

SR1

語音辨識模組

版本: V2.0

產品介紹: 利基 SR1 語音辨識模組，可自行錄製語音命令，透過 CmdBUS 與 Basic Commander 連接，達到語音控制的應用。

應用方向:

- 取代一般按鍵式開關，透過語音指令達到控制目的。

產品特色:

- 可自行定義四十組語音指令。
- 內建十組語音指令。
- 操作簡單。
- 使用漢語拼音輸入。

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

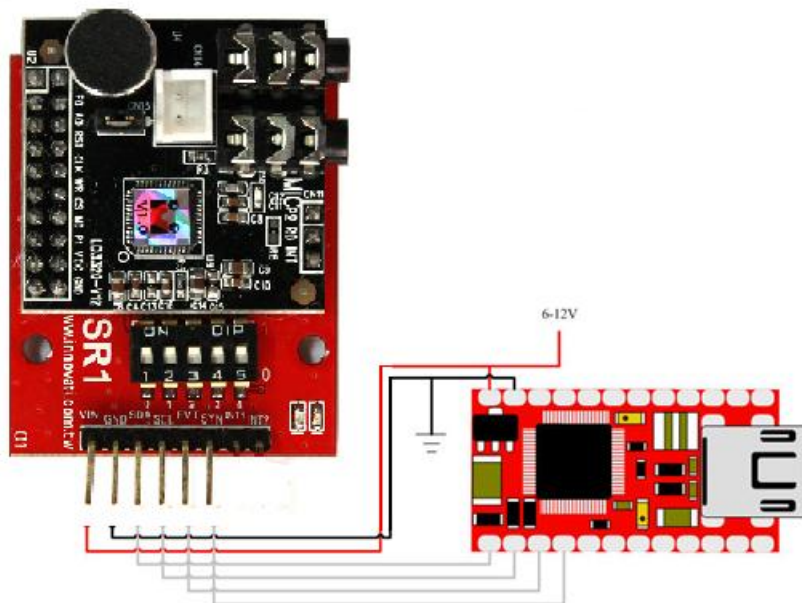


圖 1: 模組連接範例

產品規格:

模組編號設定開關，由右至左以二進制設定SR1的模組編號，編號可以讓BASIC Commander®操控時，判斷想要控制的模組。

小型麥克風
語音輸入用



cmdBUS™接腳，將此處腳位與BASIC Commander®對應腳位相接，即可透過BASIC Commander®操控SR1(連接時請注意腳位對應，將Vin對接BASIC Commander®上的Vin腳位，若是腳位錯誤可能造成模組損毀)

圖 2: 模組腳位與開關介紹

操作注意事項:

模組操作溫度 0 °C ~ 70 °C

模組儲存溫度 -50 °C ~ 125 °C

模組下達指令的方式可分為兩種：cmdBUS、I2C 控制方式

cmdBUS 指令表:

下面的指令表是專供控制 SR1 的各種指令，必要輸入的指令名稱與參數，以粗底或粗斜體表示，粗體的文字在輸入時請不要更改，粗斜體的文字請自行定義適當格式的參數填入。輸入時請注意 innoBASIC™ Workshop 大寫與小寫會視為相同字。

在執行 SR1 指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral *ModuleName* As SR1 @ *ModuleID*

I2C 通訊協議(Protocol):

為了使更廣泛的使用者能控制模組，提供了部份指令的通訊協議讓使用者應用。透過通訊規格，使用者可使用 I2C 通訊協議為模組下達命令。

通訊協議常見的封包如下：

MID：模組 ID 編號，空間大小為 Byte 的變數。對應於硬體的指播開關。

CID：命令 ID 編號，空間大小為 Byte 的變數。依不同命令而改變。

Checksum1：驗證位元_1，空間大小為 Byte 的變數。

定義方式： $255 - (MID * 2) - CID$

Checksum2：驗證位元_2，空間大小為 Byte 的變數。

定義方式： $255 - (\text{Checksum1} \sim \text{Checksum2} \text{ 之間的變數總和})$

Checksum3：驗證位元_3，空間大小為 Byte 的變數。

定義方式： $255 - MID - (\text{MID} \sim \text{Checksum3} \text{ 之間的變數總和})$

Dummy：虛設位元，可為任意變數。空間大小為 Byte 的變數。

於通訊規格每筆資料空間大小階為 **Byte**，若資料空間大小超過一個 Byte 時，需將資料拆開，並由 **Low Byte** 開始傳送。

Ex: 傳送資料 Temp 為一筆空間大小為 Word 的資料，則需將 Temp 拆開，分為 Temp_L、Temp_H，並且先傳送 **Temp_L**。

Ex1 模組編號為 2，命令編號為 153，傳送參數 Byte 為 100，通訊協議為

MID+CID+Checksum1+Byte+Checksum2+Dummy 則：

MID = 2

CID = 153

Checksum1 = $255 - (2 * 2) - 153 = 98$

Byte = 100

Checksum2 = $255 - 100$

Dummy = 0~255 之間的任意數

Ex2 模組編號為 2，命令編號為 153，傳送參數 Temp 為 511，通訊協議為

MID+CID+Checksum1+Temp_L+Temp_H+Checksum2+Dummy 則：

MID = 2

CID = 153

Checksum1 = $255 - (2 * 2) - 153 = 98$

Temp_L = 255，Temp_H = 1

Checksum2 = $255 - \text{Temp}_L - \text{Temp}_H = 255$

Dummy = 0~255 之間的任意數

指令格式	指令功能
語句設定相關指令	
CmdBUS： LoadDefaultSentence(Mode)	載入預設語句 <i>Mode</i> : 第幾套預設語句。預設語句會安排在 <i>ID</i> =41~50。
I2C： MID+88+Checksum1	<i>Mode</i> = 0，不使用內建語句(預設)。 = 1，使用以下預設語句。

<p>+Mode+Checksum2+Dummy</p>	<table border="1"> <tr> <td style="text-align: center;"><i>Mode</i></td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;"><i>ID</i></td> <td></td> </tr> <tr> <td style="text-align: center;">41</td> <td>機器人前進</td> </tr> <tr> <td style="text-align: center;">42</td> <td>機器人退後</td> </tr> <tr> <td style="text-align: center;">43</td> <td>機器人停止</td> </tr> <tr> <td style="text-align: center;">44</td> <td>機器人左轉</td> </tr> <tr> <td style="text-align: center;">45</td> <td>機器人右轉</td> </tr> <tr> <td style="text-align: center;">46</td> <td>機器人左移</td> </tr> <tr> <td style="text-align: center;">47</td> <td>機器人右移</td> </tr> <tr> <td style="text-align: center;">48</td> <td>機器人站起來</td> </tr> <tr> <td style="text-align: center;">49</td> <td>機器人蹲下</td> </tr> <tr> <td style="text-align: center;">50</td> <td>機器人攻擊</td> </tr> </table>	<i>Mode</i>	1	<i>ID</i>		41	機器人前進	42	機器人退後	43	機器人停止	44	機器人左轉	45	機器人右轉	46	機器人左移	47	機器人右移	48	機器人站起來	49	機器人蹲下	50	機器人攻擊	
<i>Mode</i>	1																									
<i>ID</i>																										
41	機器人前進																									
42	機器人退後																									
43	機器人停止																									
44	機器人左轉																									
45	機器人右轉																									
46	機器人左移																									
47	機器人右移																									
48	機器人站起來																									
49	機器人蹲下																									
50	機器人攻擊																									
<p>CmdBUS : <i>State=PutSentence(Sentence)</i></p> <hr/> <p>I2C : Out : MID+89+Checksum1+ Total_Len+Sentence_Len+ Sentence(n)+Sentence(n-1)+....+ Sentence(0)+ Checksum2+Dummy</p> <p>In : MID+State+Checksum3</p> <p>Total_Len = n+4 Sentence_Len = n+1 n = 0 ~19</p>		<p>一次暫存一整個語句到暫存語句中。</p> <p>Sentence 的大小限制 20 字之內。</p> <p>如整個語句超過 20 字，請使用 PutWord 一個字一個字輸入。</p> <p>State = 0，成功。 = 1，失敗。</p>																								
<p>CmdBUS : <i>State =PutWord(mWord)</i></p> <hr/> <p>I2C : Out : MID+90+Checksum1+ Total_Len+Word_Len+ Word(n)+Word(n-1)+....+ Word(0)+ Checksum2+Dummy</p> <p>In : MID+State+Checksum3</p> <p>Total_Len = n+4 Word_Len = n+1 n = 0 ~19</p>		<p>一次暫存一個字 mWord到暫存語句中，一直累積，不需空格。</p> <p>Word 的大小限制 20 字之內。</p> <p>State = 0，成功。 = 1，失敗。</p>																								

<p>CmdBUS : ResetSentence()</p> <hr/> <p>I2C : MID+91+Checksum1+Dummy</p>	<p>清除暫存語句的指標。清除後，指標會指到暫存語句起頭的位置。</p>
<p>CmdBUS : State =GetWord(mWord)</p> <hr/> <p>I2C : Out : MID+92+Checksum1+Dummy In : MID+Total_Length+State+Word_Len+Word(n)+Word(n-1)+....+Word(0)+Checksum3</p> <p>Word_Len = n+1 n = 0 ~19</p>	<p>依序讀取暫存語句中的一個字，存放在 <i>mWord</i> 。</p> <p><i>State</i> = 0，成功。 = 1，失敗。</p>
<p>CmdBUS : State =GetSentence(Sentence)</p> <hr/> <p>I2C : Out : MID+93+Checksum1+Dummy In : MID+Total_Length+State+Sentence_Len+Sentence(n)+Sentence(n-1)+....+Sentence(0)+Checksum3</p> <p>Word_Len = n+1 n = 0 ~19</p>	<p>一次讀取整個暫存語句，存放在 <i>Sentence</i> 。</p> <p><i>State</i> = 0，成功。 = 1，失敗。</p>
<p>CmdBUS : State =SaveSentence(ID)</p> <hr/> <p>I2C : Out : MID+94+Checksum1+ID+Checksum2+Dummy In : MID+State+Checksum3</p>	<p>將暫存語句永久儲存到非揮發性記憶體中。</p> <p><i>ID</i> = 1 ~ 40 。</p> <p><i>State</i> = 0，成功。 = 1，失敗。</p>
<p>CmdBUS : State =LoadSentence(ID)</p>	<p>讀取非揮發性記憶體中的語句資料到暫存語句中。</p> <p><i>ID</i> = 1 ~ 40 。</p>

I2C : Out : MID+95+Checksum1+ID+ Checksum2+Dummy In : MID+State+Checksum3	State = 0，成功。 = 1，失敗。
CmdBUS : State =DeleteSentence(ID) <hr/> I2C : Out : MID+96+Checksum1+ID+ Checksum2+Dummy In : MID+State+Checksum3	清除非揮發性記憶體中的語句。 ID = 1 ~ 40。 State = 0，成功。 = 1，失敗。
CmdBUS : DeleteAllSentence() <hr/> I2C : MID+99+Checksum1+Dummy	清除非揮發性記憶體中的所有語句。 執行此命令後，需至少延遲 100ms 才可再執行 SR1 其他命令。
語音辨識相關指令	
CmdBUS : State =GoRecognition() <hr/> I2C : Out : MID+97+Checksum1+Dummy In : MID+State+Checksum3	啟動語音辨識。 State = 0，啟動成功。 = 1，啟動失敗。
CmdBUS : State =GetRecognition(ID) <hr/> I2C : Out : MID+98+Checksum1+Dummy In : MID+State+ID+Checksum3	取得語音辨識結果。 State = 0，辨識成功。 = 1，無法辨識。 辨識結果放在 ID (範圍 1~50)。

範例程式:

範例 1

```
Peripheral mySR1 As SR1 @ 31 '設定模組編號為 31

Dim Result, State As Byte
Dim MyString As String * 16

Sub MAIN()

    mySR1.DeleteAllSentence()
    Pause 100 ' 延遲 100 ms
    MyString = "da jia hao" ' 大家好
    mySR1.PutSentence(MyString)
    mySR1.SaveSentence(1)
    MyString = "zao an" ' 早安
    mySR1.PutSentence(MyString)
    mySR1.SaveSentence(2)
    MyString = "wu an" ' 午安
    mySR1.PutSentence(MyString)
    mySR1.SaveSentence(3)
    MyString = "wan an" ' 晚安
    mySR1.PutSentence(MyString)
    mySR1.SaveSentence(4)

    mySR1.GoRecognition()

    Do
        State = mySR1.GetRecognition(Result) '取得偵測結果
        IF State = 0 THEN
            Debug CSRXY(1, 2), Result, CLREOL
        END IF
        Pause(1)
    Loop
End Sub
```

範例 2

```
Peripheral mySR1 As SR1 @ 31 '設定模組編號為 31
Dim Result, State As Byte
Dim MyString As String * 16

Sub MAIN()

    mySR1.DeleteAllSentence()
    Pause 100 ' 延遲 100 ms
    mySR1.ResetSentence()
    MyString = "da" ' 大
    mySR1.PutWord(MyString)
    MyString = "jia" ' 家
    mySR1.PutWord(MyString)
    MyString = "hao" ' 好
    mySR1.PutWord(MyString)
    mySR1.SaveSentence(1)

    mySR1.GoRecognition()

    Do
        State = mySR1.GetRecognition(Result) '取得偵測結果
        IF State = 0 THEN
            Debug CSRXY(1, 2), Result, CLREOL
        END IF
        Pause(1)
    Loop
End Sub
```


範例 3

```

Peripheral mySR1 As SR1 @ 31 '設定模組編號為 31
Dim Result, State As Byte
Sub MAIN()



















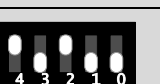













    mySR1.LoadDefaultSentence(1) ' 使用第一套內建語句
    mySR1.GoRecognition()

    Do
        State = mySR1.GetRecognition(Result) '取得偵測結果
        IF State = 0 THEN
            Debug CSRXY(1, 2), Result, CLREOL
        END IF
        Pause(1)
    Loop
End Sub

```

附錄

1. 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

2. 漢語拼音速查表

注音	漢語拼音	注音	漢語拼音
ㄅ	b	ㄆ	an
ㄆ	p	ㄇ	en
ㄇ	m	ㄏ	ang
ㄈ	f	ㄏ	eng
ㄉ	d	ㄉ	er
ㄊ	t	ㄟ	i, yi
ㄋ	n	ㄨ	u, wu
ㄌ	l	ㄩ	u, yu
ㄍ	g	ㄚ	ia, ya
ㄎ	k	ㄞ	ie, ye
ㄏ	h	ㄞ	iai, yai
ㄐ	j	ㄞ	iao, yao
ㄑ	q	ㄞ	iu, you
ㄒ	x	ㄞ	ian, yan
ㄗ	zh	ㄞ	in, yin
ㄘ	ch	ㄞ	iang, yang
ㄙ	sh	ㄞ	ing, ying
ㄨ	r	ㄞ	ua, wa
ㄨ	z	ㄞ	uo, wo
ㄨ	c	ㄞ	uai, wai
ㄨ	s	ㄞ	ui, wei
ㄨ	-i	ㄞ	uan, wan
ㄨ	a	ㄞ	un, wen
ㄨ	o	ㄞ	uang, wang
ㄨ	e	ㄞ	ong, weng
ㄨ	ê	ㄞ	ue, yue
ㄨ	ai	ㄞ	uan, yuan
ㄨ	ei	ㄞ	un, yun
ㄨ	ao	ㄞ	iong, yong
ㄨ	ou		