

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 Arminno 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 Arminno® 上對應的腳位，就可透過 Arminno® 執行操作。



需提供 VIN +6 ~ 12V



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Arminno[®]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Arminno[®]對應腳位相接，即可透過Arminno[®]操控SR1(連接時請注意腳位對應，將Vin對接Arminno[®]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

指令格式	指令功能																						
語句設定相關指令																							
LoadDefaultSentence(uint8_t Mode)	<p>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。</p> <p><i>Mode</i> = 0，不使用內建語句(預設)。 = 1，使用以下預設語句。</p> <table border="1" data-bbox="719 454 1118 1037"> <thead> <tr> <th><i>Mode</i> \ <i>ID</i></th> <th>1</th> </tr> </thead> <tbody> <tr><td>41</td><td>機器人前進</td></tr> <tr><td>42</td><td>機器人退後</td></tr> <tr><td>43</td><td>機器人停止</td></tr> <tr><td>44</td><td>機器人左轉</td></tr> <tr><td>45</td><td>機器人右轉</td></tr> <tr><td>46</td><td>機器人左移</td></tr> <tr><td>47</td><td>機器人右移</td></tr> <tr><td>48</td><td>機器人站起來</td></tr> <tr><td>49</td><td>機器人蹲下</td></tr> <tr><td>50</td><td>機器人攻擊</td></tr> </tbody> </table>	<i>Mode</i> \ <i>ID</i>	1	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> \ <i>ID</i>	1																						
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																						
uint8_t State = PutSentence(char* Sentence)	<p>一次暫存一整個語句到暫存語句中。</p> <p><i>Sentence</i> 的大小限制 20 字之內。 如整個語句超過 20 字，請使用 PutWord 一個字一個字輸入。</p> <p><i>State</i> = 0，成功。 = 1，失敗。</p>																						
uint8_t State = PutWord(char*mWord)	<p>一次暫存一個字到暫存語句中，一直累積，不需空格。</p> <p><i>mWord</i> 的大小限制 20 字之內。</p> <p><i>State</i> = 0，成功。 = 1，失敗。</p>																						
ResetSentence(void)	<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>																						
uint8_t State = GetWord(char *mWord)	<p>依序讀取暫存語句中的一個字，放在 <i>mWord</i> 中。</p> <p><i>State</i> = 0，成功。 = 1，失敗。</p>																						
uint8_t State = GetSentence(char* Sentence)	<p>一次讀取整個暫存語句。</p> <p><i>State</i> = 0，成功。</p>																						

	= 1，失敗。
uint8_t State = SaveSentence(uint8_t ID)	將暫存語句永久儲存到非揮發性記憶體中。 ID = 1 ~ 40。 State = 0，成功。 = 1，失敗。
uint8_t State = LoadSentence(uint8_t ID)	讀取非揮發性記憶體中的語句資料到暫存語句中。 ID = 1 ~ 40。 State = 0，成功。 = 1，失敗。
uint8_t State = DeleteSentence(uint8_t ID)	永久清除非揮發性記憶體中的語句。 ID = 1 ~ 40。 State = 0，成功。 = 1，失敗。
DeleteAllSentence(void)	永久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	
uint8_t State = GoRecognition(void)	啟動語音辨識。 State = 0，啟動成功。 = 1，啟動失敗。
uint8_t State = GetRecognition(uint8_t ID)	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

```
#include "arminno.h"
SR1 mySR1(31); //設定模組編號為 31

uint8_t Result, State;
char MyString[32];

int main(void)
{
    mySR1.DeleteAllSentence();
    Pause(1000); // 延遲 100 ms
    sprintf(MyString, "da jia hao"); // 大家好
    mySR1.PutSentence(MyString);
    mySR1.SaveSentence(1);
    sprintf(MyString, "zao an"); // 早安
    mySR1.PutSentence(MyString);
    mySR1.SaveSentence(2);
    sprintf(MyString, "wu an"); // 午安
    mySR1.PutSentence(MyString);
    mySR1.SaveSentence(3);
    sprintf(MyString, "wan an"); // 晚安
    mySR1.PutSentence(MyString);
    mySR1.SaveSentence(4);

    mySR1.GoRecognition();

    while(1) {
        State = mySR1.GetRecognition(Result); //取得偵測結果
        if(State == 0) {
            printf("%d\r\n", Result);
        }
        Pause(1000);
    }
}
```

範例 2

```
#include "arminno.h"
SR1 mySR1(31); //設定模組編號為 31
uint8_t Result, State;
char MyString[32];

int main(void)
{

    mySR1.DeleteAllSentence();
    Pause(1000); // 延遲 100 ms
    mySR1.ResetSentence();
    sprintf(MyString, "da"); // 大
    mySR1.PutWord(MyString);
    sprintf(MyString, "jia"); // 家
    mySR1.PutWord(MyString);
    sprintf(MyString, "hao"); // 好
    mySR1.PutWord(MyString);
    mySR1.SaveSentence(1);
    mySR1.GoRecognition();

    while(1) {
        State = mySR1.GetRecognition(Result); //取得偵測結果
        if(State == 0) {
            printf("%d\r\n", Result);
        }
        Pause(1000);
    }
}
```





















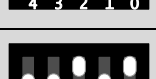
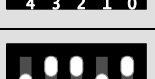
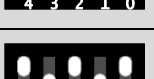

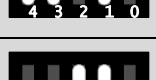
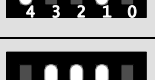
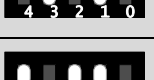
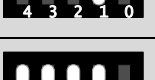
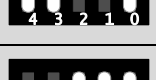
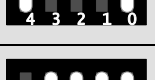
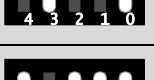
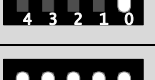
範例 3

```
#include "arminno.h"
SR1 mySR1(31); //設定模組編號為 31
uint8_t Result, State;
int main(void)
{
    mySR1.LoadDefaultSentence(1); //使用第一套內建語句
    mySR1.GoRecognition();

    while(1) {
        State = mySR1.GetRecognition(Result); //取得偵測結果
        if(State == 0) {
            printf("%d\r\n", Result);
        }
        Pause(1000);
    }
}
```

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