



EDIO-800

硬體使用手冊

版本：V.1.1.1

日期：2008.07

<http://www.epcio.com.tw>



目 錄

第一章 ...概論.....	4
1-1 功能簡介.....	4
1-2 規格說明.....	4
1.3 系統連接圖.....	6
第二章硬體安裝說明	8
2-1 安裝條件	8
2-2 組態設定	8
2-3 連接器接腳定義	11
第三章軟體安裝說明	13
3-1 安裝程式	13
3-2 驅動程式庫	14
3-2.1 有關係統界面之驅動程式	14
ISA Interface Control (EPCIO IO Page 0)	14
EDIO_init()	14
EDIO_close()	15
EDIO_ISA_SetBase()	15
EDIO_ResetModule()	15
EDIO_ISA_GetIntSource()	16
EDIO_ISA_SetWaitState()	16
EDIO_ISA_SetIrqNo()	17
EDIO_ISA_ResetIrqNo()	17
EDIO_ISA_SetIntPeriod()	17
EDIO_ISA_SetActivePage()	18
EDIO_ISA_GetActivePage()	18
3-2.2 有關遠端輸出入控制驅動程式	19
Remote Digital IO (EPCIO IO Page 5、6)	19
EDIO_RIO0_GetIntCondition()	19
EDIO_RIO1_GetIntCondition()	20
EDIO_RIO_GetInputValue()	21
EDIO_RIO_SetOutputValue()	22
EPCIO_RIO_GetTransStatus()	22
EDIO_RIO_GetMasterStatus()	23



EDIO_RIO_GetSlaveStatus()	23
EDIO_RIO_SetClockDivider()	24
EDIO_RIO_SetIntType()	24
EDIO_RIO_EnableInputInt()	25
EDIO_RIO_DisableInputInt()	25
EDIO_RIO_SetTransError()	26
EDIO_RIO_EnableSetControl()	26
EDIO_RIO_DisableSetControl()	27
EDIO_RIO_EnableSlaveControl()	27
EDIO_RIO_DisableSlaveControl()	27
EDIO_RIO_EnableTransInt()	28
EDIO_RIO_DisableTransInt()	28
3-2.3 有關近端輸出入控制驅動程式	29
Local IO Control (EPCIO IO Page 8)	29
EDIO_LIO_GetIntCondition()	29
EDIO_LIO_GetLdiInput()	30
EDIO_LIO_GetDfiInput()	30
EDIO_LIO_SetLdoOutput()	31
EDIO_LIO_EnableLdoOutput()	31
EDIO_LIO_DisableLdoOutput()	31
EDIO_LDI_SetIntType()	32
EDIO_DFI_SetIntType()	32
EDIO_LIO_EnableLdiInt()	33
EDIO_LIO_DisableLdiInt()	33
EDIO_LIO_EnableDfiInt()	34
EDIO_LIO_DisableDfiInt()	34
3-2.4 有關計時器/看門狗控制驅動程式	35
EDIO_LIO_SetTimer()	35
EDIO_LIO_EnableTimer()	35
EDIO_LIO_DisableTimer()	35
EDIO_LIO_EnableTimerInt()	36
EDIO_LIO_DisableTimerInt()	36
EDIO_LIO_SetWdogTimer()	36
EDIO_LIO_EnableWdogTimer()	36
EDIO_LIO_DisableWdogTimer()	37
EDIO_LIO_SetWdogReset()	37



EDIO_LIO_RefreshWdogTimer().....	37
3-3 範例程式	38
3-3.1 遠端輸出入範例程式	38
Example of Remote Input / Output Control.....	38
3-3.2 可規劃近端輸出入點範例程式	40
Example of Local Input / Output Control.....	40
3-3.3 計時器/看門狗範例程式	42
Example of Timer / Watch Dog Control	42
3-4 測試程式	44
附錄.....	48
IO Port Definition	48
Page 0 : Bus interface control	48
Page 5 : First set remote Digital IO	51
Page 6 : Second set remote Digital IO	54
Page 8 : LIO Control.....	55

第一章 概論

EDIO-800應用工研院機械所開發的EDIO ASIC作專屬的數位輸出入控制，透過專利的省配線化設計，利用串列通信，此板共可控制多達6組的遠端子板(EDIO-S001, 64IN/64OUT)，另外有56點具TTL準位的近端輸出入點，使系統的設計更有彈性，由於大幅簡化了配線及降低控制系統的複雜度，因此可以減少產品生產及維護的成本，提高系統的可靠度及維護性。

1-1 功能簡介

- 具備 ISA-Bus 界面
- 串列通信介面，省配線連接，總共824點I/O，其中
 - ◆ 384 點遠端輸出控制
 - ◆ 384 點遠端輸入控制
 - ◆ 28 點近端輸出入控制（可規劃）
 - ◆ 28 點近端輸入控制
- 24 bits 計時器（具中斷功能）
- 16 bits 看門狗計時器（具重置功能）

1-2 規格說明

- Size : 185mm x 122mm
- System Clock : 40MHz
- 板基址設定範圍 : 200H~3FFH
 - 設定方式 : 4 Bit DIP Switch (S1)
 - 佔用IO位址空間 : 20 H
- 可設定IRQ : 3,5,7,10,11,12,15
 - 設定方式 : 軟體規劃
- Data Bus格式 : ISA 16 Bit
- 6組 D-sub 9 pin 接頭，用來連接最多6個遠端模組（EDIO-S001）
80KHz~2.5MHz - 5 Meter in 40MHz System Clock

遠端I/O點硬體反應時間：設定1 MHz傳輸頻率時，最小0.1ms(只使用1組EDIO-S001)，最大0.3ms(使用6組EDIO-S001)

每組遠端模組前4點可設定為具中斷輸入之功能

傳輸具備失效偵測，含 CRC 碼檢測及 ECHO 碼檢測

具備傳輸失效中斷功能

- 計時器(Timer)設定範圍：< 420ms，設定最小區間 25ns

24 bits，具中斷功能

- 看門狗計時器設定範圍(16 bits)：< 27K sec，設定最小區間 25ns

reset duration (24bits)：< 420 ms，設定最小區間 25ns

- 近端專屬輸入點：28點

前7點具中斷功能

TTL輸入相容

最大輸入電壓：5.5v

與系統電位共地（無隔離）

- 近端輸出入點：28點 (7sets * 4)

設定方式：由DIP Switch 設定輸出或輸入 (S2,S3)

以一個nibble (4 bits)為單位

前8點具中斷功能

TTL輸出入相容

最大輸入電壓：5.5v

與系統電位共地（無隔離）

- 工作環境溫度 0°C ~ 55°C

1.3 系統連接圖

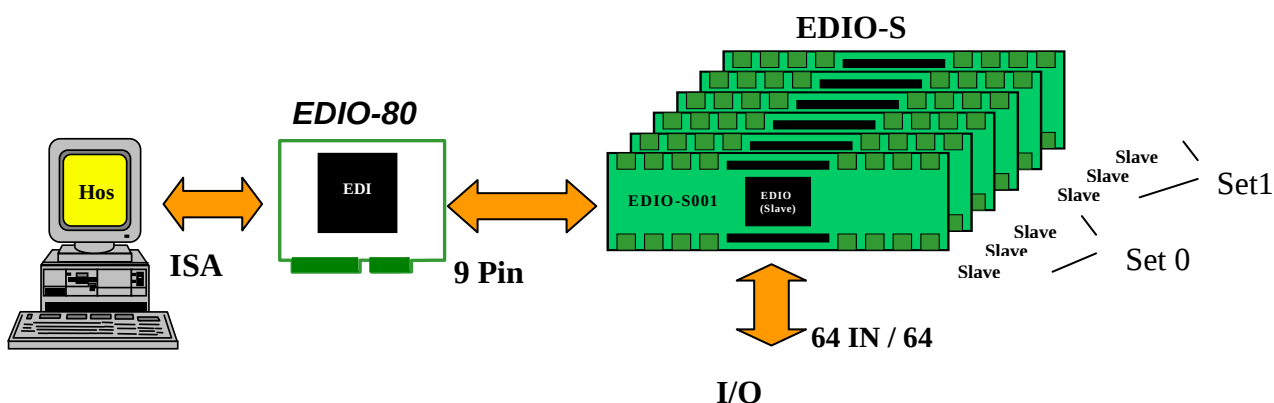


圖 1. EDIO 輸出入控制系統連接圖

圖 1 為 EDIO 系統連接圖，Host PC 透過 ISA Bus 與 EDIO-800 控制母板做資料的存取，包括輸入點的讀取，輸出點的控制，及中斷訊號處理。圖 2 所示為 EDIO-800 控制板方塊圖，圖上的 EDIO-832-01 ASIC (Master Mode) 負責在固定的時間內把遠端 IO 點的狀態做讀取與更新的工作，其中輸出部分先經過驅動電路轉成差動訊號後由 9 Pin 連接器輸出至遠端模組，輸入部分則接收由連接器輸入之差動訊號，經光耦合器後送至 ASIC 輸入，圖 3 為 EDIO-S001 功能方塊圖，EDIO-832-01 ASIC (Slave Mode) 負責接受來自於 EDIO-800 的資料並輸出及讀回實際 IO 接點狀態。

整個架構採省配線化，遠端控制模式，所有控制電路整合入 ASIC 內部，以提高產品可靠性及穩定性。

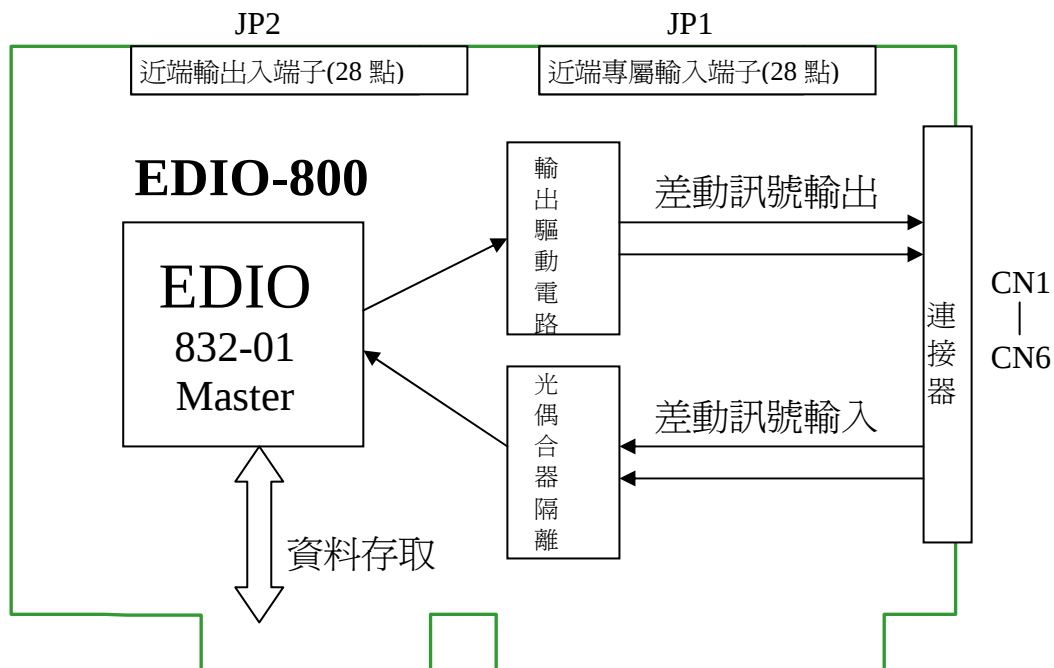


圖 2 EDIO-800 功能方塊圖

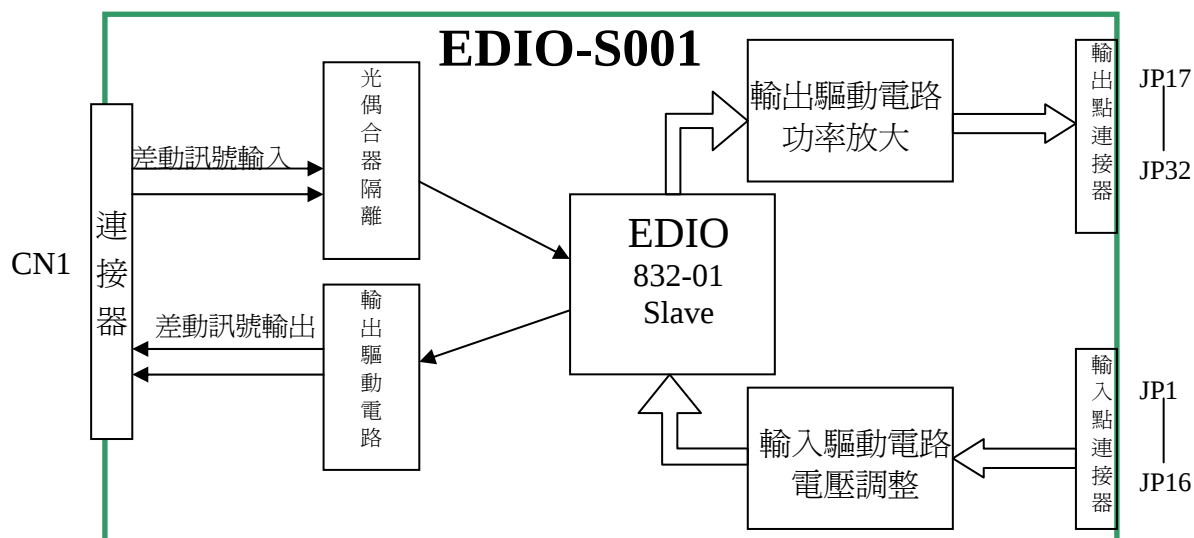


圖 3 EDIO-S001 功能方塊圖

第二章 硬體安裝說明

2-1 安裝條件

EDIO-800 為一片具備 ISA Bus 界面，且 Data Bus 為 16 Bit 之控制卡，控制卡上佔用 20H 個 IO 位置，因此 IPC 上至少必須具備一組 PC AT 界面之擴充槽，且提供連續 IO 位址空間 20H，若使用的 IO 點具備中斷功能，必須提供一組中斷資源(IRQ)供 EDIO 模組使用。

2-2 組態設定

EDIO-800 控制模組在使用上必須設定 IO 存取的基本位址，設定方式為透過板上的一 4 Bit DIP Switch (圖 4 -S1)，Switch 設定與 IO 相對位址請參考表 1，出廠時內定值為 240H。若插上 EDIO-800 控制模組後發現電腦無法正常開機，可能是 IO 位址與其他界面卡衝突，請更改 IO 位址後重試。若應用 EDIO-800 需要使用到中斷功能，設定方式請參考軟體組態設定說明(第 3-2.1 節)。

表 1. EDIO-800 控制卡板基址設定表

S1					S1				
1	2	3	4	板基址	1	2	3	4	板基址
ON	ON	ON	ON	200H	OFF	ON	ON	ON	300H
ON	ON	ON	OFF	220H	OFF	ON	ON	OFF	320H
ON	ON	OFF	ON	240H	OFF	ON	OFF	ON	340H
ON	ON	OFF	OFF	260H	OFF	ON	OFF	OFF	360H
ON	OFF	ON	ON	280H	OFF	OFF	ON	ON	380H
ON	OFF	ON	OFF	2A0H	OFF	OFF	ON	OFF	3A0H
ON	OFF	OFF	ON	2C0H	OFF	OFF	OFF	ON	3C0H
ON	OFF	OFF	OFF	2E0H	OFF	OFF	OFF	OFF	3E0H

下圖 4 為 IC 零件相對位置圖。

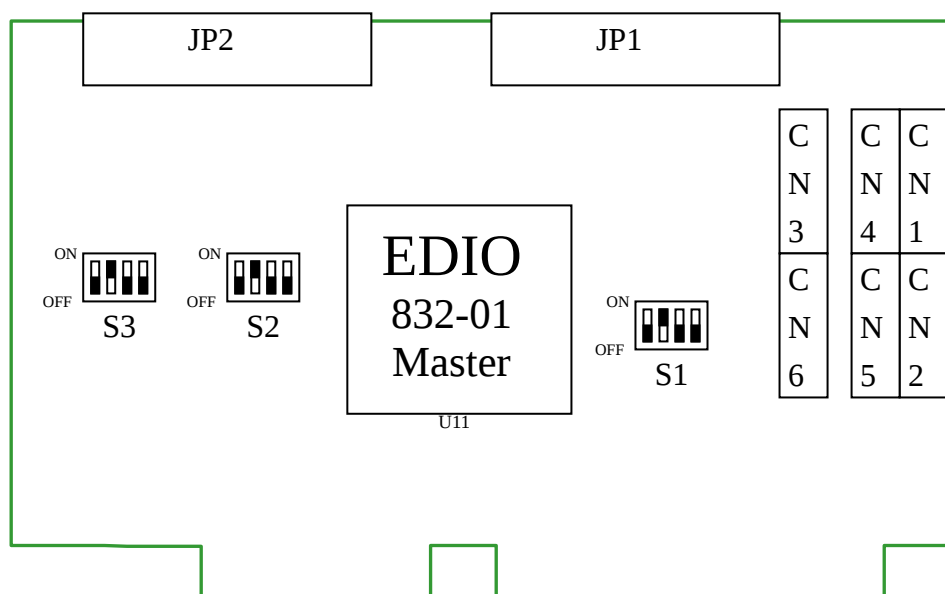


圖 4. EDIO-800 板上元件相對圖

圖 4 中 CN 1~6 為連接遠端控制模組連接器 (CN3,CN6 尚須透過 10 pin 牛角對 9 pin D-sub 的轉接線)，透過 9 Pin D-型連接線可與圖 5 中左側 D-型接頭連接，一片 EDIO-800 可與 6 片 EDIO-S001 連接，每片具有 64 點輸入及 64 點輸出，最多可達 768 點 IO 控制 (請參考 **EDIO-S001 使用手冊**)，連接器設定請參考軟體組態設定(第 3-3.2 節)。JP1 為 28 點專屬輸入點，JP2 為 28 點可規劃的輸出入接點，輸出入的規劃以每 4 點為一組，設定方式請參考軟體規劃(第 3-3.3 節)。S1 是 IO 位址設定開關，S2,S3 用來搭配軟體的設定可以規劃 JP2 的輸出或輸入 (見表 3)。

表 2. CN 1-6 的軟體定義：

CN1	SET0—SLAVE0
CN2	SET0—SLAVE1
CN3	SET0—SLAVE2
CN4	SET1—SLAVE0
CN5	SET1—SLAVE1
CN6	SET1—SLAVE2



表 3. S2,S3--可規劃的輸出入點設定：

S3					S2				
1	2	3	4		1	2	3	4	
x	SET6	SET5	SET4		SET3	SET2	SET1	SET0	
x	OUT	OUT	OUT	ON	OUT	OUT	OUT	OUT	ON
x	IN	IN	IN	OFF	IN	IN	IN	IN	OFF

註：需要搭配軟體設定

註：一個 SET 等於 4 bits 設定

SET0	LDIO0~3
SET1	LDIO4~7
SET2	LDIO8~11
SET3	LDIO12~15
SET4	LDIO16~19
SET5	LDIO20~23
SET6	LDIO24~27

2-3 連接器接腳定義

EDIO-800 9 Pin D-型連接器接腳定義如下(CN 1-6)

1	CLK/	6	CLK
2	SCS/	7	SCS
3	S2MD/	8	S2MD
4	M2SD/	9	M2SD
5	GND		

圖 5. EDIO-800 D-型連接器腳位定義

接腳 1,6 分別為 CLK\, CLK，傳送時脈差動輸出

接腳 2,7 分別為 SCS\, SCS，遠端模組選擇致能差動輸出

接腳 3,8 分別為 S2MD\, S2MD，串列資料差動輸入訊號

接腳 4,9 分別為 M2SD\, M2SD，串列資料差動輸出訊號

接腳 5 為 GND，PC 系統地電位

註：EDIO-800 和 EDIO-S001 之間的地電位,可透過 pin 5 來連接
在配線上,應考慮雜訊的嚴重性,來決定共地於否。



JP1 / JP2 34 Pin 牛角連接器接腳定義如下(CN 1-6)：

1	Gnd	2	Gnd
3	Gnd	4	Gnd
5	LDI0 / LDIO0	6	LDI1 / LDIO1
7	LDI2 / LDIO2	8	LDI3 / LDIO3
9	LDI4 / LDIO4	10	LDI5 / LDIO5
11	LDI6 / LDIO6	12	LDI7 / LDIO7
13	LDI8 / LDIO8	14	LDI9 / LDIO9
15	LDI10 / LDIO10	16	LDI11 / LDIO11
17	LDI12 / LDIO12	18	LDI13 / LDIO13
19	LDI14 / LDIO14	20	LDI15 / LDIO15
21	LDI16 / LDIO16	22	LDI17 / LDIO17
23	LDI18 / LDIO18	24	LDI19 / LDIO19
25	LDI20 / LDIO20	26	LDI21 / LDIO21
27	LDI22 / LDIO22	28	LDI23 / LDIO23
29	LDI24 / LDIO24	30	LDI25 / LDIO25
31	LDI26 / LDIO26	32	LDI27 / LDIO27
33	Gnd	34	Gnd



第三章 軟體安裝說明

3-1 安裝程式

EDIO-800 輸出入控制模組所提供的軟體組件部分包括驅動程式庫，驅動程式庫標頭檔，中斷處理函式原始檔及中斷處理函式標頭檔，發展環境為 BC3.1，以下分別就各個檔案說明

Driver_s.Lib : Memory model 為 Small mode 驅動程式庫。

Driver_m.Lib : Memory model 為 Middle mode 驅動程式庫。

Driver_l.Lib : Memory model 為 Large mode 驅動程式庫。

Driver.H : 驅動程式庫標頭檔。所有驅動函式雛型宣告皆定義於此檔案內。

編輯程式時請加入以下程式碼：

```
#include "driver.h"
```

IntHdl.C : 中斷處理副程式原始檔(Interrupt Handle)。此原始檔中包含各個模組之中斷副程式，使用者必須根據自己需求更改副程式內容。請參考 3-3 範例程式。

IntHdl.H : 中斷處理副程式標頭檔。所有中斷函式雛型宣告皆定義於此檔案內，使用時請把此檔拷貝至工作目錄下。並加入以下程式碼於開發程式中。

```
#include "inthdl.h"
```

EDIO.EXE : 測試/DEMO 程式。

EXAMPLE1.C : 範例程式 1 原始碼。

EXAMPLE2.C : 範例程式 2 原始碼。

EXAMPLE3.C : 範例程式 3 原始碼。

3-2 驅動程式庫

EDIO-800所提供的驅動程式包含系統界面、遠端輸出入、近端輸出入、計時器/看門狗四大部分，以下列出所有驅動程式名稱，並分別就各個驅動程式作說明。

3-2.1 有關係統界面之驅動程式

ISA Interface Control (EPCIO IO Page 0)

EDIO_init()

```
BOOL EDIO_init ( int base_addr,  
                 int irq_no,  
                 RIOISR  myEPCIO_RIO0_ISR,  
                 RIOISR  myEPCIO_RIO1_ISR,  
                 LIOISR  myEPCIO_LIO_ISR,
```

Argument	base_addr：EDIO 板基址 irq_no：中斷程式向量 myEPCIO_RIO0_ISR：User 自己撰寫的 RIO 第一組中斷副程式之 Function Pointer myEPCIO_RIO1_ISR：User 自己撰寫的 RIO 第二組中斷副程式之 Function Pointer myEPCIO_LIO_ISR：User 自己撰寫的 LIO 中斷副程式之 Function Pointer
Return Value	true：Initial EDIO 模組成功 False：Initial EDIO 模組失敗
Description	初始化板基址，中斷向量號碼及指定 User 的中斷服務副程式，當副程式名稱給定為 NULL 時，則呼叫內定服務副程式。



EDIO_close()

BOOL EDIO_close(void)

Argument 無

Return Value true : Close EDIO 模組成功
false : Close EDIO 模組失敗

Description 結束 EDIO 模組，本函式會關閉 EDIO 模組內所有功能，若初始化時有設定中斷功能，亦會還原中斷向量。

EDIO_ISA_SetBase()

BOOL EDIO_ISA_SetBase(WORD base)

Argument base : 板基址

BASE_200 : 板基址設定為 200h
BASE_220 : 板基址設定為 220h
BASE_240 : 板基址設定為 240h
BASE_260 : 板基址設定為 260h
BASE_280 : 板基址設定為 280h
BASE_2A0 : 板基址設定為 2A0h
BASE_2C0 : 板基址設定為 2C0h
BASE_2E0 : 板基址設定為 2E0h
BASE_300 : 板基址設定為 300h
BASE_320 : 板基址設定為 320h
BASE_340 : 板基址設定為 340h
BASE_360 : 板基址設定為 360h
BASE_380 : 板基址設定為 380h
BASE_3A0 : 板基址設定為 3A0h
BASE_3C0 : 板基址設定為 3C0h
BASE_3E0 : 板基址設定為 3E0h

Return Value true : 設定成功
false : 基址範圍設定錯誤

Description 設定 EDIO 板基址，本函式請搭配硬體板基址設定。

EDIO_ResetModule()

BOOL EDIO_ResetModule(WORD Module_no)

Argument Module_no : Reset Module 編號

RESET_RIO0 : Remote IO Set 0 Module



	<i>RESET_RIO1</i> : Remote IO Set 1 Module
	<i>RESET_LIO</i> : Local IO Module
	<i>RESET_ALL</i> : All Modules
Return Value	true : reset success False : Module_no 指定錯誤，不在 參數設定有效值內
Description	Reset 所選定的 EDIO 模組，本函式為提供使用者以軟體設定方式重致 EDIO 模組，EDIO 可單獨重致各模組，亦可重致所有模組。

EDIO_ISA_GetIntSource()

BOOL EDIO_ISA_GetIntSource(EPCIOINT *source)

Argument	source 為提供使用者以讀取中斷訊號之 Module 編號；此為一結構變數，定義如下： <pre>typedef struct _EPCIO_INT { BOOL RIO0; BOOL RIO1; BOOL LIO; }EPCIOINT;</pre> 其中當讀回值為 true 時表該功能模組發生中斷，false 表未發生中斷，各狀態變數說明如下： source->RIO0 表 Remote IO set 0 觸發中斷之狀態 source->RIO1 表 Remote IO set 1 觸發中斷之狀態 source->LIO 表 Local IO Module 觸發中斷之狀態
Description	當硬體中斷產生時應先呼叫此 函數判別中斷是由那一個 Module 所觸發，再至該 Module 所支援的 function 判斷中斷產生的源由。
See also	EDIO_get_RIO0_int_condition() , EDIO_get_RIO1_int_condition() , EDIO_get_LIO_int_condition() 。

EDIO_ISA_SetWaitState()

BOOL EDIO_ISA_SetWaitState(WORD wait_state)

Argument	wait_state : ISA Bus R/W wait state number (1 ~ 8)
----------	--

Return Value	true：設定成功。 false：wait_state 指定錯誤，不在 1~8 的有效值內
Description	以軟體方式設定 ISA Bus Read/Write Data 的存取等待時間，設定值為 ISA Bus 上 clock 的數量。

EDIO_ISA_SetIrqNo()

BOOL EDIO_ISA_SetIrqNo(WORD irqno)

Argument	irqno：ISA Bus IRQ 中斷編號 IRQ3：設定 Interrupt Request Number 為 3 IRQ5：設定 Interrupt Request Number 為 5 IRQ7：設定 Interrupt Request Number 為 7 IRQ10：設定 Interrupt Request Number 為 10 IRQ11：設定 Interrupt Request Number 為 11 IRQ12：設定 Interrupt Request Number 為 12 IRQ15：設定 Interrupt Request Number 為 15
Return Value	true：設定成功。 false：irqno 指定錯誤，不在中斷編號的有效值內
Description	以軟體方式設定 EDIO 板產生中斷時的 ISA Bus IRQ 編號
See also	EDIO_ISA_ResetIrqNo()

EDIO_ISA_ResetIrqNo()

BOOL EDIO_ISA_ResetIrqNo(void)

Argument	無
Return Value	true：取消設定成功。 false：無中斷向量值被設定過
Description	以軟體方式取消 EDIO 板產生中斷時的 ISA Bus IRQ 編號
See also	EDIO_ISA_ResetIrqNo()

EDIO_ISA_SetIntPeriod()

BOOL EDIO_ISA_SetIntPeriod(WORD period)

Argument	period：system clock number (1~255)
Return Value	true：設定成功。 false：period 指定範圍錯誤，不在指定有效值內



Description	設定 ISA Bus 中斷訊號產生時，其 Low active 週期佔用多少個 system clock。
-------------	---

EDIO_ISA_SetActivePage()

BOOL EDIO_ISA_SetActivePage(WORD page_no)

Argument	page_no : IO port active page _BUS_PAGE0_ : ISA Bus interface control page _RIO0_PAGE5_ : Remote digital I/O set 0 interface control page _RIO1_PAGE6_ : Remote digital I/O set 1 interface control page _LIO_PAGE8_ : Local digital I/O page
Return Value	true : 設定成功。 false : page_no 指定範圍錯誤。
Description	設定 IO R/W 動作時之 Active Page 編號
See also	EDIO_ISA_GetActivePage()

EDIO_ISA_GetActivePage()

BOOL EDIO_ISA_GetActivePage(WORD *page_no)

Argument	page_no : 目前 IO R/W 動作時之 Active Page 編號，讀回值內容請參考 EDIO_ISA_SetActivePage()。
Return Value	true : 讀取成功。 false : 讀取失敗。
Description	讀取目前 IO R/W 動作時之 Active Page 編號
See also	EDIO_ISA_SetActivePage()

3-2.2 有關遠端輸出入控制驅動程式

Remote Digital IO (EPCIO IO Page 5、6)

EDIO_RIO0_GetIntCondition()

BOOL EDIO_RIO0_GetIntCondition(RIOINT *source)

Argument source 為提供 User 使用以讀取 RIO0 Page 中斷訊號之源由；此為一結構變數，定義如下：

```
typedef struct _RIO_INT
```

```
{
```

```
  BOOL S0DI0;
```

```
  BOOL S0DI1;
```

```
  BOOL S0DI2;
```

```
  BOOL S0DI3;
```

```
  BOOL S1DI0;
```

```
  BOOL S1DI1;
```

```
  BOOL S1DI2;
```

```
  BOOL S1DI3;
```

```
  BOOL S2DI0;
```

```
  BOOL S2DI1;
```

```
  BOOL S2DI2;
```

```
  BOOL S2DI3;
```

```
  BOOL FAIL;
```

```
}RIOINT;
```

其中當讀回值為 true 時表該功能模組發生中斷，false 表未發生中斷，各狀態變數說明如下：

source->S0DI0 : RIO0 Slave0 DI0 interrupt happened

source->S0DI1 : RIO0 Slave0 DI1 interrupt happened

source->S0DI2 : RIO0 Slave0 DI2 interrupt happened

source->S0DI3 : RIO0 Slave0 DI3 interrupt happened

source->S1DI0 : RIO0 Slave1 DI0 interrupt happened

source->S1DI1 : RIO0 Slave1 DI1 interrupt happened

source->S1DI2 : RIO0 Slave1 DI2 interrupt happened



	source->S1DI3 : RIO0 Slave1 DI3 interrupt happened
	source->S2DI0 : RIO0 Slave2 DI0 interrupt happened
	source->S2DI1 : RIO0 Slave2 DI1 interrupt happened
	source->S2DI2 : RIO0 Slave2 DI2 interrupt happened
	source->S2DI3 : RIO0 Slave2 DI3 interrupt happened
	source->FAIL : RIO0 transmission fail interrupt happened
Return Value	true : 讀取中斷源成功 false : 無中斷發生
Description	讀取 Remote I/O (RIO0) 發出中斷的條件狀態，並清除中斷 Latch 值，等待下一次中斷發生。RIO0 可連接最多 3 組 Slave，每個 Slave 前 4 點輸入具有中斷功能，當硬體中斷發生後，可先經由 EDIO_ISA_GetIntSource () 判斷是否為 RIO0 Page 所發生，若是則呼叫本函式讀取中斷發生源。
See also	EDIO_ISA_GetIntSource()

EDIO_RIO1_GetIntCondition()

BOOL EDIO_RIO1_GetIntCondition(RIOINT *source)

Argument source 為提供 User 使用以讀取 RIO1 Page 中斷訊號之源由；此為一結構變數，定義如下：

```
typedef struct _RIO_INT
```

```
{
```

```
  BOOL S0DI0;
```

```
  BOOL S0DI1;
```

```
  BOOL S0DI2;
```

```
  BOOL S0DI3;
```

```
  BOOL S1DI0;
```

```
  BOOL S1DI1;
```

```
  BOOL S1DI2;
```

```
  BOOL S1DI3;
```

```
  BOOL S2DI0;
```

```
  BOOL S2DI1;
```

```
  BOOL S2DI2;
```

```
  BOOL S2DI3;
```

```
  BOOL FAIL;
```

```
}RIOINT;
```

其中當讀回值為 true 時表該功能模組發生中斷，false 表未發生

中斷，各狀態變數說明如下：

source->S0DI0 : RIO1 Slave0 DI0 interrupt happened
source->S0DI1 : RIO1 Slave0 DI1 interrupt happened
source->S0DI2 : RIO1 Slave0 DI2 interrupt happened
source->S0DI3 : RIO1 Slave0 DI3 interrupt happened
source->S1DI0 : RIO1 Slave1 DI0 interrupt happened
source->S1DI1 : RIO1 Slave1 DI1 interrupt happened
source->S1DI2 : RIO1 Slave1 DI2 interrupt happened
source->S1DI3 : RIO1 Slave1 DI3 interrupt happened
source->S2DI0 : RIO1 Slave2 DI0 interrupt happened
source->S2DI1 : RIO1 Slave2 DI1 interrupt happened
source->S2DI2 : RIO1 Slave2 DI2 interrupt happened
source->S2DI3 : RIO1 Slave2 DI3 interrupt happened
source->FAIL : RIO1 transmission fail interrupt happened

Return Value true : 讀取中斷源成功

 false : 無中斷發生

Description 讀取 Remote I/O (RIO1) 發出中斷的條件狀態，並清除中斷 Latch 值，等待下一次中斷發生。RIO1 可連接最多 3 組 Slave，每個 Slave 前 4 點輸入具有中斷功能，當硬體中斷發生後，可先經由 EDIO_ISA_GetIntSource() 判斷是否為 RIO1 Page 所發生，若是則呼叫本函式讀取中斷發生源。

See also **EDIO_ISA_GetIntSource()**

EDIO_ RIO_GetInputValue()

**BOOL EDIO_ RIO_GetInputValue(WORD set, WORD slave, WORD port,
 WORD *value)**

Argument set : Remote IO set number selection
 RIO_SET0 : Remote I/O Set 0
 RIO_SET1 : Remote I/O Set 1
 slave : Slave number selection in a set
 RIO_SLAVE0 : Remote I/O slave 0 in a set
 RIO_SLAVE1 : Remote I/O slave 1 in a set
 RIO_SLAVE2 : Remote I/O slave 2 in a set
 port : Digital Input port number selection in a slave
 RIO_PORT0 : DI 0 ~ DI 15 in a slave
 RIO_PORT1 : DI 16 ~ DI 31 in a slave



	<i>RIO_PORT2</i> : DI 32 ~ DI 47 in a slave
	<i>RIO_PORT3</i> : DI 48 ~ DI 63 in a slave
	value : variable name to read back the Digital input data
Return Value	true : 設定成功 false : 指定之參數 set, slave, port 不在設定範圍內。
Description	對於 Remote IO 指定的 set、slave、port 讀取目前 digital input 的訊號狀態。

EDIO_RIO_SetOutputValue()

**BOOL EDIO_RIO_SetOutputValue(WORD set, WORD slave, WORD port ,
WORD value)**

Argument	set : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1 slave : Slave number selection in a set <i>RIO_SLAVE0</i> : Remote I/O slave 0 in a set <i>RIO_SLAVE1</i> : Remote I/O slave 1 in a set <i>RIO_SLAVE2</i> : Remote I/O slave 2 in a set port : Digital output port number selection in a slave <i>RIO_PORT0</i> : DI 0 ~ DI 15 in a slave <i>RIO_PORT1</i> : DI 16 ~ DI 31 in a slave <i>RIO_PORT2</i> : DI 32 ~ DI 47 in a slave <i>RIO_PORT3</i> : DI 48 ~ DI 63 in a slave value : 16 bits output data
Return Value	true : 設定成功 false : 指定之參數 set, slave, port 不在設定範圍內。
Description	設定指定之 set , slave , port 編號之 16 bit digital output 輸出訊號狀態值。

EPCIO_RIO_GetTransStatus()

BOOL EDIO_RIO_GetTransStatus(WORD set_no, WORD *status)

Argument	set_no : Remote IO set number selection status : RIO master/slave receive data status 0 : RIO set number is working 1 : RIO set number is stop
----------	---

Return Value	true : 讀取狀態成功 false : 指定之參數 set 不在設定範圍內。
Description	讀取目前 RIO set number 傳輸狀態。若傳輸已停止，則呼叫 EDIO_RIO_GetMasterStatus()及 EDIO_RIO_GetSlaveStatus()分辨產生錯誤的為 Master 端或 Slave 端。
See also	EDIO_RIO_GetMasterStatus() , EDIO_RIO_GetSlaveStatus()

EDIO_RIO_GetMasterStatus()

BOOL EDIO_RIO_GetMasterStatus(WORD set_no, WORD *status)

Argument	set_no : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1 status : RIO master receive data status 0 : RIO master receive data OK 1 : RIO master receive data FAIL
Return Value	true : 讀取狀態成功 false : 指定之參數 set 不在設定範圍內。
Description	讀取目前 RIO set number Master 端傳送資料到 Slave 的狀態。若傳送發生錯誤，再由 EDIO_RIO_GetSlaveFail() 分辨產生錯誤的 slave number。
See also	EDIO_RIO_GetSlaveStatus() , EDIO_RIO_GetSlaveFail()

EDIO_RIO_GetSlaveStatus()

BOOL EDIO_RIO_GetSlaveStatus(WORD set_no, WORD *status)

Argument	set_no : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1 status : RIO slave receive data status 0 : RIO slave receive data OK 1 : RIO slave receive data FAIL
Return Value	true : 讀取狀態成功 false : 指定之參數 set 不在設定範圍內。
Description	讀取目前 RIO set number Slave 端接收 Master 資料的狀態，若接



受發生錯誤，再由 EDIO_RIO_GetSlaveFail() 分辨產生錯誤的 slave number。

See also **EDIO_RIO_GetMasterStatus()**，**EDIO_RIO_GetSlaveFail()**

EDIO_RIO_SetClockDivider()

BOOL EDIO_RIO_SetClockDivider(WORD set_no, WORD divider)

Argument	set_no : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1 divider : Remote IO clock divider (0~255)
Return Value	true : 設定成功 false : 指定之參數 se_no 不在設定範圍內。
Description	設定 Remote IO 傳送資料的時脈頻率。傳送頻率為 System clock 除以 4(divider+1)，內定設定值為 0。

EDIO_RIO_SetIntType()

BOOL EDIO_RIO_SetIntType(WORD set, WORD slave, WORD input, WORD type)

Argument	set : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1 slave : Remote slave number selection <i>RIO_SLAVE0</i> : Remote I/O slave 0 in a set <i>RIO_SLAVE1</i> : Remote I/O slave 1 in a set <i>RIO_SLAVE2</i> : Remote I/O slave 2 in a set input : Slave DI number <i>RIO_DI0</i> : Remote I/O input 0 in a slave <i>RIO_DI1</i> : Remote I/O input 1 in a slave <i>RIO_DI2</i> : Remote I/O input 2 in a slave <i>RIO_DI3</i> : Remote I/O input 3 in a slave type : interrupt trigger type <i>RIO_INT_RISE</i> : Rising edge trigger <i>RIO_INT_FALL</i> : Falling edge trigger <i>RIO_INT_LEVEL</i> : level change trigger
----------	--



Return Value	true：設定成功 false：指定之參數 set，slave，input 或 type 不在設定範圍內。
Description	設定 Remote IO digital input 點中斷訊號觸發方式為「前緣觸發」、「後緣觸發」或「轉態觸發」。本函式設定完成後，必須呼叫 EDIO_RIO_EnableInputInt() 函式，開啟中斷功能。
See also	EDIO_RIO_EnableInputInt()

EDIO_RIO_EnableInputInt()

BOOL EDIO_RIO_EnableInputInt(WORD set, WORD slave, WORD input)

Argument	set：Remote IO set number selection <i>RIO_SET0</i> ：Remote I/O Set 0 <i>RIO_SET1</i> ：Remote I/O Set 1 slave：Remote set-slave number <i>RIO_SLAVE0</i> ：Remote I/O slave 0 in a set <i>RIO_SLAVE1</i> ：Remote I/O slave 1 in a set <i>RIO_SLAVE2</i> ：Remote I/O slave 2 in a set input：Slave DI number <i>RIO_DI0</i> ：Remote I/O input 0 in a slave <i>RIO_DI1</i> ：Remote I/O input 1 in a slave <i>RIO_DI2</i> ：Remote I/O input 2 in a slave <i>RIO_DI3</i> ：Remote I/O input 3 in a slave
Return Value	true：設定成功 false：指定之參數 set，slave，input 不在設定範圍內。
Description	Remote IO 每一 set 具有 3 組 slave，每一組 Slave 的前四點 digital input (DI0、DI1、DI2、DI3) 可引發中斷。此函數致能 DI0~DI3 中斷功能。
See Also	EDIO_RIO_SetIntType

EDIO_RIO_DisableInputInt()

BOOL EDIO_RIO_DisableInputInt(WORD set, WORD slave, WORD input)

Argument	set：Remote IO set number selection <i>RIO_SET0</i> ：Remote I/O Set 0 <i>RIO_SET1</i> ：Remote I/O Set 1 slave：Remote slave number selection <i>RIO_SLAVE0</i> ：Remote I/O slave 0 in a set
----------	---

	<i>RIO_SLAVE1</i> : Remote I/O slave 1 in a set
	<i>RIO_SLAVE2</i> : Remote I/O slave 2 in a set
	input : Slave DI number
	<i>RIO_DI0</i> : Remote I/O input 0 in a slave
	<i>RIO_DI1</i> : Remote I/O input 1 in a slave
	<i>RIO_DI2</i> : Remote I/O input 2 in a slave
	<i>RIO_DI3</i> : Remote I/O input 3 in a slave
Return Value	true : 設定成功 false : 指定之參數 set , slave , input 不在設定範圍內
Description	Remote IO 每一 set 具有 3 組 slave, 每一組 Slave 的前四點 digital input 點(DI0、DI1、DI2、DI3)可引發中斷。此函數取消 DI0~DI3 中斷功能。

EDIO_RIO_SetTransError()

BOOL EDIO_RIO_SetTransError (WORD time)

Argument	time : Maximum transmission error time (0~15)
Return Value	true : 設定成功 false : 指定之參數 time 不在設定範圍內
Description	設定 Remote IO 最大傳輸錯誤重送次數。

EDIO_RIO_EnableSetControl()

BOOL EDIO_RIO_EnableSetControl(WORD set_no)

Argument	set_no : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1
Return Value	true : 設定成功 false : 指定之參數 set_no 不在設定範圍內
Description	開啟指定之 Remote IO set 之控制功能，而該 Set 之中之各個 Slave 功能需再呼叫 EDIO_RIO_EnableSlaveControl 才會開啟。
See Also	EDIO_RIO_EnableSlaveControl() , EDIO_RIO_DisableSlaveControl()

EDIO_RIO_DisableSetControl()

BOOL EDIO_RIO_DisableSetControl(WORD set_no)

Argument	set_no : Remote IO set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1
Return Value	true : 關閉成功 false : 指定之參數 set_no 不在設定範圍內
Description	關閉指定之 Remote IO set 之控制功能，該 Set 之中的所有 Slave 功能也一併關閉。
See also	EDIO_RIO_EnableSlaveControl() , EDIO_RIO_Disable SlaveControl()

EDIO_RIO_EnableSlaveControl()

BOOL EDIO_RIO_EnableSlaveControl(WORD set, WORD slave)

Argument	set : Remote I/O set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1 slave : Remote I/O slave number <i>RIO_SLAVE0</i> : Remote I/O slave 0 in a set <i>RIO_SLAVE1</i> : Remote I/O slave 1 in a set <i>RIO_SLAVE2</i> : Remote I/O slave 2 in a set
Return Value	true : 開啟成功 false : 指定之參數 set,slave 不在設定範圍內
Description	開啟指定的 Remote IO Slave 功能。Slave 功能開啟後，必須再呼叫 EDIO_RIO_EnableSetControl() 開啟 Set 功能，I/O 模組才開始做傳送與接收。
See also	EDIO_RIO_EnableSetControl() , EDIO_RIO_DisableSetControl()

EDIO_RIO_DisableSlaveControl()

BOOL EDIO_RIO_DisableSlaveControl(WORD set, WORD slave)

Argument	set : Remote I/O set number selection <i>RIO_SET0</i> : Remote I/O Set 0
----------	---



	<i>RIO_SET1</i> : Remote I/O Set 1
	slave : RIO set-slave number
	<i>RIO_SLAVE0</i> : Remote I/O slave 0 in a set
	<i>RIO_SLAVE1</i> : Remote I/O slave 1 in a set
	<i>RIO_SLAVE2</i> : Remote I/O slave 2 in a set
Return Value	true : 關閉成功 false : 指定之參數 set , slave 不在設定範圍內
Description	關閉指定的 Remote IO Slave 功能。本函式可單獨關閉指定的 Slave 功能。
See also	EDIO_RIO_EnableSetControl() , EDIO_RIO_DisableSetControl()

EDIO_RIO_EnableTransInt()

BOOL EDIO_RIO_EnableTransInt (WORD set)

Argument	set : Remote I/O set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1
Return Value	true : 開啟成功 false : 指定之參數 set 不在設定範圍內
Description	開啟 Remote IO 「Transmission Error」 中斷功能。

EDIO_RIO_DisableTransInt()

BOOL EDIO_RIO_DisableTransInt (WORD set_no)

Argument	set_no : Remote I/O set number selection <i>RIO_SET0</i> : Remote I/O Set 0 <i>RIO_SET1</i> : Remote I/O Set 1
Return Value	true : 關閉成功 false : 指定之參數 set 不在設定範圍內
Description	關閉 Remote IO 「Transmission Error」 中斷功能。

3-2.3 有關近端輸出入控制驅動程式

Local IO Control (EPCIO IO Page 8)

EDIO_LIO_GetIntCondition()

BOOL EDIO_LIO_GetIntCondition (LIOINT *source)

Argument source 為提供 User 使用以讀取 LIO Page 中斷訊號之源由；此為一結構變數，定義如下：

```
typedef struct _LIO_INT
```

```
{  
    BOOL LDI0;  
    BOOL LDI1;  
    BOOL LDI2;  
    BOOL LDI3;  
    BOOL LDI4;  
    BOOL LDI5;  
    BOOL LDI6;  
    BOOL LDI7;  
    BOOL DFI0;  
    BOOL DFI1;  
    BOOL DFI2;  
    BOOL DFI3;  
    BOOL DFI4;  
    BOOL DFI5;  
    BOOL DFI6;  
    BOOL TIMER;  
}LIOINT;
```

其中當讀回值為 true 時表該功能模組發生中斷，false 表未發生中斷，各狀態變數說明如下：

source->LDI0：Local Digital Input 第 0 點 interrupt happened

source->LDI1：Local Digital Input 第 1 點 interrupt happened

source->LDI2：Local Digital Input 第 2 點 interrupt happened

source->LDI3：Local Digital Input 第 3 點 interrupt happened

source->LDI4：Local Digital Input 第 4 點 interrupt happened

source->LDI5：Local Digital Input 第 5 點 interrupt happened

source->LDI6：Local Digital Input 第 6 點 interrupt happened

	source->LDI7 : Local Digital Input 第 7 點 interrupt happened
	source->DFI0 : Double Function input 第 0 點 interrupt happened
	source->DFI1 : Double Function input 第 1 點 interrupt happened
	source->DFI2 : Double Function input 第 2 點 interrupt happened
	source->DFI3 : Double Function input 第 3 點 interrupt happened
	source->DFI4 : Double Function input 第 4 點 interrupt happened
	source->DFI5 : Double Function input 第 5 點 interrupt happened
	source->DFI6 : Double Function input 第 6 點 interrupt happened
	source->TIMER : Timer interrupt happened
Return value	true : 讀取中斷源成功 false : 無中斷發生
Description	讀取 LIO Page 中斷發生的原因，並清除中斷 Latch 值，等待下一次中斷發生。當硬體中斷發生後，可先經由 EDIO_ISA_GetIntSource() 判斷是否為 LIO Page 所發生，若是則呼叫本函式讀取中斷發生源。
See also	EDIO_ISA_GetIntSource()

EDIO_LIO_GetLdiInput()

BOOL EDIO_LIO_GetLdiInput (DWORD *input)

Argument	input : Local I/O 輸入狀態值
Return Value	true : 讀取成功 false : 讀取不成功
Description	讀取 Local LDI0 ~ LDI27 數位輸入訊號值。Local I/O 具有 28 個可規劃為輸入或輸出的點，Bit0~Bit27 為 LDI0~LDI27，Bit28~Bit31 不代表輸入訊號值，當規劃為輸出模式時，則讀回值為輸出狀態。

EDIO_LIO_GetDfiInput ()

BOOL EDIO_LIO_GetDfiInput (DWORD *input)

Argument	input : Double function Input 輸入狀態值
Return Value	true : 讀取成功 false : 讀取不成功
Description	讀取 Double Function DFI0 ~ DFI19 數位訊號輸入狀態值。Double function Input 具有 20 點，Bit0~Bit19 為 DI0~DI19，

EDIO_LIO_SetLdoOutput ()

BOOL EDIO_LIO_SetLdoOutput (DWORD value)

Argument	value : LIO digital output DO0 ~ DO27 signal status
Return Value	true : 寫入成功 False : 寫入不成功
Description	輸出 LIO output LDO0 ~ LDO27 數位訊號狀態。此函式需先規劃 LIO 輸出功能。
See also	EDIO_LIO_EnableLdoOutput ()

EDIO_LIO_EnableLdoOutput()

BOOL EDIO_LIO_EnableLdoOutput (WORD port)

Argument	port : Local IO digital output enable port (每 4 點為一個 port) <i>LIO_OUT_EN0</i> : Port 0 is LDO0 ~ LDO3 <i>LIO_OUT_EN1</i> : Port 1 is LDO4 ~ LDO7 <i>LIO_OUT_EN2</i> : Port 2 is LDO8 ~ LDO11 <i>LIO_OUT_EN3</i> : Port 3 is LDO12 ~ LDO15 <i>LIO_OUT_EN4</i> : Port 4 is LDO16 ~ LDO19 <i>LIO_OUT_EN5</i> : Port 5 is LDO20 ~ LDO23 <i>LIO_OUT_EN6</i> : Port 6 is LDO24 ~ LDO27
Return Value	true : 開啟成功 false : 指定之參數 port 不在設定範圍內
Description	Local IO digital output 以 4 點為一個 port，28 點 Input/Output 共分為 Port 0 ~ Port 6，此函式可單獨啟動某一個 port 的 output 功能，初值設定所有 port 之 output 為 disable 狀態。

EDIO_LIO_DisableLdoOutput()

BOOL EDIO_LIO_DisableLdoOutput (WORD port)

Argument	port : Local IO digital output dividing port (每 4 點為一個 port) <i>LIO_OUT_EN0</i> : Port 0 is LDO0 ~ LDO3 <i>LIO_OUT_EN1</i> : Port 1 is LDO4 ~ LDO7 <i>LIO_OUT_EN2</i> : Port 2 is LDO8 ~ LDO11
----------	---



	<i>LIO_OUT_EN3</i> : Port 3 is LDO12 ~ LDO15
	<i>LIO_OUT_EN4</i> : Port 4 is LDO16 ~ LDO19
	<i>LIO_OUT_EN5</i> : Port 5 is LDO20 ~ LDO23
	<i>LIO_OUT_EN6</i> : Port 6 is LDO24 ~ LDO27
Return Value	true : 關閉成功 false : 指定之參數 port 不在設定範圍內
Description	此函式可單獨關閉某一個 port 的 output 功能。

EDIO_LDI_SetIntType()

BOOL EDIO_LDI_SetIntType (WORD inputno, WORD type)

Argument	inputno : digital input 中斷觸發 LDI 接點編號 (LDI0~LDI7) type : 中斷觸發型態 <i>LIO_INT_RISE</i> : Rising edge trigger (default) <i>LIO_INT_FALL</i> : Falling edge trigger <i>LIO_INT_LEVEL</i> : Level change trigger
Return Value	true : 設定成功 false : 指定之參數 port 不在設定範圍內
Description	設定 LIO Page 中具中斷型態之 Local digital input 接點，中斷觸發型態為上緣觸發或是下緣觸發或是轉態觸發。本函式設定完後必須再設定 EPCIO_LIO_EnableLdiInt()
See also	EDIO_LIO_EnableLdiInt()

EDIO_DFI_SetIntType()

BOOL EDIO_DFI_SetIntType (WORD inputno, WORD type)

Argument	inputno : digital input 中斷觸發 DFI 接點編號 (DFI0~DFI6) type : 中斷觸發型態 <i>LIO_INT_RISE</i> : Rising edge trigger (default) <i>LIO_INT_FALL</i> : Falling edge trigger <i>LIO_INT_LEVEL</i> : Level change trigger
Return Value	true : 設定成功 false : 指定之參數 inputno, type 不在設定範圍內
Description	設定 LIO Page 中具中斷型態之 Double function input 接點，其中斷觸發型態為上緣觸發或是下緣觸發或是轉態觸發。本函式設定完後必須再設定 EPCIO_LIO_EnableDfiInt()
See also	EDIO_LIO_EnableDfiInt()

EDIO_LIO_EnableLdiInt()

BOOL EDIO_LIO_EnableLdiInt (WORD point)

Argument	point : LIO input point <i>LIO_LDI0</i> : Local digital input 0 interrupt <i>LIO_LDI1</i> : Local digital input 1 interrupt <i>LIO_LDI2</i> : Local digital input 2 interrupt <i>LIO_LDI3</i> : Local digital input 3 interrupt <i>LIO_LDI4</i> : Local digital input 4 interrupt <i>LIO_LDI5</i> : Local digital input 5 interrupt <i>LIO_LDI6</i> : Local digital input 6 interrupt <i>LIO_LDI7</i> : Local digital input 7 interrupt
Return Value	true : 設定成功 false : 指定之參數 point 不在設定範圍內
Description	開啟 LIO 中斷觸發輸出。呼叫本函數前請先呼叫 EDIO_LDI_SetIntType()設定中斷型式
See also	EDIO_LDI_SetIntType()

EDIO_LIO_DisableLdiInt()

BOOL EDIO_LIO_DisableLdiInt (WORD point)

Argument	point : LIO interrupt point <i>LIO_LDI0</i> : Local digital input 0 interrupt <i>LIO_LDI1</i> : Local digital input 1 interrupt <i>LIO_LDI2</i> : Local digital input 2 interrupt <i>LIO_LDI3</i> : Local digital input 3 interrupt <i>LIO_LDI4</i> : Local digital input 4 interrupt <i>LIO_LDI5</i> : Local digital input 5 interrupt <i>LIO_LDI6</i> : Local digital input 6 interrupt <i>LIO_LDI7</i> : Local digital input 7 interrupt
Return Value	true : 設定成功 false : 指定之參數 point 不在設定範圍內
Description	抑止 LIO Page 具中斷功能的 Local Digital Input 點中斷觸發條件。

EDIO_LIO_EnableDfiInt()

BOOL EDIO_LIO_EnableDfiInt (WORD point)

Argument	point : DFI input point <i>LIO_DFI0</i> : Double function input 0 interrupt <i>LIO_DFI1</i> : Double function input 1 interrupt <i>LIO_DFI2</i> : Double function input 2 interrupt <i>LIO_DFI3</i> : Double function input 3 interrupt <i>LIO_DFI4</i> : Double function input 4 interrupt <i>LIO_DFI5</i> : Double function input 5 interrupt <i>LIO_DFI6</i> : Double function input 6 interrupt
Return Value	true : 設定成功 false : 指定之參數 point 不在設定範圍內
Description	開啟 Double function input 中斷觸發輸出。呼叫本函數前請先呼叫 EDIO_DFI_SetIntType() 設定中斷型式
See also	EDIO_DFI_SetIntType()

EDIO_LIO_DisableDfiInt()

BOOL EDIO_LIO_DisableDfiInt (WORD point)

Argument	point : LIO interrupt point <i>LIO_DFI0</i> : Double function input 0 interrupt <i>LIO_DFI1</i> : Double function input 1 interrupt <i>LIO_DFI2</i> : Double function input 2 interrupt <i>LIO_DFI3</i> : Double function input 3 interrupt <i>LIO_DFI4</i> : Double function input 4 interrupt <i>LIO_DFI5</i> : Double function input 5 interrupt <i>LIO_DFI6</i> : Double function input 6 interrupt
Return Value	true : 設定成功 false : 指定之參數 point 不在設定範圍內
Description	抑止 LIO Page 具中斷功能之 Double Function Input 點中斷觸發條件。

3-2.4 有關計時器/看門狗控制驅動程式

EDIO_LIO_SetTimer()

BOOL EDIO_LIO_SetTimer (DWORD value)

Argument	value：24 bit 計時器內容值，可設定範圍 (0 ~ 2 ²⁴)
Return Value	true：設定成功 false：指定之參數 point 不在設定範圍內
Description	設定 LIO Page 計時器之計時時間，配合呼叫 EDIO_LIO_EnableTimer()及 EDIO_LIO_EnableTimer_Int()，計時終了將發出 LIO Timer 中斷觸發功能。Timer 計時時間之單位為 Syatem Clock。
See also	EDIO_LIO_EnableTimer()，EDIO_LIO_EnableTimerInt()

EDIO_LIO_EnableTimer()

BOOL EDIO_LIO_EnableTimer (void)

Argument	無
Return Value	true：開啟成功 flase：開啟不成功
Description	開啟 Timer 計時功能。呼叫本函式前請先設定 Timer 計時器值
See also	EDIO_LIO_SetTimer()

EDIO_LIO_DisableTimer()

BOOL EDIO_LIO_DisableTimer (void)

Argument	無
Return Value	true：關閉成功 flase：關閉不成功
Description	關閉 Timer 計時功能。

EDIO_LIO_EnableTimerInt()

BOOL EDIO_LIO_EnableTimerInt (void)

Argument	無
Return Value	true：開啟成功 false：開啟不成功
Description	開啟 Timer 中斷觸發功能。呼叫本函式前請先設定 Timer 計時器值並開啟 Timer 計時功能
See also	EDIO_LIO_SetTimer()，EDIO_LIO_EnableTimer()

EDIO_LIO_DisableTimerInt()

BOOL EDIO_LIO_DisableTimerInt (void)

Argument	無
Return Value	true：關閉成功 false：關閉不成功
Description	關閉 Timer 中斷觸發功能。

EDIO_LIO_SetWdogTimer()

BOOL EDIO_LIO_SetWdogTimer (WORD value)

Argument	value：watch dog timer value
Return Value	true：設定成功 false：設定不成功
Description	設定 Watch Dog 計時器計時時間。Watch Dog 計時器的 Time Base 為使用 LIO timer 輸出，請呼叫 EDIO_LIO_SetTimer()函式設定。啟動 Watch Dog 功能請呼叫 EDIO_LIO_EnableWdogTimer()
See also	EDIO_LIO_SetTimer()，EDIO_LIO_EnableWdogTimer()

EDIO_LIO_EnableWdogTimer()

BOOL EDIO_LIO_EnableWdogTimer (void)

Argument	無
Return Value	true：開啟成功



	False：開啟不成功
Description	開啟 Watch Dog Timer 功能。請先設定 EPCIO_LIO_SetWdogTimer() 函式。
See also	EDIO_LIO_SetWdogTimer()

EDIO_LIO_DisableWdogTimer()

BOOL EDIO_LIO_DisableWdogTimer (void)

Argument	無
Return Value	true：關閉成功 False：關閉不成功
Description	關閉 Watch Dog Timer 功能。

EDIO_LIO_SetWdogReset()

BOOL EDIO_LIO_SetWdogReset (DWORD value)

Argument	value：設定 24 bit 的 reset 暫存器，設定範圍(0 ~ 2 ²⁴)
Return Value	true：設定成功 false：指定之參數 point 不在設定範圍內
Description	設定 Watch Dog 計時器 reset 訊號持續時間，看門狗計時器 Time out 後將觸發 Reset 動作，Reset 維持時間可透過本函式規劃。設定單位為 System Clock。

EDIO_LIO_RefreshWdogTimer()

BOOL EDIO_LIO_RefreshWdogTimer (void)

Argument	無
Return Value	true：設定成功 False：設定不成功
Description	更新 Watch Dog 計時器時間。當使用者開啟 Watch Dog 功能後，必須在固定時間內呼叫本函式清除 Watch Dog 計時器內容。否則一但 Time Out 後將發生 Reset 動作。

3-3 範例程式

3-3.1 遠端輸出入範例程式

Example of Remote Input / Output Control

```
/******  
/*目的: 利用 EPCIO 所提供驅動程式做 RIO 遠端數位輸出入控制 *  
/*功能: 本範例程式規劃 RIO 傳輸時脈, 傳輸錯誤容許次數, 傳輸錯誤中斷功*  
/*      能, 輸入點中斷功能等, 讀取遠端狀態及寫出遠端狀態. *  
/*作者: Chen Wen-Chuan 2000.06.21 *  
/*TOOL: Borland C 3.1 *  
/******  
  
#include <stdio.h>  
#include <conio.h>  
#include "epciodev.h"  
#include "epcioint.h"  
  
#define Type          0x1  
#define ClkDiv        0x9  
#define MaxErr        0xA  
  
void rio_isr_function(RIOINT rio_int_cond);  
  
WORD input,isr_count=0,isr_err_count=0,Base;  
void main()  
{  
clrscr();  
printf("Input Base Address in HEX (200h,220h...3C0h,3E0h) : ");  
scanf("%x",&Base);  
  
if(!EDIO_init(Base,IRQ5,rio_isr_function,rio_isr_function, NULL))//初始化 EDIO  
模組  
{  
printf("Initialize Error");  
return;  
}  
EDIO_ResetModule(RESET_ALL);  
EDIO_RIO_SetClockDivider(RIO_SET0,ClkDiv);//設定第一組傳輸時脈為 1
```



MHz

```
EDIO_RIO_SetTransError(RIO_SET0,MaxErr);//設定最大傳輸錯誤 10 次
EDIO_RIO_EnableTtransInt(RIO_SET0);//設定傳輸失效中斷
EDIO_RIO_EnableSlaveControl(RIO_SET0, RIO_SLAVE0);
EDIO_RIO_EnableSetControl(RIO_SET0);//開始傳輸
EDIO_RIO_SetIntType(RIO_SET0, RIO_SLAVE0, RIO_DI0, RIO_INT_RISE);
EDIO_RIO_EnableInputInt(RIO_SET0, RIO_SLAVE0, RIO_DI0);
```

```
while(!kbhit())
{
    gotoxy(10,5);
    EDIO_RIO_GetInputValue(RIO_SET0,RIO_SLAVE0,RIO_PORT0,&input);
    printf("Rio input value %x\n",input ); //讀取 Slave0 輸入狀態
    gotoxy(10,6);
    printf("Rio isr count is %d\n",isr_count);
    gotoxy(10,7);
    printf("Rio transfor error count is %d\n",isr_err_count);
}
EDIO_close();//關閉 EDIO 模組
}
```

```
void rio_isr_function(RIOINT rio_int_cond)//中斷服務程式
{
    if(rio_int_cond.S0DI0)//if DI0 input then
        isr_count++; //isr_count + 1
    if(rio_int_cond.FAIL) //if transfer FAIL
        isr_err_count++; //isr_err_count + 1

    EDIO_RIO_SetOutputValue(RIO_SET0,RIO_SLAVE0,RIO_PORT0,isr_count);//
    輸出至遠端模組
}
```

3-3.2 可規劃近端輸出入點範例程式

Example of Local Input / Output Control

```
/******  
/*目的: 利用 EDIO 所提供驅動程式做 LIO 遠端數位輸出入控制 *  
/*功能: 本範例程式規劃 LIO 輸出功能, 中斷型式, 開啟中斷功能, 讀取輸 *  
/*      入及寫出輸出 *  
/*作者: Chen Wen-Chuan 2000.06.21 *  
/*TOOL: Borland C 3.1 *  
/******  
#include <stdio.h>  
#include <conio.h>  
#include "epciodev.h"  
#include "epcioint.h"  
  
WORD isr_count=0;  
void lio_isr_function(LIOINT);  
  
void main()  
{  
    DWORD input;  
    WORD Base;  
  
    clrscr();  
    printf("Input Base Address in HEX (200h,220h...3C0h,3E0h) : ");  
    scanf("%x",&Base);  
  
    if(!EDIO_init(Base,IRQ5,NULL,NULL,lio_isr_function))//初始化 EDIO 模組  
    {  
        printf("Initialize Error");  
        return;  
    }  
    EDIO_ResetModule(RESET_ALL);//重致 EDIO 模組  
    EDIO_LIO_EnableLdoOutput(LIO_OUT_EN0);//開啟輸出功能  
    EDIO_LDI_SetIntType(LIO_LDI0,LIO_INT_RISE);//設定輸入中斷型式  
    EDIO_LIO_EnableLdiInt(LIO_LDI0);//開啟輸入點中斷功能  
  
    while(!kbhit())  
    {  
        gotoxy(5,10);  
        EDIO_LIO_GetLdiInput(&input);  
        printf("Local Digital Input %lx\n",input);  
        gotoxy(5,11);
```



```
    EDIO_LIO_GetDfiInput(&input);
    printf("Double Function Input %lx\n",input);
    gotoxy(5,12);
    printf("ISR interrupt count %10d\n",isr_count);

}
EDIO_close();//關閉 EDIO 模組
}

void lio_isr_function(LIOINT lioint)//LIO 中斷副程式
{
    if(lioint.LDI0)//if LDI0 input from LOW to HIGH(Interrupt happened)
    isr_count++;          //isr_count + 1
}
```

3-3.3 計時器/看門狗範例程式

Example of Timer / Watch Dog Control

```
/******  
/*目的: 利用 EPCIO 所提供驅動程式做 LIO 內建計時器與看門狗計時器控制*  
/*功能: 本範例程式規劃 LIO 計時器功能, 計時器中斷功能, 看門狗計時器,*  
/*      看門狗計時器 Time out 重致規劃*  
/*作者: Chen Wen-Chuan 2000.06.21*  
/*TOOL: Borland C 3.1*  
/******  
#include <stdio.h>  
#include <conio.h>  
#include "epciodev.h"  
#include "epcioint.h"  
  
#define Timer_value 0x200000  
#define Wdg_value 100  
#define Rst_value0x100  
  
int timer_int=0;  
void lio_isr_function(LIOINT);  
  
void main()  
{  
WORD Base;  
  
//clrscr();  
printf("Input Base Address in HEX (200h,220h...3C0h,3E0h) : ");  
scanf("%x",&Base);  
  
if(!EDIO_init(Base,IRQ5,NULL,NULL, lio_isr_function )//初始化 EDIO 模組  
{  
printf("Initial Fail");  
return;  
}  
EDIO_ResetModule(RESET_ALL);//重致 EDIO 模組  
EDIO_LIO_SetTimer(Timer_value);//設定計時器時間  
EDIO_LIO_EnableTimer_int();//開啟計時器中斷功能  
EDIO_LIO_EnableTimer();//開啟計時器功能  
  
EDIO_LIO_SetWdogReset(Rst_value);//設定 Reset 時間  
EDIO_LIO_SetWdogTimer(Wdg_value);//設定看門狗計時器時間
```



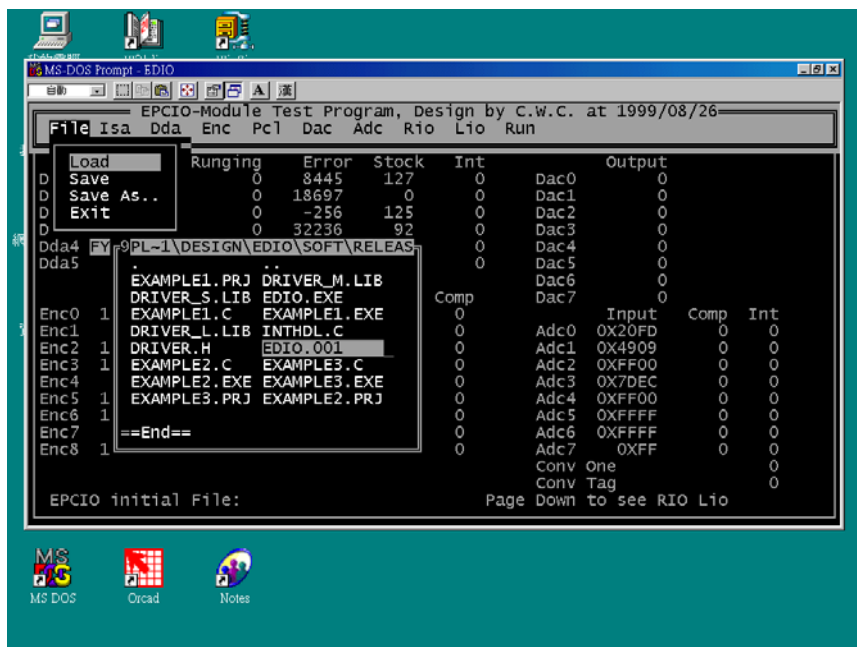
EDIO_LIO_EnableWdogTimer();//開啟看門狗計時器功能

```
while(!kbhit())
{
    EDIO_LIO_RefreshWdogTimer();//清除看門狗計數器內容
    gotoxy(10,5);
    printf("Timer Interrupt counter is %d\n",timer_int);
    gotoxy(10,6);
    printf("Hit any key to see Watch dog Time out and reset happen\n");
}
getch();
while(!kbhit())
{
    gotoxy(10,5);
    printf("Timer Interrupt counter is %d\n",timer_int);
    gotoxy(10,6);
    printf("Waiting for watch dog Time out...");
}

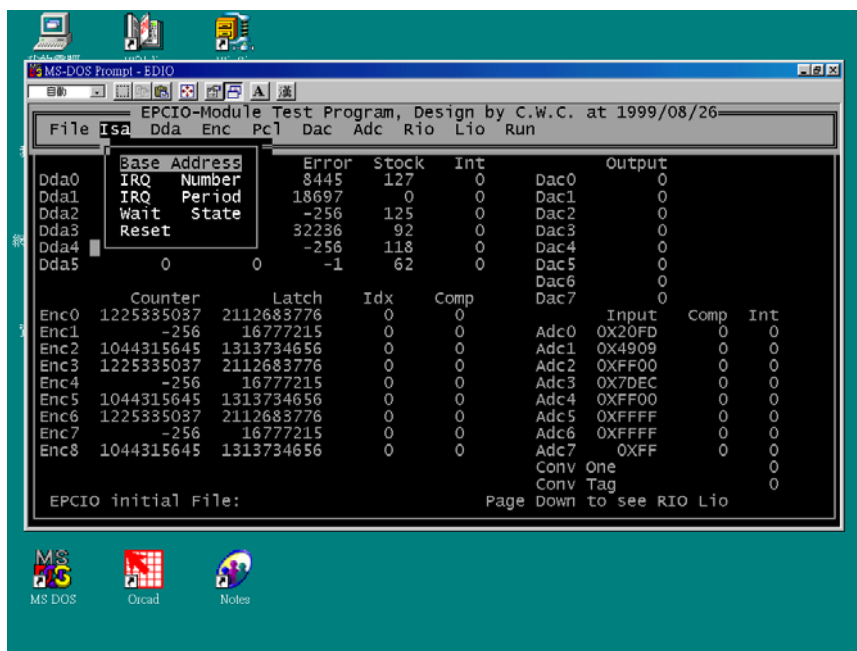
EDIO_close();//關閉 EDIO 模組
}

void lio_isr_function(LIOINT lioint)//LIO 中斷副程式
{
    if(lioint.TIMER)//if Timer interrupt happened
        timer_int++;           //timer_int + 1
}
```

3-4 測試程式

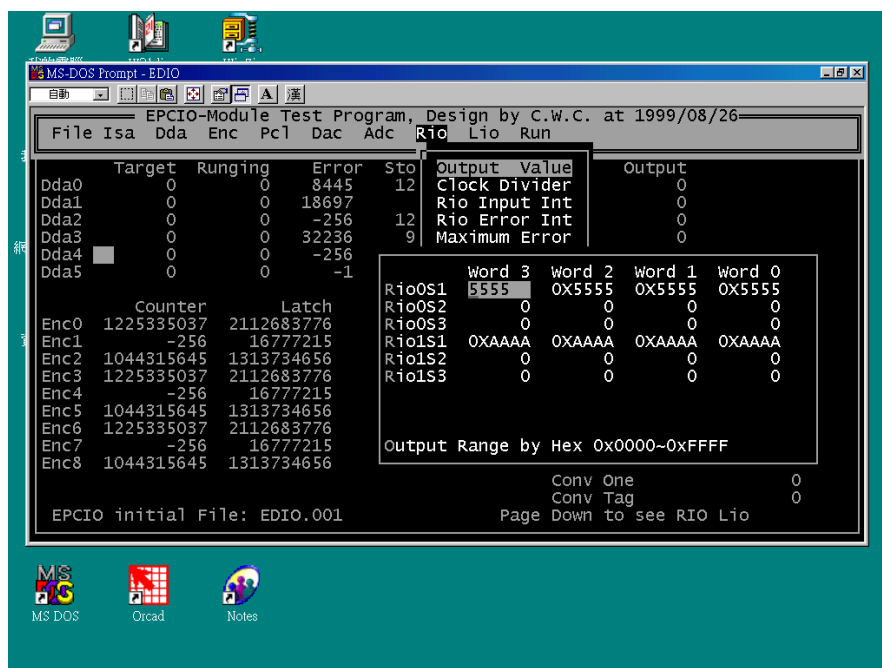


步驟 1：請利用 File->Load 載入功能載入 EDIO.001 設定檔，內定板基址為 0x240H，IRQ 為 3，若硬體模組的設定值有所變動，請利用 Isa->Base Address 及 IRQ Number 修改。

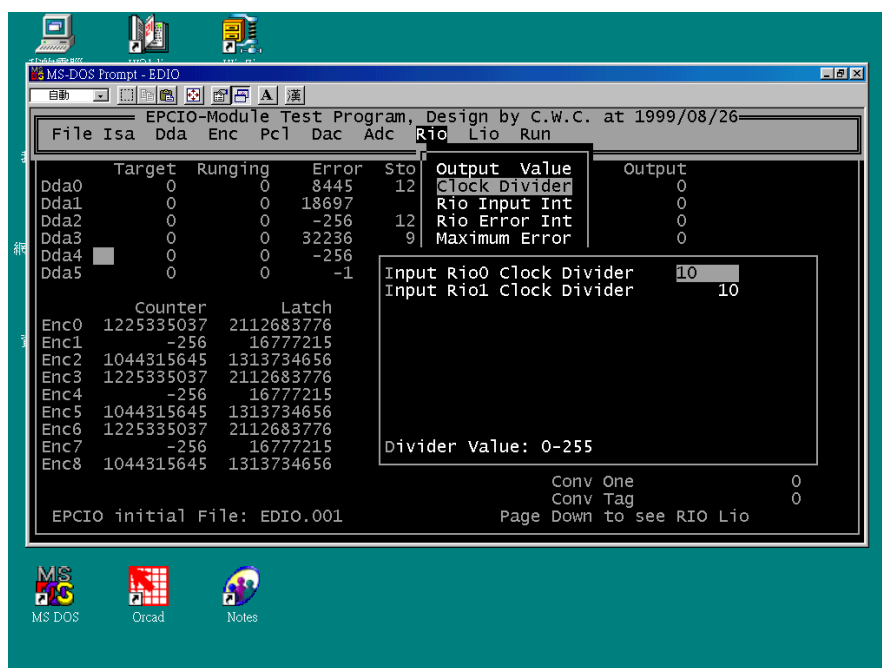


步驟 2：請利用 Rio->Output Value 修改輸出值設定，內定輸出址為 0x5555

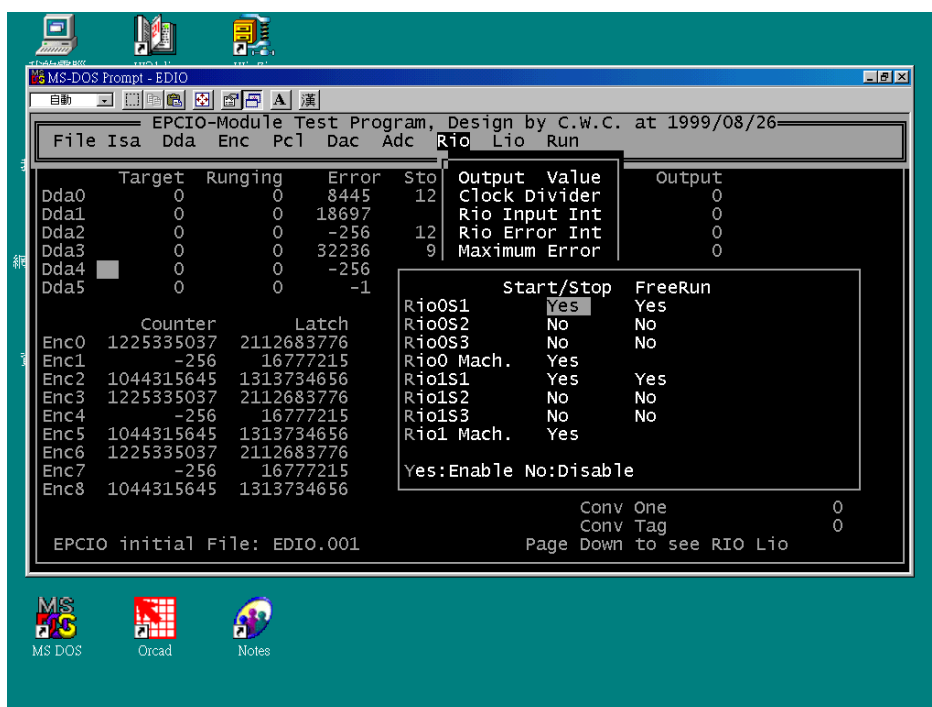
及 0xAAAA。



步驟 3：更改傳輸時脈請利用 Rio->Clock Divider 做修改，內定傳輸時脈除頻值設為 10，傳輸時脈約 $20\text{MHz}/2/(10+1)=1\text{Mhz}$



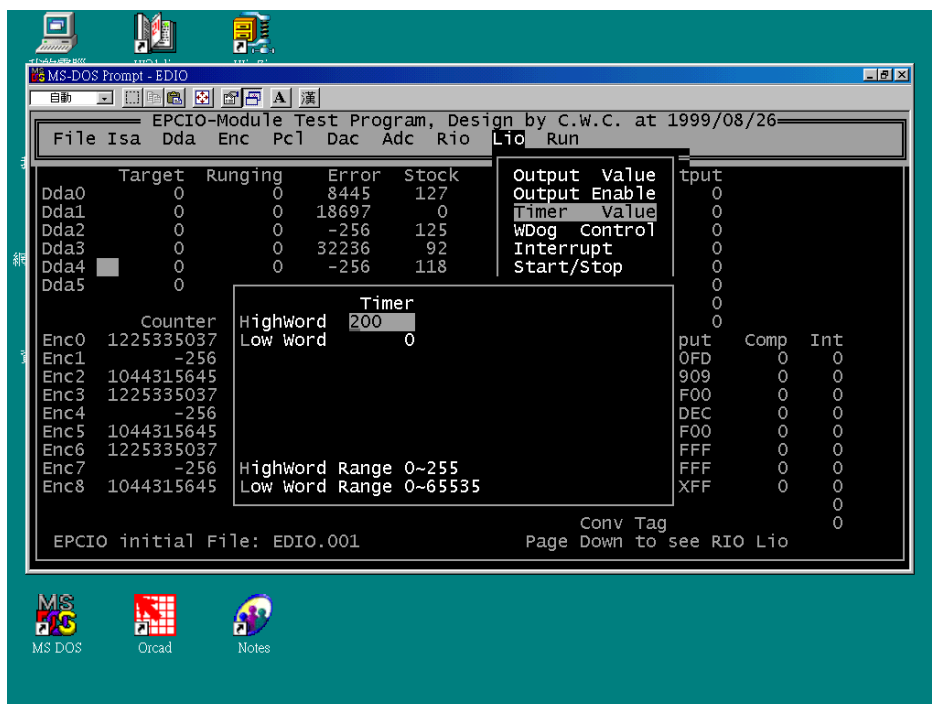
步驟 4：設定 Rio->Start/Stop 開始傳輸，內定設定為 Enable 兩組遠端模組的第一個 Slave。



步驟 5：設定完成後請到 Run -> <Enter> 開始傳輸。若要停止，請押 Stop-><Enter>

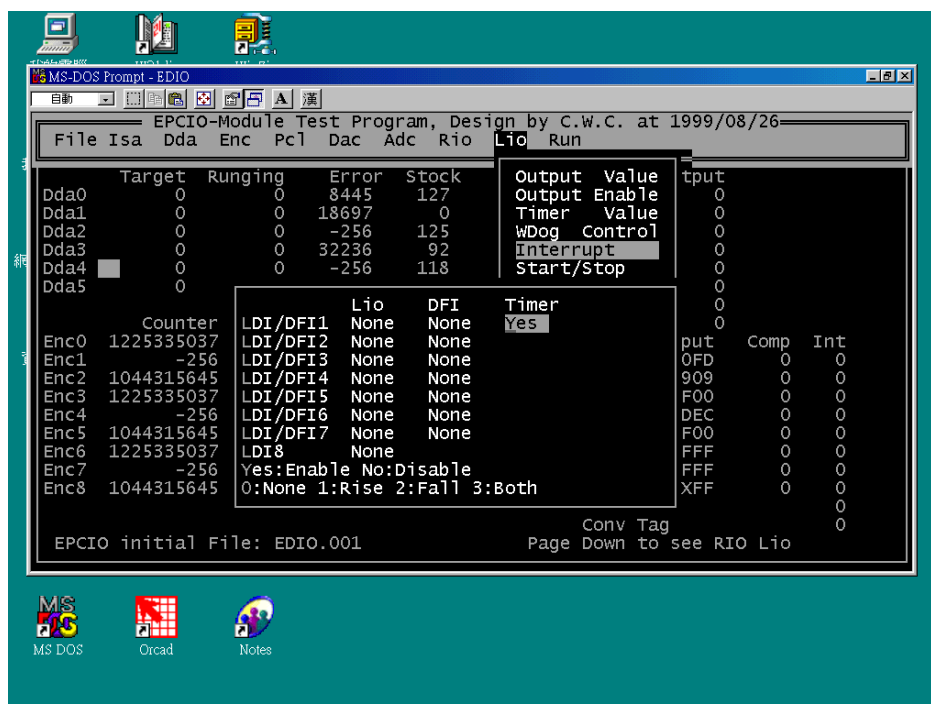
步驟 6: 若要設定跑馬燈功能, 請回到步驟 4, 並設定 FreeRun 功能 Enable。

步驟 7: 設定 Lio->Timer Value，設定時間越大，跑馬燈閃爍越慢。



步驟 8: 設定 Lio->Interrupt->Timer 為 Enable 狀態，並至

Lio->Start/Stop->Timer 設定為 Enable 狀態。



步驟 9: 設定完成後請到 Run -> <Enter> 開始傳輸。程式執行時可用 Page Down , Page Up 切換顯示螢幕觀看結果。



附錄

IO Port Definition

Page 0 : Bus interface control

P.A.	Read	Write
0	Interrupt Index *1	Software Reset *2
1		ISA Bus Wait State *3
2		Interrupt Channel & period *4
3		Interrupt Mode*5
4		Software Clear Interrupt*6
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	Read the present R/W page*7	Write the next R/W page register*7

*P.A. : Port Address

*1 [RD Addr. 0] Interrupt Index

Meaning of bit value :

D15~D9 No Meaning

D8~D0 Interruption Index

Bit	Interrupt Source	Value Meaning
D8	Reserved	0 No Interruption Happened 1 Interruption Happened
D7	Local IO	
D6	ADC	
D5	Remote IO 1	
D4	Remote IO 0	
D3	Reserved	
D2	Reserved	
D1	Reserved	
D0	Reserved	



*2 [WR Addr 0] Software Reset

Meaning of bit value :

D15~D9 Reserved (Fill with 0 when Write)

D8~D0 Software reset

Bit	Affected Module	Value Meaning
D9	All Module	0 Not Reset (default) 1 Reset Module
D8	Peripheral	
D7	Reserved	
D6	Reserved	
D5	Local IO	
D4	Reserved	
D3	Remote IO 1	
D2	Remote IO 0	
D1	Reserved	
D0	Reserved	

*3 [WR Addr 1] ISA Bus Wait State

Meaning of bit value :

D15~D3 Reserved (Filled with 0)

D2~D0 Wait State Number

[0 0 0]	8 Wait state (default)
[0 0 1]	7 Wait state
[0 1 0]	6 Wait state
[0 1 1]	5 Wait state
[1 0 0]	4 Wait state
[1 0 1]	3 Wait state
[1 1 0]	2 Wait state
[1 1 1]	1 Wait state

*4 [WR Addr 2] Interrupt Channel & period

Meaning of bit value :

D15~D8 Interrupt channel selection

Bit	Interrupt Channel	Value Meaning
D15	7	0 Disable(default) 1 Enable
D14	6	
D13	5	
D12	4	
D11	3	
D10	2	
D9	1	

D8 Reserved (Filled with 0)

D7~D0 Interruption signal low period (default is 0)



*5 [RD/WR Addr 3] Write Interrupt Mode

Meaning of Bits value

D15~D2 Reserved (Filled with 0)

D1~D0 Interrupt Mode Select

[0 0] Active Rising Edge(Default)

[0 1] Active Falling Edge

[1 0] Active High Level

[1 1] Active Low Level

*6 [RD/WR Addr 4] Software Clear Interrupt

Meaning of Bits value

D15~D0 Reserved (Filled with 0)

*7 [RD/WR Addr 15] Read/Write the page register

Meaning of bit value :

D15~D4 Reserved (Fill with 0 when Write)

D3~D0 Next Read/Write page

[0 0 0 0] Page 0 System Page

[0 1 0 1] Page 5 Remote IO 0

[0 1 1 0] Page 6 Remote IO 1

[1 0 0 0] Page 8 Local IO

Others Page 1~4,7,9~15 Reserved for future

Page 5 : First set remote Digital IO

P.A.	Read	Write
0	Set 1 Remote slave 0 input of channel 15~0	Set 1 Remote slave 0 output of channel 15~0
1	Set 1 Remote slave 0 input of channel 31~16	Set 1 Remote slave 0 output of channel 31~16
2	Set 1 Remote slave 0 input of channel 47~32	Set 1 Remote slave 0 output of channel 47~32
3	Set 1 Remote slave 0 input of channel 63~48	Set 1 Remote slave 0 output of channel 63~48
4	Set 1 Remote slave 1 input of channel 15~0	Set 1 Remote slave 1 output of channel 15~0
5	Set 1 Remote slave 1 input of channel 31~16	Set 1 Remote slave 1 output of channel 31~16
6	Set 1 Remote slave 1 input of channel 47~32	Set 1 Remote slave 1 output of channel 47~32
7	Set 1 Remote slave 1 input of channel 63~48	Set 1 Remote slave 1 output of channel 63~48
8	Set 1 Remote slave 2 input of channel 15~0	Set 1 Remote slave 2 output of channel 15~0
9	Set 1 Remote slave 2 input of channel 31~16	Set 1 Remote slave 2 output of channel 31~16
10	Set 1 Remote slave 2 input of channel 47~32	Set 1 Remote slave 2 output of channel 47~32
11	Set 1 Remote slave 2 input of channel 63~48	Set 1 Remote slave 2 output of channel 63~48
12	Set 1 Remote IO status*1	Set 1 Remote IO clock divider & Slave 2 IO Interrupt Control*3
13	Set 1 Remote IO Interruption latch*2	Set 1 Remote IO maximum error、slave enable & Slave 1 IO Interrupt Control*4
14		Slave 0 IO Interrupt Control & Transmission Failure interruption control*5
15	Read the present R/W page	Write the next R/W page register

*P.A. : Port Address

*1 [RD Addr 12] Read Set 1 Remote IO stop status

Meaning of bit value :

D15~D5 No Meaning

D4 Set 1 Remote IO enable/Disable status

0 Disabled

1 Enable

D3~D2 Master/Slave Receive data status

	Source	Value	Meaning
D3	Slave	0	No error
D2	Master	1	Receive failure

D1~D0 Fail slave

[0 1] Slave 0

[1 0] Slave 1

[1 1] Slave 2

*2 [RD Addr 13] Set 1 Remote IO Interruption latch

Meaning of bit value :

D15~D13 No meaning

D12~D0 Interrupt Latch

Bit	Interruption Source	Value Meaning	
D12	Transmission Failure	0	No interruption Happened
D11	Slave 2 DI 3	1	Interruption Happened
D10	Slave 2 DI 2		
D9	Slave 2 DI 1		
D8	Slave 2 DI 0		
D7	Slave 1 DI 3		
D6	Slave 1 DI 2		
D5	Slave 1 DI 1		
D4	Slave 1 DI 0		
D3	Slave 0 DI 3		
D2	Slave 0 DI 2		
D1	Slave 0 DI 1		
D0	Slave 0 DI 0		

*3 [WR Addr 12] Set 1 Remote IO clock divider & Slave 2 Interrupt Control

Meaning of bit value :

D15~D8 Slave 2 IO interruption Control

Bit	Interruption Source	Value Meaning	
[D15 D14]	Slave 2 DI 3	[0 0]	Disable Interrupt(default)
[D13 D12]	Slave 2 DI 2	[0 1]	Rising Edge Interrupt
[D11 D10]	Slave 2 DI 1	[1 0]	Falling Edge Interrupt
[D9 D8]	Slave 2 DI 0	[1 1]	Both Edge Interrupt

D7~D0 Set 1 Remote IO clock divider

*4 [WR Addr 13] Set 1 Remote IO maximum error 、 slave enable & Slave 1 IO

Interrupt Control

Meaning of bit value :

D15~D8 Slave 1 IO interruption Control

Bit	Interruption Source	Value Meaning	
[D15 D14]	Slave 1 DI 3	[0 0]	Disable Interrupt(default)
[D13 D12]	Slave 1 DI 2	[0 1]	Rising Edge Interrupt
[D11 D10]	Slave 1 DI 1	[1 0]	Falling Edge Interrupt
[D9 D8]	Slave 1 DI 0	[1 1]	Both Edge Interrupt

D7~D4 Maximum transmission error

D3 Enable Set 1 remote IO

0 Disable (default)

1 Enable

D2~D0 Enable Set 1 Slave 2~0

Bit	Slave	Value Meaning	
D2	2	0	Disable (default)
D1	1	1	enable
D0	0		

*5 [WR Addr 14] Slave 0 IO and transmission Interrupt Control

Meaning of bit value :

D15~D9 Reserved(Filled with 0)

D8 Enable Transmission Failure interruption

0 Disable (default)

1 Enable

D7~D0 Slave 0 IO interruption Control

Bit	Interruption Source	Value Meaning	
[D7 D6]	Slave 0 DI 3	[0 0]	Disable Interrupt(default)
[D5 D4]	Slave 0 DI 2	[0 1]	Rising Edge Interrupt
[D3 D2]	Slave 0 DI 1	[1 0]	Falling Edge Interrupt
[D1 D0]	Slave 0 DI 0	[1 1]	Both Edge Interrupt

Page 6 : Second set remote Digital IO

P.A.	Read	Write
0	Set 2 Remote slave 0 input of channel 15~0	Set 2 Remote slave 0 output of channel 15~0
1	Set 2 Remote slave 0 input of channel 31~16	Set 2 Remote slave 0 output of channel 31~16
2	Set 2 Remote slave 0 input of channel 47~32	Set 2 Remote slave 0 output of channel 47~32
3	Set 2 Remote slave 0 input of channel 63~48	Set 2 Remote slave 0 output of channel 63~48
4	Set 2 Remote slave 1 input of channel 15~0	Set 2 Remote slave 1 output of channel 15~0
5	Set 2 Remote slave 1 input of channel 31~16	Set 2 Remote slave 1 output of channel 31~16
6	Set 2 Remote slave 1 input of channel 47~32	Set 2 Remote slave 1 output of channel 47~32
7	Set 2 Remote slave 1 input of channel 63~48	Set 2 Remote slave 1 output of channel 63~48
8	Set 2 Remote slave 2 input of channel 15~0	Set 2 Remote slave 2 output of channel 15~0
9	Set 2 Remote slave 2 input of channel 31~16	Set 2 Remote slave 2 output of channel 31~16
10	Set 2 Remote slave 2 input of channel 47~32	Set 2 Remote slave 2 output of channel 47~32
11	Set 2 Remote slave 2 input of channel 63~48	Set 2 Remote slave 2 output of channel 63~48
12	Set 2 Remote IO status	Set 2 Remote IO clock divider & Slave 2 IO Interrupt Control
13	Set 2 Remote IO Interruption latch	Set 2 Remote IO maximum error 、 slave enable & Slave 1 IO Interrupt Control
14		Slave 0 IO Interrupt Control & Transmission Failure interruption control
15	Read the present R/W page	Write the next R/W page register

*P.A. : Port Address

Page 8 : LIO Control

P.A.	Read	Write
0	Local Digital IO LDIO[15:0] *1	Local Digital Output LDO[15:0] *6
1	Local Digital IO LDIO[31:16] *2	Local Digital Output LDO[31:16] *7
2	Local Double Function Digital Input DFI[15:0] *3	Enable Local Digital Output, Timer and Watch Dog Enable *8
3	Local Double Function Digital Input DFI[31:16] *4	Local Digital Input Interrupt Control *9
4	Local Digital Input & Timer Interrupt Latch*5	Local Double Function Input & Timer Interrupt Control *10
5		Timer Value low word *11
6		Timer Value high word *12
7		Watch Dog Reset duration low word *13
8		Watch Dog Reset duration high word *14
9		Watch Dog Timer value *15
10		Refresh Watch Dog *16
11		
12		
13		
14		
15	Read the present R/W page	Write the next R/W page register

*P.A. : Port Address

*1 [RD Addr 0] Read Local Digital IO LDIO[15:0]

Meaning of bit value :

D15~D12 Local Digital IO LDIO[15:0]

*2 [RD Addr 1] Read Local Digital IO LDIO[31:16]

Meaning of bit value :

D15~D0 Local Digital IO LDIO[31:16]

*3 [RD Addr 2] Read Local Double Function Digital Input DFI[15:0]

Meaning of bit value :

D15~D0 Local Double Function Digital Input DFI[15:0]

*4 [RD Addr 3] Read Local Double Function Digital Input DFI[31:16]

Meaning of bit value :

D15~D0 Local Double Function Digital Input DFI[31:16]

*5 [RD Addr 4] Read Local Digital Input & Timer Interrupt Latch

Meaning of bit value :

Bit	Interrupt Source	Value
D15	Timer	0 No Interrupt happen 1 Interrupt happened
D14~D8	DFI[6:0]	
D7~D0	LDI[7:0]	

*6 [WR Addr 0] Write Local Digital Output LDO[15:0]

Meaning of bit value :

D15~D0 Local Digital Output LDO[15:0]

*7 [WR Addr 1] Local Digital Output LDO[31:16]

Meaning of bit value :

D15~D0 Local Digital Output LDO[31:16]

*8 [WR Addr 2] Enable Local Digital Output, Timer and Watch Dog Enable

Meaning of bit value :

D15~D10 Reserved (filled with 0)

D9 Enable Watch Dog Timer

0 Disable(default)

1 Enable

D8 Enable Timer

0 Disable(default)

1 Enable

D7~D0 Enable Local Digital Output

Meaning of bit value :

Bit	Control Local Digital Output	Value
D7	LDO[31:28]	0 Disable Local Digital Output(default) 1 Enable Local Digital Output
D6	LDO[27:24]	
D5	LDO[23:20]	
D4	LDO[19:16]	
D3	LDO[15:12]	
D2	LDO[11:8]	
D1	LDO[7:4]	
D0	LDO[3:0]	

*9 [WR Addr 3] Local Digital Input Interrupt trigger edge control

Meaning of bit value :

D15~D0 Local Digital Input Interrupt Control

Bit	Interrupt Source	Value Meaning	
[D15 D14]	LDI7	[0 0]	Disable Interrupt(default)
[D13 D12]	LDI6	[0 1]	Rising Edge Interrupt
[D11 D10]	LDI5	[1 0]	Falling Edge Interrupt
[D9 D8]	LDI4	[1 1]	Both Edge Interrupt
[D7 D6]	LDI3		
[D5 D4]	LDI2		
[D3 D2]	LDI1		
[D1 D0]	LDI0		

*10 [WR Addr 4] Double Functions Input & Timer Interrupt Control

Meaning of bit value :

D15 Reserved (filled with 0)

D14 Timer Interrupt Control

0 Disable (default)

1 Enable

D13~D0 Double Functions Input Interrupt Control

Bit	Interrupt Source	Value Meaning	
[D13 D12]	DFI6	[0 0]	Disable Interrupt(default)
[D11 D10]	DFI5	[0 1]	Rising Edge Interrupt
[D9 D8]	DFI4	[1 0]	Falling Edge Interrupt
[D7 D6]	DFI3	[1 1]	Both Edge Interrupt
[D5 D4]	DFI2		
[D3 D2]	DFI1		
[D1 D0]	DFI0		

*11 [WR Addr 5] Timer Value low word

Meaning of bit value :

D15~D0 Timer Value low word [15:0]

*12 [WR Addr 6] Timer Value high word

Meaning of bit value :

D15~D8 Reserved (filled with 0)

D7~D0 Timer Value high word [23:16]

*13 [WR Addr 7] Watch Dog Reset duration low word

Meaning of bit value :

D15~D0 Timer Value low word [15:0]



*14 [WR Addr 8] Watch Dog Reset duration high word

Meaning of bit value :

D15~D8 Reserved (filled with 0)

D7~D0 Timer Value high word [23:16]

*15 [WR Addr 9] Watch Dog timer value

Meaning of bit value :

D15~D0 Watch Dog Timer Value [15:0]

*16 [WR Addr 10] Refresh Watch Dog Timer

Meaning of bit value :

D15~D0 Reserved (filled with 0)