

Joystick 2A

二軸搖桿與搖桿按鍵模組

版本: V2.0



產品介紹: 利基 Joystick 2A 模組提供簡易的設定與位置取得指令,讓使用者可以規劃符合自己需求的搖桿。透過 cmdBUS 與利基的 Ozone 連接,就可以執行各種專屬的應用指令。不論是機器手臂,機器人操縱,都可以直覺的達成。提供直角坐標與極座標兩種定址方式,使用者可以根據需求,任意更換所要的回傳座標系。除了平面的控制,還可以配合按鍵,控制更複雜的應用。

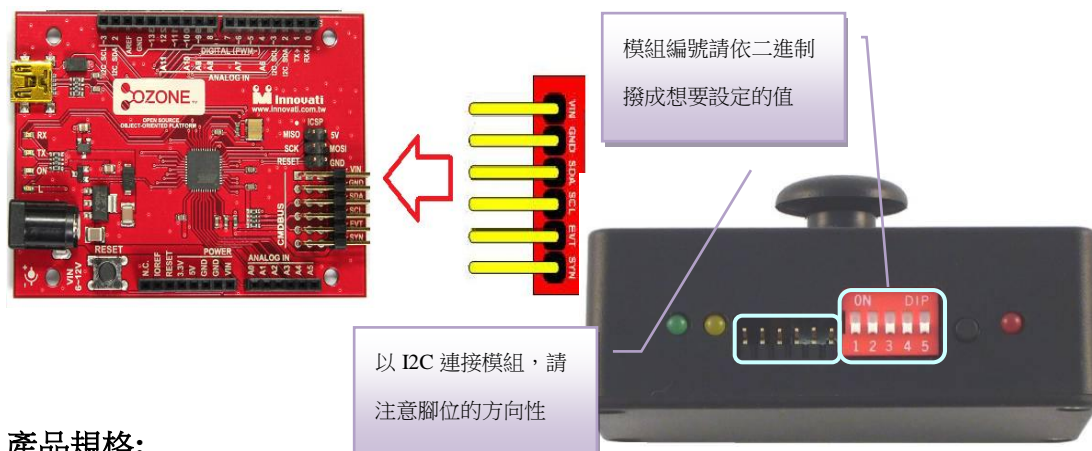
應用方向:

- 連結機器手臂,以極座標直接設定手臂所要旋轉的角度。
- 加裝配合無線輸出,控制各種遙控車,飛機等應用。
- 各種測試機具的操作。
- 搭配馬達模組做馬達加速控制,可以搭配按鍵做定速功能。
- 控制利基應用科技的各項應用套件。

產品特色:

- 設定容易,只要使用 cmdBUS 連接 Ozone,就可以用專屬的指令做各種應用。
- 二軸操作,可以移動搖桿在平面做兩軸的操作。
- 提供兩種回傳座標值:直角坐標與極座標,可以隨時選擇所要的回傳方式,或是混合使用。
- 可以回傳四向與八向搖桿位置,快速直覺應用到各種基本控制。
- 回傳刻度可以設定 128 個刻度值,極座標角度範圍值可以設定 360 個刻度值。
- 提供搖桿按鍵,能將整個搖桿當做一個大按鍵,下押就可以觸動按鍵效果。
- 提供校正功能,並有校正按鈕,操作中可以隨時中斷,進行搖桿的校正。
- 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:

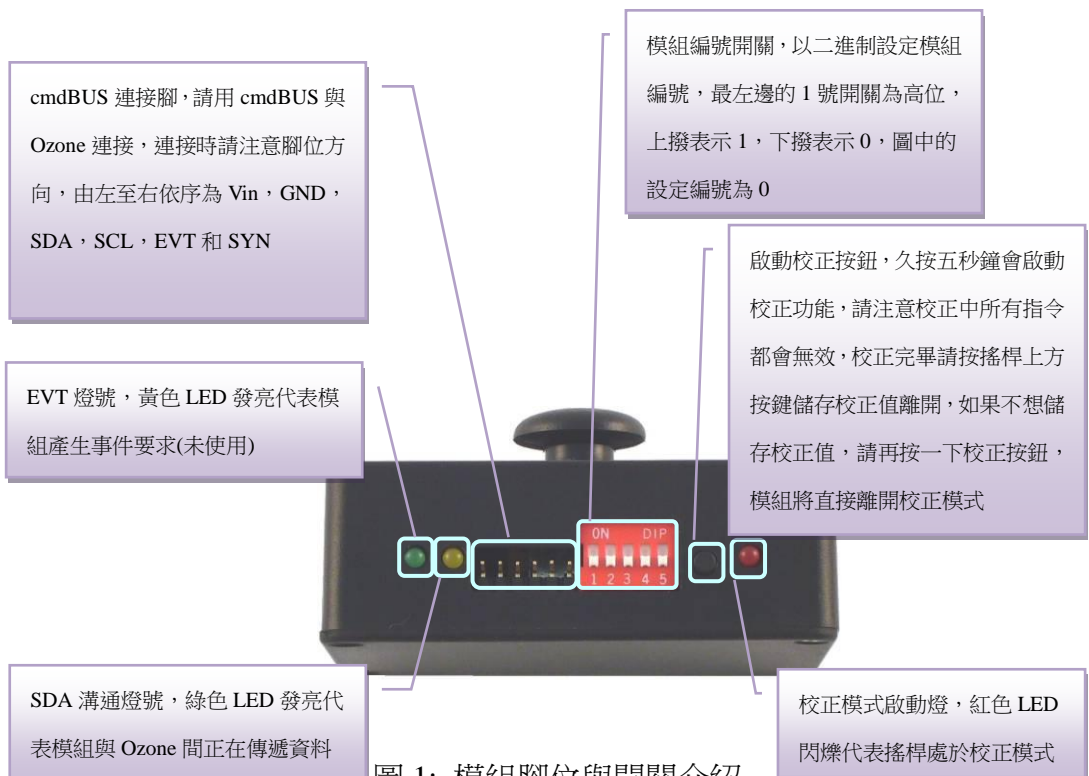


圖 1: 模組腳位與開關介紹

進入校正模式後(紅色 LED 燈閃爍時)，請將搖桿推到頂點，再沿著頂點繞兩圈，以取得 XY 軸向的最大與最小值，最後將搖桿靜置於中心點，等候三秒，讓搖桿記錄完 XY 軸的中心點值，最後再按下搖桿，結束校正模式。
 若是不小心啟動校正模式，可以再按下校正鈕，就可以離開校正模式。
 錯誤的校正可能會造成不正確的回傳值!

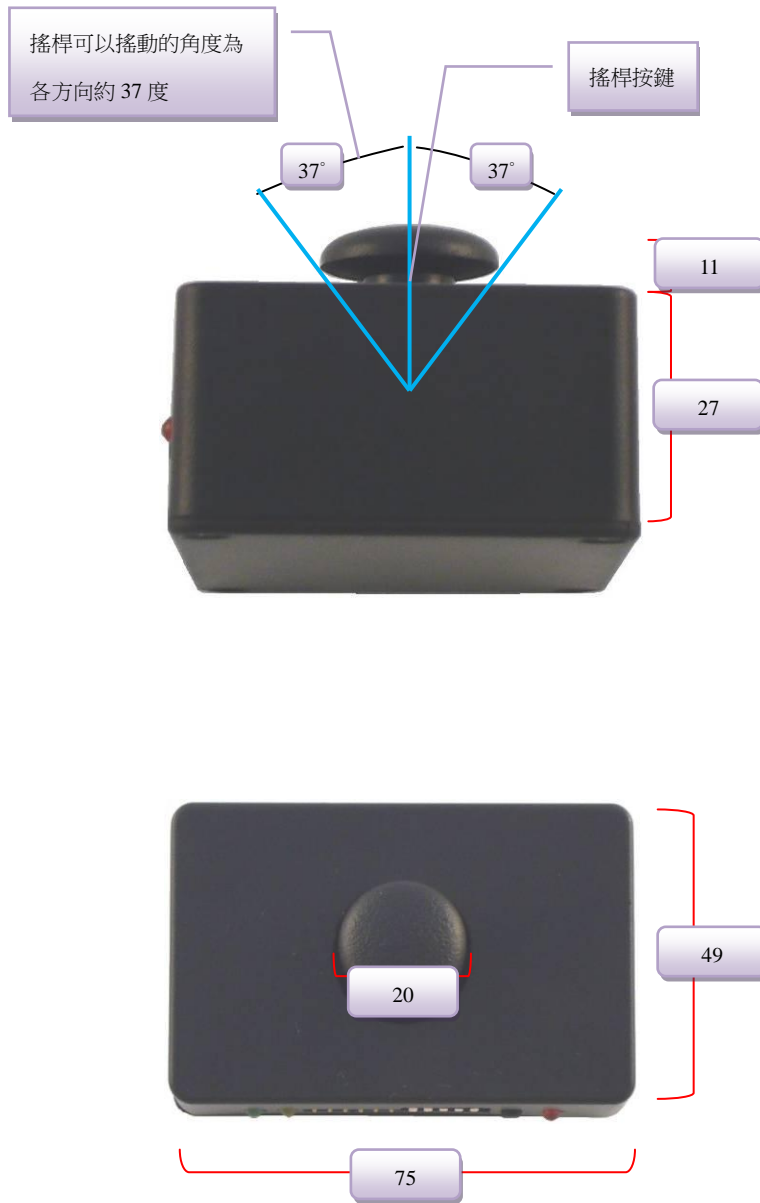


圖 2: 搖桿規格 (單位 mm)

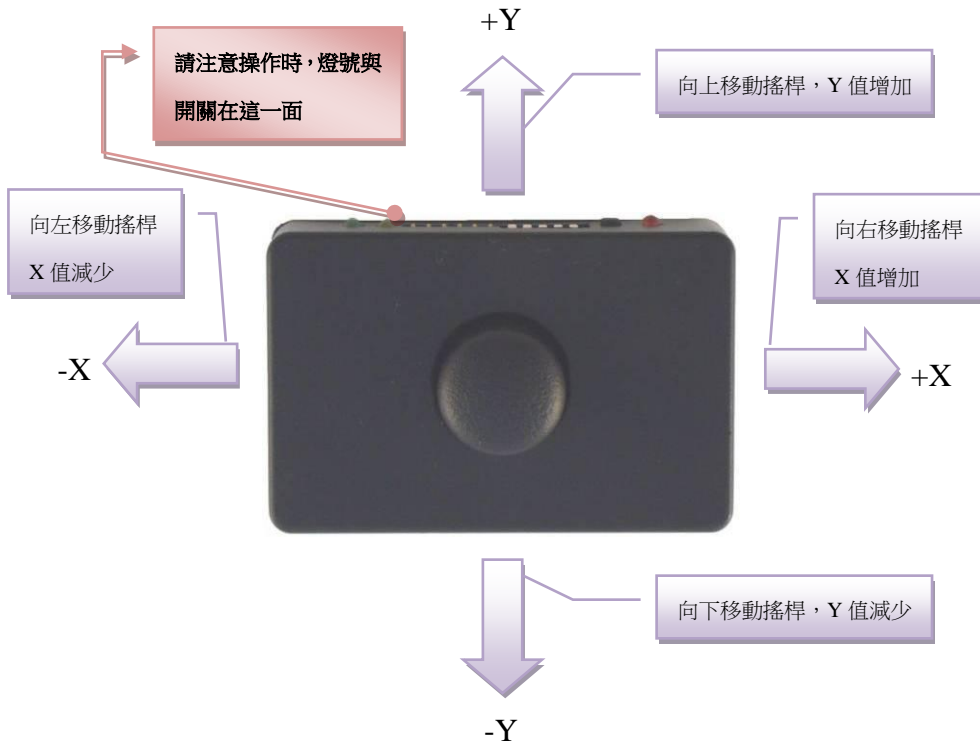


圖 3: 直角座標系操作軸向

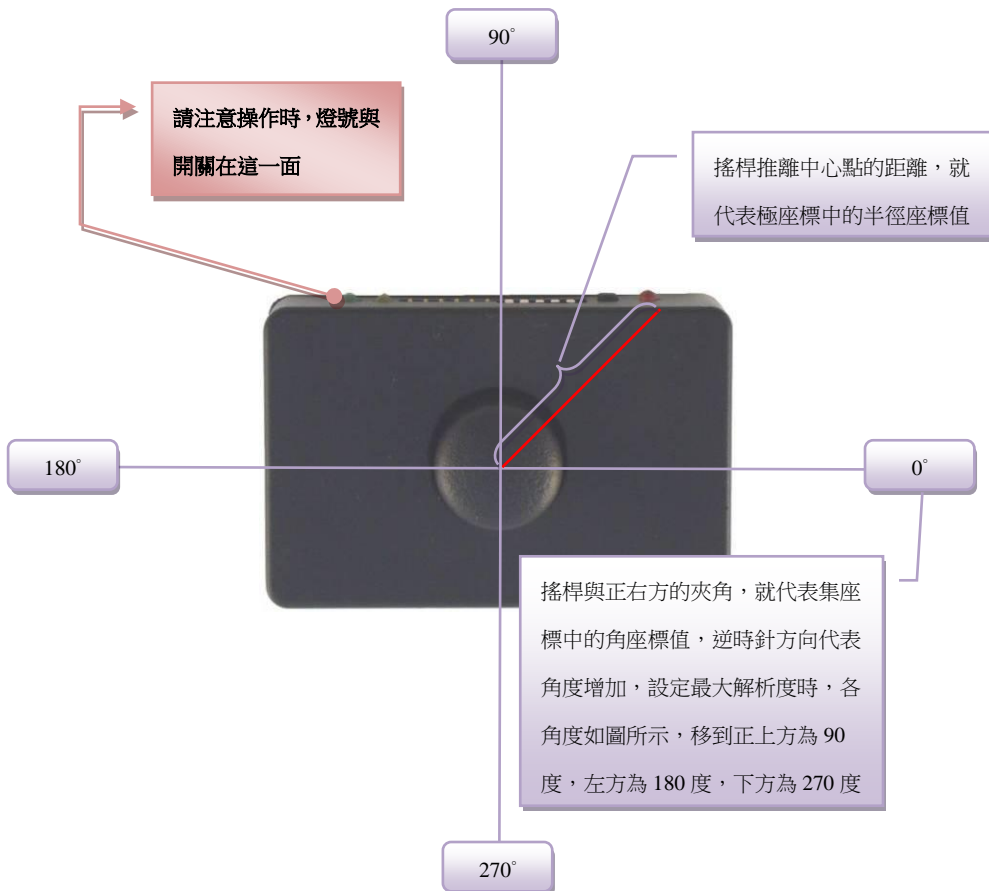


圖 4: 極座標系操作軸向

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{IN}	Conditions				
I _{IN}	Operating Current	7.5	—	—	8	—	mA

表 1: 工作電流特性 (於 25 °C 之環境)

操作注意事項:

操作溫度 0 °C ~ 70 °C

儲存溫度 0 °C ~ 70 °C

指令格式	指令功能
校正搖桿相關指令	
StartCalibration(void)	啟動搖桿校正模式。執行此命令後，搖桿會進入校正模式，此時請將搖桿推到頂點，再沿著頂點繞兩圈，以取得 XY 軸向的最大與最小值，最後將搖桿靜置於中心點，等候三秒，讓搖桿記錄完 XY 軸的中心點值，最後再押下搖桿，結束校正模式。
取得搖桿座標相關指令	
GetXY(int8_t &X, int8_t &Y)	以直角坐標系，取得搖桿現在的 XY 軸座標， X 為 X 軸向的座標刻度值， Y 為 Y 軸向的座標刻度值，兩刻度值預設範圍都是 -127~127。 X ， Y 回傳值為 -127~127 之間的整數值。
GetPolarBinaryRadian(uint8_t &Radius, uint16_t &Angle)	取得整數值表示的搖桿極座標值， Radius 是半徑值，回傳值為 0~127 之間的整數值， Angle 是角度值，回傳值為 0~359 之間的整數值，以 X 軸為零度，逆時針方向增加角度值。半徑刻度預設範圍是 0~127，角度刻度預設範圍是 0~359。
uint8_t Dir = Get4WayStatus(void)	取得以四向表示的搖桿位置。 Dir 是方向值，只會有 0~4 的回傳值，分別代表： 0 → 搖桿位於中心點 1 → 搖桿位置於右方 2 → 搖桿位置於下方

	<p>3 → 搖桿位置於左方 4 → 搖桿位置於上方</p>
<p>uint8_t Dir = Get8WayStatus(void)</p>	<p>取得以八向表示的搖桿位置。<i>Dir</i> 是方向值，只會有 0~8 的回傳值，分別代表：</p> <p>0 → 搖桿位於中心點 1 → 搖桿位置於右方 2 → 搖桿位置於右下方 3 → 搖桿位置於下方 4 → 搖桿位置於左下方 5 → 搖桿位置於左方 6 → 搖桿位置於左上方 7 → 搖桿位置於上方 8 → 搖桿位置於右上方</p>
<p>按鈕應用相關指令</p>	
<p>uint8_t Sta = GetButtonStatus(void)</p>	<p>取得按鍵的狀態，放在 <i>Sta</i> 參數中，回傳值如下：</p> <p>關閉連續按鍵功能時 0: 按鍵被壓下 1: 沒有按按鍵</p> <p>開啟連續按鍵功能時 0: 沒有偵測到新的按鍵 1: 按鍵剛被按下時，或是按鍵久按到達 Repeat Time 所設定的時間，以及久按超過 Repeat Time 設定時間後，每隔 Repeat Rate 所設定的時間，就會再次設定為 1，在沒有執行 GetButtonStatus 指令前，都會保持 1 的值。</p>

範例程式:

(偵測搖桿移動，並在偵測到時，回傳 XY 值)

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

```
JoyStick2A myJoy(10); // 設定模組編號為 10
```

```
int8_t sX, sY;
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```
}
```











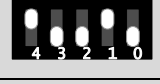





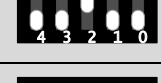
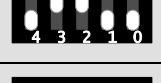
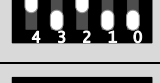
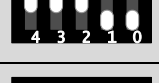




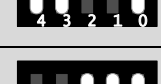



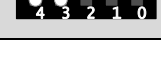
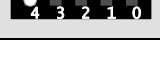

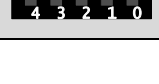
```
void loop()
{
  myJoy.GetXY(sX, sY);
  Serial.print("X=");
  Serial.println(sX);
  Serial.print("Y=");
  Serial.println(sY);
  delay(200);
}
```

附錄

1. 已知問題:

- V1.0 版本，Restore 指令只會將值存於 RAM 中，斷電後重新開啟，會回到最後的設定值。

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