

CSR200 型馬達控制器

使用說明書

目錄

- 一、CSR200 型馬達控制器簡介
- 二、系統參數
 - 1. 系統參數說明
 - 2. 系統參數的讀取與更新
 - 3. 系統參數一覽表
 - 4. 各系統參數詳細說明
- 三、指令
 - 1. 指令說明
 - 2. 指令一欄表
 - 3. 各指令詳細說明
- 四、函數
 - 1. 函數說明
 - 2. 函數一覽表
- 五、程式編輯
 - 1. 變數說明
 - 2. 變數應用例
 - 3. 指標說明
 - 4. 程式用例
- 六、面板
 - 1. 鍵盤
 - 2. LCD
 - 3. 指撥開關
- 七、與 PC 連線
 - 1. 使用終端機程式
 - 2. 程式上傳與下載
- 八、擴充模組

一、CSR200 型馬達控制器簡介

CSR200 為一通用型多功能的馬達控制器，體積雖小，但功能強大。可以控制伺服馬達，也可以控制步進馬達，更兼具有一般 PLC 及人機介面的各項功能。對自動控制的各種應用需求，CSR200 提供了 All In One 的完全解決方案。

CSR200 的主要規格及特性如下：

1. 可以做開回路控制，也可以做閉迴路控制。
一般步進馬達，只能以開迴路的方式控制，所以在高速應用時，難免有失步的顧慮。但 CSR200 對步進馬達，卻可以提供全閉迴路的控制。普通的步進馬達，只要再加裝一簡單的編碼器，就可以用全閉迴路，類似控制伺服馬達的方法來控制，使步進馬達再也沒有所謂失步的問題。不但如此，高解析度的編碼器與微步驅動的步進馬達合用，更可以達成極精細的定位精度。再加上步進馬達原來就有自鎖扭矩大而且穩定的特性，整個系統構成一完美的夢幻組合。其整體性能，即使高價位的伺服馬達系統，也難以比擬。
2. 除了一般脈波輸出方式外，也可以提供 PWM 波輸出方式。
CSR200 對一般馬達驅動器，可以提供脈波輸出控制方式。對 DC 伺服馬達，更可以提供 PWM 波輸出，以最有效率的方式，直接控制 DC 伺服馬達。
3. 最多能同時控制 32 組馬達。
每一台 CSR200 可同時控制兩組馬達的運動，包括兩軸直線補間及圓弧補間。超過兩軸以上的系統，更可以串聯的方式運用。最多可串聯 16 台 CSR200，能同時控制 32 組馬達的動作。
4. 超強的 PLC 功能。
除馬達的控制外，CSR200 還附有 8 組輸出及 16 組輸入功能，可以涵蓋一般 PLC 的用途。而且所有的輸入輸出動作，均可以由馬達回授位置，直接觸發，有如“跑馬射箭”，效率及準確性，均非一般 PLC 所能相比。
5. 完整的人機介面功能。
CSR200 面板上的 LCD，可顯示 4 行中文或英文。28 鍵 KB，可輸入所有的數字，英文字母及常用符號。LCD 的顯示畫面及 KB 功能定義，均可由使用者的程式自行規劃。
6. 類比信號輸入界面。
CSR200 除一般數位輸出入界面外，尚有兩組類比信號輸入埠，可輸入 0-5V 電壓信號。也可配合 VR 式手搖桿，做調速或教導方式輸入應用。
7. 行進間調速。
CSR200 在馬達運轉進行中，仍可隨時調整速度，並立刻做加減速反應。而且這種臨時的速度改變，不會影響預先已設定好的定位距離。

8. 可以完全獨立單機操作。

CSR200 有獨立的電源，面板上有 LCD，也有功能完整的 KB，可以直接在面板上操作或編輯程式，完全不需要其他的電腦。

當然 CSR200 也可以透過 RS232 介面與 PC 或 PDA 連線，分享應用資源，使操作更方便，功能應用更有彈性。

9. 完整的指令集。

CSR200 有超過 100 種指令的指令集，並附有功能強大的函數功能，變數功能等。讓使用者可以用很簡單且人性化的語法，輕鬆發展自己的應用程式。

10. 程式、數據及系統參數，以分散化方式管理。

CSR200 對使用者的應用程式，大量或反覆使用的數據，以及控制系統本身的參數，均以分開的方式管理。使用者不但可以將編輯程式與修改參數兩種工作分開，以簡化程式。大量的數據，又可以被不同的程式反覆應用，提高系統記憶體的使用效率。同一組控制器，對應不同的工作切換，也更為簡單。

使用者對程式、數據或系統參數，可以分別操作，將其儲存在系統快閃記憶體，不受電源關閉影響。也可以透過 RS232 介面，上載至 PC 儲存，或隨時由 PC 下載應用。

二、系統參數

2-1. 系統參數說明

CSR200 為一通用型多功能的馬達控制器，可以做很多種不同的功能變化。一般而言，開機後 CSR200 各項功能的預設值及啓動程序，就是依據系統參數來決定。在打開電源後，CSR200 會自動讀取儲存在系統記憶體中的各項系統參數。然後即依這些參數，設定 CSR200 的各種工作模式。

CSR200 所有的系統參數均儲存在系統記憶體中，所以即使關閉電源也不會受到影響。所有的系統參數均可隨時讀取檢視，也可依使用者需要重新設定新值。

系統參數會直接影響 CSR200 的功能設定。不正確的系統參數，甚至會造成系統無法正常開機。在大多數的場合，使用者並無必要經常更改系統參數。若因系統參數錯誤，造成無法穩定開機的狀況時。可執行“DF”指令，或在開啓電源的同時，持續按著面板“重置”鍵，即可將所有系統參數改回安全預設值。再逐一更正各項設定，恢復系統功能。

2-2. 系統參數的讀取與更新

所有的系統參數均可隨時讀取檢視，也可隨時重新設定新值。

讀取系統參數，可執行“? 參數名稱”指令，CSR200 就會立即回應該參數現值。

使用例：

由終端機輸入“? XVM”。

CSR200 會立即將 XVM (X 軸工作速度) 的現值，以 ASCII 字串方式回應給終端機。

重新設定系統參數，可執行“參數名稱=新值”指令。

使用例：

由終端機輸入“XVM=200”。

CSR200 就會將 X 軸的工作速度，設定為 200。以後再收到“XM”等指令，CSR200 就會以 200 的速度去執行。

在程式模式，使用者除依上述方法，直接讀取或更改系統參數外。也可以藉由變數，來讀取或更改系統參數。

使用例：

SR #VEL；	宣告實數變數#VEL
#VEL=XVM；	讀取系統參數 XVM，將其儲存在變數#VEL 中
XVM=200；	將系統參數 XVM 改為 200
XMA 500；	以新設的速度 200，將 X 軸移動至座標 500 位置
XVM=#VEL；	將變數#VEL 的值存回系統參數 XVM，還原原來狀況

系統參數均可隨時重新設定新值。更改後的新值，一般而言，將立即生效，但並不會被自動儲存。所以關機後，下次開機仍是原設定值。要將更新後的系統參數儲存到系統記憶體，必須執行“SAVE C”指令。執行過“SAVE C”指令後，設定值即可被永久儲存。但原來儲存在系統記憶體的舊值，將被新值取代，即使關機後再重新開機也無法復原。

2-3. CSR200 系統參數一覽表

CSR200 系統參數一覽表

參數名稱	英文助憶	設定範圍	預設值	功能說明
XSV	X Servo mode	0~4	0	X 軸伺服模式
YSV	Y Servo mode	0~4	0	Y 軸伺服模式
XHM	X Home Mode	0~31	0	X 軸回原點模式
YHM	Y Home Mode	0~31	0	Y 軸回原點模式
XHN	X Home sensor No	0~31	0	X 軸原點開關輸入埠編號
YHN	Y Home sensor No	0~31	0	Y 軸原點開關輸入埠編號
XHC				
YHC				
XEDI	X Encoder Direct	0~1	0	X 軸編碼器方向
YEDI	Y Encider Direct	0~1	0	Y 軸編碼器方向
XSC2	SCale	0~32767.9999	2000	每單位距離相對的編碼器步數分頻的電子減速比之分母值。(馬達解析度)
XSC1	SCale	0~32767.9999	10	每單位距離相對的步數分頻的電子減速比之分子值。(螺桿每轉前進 mm)
XESC	X Encoder Scale	1~65535	1	X 軸編碼器回授一脈波所相對應的輸出脈波數。 ES=馬達解析度÷編碼器解析度
YESC	Y Encoder Scale	1~65535	1	Y 軸編碼器回授一脈波所相對應的輸出脈波數。
XTM				閉回路控制用參數，其實際物理意義及用法，請參考系統參數詳細說明。
YTM				同上
XTV		1~65535	16	同上
YTV		1~65535	16	同上
XTA		1~256	2	同上
YTA		1~256	2	同上
XTB		1~256	2	同上
YTB		1~256	2	同上

XCLP XCL+	X Clip Positive	-32767.9999~ 32767.9999	32767	X 軸正極限
XCLM XCL-	X Clip Negative	-32767.9999~ 32767.9999	-32767	X 軸負極限
YCLM YCL+	Y Clip Right	-32767.9999~ 32767.9999	32767	Y 軸左極限
YCLM YCL-	Y Clip Left	-32767.9999~ 32767.9999	-32767	Y 軸右極限
XEL	X Error Limit	1~65535	65535	
YEL	Y Error Limit	1~65535	65535	
XHC	X Home Coordinate	-32767.9999~ 32767.9999	0	X 軸原點座標
YHC	Y Home Coordinate	-32767.9999~ 32767.9999	0	Y 軸原點座標
XVF	X Velocity Fast	0~32767.9999	100	X 軸高速
YVF	Y Velocity Fast	0~32767.9999	100	Y 軸高速
XVM	X Velocity Move	0~32767.9999	50	X 軸工作速度
YVM	Y Velocity Move	0~32767.9999	50	Y 軸工作速度
XVB	X Velocity Base	0~32767.9999	1	X 軸起動速度
YVB	Y Velocity Base	0~32767.9999	1	Y 軸起動速度
XVA	X Velocity Accelerate	0~32767.9999	1000	X 軸加速度
YVA	Y Velocity Accelerate	0~32767.9999	1000	Y 軸加速度
XVH	X Velocity Home	0~32767.9999	1000	X 軸回原點速度
YVH	Y Velocity Home	0~32767.9999	1000	Y 軸回原點速度
XVJ	X Velocity Jog	0~32767.9999	1000	X 軸 JOG 速度
YVJ	Y Velocity Jog	0~32767.9999	1000	Y 軸 JOG 速度
XKP	X Proportional Gain	0~65535	1	X 軸比例控制器增益
YKP	Y Proportional Gain	0~65535	1	Y 軸比例控制器增益
XKD	X Derivative Gain	0~65535	1	X 軸微分控制器增益
YKD	Y Derivative Gain	0~65535	1	Y 軸微分控制器增益
XKI	X Integral Gain	0~65535	0	X 軸積分控制器增益
YKI	Y Integral Gain	0~65535	0	Y 軸積分控制器增益
XPM	Pulse mod	0~1	0	1 pulse 2pulse 切換

2-4. 各系統參數詳細說明

參數名稱： XSV

YSV

英文助憶： X Servo mode

Y Servo mode

功能： 伺服模式

設定範圍： 0-4

預設值： 0

功能說明： XSV 決定 X 軸的伺服控制模式，YSV 決定 Y 軸的伺服控制模式，有下列各種設定選項；

0：開迴路控制

1：半閉迴路控制。運動時為開迴路控制，但在終點自動做到位檢查，並自動修正。

2：全閉迴路控制。

3：PWM 輸出閉迴路控制。

此種控制方式，最適合用於 DC 伺服馬達，配合可接受 PWM 信號輸入的馬達驅動器（如擎翔的 CSV201），可直接對伺服馬達做最有效率的控制。

PWM 輸出只要再經過簡單的積分或濾波處理，也可轉為連續的類比信號。（請參考附錄應用電路）

4：手搖輪模式。

參數名稱： XHM
YHM

英文助憶： X Home Mode
Y Home Mode

功能： 回原點模式

設定範圍： 0-31

預設值： 0

功能說明： XHM 決定 X 軸回原點的模式，YHM 決定 Y 軸回原點的模式。
其功能依二進位各位元決定。

位元 0 0：回原點時，先往座標負方向尋找原點。
1：回原點時，先往座標正方向尋找原點。

位元 1 0：進原點時，使用原點開關信號。
1：進原點時，使用失步信號。

位元 2 0：出原點時，使用原點開關信號。
1：出原點時，使用編碼器 Index 信號。

位元 3 0：原點開關為常閉極性。
1：原點開關為常開極性。

位元 4 0：回原點後更新座標。
1：回原點後不更新座標。

參數名稱： XSC

YSC

英文助憶： X Scale

Y Scale

參數意義： 距離單位比例，即 CSR200 內部脈波單位與距離單位的比值。

SC 在實際應用時，也可以將其視為移動一單位距離所需輸出脈波數。

XSC 為 X 軸的距離單位比例，YSC 為 Y 軸的距離單位比例。兩軸可以分開做不同的設定。

設定範圍： 0-32767.9999

預設值： 0

功能說明： CSR200 內部的計算單位〈脈波〉，與使用者所慣用的距離單位〈如 mm 或 inch 等〉未必一致。但實際應用時，使用者並不需要特別去注意 CSR200 內部的計算單位。使用者只要先以 SC 參數來訂定自己所習慣使用的單位與 CSR200 內部的脈波單位的比例關係。以後所有與距離有關的參數，如座標或速度等，就可以使用自己所習慣使用的單位了。

SC 的訂定方法，可以公式表示如下，

$SC = \text{馬達解析度} \div \text{馬達每轉進給距離}$ 。其中

馬達解析度：對步進馬達而言，為驅動馬達旋轉一轉所需的脈波數。

對以 PWM 模式控制的伺服馬達而言，則為馬達上所附的編碼器，每轉所回授的步數。

馬達每轉進給距離：馬達旋轉一轉所走的距離，若為螺桿驅動的系統，此即為螺桿的節距(Pitch)。若為皮帶傳動的系統，則為驅動輪每轉的齒數乘以齒距。

使用例：

- 1、步進馬達系統，使用微步驅動，每轉分 10,000 步。使用皮帶驅動。驅動齒輪為 20 齒，皮帶齒距為 2mm。則

$$SC = 10000 / (20 \times 2) = 250。$$

以上計算結果，其實際意義就是在這個系統中，每 mm 距離相當於 250 個脈波。

- 2、DC 伺服馬達，編碼器解析度為每轉 2000 步。驅動使用螺距為 10mm 的滾珠螺桿。則

$$SC = 2000 / 10 = 200。$$

參數名稱： XESC
YESC

英文助憶： X Encoder Scale
Y Encoder Scale

參數意義： 編碼器脈波比，即編碼器回授一步所相對應的輸出脈波數。
XESC 為 X 軸的編碼器脈波比，YESC 為 Y 軸的編碼器脈波比，可以分開做不同的設定。

設定範圍： 0-65536

預設值： 0

功能說明： 此項參數只有在以閉迴路模式控制步進馬達時才有用〈SV=2〉。
若以公式表示，

$ESC = \text{馬達解析度} \div \text{編碼器解析度}$ 。其中

馬達解析度：為驅動馬達旋轉一轉所需的脈波數，

編碼器解析度：編碼器一轉所回授的步數。

此項參數只能設為整數，不接受小數。也就是說，步進馬達微步驅動的解析度，必須是編碼器解析度的整數倍。

使用例：

步進馬達系統，使用微步驅動，每轉分 10,000 步。編碼器每轉 500cycle，解析為 2000 步。則

$SC = 10000 / 2000 = 5$

- 參數名稱： XVF，XVM
YVF，YVM
- 英文助憶： X Velocity Fast，X Velocity Move
Y Velocity Fast，Y Velocity Move
- 參數意義： 設定速度
XVF 決定 X 軸的高速，
XVM 決定 X 軸的工作速度，
YVF 決定 Y 軸的高速，
YVM 決定 Y 軸的工作速度。
CSR200 X 軸及 Y 軸可以分別設定不同的速度參數。但在兩軸同動的場合，
則以 X 軸的速度參數為共同速度參數。
- 設定範圍： 0-32767.9999
- 預設值： 0
- 功能說明： 所有的速度參數，其單位為距離除以時間。若 SC 參數已訂定，使用者應可
將其設定為自己所慣用的單位，如 mm 或 inch 等〈請參照 SC 系統參數功能
說明〉。時間單位在 CSR200 中則固定為秒。所以速度參數的單位可以被設
定為 mm/sec 或 inch/sec 等。

CSR200 將速度參數分為高速及工作速度兩種模式，可藉由不同的運動指令選用不同的速度。一般而言，高速模式用於單純的位置移動，只要求速度，不需要準確度的場合。而工作速度模式則被用在需要準確度的場合。

以銑床應用為例，刀具位置的移動可使用高速，其值由 VF 參數設定。而在切削時，則應使用工作速度模式，其值由 VM 參數設定。

使用例：

步進馬達系統，使用微步驅動，每轉分 10,000 步。使用皮帶驅動。驅動齒輪為 20 齒，皮帶齒距為 2mm。則 SC 應設定為

$$SC=10000/(20 \times 2)=250。$$

若設定 XVF=500，XVM=200，
YVF=400，YVM=100

其意義為 X 軸的高速為 500mm/sec，工作速度為 200mm/sec
Y 軸的高速為 400mm/sec，工作速度為 100mm/sec

此時若執行“XMA 1200”指令，

X 軸會以 200mm/sec 的速度，移動至座標 1200mm 的位置。

參數名稱： XVB，YVB
英文助憶： X Velocity Base，X Velocity Base
參數意義： 設定起始速度
XVB 決定 X 軸的起始速度，
YVB 決定 Y 軸的起始速度。
CSR200 X 軸及 Y 軸可以分別設定不同的起始速度參數。但在兩軸同動的情況，則以 X 軸的起始速度參數為共同起始速度。
設定範圍： 0-32767.9999
預設值： 0
功能說明：

有關本參數應用單位的說明，請參照 VF,VM 系統參數的功能說明。

起始速度即運動開始時的初速度，也等於運動終了時的終速度。

CSR200 將所有的運動指令的執行方式固定為如下述模式：

由 VB 參數所設定的起始速度開始移動，然後由 VA 參數所設定的加速度逐漸加速。待加速至 VF 或 VM 參數所設定的最高速時，即改為等速運動。待快到終點時，再以 VA 參數所設定的加速度開始減速，直到減至由 VB 參數所設定的起始速度，即立即停止。

起始速度的設定，對系統整體運動效率的影響很大，尤其是當短距離的運動比較多時。起始速度設的太低，運動會顯得比較遲緩，效率降低。相對的，起始速度設的太高，則會在運動起點和終點時，引發過大的震動。

使用例：

參數名稱： XVA，YVA
英文助憶： X Velocity Accelerate，X Velocity Accelerate
參數意義： 設定加速度

XVA 決定 X 軸的加速度，
YVA 決定 Y 軸的加速度。

CSR200 X 軸及 Y 軸可以分別設定不同的加速度參數。但在兩軸同動的場合，則以 X 軸的加速度參數為共同加速度。

設定範圍： 0-32767.9999

預設值： 0

功能說明：

加速度的單位應為距離除以時間平方。若 SC 參數已訂定，使用者應可將距離單位設定為自己所慣用的單位，如 mm 或 inch 等〈請參照 SC 系統參數功能說明〉。時間單位在 CSR200 中則固定為秒。所以加速度參數的單位可以被設定為 mm/sec²或 inch/sec²等。

CSR200 將所有的運動指令的執行方式固定為如下述模式：

由 VB 參數所設定的起始速度開始移動，然後由 VA 參數所設定的加速度逐漸加速。待加速至 VF 或 VM 參數所設定的最高速時，即改為等速運動。待快到終點時，再以 VA 參數所設定的加速度開始減速，直到減至由 VB 參數所設定的起始速度，即立即停止。

加速度的設定，對系統整體運動效率的影響很大，尤其是當短距離的運動比較多，或起始速度設定比較低時。加速度設的太低，運動會顯得比較平緩，但效率也比較低。相對的，加速度設高一點，則運動會顯得比較敏捷，但也比較不穩定。加速度設的太高，在閉迴路系統，會引發很大的震動。在開迴路系統，更會造成失步問題。

使用例：

參數名稱： XTV，YTV
英文助憶： X Timer Velocity，Y Timer Velocity
參數意義： 設定閉迴路步進馬達控制時，對誤差的修正速度
XTV 決定 X 軸對誤差的修正速度，
YTV 決定 Y 軸對誤差的修正速度。
設定範圍： 0-65536
預設值： 1280
功能說明：

本參數只有應用在對步進馬達做閉迴路的控制模式，開迴路控制模式本參數並未使用到。

步進馬達在未失步時，本參數實際並沒有發生作用。而一旦發生失步，也就是有誤差發生時，閉迴路控制系統就要去修正這個誤差。本參數就是用於決定修正誤差的速度。設定值小，就會以比較慢的速度修正誤差。慢慢修正的好處是比較穩定。但缺點是，若修正速度比系統原來的運動速度還慢的話，一旦發生失步，就永遠追不上了。相對的，本參數若設的較大，有誤差發生時，會以比較快的速度去修正誤差。但這樣也會使系統比較不穩定，有時甚至還會造成誤差的反覆追逐現象。

使用者若要調整此參數，可由小到大，慢慢增加，以實際實驗結果為依據，決定最佳值。

參數名稱： XTA，YTA

英文助憶： X Timer Accelerate，Y Timer Accelerate

參數意義： 設定閉迴路步進馬達控制時，對誤差的反應速度

XTA 決定 X 軸對誤差的反應速度，

YTA 決定 Y 軸對誤差的反應速度。

CSR200 X 軸及 Y 軸可以分別設定不同的對誤差的反應速度。

設定範圍： 0-256

預設值： 1

功能說明：

本參數只有應用在對步進馬達做閉迴路的控制模式，開迴路控制模式本參數並未使用到。

本參數決定對誤差的反應速度，設定值愈大，對誤差反應愈敏銳，但相對的，系統也愈不穩定，甚至會造成發散性震動。

本參數實用值，約為 1 至 5 之間，不建議使用過大的設定值。使用者若要調整此參數，可由小到大，慢慢增加，以實際實驗結果為依據，決定最佳值。

使用例：

參數名稱： XTB，YTB

英文助憶： X Timer Base speed，Y Timer Base speed

參數意義： 設定閉迴路步進馬達控制時，對誤差的反應啓動速度

XTB 決定 X 軸對誤差的反應速度，

YTB 決定 Y 軸對誤差的反應速度。

CSR200 X 軸及 Y 軸可以分別設定不同的對誤差的反應啓動速度。

設定範圍： 0-256

預設值： 1

功能說明：

本參數只有應用在對步進馬達做閉迴路的控制模式，開迴路控制模式本參數並未使用到。

本參數決定對誤差的反應速度，設定值愈大，對誤差反應愈敏銳，但相對的，系統也愈不穩定，甚至會造成反復性震動。

本參數實用值，約為 1 至 5 之間，不建議使用過大的設定值。使用者若要調整此參數，可由小到大，慢慢增加，以實際實驗結果為依據，決定最佳值。

使用例：

參數名稱： XCLP，XCL+
XCLM，XCL-
YCLP，YCL+
YCLM，YCL-
英文助憶： X Clip Plus，X Clip Minus
Y Clip Plus，Y Clip Minus
參數意義： 設定運動位置極限
XCLP(或 XCL+)，決定 X 軸運動往座標正方向移動的位置極限，
XCLM(或 XCL-)，決定 X 軸運動往座標負方向移動的位置極限，
YCLP(或 YCL+)，決定 Y 軸運動往座標正方向移動的位置極限，
YCLM(或 YCL-)，決定 Y 軸運動往座標負方向移動的位置極限。
設定範圍： -65535.9999-65535.9999
預設值： CLP=65535
CLM=-65535

功能說明：

本參數提供軟體極限功能，可以做為系統的一道安全保障。所有的運動指令會自動被限制在極限範圍內動作。設定好的各軸的極限位置，是存於系統參數，和一般應用程式可以分開管理。以後即使發生程式操作錯誤，要求超過極限範圍的運動，會被自動限制。

本參數所用的單位，就是座標系統的距離單位，是由 SC 參數訂定。使用者應可將距離單位設定為自己所慣用的單位，如 mm 或 inch 等〈請參照 SC 系統參數功能說明〉。參數值可以為負數，也可以接受小數。

負極限是指往負座標方向的極限，正極限是指往正座標方向的極限，兩者數值都不限正負，但實際設定時，負極限應小於正極限。

使用例：

參數名稱： XHC，YHC
英文助憶： X Home Coordinate
Y Home Coordinate
參數意義： 設定原點座標
XHC 為 X 軸的原點座標，
YHC 為 Y 軸的原點座標。
設定範圍： -65535.9999-65535.9999
預設值： 0
功能說明：

本參數是配合原點感測器〈Origin position sensor〉及回原點指令使用。當 CSR200 收到回原點指令後，即會依 HM 系統參數所指定的方向去尋找原點，也就是一方面移動位置，一方面隨時檢查原點感測器的輸入，待收到原點訊號時，在那一瞬間的位置，即是座標系統原點位置的參考點。但參考點並不一定就是原點，也就是說收到原點訊號時的位置，其座標並不一定要設為 0，而可以由本參數任意設定。

本參數所用的單位，就是座標系統的距離單位，是由 SC 參數訂定。使用者應可將距離單位設定為自己所慣用的單位，如 mm 或 inch 等〈請參照 SC 系統參數功能說明〉。參數值可以為負數，也可以接受小數。

本參數可配合 HM 系統參數一起使用，開機後可以設定 CSR200 自動去尋找原點，並自動設定原點座標，再由 SC 系統參數設定距離單位，再由 CL 及 CR 系統參數設定極限範圍，整個座標系統就可自動建立。而這些動作可全部由系統參數決定，開機即自動完成，與應用程式完全獨立。

使用例：

參數名稱： XKP，YKP
英文助憶： X Proportional Gain
Y Proportional Gain
參數意義： 設定伺服馬達比例控制器增益。
XPG 為 X 軸的比例控制器增益，
YPG 為 Y 軸的比例控制器增益。
設定範圍： 0-65536
預設值： 0
功能說明：

本參數只有在以 PWM 輸出，控制 DC 伺服馬達的模式使用。其他控制模式並未用到。

CSR200 控制 DC 伺服馬達，是使用全閉迴路控制。所運用的控制法則〈Control algorithm〉包括三種數學運算，分別是比例控制器、微分控制器及積分控制器。本參數即是比例控制器所用的增益。

所謂比例控制器，就是輸出與誤差成正比例的控制器，而所謂增益就是它的比例值。以公式表示，其關係如下

$$PWM = ERR \times PG \quad \text{其中，}$$

PWM = 比例控制器的輸出。
ERR = 位置誤差

由上式即可看出，對同樣的誤差而言，比例控制器的輸出，與 PG 成正比，PG 愈大，輸出就愈大，控制器對誤差就愈敏感。實務上，將 PG 調大，確實可以減小誤差，但控制也比較不穩定。

使用例：

參數名稱： XKI，YKI

英文助憶： X Integral Gain

Y Integral Gain

參數意義： 設定伺服馬達積分控制器增益。
XPG 為 X 軸的積分控制器增益，
YPG 為 Y 軸的積分控制器增益。

設定範圍： 0-65536

預設值： 0

功能說明：

本參數只有在以 PWM 輸出，控制 DC 伺服馬達的模式使用。其他控制模式並未用到。

CSR200 控制 DC 伺服馬達，是使用全閉迴路控制。所運用的控制法則〈Control algorithm〉包括三種數學運算，分別是比例控制器、微分控制器及積分控制器。本參數即是積分控制器所用的增益。

所謂積分控制器，就是輸出與誤差的時間積分成正比例的控制器，而 IG 就是兩者的比值。以公式表示，其關係如下

$$PWM = \int ERR dt \times IG \quad \text{其中，}$$

PWM = 積分控制器的輸出。

ERR = 位置誤差

積分控制器，也可視為輸出的變化率正比於誤差，以公式表示，其關係如下

$$\frac{dPWM}{dt} = IG \times ERR$$

當誤差變大時，控制器輸出快速變化，以修正誤差，當誤差變小時，控制器的輸出變化會變慢。只要有誤差，控制器的輸出將持續變化。

積分控制器主要的功能是消除系統的穩態誤差，但很容易由於積分，控制變得比較不穩定。

使用例：

參數名稱： XKD，YKD

英文助憶： X Derivative Gain

Y Derivative Gain

參數意義： 設定伺服馬達微分控制器增益。
XKD 為 X 軸的微分控制器增益，
YKD 為 Y 軸的微分控制器增益。

設定範圍： 0-65536

預設值： 0

功能說明：

本參數只有在以 PWM 輸出，控制 DC 伺服馬達的模式使用。其他控制模式並未用到。

CSR200 控制 DC 伺服馬達，是使用全閉迴路控制。所運用的控制法則〈Control algorithm〉包括三種數學運算，分別是比例控制器、微分控制器及積分控制器。本參數即是微分控制器所用的增益。

所謂微分控制器，就是輸出與誤差的變動率成正比例的控制器，而 DG 就是兩者的比值。以公式表示，其關係如下

$$PWM = \frac{dErr}{dt} \times DG \quad \text{其中}$$

PWM = 微分控制器的輸出
ERR = 位置誤差
KD = 微分控制器增益

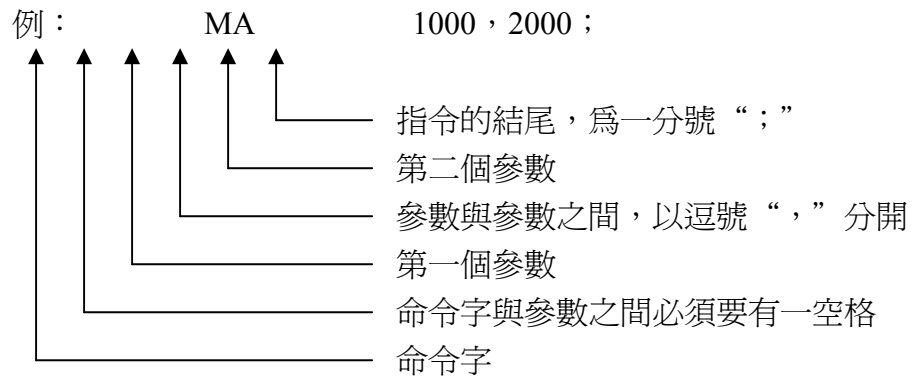
當誤差變動率變大時，控制器輸出也就愈大。

使用例：

三、指令

3.1 指令語法說明

CSR200 的指令，是由命令字與參數兩部分所構成，命令字均為大寫英文字母，基本上是該指令功能的英文簡寫。參數則多為數字或代表數字的變數符號等。命令字與參數之間必須要有一空格分開。指令的結尾，則為一分號“；”。



3.2CSR200 指令一覽表

CSR200 指令一覽表

單軸運動指令

XMA x	(X Move Absolutely)	以工作速度移動 X 軸至絕對座標 x 處。
XMR x	(X Move Relatively)	以工作速度移動 X 軸至相對座標 x 處。
XA x	(X move Absolutely)	依系統現在的速度模式，移動 X 軸至絕對座標 x 處。
XR x	(X move Relatively)	依系統現在的速度模式，移動 X 軸至相對座標 x 處。
X x	(X move)	依系統現在的速度模式及座標模式，移動 X 軸至座標 x 處。
YMA y	(Y Move Absolutely)	以工作速度移動 Y 軸至絕對座標 y 處。
YMR y	(Y Move Relatively)	以工作速度移動 Y 軸至相對座標 y 處。
YA y	(Y move Absolutely)	依系統現在的速度模式，移動 Y 軸至絕對座標 y 處。
YR y	(Y move Relatively)	依系統現在的速度模式，移動 Y 軸至相對座標 y 處。
Y y	(Y move)	依系統現在的速度模式及座標模式，移動 Y 軸至座標 y 處。

兩軸直線運動指令

MA x, y	(Move Absolutely)	以工作速度，同時移動 X 軸及 Y 軸，至絕對座標 x,y 處。
MR x, y	(Move Relatively)	以工作速度，同時移動 X 軸及 Y 軸，至相對座標 x,y 處。
M x, y	(Move)	依系統現在的座標模式，以工作速度，同時移動 X 軸及 Y 軸，至座標 x,y 處。

兩軸圓弧指令

AAL x, y, r	(Arc Absolute Left)	以絕對座標 x,y 為終點，r 為半徑，往左轉方向做圓弧移動。
AAR x, y, r	(Arc Absolute Right)	以絕對座標 x,y 為終點，r 為半徑，往右轉方向

ARL x, y, r (Arc Relative Left)	做圓弧移動。 以相對距離 x, y 為終點， r 為半徑，往左轉方向做圓弧移動。
ARR x, y, r (Arc Relative Right)	以相對距離 x, y 為終點， r 為半徑，往右轉方向做圓弧移動。

JOG 運動指令

XJG a (X Jog by Analog input)	X 軸依手搖桿類比輸入移動。 $(1 \leq a \leq 2)$
XJG + (X Jog +)	X 軸往正座標方向移動。
XJG - (Y Jog -)	X 軸往負座標方向移動。
XJG 0 (X Jog stop)	X 軸減速停止。
YJG a (Y Jog by Analog input)	Y 軸依手搖桿類比輸入移動。 $(1 \leq a \leq 2)$
YJG + (Y Jog +)	Y 軸往正座標方向移動。
YJG - (Y Jog -)	Y 軸往負座標方向移動。
YJG 0 (Y Jog stop)	Y 軸減速停止。

回原點指令

XH m, n (X Home)	X 軸回原點。 m 為回原點模式， n 為做為原點的輸入埠 $(1 \leq n \leq 16)$ 。
YH m, n (Y Home)	Y 軸回原點。 m 為回原點模式， n 為做為原點的輸入埠 $(1 \leq n \leq 16)$ 。

座標設定指令

CS x, y (Coordinate Set)	設定現在位置座標。
XCS x (X Coordinate Set)	設定 X 軸現在位置座標。
YCS y (Y Coordinate Set)	設定 Y 軸現在位置座標。

速度控制指令

VF r	(Velocity Fast)	設定高速。
VM r	(Velocity Move)	設定工作速度。
VB r	(Velocity Base)	設定起始速度。
VA r	(Velocity Accelerate)	設定加速度。
VH r	(Velocity Home)	設定回原點移動的速度。
VJ r	(Velocity Jog)	設定 JOG 移動的速度。
AVM a	(Analog VM)	依類比輸入埠的值，調變工作速度。 $(1 \leq a \leq 2)$
XAVM a		依類比輸入埠的值，調變 X 軸的工作速度。
YAVM a		依類比輸入埠的值，調變 Y 軸的工作速度。

輸出介面控制指令

SET Pn	(Set Port)	設定第 n 輸出埠為 On。 $(1 \leq n \leq 8)$
CLR Pn	(Clear Port)	清除第 n 輸出埠為 Off。
CHG Pn	(Change Port)	變更第 n 輸出埠的狀態。原來為 Off 改為 On，原來為 On 改為 Off。
PLS Pn, tm	(Pulse Port)	由第 n 輸出埠輸出一脈波，tm 為脈波寬度，單位為 4msec。
OUTP n	(Output Port)	同時設定全部 8 個輸出埠的狀態。 $(0 \leq n \leq 255)$

運動同步輸出控制指令

XSET Pn, x	(Set Port by X)	在 X 軸移動到座標 x 時，設定第 n 輸出埠為 On。 $(1 \leq n \leq 8)$
XCLR Pn, x	(Clear Port by X)	在 X 軸移動到座標 x 時，清除第 n 輸出埠為 Off。 $(1 \leq n \leq 8)$
XCHG Pn, x1, x2...x16	(Change Port by X)	在 X 軸移動到座標 x 時，變更第 n 輸出埠的狀態。原來為 Off 改為 On，原來為 On 改為 Off。本指令可一次設定 16 點 x 座標，配合運動控制指令，依到達位置，連續變更輸出埠的狀態。
XPLS Pn, tm, x1, x2, ...x16	(Pulse Port by X)	在 X 軸移動到座標 x 時，由第 n 輸出埠輸出一脈波，tm 為脈波寬度，單位為 4msec。本指令可一次設定 16 點 x 座標，配合運動控制指令，依到達位置，連續輸出脈波。
YSET Pn, x	(Set Port by Y)	
YCLR Pn, x	(Clear Port by Y)	

YCHG Pn ,x1 ,x2...x16 (Change Port by Y)
 YPLS Pn ,tm ,x1 ,x2 ,...x16 (Pulse Port by Y)

程式流程控制指令

G addr	(Go)	由指定位址開始執行程式。若未指定位址，即由位址 0 開始執行。 addr 參數可為絕對位址，址標或行號。
JP addr	(Jump)	程式無條件跳躍至指定位址。
J1 i ,addr	(Jump at Input)	當指定的輸入埠為 On 時，程式跳躍至指定位址。 $(1 \leq i \leq 18)$
JN1 i ,addr	(Jump at No Input)	當指定的輸入埠為 Off 時，程式跳躍至指定位址。 $(1 \leq i \leq 18)$
JZ #int ,addr	(Jump at Zero)	當指定的變數為 0 時，程式跳躍至指定位址。
JNZ #int ,addr	(Jump at Not Zero)	當指定的變數不為 0 時，程式跳躍至指定位址。
JE c ,#var ,addr	(Jump at Equal)	當變數#var 等於 c 時，程式跳躍至指定位址。其中 c 可為常數，變數或函數，但必須與#var 同類型。
JNE c ,#var ,addr	(Jump at Not Equal)	當變數#var 不等於 c 時，程式跳躍至指定位址。其中 c 可為常數，變數或函數，但必須與#var 同類型。
JG #var ,c ,addr	(Jump at Great than)	當變數#varc 大於 c 時，程式跳躍至指定位址。其中 c 可為常數，變數或函數，但必須與#var 同類型。
JNG #var ,c , addr	(Jump at Not Great than)	當變數#varc 不大於 c 時，程式跳躍至指定位址。其中 c 可為常數，變數或函數，但必須與#var 同類型。
JTI i ,tm ,addr	(Jump depends on Timer and Input)	當輸入埠 i 為 On，或在參數 tm 設定的時間內變為 On，程式跳躍至指定位址。
JNTI i ,tm ,addr	(Jump depends on Timer and No Input)	當輸入埠 i 為 Off，或在參數 tm 設定的時間內變為 Off，程式跳躍至指定位址。
CALL addr	(Call)	呼叫副程式。
RET	(Return)	由副程式返回。若原來已在最上層程式，則結束程式執行。
WI i	(Wait Input)	若輸入埠 i 為 Off，則程式暫停。等待輸入埠 I 轉為 On 後，程式再繼續執行。

WNI i	(Wait No Input)	若輸入埠 i 為 On，則程式暫停。等待輸入埠 I 轉為 Off 後，程式再繼續執行。
WT tm	(Wait)	暫停一段時間再繼續。參數 tm 為暫停的時間，單位為 msec。
DN	(Done)	確定先前所下指令已全部執行完畢再繼續。
SG	(Start Go)	程式執行時，可由此指令作單步執行動作。

馬達控制指令

HON	(Hold On)	啟動馬達控制。
HOFF	(Hold Off)	暫停馬達控制。
SV n	(Servo Mode)	設定伺服控制模式。 n=0 開迴路控制， n=1 半閉迴路控制， n=2 全閉迴路控制， n=3 PWM 輸出閉迴路控制 (DC 伺服馬達用)。

面板控制指令

CLR D	(Clear Disp)	清除 LCD 顯示，並將指標(Cursor)移回左上角原點。
POS n	(Position)	設定 LCD 指標(Cursor)位置，後續 DISP 指令即以此位置開始印出。(0 ≤ n ≤ 63)
DISP #var	(Display a variable)	由 LCD 印出一變數。
DISP "..."	(Display a string)	由 LCD 印出一字串。
SET C	(Set Cursor)	顯示 LCD 指標(Cursor)
CLR C	(Clear Cursor)	不顯示 LCD 指標(Cursor)，但 POS 功能仍在。
INK In	(Input an Integer from KB)	由 KB 輸入一 n 位數的整數。
INK Rn	(Input a Real from KB)	由 KB 輸入一 n 位數(含小數點)的小數。
JFn addr	(Jump at key pressed)	檢查面板 Fn 鍵，當被按下時，程式跳躍至指定位址。各鍵名稱 Fn 如下
	n=1-4	F1-F4
	n=P	前目錄(Previous menu)
	n=M	主目錄(Main menu)
	n=G	啟動 (Go)
	n=S	停止 (Stop)
	n=T	重置 (reseT)

$n=U,D,L,R$ 上(Up),下(Down),左(Left),右(R)

On Fn, addr	設定當 Fn 鍵被按下時，執行位於 addr 處的副程式。
Off Fn, addr	設定當 Fn 鍵 Off 時，執行位於 addr 處的副程式。
CLROn Fn	解除 Fn 鍵 On 副程式設定。
CLROff Fn	解除 Fn 鍵 Off 副程式設定。

程式管理指令

PG (Program Generate)	進入程式編輯模式
PA (Program Append)在	原有程式最後再繼續附加新程式
PL (Program List)	列示程式
PE n (Program Edit)	修改程式， n 為要修改的程式行號。
PI n (Program Insert)	插入一行程式， n 為要插入的程式行號。
PD n (Program Delete)	刪除一行程式， n 為要刪除的程式行號。
UL (Up Load)	上傳程式至終端機(PC)。
DL (Down Load)	由終端機(PC)下傳程式。
SAVE P (Save Program)	儲存程式至 Flash Rom

資料庫管理指令

DATA idx, r	儲存資料 r 至 idx 指定的資料庫位址 ($0 \leq idx \leq 2048$)
SAVE D (Save Data)	儲存全部資料庫資料至 Flash Rom

變數指令

CLR VAR	清除變數。
SI #var	宣告整數變數。
SR #var	宣告實數變數。
#var=data	設定變數， data 可以下列方式設定 常數，變數，系統參數，資料庫資料，函數或 以上數據來源的數學組合

系統參數管理指令

參數名稱=data	設定系統參數
SAVE C (Save Config)	儲存全部系統參數至 Flash Rom

數學函數

SIN(r)	正弦函數
COS(r)	餘弦函數
TAN(r)	正切函數
ATN(r)	反正切函數
SQR(r)	開平方

讀取指令

?XPC	(Pulse)	讀取目前 PULSE 位置
?XPE	(Encoder)	讀取編碼器位置
?XSV	(Mode)	讀取目前使用控制模式 (0~4)
?XVM	(Speed)	讀取目前工作速度
?XVA	(VA)	讀取目前加速度
?XVB	(VB)	讀取目前啓動速度
?XEL	(Encoder err)	讀取目前設定馬達與編碼器誤差值容許範圍
?IN n	(Input)	讀取第 n 輸入埠的狀態。 (1 ≤ n ≤ 18)
?AIN a	(Input)	讀取類比輸入埠的狀態。 (1 ≤ a ≤ 2)
?SW n	(Switch)	讀取指撥開關。 (1 ≤ n ≤ 2)
?ST	(Status)	讀取系統狀況
?SN	(Station Number)	讀取模組編號
?ID	(IDentify Number)	讀取 ID
?VER	(Version Number)	讀取軟體編號
?@idx		讀取 idx 位址的資料庫資料
?#var		讀取變數
?參數		
?數學函數		

RS232 輸出入指令

GETI	(Get an Integer)	由 RS232 輸入一整數
GETR	(Get a Real number)	由 RS232 輸入一小數
OUT #var	(Output)	由 RS232 輸出一數字
OUT “...”	(Output)	由 RS232 輸出一字串

擴充模組指令

[...]

3.3 指令詳細說明

指令： XFA (X Fast move Absolute)

YFA (Y Fast move Absolute)

語法： XFA x ;

YFA x ;

參數範圍： $-32,767.9999 \leq x \leq +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

參數 x 另受到 XCLP(X 軸正極限)及 XCLM(X 軸負極限)所設定的範圍限制，而參數 y 也另受到 YCLP(Y 軸正極限)及 YCLM(Y 軸負極限)所設定的範圍限制。

模式： 立即，程式

功能： 快速移動 X 軸或 Y 軸至指定位置，此位置由絕對座標設定。

說明： XFA 指令只移動 X 軸，Y 軸位置不受影響。

YFA 指令只移動 Y 軸，X 軸位置不受影響。

XFA 移動速度為系統內定的 X 軸高速，此速度可由 VF 或 XVF 指令設定。

YFA 移動速度為系統內定的 Y 軸高速，此速度可由 VF 或 YVF 指令設定。

x 參數即為終點的 x 座標，y 參數即為終點的 y 座標。參數可以有正負號，也可以使用小數表示。

執行 XFA 指令(或 YFA 指令)後，會將系統內定的速度模式改為高速模式，座標模式改為絕對座標。以後只要使用 X 指令(或 Y 指令)，即可再做同樣動作。

範例：

XFA 23.5; 執行本指令會將 X 軸以高速移動至 X 座標 23.5 的位置。。

X 50; 繼續執行本指令，會延用前指令所設的速度模式及座標模，將 X 軸以高速移動至 X 座標 50 的位置。

指令： XFR (X Fast move Relative)

YFR (Y Fast move Relative)

語法： XFR x

YFR y ;

參數範圍： $-32,767.9999 < x < +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

參數 x 另受到 XCLP(X 軸正極限)及 XCLM(X 軸負極限)所設定的範圍限制，而參數 y 也另受到 YCLP(Y 軸正極限)及 YCLM(Y 軸負極限)所設定的範圍限制。

模式： 立即，程式

功能： 快速移動 X 軸或 Y 軸至指定位置，此位置由相對座標距離設定。

說明： XFR 指令只移動 X 軸，Y 軸位置不受影響。

YFR 指令只移動 Y 軸，X 軸位置不受影響。

XFR 移動速度為系統內定的 X 軸高速，此速度可由 VF 或 XVF 指令設定。

YFR 移動速度為系統內定的 Y 軸高速，此速度可由 VF 或 YVF 指令設定。

x 參數為終點相對於現在位置的 x 座標距離。y 參數為終點相對於現在位置的 y 座標距離。參數可以有正負號，也可以使用小數表示。

執行 XFR 指令(或 YFR 指令)後，會將系統內定的速度模式改為高速模式，座標模式改為及相對座標。以後只要使用 X 指令(或 Y 指令)，即可再做同樣動作。

範例：

設若現在 X 軸位置在座標 12.4 處。

XFA 23.5; 執行本指令會將 X 軸以高速移動至 X 座標 35.9 的位置。

指令： XMA (X Move Relative)

YMA (Y Move Relative)

語法： XMA x ;

YMA y ;

參數範圍： $-32,767.9999 < x < +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

此參數另受到 XCR 及 XCL 所設定的範圍限制。

模式： 立即，程式

功能： 以系統設定的工作速度，移動 X 軸至指定位置，此位置由絕對座標設定。

說明： 本指令只移動 X 軸，Y 軸位置不變。

移動速度為系統內定的 X 軸工作速度，此速度可由 VM 或 XVM 指令重新設定。

x 參數即為終點的 x 座標，可以有正負號，也可以使用小數表示。

執行本指令後，會將系統內定的速度模式改為工作速度模式，座標模式改為及絕對座標。以後只要使用 X 指令，即可再做同樣動作。

範例：

XMA 23.5; 執行本指令會將 X 軸以工作速度移動至 X 座標 23.5 的位置。

指令： XMR (X Move Relative)

YMR (Y Move Relative)

語法： XMR x；

YMR y；

參數範圍： $-32,767.9999 < x < +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

此參數另受到 XCR 及 XCL 所設定的範圍限制。

模式： 立即，程式

功能： 快速移動 X 軸至指定位置，此位置由相對座標距離設定。

說明： 本指令只移動 X 軸，Y 軸位置不變。

移動速度為系統內定的 X 軸工作速度，此速度可由 VM 或 XVM 指令重新設定。

x 參數為終點相對於現在位置的座標距離，可以有正負號，也可以使用小數表示。

執行本指令後，會將系統內定的速度模式改為工作速度模式，座標模式改為及相對座標。以後只要使用 X 指令，即可再做同樣動作。

範例：

設若現在 X 軸位置在座標 12.4 處。

XFA 23.5; 執行本指令會將 X 軸以高速移動至 X 座標 35.9 的位置。999

指令： MA (Move Absolute)

語法： MA xa,ya

參數範圍： $-32,767.9999 < xa < +32,767.9999$

$-32,767.9999 \leq ya \leq +32,767.9999$

模式： 立即，程式

功能： 以系統現在的設定速度，同時移動 X 軸及 Y 軸至指定位置，此位置由絕對座標設定。

說明： 本指令會同時移動 X 軸及 Y 軸，且行走途徑為一直線。

移動速度為系統現在的設定速度，此速度可由 VH 指令重新設定。

xa,ya 參數即為終點的(x,y)絕對座標。

相關指令： VH 設定速度

VA 設定加速度

SC 設定座標單位距離

範例：

指令： MR (Move Relative)

語法： MR xr,yr

參數範圍： $-32,767.9999 < xr < +32,767.9999$

$-32,767.9999 \leq yr \leq +32,767.9999$

模式： 立即，程式

功能： 以系統現在的設定速度，同時移動 X 軸極 Y 軸至指定位置，此指定位置由相對座標距離設定。

說明： 本指令會同時移動 X 軸及 Y 軸，且行走途徑為一直線。

移動速度為系統現在的設定速度，此速度可由 VH 指令重新設定

xr, yr 參數為終點相對於現在位置的座標距離。

相關指令： VH 設定速度

VA 設定加速度

SC 設定座標單位距離

範例：

指令： AAR (Arc Absolute Right)

語法： AAR xa,ya,r

參數範圍： $-32,767.9999 < xa < +32,767.9999$

$-32,767.9999 \leq ya \leq +32,767.9999$

$-9,999,999.999 < r < +9,999,999.999$

模式： 立即，程式

功能： 以絕對座標指定終點及半徑，做右轉圓弧。

說明： 本指令會以右轉方向〈順時針〉走一圓弧。

圓弧的起點為現在位置，圓弧的終點由 xa,ya 參數設定，圓弧的半徑由 r 參數設定。

xa,ya 參數即為終點的(x,y)絕對座標。

r 參數若為正數，為小於或等於 180°的圓弧。r 若為負數，則為大於 180°的圓弧。

移動速度為系統現在的設定速度，此速度可由 VH 指令重新設定。

相關指令： VH 設定速度

VA 設定加速度

SC 設定座標單位距離

範例：

指令： AAL (Arc Absolute Left)

語法： AAL xa,ya,r

參數範圍： $-32,767.9999 < xa < +32,767.9999$

$-32,767.9999 \leq ya \leq +32,767.9999$

$-9,999,999.999 < r < +9,999,999.999$

模式： 立即，程式

功能： 以絕對座標指定終點及半徑，做左轉圓弧。

說明： 本指令會以左轉方向〈逆時針〉走一圓弧。

圓弧的起點為現在位置，圓弧的終點由 xa,ya 參數設定，圓弧的半徑由 r 參數設定。

xa,ya 參數即為終點的(x,y)絕對座標。

r 參數若為正數，為小於或等於 180°的圓弧。r 若為負數，則為大於 180°的圓弧。

移動速度為系統現在的設定速度，此速度可由 VH 指令設定。

相關指令： VH 設定速度

範例：

指令： ARR (Arc Relative Right)

語法： ARR xr,yr,r

參數範圍： $-32,767.9999 < xr < +32,767.9999$

$-32,767.9999 \leq yr \leq +32,767.9999$

$-9,999,999.999 < r < +9,999,999.999$

模式： 立即，程式

功能： 以相對座標指定終點及半徑，做右轉圓弧。

說明： 本指令會以右轉方向〈順時針〉走一圓弧。

圓弧的起點為現在位置，圓弧的終點由 x,y 參數設定，圓弧的半徑由 r 參數設定。

xr, yr 參數為終點相對於現在位置的座標距離。

r 參數若為正數，為小於或等於 180°的圓弧。r 若為負數，則為大於 180°的圓弧。

移動速度為系統的設定速度，此速度可由 VH 指令設定。

相關指令： VH 設定速度

VA 設定加速度

SC 設定座標單位距離

範例：

指令： ARL (Arc Relative Left)

語法： ARL xr,yr,r

參數範圍： $-32,767.9999 < xr < +32,767.9999$

$-32,767.9999 \leq yr \leq +32,767.9999$

$-9,999,999.999 < r < +9,999,999.999$

模式： 立即，程式

功能： 以相對座標指定終點及半徑，做左轉圓弧。

說明： 本指令會以左轉方向〈逆時針〉走一圓弧。

圓弧的起點為現在位置，圓弧的終點由 x,y 參數設定，圓弧的半徑由 r 參數設定。

xr, yr 參數為終點相對於現在位置的座標距離。

r 參數若為正數，為小於或等於 180°的圓弧。r 若為負數，則為大於 180°的圓弧。

移動速度為系統的設定速度，此速度可由 VH 指令設定。

相關指令： VH 設定速度

範例：

指令： XJG + (X Jog Forward)
YJG + (Y Jog Forward)

語法： XJG +
YJG +

模式： 立即

功能： 模擬 JOG 輸入，設定 X 軸(或 Y 軸)往正方向運動。

說明： 本指令模擬 JOG 輸入，設定 X 軸(或 Y 軸)往正方向移動。
控制器必須在待機狀態〈先前輸入的指令都已執行完畢〉，才能執行本指令。
移動速度為系統的設定速度，此速度可由 VJ 指令設定。
收到本指令後，X 軸(或 Y 軸)開始往正方向移動，以系統現在的加速度加速。
待加至系統的設定速度，即改為等速運動。以後即繼續保持在等速運動狀態，
必須再收到 XJG 0(或 YJG 0)指令，X 軸(或 Y 軸)才會減速停止。

相關指令： XJG0 停止 X 軸 JOG 移動
YJG0 停止 Y 軸 JOG 移動
VJ 設定速度
VA 設定加速度

範例：

指令： XJG - (X Jog Reverse)
YJG - (Y Jog Reverse)

語法： XJG -
YJG -

模式： 立即

功能： 模擬 JOG 輸入，設定 X 軸(或 Y 軸)往負方向運動。

說明： 本指令模擬 JOG 輸入，設定 X 軸(或 Y 軸)往負方向移動。
控制器必須在待機狀態〈先前輸入的指令都已執行完畢〉，才能執行本指令。
移動速度為系統的設定速度，此速度可由 VJ 指令設定。
收到本指令後，X 軸(或 Y 軸)開始往負方向移動，以系統現在的加速度加速。
待加至系統的設定速度，即改為等速運動。以後即繼續保持在等速運動狀態，
必須再收到 XJG 0(或 YJG 0)指令，X 軸(或 Y 軸)才會減速停止。

相關指令： XJG0 停止 X 軸 JOG 移動
YJG0 停止 Y 軸 JOG 移動
VJ 設定速度
VA 設定加速度

範例：

指令： XJG 0 (X Jog Stop)

YJG 0 (Y Jog Stop)

語法： XJG 0;

YJG 0;

模式： 立即

功能： 模擬 Jog Off 輸入，停止 X 軸(或 Y 軸)Jog 運動。

說明： 控制器正在執行 XJG 或 YJG 指令時，執行本指令才有意義。

收到本指令後，x 軸(或 Y 軸)即進行減速至停止。減速度同加速度，可由 VA 指令設定。

相關指令： XJG + 啟動 X 軸往正方向運動

XJG - 啟動 X 軸往負方向運動

YJG + 啟動 Y 軸往正方向運動

YJG - 啟動 Y 軸往負方向運動

VH 設定速度

VA 設定加速度

範例：

指令： XH (X Home)

YH (Y Home)

語法： XH m,n

YH m,n

模式： 立即，程式

功能： 尋找 X 軸(或 Y 軸)原點。

說明： 本指令有兩個參數，說明如下

m 參數決定 X 軸回原點的模式。其功能依二進位各位元決定。

位元 0 0：回原點時，先往座標負方向尋找原點。

1：回原點時，先往座標正方向尋找原點。

位元 1 0：進原點時，使用原點開關信號。

1：進原點時，使用失步信號。

位元 2 0：出原點時，使用原點開關信號。

1：出原點時，使用編碼器 Index 信號。

位元 3 0：原點開關為常閉極性。

1：原點開關為常開極性。

位元 4 0：完成回原點的機械動作後，更新 X 軸(或 Y 軸)座標。座標新值由系統參數 XHC(或 YHC)定義。

1：完成回原點的機械動作後，不更新 X 軸(或 Y 軸)座標。

n 參數為 X 軸(或 Y 軸)原點開關銜接的輸入埠編號。

執行本指令，若未輸入參數，則參數 m 取系統參數 XHM(或 YHM)。參數 n 取用系統參數 XHN(或 YHN)。

範例：

XHM 0,3;

執行本指令，X 軸即開始往負方向移動，待收到 X 軸原點信號才停止。

X 軸將以系統現在的加速度加速，待加至系統參數 XVH 的設定速度，就改為等速運動。以後即繼續保持在等速運動狀態。待收到 X 軸原點信號，即進行減速至停止。

然後再以慢速倒車，準確地回到並停在收到原點信號的那一點上。

指令： CS (Coordinate Set)

語法： CS x, y ;

參數範圍： $-32,767.9999 < x < +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

模式： 立即，程式

功能： 訂定現在位置的座標。

說明：

系統必須在待機狀態〈先前輸入的指令都已執行完畢〉，才能執行本指令。

執行本指令，會將參數 x 及參數 y 定為現在位置的座標(x,y)。

若參數 $x=0$ ， $y=0$ ，執行本指令，即將現在位置設為座標原點(0,0)。

座標系統單位距離，可由 SC 指令設定。

本指令同時設定現在位置的 X 軸及 Y 軸座標。若只要設定 X 軸(或 Y 軸)的座標，可使用 XCS(或 YCS)指令。

範例：

指令： SC (SCale)
語法： SC sc
參數範圍： -9,999.99 < sc < +9,999.99
模式： 立即，程式
功能： 訂定座標系統的單位距離。
說明：

執行本指令，會將參數 sc 定為座標系統每一單位的原始基本距離。
系統原始的基本距離單位，在開迴路系統為輸出脈波(pulse)，即控制器每送出一脈波，機台所走的距離。在閉迴路系統則為回授脈波，亦即每收到一回授脈波，機台的移動距離。這種單位，在估算距離或速度時，使用者若覺得並不很方便，使用者可藉由本指令，重定座標系統的距離單位為慣用單位，如 mm 或 inch 等。
參數 sc 的意義為，在新座標系統中，每一單位等於多少原始基本距離單位。系統必須在待機狀態〈先前輸入的指令都已執行完畢〉，才能執行本指令。執行本指令後，座標系統的原點(0,0)的位置不會改變。現在位置的座標則會自動更新。系統現在設定的速度與加速度，實際值不會改變，但在讀取時〈指令 RVH,RVA 等〉，讀到的數值會被更新為以新單位表示的數值。

範例一：

開迴路系統

假設在本例中使用步進馬達，皮帶驅動機台。馬達每轉一轉，機台移動 80mm。並假設使用每轉 5000 分割的微步驅動器。

在本例中，系統原始的基本距離單位〈控制器每送出一脈波，機台所走的距離〉為

$$80\text{mm}/5000=0.016\text{mm}$$

使用者若不想使用這種距離單位，可執行下列指令，即可將座標單位改為習慣的 mm

$$\text{SC } 62.5; \quad 1/0.016=62.5$$

範例二：

閉迴路系統

假設在本例中使用光學尺做位置回授，光學尺的解析度為 0.005mm

在本例中，系統原始的基本距離單位即為 0.005mm。〈閉迴路系統中，系統的基本距離單位即回授解析度，與馬達解析度或傳動比無關，使用者若不想使用這種距離單位，可執行下列指令，即可將座標單位改為習慣的 mm

$$\text{SC } 200; \quad 1/0.005=200$$

指令： VF (Velocity Fast)

語法： VF v

參數範圍： $0 < v < 32,767.9999$

模式： 立即，程式

功能： 設定 X 軸及 Y 軸的高速。

說明： CSR200 有兩組內建速度參數，一組為供高速移動時使用，另一組為工作速度。本指令就是用來設定系統高速移動時的速度。當執行 FA 指令或 FR 指令時，X 軸及 Y 軸就會以本指令設定的速度移動。

參數 v 的單位為：系統座標單位距離/每秒。其中系統座標單位距離可由 SC 指令設定。

本指令會同時設定 X 軸及 Y 軸的工作速度。若要單獨設定 X 軸或 Y 軸的工作速度，可以藉由設定系統參數 XVF 或 YVF 的方式達成。

範例：

指令： VM (Velocity Move)

語法： VM v

參數範圍： $0 < v < 32,767.9999$

模式： 立即，程式

功能： 設定 X 軸及 Y 軸的工作速度。

說明： CSR200 有兩組內建速度參數，一組為供高速移動時使用，另一組即為工作速度。本指令就是用來設定系統的工作速度。當執行 MA 指令或 MR 指令時，X 軸及 Y 軸就會以本指令設定的工作速度移動。

參數 v 的單位為：系統座標單位距離/每秒。其中系統座標單位距離可由 SC 指令設定。

本指令會同時設定 X 軸及 Y 軸的工作速度。若要單獨設定 X 軸或 Y 軸的工作速度，可以藉由設定系統參數 XVM 或 YVM 的方式達成。

範例：

指令： VA (Velocity Accelerate)

語法： VA a

參數範圍： $0 < a < 32,767.9999$

模式： 立即，程式

功能： 同時設定 X 軸及 Y 軸的加速度。

說明： 本指令設定運動起動時的加速度，以及減速停止時的減速度。

參數 a 的單位為：系統座標單位距離/每秒每秒。其中系統座標單位距離可由 SC 指令設定。

本指令會同時設定 X 軸及 Y 軸的加速度。若要單獨設定 X 軸或 Y 軸的加速度，可以藉由設定系統參數 XVA 或 YVA 的方式達成。

範例：

指令： SET P (SET output)

語法： SET Pn

參數範圍： $1 \leq n \leq 8$

模式： 立即，程式

功能： 設定輸出埠為開(On)狀態。

說明： CSR200 控制器共有 8 個供一般用途的輸出，每一輸出均可單獨控制輸出的 On/Off 狀態。

本指令的功能就是將參數 n 指定的輸出埠設定為「開」的狀態，也就是將其輸出光耦合器的電晶體設定為 On 的狀態。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

範例：

指令： CLR P (Clear output)

語法： CLR Pn

參數範圍： $1 \leq n \leq 8$

模式： 立即，程式

功能： 設定輸出為關(Off)狀態。

說明： CSR200 控制器共有 8 個供一般用途的輸出，每一輸出均可單獨控制輸出的 On/Off 狀態。

本指令的功能就是將參數 n 指定的輸出設定為「關」的狀態，也就是將其輸出光耦合器的電晶體設定為 Off 的狀態。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

範例：

指令： CHG P (Change output)

語法： CHG Pn

參數範圍： $1 \leq n \leq 8$

模式： 立即，程式

功能： 設定輸出為關(Off)狀態。

說明： CSR200 控制器共有 8 個供一般用途的輸出，每一輸出均可單獨控制輸出的 On/Off 狀態。

執行本指令，將改變參數 n 指定的輸出埠的輸出狀態，原來是輸出 On 的會改成 OFF，原來輸出 Off 的會改成 On。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

範例：

指令： PLS P (Pulse output)

語法： PLS Pn,tm

參數範圍： $1 \leq n \leq 8$

$1 \leq tm \leq 65535$

模式： 立即，程式

功能： 由指定的輸出埠輸出一脈波。

說明： CSR200 控制器共有 8 個供一般用途的輸出，每一輸出均可單獨控制輸出的 On/Off 狀態。

執行本指令，將由參數 n 指定的輸出埠輸出一脈波。詳細動作就是將輸出光耦合器的電晶體 Turn On 一段時間後再 Turn off。電晶體 Turn On 的時間，也就是脈波寬度，是由參數 tm 設定，單位為 4msec。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

範例：

指令： XSET P (SET output by X position)

YSET P (SET output by Y position)

語法： XSET Pn,x

YSET Pn,y

參數說明： n 輸出口 No.

x X 軸座標位置

y Y 軸座標位置

參數範圍： $1 \leq n \leq 8$

$-32,767.9999 \leq x \leq +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

模式： 立即，程式

功能： 在 X 軸〈或 Y 軸〉位置到達指定座標時，將指定的輸出設定為 On 狀態。

說明： 本指令與 SET Pn 指令類似。但執行本指令，並不會立即改變輸出狀態。而是在 X 軸〈或 Y 軸〉到達參數 x 指定的位置時，才將參數 n 指定的輸出口設定為 On 狀態(輸出光耦合器的電晶體 On)。輸出狀態改變設定時，X 軸〈或 Y 軸〉的運動不受任何影響，並不需要做減速或停車配合。

本指令較適合在閉迴路模式下使用。在開迴路模式下使用，輸出口設定改變的位置，會有些許延遲誤差。

本指令可配合各種運動控制指令使用，包括手動及 Jog 移動。

本指令只在下達後第一次 X 軸〈或 Y 軸〉到位時執行，以後即使再重複到位，也不再動作。

本指令下達後若要取消，可再下達沒有 x 參數〈或 y 參數〉的同樣指令即可。有關輸出電路的使用及配線方法，請參照“輸出輸入電路”。

相關指令：

範例：

指令： XCLR P (Clear output by X position)

YCLR P (Clear output by Y position)

語法： XCLR Pn,x

YCLR Pn,y

參數說明： n 輸出口 No.

x X 軸座標位置

y Y 軸座標位置

參數範圍： $1 \leq n \leq 8$

$-32,767.9999 \leq x \leq +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

模式： 立即，程式

功能： 在 X 軸〈或 Y 軸〉位置到達指定座標時，將指定的輸出設定為 Off 狀態。

說明： 本指令與 CLR P 指令類似。但執行本指令，並不會立即改變輸出狀態。而是在 X 軸〈或 Y 軸〉到達參數 x 指定的位置時，才將參數 n 指定的輸出口設定為 Off 狀態(輸出光耦合器的電晶體 Off)。輸出狀態改變設定時，X 軸〈或 Y 軸〉的運動不受任何影響，並不需要做減速或停車配合。

本指令較適合在閉迴路模式下使用。在開迴路模式下使用，輸出口設定改變的位置，會有些許延遲誤差。

本指令可配合各種運動控制指令使用，包括手動及 Jog 移動。

本指令只在下達後第一次 X 軸〈或 Y 軸〉到位時執行，以後即使再重複到位，也不再動作。

本指令下達後若要取消，可再下達沒有 x 參數〈或 y 參數〉的同樣指令即可。有關輸出電路的使用及配線方法，請參照“輸出輸入電路”。

相關指令：

範例：

指令： XCHG P (Clear output by X position)

YCHG P (Clear output by Y position)

語法： XCHG Pn,x1,x2,...

YCHG Pn,y1,y2,...

參數說明： n 輸出口 No.

x1,x2,... X 軸座標位置，最多可輸入 16 個位置。

y1,y2,... Y 軸座標位置，最多可輸入 16 個位置。

參數範圍： $1 \leq n \leq 8$

$-32,767.9999 \leq x \leq +32,767.9999$

$-32,767.9999 \leq y \leq +32,767.9999$

模式： 立即，程式

功能： 在 X 軸〈或 Y 軸〉位置到達指定座標時，改變參數 n 指定的輸出埠的狀態，原來為 On 的改為 Off，原來為 Off 的改為 On。

說明： 本指令與 CHG P 指令類似。但執行本指令，並不會立即改變輸出狀態。而是在 X 軸〈或 Y 軸〉到達參數 x 指定的位置時，才改變參數 n 指定的輸出埠的狀態，原來為 On 的改為 Off，原來為 Off 的改為 On。

可一次輸入最多 16 個位置參數，即可依輸入順序，在 X 軸〈或 Y 軸〉到達指定位置時，反復改變輸出埠的狀態。

輸出狀態改變設定時，X 軸〈或 Y 軸〉的運動不受任何影響，並不需要做減速或停車配合。

本指令較適合在閉迴路模式下使用。在開迴路模式下使用，輸出口設定改變的位置，會有些許延遲誤差。

本指令可配合各種運動控制指令使用，包括手動及 Jog 移動。

本指令只在下達後第一次 X 軸〈或 Y 軸〉到位時執行，以後即使再重複到位，也不再動作。

本指令下達後若要取消，可再下達沒有 x 參數〈或 y 參數〉的同樣指令即可。

有關輸出電路的使用及配線方法，請參照“輸出輸入電路”。

範例： XMA 2000; 將 X 軸移到座標 2000 位置

CLR P2; 將輸出 2 設為 off

CPLBMX 2,2500,3000,3500,4000; 設定改變輸出 2 的 X 軸的座標位置

XMA 5000 將 X 軸移到座標 5000 位置

執行以上程式，輸出 2 會配合 X 軸的移動，

在到達 2500 位置時 On，

在到達 3000 位置時 Off，

在到達 3500 位置時 On，

在到達 4000 位置時 Off。

指令： XPLS P (Pulse output by X position)

YPLS P (Pulse output by Y position)

語法： XPLS Pn,tm,x1,x2,...

YPLS Pn,tm,y1,y2,...

參數說明： n 輸出口 No.

x1,x2,... X 軸座標位置，最多可輸入 16 個位置。

y1,y2,... Y 軸座標位置，最多可輸入 16 個位置。

參數範圍： $1 < n < 8$

$1 < tm < 65535$

$-32,767.9999 < x < +32,767.9999$

$-32,767.9999 < y < +32,767.9999$

模式： 立即，程式

功能： 在 X 軸〈或 Y 軸〉位置到達指定座標時，由指定的輸出埠輸出脈波。

說明： 本指令與 PLS P 指令類似。但執行本指令，並不會立即輸出脈波。而是在 X 軸〈或 Y 軸〉到達參數 x 指定的位置時，才由參數 n 指定的輸出埠輸出一脈波，也就是將輸出光耦合器的電晶體 Turn On 一段時間後再 Turn off。電晶體 Turn On 的時間，也就是脈波寬度，是由參數 tm 設定，單位為 4msec。

可一次輸入最多 16 個位置參數，即可依輸入順序，在 X 軸〈或 Y 軸〉到達指定位置時，重復輸出脈波。

輸出脈波時，X 軸〈或 Y 軸〉的運動不受任何影響，並不需要做減速或停車配合。

本指令較適合在閉迴路模式下使用。在開迴路模式下使用，輸出脈波的位置，會有些許延遲誤差。

本指令可配合各種運動控制指令使用，包括手動及 Jog 移動。

本指令只在下達後第一次 X 軸〈或 Y 軸〉到位時執行，以後即使再重復到位，也不再動作。

本指令下達後若要取消，可再下達沒有 x 參數〈或 y 參數〉的同樣指令即可。有關輸出電路的使用及配線方法，請參照“輸出輸入電路”。

範例：

指令： JP (JumP)

語法： JP @addr

JP \$label

JP Ln

參數說明： addr 直接位址

\$label 標籤位址

Ln 變數位址

參數範圍： $0 < \text{addr} < 8192$

模式： 程式

功能： 將程式無條件跳躍至參數所指定的位址繼續執行

說明： 本指令能改變程式流程，無條件跳躍至參數所指定的位址。

代表位址的參數，可以標籤或變數方式表示。標籤位址必須以\$起始，變數位址必須以#起始。

相關指令：

範例：

指令： JI (Jump at Input)

語法： JI n,@addr

JI n,\$label

參數說明： n 輸入口 No.

addr 直接位址

\$label 標籤位址

參數範圍： $1 < n < 16$

$0 < \text{addr} < 1024$

模式： 程式

功能： 參數 n 指定的輸入為 On 時，程式跳躍至參數 addr 所指定的位址繼續執行

說明： 本指令能依輸入的狀態，決定程式流程方向。當參數 n 指定的輸入口為 On (輸入光耦合器的 LED 有電流致能) 時，程式跳躍至參數 addr 所指定的位址。反之，參數 n 指定的輸入口為 Off (輸入光耦合器的 LED 沒有電流致能) 時，程式不跳躍，繼續執行下一指令。

代表位址的參數，可以標籤或變數方式表示。標籤位址必須以 \$ 起始，變數位址必須以 # 起始。

本指令與 JPNI 指令類似，僅輸入口的狀態條件相反。

有關輸入電路的使用及配線方法，請參照“輸出輸入電路”。

相關指令：

範例：

指令： JNl (Jump at No Input)

語法： JNl n,addr

JNl n,\$label

參數說明： n 輸入口 No.

addr 直接位址

\$label 標籤位址

參數範圍： $1 < n < 16$

$0 < \text{addr} < 1024$

模式： 程式

功能： 參數 n 指定的輸入為 On 時，程式跳躍至參數 addr 所指定的位址繼續執行

說明： 本指令能依輸入的狀態，決定程式流程方向。當參數 n 指定的輸入口為 Off
〈輸入光耦合器的 LED 沒有電流致能〉時，程式跳躍至參數 addr 所指定的位址。反之，參數 n 指定的輸入口為 On 〈輸入光耦合器的 LED 有電流致能〉時，程式不跳躍，繼續執行下一指令。

代表位址的參數，可以標籤或變數方式表示。標籤位址必須以\$起始，變數位址必須以#起始。

本指令與 JPNl 指令類似，僅輸入口的狀態條件相反。

有關輸入電路的使用及配線方法，請參照“輸出輸入電路”。

相關指令：

範例：

指令： JTI (Jump depend on Timer and Input)

語法： JTI n,tm,addr

參數說明： n 輸入口 No.
tm 延遲時間
addr 直接位址

參數範圍： $1 < n < 16$

$0 < tm < 65536$

$0 < addr < 1024$

模式： 程式

功能： 參數 n 指定的輸入為 On 時，程式跳躍至參數 addr 所指定的位址繼續執行

說明： 本指令與 JI 相似，依輸入的狀態，決定程式流程方向。若參數 n 指定的輸入口為 On (輸入光耦合器的 LED 有電流致能) 時，或在 tm 設定的時間內，由 Off 轉為 On，程式跳躍至參數 addr 所指定的位址。反之，但若參數 n 指定的輸入口一直為 Off (輸入光耦合器的 LED 沒有電流致能) 時，程式不跳躍，繼續執行下一指令。

代表位址的參數，可以標籤或變數方式表示。標籤位址必須以 \$ 起始，變數位址必須以 # 起始。

本指令與 JNTI 指令類似，僅輸入口的狀態條件相反。

有關輸入電路的使用及配線方法，請參照“輸出輸入電路”。

相關指令：

範例：

指令： JNTI (Jump depend on Timer and No Input)

語法： JNTI n,tm,addr

參數說明： n 輸入口 No.
tm 延遲時間
addr 直接位址

參數範圍： $1 < n < 16$

$0 < tm < 65536$

$0 < addr < 1024$

模式： 程式

功能： 參數 n 指定的輸入為 On 時，程式跳躍至參數 addr 所指定的位址繼續執行

說明： 本指令與 JNTI 相似，由輸入口 On/Off 的狀態，決定程式流程方向，並可加上時間限制。

若參數 n 指定的輸入口為 On (輸入光耦合器有電流致能) 時，或在 tm 設定的時間內，由 Off 轉為 On，程式跳躍至參數 addr 所指定的位址。反之，但若參數 n 指定的輸入口一直為 Off (輸入光耦合器沒有電流致能) 時，程式不跳躍，繼續執行下一指令。

代表位址的參數，可以標籤或變數方式表示。標籤位址必須以 \$ 起始，變數位址必須以 # 起始。

本指令與 JNTI 指令類似，僅輸入口的狀態條件相反。

有關輸入電路的使用及配線方法，請參照“輸出輸入電路”。

相關指令：

範例：

指令： SI (Set Intergel variable)
語法： SI #var
SI #var=data
SI #var1,#var2,#var3...
SI #var1=data1,#var2=data2,#var3=dat3...

參數說明：

參數範圍：

模式： 程式，立即

功能： 宣告一整數變數

說明： 本指令宣告一整數變數。所謂宣告，即是要求在系統的記憶體中空出一空間，儲存這個變數。變數一定要先宣告才可以使用。

整數變數只能儲存小數，若將一整數常數存到整數變數，該常數會自動被轉成小數型式。但若將一整數變數存到小數變數，則會造成錯誤。

變數的名稱是一組以#開頭的字串構成，字串中可包括大小寫英文字母及數字，也可以使用中文，但總長不能超過 12 個字(或 6 個中文字)。

例如#ABC，#Count100，#參數 1 等，都是可接受的變數名稱。

設定變數新值，有下列多種方法：

在宣告的同時設定其值

```
SI #AA=125.3;
```

以常數設定

```
#AA=200;
```

以變數設定

```
#AA=#BB;
```

```
#AA=ST;
```

```
#AA=@1;
```

以函數設定

```
#AA=COS(30);
```

以運算式設定

```
#AA=#AA+1;
```

以其它模組的傳值設定

```
#AA=[ST];
```

相關指令：

CLR VAR 清除變數

指令： SR (Set Real variable)
語法： SR #var
SR #var=data
SR #var1,#var2,#var3...
SR #var1=data1,#var2=data2,#var3=dat3...

參數說明：

參數範圍：

模式： 程式，立即

功能： 宣告一小數變數

說明： 本指令宣告一小數變數。所謂宣告，即是要求在系統的記憶體中空出一空間，儲存這個變數。變數一定要先宣告才可以使用。

小數變數只能儲存小數，若將一整數常數存到小數變數，該常數會自動被轉成小數型式。但若將一整數變數存到小數變數，則會造成錯誤。

變數的名稱是一組以#開頭的字串構成，字串中可包括大小寫英文字母及數字，也可以使用中文，但總長不能超過 12 個字(或 6 個中文字)。

例如#ABC，#Count100，#參數 1 等，都是可接受的變數名稱。

設定變數新值，有下列多種方法：

在宣告的同時設定其值

```
SR #AA=125.3;
```

以常數設定

```
#AA=2.4;
```

以變數設定

```
#AA=#BB;
```

```
#AA=XVF;
```

```
#AA=@1;
```

以函數設定

```
#AA=COS(30);
```

以運算式設定

```
#AA=#AA*2;
```

以其它模組的傳值設定

```
#AA=[PCX];
```

相關指令：

CLR VAR 清除變數

四、函數

4-1. 函數說明

CSR200 內建有一些非常實用的數學函數。利用這些數學函數的計算能力，使用者就可以用很簡單的程式語法，完成很複雜的功能變化。

CSR200 有下列函數

SIN(r) 正弦函數
COS(r) 餘弦函數
TAN(r) 正切函數
ATN(r) 反正切函數
SQR(r) 開平方

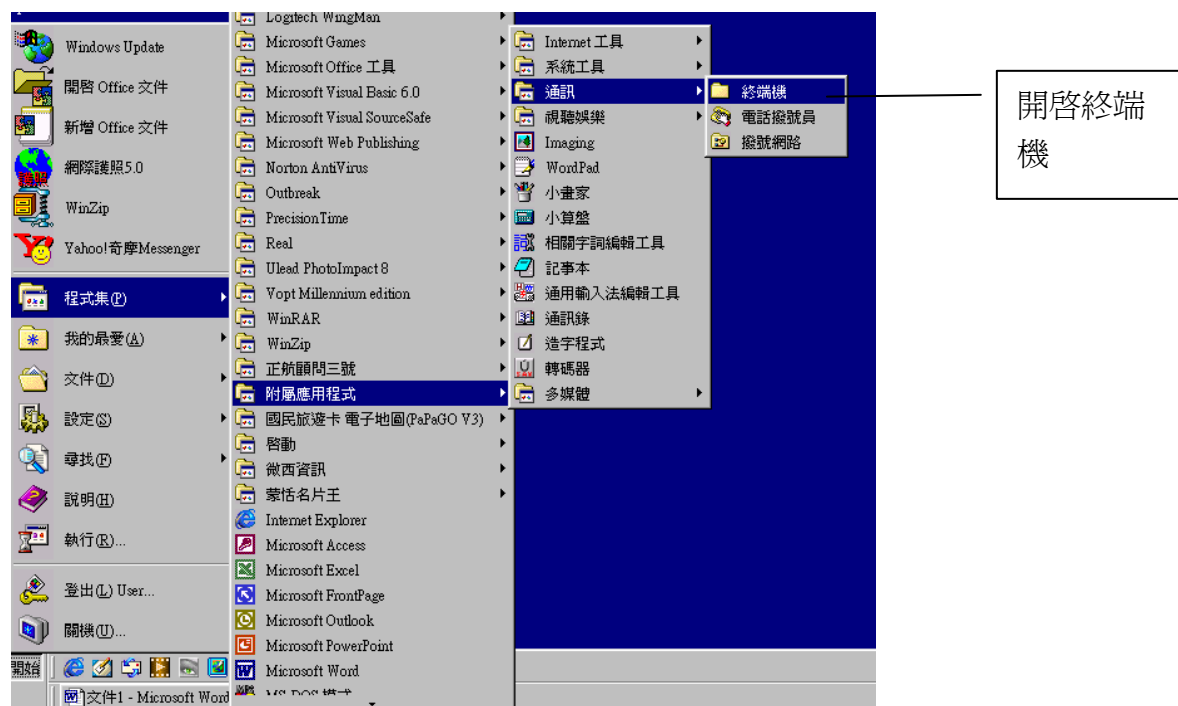
4-2. 函數使用方法例

```
?SIN(30.5)  
SR #A=30.5  
SR #R  
#R=SIN(#A)
```

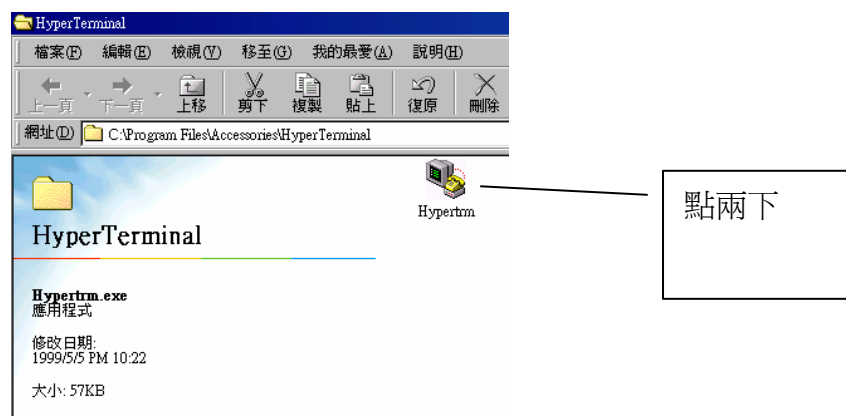
指撥開關設定及用法說明

開關編號	設定狀態	設定功能
1	On	電源開啓，程式不會自動啓動。必須等按”啓動”鍵或收到”G”指令後，程式才會啓動。
	Off	電源開啓後，程式即自動啓動。
2	On	啓用回應功能(Echo back)。RS232 收到任何訊息，都會回應同樣訊息，如同回音一樣，方便終端機模式的通訊。
	Off	關閉回應功能。
3	On	設定本模組爲最高模組。單片模組應用，或多片模組串聯應用時的第一片，本開關必須設定爲 On。
	Off	設定本模組非最高模組。
4	On	設定本模組爲最終模組。單片模組應用，或多片模組串聯應用時的最後一片，本開關必須設定爲 On。
	Off	設定本模組非最終模組。
5	On	啓用極限開關功能。使用 IN1,IN2 做爲 X 軸的正負極限開關，IN4,IN5 做爲 Y 軸的正負極限開關。
	Off	關閉極限開關功能。IN1,IN2,IN4,IN5 只當做一般輸入口用。

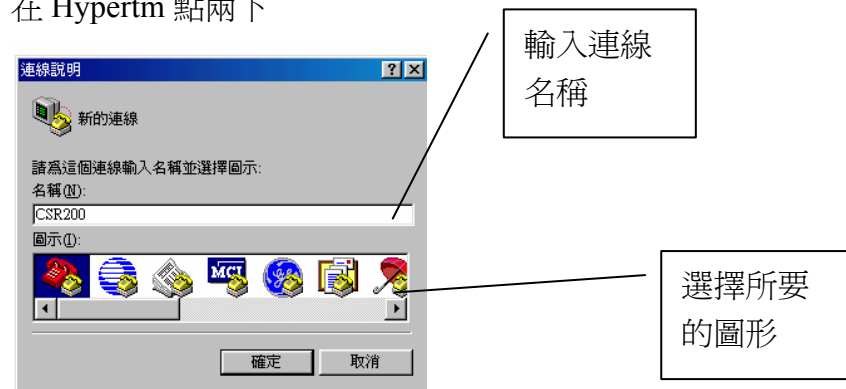
7-1. CSR200 控制器連線圖示



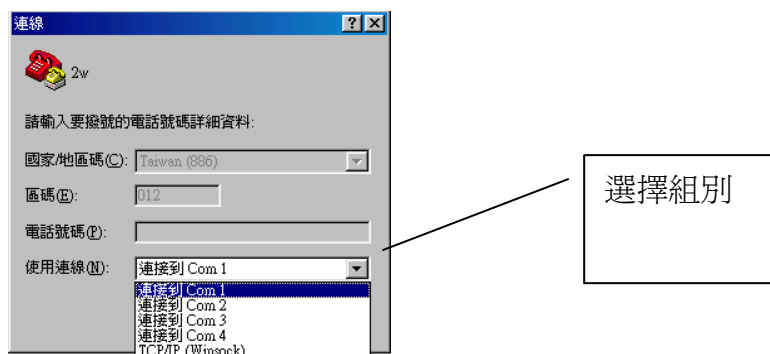
先在程式集→附屬應用程式→通訊→終端機 內開啟終端機
(若無終端機,可在控制台→新增/移除程式→WINDOWS 安裝程式內的通訊開啓即可)



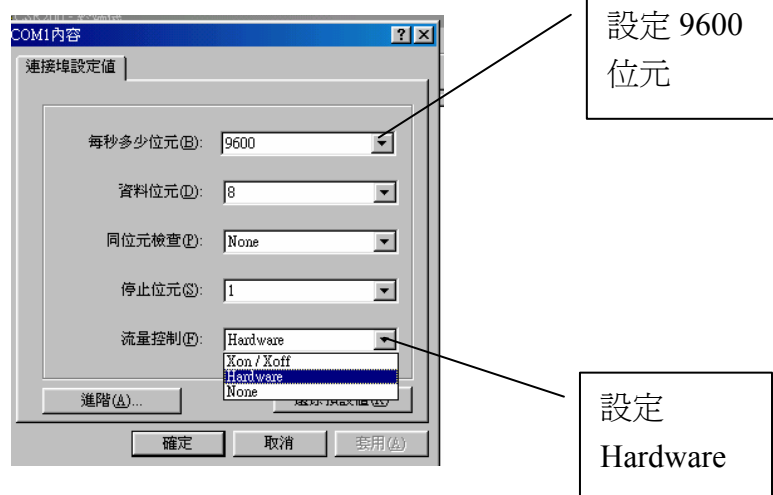
在 Hypertm 點兩下



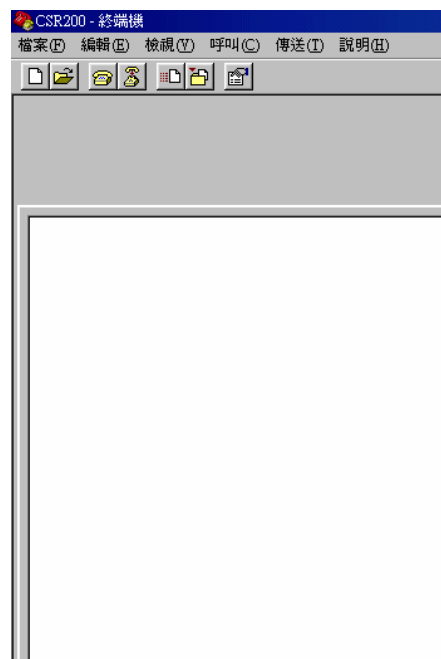
在名稱內輸入連線代號,及選擇圖形後按確定



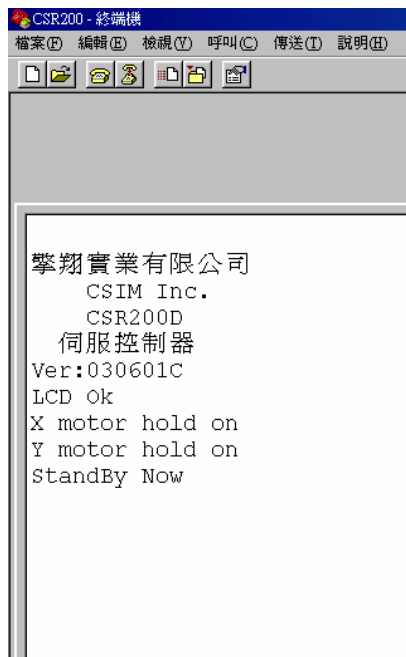
連線方格內選擇 RS232 的連線組別



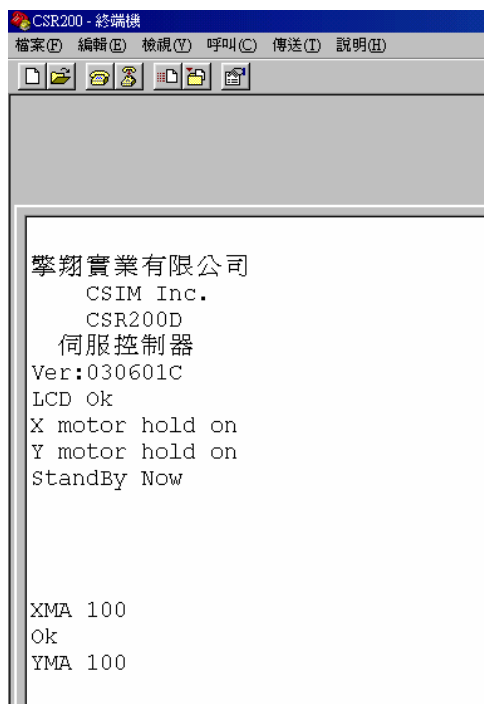
在 COM#設定完按確定



完成後電腦終端機即成待機模式



將 CSR200 控制器與電腦連線後,電源開起終端機螢幕即出現如上圖所示,則表示連線成功



此時下達 CSR200 的指令或程式,馬達就可開始動作