

PIO-D48 系列用户手册

48 数字量输出入通道卡

简体中文版, 版本 3.5, 2018 年 6 月

支援

包含 PIO-D48, PIO-D48U, PIO-D48SU, PEX-D48。

承诺

郑重承诺：凡泓格科技股份有限公司产品从购买即日起一年内无任何材料性缺损。

免责声明

凡使用本系列产品除产品质量所造成的损害，泓格科技股份有限公司不承担任何法律责任。泓格科技股份有限公司有义务提供本系列产品可靠而详尽数据，但保留修订权利，且不承担使用者非法利用数据对第三方所造成侵害构成的法律责任。

版权

版权所有© 2018 泓格科技股份有限公司，保留所有权力。

商标

手册中所涉及所有公司商标，商标名称及产品名称分别属于该商标或名称的拥有者所有。

与我们联系

如有任何问题欢迎联系我们，我们将会为您提供完善的咨询服务。

Email: service@icpdas.com, service.icpdas@gmail.com


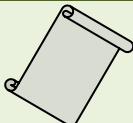

目录

产品清单	3
1. 绪论	4
1.1 特点	5
1.2 规格	6
2. 硬件结构	7
2.1 板卡布局	7
2.2 I/O 口位置	9
2.3 CARD ID 开关	10
2.4 引脚分配	11
2.5 启动 I/O 运作	12
2.6 DI/DO 口结构	13
2.7 中断运行	14
2.7.1 中断结构图	15
2.7.2 INT_CHAN_0	16
2.7.3 INT_CHAN_1	17
2.7.4 INT_CHAN_2	18
2.7.5 INT_CHAN_3	19
3. 安装硬件装置	20
4. 软件安装向导	24
4.1 开始安装使用--取得驱动安装程序	24
4.2 PNP 驱动程序安装	26
4.3 确认板卡安装成功	28
4.3.1 如何开启设备管理器	28
4.3.2 确认板卡是否正确安装	30
5. 测试 PIO-D48 系列卡	31
5.1 自我测试接线	31
5.1.1 PIO-D48(U) 及 PEX-D48	31
5.1.2 PIO-D48SU	32
5.2 执行测试程序	33
6. I/O 控制寄存器	36

6.1 如何找到 I/O 地址	36
6.2 分配 I/O 地址	39
6.3 I/O 地址映像	41
6.3.1 RESET\ 控制寄存器	42
6.3.2 AUX 状态寄存器	42
6.3.3 INT 屏蔽控制寄存器	43
6.3.4 中断极性控制寄存器	44
6.3.5 读/写 I/O 口	45
6.3.6 读/写 8254	47
6.3.7 读/写 Clock/Int 控制寄存器	49
6.3.8 读 Card ID 寄存器	49
7. 示例程序	50
7.1 WINDOWS DEMO 程序	50
7.2 DOS DEMO 程序	52
附录: 端子板	53
A1. DB-37, DN-37, DN-50 及 DN-100	53
A2. DB-8125	54
A3. ADP-37/PCI 及 ADP-50/PCI	54
A4. DB-24P/DB-24PD 光电隔离输入端子板	55
A5. DB-24R/DB-24RD 继电器输入端子板	56
A6. 端子板对照表	57

产品清单

硬纸盒包装内包括以下项目：

	一张 PIO-D48/D48U/D48SU, PEX-D48 系列板卡
	一张 快速入门指南
	一张 软件安装的 PCI 光盘

注意：

如发现产品包装内的配件有任何损坏或遗失，请保留完整包装盒及配件，尽快联系我们，我们将有专人快速为您服务。

1. 绪论

PIO-D48SU/PIO-D48U/PEX-D48 板卡是泓格新上市并符合 RoHS 环保规范的产品，新的 PIO-D48U/D48SU 及 PEX-D48 的设计可直接兼容于 PIO-D48。

PIO-D48U/D48SU 支持 3.3 V/5 V PCI bus 接口, PEX-D48 支持 PCI Express 接口。并提供 48 个符合 TTL 规范的数字输入输出信道,它由六个 8 位的双向 I/O 端口所组成,这些埠分别叫作埠 A(PA)、埠 B(PB)、埠 C(PC) 而埠 C 又可以被分成二个半宽度 (4-bit) 的埠,每个 Port 的初始设定皆为输入模式。

PIO-D48U/D48SU 及 PEX-D48 还有其他实用的功能,第一种是 Card ID 指拨开关,让使用者可以自由设定每张板卡的标识符。当系统同时使用多张 PIO-D48U/D48SU 或 PEX-D48 板卡时,使用者可以迅速而简单区别这些同型号的板卡。第二种是 DI Pull High/Low 设定功能,数字输入端口可设定为 pull-high 或 pull-low,当信号线脱落或断线时,该 DI 值会相对维持 High 或 Low 的状态(非浮动)。

此系列卡支持在 Linux、DOS、Windows 98/NT/2000、32/64-Bit Windows XP/2003/2008/7/8 及 Windows 10 等操作系统环境下使用,还提供有动态函数库及 Active X 控件使开发更加容易及简单易懂的各种语言范例程序,如 Turbo C++、Borland C++、Microsoft C++、Visual C++、Borland Delphi、Borland C++ Builder、Visual Basic、Visual C#.NET、Visual Basic.NET 及 LabVIEW 等,让用户能够快速的上手来使用。

1.1 特点

- PIO-D48 为 PCI bus 接口，支持 +5 V PCI bus 插槽
- PIO-D48U/D48SU 为 Universal PCI 接口，支持 +3.3 V 及 +5 V PCI bus 插槽
- PEX-D48 为 PCI Express 接口，支持 PCI Express x 1 插槽
- 最多可提供 48 个数字输入输出信道
- 双向 I/O 信道可以软件方式设定为 Input/Output 端口
- 内建 I/O line 缓冲区
- 六个 8-bit 埠可分别规化为 Input/Output
- 四个中断源
- 仿真两个工业标准的 8255 模式 0
- 高输出驱动能力
- PIO-D48U/D48SU: Digital I/O 反应速度最高可达 1 μ s (1 MHz)
- PEX-D48: Digital I/O 反应速度最高可达 500 KHz
- 一个 16 位的事件计数器
- 一个 32 位的内部程序计数器
- Card ID 功能为 PIO-D48U/D48SU/PEX-D48 仅有
- 数字输入端口可设定为 Pull-high 或 Pull-low
- 短卡设计
- 可直接连接 DB-24PR, DB-24PD, DB-24RD, DB-24PRD, DB-16P8R, DB-24POR, DB-24SSR, DB-24C 或者其他兼容 OPTO-22 规格的端子板

1.2 规格

板卡名称	PIO-D48 (停产)	PIO-D48U	PIO-D48SU	PEX-D48
可编程数字输入输出				
通道数	48			
数字输入				
兼容性	5 V/TTL			
输入电压	Logic 0: 0.8 V max.; Logic 1: 2.0 V min.			
响应速度	1 MHz			500 KHz
数字输出				
兼容性	5 V/TTL			
输出电压	Logic 0: 0.4 V max. ; Logic 1: 2.4 V min.			
输出能力	Sink: 64mA @ 0.8 V ; Source: 32 mA @ 2.0 V			
响应速度	1 MHz			500 KHz
公共				
总线型态	5 V PCI, 32-bit, 33 MHz	3.3 V/5 V Universal PCI, 32-bit, 33 MHz		PCI Express x1
数据总线	8-bit			
卡 ID	No	Yes(4-bit)		
I/O 连接头	Female DB37 x 1 50-pin box header x 1		Female SCSI II 100 pin x 1	Female DB37 x 1 50-pin box header x 1
尺寸(长 x 宽 x 高)	156 mm x 105 mm x 22mm		140 mm x 97 mm x 22 mm	172 mm x 112 mm x 22 mm
耗电量	900 mA @ +5 V			1500 mA @ +3.3 V 0 mA @ +12 V
运行温度	0 ~ 60 °C			
储存温度	-20 ~ 70 °C			
周围环境相对湿度	5 ~ 85%相对湿度, 非冷凝(non-condensing)			

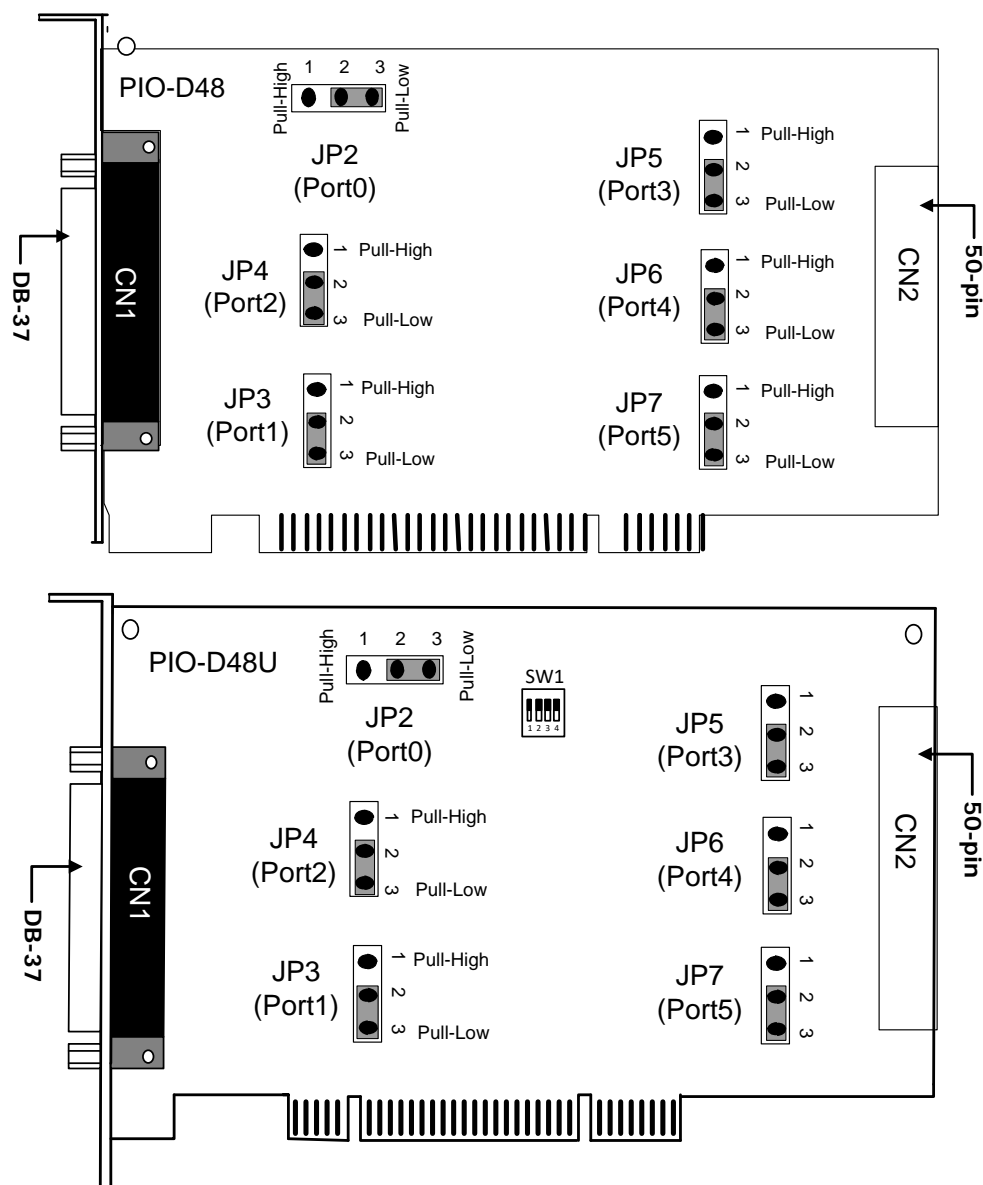
注意:

I/O 速度取决于 I/O 卡，总线速度，CPU 速度和系统负载。任何条件更改都可能导致 I/O 速度不同。

2. 硬件结构

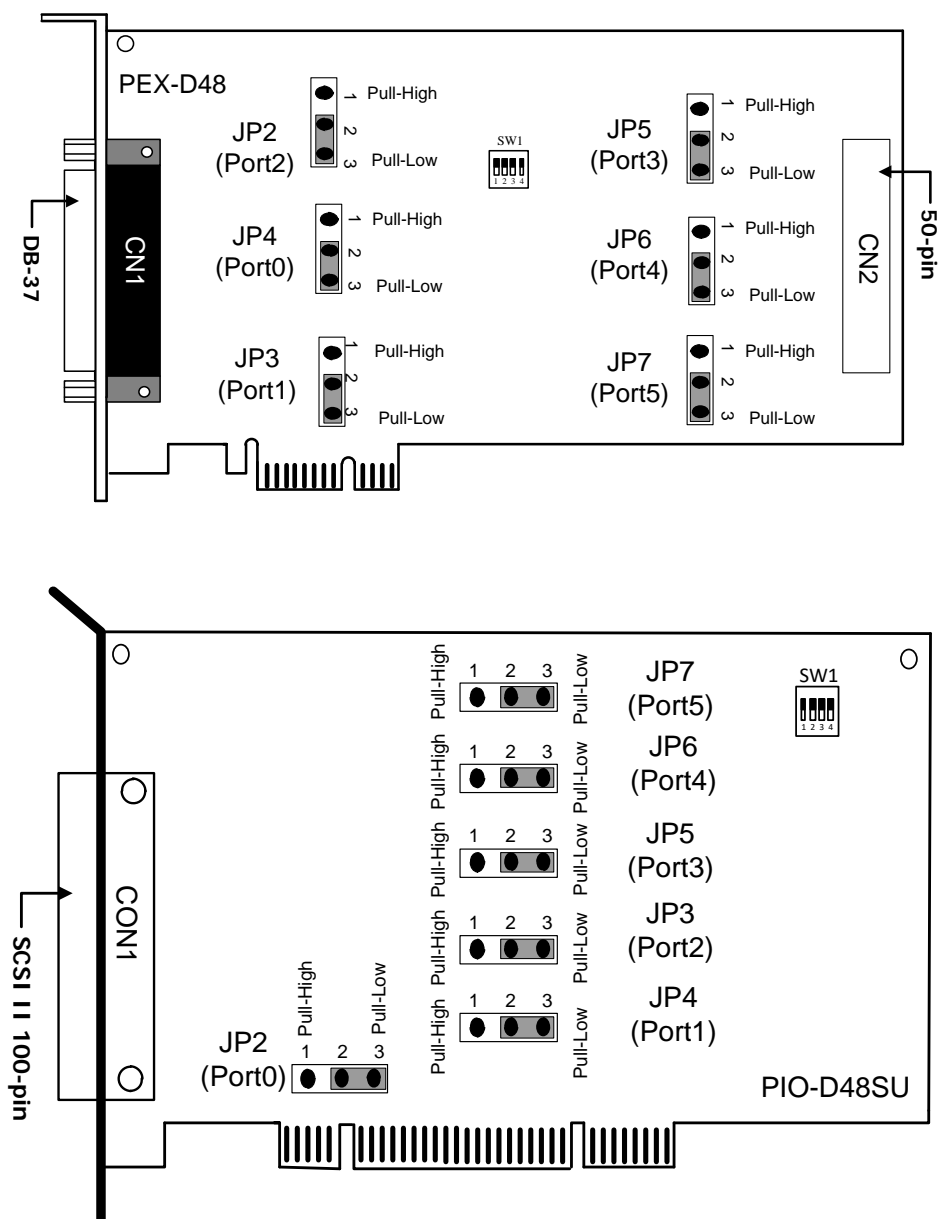
2.1 板卡布局

➤ PIO-D48/D48U 板卡布局:



注意: 预设设定: JP2/3/4/5/6/7=2-3 短接 = Pull-Low, 详细关于 DI Pull-high/low 讯息, 请参考 [第 2.2 节 “I/O 口位置”](#)。

➤ PEX-D48/PIO-D48SU 板卡布局:



注意: 预设设定: JP2/3/4/5/6/7=2-3 短接 = Pull-Low, 详细关于 DI Pull-high/low 讯息, 请参考 [章节 2.2 “I/O 口位置”](#)。

2.2 I/O 口位置

PIO-D48 系列卡具有 48 路 TTL 数字量输入输出通道，由 6 个 8 位双向口组成。当复位或上电时所有口均被设定为输入通道，并且这些通道通过跳线 JP2 ~ JP7 可选择置上升沿或下降沿。I/O 信道配置如下：

PIO-D48/D48U 连接头	CN1 (DB37)	CN2 (50-pin)
PA0 ~ PA7	Port0 (JP2 设定 pull-high/low)	Port3 (JP5 设定 pull-high/low)
PB0 ~ PB7	Port1 (JP3 设定 pull-high/low)	Port4 (JP6 设定 pull-high/low)
PC0 ~ PC7	Port2 (JP4 设定 pull-high/low)	Port5 (JP7 设定 pull-high/low)

PEX-D48 连接头	CN1 (DB37)	CN2 (50-pin)
PA0 ~ PA7	Port0 (JP4 设定 pull-high/low)	Port3 (JP5 设定 pull-high/low)
PB0 ~ PB7	Port1 (JP3 设定 pull-high/low)	Port4 (JP6 设定 pull-high/low)
PC0 ~ PC7	Port2 (JP2 设定 pull-high/low)	Port5 (JP7 设定 pull-high/low)

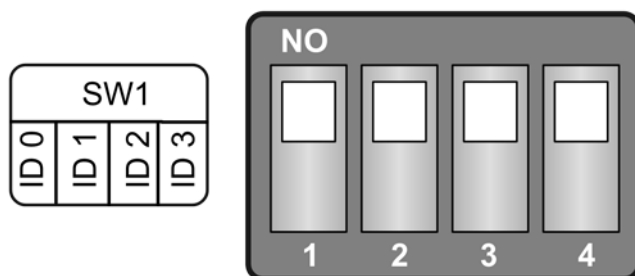
PIO-D48SU 连接头	CON1 (SCSI II 100-pin)
PA00 ~ PA07	Port0 (JP2 设定 pull-high/low)
PB00 ~ PB07	Port1 (JP3 设定 pull-high/low)
PC00 ~ PC07	Port2 (JP4 设定 pull-high/low)
PA10 ~ PA17	Port3 (JP5 设定 pull-high/low)
PB10 ~ PB17	Port4 (JP6 设定 pull-high/low)
PC10 ~ PC17	Port5 (JP7 设定 pull-high/low)

注意:

此板卡是双向 I/O 设计，供电后内定为 DI 模式。在正确切换至 DO 模式前，当 Jumper 设为 pull-high 时，该 DI 通道可能会造成 Active-High 的 DO 设备 (例:DB-24R/24PR/24C) 作动。或 Jumper 设为 pull-low 时，该 DI 通道可能会造成 Active-Low 的 DO 设备作动。请依外接设备特性选择适当的 DI pull-high/low Jumper 设定。

2.3 Card ID 开关

PIO-D48U/D48SU 及 PEX-D48 在硬件上新增 Card ID 指拨开关，让使用者可以自由设定每张板卡的识别码。当系统同时使用多张 PIO-D48U/D48SU 或 PEX-D48 板卡时，使用者可以迅速而简单区别这些同型号的板卡。出厂预设 Card ID 为 0x0。详细的 SW1 Card ID 设定，请参考至表 2.1。



(预设设定)

表 2.1 (*)预设设定; OFF → 1; ON → 0

Card ID (Hex)	1 ID0	2 ID1	3 ID2	4 ID3
(*) 0x0	ON	ON	ON	ON
0x1	OFF	ON	ON	ON
0x2	ON	OFF	ON	ON
0x3	OFF	OFF	ON	ON
0x4	ON	ON	OFF	ON
0x5	OFF	ON	OFF	ON
0x6	ON	OFF	OFF	ON
0x7	OFF	OFF	OFF	ON
0x8	ON	ON	ON	OFF
0x9	OFF	ON	ON	OFF
0xA	ON	OFF	ON	OFF
0xB	OFF	OFF	ON	OFF
0xC	ON	ON	OFF	OFF
0xD	OFF	ON	OFF	OFF
0xE	ON	OFF	OFF	OFF
0xF	OFF	OFF	OFF	OFF

2.4 引脚分配

PIO-D48 系列卡连接器引脚分配参考下图。

所有数字量输入或数字量输出信号均为 TTL 电相兼容。

Pin Assignment	Terminal No.	Pin Assignment
N.C.	01	20 +5 V
N.C.	02	21 GND
PB_7	03	22 PC_7
PB_6	04	23 PC_6
PB_5	05	24 PC_5
PB_4	06	25 PC_4
PB_3	07	26 PC_3
PB_2	08	27 PC_2
PB_1	09	28 PC_1
PB_0	10	29 PC_0
GND	11	30 PA_7
N.C.	12	31 PA_6
GND	13	32 PA_5
N.C.	14	33 PA_4
GND	15	34 PA_3
N.C.	16	35 PA_2
GND	17	36 PA_1
+5 V	18	37 PA_0
GND	19	

CN1 (Female DB-37)

Pin Assignment	Terminal No.	Pin Assignment
PC_7	01	02 GND
PC_6	03	04 GND
PC_5	05	06 GND
PC_4	07	08 GND
PC_3	09	10 GND
PC_2	11	12 GND
PC_1	13	14 GND
PC_0	15	16 GND
PB_7	17	18 GND
PB_6	19	20 GND
PB_5	21	22 GND
PB_4	23	24 GND
PB_3	25	26 GND
PB_2	27	28 GND
PB_1	29	30 GND
PB_0	31	32 GND
PA_7	33	34 GND
PA_6	35	36 GND
PA_5	37	38 GND
PA_4	39	40 GND
PA_3	41	42 GND
PA_2	43	44 GND
PA_1	45	46 GND
PA_0	47	48 GND
+5 V	49	50 GND

CN2 (50-pin box header)

Pin Assignment	Terminal No.	Pin Assignment
PA_00	01	51 PA_10
PA_01	02	52 PA_11
PA_02	03	53 PA_12
PA_03	04	54 PA_13
PA_04	05	55 PA_14
PA_05	06	56 PA_15
PA_06	07	57 PA_16
PA_07	08	58 PA_17
PB_00	09	59 PB_10
PB_01	10	60 PB_11
PB_02	11	61 PB_12
PB_03	12	62 PB_13
PB_04	13	63 PB_14
PB_05	14	64 PB_15
PB_06	15	65 PB_16
PB_07	16	66 PB_17
PC_00	17	67 PC_10
PC_01	18	68 PC_11
PC_02	19	69 PC_12
PC_03	20	70 PC_13
PC_04	21	71 PC_14
PC_05	22	72 PC_15
PC_06	23	73 PC_16
PC_07	24	74 PC_17
GND	25	75 GND
-	26	76 -
-	27	77 -
-	28	78 -
-	29	79 -
-	30	80 -
-	31	81 -
-	32	82 -
-	33	83 -
-	34	84 -
-	35	85 -
-	36	86 -
-	37	87 -
-	38	88 -
-	39	89 -
-	40	90 -
-	41	91 -
-	42	92 -
-	43	93 -
-	44	94 -
-	45	95 -
-	46	96 -
-	47	97 -
-	48	98 -
-	49	99 -
+ 5 V	50	100 + 5 V

CON1 (100-pin SCSI Connector)

2.5 启动 I/O 运作

当 PC 第一次运行，所有与运行有关的数字量 I/O 通道不可用。注意数字量 I/O 通道的每个口是激活或禁用是由 RESET\ 信号决定的，参考 [章节 6.3.1 “RESET\ 控制寄存器”](#) 讲述更多信息。电源开启状态所有 DI/DO 口如下：

- DI/DO 操作为每个口是禁用。
- DI/DO 口都配置成数字量输入口。
- DO 锁存寄存器未定义(参考至 [章节 2.6 “DI/DO 口结构”](#))。

用户在使用数字量 I/O 口之前需要做一些初始化工作。参照以下几个步骤：

步骤 1: 找到 PIO/PISO 板卡映像地址。(参考至 [章节 6.1 “如何找到 I/O 地址”](#))

步骤 2: 激活所有数字量 I/O 运行。(参考至 [章节 6.3.1 “RESET\ 控制寄存器”](#))

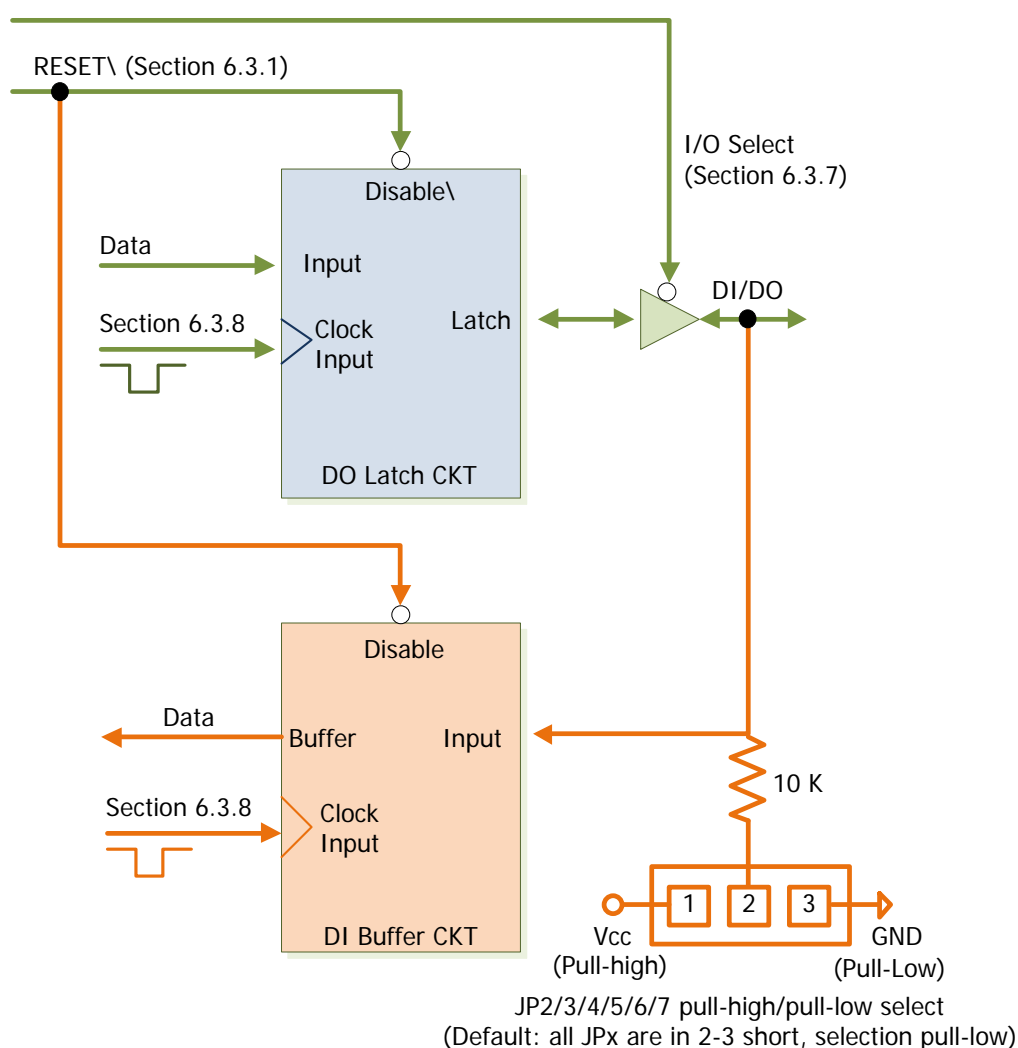
步骤 3: 设定第一个三个端口到指定的 D/I/O 状态并且发送初始值到每个 D/O 口。
(参考至 [章节 6.3.7 “读/写 Clock/Int 控制寄存器”](#))

注意：更多初始化数字量 I/O 端口信息，请参考 DEMO1.C 范例程序。

2.6 DI/DO 口结构

PIO-D48 系列卡数字量 I/O 口控制体系结构如下图，其控制信息操作方法如下：

- RESET\ 为低电平状态 → 所有 DI/DO 操作禁用。
- RESET\ 为高电平状态 → 所有 DI/DO 操作激活。
- 若 D/I/O 设置为 D/I 通道 → D/I= 通过跳线 JP2/3/4/5/6/7 选择外部输入信号为上升沿或下降沿 (1-2-ON=上升沿，2-3-ON=下降沿)。
- 若 DI/DO 设置为 DO 通道 → DI = 读回 DO 值。
- 若 DI/DO 设置为 DI 通道 → 数据发送到数字量输入通道时将仅仅只改变 DO 锁存寄存器，一旦通道设置成数字量输出时，储存在锁存寄存器的数据将被立即发送。



2.7 中断运行

板卡 PIO-D48 系列卡有四个中断源，分别为 INT_CHAN_0、INT_CHAN_1、INT_CHAN_2 及 INT_CHAN_3。其信号源参考如下：

INT_CHAN_0: PC3/PC7 from port-2 (参考[章节 2.7.2 “INT_CHAN_0”](#))

INT_CHAN_1: PC3/PC7 from port-5 (参考[章节 2.7.3 “INT_CHAN_1”](#))

INT_CHAN_2: Cout0 (参考[章节 2.7.4 “INT_CHAN_2”](#))

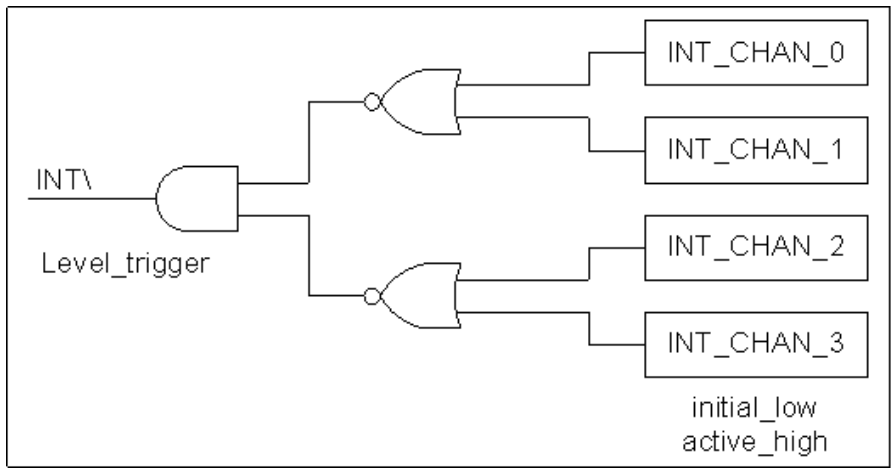
INT_CHAN_3: Cout2 (参考[章节 2.7.5 “INT_CHAN_3”](#))

注意：在 DOS 示例程序中 DEMO4.C、DEMO7.C、DEMO8.C、DEMO9.C 和 DEMO10.C 为单一信号中断源示例程序，DEMO11.C 为多信号中断源示例程序。若仅仅使用一个中断信号源，则中断服务程序没有必要分别中断源，但若有多多个中断源，那么中断服务程序必须按以下方式分辨激活的中断源：

1. 读取所有中断信号源最新状态（参考 [章节 6.3.2 “AUX 状态寄存器”](#)）
2. 对比信号源新旧状态，分别已激活信号。
3. 若 INT_CHAN_0 激活，则 INT_CHAN_0 和其同相反相信号均可使用。
4. 若 INT_CHAN_1 激活，则 INT_CHAN_1 和其同相反相信号均可使用。
5. 若 INT_CHAN_2 激活，则 INT_CHAN_2 和其同相反相信号均可使用。
6. 若 INT_CHAN_3 激活，则 INT_CHAN_3 和其同相反相信号均可使用。
7. 更新中断状态。

注意：如果中断信号太短，现在状态可能和原来状态相同，因此，在中断服务程序被执行前中断信号需要保持激活一段时间（Hold time）。这个保持时间在不同的操作系统是不同的，可能从数毫秒到 1 秒。在一般情况下，20 ms 对大部份操作系统都已足够。

2.7.1 中断结构图



`INT\` 中断输出信号为低电平触发器。若 `INT\` 产生一个低脉冲，则 **PIO-D48** 系列卡将使 **PC** 产生一次中断。若 `INT\` 设定为持续的低电平，那么 **PIO-D48** 系列卡将一直占用 **PC** 中断。因此 `INT_CHAN_0/1/2/3` 必须由脉冲信号控制，且稳定在低电平状态加以高电平脉冲发生来中断 **PC** 机。

由于 `INT_CHAN_0/1/2/3` 优先级相同，如果所有四个信号同时激活，那么 `INT\` 将仅仅只激活一次。因此，中断服务程序不得不读取多路中断的全通道的状态（详情请参考 [章节 2.7 “中断运行”](#)）。

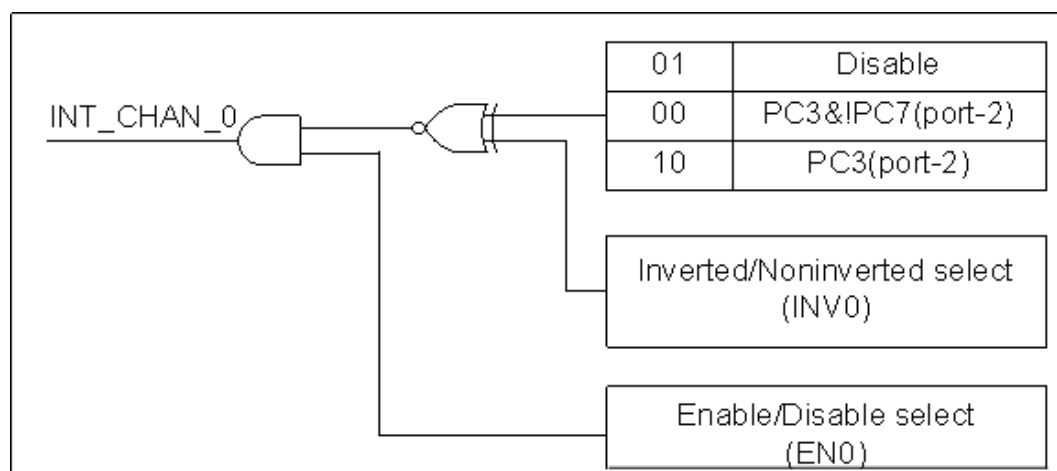
DEMO11.C	为 <code>INT_CHAN_0</code> 及 <code>INT_CHAN_1</code>
----------	---

而仅仅只有一个中断信号，则中断服务程序没有必要读取中断源状态。

注意，**DEMO4.C** 到 **DEMO5.C** 示例程序均为 **DOS** 操作系统下的单通道中断。

DEMO4.C	仅有 <code>INT_CHAN_3</code>
DEMO7.C	仅有 <code>INT_CHAN_2</code>
DEMO8.C	仅有 <code>INT_CHAN_0</code>
DEMO9.C	仅有 <code>INT_CHAN_0</code>
DEMO10.C	仅有 <code>INT_CHAN_1</code>

2.7.2 INT_CHAN_0



通常以低电平为标准，配合高电平脉冲的产生来中断 PC。

INT_CHAN_0 通过编程可等效于 PC3&!PC7 或 PC3，方法如下(详情请参考 [章节 6.3.7 “读/写 Clock/Int 控制寄存器”](#))：

- CTRL_D3=0, CTRL_D2=1 → INT_CHAN_0=禁用
- CTRL_D3=1, CTRL_D2=0 → INT_CHAN_0= port-2 的 PC3
- CTRL_D3=0, CTRL_D2=0 → INT_CHAN_0= port-2 的 PC3&IPC7

EN0 可用于激活/禁用 INT_CHAN_0，方法如下 (详情请参考 [章节 6.3.3 “INT 屏蔽控制寄存器”](#))：

- EN0=0 → INT_CHAN_0=禁用
- EN0=1 → INT_CHAN_0=激活

INV0 可用于设定 PC3 或 PC3&IPC7 同相/反相，方法如下 (详情请参考 [章节 6.3.4 “中断极性控制寄存器”](#))：

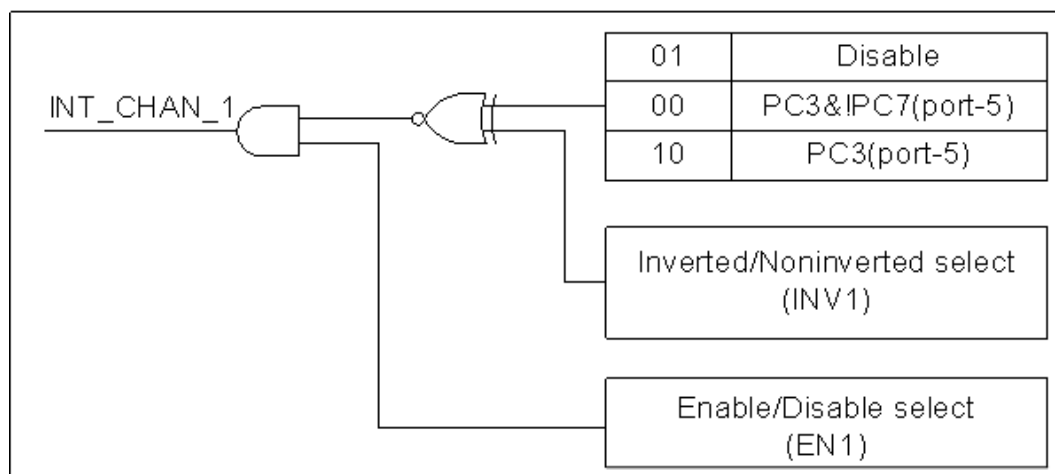
- INV0=0 → INT_CHAN_0=反相状态 (PC3 或 port-2 的 PC3&IPC7)
- INV0=1 → INT_CHAN_0=同相状态 (PC3 或 port-2 的 PC3&IPC7)

更多信息请参考下面 DOS 示例程序：

DEMO8.C 仅有 INT_CHAN_0 (Port-2 的 PC3)

DEMO9.C 仅有 INT_CHAN_0 (Port-2 的 PC3&IPC7)

2.7.3 INT_CHAN_1



INT_CHAN_1 通常以低电平为标准，配合高电平脉冲的产生来中断 PC。

INT_CHAN_1 通过编程可等效于 PC3&!PC7 或 PC3，方法如下(详情请参考 [章节 6.3.7 “读/写 Clock/Int 控制寄存器”](#))：

- CTRL_D5=0, CTRL_D4=1 → INT_CHAN_1=禁用
- CTRL_D5=1, CTRL_D4=0 → INT_CHAN_1= port-5 的 PC3
- CTRL_D5=0, CTRL_D4=0 → INT_CHAN_1= port-5 的 PC3&!PC7

EN1 可用于激活/禁用 INT_CHAN_0，方法如下 (详情请参考[章节 6.3.3 “INT 屏蔽控制寄存器”](#))：

- EN1=0 → INT_CHAN_1=禁用
- EN1=1 → INT_CHAN_1=激活

INV1 可用来设定 PC3 或 PC3&!PC7 同相/反相，方法如下 (详情请参考 [章节 6.3.4 “中断极性控制寄存器”](#))：

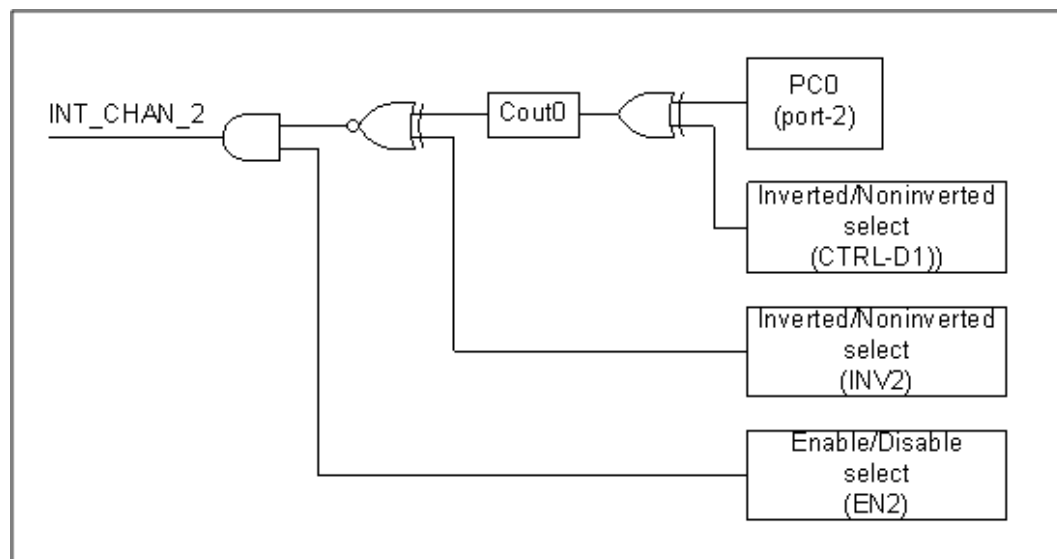
- INV1=0 → INT_CHAN_1=反相状态 (PC3 或 port-5 的 PC3&!PC7)
- INV1=1 → INT_CHAN_1=同相状态 (PC3 或 port-5 的 PC3&!PC7)

更多信息请参考下面 DOS 示例程序:

DEMO10.C	仅有 INT_CHAN_1 (Port-5 的 PC3&!PC7)
----------	-----------------------------------

注意：高电平脉冲产生，参考 [章节. 2.7.2 “INT_CHAN_0”](#)。

2.7.4 INT_CHAN_2



INT_CHAN_2 通常以低电平为标准，配合高电平脉冲的产生来中断 PC。

PC0 (port-2) 可通过编程设定反向/同相状态，方法如下：（参考 [章节 6.3.7 “读/写 Clock/Int 控制寄存器”](#)）

- CTRL_D1=0 → Cin0 = port-2 的 PC0
- CTRL_D1=1 → Cin0 = port-2 的!PC0

EN2 可用于激活/禁用 INT_CHAN_2，方法如下（参考 [章节 6.3.3 “INT 屏蔽控制寄存器”](#)）：

- EN2=0 → INT_CHAN_2 = 禁用
- EN2=1 → INT_CHAN_2 = 激活

INV2 用来设定 Cout0 同相/反相状态，方法如下（参考 [章节 6.3.4 “中断极性控制寄存器”](#)）：

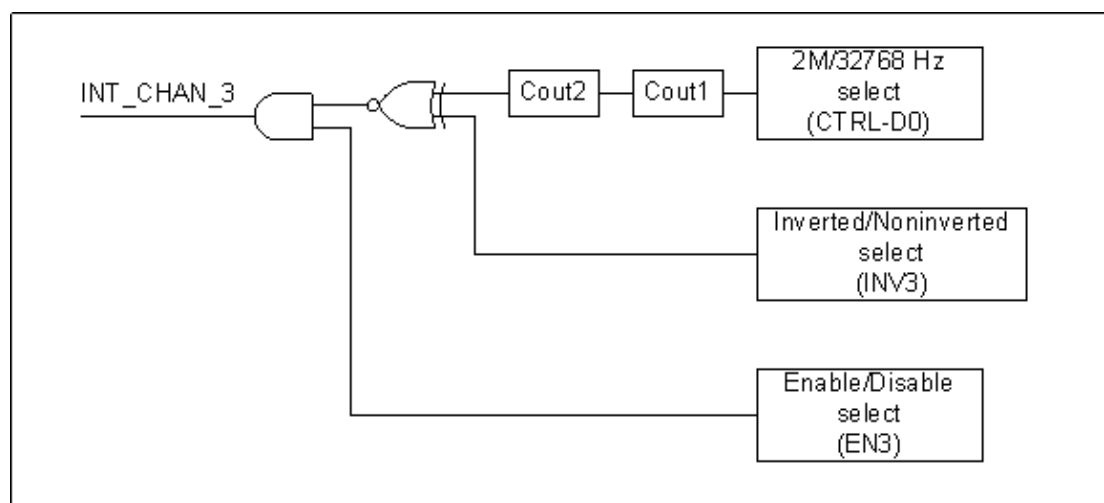
- INV2=0 → INT_CHAN_2 = 反相状态 (Cout0)
- INV2=1 → INT_CHAN_2 = 同相状态 (Cout0)

更多信息请参考下面 DOS 示例程序：

DEMO7.C	仅有 INT_CHAN_2 (Cout0)
---------	-----------------------

注意：高电平脉冲产生，参考 [章节. 2.7.2 “INT_CHAN_0”](#)。

2.7.5 INT_CHAN_3



INT_CHAN_3 通常以低电平为标准，配合高电平脉冲的产生来中断 PC。

Cin1 可通过编程设定 2MHz 或 32768Hz 时钟脉冲源，方法如下：（参考 [章节 6.3.7 “读/写 Clock/Int 控制寄存器”](#)）

- CTRL_D0=0 → Cin1 = 2 MHz 时钟脉冲源
- CTRL_D0=1 → Cin1 = 32768 Hz 时钟脉冲源

EN3 可用于激活/禁用 INT_CHAN_3，方法如下（详情请参考 [章节 6.3.3 “INT 屏蔽控制寄存器”](#)）：

- EN3=0 → INT_CHAN_3 = 禁用
- EN3=1 → INT_CHAN_3 = 激活

INV2 可用来设定 Cout0 同相/反相状态，方法如下（详情请参考 [章节 6.3.4 “中断极性控制寄存器”](#)）：

- INV2=3 → INT_CHAN_3 = 反相 (Cout2)
- INV2=3 → INT_CHAN_3 = 同相 (Cout2)

更多信息请参考下面 DOS 示例程序：

DEMO4.C	仅有 INT_CHAN_3 (Cout2)
---------	-----------------------

注意：高电平脉冲产生，参考 [章节. 2.7.2 “INT_CHAN_0”](#)。

3. 安装硬件装置

注意：建议先安装软件驱动程序，因为有些操作系统（如，Windows 2000）可能会要求您重新启动计算机。因此可减少您重新启动计算机开机的次数。

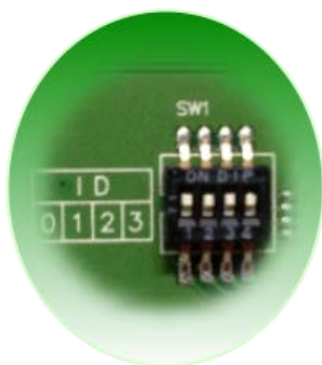
依照下列步骤来完成安装：

步骤 1：安装 PIO-D48 系列卡的软件驱动程序。



详细软件驱动程序安装信息，请参考至 [章节 4 “软件安装向导”](#)。

步骤 2：设定 SW1 DIP-Switch 来配置 Card ID。

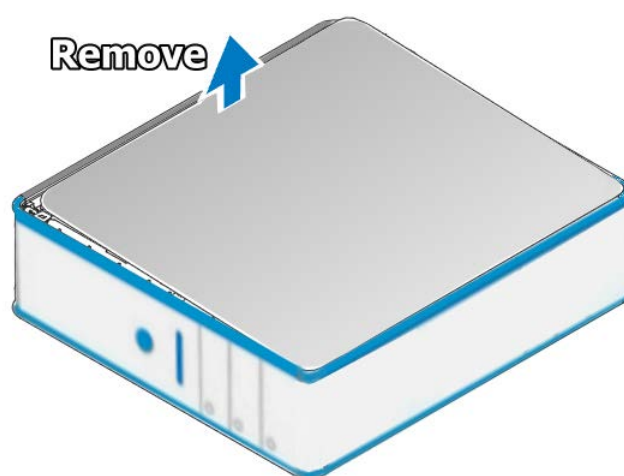


详细 Card ID (SW1) 设定，请参考至 [章节 2.3 “Car ID 开关”](#)。

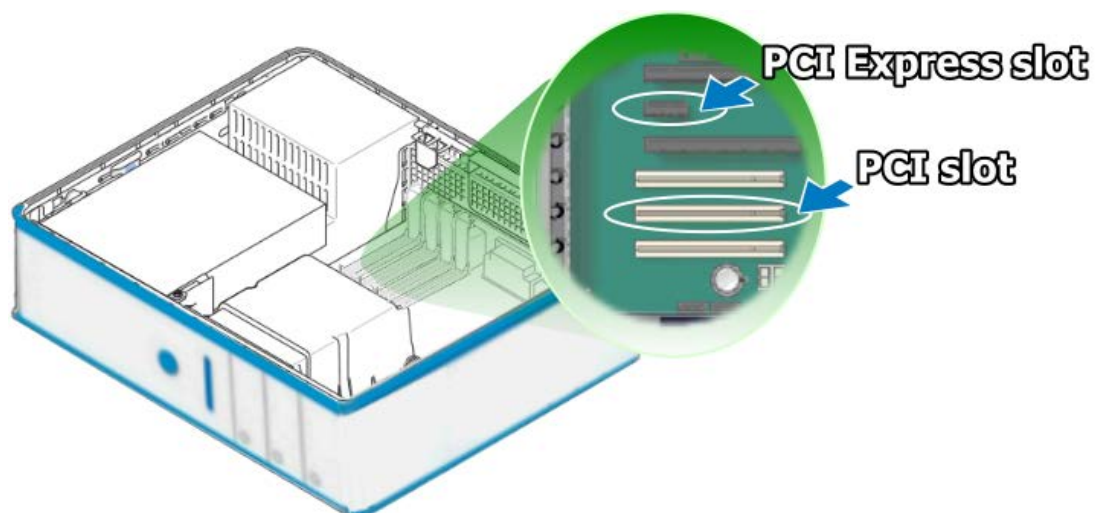


步骤 3: 关掉计算机电源。

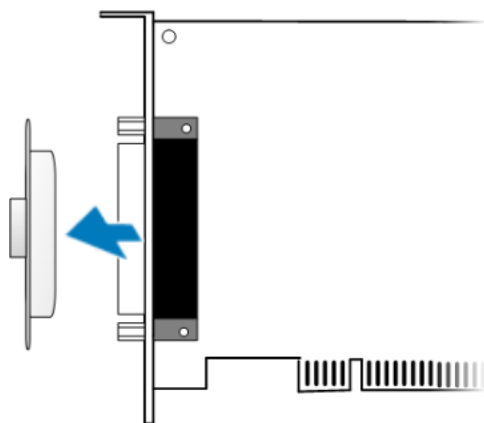
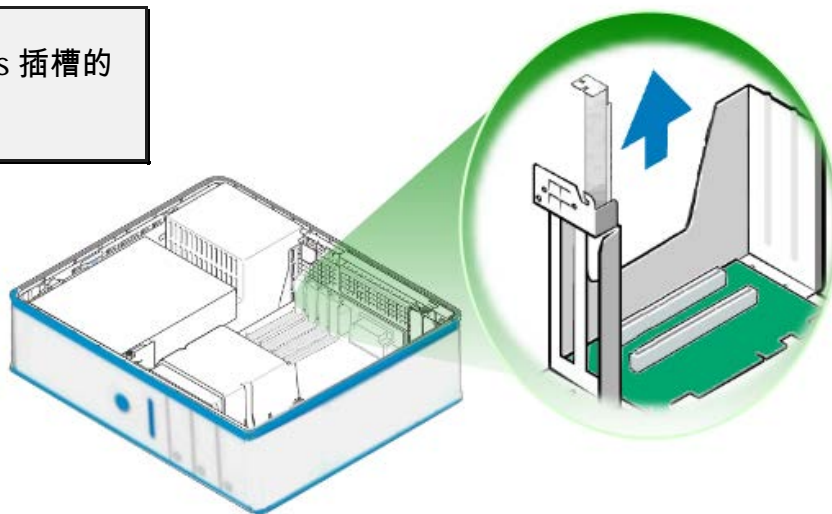
步骤 4: 打开计算机机壳。



步骤 5: 选择未使用的 PCI/PCI Express 插槽。

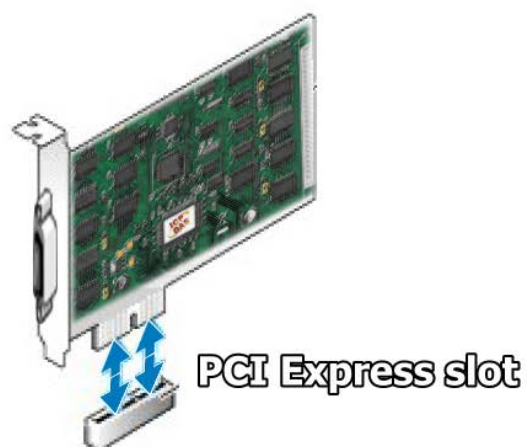
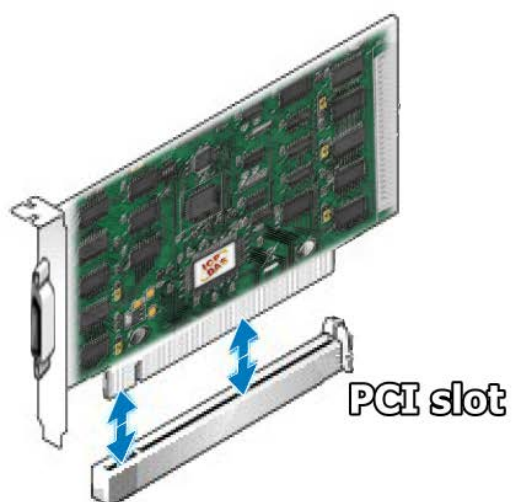


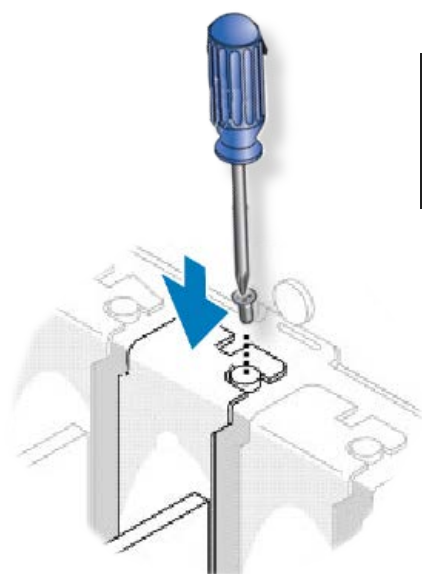
步骤 6: 移除 PCI/PCI Express 插槽的
保护装置。



步骤 7: 移除 PIO-D48 系列卡接头上的保护装
置。

步骤 8: 小心插入 PIO-D48 系列卡至 PCI/PCI Express 插槽。

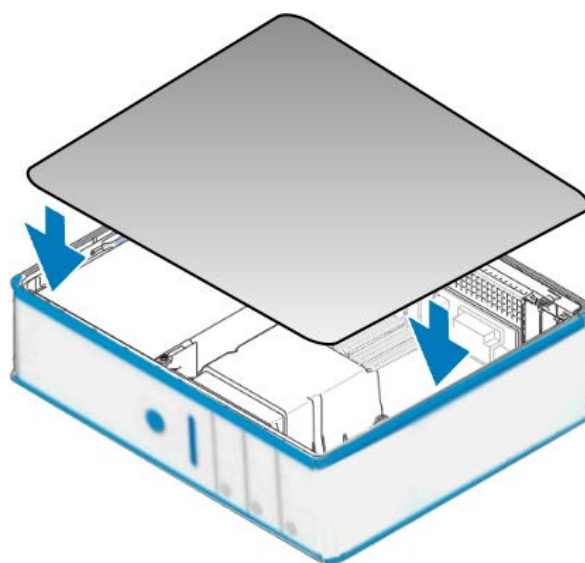




步骤 9：并以螺丝固定住。

确认 PIO-D48 系列卡已正确且牢固的安装在计算机主板上。

步骤 10：装回计算机机壳。



步骤 11：启动计算机电源。



进入 Windows 后，请依照提示讯息完成即插即用驱动安装，请参考至 [章节 4 “软件安装向导”](#)。

4. 软件安装向导

PIO-D48 系列板卡支持在 DOS、32-/64-bit Windows XP/2003/2008/7/8 及 Windows 10 等操作系统环境下使用。本章节将详细介绍如何取得驱动安装执行档、驱动安装程序以及验证板卡是否正确安装...等。

4.1 开始安装使用--取得驱动安装程序

PIO-D48 系列卡驱动程序安装执行文件，可从随机出货的配件 CD 软件光盘中或从泓格的软件网站中下载。请参考表 4-1 及 4-2 来选择适当的驱动程序。

表 4-1: UniDAQ Driver/SDK (建议新用户安装此驱动程序)

操作系统	Windows 2000, 32/64-bit Windows XP, 32/64-bit Windows 2003, 32/64-bit Windows Vista, 32/64-bit Windows 7, 32/64-bit Windows 2008, 32/64-bit Windows 8, 32/64-bit Windows 10
名称	UniDAQ Driver/SDK (unidaq_win_setup_xxxx.exe)
CD-ROM	CD:\\ NAPDOS\\PCI\\UniDAQ\\DLL\\Driver\\
下载网站	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/driver/
安装程序	详细 UniDAQ 驱动程序安装，可参考至 UniDAQ DLL 软件使用手册。 手册下载位置： CD:\\NAPDOS\\PCI\\UniDAQ\\Manual\\ http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/manual/

表 4-2: PIO-DIO Series Classic Driver (建議已安裝使用過 PIO-DIO 系列卡的原用戶使用此驅動程式)

操作系统	Windows 95/98/ME, Windows NT, Windows 2000, 32-bit Windows XP, 32-bit Windows 2003, 32-bit Windows Vista, 32-bit Windows 7, 32-bit Windows 8, 32-bit Windows 10
名称	PIO-DIO Series Classic Driver(PIO_DIO_Win__vxxx.exe)
CD-ROM	CD:\\ NAPDOS\\PCI\\PIO-DIO\\DLL_OCX\\Driver\\
下载网站	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/driver/
安装程序	<p>详细 PIO-DIO 驱动程序安装，可参考至 PIO-DIO DLL 软件使用手册。</p> <p>手册下载位置:</p> <p>CD:\\NAPDOS\\PCI\\PIO-DIO\\Manual\\</p> <p>http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/driver/</p>

4.2 PnP 驱动程序安装

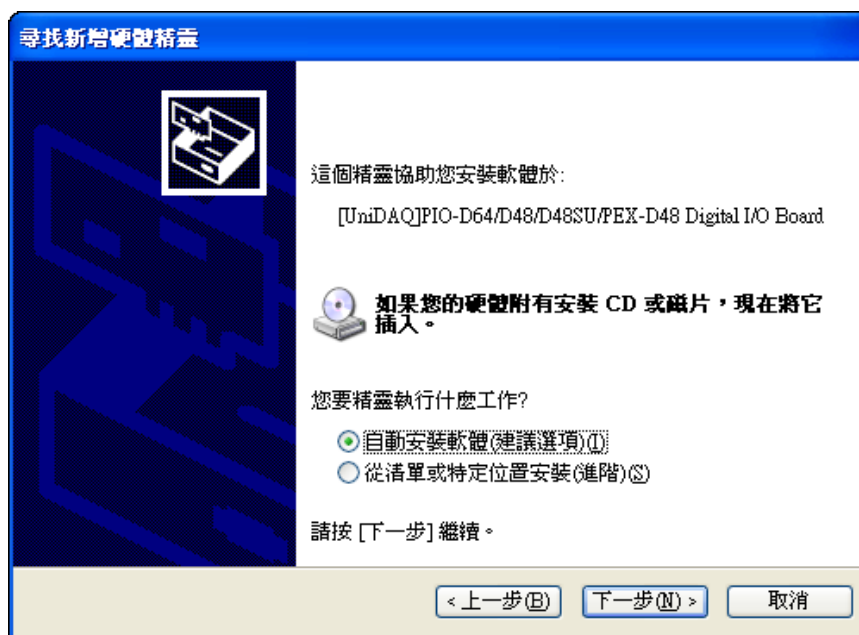
步骤 1: 关闭计算机电源，并安装 PIO-D48 系列卡至计算机中。

详细 PIO-D48 系列卡硬件安装，请参考至 [章节 3 “安装硬件装置”](#)。

步骤 2: 开启计算机电源来完成即插即用驱动安装。

注意: 有些作系统 (如, **Windows 7/8/10**) 会找到新硬件后, 将自动完成即插即用驱动安装, 因此将会跳过步骤 3 到步骤 5。

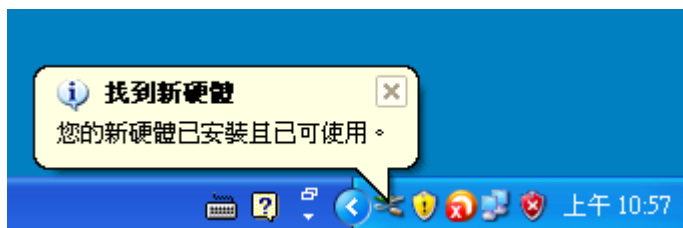
步骤 3: 选择 “自动安装软件 (建议选项)(I)” 后, 按 “下一步(N)>” 按钮到下一个画面。



步骤 4: 按下“完成”按钮，来完成安装。



步骤 5: 显示“您的新硬件已安装且已可使用”讯息。



4.3 确认板卡安装成功

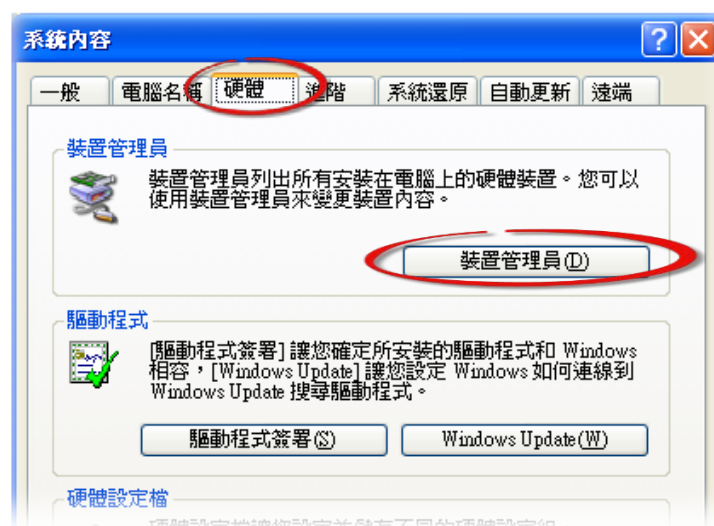
请到装置管理员中来确认您的 PIO-D48 系列板卡已正确的安装到 PC 中，依照您的操作系统，参考至下列来开启您的装置管理员。

4.3.1 如何开启设备管理器

➤ Microsoft Windows 2000/XP

步骤 1: 单击“开始”→“控制台(C)”，开启控制台后，再双击“系统”icon 来开启“系统内容”配置框。

步骤 2: 单击“硬件”标签后，再单击“设备管理器(D)”按钮。



➤ Microsoft Windows 2003

步骤 1: 单击“开始”→“系统管理工具”→“计算机管理”。

步骤 2: 在“系统工具”控制台树中，单击“设备管理器”。



➤ Microsoft Windows 7/10

步骤 1: 单击“开始”→“控制台(C)”→“系统及安全性”。

步骤 2: 然后在“系统”下方单击“设备管理器”。

或者是，

步骤 1: 单击“开始 Start”按钮。

步骤 2: 在搜寻字段中输入设备管理员，再按 Enter 键。

注意：您必须以系统管理员的身份登入，才能变更「设备管理器」内的设定。其他使用者可以检视设定，但无法进行变更。

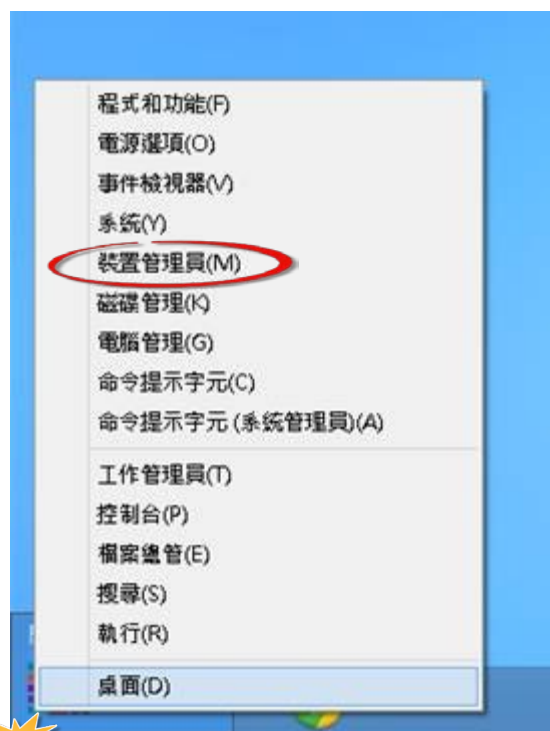


➤ Microsoft Windows 8

步骤 1: 将鼠标移至左下角，在出现“开始”的小图标上按鼠标右键。

步骤 2: 在功能列表中点选“设备管理器”。

或者是，可按快速组合键 [Windows Key] + [X] 来开启功能列表，在点选“设备管理器”。



4.3.2 确认板卡是否正确安装

检查 PIO-D48 系列板卡是否正确安装，如已安装完成，装置管理员中将显示 PIO-D48 板卡名称于 DAQCard 项目下，如下图所示：



5. 测试 PIO-D48 系列卡

此章节将详细介绍自我测试步骤。您可依照下列步骤来确认 PIO-D48 系列卡是否能正常启动。在自我测试前，您必须先完成软件驱动程序及硬件的安装。详细软硬件安装信息请参考至 [第 3 章“安装硬件装置”](#) 及 [第 4 章“软件安装向导”](#)。

5.1 自我测试接线

5.1.1 PIO-D48(U) 及 PEX-D48

在自我测试接线前，请先准备下列项目：

☒ A CA-3710 Cable

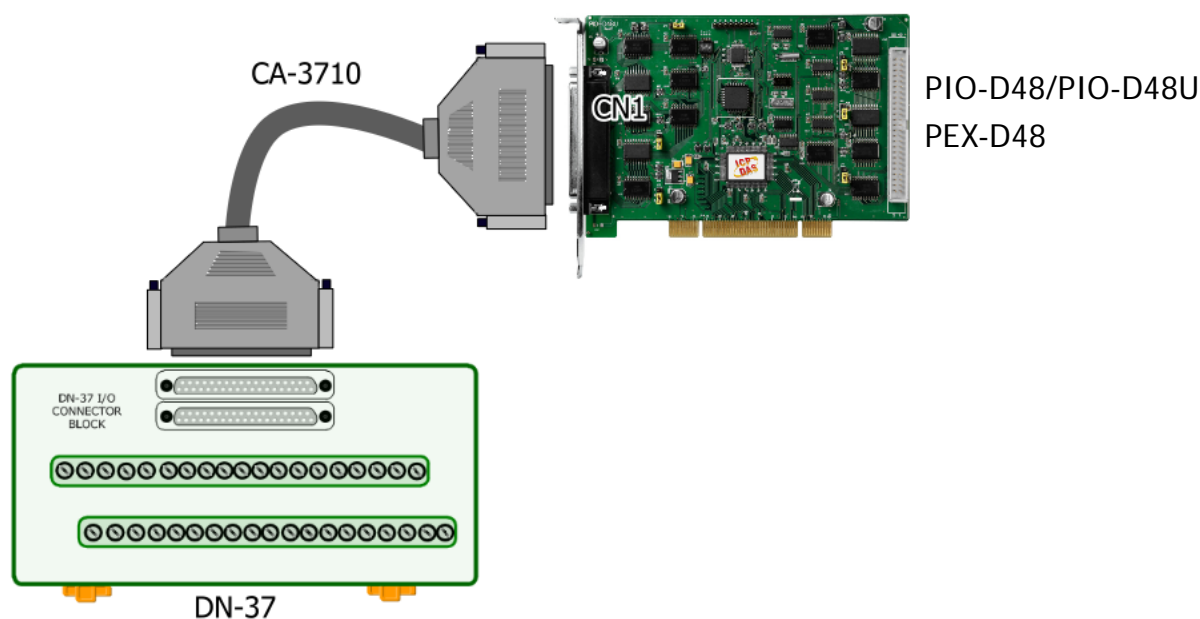
(选购品，产品网页 http://www.icpdas.com/products/Accessories/cable/cable_selection.htm)

☒ A DN-37 接线端子板

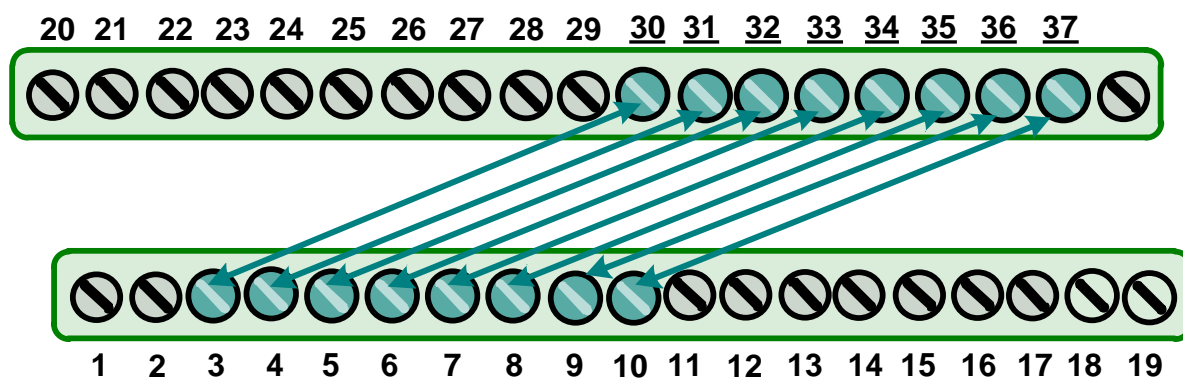
(选购品，产品网页

http://www.icpdas.com/root/product/solutions/pc_based_io_board/daughter_boards/dn-37.html)

步骤 1: 使用 CA-3710 Cable 将 DN-37 (选购品) 连接至 PIO-D48(U)/PEX-D48 卡的 CON1



步骤 2: 将 Port0 (PA0 ~ PA7) 连接至 Port1 (PB0 ~ PB7)。



5.1.2 PIO-D48SU

在自我测试前，请先准备下列项目：

☒ A CA-SCSI100-15 Cable

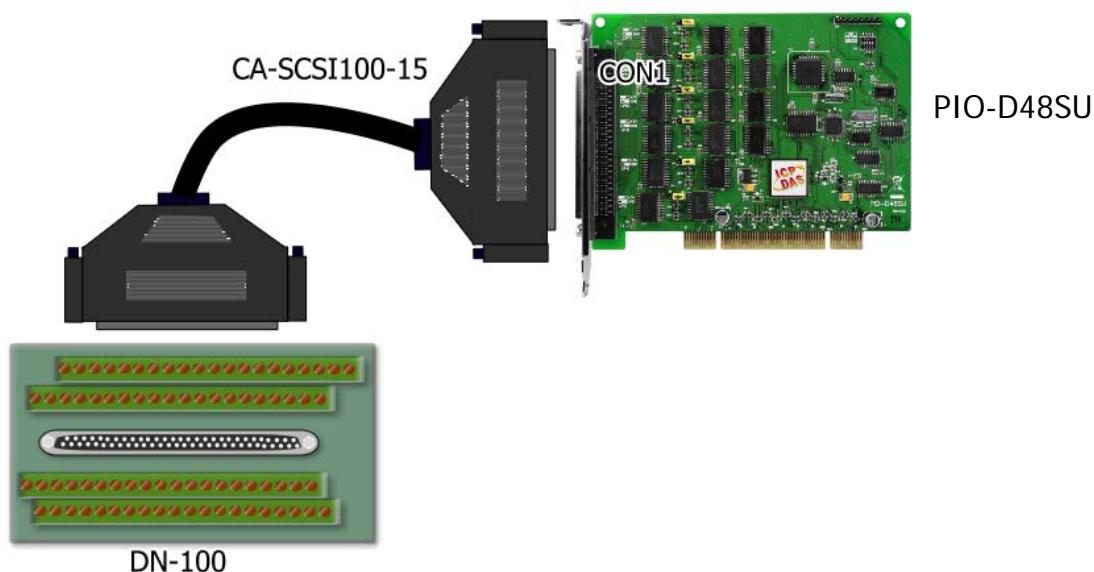
(选购品，产品网页 http://www.icpdas.com/products/Accessories/cable/cable_selection.htm)

☒ A DN-100 接线端子板

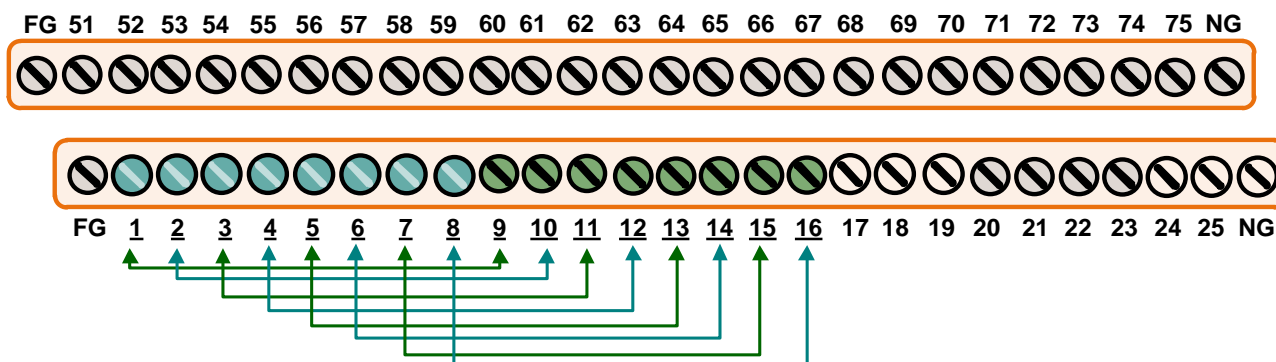
(选购品，产品网页

http://www.icpdas.com/root/product/solutions/pc_based_io_board/daughter_boards/dn-100.html)

步骤 1: 使用 CA-SCSI100-15 Cable 将 DN-100 (选购品) 连接至 PIO-D48SU 卡的 CON1。



步骤 2: 将 Port0 (PA00 ~ PA07) 连接至 Port1 (PB00 ~ PB07)。

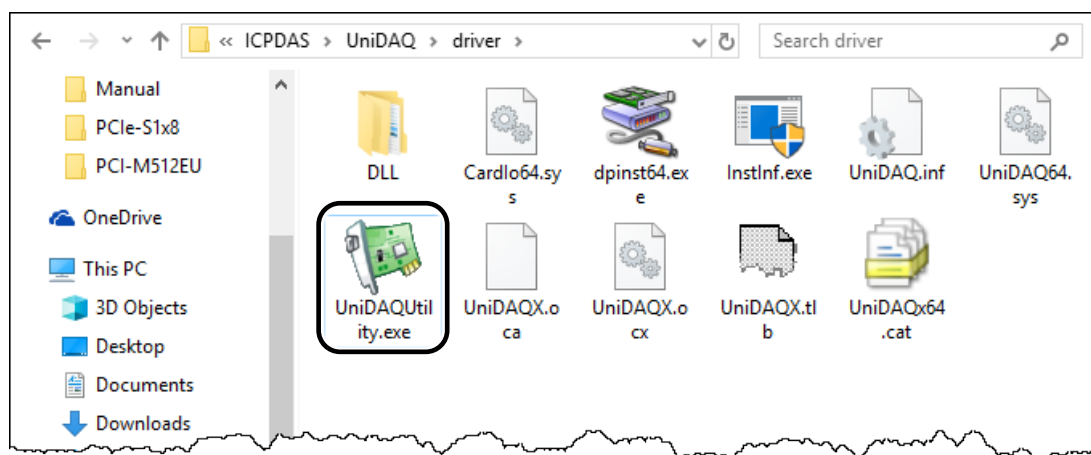


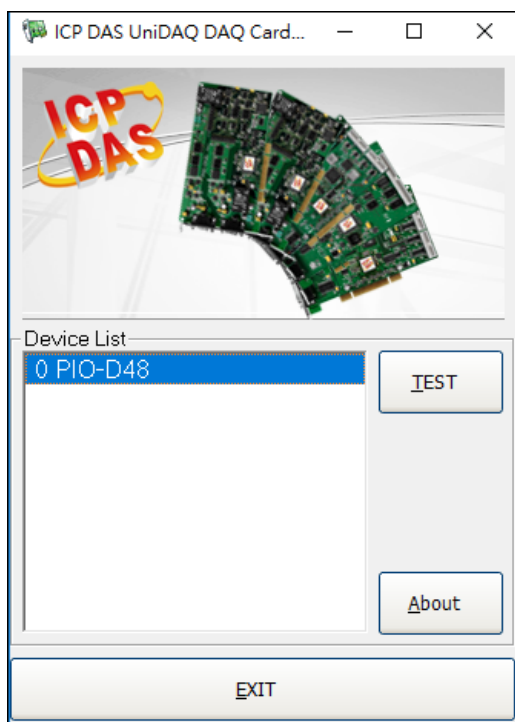
5.2 执行测试程序

下面自我测试范例为安装 UniDAQ 驱动程序。如果您安装的驱动程序为 PIO-DIO series classic，请参考至 PIO-D48 的快速入门指南来执行自我测试步骤。

(http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/manual/quickstart/d48/pio-d48_quickstart_cht.pdf)

步骤 1: 双击 “UniDAQUtility.exe”，开始进行测试。在完成 UniDAQ 驱动程序后，此 UniDAQ Utility.exe 将被安装放置默认路径 C:\ICPDAS\UniDAQ\Driver\下。





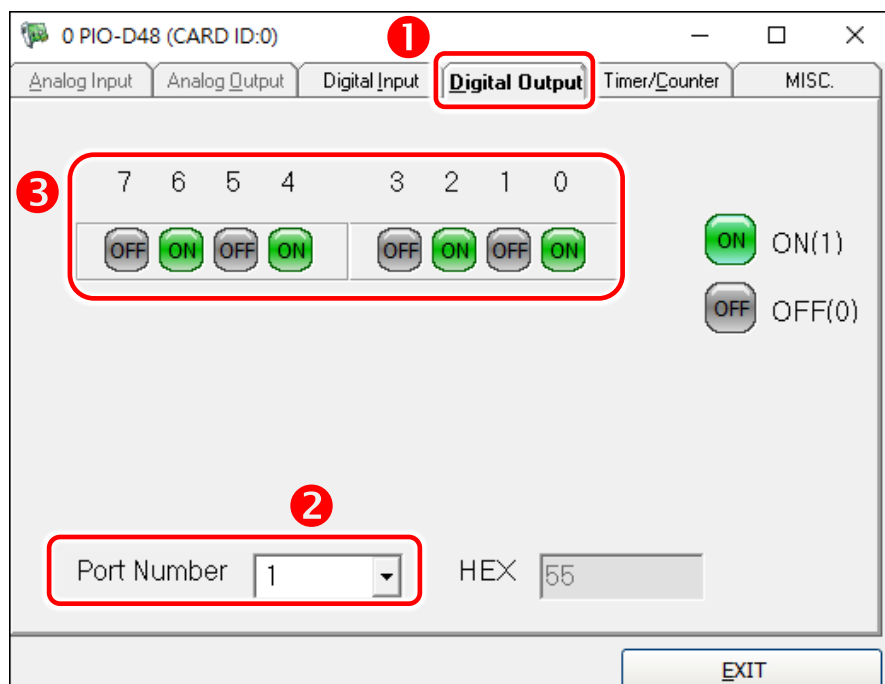
步骤 2: 确认一张 PIO-D48 系列板卡成功安装至计算机上。注意：数字 0 为第一张卡。

步骤 3: 按下 “**TEST**” 按钮，开始测试。

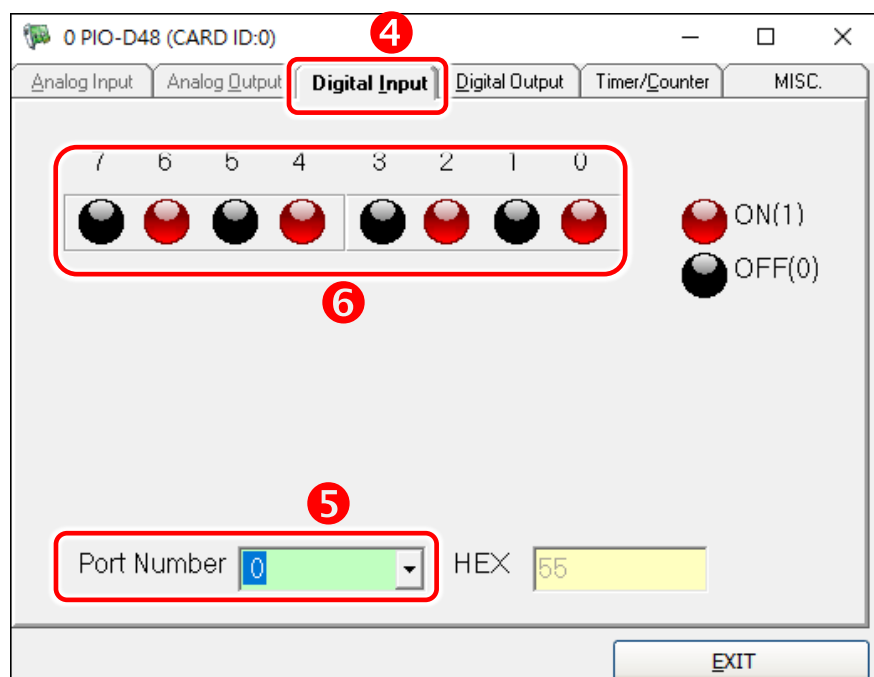
注意: PEX-D48、PIO-D48SU、PIO-D48U 的软件完全与 PIO-D48 软件兼容。

步骤 4: DIO 功能测试结果。

1. 点选 “**Digital Output**” 项目。
2. 从 “**Port Number**” 下拉式选单，选择 “**Port 1**”。
3. 点选 **DO** 通道 0, 2, 4, 6 为 **ON** 起。



4. 点选 “Digital Input” 项目。
5. 从 “Port Number” 下拉式选单，选择 “Port 0”。
6. DO 相对应的 DI 通道 0, 2, 4, 6 需显示为 High 的状态 (红灯亮起)。



6. I/O 控制寄存器

6.1 如何找到 I/O 地址

在上电后即插即用 BIOS 将分配适当的一个 I/O 地址到每个 PIO/PISO 板卡。

PIO-D48 系列板卡 ID 如下：

表 6-1:

PIO-D48(U) (Rev 1.x)		PIO-D48(U)/PEX-D48 (Rev 2.0 或更新版本)	
Vendor ID	0xE159	Vendor ID	0xE159
Device ID	0x0002	Device ID	0x0001
Sub-Vendor ID	0x80	Sub-Vendor ID	0x0080
Sub-Device ID	0x01	Sub-Device ID	0x01
Sub-Aux ID	0x30	Sub-Aux ID	0x30

表 6-2:

PIO-D48SU (Rev 1.0 或更新版本)	
Vendor ID	0xE159
Device ID	0x0001
Sub-Vendor ID	0x0080
Sub-Device ID	0x0001
Sub-Aux ID	0x0030

- 使用列下功能函数能够让您识别 PIO/PISO 系列板卡各 Sub IDs:
 - PIO_DriverInit(&wBoard, wSubVendor, wSubDevice, wSubAux)
 - PIO_GetConfigAddressSpace(wBoardNo, *wBase, *wlrq, *wSubVendor, *wSubDevice, *wSubAux, *wSlotBus, *wSlotDevice)
 - Show_PIO_PISO(wSubVendor, wSubDevice, wSubAux)

详细功能函数定义及说明，请参考至 PIO-DIO DLL 软件使用手册或 [章节 6.3 “I/O 地址映像”](#)。
以下列举重要的驱动程序功能参数说明：

- 资源分配信息：
 - **wBase:** 板卡的基础地址
 - **wlrq:** 板卡正在使用的 IRQ
- PIO/PISO 识别板卡信息：
 - **wSubVendor:** 板卡的 subVendor ID
 - **wSubDevice:** 板卡的 subDevice ID
 - **wSubAux:** 板卡的 subAux ID
- PC 插槽信息：
 - **wSlotBus:** 板卡的 Slot Bus 编码值
 - **wSlotDevice:** 板卡的 Slot Device ID 值

使用 PIO_PISO.EXE utility 工具程序，能够更便利且快速来检测并显示安装在计算机主机中所有的 PIO/PISO 系列卡，更详细讯息参考至下页 [PIO PISO.EXE Utility](#)。

➤ PIO_PISO.EXE Utility

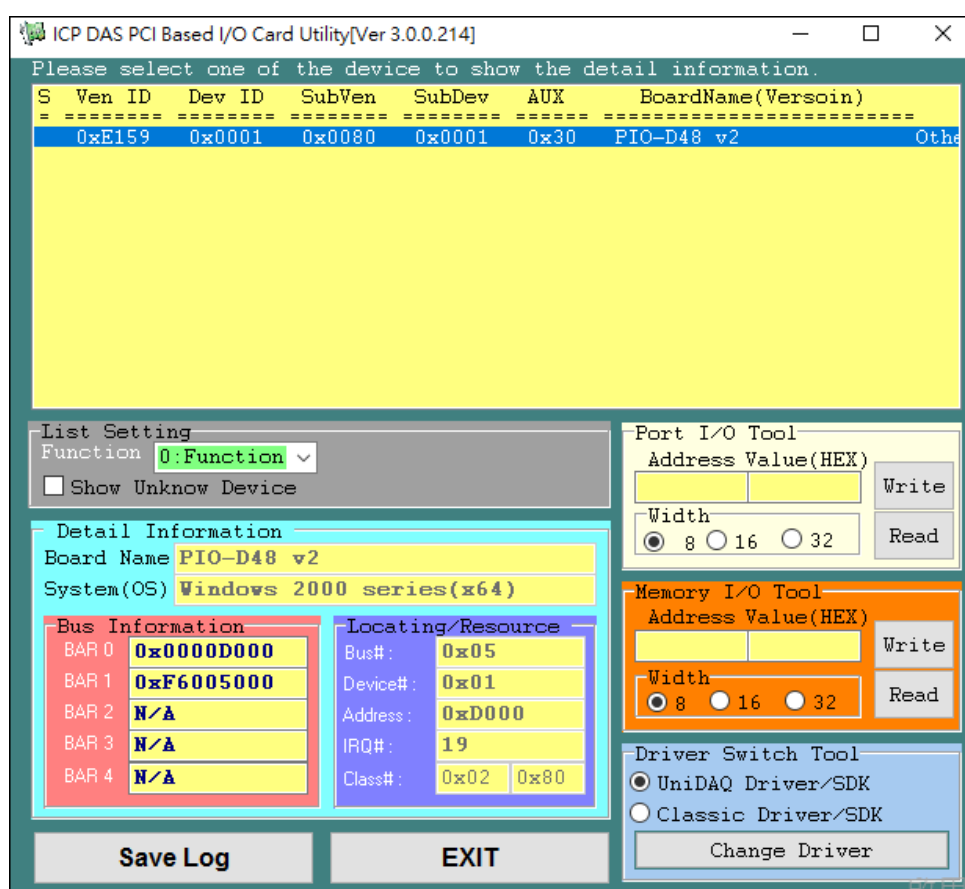
工具程序 PIO_PISO.EXE 适用于所有的 PIO/PISO 系列卡，且能够显示 PIO/PISO 系列卡安装在计算机主机上的所有硬件信息 (如: Sub-Vender, Sub-Device 和 Sub-Aux ID)，具体参数参考表 6-1 及表 6-2。如果 PIO_PISO.EXE utility 找不到 PIO/PISO 系列卡时，请尝试使用另一个 PCI 插槽，然后再开启 PIO_PISO.EXE utility 搜寻一次。

- 执行 PIO_PISO.EXE utility 将可取得下列信息:
- 显示安装在计算机主机上的所有 PIO/PISO 系列卡
- 显示分配给每个 PIO/PISO 系列卡的所有资源
- 显示 PIO/PISO 系列卡的 wSlotBus 及 wSlotDevice 识别值

PIO_PISO.exe utility 具程序下载位置:

CD:\NAPDOS\PCI\Utility\Win32\PIO_PISO\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/utility/win32/pio_piso/



6.2 分配 I/O 地址

新用户可使用 **Card ID** 功能来识别板卡。

即插即用 BIOS 将分配 PIO/PISO 板卡适当的 I/O 地址。假如只有一块 PIO/PISO 卡，用户能够确定这块卡为 **card_0**。如果有两个 PIO/PISO 卡在系统中，用户将很难找到哪一块卡是 **card_0**。软件驱动最多能够支持 16 块卡。所以用户可以安装 16 块 PIO/PISO 板卡在一台 PC 中。下面的方法说明查找和确定 **card_0** 和 **card_1**：

通过主板 ROM BIOS 去自动分配 PIO/PISO 板卡的 I/O 地址。用户同样可以再分配 I/O 地址。强烈推荐用户不要去改变 I/O 地址，即插即用 BIOS 将会很好的去分配每个 PIO/PISO 板卡的 I/O 地址。

下面的方式是使用 “wSlotBus” 和 “wSlotDevice” 来简单的识别 **card_0**:

步骤 1: 移除 PC 中所有 PIO-D48 系列卡。

步骤 2: 安装一块 PIO-D48 系列卡到 PC 的第一个 PCI 插槽，运行 PIO_PISO.EXE。

然后记录 wSlotBus1 和 wSlotDevice1 的信息。

步骤 3: 移除 PC 中所有 PIO-D48 系列卡。

步骤 4: 安装一块 PIO-D48 系列卡在 PC 的第二个插槽并运行 PIO_PISO.EXE。

然后记录 wSlotBus2 和 wSlotDevice2 的信息。

步骤 5: 重复步骤 3 和 4 到每个 PCI 插槽并记录所有 wSlotBus 和 wSlotDevice 信息。

记录的信息也许与下表相似:

表 6-3

PC's PCI Slot	Locating/Resource	
	wSlotBus (Bus#)	wSlotBus (Device#)
Slot_1	0	0x07
Slot_2	0	0x08
Slot_3	0	0x09
Slot_4	0	0x0A
PCI-BRIDGE		
Slot_5	1	0x0A
Slot_6	1	0x08
Slot_7	1	0x09
Slot_8	1	0x07

上面所记录的是在一台 PC 机上的 wSlotBus 和 wSlotDevice 信息。这些值将被映射到 PC 的物理插槽。任何 PIO-D48 系列卡的映像将不能被改变。因此，下面三个步骤就能够使用这个信息确定 PIO-D486 系列卡：

步骤 1: 利用表 6-1 及 6-2 wSlotBus 和 wSlotDevice 信息。

步骤 2: 输入板卡号到函数 PIO_GetConfigAddressSpace(...) 去获得板卡的详细信息，尤其是 wSlotBus 和 wSlotDevice 信息。

步骤 3: 用户可以识别一个指定的 PIO-D48 系列板卡，通过步骤 1 和步骤 2 得到的 wSlotBus 和 wSlotDevice 数据来比较。

注意：通常 PIO-D48 系列卡安装在插槽 0 就是 card0，安装在插槽 1 就是 card1。

6.3 I/O 地址映像

通过主板 ROM BIOS 去自动分配 PIO/PISO 板卡的 I/O 地址。用户同样可以再分配 I/O 地址。强烈推荐用户不要去改变 I/O 地址，即插即用 BIOS 将会很好的去分配每个 PIO/PISO 板卡最适合的 I/O 地址。PIO-D48 系列卡的 I/O 地址见下表。

表 6-4:

Address	Read/读	Write/写
wBase+0	-	RESET\ 控制寄存器
wBase+5	INT 屏蔽控制寄存器	相同
wBase+7	Aux 引脚状态寄存器	相同
wBase+0x2a	INT 极性控制寄存器	相同
wBase+0xc0	Read Port0 (PA)	Write Port0 (PA)
wBase+0xc4	Read Port1 (PB)	Write Port1 (PB)
wBase+0xc8	Read Port2 (PC)	Write Port2 (PC)
wBase+0xcc	-	Write Control Word (Port0/1/2)
wBase+0xd0	Read Port3 (PA)	Write Port3 (PA)
wBase+0xd4	Read Port4 (PB)	Write Port4 (PB)
wBase+0xd8	Read Port5 (PC)	Write Port5 (PC)
wBase+0xdc	-	Write Control Word (Port3/4/5)
wBase+0xe0	Read 8254-Counter0	Write 8254-Counter0
wBase+0xe4	Read 8254-Counter1	Write 8254-Counter1
wbase+0xe8	Read 8254-Counter2	Write 8254-Counter2
wBase+0xec	Read 8254 Control Word	Writer 8254 Control Word
wBase+0xf0	Read Clock/Int Control Word	Write Clock/Int Control Word
wBase+0xf4	Read Card ID	-

注意：有关 wBase 详情请参考 [章节 6.1 “如何找到 I/O 地址”](#)。

6.3.1 RESET\控制寄存器

(Read/Write): wBase+0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RESET\

当 PC 的电源第一次开启，RESET\ 信号是低电平状态。这将禁用所有 D/I/O 操行。用户在使用任何 D/I/O 命令前，必须将 RESET\ 信号置于高电平状态。

示例:

```

outputb (wBase,1);          /* RESET\=High →所有 D/I/O 启动*/
outputb (wBase,0);          /* RESET\=Low →所有 D/I/O 禁用*/
    
```

6.3.2 AUX 状态寄存器

(Read/Write): wBase+7

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Aux7	Aux6	Aux5	Aux4	Aux3	Aux2	Aux1	Aux0

Aux0=INT_CHAN0、Aux1=INT_CHAN_1、Aux2=INT_CHAN_2、Aux3=INT_CHAN_3、Aux7~4=Aux-ID。

Aux 0 ~ 3 作为中断源，为分辨中断源中断服务程序必须读取该这个寄存器。

更多信息参考 [章节 2.7 “中断运行”](#)。

6.3.3 INT 屏蔽控制寄存器

(Read/Write): wBase+5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	EN3	EN2	EN1	EN0

EN0=0→ 禁用 INT_CHAN_0 作为中断信号 (默认)

EN0=1→ 激活 INT_CHAN_0 作为中断信号

EN1=0→ 禁用 INT_CHAN_1 作为中断信号 (默认)

EN1=1→ 激活 INT_CHAN_1 作为中断信号

EN2=0→ 禁用 INT_CHAN_2 作为中断信号 (默认)

EN2=1→ 激活 INT_CHAN_2 作为中断信号

EN3=0→ 禁用 INT_CHAN_3 作为中断信号 (默认)

EN3=1→ 激活 INT_CHAN_3 作为中断信号

示例:

```
outportb(wBase+5,0);    /*禁用所有中断*/
outportb(wBase+5,1);    /*激活中断 INT_CHAN_0 */
outportb(wBase+5,2);    /*激活中断 INT_CHAN_1 */
outportb(wBase+5,4);    /*激活中断 INT_CHAN_2 */
outportb(wBase+5,8);    /*激活中断 INT_CHAN_3 */
outportb(wBase+5,0x0f); /*激活所有四通道中断*/
```

6.3.4 中断极性控制寄存器

(Read/Write): wBase+0x2A

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	INV3	INV2	INV1	INV0

该寄存器提供函数控制中断信号源反相或同相作用。具体应用案例如下：

INV0=0→ 选择 interrupt_channel_0 反相信号

INV0=1→ 选择 interrupt_channel_0 相同信号

INV1= 控制中断 channel_1

INV2= 控制中断 channel_2

INV3= 控制中断 channel_3

示例:

```
/*选择所有 4 通道的中断源信号正相输入*/
```

```
outportb(wBase+0x2a,0x0f);
```

```
/*选择所有 4 通道中断源信号反相输入*/
```

```
outportb(wBase+0x2a,0x00);
```

```
/*选择正相输入 INT_CHAN_0 */
```

```
/*选择其它通道反相输入*/
```

```
outportb(wBase+0x2a,0x0e);
```

```
/*选择正相输入 INT_CHAN_0, INT_CHAN_1*/
```

```
/*选择其它通道反相输入*/
```

```
outportb(wBase+0x2a,0x0c);
```

更多信息参考 [章节 2.7 “中断运行”](#) 与 DEMO5.C (DOS)。

6.3.5 读/写 I/O 口

➤ 控制命令

(Read/Write): wBase+0xcc/ 0xdc

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	PA	PC-H	0	PB	PC-L

PIO-D48 系列卡拥有 6 路 8 位 I/O 端口，每路 I/O 口均可由程序定义为 D/I 或 D/O 口。当 PC 启动时，所有 12 路端口默认为 D/I 端口。

(读/写)→ wBase+0xcc = 设定 Port-0, Port-1, Port-2 为 D/I 或 D/O

(读/写)→ wBase+0xdc = 设定 Port-3, Port-4, Port-5 为 D/I 或 D/O

PA/ PB/ PC-H/ PC-L : 1→ 输入端口, 0→ 输出端口.

PC-H: PC 高字节

PC-L: PC 低字节

➤ 读/写 8 位数据

(Read/Write): wBase+0xc0/ 0xc4/ 0xc8/ 0xd0/ 0xd4/ 0xd8

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

(读/写)

wBase+0xc0= Port-0 (PA)

wBase+0xc4= Port-1 (PB)

wBase+0xc8= Port-2 (PC)

wBase+0xd0= Port-3 (PA)

wBase+0xd4= Port-4 (PB)

wBase+0xd8= Port-5 (PC)

示例:

```
outportb(wBase+0xcc,0x80);    /* port-0, port-1, port-2 为 D/O port */
outportb(wBase+0xc0,V1);      /* 写入 port_0 (PA)*/
outportb(wBase+0xc4,V2);      /* 写入 port_1 (PB)*/
outportb(wBase+0xc8,V3);      /* 写入 port_2 (PC)*/

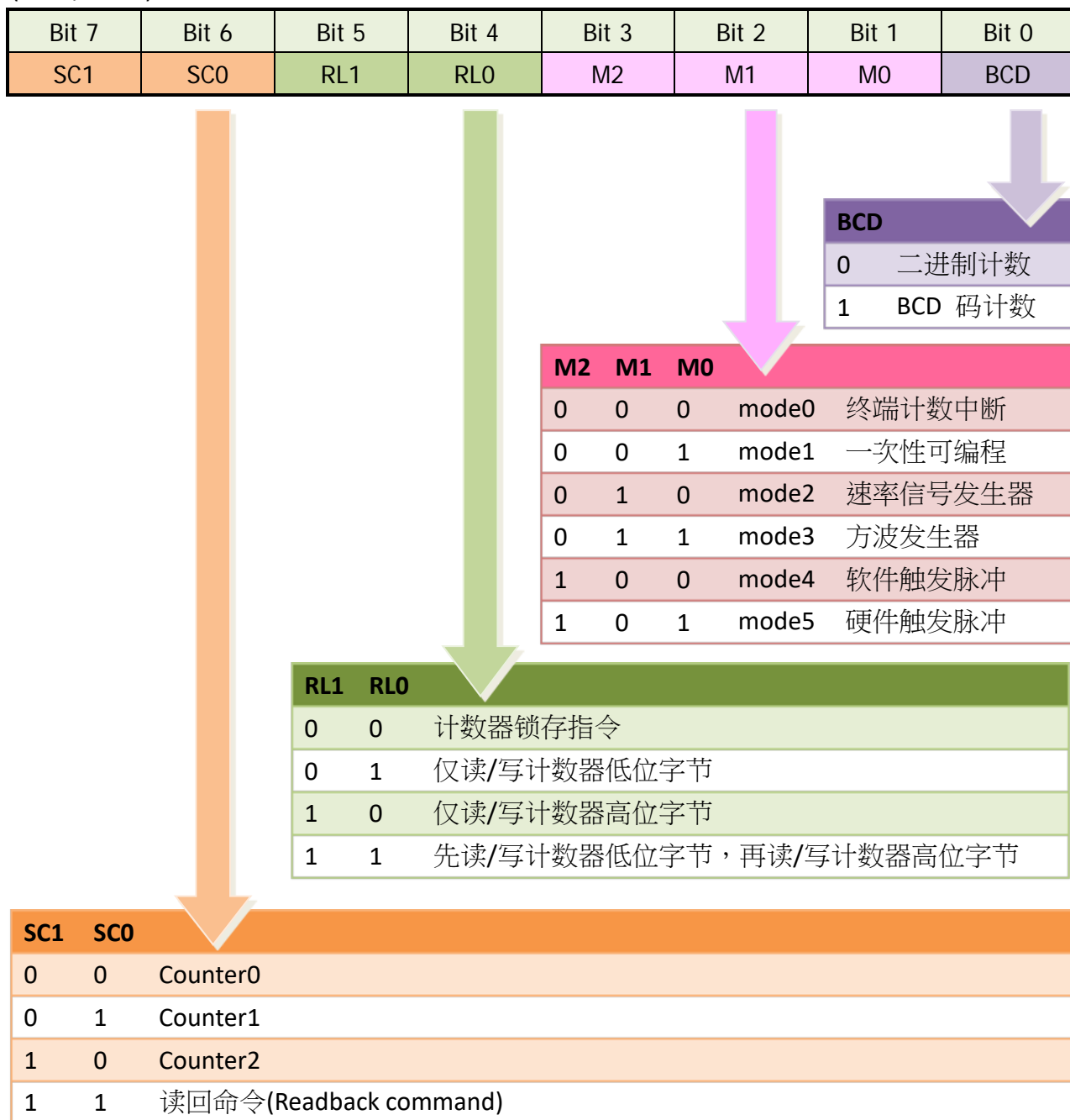
outportb(wBase+0xdc,0x9B);    /* port-3, port-4, port-5 为 D/I port */
V1=inportb(wBase+0xd0);       /* 读取 port_3 (PA)*/
V2=inportb(wBase+0xd4);       /* 读取 port_4 (PB)*/
V3=inportb(wBase+0xd8);       /* 读取 port_5 (PC)*/
```

注意: I/O 速度取决于 I/O Card、Bus 速度、CPU 速度以及系统的负载。所以以上任何一个因素改变都有可能造成 I/O 速度不同。

6.3.6 读/写 8254

➤ 8254 控制命令

(Read/Write): wBase+0xec



➤ 读/写 8254 的 8 位数据

(Read/Write): wBase+0xe0/ 0xe4/ 0xe8

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

(读/写): wBase+0xec= 8254 控制命令

(读/写): wBase+0xe0= 8254-counter-0

(读/写): wBase+0xe4= 8254-counter-1

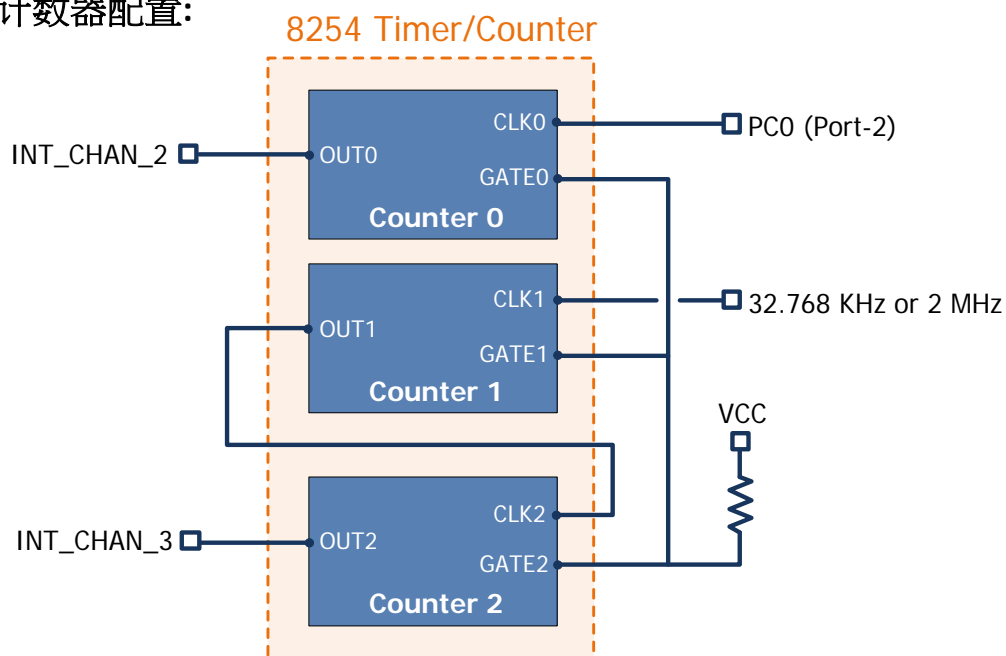
(读/写): wBase+0xe8= 8254-counter-2

示例:

```

outputb(wBase+0xec,0x30);    /* Counter0, mode-0 */
outputb(wBase+0xe0,0xff);    /* 写入第一个 low byte */
outputb(wBase+0xe4,0xff)     /* 写入第二个 high byte */
/* 然后 Counter0 将从 0xffff 开始向下计数 */
    
```

➤ 8254 计数器配置:



更多相关信息请参考如下示例程序:

Int2 Demo Counter0 (使用中斷 INT_CHAN_2)

Int3 Demo Counter1 ~ Counter 2 (使用中斷 Int_CHAN_3)

6.3.7 读/写 Clock/Int 控制寄存器

(Read/Write): wBase+0xf0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	CTRL-D5	CTRL-D4	CTRL-D3	CTRL-D2	CTRL-D1	CTRL-D0

➤ **CTRL-D0:** 选择 CLK1 为时钟资源 (参考 [章节 6.3.6 “读/写 8254”](#))

0→ 2 MHz

1→ 32.768 KHz

➤ **CTRL-D1:** port-2 中 PC0 的反相或同相设定 (参考 [章节 2.7.4 “INT CHAN 2”](#))

0→ non-invert

1→ invert

➤ **CTRL-D3, CTRL-D2:** 中断资源选择 (参考 [章节 2.7.2 “INT CHAN 0”](#))

01 : 禁用 PC3 & !PC7 (of port-2) 为中断资源

10 : INT_CHAN_0=PC3 of port-2

00 : INT_CHAN_0=PC3 & !PC7 of port-2

➤ **CTRL-D5, CTRL-D4:** 中断资源选择(参考 [章节 2.7.3 “INT CHAN 1”](#))

01 : 禁用 PC3 & !PC7 (of port-5) 为中断资源

10 : INT_CHAN_1=PC3 of port-5

00 : INT_CHAN_1=PC3&!PC7 of port-5

6.3.8 读 Card ID 寄存器

(Read): wBase+0xf4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	ID3	ID2	ID1	ID0

示例:

```
wCardID=inportb(wBase+0xf4);    /* 读取 Card ID */
```

注意: 仅有 PIO-D48U/D48SU 及 PEX-D48 (1.0 版或更新版本) 支持 Card ID 功能。

7. 示例程序

7.1 Windows Demo 程序

如果 DLL 驱动没有正确安装那么所有 DEMO 程序将不能正常工作。在 DLL 驱动安装过程的时候，安装程序将注册适当的内核驱动到操作系统中，并且拷贝 DLL 驱动和 DEMO 程序到适当的位置，你可以选择 (Win98/Me/NT/2000 and 32-/64-bit Win XP/2003/2008/7/8/10) 驱动软件包。一次完整的驱动安装，下列出现相关的 DEMO 程序、库文件、声明的头档在不同的运行环境：

➤ PIO-DIO Series Classic 驱动程序的 Demo 程序:

取得示例程序位置:

CD:\NAPDOS\PCI\PIO-DIO\DLL_OCX\Demo\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/demo/

<ul style="list-style-type: none">BCB4 → for Borland C++ Builder 4PIODIO.H → Header filesPIODIO.LIB → Linkage library for BCB only	<ul style="list-style-type: none">Delphi4 → for Delphi 4PIODIO.PAS → Declaration files
<ul style="list-style-type: none">VC6 → for Visual C++ 6PIODIO.H → Header filesPIODIO.LIB → Linkage library for VC only	<ul style="list-style-type: none">VB6 → for Visual Basic 6PIODIO.BAS → Declaration files
<ul style="list-style-type: none">VB.NET2005 → for VB.NET2005PIODIO.vb → Visual Basic Source files	<ul style="list-style-type: none">CSharp2005 → for C#.NET2005PIODIO.cs → Visual C# Source files

详细 PIO-DIO 系列的 DLL 函式，请参考至 PIO-DIO DLL 软件使用手册
(CD:\NAPDOS\PCI\PIO-DIO\Manual)

➤ UniDAQ SDK 驱动程序的 Demo 程序:

取得示例程序位置:

CD:\NAPDOS\PCI\UniDAQ\DLL\Demo\

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidag/dll/demo/>

<ul style="list-style-type: none">⊕ BCB6 → for Borland C++ Builder 6UniDAQ.H → Header filesUniDAQ.LIB → Linkage library for BCB only	<ul style="list-style-type: none">⊕ Delphi6 → for Delphi 6UniDAQ.PAS → Declaration files
<ul style="list-style-type: none">⊕ VB6 → for Visual Basic 6UniDAQ.BAS → Declaration files	<ul style="list-style-type: none">⊕ CSharp2005 → for C#.NET2005UniDAQ.cs → Visual C# Source files
<ul style="list-style-type: none">⊕ VC6 → for Visual C++ 6UniDAQ.H → Header filesUniDAQ.LIB → Linkage library for VC only	<ul style="list-style-type: none">⊕ VB.NET2005 → for VB.NET2005UniDAQ.vb → Visual Basic Source files
<ul style="list-style-type: none">⊕ VC.NET2005 → for VC.NET2005 (32-bit)UniDAQ.H → Header filesUniDAQ.LIB → Linkage library for VC only	<ul style="list-style-type: none">⊕ VC.NET2005 → for VC.NET2005 (64-bit)UniDAQ.H → Header filesUniDAQ.LIB → Linkage library for VC only

详细 UniDAQ DLL 函式，请参考至 UniDAQ 软件使用手册
(CD:\NAPDOS\PCI\UniDAQ\Manual\)

7.2 DOS Demo 程序

取得示例程序位置:

CD:\NAPDOS\PCI\PIO-DIO\DOS\D48\PIOD48\

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dos/d48/piod48/>

- ✦ \TC*. * → for Turbo C 2.xx or above
- ✦ \MSC*. * → for MSC 5.xx or above
- ✦ \BC*. * → for BC 3.xx or above

- ✦ \TC\LIB*. * → for TC Library
- ✦ \TC\DEMO*. * → for TC demo program
- ✦ \TC\DIAG*. * → for TC diagnostic program
- ✦ \TC\LIB\PIO.H → TC Declaration File
- ✦ \TC\LIB\TCPIO_L.LIB → TC Large Model Library File
- ✦ \TC\LIB\TCPIO_H.LIB → TC Huge Model Library File

Demo 程序列表:

DEMO1: D/O Demo

DEMO2: D/I Demo

DEMO3: D/I/O Demo

DEMO4: Timer Interrupt of INT_CHAN_3

DEMO5: Event Counter (no interrupt) of INT_CHAN_2 (init_HIGH & active_Low)

DEMO6: Event Counter (no interrupt) of INT_CHAN_2 (init_Low & active_HIGH)

DEMO7: Down-Counter (interrupt) of INT_CHAN_2 (init_HIGH & active_Low)

DEMO8: Interrupt demo of INT_CHAN_0 (PC7 of Port2 don't care)

DEMO9: Interrupt demo of INT_CHAN_0 (PC7 of Port2 interrupt is enable)

DEMO10: Interrupt demo of INT_CHAN_1 (PC7 of Port5 interrupt is disable)

DEMO11: Interrupt demo of INT_CHAN_0 & INT_CHAN_1

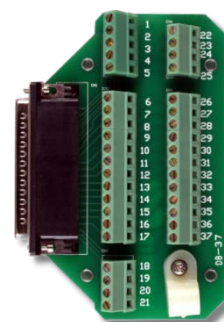
详细 DOS 的 DLL 函式, 请参考至 PIO-DIO DLL 软件使用手册
(CD:\NAPDOS\PCI\PIO-DIO\Manual\)

附录：端子板

A1. DB-37, DN-37, DN-50 及 DN-100

➤ **DB-37:**

DB-37 是一个 37 针的端子板可以很方便的进行接线。使用 37-pin cable (如: CA-3710) 将 DB-37 连接至 PIO-D48 的 CON1。



DB-37

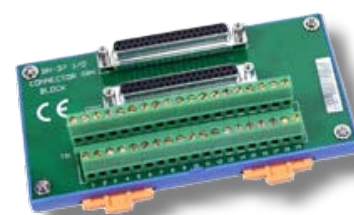
➤ **DN-37 及 DN-50:**

DN-37 是一个带 DIN 安装导轨和 37 芯 D 型插头的 I/O 接线板。
DN-50 是一个带 DIN 安装导轨和 50 针的公头扁平电缆。可以很方便的进行接线。

使用 37-pin cable (如: CA-3710) 将 DN-37 连接至 PIO-D48 的 CON1, 再使用 50-pin cable (如: CA-5002) 将 DN-50 连接至板卡的 CN1/CN2/CN3。



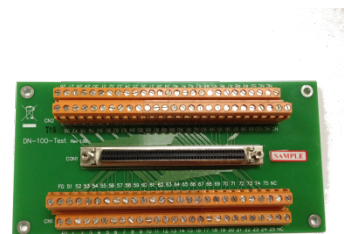
DN-50



DN-37

➤ **DN-100:**

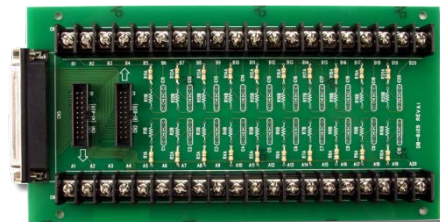
DN-100 是一个 100 针的 SCSI II 接头端子板可以很方便的进行接线。使用 100-pin SCSI II cable (如: CA-SCSI100-15) 将 DN-100 连接至 PIO-D48SU 的 CON1。



DN-100

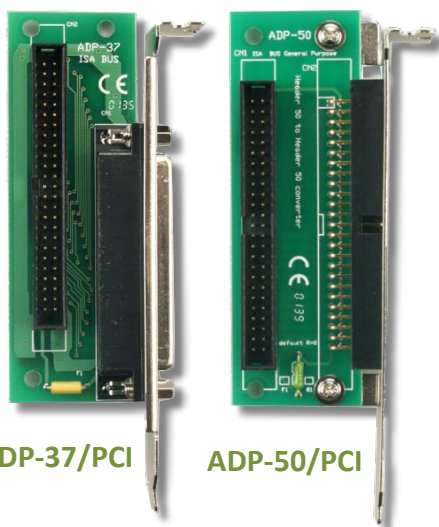
A2. DB-8125

DB-8125 是一个带 1 米 D 型接头 37 芯电缆的螺钉端子板。DB8125 由一个 DB-37 和两个 20 芯扁平电缆组成。使用 37-pin cable (如: CA-3710) 将 DB-8125 连接至 PIO-D48 的 CON1。



DB-8125

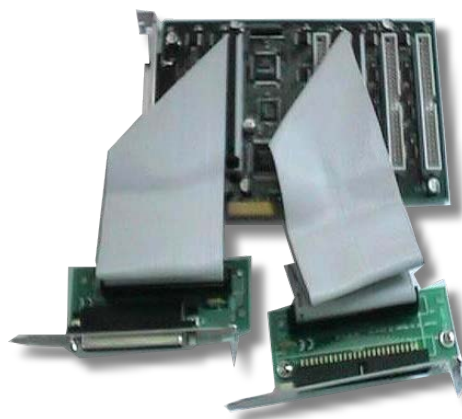
A3. ADP-37/PCI 及 ADP-50/PCI



ADP-37/PCI

ADP-50/PCI

ADP-37/PCI 和 ADP-50/PCI 分为 50 针的 PCI 插槽的端子档板。它们一边可接连 50 针的接线头，另一端可连接 PC 上相应的插槽上。参考如下：



ADP-37/PCI：用于 PCI 总线卡的 50 针电缆到 37 针 D 型连接器的插槽挡板。

ADP-50/PCI：用于 PCI 总线卡的 50 针电缆连接到机箱的插槽挡板。

A4. DB-24P/DB-24PD 光电隔离输入端子板

DB-24 是 24 通道隔离数字量输入端子板。

DB-24P 的光隔离输入由光电耦合器带一个电流检测用电阻组成。你可以用它来检测从 TTL 电平到 24 V 直流信号。也可以用来检测宽范围的交流信号。还可以用此卡来隔离计算机和工业环境中常发生的共模电压，地环流以及电压尖峰。见图 A4-1。表 A4-1 是用来比较 DB-24P 和 DB-24PD。

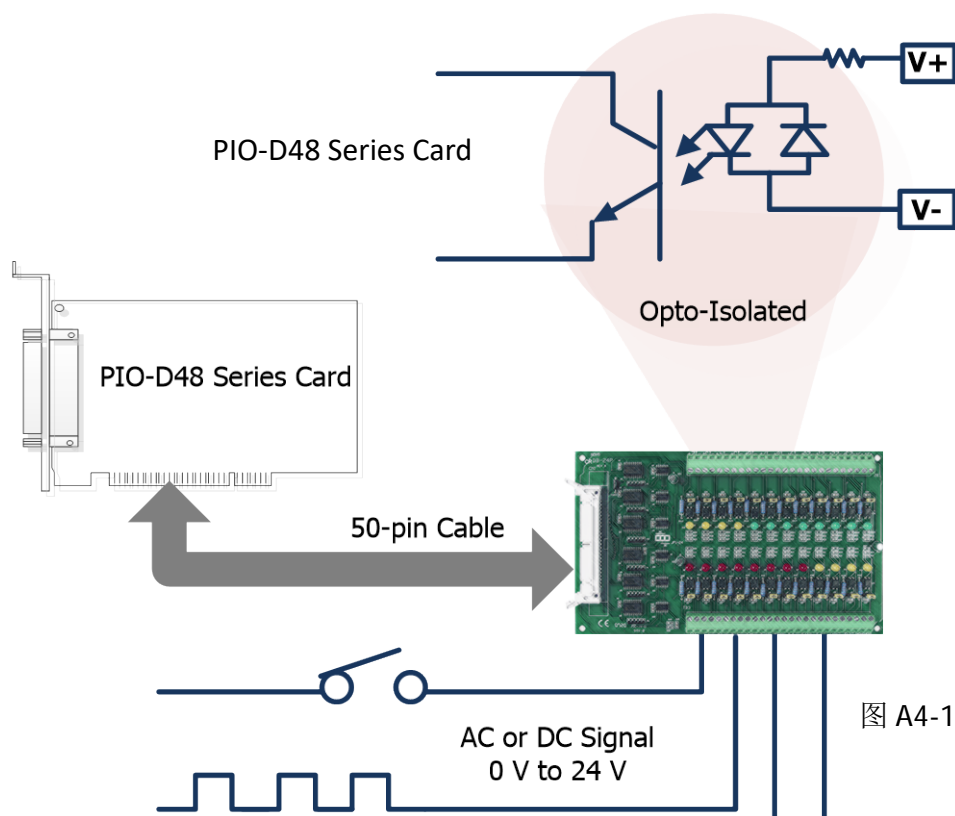
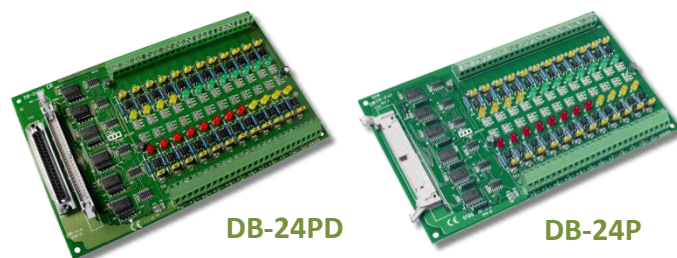


图 A4-1

表 A4-1:

	DB-24P	DB-24PD
50-pin Flat-Cable Header	Yes	Yes
D-sub 37-pin Header	No	Yes
Other Specifications	Same	

A5. DB-24R/DB-24RD 继电器输入端子板

DB-24R 有 24 个 C 型继电器，通过可编程使得机电式继电器可直接通断负载。每路继电器可控制一个 0.5 A/110 V 或 1A/24 V_{DC} 的负载。通过 50 针与 OPTO-22 兼容的连接器或 20 针扁平电缆线连接器，给对应通道的功率继电器加上 5 V 电压信号来激活其工作。每个通道一个 LED，共有 24 个高亮度 LED，当与之关联的继电器接通时发亮。为避免你的 PC 过载，本板卡需要一个+12 V_{DC} 或+24 V_{DC} 的外界电源供电。见图 A5-1。

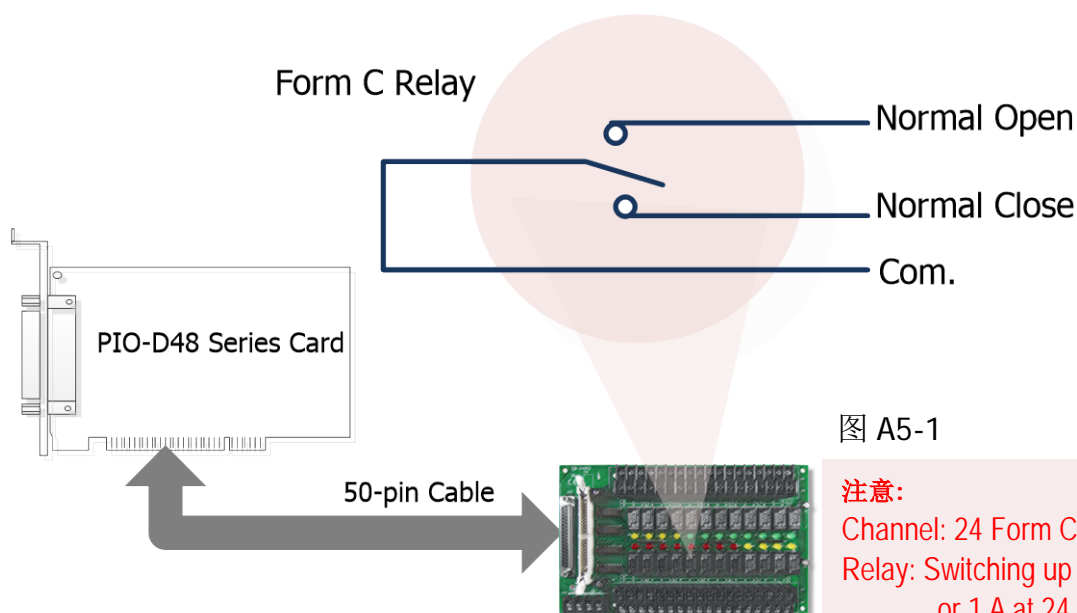
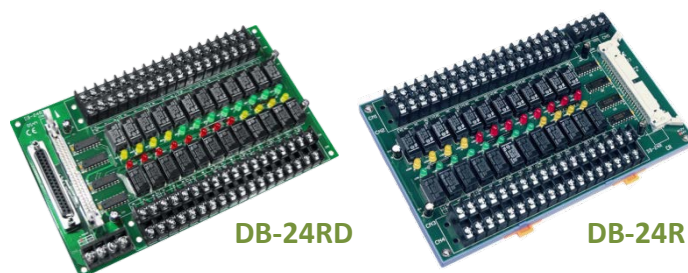


图 A5-1

注意:
Channel: 24 Form C Relay
Relay: Switching up to 0.5 A at 110 V_{AC}
or 1 A at 24 V_{DC}

表 A5-1:

	DB-24R	DB24RD
50-pin Flat-Cable Header	Yes	Yes
D-sub 37-pin Header	No	Yes
Other Specifications	Same	

表 A5-2:

DB-24R, DB-24RD	24 * Relay (120 V, 0.5 A)
DB-24PR, DB-24PRD	24 * Power Relay (250 V, 5 A)
DB-24POR	24 * PhotoMOS Relay (350 V, 0.1 A)
DB-24SSR	24 * SSR (250 V _{AC} , 4 A)

A6. 端子板对照表

端子板对照表 A6-1 为 PIO-D48 系列板卡，参考如下：

表 A6-1:

I/O Card	-	PIO-D48 PIO-D48U PEX-D48	PIO-D48 PIO-D48U PEX-D48	PIO-D48SU
Cable/ Daughter Boards	20-Pin Flat-Cable	50-Pin Flat-Cable	37-Pin D-sub	100-Pin SCSI II
DB-37	NO	NO	Yes	NO
DN-37	NO	NO	Yes	NO
ADP-37/PCI	NO	Yes	Yes	NO
ADP-50/PCI	NO	Yes	NO	NO
DB-24P	NO	Yes	NO	NO
DB-24PD	NO	Yes	Yes	NO
DB-16P8R	NO	Yes	Yes	NO
DB-24R	NO	Yes	NO	NO
DB-24RD	NO	Yes	Yes	NO
DB-24C	Yes	Yes	Yes	NO
DB-24PR	Yes	Yes	NO	NO
DB-24PRD	NO	Yes	Yes	NO
DB-24POR	Yes	Yes	Yes	NO
DB-24SSR	NO	Yes	Yes	NO
DN-100	NO	NO	NO	Yes