

Gamepad PS

PS2 搖桿控制模組

版本： V2.0



產品介紹：

GamepadPS 模組提供簡易的設定與位置取得指令，搭配 12 個按鈕，讓使用者規劃符合自己需求的操作模式。透過 cmdBUS 與 Ozone 連接，可以用簡單的指令與 PS2 搖桿溝通，取得按鍵資訊製作專屬的應用指令。

應用方向：

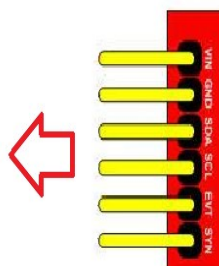
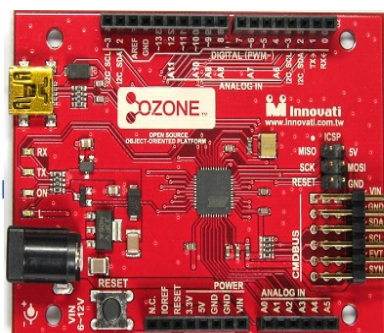
- 連結機器人，設定按鍵達成控制動作與行進等行為。
- 各種測試機具的操作。
- 可與無線 PS2 搖桿結合，控制各種遙控車、飛機等應用。
- 控制利基應用科技的各項應用套件。

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。
- 操縱桿部份，可設定類比回傳、四向或八向操縱桿位置回傳。
- 方向鍵，可設定四向或八向位置回傳。
- 十二個功能鍵，可單獨控制或組合控制。
- 提供校正功能，並有校正按鈕，操作中可以隨時中斷，進行操縱桿的校正。
- 可自行定義搖桿震動強度與時間。
- 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，連接上 PS2 搖桿後，就可透過 Ozone 執行操作。

連接所選用的 PS2 搖桿



產品規格：

連接所選用的 PS2 搖桿

以 cmdBUS 連接模組，請注意腳位的方向性。

cmdBUS 連接腳，請用 cmdBUS 與 Ozone 連接，連接時請注意腳位方向，由左至右依序為：Vin、Gnd、SDA、SCL、EVT、SYN。

模組編號開關，以二進制設定模組編號，最左邊的 1 號開關為高位，上撥表示 1，下撥表示 0，圖中的設定編號為 31。

啟動校正按鈕，久按三到五秒會啟動校正。請注意校正中所有指令都會無效。校正指示燈，當啟動校正時會恆亮，校正結束後熄滅。若為閃爍表示校正失敗。

功能鍵：L1、L2

方向鍵

左邊操縱桿 (L JoyStick)
功能鍵：L3

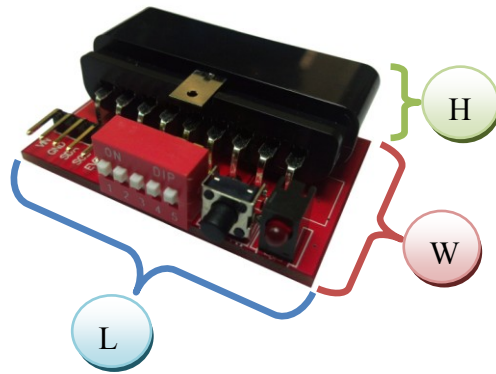
功能鍵：Select、Start

功能鍵：R1、R2

功能鍵：
△、○、×、□

右邊操縱桿 (R JoyStick)
功能鍵：R3





L * W * H : 47 * 31 * 16 (mm)

操作注意事項:

模組操作溫度 -40 °C ~ 123.8 °C

模組儲存溫度 -40 °C ~ 125 °C

本模組適用於原廠 PS2 搖桿，副廠 PS2 搖桿不在保正範圍內。

校正模式操作方式：

- 1：久按校正鈕(或以軟體方式)，進入校正模式。校正燈恆亮。
- 2：請將想要校正的操縱桿推到頂點，再沿著頂點繞兩圈以取得 XY 軸的最大與最小值。
- 3：最後將操縱桿靜置於中心點，等候三秒鐘，讓操縱桿記錄 XY 軸的中心點值。
- 4：按下功能鍵(△、○、×、□)，完成校正。校正燈熄滅。

※若校正燈閃爍，表示校正失敗，請重新校正。

若誤入校正模式，可再按一下校正鈕，離開校正。

指令格式	指令功能
校正搖桿相關指令	
StickCalibration(void)	同時啟動左右邊操縱桿校正模式。 執行此命令後，操縱桿會進入校正模式校正 LED 長亮，此時請將操縱桿推到頂點，再沿著頂點繞兩圈，以取得 XY 軸向的最大與最小值，最後將操縱桿靜置於中心點等候三秒，讓操縱桿記錄完 XY 軸的中心點值，最後再按壓功能鍵離開，校正 LED 燈熄滅完成校正。 ※校正 LED 燈閃爍，表示校正失敗。 功能鍵：△、○、×、□
應用相關指令	
GetLXYPos(int8_t &POSx, int8_t &POSy)	取得左邊操縱桿座標值。 回傳 XY 座標，分別儲存於 POSx，POSy 中，預設範圍為 -127~+127。
GetRXYPos(int8_t &POSx, int8_t &POSy)	取得右邊操縱桿座標值。 回傳 XY 座標，分別儲存於 POSx，POSy 中，預設範圍為 -127~+127。
GetL4WayValue(uint8_t &Dir)	以四向表示方式，取得左邊操縱桿位置。 回傳值存放於 Dir 是方向值，只會有 0~4 的回傳值，分別代表： 0：操縱桿位於中心點 1：操縱桿位於右方→ 2：操縱桿位於下方↓ 3：操縱桿位於左方← 4：操縱桿位於上方↑
GetR4WayValue(uint8_t &Dir)	以四向表示方式，取得右邊操縱桿位置。 回傳值存放於 Dir 是方向值，只會有 0~4 的回傳值，分別代表： 0：操縱桿位於中心點 1：操縱桿位於右方→ 2：操縱桿位於下方↓ 3：操縱桿位於左方← 4：操縱桿位於上方↑
GetL8WayValue(uint8_t &Dir)	以八向表示方式，取得左邊操縱桿位置。 回傳值存放於 Dir 是方向值，只會有 0~8 的回傳值，分別代表： 0：操縱桿位於中心點 1：操縱桿位於右方→ 2：操縱桿位於右下方↘3：操縱桿位於下方↓ 4：操縱桿位於左下方↙5：操縱桿位於左方← 6：操縱桿位於左上方↖7：操縱桿位於上方↑ 8：操縱桿位於右上方↗

GetR8WayValue(uint8_t &Dir)	以八向表示方式，取得右邊操縱桿位置。 回傳值存放於 Dir 是方向值，只會有 0~8 的回傳值，分別代表： 0：操縱桿位於中心點 1：操縱桿位於右方→ 2：操縱桿位於右下方↘3：操縱桿位於下方↓ 4：操縱桿位於左下方↙5：操縱桿位於左方← 6：操縱桿位於左上方↖7：操縱桿位於上方↑ 8：操縱桿位於右上方↗																																								
uint8_t Status = GetKeyStatus(void)	取得按鈕狀態存放於 Status 中。 啟動 = 1，關閉 = 0 <table><tr><td rowspan="13">Status</td><td>Bit</td><td>對應按鈕</td><td>十進制</td></tr><tr><td>0</td><td>△</td><td>1</td></tr><tr><td>1</td><td>○</td><td>2</td></tr><tr><td>2</td><td>×</td><td>4</td></tr><tr><td>3</td><td>□</td><td>8</td></tr><tr><td>4</td><td>L1</td><td>16</td></tr><tr><td>5</td><td>R1</td><td>32</td></tr><tr><td>6</td><td>L2</td><td>64</td></tr><tr><td>7</td><td>R2</td><td>128</td></tr><tr><td>8</td><td>Select</td><td>256</td></tr><tr><td>9</td><td>Start</td><td>512</td></tr><tr><td>10</td><td>L3</td><td>1024</td></tr><tr><td>11</td><td>R3</td><td>2048</td></tr></table> EX：若 Status = 3 則△、○被啟動。	Status	Bit	對應按鈕	十進制	0	△	1	1	○	2	2	×	4	3	□	8	4	L1	16	5	R1	32	6	L2	64	7	R2	128	8	Select	256	9	Start	512	10	L3	1024	11	R3	2048
Status	Bit		對應按鈕	十進制																																					
	0		△	1																																					
	1		○	2																																					
	2		×	4																																					
	3		□	8																																					
	4		L1	16																																					
	5		R1	32																																					
	6		L2	64																																					
	7		R2	128																																					
	8		Select	256																																					
	9		Start	512																																					
	10		L3	1024																																					
	11	R3	2048																																						
GetDir4Way(uint8_t &Dir)	取得方向鍵狀態，以四向方式回傳。 回傳值存放於 Dir，只會有 0~4 的回傳值，分別代表： 0：無方向 1：右方→ 2：下方↓ 3：左方← 4：上方↑																																								
GetDir8Way(uint8_t &Dir)	取得方向鍵狀態，以八向方式回傳。 回傳值存放於 Dir，只會有 0~8 的回傳值，分別代表： 0：無方向 1：右方→ 2：右下方↘ 3：下方↓ 4：左下方↙ 5：左方← 6：左上方↖ 7：上方↑ 8：右上方↗																																								
StartVib(uint8_t Time, uint8_t Level)	啟動搖桿震動功能。經由 Time 設定震動時間，Level 設定震動強度。 Time：可設定範圍為 0~255 的整數。 0：持續指動直到下達 StopVib 指令。 1 為 1 秒，之後每加 1 則增加 100 ms。																																								

	<i>Level</i> ：可設定範圍為 0~255 的整數。 0：不震動，1~255 數字越大震動越強。
StopVib(void)	停止搖桿震動功能。

範例程式：

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

```
GamepadPS PS(2);          //    設定模組編號為 2
```

```
uint16_t wStatus;
```

```
uint8_t b4Dir;
```

```
void setup()
```

```
{
```

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

```
}
```

```
void loop()
```

```
{
```

```
    wStatus = PS.GetKeyStatus();
```

```
    Serial.print("KeyStatus=");
```

```
    Serial.println(wStatus);
```

```
    PS.GetDir4Way(b4Dir);
```

```
    Serial.print("Dir=");
```

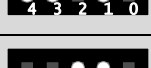
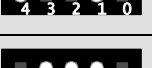
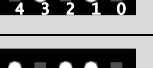
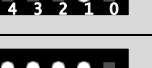
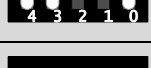
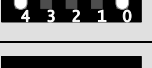
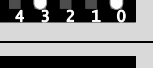
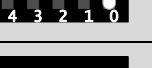
```
    Serial.println(b4Dir);
```

```
    delay(200);
```

```
}
```

附錄

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