

# Gamepad BT

## 藍牙搖桿控制模組

版本： V2.0



### 產品介紹：

GamepadBT 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藍牙搖桿溝通，取得按鍵資訊製作專屬的應用指令。

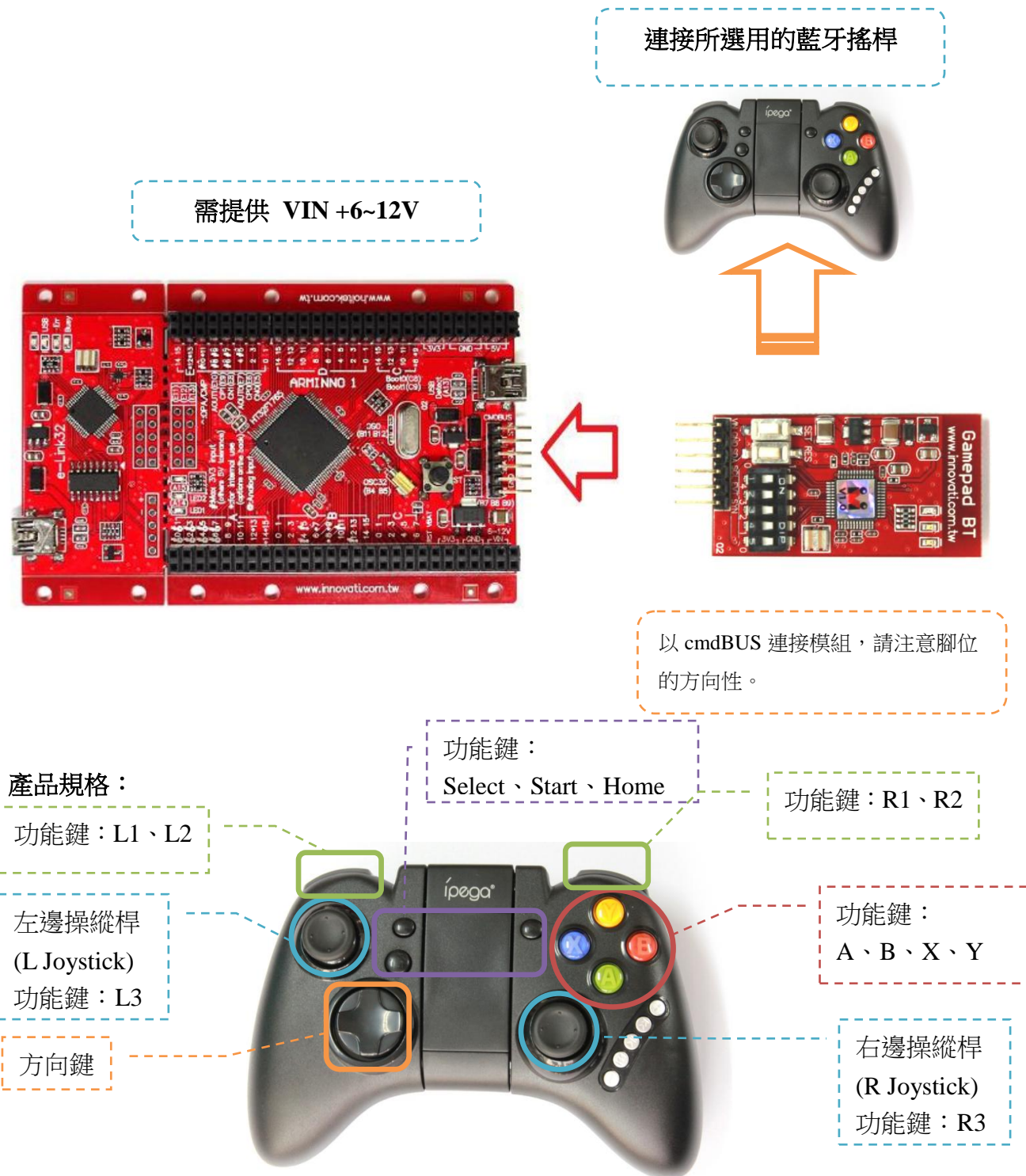
### 應用方向：

- 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。
- 與無線搖桿結合，控制各種搖控車、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。
- 可自行定義按鈕功能，包括按鍵的連續觸發起動時間，以及連續觸發的速率，都可透過指令設定。
- 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 Arminno 上對應的腳位，連接上藍牙搖桿後，就可透過 Arminno 執行操作。



#### 操作注意事項:

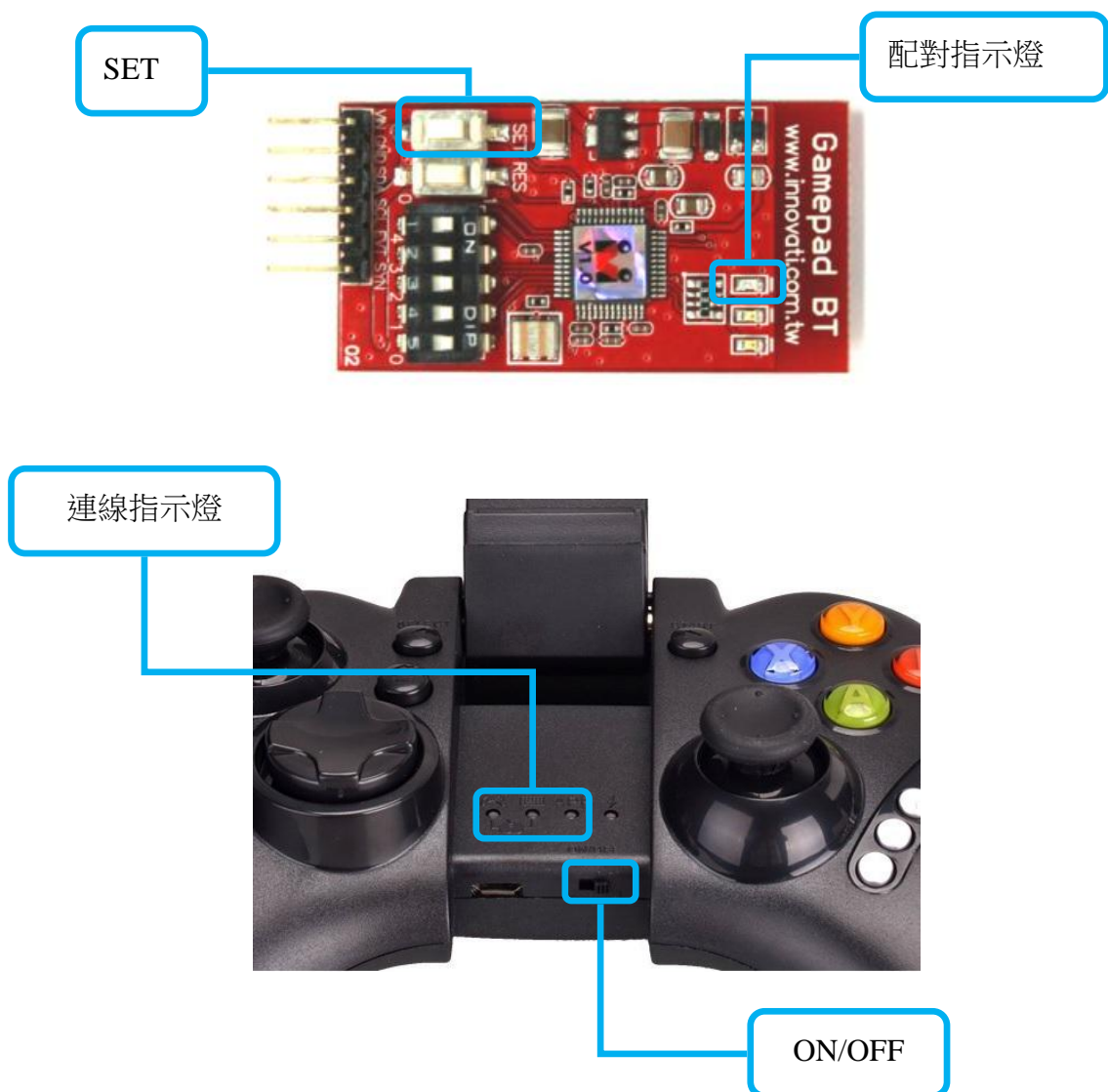
模組操作溫度 -40 °C ~ 123.8 °C

模組儲存溫度 -40 °C ~ 125 °C

本模組適用於原廠藍牙搖桿，副廠藍牙搖桿不在保證範圍內。

## 如何配對

1. 模組接上 CmdBUS 並提供電源，請確認模組的**配對指示燈**保持閃爍。
  2. 打開手把中間的蓋子，將電源開關撥至 ON。此時手把上的所有 LED 燈仍保持熄滅。  
如未熄滅，請重新 OFF-ON 一次，或是長按住 HOME 讓手機進入待機模式。
  3. 在待機模式下先按住手把 X 並保持，再按住 HOME，待手把上**連線指示燈**開始閃爍即可全部放開。**連線指示燈**會閃爍一段時間後保持一顆恒亮，即表示配對成功
- 註：1) 如果手把進入睡眠模式(LED 燈全滅) -可按住 X，再按住 HOME 來喚醒手把。  
2) 如果模組進入睡眠模式(**配對燈滅**)，或是一直無法成功配對-(二擇一)  
2-1 可重新給 CmdBUS 上電。  
2-2 或是按住 RES 鍵約 6 秒。放開 RES 後再快速按模組上的 **SET** 鍵兩次。  
此時模組**配對指示燈**會快速閃爍並**重新**進入配對模式。



指令格式	指令功能																																								
設定相關指令																																									
SetKeyRepeatFunc(uint16_t Key_ID)	設定是否啟動重複輸入判定。 啟動 = 1，關閉 = 0																																								
	<table><tr><td rowspan="12">Key_ID</td><td>Bit</td><td>對應按鈕</td><td>十進制</td></tr><tr><td>0</td><td>Y</td><td>1</td></tr><tr><td>1</td><td>B</td><td>2</td></tr><tr><td>2</td><td>A</td><td>4</td></tr><tr><td>3</td><td>X</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>	Key_ID	Bit	對應按鈕	十進制	0	Y	1	1	B	2	2	A	4	3	X	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
	Key_ID		Bit	對應按鈕	十進制																																				
			0	Y	1																																				
			1	B	2																																				
			2	A	4																																				
			3	X	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																																						
EX：若想啟動 Y、B 則 Key_ID 可設定為： &B11(二進制)，或 3(十進制)。																																									
GetKeyRepeatFunc(uint16_t& Key_ID)	取得是否啟動重複輸入判定設定。 啟動 = 1，關閉 = 0																																								
	<table><tr><td rowspan="12">Key_ID</td><td>Bit</td><td>對應按</td><td>十進制</td></tr><tr><td>0</td><td>Y</td><td>1</td></tr><tr><td>1</td><td>B</td><td>2</td></tr><tr><td>2</td><td>A</td><td>4</td></tr><tr><td>3</td><td>X</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>	Key_ID	Bit	對應按	十進制	0	Y	1	1	B	2	2	A	4	3	X	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
	Key_ID		Bit	對應按	十進制																																				
			0	Y	1																																				
			1	B	2																																				
			2	A	4																																				
			3	X	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																																						
SetRepeatTime(uint8_t Time)	設定重複輸入判定時間值。 經由 Time 設定，可輸入範圍為 0~255 之間的整數值，單位為 10 ms。																																								
	取得重複輸入判定時間設定值。 回傳值存放於 Time。回傳範圍為 0~255 之間的整數值，單位為 10 ms。																																								

<b>SetRepeatRate(uint8_t Rate)</b>	設定重複輸入判定速率值。 經由 <b>Rate</b> 設定，可輸入範圍為 0~255 之間的整數值，單位為 10 ms。
<b>GetRepeatRate(uint8_t&amp; Rate)</b>	取得重複輸入判定速率設定值。 回傳值存放於 <b>Rate</b> 。回傳範圍為 0~255 之間的整數值，單位為 10 ms。
<b>應用相關指令</b>	
<b>GetLXYPos(int8_t&amp; POSx, int8_t&amp; POSy)</b>	取得 <b>左邊</b> 操縱桿座標值。 回傳 XY 座標，分別儲存於 <b>POSx</b> ， <b>POSy</b> 中，預設範圍為 -127~+127。
<b>GetRXYPos(int8_t&amp; POSx, int8_t&amp; POSy)</b>	取得 <b>右邊</b> 操縱桿座標值。 回傳 XY 座標，分別儲存於 <b>POSx</b> ， <b>POSy</b> 中，預設範圍為 -127~+127。
<b>GetL4WayValue(uint8_t&amp; Dir)</b>	以四向表示方式，取得 <b>左邊</b> 操縱桿位置。 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： <b>0</b> ：操縱桿位於中心點 <b>1</b> ：操縱桿位於右方→ <b>2</b> ：操縱桿位於下方↓ <b>3</b> ：操縱桿位於左方← <b>4</b> ：操縱桿位於上方↑
<b>GetR4WayValue(uint8_t&amp; Dir)</b>	以四向表示方式，取得 <b>右邊</b> 操縱桿位置。 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： <b>0</b> ：操縱桿位於中心點 <b>1</b> ：操縱桿位於右方→ <b>2</b> ：操縱桿位於下方↓ <b>3</b> ：操縱桿位於左方← <b>4</b> ：操縱桿位於上方↑
<b>GetL8WayValue(uint8_t&amp; Dir)</b>	以八向表示方式，取得 <b>左邊</b> 操縱桿位置。 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： <b>0</b> ：操縱桿位於中心點 <b>1</b> ：操縱桿位於右方→ <b>2</b> ：操縱桿位於右下方↘ <b>3</b> ：操縱桿位於下方↓ <b>4</b> ：操縱桿位於左下方↙ <b>5</b> ：操縱桿位於左方← <b>6</b> ：操縱桿位於左上方↖ <b>7</b> ：操縱桿位於上方↑ <b>8</b> ：操縱桿位於右上方↗
<b>GetR8WayValue(uint8_t&amp; Dir)</b>	以八向表示方式，取得 <b>右邊</b> 操縱桿位置。 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：




























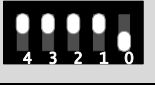
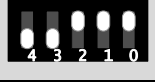


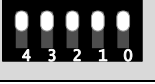
	<p>0：操縱桿位於中心點    1：操縱桿位於右方→ 2：操縱桿位於右下方↘ 3：操縱桿位於下方↓ 4：操縱桿位於左下方↙ 5：操縱桿位於左方← 6：操縱桿位於左上方↖ 7：操縱桿位於上方↑ 8：操縱桿位於右上方↗</p>																																								
<p>uint16_t Status = GetKeyStatus(void)</p>	<p>取得按鈕狀態存放於 <i>Status</i> 中。 啟動 = 1，關閉 = 0</p> <table><tr><td rowspan="12"><i>Status</i></td><td>Bit</td><td>對應按鈕</td><td>十進制</td></tr><tr><td>0</td><td>Y</td><td>1</td></tr><tr><td>1</td><td>B</td><td>2</td></tr><tr><td>2</td><td>A</td><td>4</td></tr><tr><td>3</td><td>X</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> <p>EX：若 Status = 3 則 Y、B 被啟動。</p>	<i>Status</i>	Bit	對應按鈕	十進制	0	Y	1	1	B	2	2	A	4	3	X	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
<i>Status</i>	Bit		對應按鈕	十進制																																					
	0		Y	1																																					
	1		B	2																																					
	2		A	4																																					
	3		X	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>GetDir4Way(uint8_t&amp; Dir)</p>	<p>取得方向鍵狀態，以四向方式回傳。 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↓ 3：左方←      4：上方↑</p>																																								
<p>GetDir8Way(uint8_t&amp; Dir)</p>	<p>取得方向鍵狀態，以八向方式回傳。 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↘ 3：下方↓      4：左下方↙    5：左方← 6：左上方↖    7：上方↑      8：右上方↗</p>																																								
<p>GetConnect(uint8_t&amp; Status)</p>	<p>取得控制器連接狀態。 存放於 <i>Status</i> 中，回傳值為 0、1。 分別代表：</p> <p>0：沒有偵測到搖桿，1：搖桿正常連接</p>																																								

## 範例程式:

```
#include "arminno.h"
GamepadBT BT(31);           // 設定模組編號
uint8_t b4Dir;              // 儲存取得的方向值
uint8_t b8WayL, b8WayR;     // 儲存取得的操縱桿方向值
uint16_t wStatus;           // 儲存取得的按鈕狀態值
int main(void)
{
    while(1) {
        BT.GetDir4Way(b4Dir);           // 以四向回傳方式，取得方向鍵狀態
        printf("Dir4=%d\r\n", b4Dir);   // 顯示
        BT.GetR8WayValue(b8WayR);       // 以八向回傳方式，取得右邊操縱桿狀態
        printf("Dir8R=%d\r\n", b8WayR); // 顯示
        BT.GetL8WayValue(b8WayL);       // 以八向回傳方式，取得左邊操縱桿狀態
        printf("Dir8L=%d\r\n", b8WayL); // 顯示
        wStatus = BT.GetKeyStatus();     // 取得按鍵狀態
        printf("Key Status %04X\r\n", wStatus); // 顯示
        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