

分類/Classification	<input type="checkbox"/> tDS	<input type="checkbox"/> tGW	<input type="checkbox"/> PETL/tET/tPET	<input type="checkbox"/> DS/PDS/PPDS	<input type="checkbox"/> tM-752N
	<input checked="" type="checkbox"/> I/O Card	<input type="checkbox"/> VXC Card	<input type="checkbox"/> VxComm	<input type="checkbox"/> Other	
作者/Author	Albert	日期/Date	2015-04-02	編號/No.	FAQ-022

Q：如何處理 DI/DO 多通道的資料傳輸？

A：一般的 Digital I/O 卡每個埠有 8 個通道，在進行資料傳輸時，不論輸入/輸出皆是使用 16 進制的方式來處理，以下將介紹如何將通道數對應到 16 進制上。

當我們設置好了一個 8-Bit 的 DO 埠時，他的資料格式可以表示成以下式子：

D/O byte = 0000 0000(bit) = 0x00 (hex) (附錄 1)，要注意的是，資料從左到右分別為 Bit 7 到 Bit 0，Bit 7~Bit 4 為一組，Bit 3~Bit 0 為一組，如以下表格所示，有關每個埠號的通道位置，請參考每塊板卡的通道設定。

	Bit 7 ~ Bit 4	Bit 3 ~ Bit 0
D/O byte	0 0 0 0	0 0 0 0

● 操作範例說明

➤ D/O 輸出範例

1. 設定"Bit 2"輸出：D/O = 0000 0100 (bit) = 0x04 (hex)
2. 接著設定" Bit 5 和 6"輸出：D/O = 0110 0100 (bit) = 0x64 (hex)
3. 接著設定" Bit 7"輸出，但是"Bit 2"不輸出：D/O = 1110 0000 (bit) = 0xE0 (hex)

➤ D/I 輸入範例

1. 假設 DI 埠收到一組資料 0x3D
=> DI = 0x3D (hex) = 0011 1101 (bit)
=> 表示 Bit 0、2、3、4、5 有值，也就是通道 0、2、3、4、5 有資料輸入
2. 假設 DI 埠收到一組資料 0x80
=> DI = 0x80 (hex) = 1000 0000 (bit)
=> 表示只有 Bit 7 有值，也就是只有通道 7 有資料輸入

優點：多通道的資料傳輸在讀寫一次資料時，至少可操作 8 個通道的數值，可有效的提高處理效率。

缺點：如果只想要單獨處理某通道的數值，則需要再進行數值的換算。可參考以下範例。

- Binary Operation (for C/C++)

Mask Computing

Mask = 0000 0001 left-shift 3 bits => 0000 1000

Mask = Invert Mask => 1111 0111

Mask Off for Bit N

Mask = $\sim (1 \ll N)$

Result = Data & Mask

Mask Off for Bit 3

Data	1010	1010
AND	1111	0111
Result	1010	0010

Mask Computing

Mask = 0000 0001 left-shift 3 bits => 0000 1000

Set On for Bit N

Mask = $1 \ll N$

Result = Data | Mask

Set On for Bit 3

Data	1010	0010
OR	0000	1000
Result	1010	1010

Get Bit 3 Status (Data = 1010 1010)

Data right-shift 3 bits = 0001 0101

Data mask off with 0000 0001

0001 0101 & 0000 0001 = 0000 0001

Get Bit N Status

Result = (Data >> N) & 1

Get Bit 3 Status

Data	1010	0010
Shift	0001	0101
AND	0000	0001
Result	0000	0001

- Binary Operation (for VB)

Mask Computing

Mask = 0000 0001 left-shift 3 bits => 0000 1000

Mask = Invert Mask => 1111 0111

Mask Off for Bit N

Mask = Not (2 ^ N)

Result = Data and Mask

Mask Off for Bit 3		
Data	1010	1010
AND	1111	0111
Result	1010	0010

Mask Computing

Mask = 0000 0001 left-shift 3 bits => 0000 1000

Set On for Bit N

Mask = 2 ^ N

Result = Data or Mask

Set On for Bit 3		
Data	1010	0010
OR	0000	1000
Result	1010	1010

Get Bit 3 Status (Data = 1010 1010)

Data right-shift 3 bits = 0001 0101

Data mask off with 0000 0001

0001 0101 & 0000 0001 = 0000 0001

Get Bit N Status

Result = (Data \ (2 ^ N)) and 1

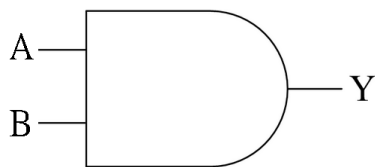
Get Bit 3 Status		
Data	1010	0010
Shift	0001	0101
AND	0000	0001
Result	0000	0001

附錄 1：二進位轉 16 進位

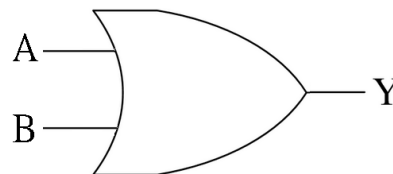
Binary	Hexadecimal	Binary	Hexadecimal
0 0 0 0	0x0	1 0 0 0	0x8
0 0 0 1	0x1	1 0 0 1	0x9
0 0 1 0	0x2	1 0 1 0	0xA (10)
0 0 1 1	0x3	1 0 1 1	0xB (11)
0 1 0 0	0x4	1 1 0 0	0xC (12)
0 1 0 1	0x5	1 1 0 1	0xD (13)
0 1 1 0	0x6	1 1 1 0	0xE (14)
0 1 1 1	0x7	1 1 1 1	0xF (15)

附錄 2：Binary Operation (Bitwise)

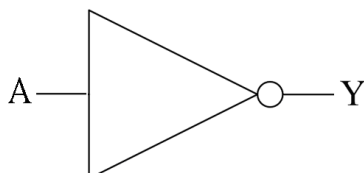
AND	0	1
0	0	0
1	0	1



OR	0	1
0	0	1
1	1	1



NOT	
0	1
1	0



XOR	0	1
0	0	1
1	1	0

