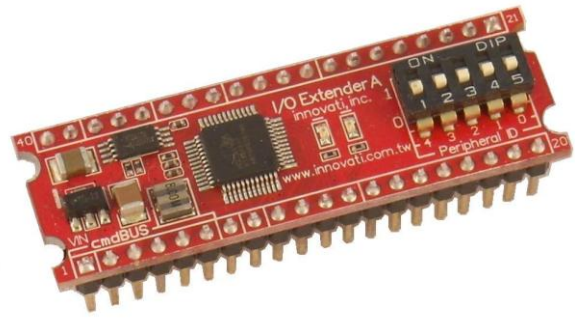


I/O Extender A

輸出入擴充模組

版本: V2.0



產品介紹: 透過利基 I/O Extender A 模

組，使用者可以取得除了 Ozone 外，增加 3 個埠，共 24 個腳位，每個腳位都能以類似的指令加上模組名稱，做出與 Ozone 上各腳位相近的效能。除此之外，更增加輸入 Pulse 的計數等多樣功能。

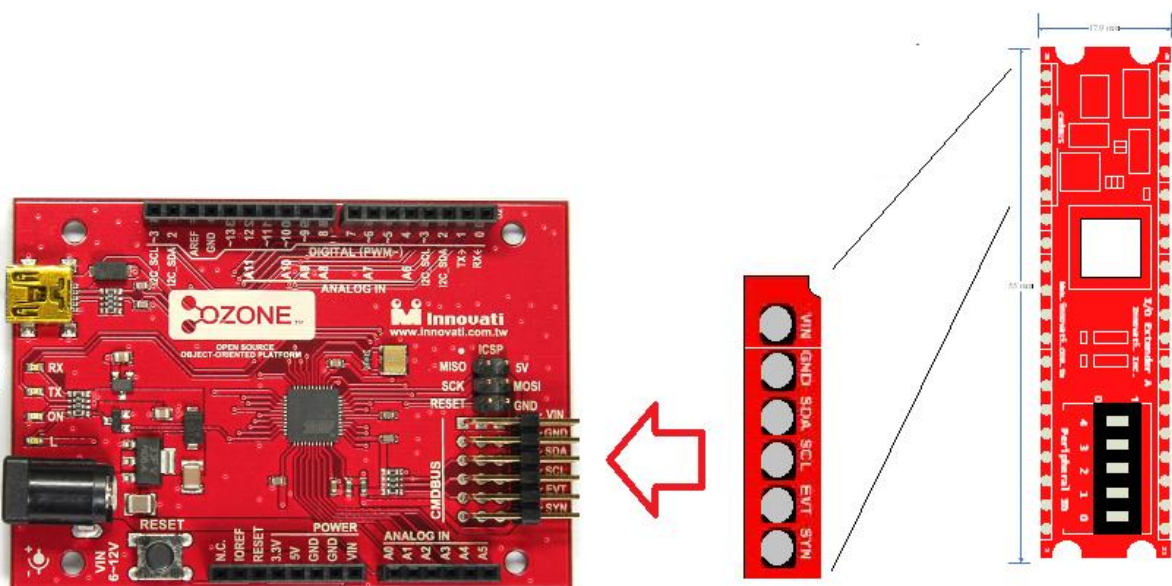
應用方向:

- 可以同時增加獨立控制多個燈號的顯示變化。
- 應用 AD 轉換腳位，可以將類比訊號轉為數位處理。

產品特色:

- 三個埠各含八個腳位，共二十四根腳位可提供增加應用。
- 獨立腳位訊號寬度量測與訊號個數計數，精準度可至微秒(μs)。
- 八通道類比轉數位(AD)量測輸入。
- 可透過 I2C 方式，下達指令。

連接方式: 直接將 ID 開關撥至欲設定的編號，再將 cmdBUS 連接至 Ozone 上對應的腳位，就可透過 Ozone 執行操作。



產品規格:

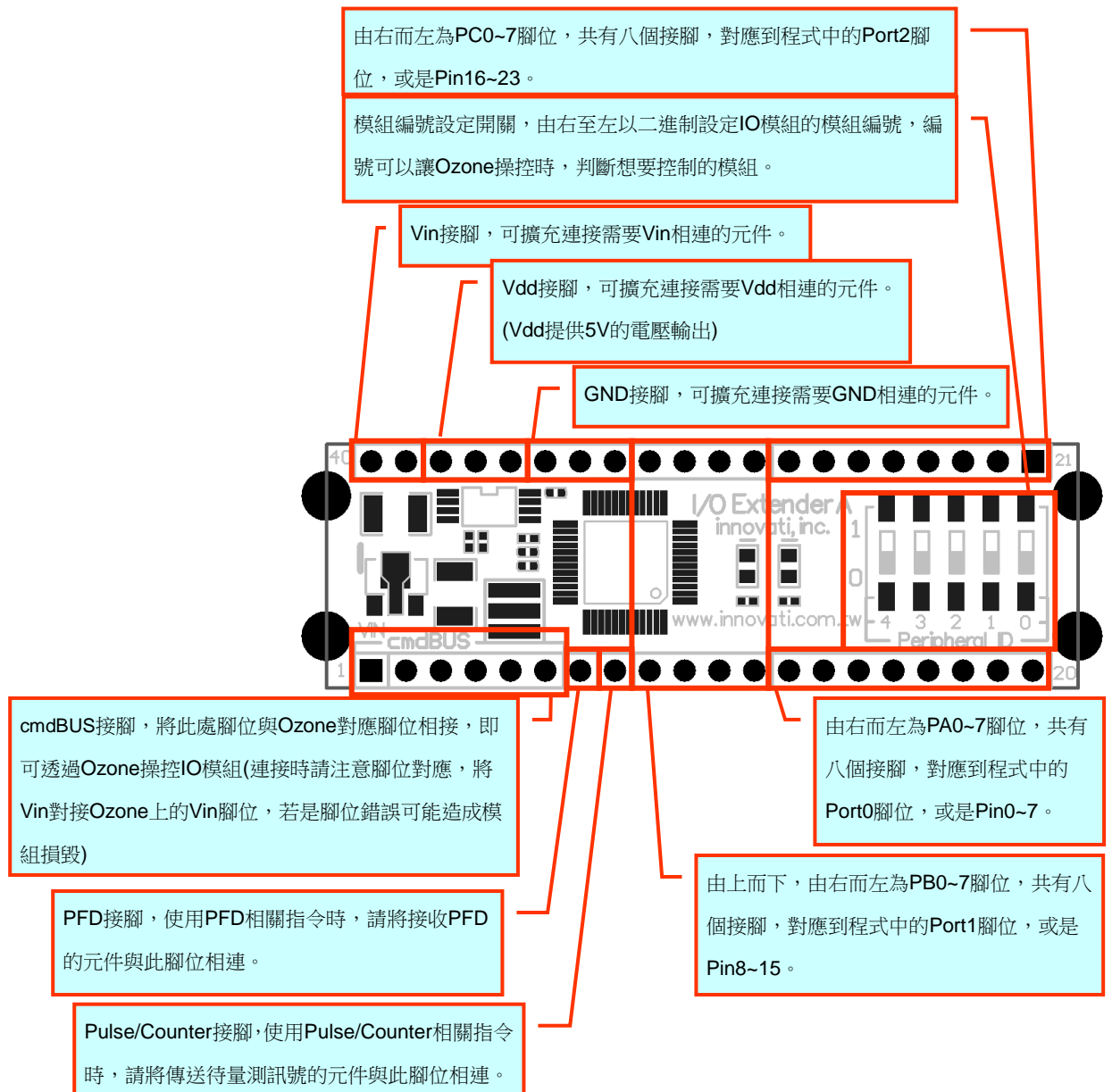


圖 1: 模組腳位與開關介紹

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{IN} = 7.5	Conditions				
I _{DD}	Operating Current	—	No I/O	—	4	—	mA
V _{IL}	Input Low Voltage for I/O Ports.	—	—	0	—	1.5	V
V _{IH}	Input High Voltage for I/O Ports.	—	—	3.5	—	5	V
V _{OH}	I/O Port output high voltage	—	No loading	—	5	—	V

V _{OL}	I/O Port output low voltage	—	No loading	—	0	—	V
I _{OL}	I/O Port Sink Current	—	V _{load} =0.1V _{OH}	10	20	—	mA
I _{OH}	I/O Port Source Current	—	V _{load} =0.9V _{OH}	-5	-10	—	mA
V _{AD}	A/D Input Voltage	—	—	0	—	5	V
I _{ADC}	Additional Power Consumption if A/D Converter is Used	—	—	—	1.5	3	mA

表 1: 電壓電流特性 (於 25 °C 之環境)

操作注意事項:

操作溫度 0 °C~70°C
 儲存溫度 -50 °C~125°C

指令格式	指令功能
各埠腳位讀取寫入相關指令	
High(uint8_t BitNum)	設定 BitNum 所指定腳位輸出高電位 (BitNum 範圍為 0~23, 若設定超出此範圍則不動作) *1
In(uint8_t BitNum, uint8_t &Value)	取得 BitNum 所指定腳位的電位, 為 0 或 1, 存放於 Value (BitNum 範圍為 0~23, 若設定超出此範圍則不動作) *2
Low (uint8_t BitNum)	設定 BitNum 所指定腳位輸出低電位 (BitNum 範圍為 0~23, 若設定超出此範圍則不動作) *1
PulseOut(uint8_t BitNum, uint8_t Mode, uint16_t Width)	從 BitNum 所指定的腳位, 輸出以毫秒(ms) 為單位, 長度為 Width 的 pulse, Width 可以輸入 0~65535 的整數值, Mode 為 0 則輸出低電位, 若 Mode 為 1, 則輸出高電位 (BitNum 範圍為 0~23, Mode 則只可輸入 0 或 1, 當輸入超出此範圍的值, 指令都不會被執行) *3
ReadPort(uint8_t PortNum, uint8_t &Value)	根據 PortNum 選擇要讀取的埠, PortNum 可以輸入 0~2 的整數值, 再以 Value 儲存該埠各腳位之電位, 回傳值為 0~255 的整數值 *4

WritePort(uint8_t PortNum, uint8_t Value)	根據 PortNum 選擇要輸出的埠，請輸入 0~2 之間的整數值，再以 Value 輸出該埠各腳位之電位 Value 可以為範圍在 0~255 之間的整數值 *4
設定腳位模式相關指令 (讀取或設定皆以 0 為輸出模式，1 為輸入模式)	
SetDirPort(uint8_t Port, uint8_t Dir)	由 Dir 設定 Port 所指定的埠各腳位設定的輸出入模式， Port 請輸入 0~2 之間的整數值， Dir 請輸入 0~255 間的整數值
AD 轉換相關指令	
GetADC (uint8_t ChanNum, uint16_t &Value)	取得 ChanNum ，範圍為 0~7 之間的整數值，指定頻道的電位轉換值，存放在 Value 中 (Value 範圍為 0~1023，當輸入電壓高於 VDD 則皆以 1023 表示)
SetADC (uint8_t ChanNum)	根據 ChanNum 值設定 ADC 所使用的頻道如下： 0→ADC 轉換關閉 1→CH0 開啟 2→CH0~CH1 開啟 3→CH0~CH2 開啟 4→CH0~CH3 開啟 5→CH0~CH4 開啟 6→CH0~CH5 開啟 7~8→CH0~CH7 開啟 其中 CH0~CH7 分別是 PA0~PA7
計時器(Timer)相關指令	
GetTimer(uint16_t &Timer)	讀取計時器值存放於 Timer 中 (計時器值在每次啟動計算就會清為零)，回傳值為 0~65535 之間的整數值
SetTimer(uint8_t Mode, uint16_t Timer)	以 Mode 設定計時器的模式，1 為重複模式，0 為單次模式，並以 Timer 設定所要計時的時間值，單位為毫秒(ms)，請輸入 0~65535 之間的整數值
TimerOff(void)	停止計時器計時
TimerOn(void)	啟動計時器計時

*1: 此命令會將腳位設為輸出模式

*2: 此命令會將腳位設為輸入模式

*3: 當原先輸出電位已經是高電位，又再要求輸出高電位的 Pulse，模組就會先傳送一小段的低電位再轉為高電位

*4: 此命令不會改變腳位之輸出入模式，請使用 SetDirPort 相關指令來幫助轉換輸出入模式

範例程式:

```
#include <ozone.h>
```

```
IOExtenderA myIO(2);          // 設定模組編號為 2

uint8_t PinLevel;
uint16_t ChValue;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    myIO.High(0);              // 由第 0 腳位輸出高電位
    myIO.Low(1);               // 由第 1 腳位輸出低電位
    myIO.In(2, PinLevel);     // 取得第 2 腳位輸入之電位
    myIO.PulseOut(3, 1, 2);   // 由第 3 腳輸出 2ms 的高電位波形

































    myIO.SetADC (1);          // 開啟 PB0 作為 ADC 轉換的輸入
    myIO.GetADC (0, ChValue); // 取得 CH0 的類比輸入轉換為數位的值，存入 ChValue 中

    myIO.SetTimer(0, 1000);   // 設定倒數計時 1 秒
    myIO.TimerOn();
}
```

附錄

1. 已知問題:

2. 模組編號開關對應編號表:

	0		8		16		24
	1		9		17		25
	2		10		18		26
	3		11		19		27
	4		12		20		28
	5		13		21		29
	6		14		22		30
	7		15		23		31