

# 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 Arminno 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：

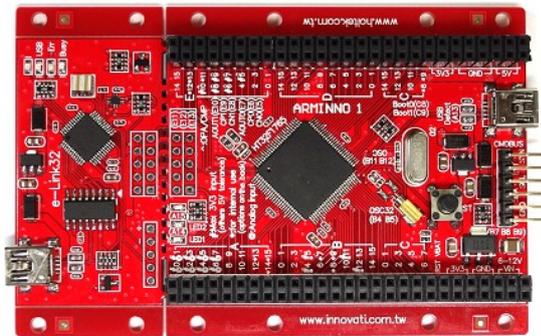
- 連結機器人，設定按鍵達成控制動作與行進等行為。
- 各種測試機具的操作。
- 可與無線 PS2 搖桿結合，控制各種遙控車、飛機等應用。
- 控制利基應用科技的各項應用套件。

### 產品特色：

- 設定容易，只要使用 cmdBUS 連接 Arminno，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 cmdBUS 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 Arminno 上對應的腳位，連接上 PS2 搖桿後，就可透過 Arminno 執行操作。

連接所選用的 PS2 搖桿



產品規格：

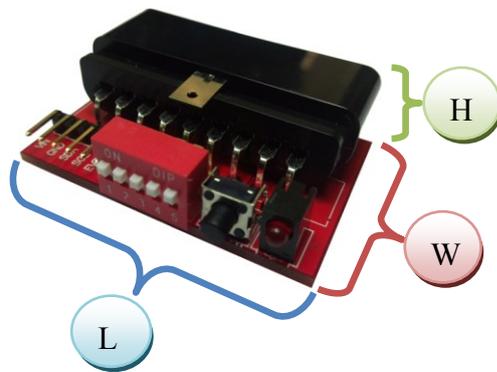
連接所選用的 PS2 搖桿

以 cmdBUS 連接模組，請注意腳位的方向性。

cmdBUS 連接腳，請用 cmdBUS 與 Arminno 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。



**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  |   |
| <p style="text-align: center;"><b>StickCalibration(void)</b></p>                        | <p>同時啟動左右邊操縱桿校正模式。<br/>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。<br/>※校正 LED 燈閃爍，表示校正失敗。<br/>功能鍵：△、○、×、□</p>   |
| 應用相關指令  |   |
| <p style="text-align: center;"><b>GetLXYPos(int8_t &amp;POSx, int8_t &amp;POSy)</b></p> | <p>取得<b>左邊</b>操縱桿座標值。<br/>回傳 XY 座標，分別儲存於 <b>POSx</b>，<b>POSy</b> 中，預設範圍為 -127~+127。</p>   |
| <p style="text-align: center;"><b>GetRXYPos(int8_t &amp;POSx, int8_t &amp;POSy)</b></p> | <p>取得<b>右邊</b>操縱桿座標值。<br/>回傳 XY 座標，分別儲存於 <b>POSx</b>，<b>POSy</b> 中，預設範圍為 -127~+127。</p>   |
| <p style="text-align: center;"><b>GetL4WayValue(uint8_t &amp;Dir)</b></p>               | <p>以四向表示方式，取得<b>左邊</b>操縱桿位置。<br/>回傳值存放於 <b>Dir</b> 是方向值，只會有 0~4 的回傳值，分別代表：<br/><b>0</b>：操縱桿位於中心點   <b>1</b>：操縱桿位於右方→<br/><b>2</b>：操縱桿位於下方↓   <b>3</b>：操縱桿位於左方←<br/><b>4</b>：操縱桿位於上方↑</p>  |
| <p style="text-align: center;"><b>GetR4WayValue(uint8_t &amp;Dir)</b></p>               | <p>以四向表示方式，取得<b>右邊</b>操縱桿位置。<br/>回傳值存放於 <b>Dir</b> 是方向值，只會有 0~4 的回傳值，分別代表：<br/><b>0</b>：操縱桿位於中心點   <b>1</b>：操縱桿位於右方→<br/><b>2</b>：操縱桿位於下方↓   <b>3</b>：操縱桿位於左方←<br/><b>4</b>：操縱桿位於上方↑</p>  |
| <p style="text-align: center;"><b>GetL8WayValue(uint8_t &amp;Dir)</b></p>               | <p>以八向表示方式，取得<b>左邊</b>操縱桿位置。<br/>回傳值存放於 <b>Dir</b> 是方向值，只會有 0~8 的回傳值，分別代表：<br/><b>0</b>：操縱桿位於中心點   <b>1</b>：操縱桿位於右方→<br/><b>2</b>：操縱桿位於右下方↘<b>3</b>：操縱桿位於下方↓<br/><b>4</b>：操縱桿位於左下方↙<b>5</b>：操縱桿位於左方←<br/><b>6</b>：操縱桿位於左上方↖<b>7</b>：操縱桿位於上方↑<br/><b>8</b>：操縱桿位於右上方↗</p> |

| <p><b>GetR8WayValue(uint8_t &amp;Dir)</b></p>       | <p>以八向表示方式，取得右邊操縱桿位置。<br/>回傳值存放於 <i>Dir</i> 是方向值，只會有 0~8 的回傳值，分別代表：</p> <p>0：操縱桿位於中心點    1：操縱桿位於右方→<br/>2：操縱桿位於右下方↘ 3：操縱桿位於下方↓<br/>4：操縱桿位於左下方↙ 5：操縱桿位於左方←<br/>6：操縱桿位於左上方↖ 7：操縱桿位於上方↑<br/>8：操縱桿位於右上方↗</p>  |        |      |      |     |               |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |   |    |    |   |    |    |   |    |     |   |        |     |   |       |     |    |    |      |    |    |      |
|---|---|--------|------|------|-----|---------------|---|---|---|---|---|---|---|---|---|---|---|---|---|----|----|---|----|----|---|----|----|---|----|-----|---|--------|-----|---|-------|-----|----|----|------|----|----|------|
| <p><b>uint8_t Status = GetKeyStatus(void)</b></p>   | <p>取得按鈕狀態存放於 <i>Status</i> 中。<br/>啟動 = 1，關閉 = 0</p> <table border="1" data-bbox="794 645 1433 1283"> <thead> <tr> <th></th> <th>Bit</th> <th>對應按鈕</th> <th>十進制</th> </tr> </thead> <tbody> <tr> <td rowspan="12" style="text-align: center; vertical-align: middle;"><i>Status</i></td> <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> </tbody> </table> <p>EX：若 <i>Status</i> = 3 則△、○被啟動。</p> |        | Bit  | 對應按鈕 | 十進制 | <i>Status</i> | 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 |
|   | Bit   | 對應按鈕   | 十進制  |      |     |               |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |   |    |    |   |    |    |   |    |     |   |        |     |   |       |     |    |    |      |    |    |      |
| <i>Status</i>                                       | 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 |      |     |               |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |   |    |    |   |    |    |   |    |     |   |        |     |   |       |     |    |    |      |    |    |      |
| <p><b>GetDir4Way(uint8_t &amp;Dir)</b></p>          | <p>取得方向鍵狀態，以四向方式回傳。<br/>回傳值存放於 <i>Dir</i>，只會有 0~4 的回傳值，分別代表：</p> <p>0：無方向    1：右方→    2：下方↓<br/>3：左方←    4：上方↑</p>  |        |      |      |     |               |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |   |    |    |   |    |    |   |    |     |   |        |     |   |       |     |    |    |      |    |    |      |
| <p><b>GetDir8Way(uint8_t &amp;Dir)</b></p>          | <p>取得方向鍵狀態，以八向方式回傳。<br/>回傳值存放於 <i>Dir</i>，只會有 0~8 的回傳值，分別代表：</p> <p>0：無方向    1：右方→    2：右下方↘<br/>3：下方↓    4：左下方↙    5：左方←<br/>6：左上方↖    7：上方↑    8：右上方↗</p>   |        |      |      |     |               |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |   |    |    |   |    |    |   |    |     |   |        |     |   |       |     |    |    |      |    |    |      |
| <p><b>StartVib(uint8_t Time, uint8_t Level)</b></p> | <p>啟動搖桿震動功能。經由 <i>Time</i> 設定震動時間，<i>Level</i> 設定震動強度。</p> <p><i>Time</i>：可設定範圍為 0~255 的整數。<br/>0：持續指動直到下達 StopVib 指令。<br/>1 為 1 秒，之後每加 1 則增加 100 ms。</p>   |        |      |      |     |               |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |   |    |    |   |    |    |   |    |     |   |        |     |   |       |     |    |    |      |    |    |      |

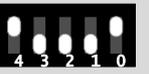
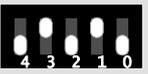
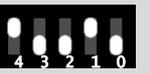
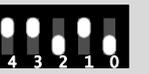
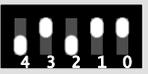
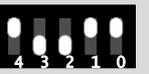
|                      |  |
|----------------------|--|
|                      | <i>Level</i> ：可設定範圍為 0~255 的整數。<br>0：不震動，1~255 數字越大震動越強。 |
| <b>StopVib(void)</b> | 停止搖桿震動功能。  |

### 範例程式:

```
#include "arminno.h"
GamepadPS PS (2);      // 設定模組編號為 2
unsigned short wStatus;
unsigned char b4Dir;
int main(void)
{
    while(1)
    {
        wStatus = PS.GetKeyStatus();
        printf("KeyStatus= %d\r\n", wStatus);
        PS.GetDir4Way(b4Dir);
        printf("Dir=%d\r\n", b4Dir);
        Pause(2000);
    }
}
```

# 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 |