被抹除。同時這個函式也會停止目前PMDK板卡的運動。重置後，DSP重新取出並執行Flash內的程式需要一些時間，所以最好在程式中用迴圈檢查PMDK是否ready，ready後才可以下命令。請參考函式pmdk\_board\_status\_get()。

### 參數：

hCard：傳入的Handle值。

### 回傳值：

請參考附錄。

## 2.1.4 pmdk\_dip\_switch\_get

short pmdk\_dip\_switch\_get (HANDLE hCard, U16\* pCardID)

### 功能：

這個函式用來取得PMDK板卡的指撥ID號碼。由於驅動程式已經將Handle與CardID設成同一數值，所以這函式已經不再具有作用。

### 參數：

hCard：傳入的Handle值。

pCardID：指標指向PMDK的Card ID參數。

### 回傳值：

請參考附錄。

## 2.1.5 pmdk\_int\_event\_get

short pmdk\_int\_event\_get(HANDLE hCard, HANDLE\* phEvent)

### 功能：

這個函式用來獲得目前PMDK卡產生的event號碼。在Windows程式中可以等待此event來驅動某一程序的執行。

### 參數：

hCard：傳入的Handle值。

phEvent：指標變數指到 event 號碼。

### 回傳值：

請參考附錄。

## 2.1.6 pmdk\_int\_code\_get

short pmdk\_int\_code\_get (HANDLE hCard, I16\* pINTCode)

### 功能：

這個函式用來獲得DSP透過DPRAM送來的中斷值(int code)。這是DSP程式寫到DPRAM對PC產生中斷所傳遞的值。客戶可以定義這些值並用於PC與DSP的溝通。

### 參數：

hCard：傳入的Handle值。

pINTCode：指標變數指到 int code。

### 回傳值：

請參考附錄。

## 2.1.7 pmdk\_board\_status\_get

short pmdk\_board\_status\_get (HANDLE hCard, U16\* pStatus)

### 功能：

這個函式用來獲得目前的PMDK卡ready狀態。

### 參數：

hCard：傳入的Handle值。

pStatus：用以儲存目前PMDK卡ready狀態的指標參數。

<b>Bit</b>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Definition</b>	READY	-	-	-	-	-	-	-
<b>Bit</b>	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
<b>Definition</b>	-	-	-	-	-	-	-	-

### 回傳值：

請參考附錄。



**範例：**

```
U16 state, PMDK_ready;  
HANDLE hCard  
PMDK_ready = 0;  
do  
{  
    pmdk_board_status_get( hCard, &state );  
    PMDK_ready = (state & 0x80)>>7;  
    Sleep(10);  
} while (PMDK_ready == 0);
```

## 2.1.8 pmdk\_cpld\_version\_read

short pmdk\_cpld\_version\_read (HANDLE hCard, U16\* pPLDVersion)

**功能：**

這個函式用來獲得目前的CPLD版本。

**參數：**

hCard：傳入的Handle值。

pPLDVersion：用以儲存 PMDK CPLD版本的指標參數。

**回傳值：**

請參考附錄。

## 2.1.9 pmdk\_cpld\_offset\_write

short pmdk\_cpld\_offset\_write (HANDLE hCard, U32 dwOffset, U16 wData)

**功能：**

這個函式用來寫入16-bit資料到CPLD的暫存器。

**參數：**

hCard：傳入的Handle值。

dwOffset：這是CPLD暫存器位址(offset)。

wData: 要寫入的資料。

**回傳值：**

請參考附錄。

## 2.1.10 pmdk\_cpld\_offset\_read

short pmdk\_cpld\_offset\_read (HANDLE hCard, U32 dwOffset, U16\* pData)

**功能：**

這個函式用來由CPLD的暫存器中讀出16-bit資料。

**參數：**

hCard：傳入的Handle值。

dwOffset：這是CPLD暫存器位址(offset)。

pData: 指標參數指到被讀出的資料。

**回傳值：**

請參考附錄。

## 2.2 DPRAM讀寫函式

DPRAM 的大小為 0x800 (2048) bytes，其區塊(Block)偏移位址如下表所定義。存取 DPRAM 的基本單位為 WORD (two bytes)且起始位址必須為偶數偏移位址。DPRAM 每一個區塊包含 32 bytes (16 words)總共 64 個區塊如下表所示。當使用 API 去讀寫 DPRAM 時，偏移位址的範圍不能超過兩個區塊。例如 `pmdk_dpram_offset_write_U32()` 的 `dwOffset` 值不能為 0x01E，因為此函數需寫 4 個 bytes 會超過區塊 0 (Block 0)的範圍。相反地，可以為 0x00E，因為所有記憶體位址皆位於區塊 0 之內。當使用 `pmdk_dpram_entry_write()` 去寫資料，起始位址定義為區塊的起始位址(0x20 的整數倍)，資料長度必須為偶數且不能超過 16 個 words (0x20 bytes)。

位址 0x7FC-0x7FD 與 0x7FE - 0x7FF 有特殊的動作，當 PC 端寫資料到 0x7FE - 0x7FF 會對 DSP 端產生中斷。相對地，當 DSP 端寫資料到 0x7FC-0x7FD 會對 PC 端產生中斷。用戶應該好好利用此特點去做 PC 與 DSP 之間的高速通訊。

**注意：有關 Block 的限制只有在 PC 端，DSP 端無此限制。**

表 1: DPRAM 的區塊偏移位址

<b>Block</b> <b>0</b>	0x000 - 0x001 (2 bytes)	0x002 - 0x003 (2 bytes)	0x004 - 0x005 (2 bytes)	0x006 - 0x007 (2 bytes)	0x008 - 0x009 (2 bytes)	0x00A - 0x00B (2 bytes)	0x00C - 0x00D (2 bytes)	0x00E - 0x00F (2 bytes)
	0x010 - 0x011 (2 bytes)	0x012 - 0x013 (2 bytes)	0x014 - 0x015 (2 bytes)	0x016 - 0x017 (2 bytes)	0x018 - 0x019 (2 bytes)	0x01A - 0x01B (2 bytes)	0x01C - 0x01D (2 bytes)	0x01E - 0x01F (2 bytes)
<b>Block</b> <b>1</b>	0x020 - 0x021 (2 bytes)	0x022 - 0x023 (2 bytes)	0x024 - 0x025 (2 bytes)	0x026 - 0x027 (2 bytes)	0x028 - 0x029 (2 bytes)	0x02A - 0x02B (2 bytes)	0x02C - 0x02D (2 bytes)	0x02E - 0x02F (2 bytes)
	0x030 - 0x031 (2 bytes)	0x032 - 0x033 (2 bytes)	0x034 - 0x035 (2 bytes)	0x036 - 0x037 (2 bytes)	0x038 - 0x039 (2 bytes)	0x03A - 0x03B (2 bytes)	0x03C - 0x03D (2 bytes)	0x03E - 0x03F (2 bytes)
...	...							
<b>Block</b> <b>63</b>	0x7E0 - 0x7E1 (2 bytes)	0x7E2 - 0x7E3 (2 bytes)	0x7E4 - 0x7E5 (2 bytes)	0x7E6 - 0x7E7 (2 bytes)	0x7E8 - 0x7E9 (2 bytes)	0x7EA - 0x7EB (2 bytes)	0x7EC - 0x7ED (2 bytes)	0x7EE - 0x7EF (2 bytes)
	0x7F0 - 0x7F1 (2 bytes)	0x7F2 - 0x7F3 (2 bytes)	0x7F4 - 0x7F5 (2 bytes)	0x7F6 - 0x7F7 (2 bytes)	0x7F8 - 0x7F9 (2 bytes)	0x7FA - 0x7FB (2 bytes)	0x7FC - 0x7FD (2 bytes)	0x7FE - 0x7FF (2 bytes)

## 2.2.1 pmdk\_dpram\_offset\_write

short pmdk\_dpram\_offset\_write(HANDLE hCard, U32 dwOffset, U16 wData)

**功能：**

寫16-bit資料到DPRAM指定位置上。

**參數：**

hCard：傳入的Handle值。

dwOffset: DPRAM的指定位置(offset)，必須為偶數。

wData: 將寫入的16-bit資料。

**回傳值：**

請參考附錄。

## 2.2.2 pmdk\_dpram\_offset\_read

short pmdk\_dpram\_offset\_read(HANDLE hCard, U32 dwOffset, U16\* pData)

**功能：**

在DPRAM指定位置上讀出16-bit資料。

**參數：**

hCard：傳入的Handle值。

dwOffset: DPRAM的指定位置(offset)，必須為偶數。

pData: 指標指向讀出的16-bit資料。

**回傳值：**

請參考附錄。

## 2.2.3 pmdk\_dpram\_offset\_write\_U32

short pmdk\_dpram\_offset\_write\_U32(HANDLE hCard, U32 dwOffset, U32 dwData)

### 功能：

寫32-bit資料到DPRAM指定位置上。

### 參數：

hCard：傳入的Handle值。

dwOffset: DPRAM的指定位置(offset)，必須為偶數且值不能為0x01E，因為此函數需寫4個bytes會超過區塊0的範圍。相反地可以為0x00E，因為所有記憶體位址皆位於區塊0之內(參考表1)。

dwData: 將寫入的32-bit資料。

### 回傳值：

請參考附錄。

## 2.2.4 pmdk\_dpram\_offset\_read\_U32

short pmdk\_dpram\_offset\_read\_U32(HANDLE hCard, U32 dwOffset, U32\* pData)

### 功能：

在DPRAM指定位置上讀出32-bit資料。

### 參數：

hCard：傳入的Handle值。

dwOffset: DPRAM的指定位置(offset)，必須為偶數且值不能為0x01E，因為此函數需寫4個bytes會超過區塊0的範圍。相反地可以為0x00E，因為所有記憶體位址皆位於區塊0之內(參考表1)。

pData: 指標指向讀出的32-bit資料。

### 回傳值：

請參考附錄。

## 2.2.5 pmdk\_dpram\_entry\_write

```
short pmdk_dpram_entry_write(HANDLE hCard, U32 dwOffset, U16 wDataLength, U16* pData)
```

### 功能：

在DPRAM指定位置開始寫入連續16-bit資料陣列。

### 參數：

hCard：傳入的Handle值。

dwOffset: DPRAM的指定位置(offset)，必須為0x20的整數倍。

wDataLength: 將寫入的資料陣列的長度，其值不能超過16。

pData: 資料陣列的起始位置。

### 回傳值：

請參考附錄。

## 2.2.6 pmdk\_dpram\_entry\_read

```
short pmdk_dpram_entry_read(HANDLE hCard, U32 dwOffset, U16 wDataLength, U16* pData)
```

### 功能：

在DPRAM指定位置開始連續讀出16-bit資料陣列。

### 參數：

hCard：傳入的Handle值。

dwOffset: DPRAM的指定位置(offset)，必須為0x20的整數倍。

wDataLength: 將讀出的資料陣列的長度，其值不能超過16。

pData: 資料陣列的起始位置。

### 回傳值：

請參考附錄。

## 回傳值（錯誤碼）

所有回傳的錯誤碼定義：

PMDK_ERROR_SUCCESS	0
PMDK_ERROR_SCAN_INDEX	-1
PMDK_ERROR_CLOSE_DEVICE	-2
PMDK_ERROR_INVALID_HANDLE	-3
PMDK_ERROR_DEV_IO_CONTROL	-4
PMDK_ERROR_EVENT_CREATE	-5
PMDK_ERROR_INT_EVENT_ATTACH	-6
PMDK_ERROR_INT_EVENT_DETACH	-7
PMDK_ERROR_INVALID_EVENT_HANDLE	-8
PMDK_ERROR_INVALID_DATA_OFFSET	-9
PMDK_ERROR_INVALID_DATA_LENGTH	-10
PMDK_ERROR_DWORD_CROSS_PAGE	-11