

CSBL600

AC 伺服馬達控制器

使用說明書

Ver. 041229

目錄

- 一、 CSBL600 無刷伺服馬達控制器簡介
- 二、系統設定
 - 1. 指撥開關設定一覽表
 - 2. 指撥開關設定說明
- 三、系統參數
 - 1. 系統參數的讀取與更新
 - 2. 系統參數一覽表
 - 3. 各系統參數詳細說明
- 四、指令
 - 1. 指令說明
 - 2. 指令一覽表
 - 3. 各指令詳細說明
- 五、函數與變數
 - 1. 函數說明
 - 2. 函數一覽表
- 六、與 PC 連線
 - 1. 啓用 Window “終端機” 軟體
 - 2. 第一次在 PC 上使用 “終端機” 軟體與 CSBL600 連線
 - 3. 編輯 CSBL600 程式
 - 4. 利用外部文字處理程式編寫程式
 - 5. 上傳 CSBL600 的程式至 PC
- 七、輸出入埠電路與配線
 - 1. CSBL600 輸出入埠腳位定義
 - 2. 輸入埠電路
 - 3. 輸入電路外部配線範例
 - 4. 輸出電路
 - 5. 輸出電路外部配線範例
 - 6. RS232 接頭定義
 - 7. 編碼器接頭定義

八、使用 CSBL600 注意事項

1. 安裝場所請注意事項
2. 配綫請注意事項

一、CSBL600 無刷伺服馬達控制器簡介

CSBL600 為一多功能的 AC 伺服馬達控制器，可以用即時命令，控制伺服馬達的所有動作。也可以用執行程式的方式，設定伺服馬達的動作程序。此外 CSBL600 也兼具了伺服馬達驅動器功能，只要有 AC110V 或 220V 電源，不需再外加任何其它電路，就可以直接驅動 750W 以下的 AC 伺服馬達。

CSBL600 的主要規格及特性如下：

1. 兼具控制器與驅動器功能。

CSBL600 是一功能完備的 AC 伺服馬達控制器，可以接受立即指令，也可以執行程式。更可以如一般 AC 伺服馬達驅動器的用法，接受來自其它控制器或 PLC 的脈波控制信號驅動伺服馬達。

2. 使用 AC110V 或 220V 電源。

直接使用 AC110V 或 220V 電源，不需再外加變壓器或 DC 電源供應器，即可使用。

3. 超強的 PLC 功能。

除馬達的控制外，CSBL600 還附有 8 組輸入及 4 組輸出功能，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，非一般 PLC 所能相比。

4. 可以與電腦連線使用，也可以單機應用。

CSBL600 可以透過 RS232 介面與 PC 或其它裝置連線，分享應用資源，或接受來自 PC 的指令控制。也可以將程式儲存在 CSBL600 的 Flash ROM 中，配合自動啟動（AutoRun）功能，單機獨立運用。

5. 行進間調速。

CSBL600 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程式動作及定位距離。

6. 完整的指令集。

CSBL600 有超過 90 種指令的指令集，及 40 組以上的系統參數，並附有功能強大的函數功能，變數功能及浮點運算能力等。讓使用者可以用很簡單的語法，發展功能複雜的應用程式。

7. 程式、數據及系統參數，以分散化方式管理。

CSBL600 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。

二、系統設定

CSBL600 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 CSBL600 各項功能的預設值及啟動程序，是依據兩項輸入設定來決定。其一為指撥開關，另一則為儲存在系統記憶體中的系統參數。本章將詳細說明有關指撥開關的設定，以及跟指撥開關設定有關的輸入輸出信號。系統參數則在下一章中再做說明。

2-1. 指撥開關設定一覽表

開關編號	設定狀態	設定功能		備註
1	ON	設定 CSBL600 為驅動器模式		請參考 2-2-1. 說明
	OFF	設定 CSBL600 為控制器模式		
2	ON	驅動器模式	脈波及方向輸入	請參考 2-2-2 及 2-2-3 說明
		控制器模式	電源開啓後，程式即自動啟動（Auto Run）。	
	OFF	驅動器模式	CW/CCW 雙脈波輸入	
		控制器模式	電源開啓，程式不會自動啟動。必須等收到“G”指令後，程式才會啟動	
3	ON	關閉 RS232 埠的回應功能〈Echo Off〉		請參考 2-2-4. 說明
	OFF	開啓 RS232 埠的回應功能〈Echo On〉		
4	ON	保留給使用者，可由程式規劃用途		請參考 2-2-5. 說明
	OFF			
5	ON	保留給使用者，可由程式規劃用途		
	OFF			
6	ON	保留給使用者，可由程式規劃用途		
	OFF			

2-2. 指撥開關設定說明

2-2-1. CSBL600 有兩種使用模式，一為驅動器模式，另一為控制器模式，兩者的選擇是由指撥開關 SW1 決定。當 SW1 為 ON 時，CSBL600 被設定為驅動器模式。當 SW1 為 OFF 時，CSBL600 被設定為控制器模式。

選擇驅動器模式，一般是搭配另一控制器或 PLC 共同使用。此時 CSBL600 就如同一般 AC 伺服馬達的驅動器，直接接受來自其它控制器或 PLC 的脈波控制信號，每收到一脈波即驅動伺服馬達旋轉一步。

若選擇控制器模式，則伺服馬達的運動完全由 CSBL600 自行控制。使用者可以用立即指令，也可以用執行程式方式，控制馬達運轉。

- 2-2-2. 指撥開關 SW2 的功能，在驅動器模式和控制器模式時會不一樣。在當驅動器使用時，SW2 被用做選擇控制馬達動作的輸入信號為單脈波信號（脈波及方向）或雙脈波信號（正轉脈波及逆轉脈波）。在當控制器使用時，SW2 被用做選擇當電源開啓時，程式要不要自動啓動（Auto Run）。
- 2-2-3. CSBL600 在驅動器模式時，也可藉由設定系統參數 QEP，選擇使用手搖輪或編碼器的輸出來控制馬達動作。此時指撥開關 SW2 的設定即被忽略。
- 2-2-4. CSBL600 可以透過 RS232 介面與 PC 或其他裝置連線。指撥開關 SW3 的功能，就是設定當 RS232 收到任一字元時，是否要回傳（Echo Back）。這項回傳功能，一般是配合 PC 的終端機軟體使用。在使用其他軟體與 CSBL600 連線時，若用不到此功能，可將其關閉，以免反造成干擾。
- 2-2-5. 使用者可以隨時用“?SW”指令讀取指撥開關的狀態。也可以在程式中插入“JS”或“JNS”指令，依指撥開關的設定狀態變更程式流程。

2-3. 輸入及輸出埠說明

CSBL600 有 8 個輸入埠及 4 個輸出埠。在控制器模式時，所有的輸入及輸出，均可由程式或指令直接控制。但在驅動器模式時，CSBL600 部份輸入及輸出會被設定成專用功能，如下表：

輸入埠編號	功能	備註
輸入 1	脈波輸入或正轉脈波輸入	
輸入 2	方向輸入或逆轉脈波輸入	
輸入 3	關閉馬達電流（Hold Off）	
輸入 4	重置(Reset)	
輸入 5	強制關機（Snap Off）	請參考 2-3-1.說明
輸入 6		
輸入 7	正極限	
輸入 8	負極限	

輸出埠編號	功能	備註
輸出 1	到位	請參考 2-3-2 說明
輸出 2	跳機（Crash）	請參考 2-3-3 說明
輸出 3	備妥（Ready）	請參考 2-3-4 說明
輸出 4		

- 2-3-1. CSBL600 在驅動器模式時，輸入埠 3 為關閉馬達電流（Hold Off），輸入埠 5 為強制關機（Snap Off），兩者動作類似，都可以用來關閉馬達電流。但輸入埠 3（Hold Off）可以隨時作動，也可以隨時解除。而輸入埠 5（Snap Off）一旦被觸發，就必須以重置（Reset）方式才可以解除。一般應用時，輸入埠 3（Hold Off）多是由控

制器直接控制。輸入埠 5 (Snap Off) 則多被當做安全保護裝置 (如溫度感應開關等) 的輸入埠。

輸入 3 的作動極性可由參數 HFP 設定。輸入 5 的作動極性可由參數 SNP 設定。

- 2-3-2. CSBL600 在驅動器模式時，馬達是否已”到位”的判定，是以系統參數 ELP 為允許誤差的依據。當馬達位置誤差小於或等於 ELP 時，即判定為到位。此時輸出埠 1 即會有輸出(Turn On)。(請參考 ELP 等系統參數的說明)
- 2-3-3. 當馬達實際位置與命令位置的誤差超過 EL 參數所設定的極限時，CSBL600 為了保障系統安全，會關閉對馬達的所有功率輸出，讓馬達緊急停俾，這就是所謂“跳機”(Crash)。當發生這種狀況時，CSBL600 會藉由輸出埠 2 送出跳機信號給控制器或系統中其它裝置。跳機信號的型態(Turn On 或 Turn Off)，可由參數 CRP 設定。在強制關機 (Snap Off) 時，CSBL600 也會送出跳機信號。
- 2-3-4. 在驅動器模式下打開電源，CSBL600 完成開機動作後，就會由輸出埠 3 送出備妥信號(Ready)。當發生跳機或被強制關機 (Snap Off) 時，除了送出跳機信號外，CSBL600 也會同時關閉備妥信號。
- 2-3-5. 其它有關於 CSBL600 輸出入埠的脚位及電路配綫等詳細資料，請參關第七章的說明。

三、系統參數

系統參數是被用來做功能設定的軟體參數。這些參數是儲存在系統的快閃記憶體（Flash ROM）中，所以即使關閉電源也不會被清除。藉由設定系統參數，可以設定 CSBL600 的各項功能，如速度、距離單位等。

系統參數會直接影響 CSBL600 的功能設定。不正確的系統參數，甚至會造成系統無法正常工作。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，造成系統無法正常工作時，可執行“DF”指令，將所有系統參數改回安全預設值。再逐一調整各參數，即可重新設定系統功能。

3-1. 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，CSBL600 就會立即回應該參數的現值。

使用例：

由終端機輸入“?VM”。

CSBL600 會立即將 VM〈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“VM=200”。

CSBL600 就會將工作速度，設定為 200。以後再收到“MA”等指令，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。

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。

3-2. CSBL-600 系統參數一覽表

CSBL600 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
ECR	EnCoder Resolution	1~65535		編碼器解析度
POL	POLe	1~255		馬達磁極數
PHO	PHase Origin	1~65535		馬達相位原點
VF	Velocity Fast	正數	500	高速
VM	Velocity Move	正數	200	工作速度
VA	Velocity Accelerate	正數	500	加速度
VB	Velocity Base	正數	5	起始速度
VJ	Velocity Jog	正數	200	JOG 速度
VH	Velocity Home	正數	50	回原點速度
CLP CL+	CLip Plus		5000	正極限
CLM CL-	CLip Minus		-5000	負極限
ECL	Enable CLip	0~1	0	啓用軟體極限保護
EHC	Enable Hard Clip	0~1	0	啓用硬體極限保護
EL	Error Limit	1~65535	200	誤差極限
ELH	Error Limit at Home process	1~65535	200	執行回原點動作時的誤差極限
ELI	Error Limit Integral	1~65535	8000	誤差累積極限
ELP	Position Error Limit	1~65535	40	到位允許誤差
MDI	Motor Direction	0~1	1	馬達方向
EDI	Encoder Direct	0~1	1	編碼器方向
HC	Home Coordinate		0	原點座標
HM	Home Mode	0~127	6	回原點模式
HN	Home sensor No.	1~8	5	原點開關輸入埠編號

參數名稱	英文助憶	設定範圍	預設值	功能說明
TL	Torque Limit	0~255	255	馬達扭力極限
VL	Voltage Limit	0~255	Auto	電壓極限(煞車)
SC1	SCale1	1~65535	10	每單位距離相對的編碼器步數 (Encoder Counts)
SC2	SCale2	1~65535	2000	
RSC1	Input Scale 1	1~65535	1	在驅動器模式下，每輸入一脈波馬達旋轉步數
RSC2	Input Scale 2	1~65535	1	
KP	Proportional Gain	0~4.0	1.0	比例控制器增益
KPI	Proportional Gain at Idle state	0~4.0	1.0	停止時比例控制器增益
KD	Derivative Gain	0~16.0	2.0	微分控制器增益
KDI	Derivative Gain at Idle state	0~16.0	2.0	停止時微分控制器增益
KI	Integral Gain	0~2.0	0	積分控制器增益
KII	Integral Gain at Idle state	0~2.0	0	停止時積分控制器增益
FFA	Feed Forward Accelerate	0~32767	8	加速度前授參數
FFV	Feed Forward Velocity	0~32767	80	速度前授參數
KB		0~255	100	馬達啟動電流限制
KM		0~255	100	馬達運轉電流限制
AJ1	Adjust Jog input 1	0~1023	0	
AJ2	Adjust Jog input 2	0~1023	1023	
AJ3	Adjust Jog input 3	0~1023	0	
EC	ECho	0~1	開機時，由指撥開關 3 設定	
ARSV	Auto RSV	0~1	0	開機自動執行 RSV 指令
HFP	Hold oFf Polarity	0~1	0	
SNP	SNap off Polarity	0~1	0	
CRP	CRash signal Polarity	0~1	0	
PHZ	PHase Z signal	0~1	0	
QEP	Quadrature Encoder Pulse	0~1	0	

3-3. 各系統參數詳細說明

參數名稱： AJ1,AJ2,AJ3
英文助憶： Adjust Jog Input
功能： 設定搖桿（Jog）輸入的修正值（Offset）
設定範圍： 0~1023
預設值： 0
功能說明：

參數名稱： ARSV
英文助憶： Auto RSV
功能： 設定開機後自動執行 RSV 指令
設定範圍： 0~1
預設值： 0
功能說明：

參數名稱： CLP，CL+

CLM，CL-

英文助憶： Clip Plus

Clip Minus

參數意義： 設定運動位置極限

CLP(或 CL+)，決定往座標正方向移動的位置極限，

CLM(或 CL-)，決定往座標負方向移動的位置極限。

設定範圍：

預設值： CLP=+5000

CLM=-5000

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，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： CRP

英文助憶： Crash signal Polarity

功能： 設定在驅動器模式時，跳機（Crash）輸出信號的極性

設定範圍： 0~1

預設值： 0

功能說明：

CRP=1，設定平常輸出埠 2 致能（Enable）。當系統跳機時，輸出埠 2 除能（Disable）。

CRP=0，設定平常輸出埠 2 除能。設定當系統跳機時，輸出埠 2 致能。

所謂出致能（Enable），是指輸出埠的光耦合器有電流作動，也就是指輸出埠的兩支接頭為通路狀態（請參考 7.4 節輸出電路）。

在驅動器模式時，CSBL600 的輸入埠 2 被用做輸出系統已發生跳機（Crash）的信號。當發生跳機時，輸入埠 2 的狀態就由本參數設定。

在控制器模式時，本參數可被忽略。

參數名稱： EC
英文助憶： Echo
功能： 設定是否開啓 RS232 埠的回應功能
設定範圍： 0~1
預設值： 開機時，由指撥開關 3 設定
功能說明：

EC=1，開啓 RS232 埠的回應功能。

EC=0，關閉 RS232 埠的回應功能。

CSBL600 若是使用模擬終端機的軟體與 PC 連線，就需要開啓 RS232 埠的回應功能。此時 CSBL600 在 RS232 收到任何字元，都會先複製回送。

參數名稱： ECL

英文助憶： Enable Clip

功能： 啓動或關閉軟體極限保護

設定範圍： 0~1

預設值： 0〈關閉〉

功能說明：

ECL=1，開啓軟體極限開關功能。

ECL=0，關閉軟體極限開關功能。

參數名稱： ECR
英文助憶： Encoder Resolution
功能： 輸入編碼器的解析度
設定範圍： 1~65535
預設值： 無
功能說明：

CSBL600 是以全閉迴路(close loop)的方式控制無刷伺服馬達。所以馬達一定要搭配編碼器(Encoder)，提供位置迴授，才可以使用。

本參數就是用以輸入編碼器的解析度，使用的單位是每轉一轉編碼器的輸出計數 (Counts/Rev)。

本參數不受“DF” 指令 (Default) 影響。

參數名稱： EDI
英文助憶： Encoder Direction
功能： 設定編碼器（Encoder）計數的方向
設定範圍： 0~1
預設值： 1
功能說明：

EDI 參數可設定編碼器計數的方向。所謂編碼器計數方向，是指馬達在指定方向旋轉時，編碼器計數是 上數或往下數。

CSBL600 在應用時，編碼器的計數方向必須和系統的座標方向一致，否則回授位置相反，馬達就不可能正常運作。若發生這種狀況，有很多方法可以解決，包括更改座標方向、對調馬達配線。而其中最簡單的修正方法，就是更改 EDI 參數。將原來是 1 的改爲 0，原來是 0 的改爲 1 即可。

參數名稱： EHC
英文助憶： Enable Hard Clip
功能： 設定是否開啓硬体極限開關功能
設定範圍： 0~1
預設值： 0
功能說明：

EHC=1，開啓硬体極限開關功能。

EHC=0，關閉硬体極限開關功能。

CSBL600 的輸入埠 6 及 7 除當一般輸入埠使用外，也可以當做左右極限開關的信號輸入口。若是要做這種用途，就必須將本參數設為 1。否則就應將本參數設為 0。

參數名稱： EL
英文助憶： Error Limit
參數意義： 設定系統可容忍誤差的極限，做為跳機（Crash）判定的基準。
設定範圍： 0-65536
預設值： 200
功能說明：

本參數的功能主要是做為一道安全保護。CSBL600 控制無刷伺服馬達，是使用全閉迴路控制。當誤差發生時，CSBL600 就會自動產生修正機制。一般而言，CSBL60 對馬達的輸出會與誤差成比例，誤差愈大，CSB600 為修正誤差的輸出也會愈大。但在實務上，為了安全考量，CSBL600 的輸出不能無限放大。所以當誤差超過某一極限時，就必須放棄自動修正機制，關閉對馬達的所有功率輸出，讓馬達緊急停俾，這就是所謂“跳機”（Crash）。

EL 參數就是用來設定 CSBL600 可容忍誤差的極限，當馬達編碼器的回授位置（Actual position）與命令位置（Command position）之間的差值超過 EL 參數時，系統即立刻跳機，並關閉所有輸出，避免失控造成危險。

EL 參數所用的單位，是 CSBL600 內部所用的基本單位，也就是編碼器的回授計數(Encoder Counts)。例：若使用每轉 2000 計數的編碼器，EL=200，實際的意義就是當馬達位置的誤差超過 200 Counts，相當於 1/10 轉時，系統即立刻跳機。

EL 值不宜設的太大，否則失去安全保護的意義。但也不能設的太小，EL 太小時系統會很敏感，速度或負載稍有變化時即容易發生跳機。

參數名稱： ELH
英文助憶： Error Limit at Home process
參數意義： 設定在執行回原點時，系統可容忍誤差的極限。
設定範圍： 0-65536
預設值： 200
功能說明：

本參數的意義及用法，與 EL 參數類似，但用途不同。HEL 參數主要是用在要以跳機（Crash）狀況做為原點位置感測的依據時。

在一未裝原點位置感測裝置(Origin Sensor)的系統，若要執行原點定位動作。有一種可行的方法，就是故意去碰撞機構極限，造成跳機。以發生跳機時的位置做為原點定位的依據（將 HM 參數的位元 1 設為 1，請參考 HM 參數說明）。在這種狀況下，就必須將跳機的敏感度提高，以免造成過重的撞擊。使用 ELH 參數，可以單獨設定在做回原點動作時可容忍誤差的極限，將其與一般運動時的容忍誤差分開設定。目的是要將其設定為較小的數值，以提高跳機的敏感度。

參數名稱： ELI
英文助憶： Error Limit Integral
參數意義： 設定系統可容忍誤差累積的極限。
設定範圍： 0-65536
預設值： 8000
功能說明：

本參數與 EL 參數類似，也是以“跳機”（Crash）方式做為系統的安全保護。但 EL 參數只是以當下的誤差的絕對值為觸發跳機（Crash）的依據，並不考慮時間累積因素。本參數則包含時間累積因素，當誤差累積超過本參數所設定的極限時，一樣會觸發跳機。

當馬達命令位置與實際位置(由 Encoder 回授取得)有誤差時，若系統不能立即修正，即使此誤差小於 EL 所設定的極限，仍會因誤差被累積，超過 ELI，而觸發跳機。

參數名稱： ELP
英文助憶： Position Error Limit
參數意義： 設定系統可容忍的到位誤差。
設定範圍： 0-65536
預設值： 40
功能說明：

參數名稱： FFA
英文助憶： Feed Forward Accelerate
參數意義： 加速度前授
設定範圍： 0-65535
預設值： 8
功能說明：

參數名稱： FFV
英文助憶： Feed Forward Velocity
參數意義： 速度前授
設定範圍： 0-65535
預設值： 80
功能說明：

參數名稱： HC
英文助憶： Home Coordinate
功能： 設定原點的座標
設定範圍： -32767.9999~32767.9999
預設值： 0
功能說明：

HC 設定機械原點座標。

在執行完回原點動作後，馬達或被馬達帶動的滑台應準確停在原點感測開關（Home sensor）作動的位置。這個位置是座標系統的參考點，一般被稱作機械原點，但機械原點並不一定就是座標的原點。也就是說機械原點的座標並不一定要為零，而是可以由 HC 參數設定為任意值。藉由這項功能，在組裝系統時會更方便，原點感測開關並不一定要被裝在座標系統的原點位置（滑台的最右側或最左側），可以被裝在任何位置。實際的應用方式是，先將原點感測開關固定後，再以 HC 參數來調整座標系統的原點位置。

若 HM 參數的位元 4 被設定成 1，則在執行完回原點動作後，座標值並不會被更動（請參考 HM 系統參數說明）。在這種狀況下，HC 參數就不會有作用。

參數名稱： HFP

英文助憶： Hold off polarity

功能： 設定在驅動器模式時，Hold Off 信號的極性

設定範圍： 0~1

預設值： 0

功能說明：

HFP=1，設定輸入埠 3 沒有輸入時，關閉馬達電流。

HFP=0，設定輸入埠 3 有輸入時，關閉馬達電流。

在驅動器模式時，CSBL600 的輸入埠 3 被用做暫時關閉馬達電流(Hold Off)信號的輸入埠。這信號的極性（有輸入時動作或沒有輸入時動作），就是由本參數設定。

在控制器模式時，本參數可被忽略。

參數名稱： HM
英文助憶： Home Mode
功能： 回原點模式
設定範圍： 0-63
預設值： 6
功能說明：

CSBL600 在執行回原點指令時，其動作模式是依 HM 參數二進位各位元決定。

- | | |
|------|--|
| 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。 |
| 位元 5 | 0：回原點的動作分兩段，先進原點，再出原點。
1：省略進原點動作。 |
| 位元 6 | 0：一般模式。
1：省略出原點動作。 |

CSBL600 回原點的方式，提供了使用者非常多的彈性，可以適應各種不同機台的狀況。舉凡回原點的方向選擇，機台上是否已備有原點感應開關，開關為常開或常閉型式等，都可以各別對應。

CSBL600 關於回原點的控制，除了本參數外。另外還有 HN 選擇原點感應開關的輸入埠，HC 設定原點的座標等，請參考這些系統參數的使用說明。

若 HM 參數的位元 4 被設定成 1，則在執行完回原點動作後，座標值並不會被更動。在這種狀況下，HC 參數就不會有作用。一般而言，這項功能是被用在量測原點感應開關的位置用。若一系統有兩個位置感測開關，這項功能可被用來量測這兩個開關的距離。

若 HM 參數的位元 5 被設定成 1，則在執行回原點時，會省略進原點動作。這項功能一般是被用在以編碼器 Index 信號為原點時。

若一系統，既無原點感應開關，又無編碼器 Index 信號。要定義原點，有一種方法，就是故意去碰撞機構極限，造成跳機，做為原點位置的依據。這時需將 HM 參數的位元 1 與位元 6 同時設成 1。

參數名稱： HN
英文助憶： Home sensor No.
功能： 設定原點感測開關信號的輸入埠
設定範圍： 1-8
預設值： 5
功能說明：

HN 設定回原點時，原點感測開關信號的輸入埠。

CSBL-600 共有 8 個供一般用途的輸入埠，每一個輸入埠都可以用做原點信號的輸入口。這項功能不只是提供配 的方便性，更可以藉此架構多原點的定位系統。在同一機台上，不同的應用場合，可以有不同的原點。

若 HM 參數的位元 1 及位元 2 被設定成 1 時，CSBL600 做回點動作時，並不需要原點感測開關（請參考 HM 系統參數的說明）。在這種場合，本參數可被忽略。

參數名稱： KD，KDI

英文助憶： Derivative Gain

Derivative Gainat Idle state

參數意義： 設定伺服馬達微分控制器增益。

KD 為馬達運動時的微分控制器增益，

KDI 為馬達靜止時的微分控制器增益。

設定範圍： 0-16.0

預設值： KD=2

KDI=1

功能說明：

CSBL600 控制無刷伺服馬達，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

$$PWMd = \frac{dErr}{dt} \times KD$$

其中，PWMd = 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。所以微分控制器可以提昇系統對瞬間誤差變動時的反應。

微分控制器可以增加系統的穩定性，一般的應用均將微分控制器增益 KD 設為大於比例控制器增益 KP 很多。但 KD 設的太大也會造成高頻振動或噪音等不穩定現象。

CSBL600 的微分控制器，馬達運動狀態的增益與馬達靜止時的增益，可以分開設定。KD 為馬達運動時的微分控制器增益，KDI 則為馬達靜止時的微分控制器增益，可以設定為不同的數值。

參數名稱： KI，KI I
英文助憶： Integral Gain
Integral Gainat Idle state
參數意義： 設定伺服馬達積分控制器增益。
KI 為馬達運動時的積分控制器增益，
KI I 為馬達靜止時的積分控制器增益。
設定範圍： 0-2.0
預設值： 0
功能說明：

CSBL600 控制無刷伺服馬達，是使用全閉迴路控制。所運用的控制法則〈Control alarithrum〉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 KI 就是兩者的比值。以公式表示，其關係如下

$$PWMi = \int Err dt \times KI$$

其中， PWMi = 積分控制器的輸出。
Err = 位置誤差
KI = 積分控制器增益

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

$$\frac{dPWMi}{dt} = KI \times Err$$

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 (Steady state error)，但很容易由於積分，控制變得比較不穩定，所以若穩態誤差問題不是很嚴重時，KI 可設成零。

CSBL600 的積分控制器，馬達運動狀態的增益與馬達靜止時的增益，可以分開設定。KI 為馬達運動時的積分控制器增益，KI I 則為馬達靜止時的積分控制器增益，可以設定為不同的數值。

參數名稱： KP，KPI
英文助憶： Proportional Gain
Proportional Gain at Idle state
參數意義： 設定伺服馬達比例控制器增益。
KP 為馬達運動時的比例控制器增益，
KPI 為馬達靜止時的比例控制器增益。
設定範圍： 0-4.0
預設值： KP=1
KPI=1
功能說明：

CSBL600 控制無刷伺服馬達，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

$$PWMp = KP \times ERR$$

其中，PWMp = 比例控制器的輸出。

Err = 位置誤差

KP = 比例控制器增益

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

CSBL-600 對比例控制器，馬達運動狀態的增益與馬達靜止時的增益，可以分開設定。KP 為馬達運動時的比例控制器增益，KPI 則為馬達靜止時的比例控制器增益，可以設定為不同的數值。一般應用，多將 KPI 設小一點，可以避免靜止時發生不穩定的振動。

參數名稱： MDI
英文助憶： Motor Direction
功能： 設定馬達旋轉的方向
設定範圍： 0~1
預設值： 1
功能說明：

MDI 參數可設定馬達旋轉方向與座標方向的關係。

CSBL600 在應用時，馬達旋轉的方向必須和系統的座標方向一致，否則應用時會很困擾。若發生這種狀況，可更改 MDI 參數。將原來是 1 的改為 0，原來是 0 的改為 1 即可解決。

參數名稱： PHO
英文助憶： Phase Origin
參數意義： 馬達換向參考原點
設定範圍： 1~65535
預設值： 無
功能說明：

設定搭配 CSBL600 使用的馬達的換向參考原點。

本參數不受“DF”指令（Default）影響。

參數名稱： PHZ

英文助憶： Phase Z signal

功能： 設定是否由輸出埠 4 輸出 Z 相位信號

設定範圍： 0~1

預設值： 0

功能說明：

PHZ=1，設定由輸出埠 4 輸出 Z 相位信號。此時輸出埠 4 即無法再做一般輸出用。

PHZ=0，設定輸出埠 4 為一般用途。

參數名稱： POL
英文助憶： Pole
參數意義： 馬達磁極對數
設定範圍： 1~255
預設值： 無
功能說明：

設定搭配 CSBL600 使用的馬達的磁極對數。(注意：POL=4 是指馬達的轉子有 4 個 N 極及 4 個 S 極，有些廠商會稱這種馬達為 8 極馬達，也就是說馬達極數是本參數的兩倍)

本參數不受“DF”指令 (Default) 影響。

參數名稱： QEP

英文助憶： Quadrature Encoder Pulse

功能： 設定在驅動器模式時，使用編碼器或手搖輪為馬達控制信號

設定範圍： 0~1

預設值： 0

功能說明：

CSBL600 在驅動器模式時，馬達的運動完全由輸入埠 1 及輸入埠 2 的輸入信號控制。在這種模式下，是否要使用編碼器的信號型式，由本參數設定。

QEP=1，輸入埠 1 及輸入埠 2 的輸入信號為“編碼器型式的信號”。

QEP=0，輸入埠 1 及輸入埠 2 的輸入信號為“脈波及方向”或“正轉脈波及逆轉脈波”，由指撥開關 SW2 設定。（請參考 2.2 節有關指撥開關設定的說明）

在控制器模式，本參數可被忽略。

參數名稱： RSC1
RSC2

英文助憶：

參數意義： 在驅動器模式時，每輸入一脈波，馬達所要走的步數。

設定範圍： 1-32767

預設值： RSC1=1

RSC2=1

功能說明：

CSBL600 在當驅動器使用時，馬達轉動的速度完全由輸入脈波控制。每輸入一脈波，馬達要轉多少步，即由 RSC 參數設定。RSC1 為倍數，RSC2 為除數。例如 RSC1=3，RSC2=2，則每輸入一個脈波，馬達要走 1.5 步。

此處所謂”步”，即 CSBL600 內部的距離計算單位，也就是編碼器回授的最小計數單位 (Encoder Count)，所以馬達只能走整數步。以前例而言，依計算馬達應走 1.5 步，但實際上只會先走 1 步，而在下一個輸入脈波時，再一次走兩步，但這種捨位不會有累積，。

在控制器模式時，本參數可被忽略。

參數名稱： SC1
SC2
英文助憶： Scale
參數意義： 每單位距離所相對的編碼器計數 (Encoder Counts)。
設定範圍： 1-32767
預設值： SC1=10
SC2=2000

功能說明：

CSBL600 內部的距離計算單位，是編碼器回授的計數(Encoder Counts)。這樣的單位，與使用者所慣用的距離單位〈如 mm 或 inch 等〉未必一致，甚至不是整數比關係。要解決這項問題，使用者可先以 SC 參數來訂定自己所習慣使用的單位與編碼器計數的比例關係。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。

對一般應用，SC 可以下列方式的訂定，

SC1 = 馬達每轉進給距離

SC2 = 編碼器解析度

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。

編碼器解析度：為馬達上所附的編碼器(Encoder)，每轉所回授的步數。

使用例：

無刷伺服馬達，編碼器解析度為每轉 8000 步。驅動使用螺距為 10mm 的滾珠螺桿。則

SC1=10

SC2=8000。

參數名稱： SNP
英文助憶： S Nap off polarity
功能： 設定在驅動器模式時，緊急關機信號的極性
設定範圍： 0~1
預設值： 0
功能說明：

SNP=1，設定輸入埠 5 沒有輸入時，緊急關閉馬達電流。

SNP=0，設定輸入埠 5 有輸入時，緊急關閉馬達電流。

在驅動器模式時，CSBL600 的輸入埠 5 被用做緊急關閉馬達電流(Snap Off)信號的輸入埠。這項信號的極性（有輸入時動作或沒有輸入時動作），就是由本參數設定。

在一般應用，緊急關閉馬達電流多被用做系統保護，例如輸入埠 5 可外接一溫控 SW，當溫度過高時即關閉馬達電流。而本參數就是用於設定這種溫控 SW 為常開型（Normal open）或常閉型（Normal close）。

在控制器模式時，本參數可被忽略。

參數名稱： TL
英文助憶： Torque Limit
參數意義： 設定馬達扭力輸出極限
設定範圍： 1~255
預設值： 255
功能說明：

本參數可設定馬達扭力輸出的極限，實際也等於 CSBL600 輸出的電流極限。

一般要做馬達扭力控制時，本參數應設小一點（1~10）。否則扭力控制的效果並不明顯。

在馬達負荷較重的場合，本參數也可用做限制 CSBL600 輸出電流，做為一種系統安全的保護。但本參數若設得太低，也會影響馬達啓動及加速性能。在一般情形，使用預設值 200 是安全的。

參數名稱： VA
英文助憶： Velocity Accelerate
參數意義： 設定加速度
設定範圍： 0-32767.9999
預設值： 500
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 CSBL-600 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。

CSB-600 處理所有的運動指令的執行方式均為如下述模式：

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，甚至可能會引發散性的震動。

使用例：

參數名稱： VB
英文助憶： Velocity Base
參數意義： 設定起始速度
設定範圍： 0-32767.9999
預設值： 5
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。

CSBL600 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。

有關本參數應用單位的說明，請參照 VF 系統參數的功能說明。

使用例：

參數名稱： VF
英文助憶： Velocity Fast
參數意義： 設定高度
設定範圍： 0-32767.9999
預設值： 500
功能說明：

CSBL600 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。

所有的速度參數，其單位為距離除以時間。若 SC 參數已訂定，使用者應可將距離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 BLDC-01 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。

使用例：

參數名稱： VH
英文助憶： Velocity Home
參數意義： 設定回原點動作時的速度
設定範圍： 0-32767.9999
預設值： 50
功能說明：

CSBL600 關於速度設定，除了一般運動指令所用的 VF 參數（設定高速用）及 VM 參數（設定工作速度用）外，在執行回原點指令時，還有一專用的 VH 參數，用來設定搜尋原點位置時的速度。

所有的速度參數，其單位為距離除以時間。若 SC 參數已訂定，使用者應可將距離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 CSBL-600 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。

使用例：

參數名稱： VJ
英文助憶： Velocity Jog
參數意義： 設定執行 JOG 指令時的速度
設定範圍： 0-32767.9999
預設值： 200
功能說明：

CSBL600 關於速度設定，除了一般運動指令所用的 VF 參數（設定高速用）及 VM 參數（設定工作速度用）外，在執行 JOG 指令時，還有一專用的 VJ 參數，用來設定連續運動時的速度。

所有的速度參數，其單位為距離除以時間。若 SC 參數已訂定，使用者應可將距離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 CSBL-600 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。

使用例：

參數名稱： VL
英文助憶： Voltage Limit
參數意義： 設定 CSBL 的電壓極限
設定範圍： 0-255
預設值： 開機後，依據偵測到的電源電壓，自動設定。
功能說明：

當馬達急劇減速時，因能量回饋，會使 CSBL600 內部的電壓徒昇，有時甚至會造成破壞。為保護系統，當能量回饋造成的電壓上升超過極限時，CSBL600 就會自動打開煞車電路，讓能量轉消耗到煞車電阻。本參數就是用來設定此電壓上升極限。

本參數若設得太低，會使煞車電路誤動作。所以在系統開機時，CSBL600 會依據偵測到的電源電壓，自動設定。使用者在一般應用，並無需再做修改或設定。

參數名稱： VM
英文助憶： Velocity Move
參數意義： 設定工作速度
設定範圍： 0-32767.9999
預設值： 200
功能說明：

CSBL600 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，其值即是由本參數設定。

所有的速度參數，其單位為距離除以時間。若 SC 參數已訂定，使用者應可將距離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 BLDC-01 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。

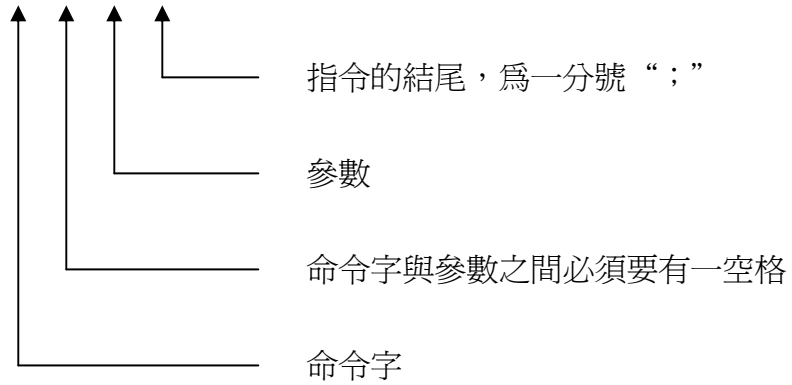
使用例：

四、指令

4.1 指令語法說明

CSBL600 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“；”。

例： MA 1000 ；



4.2 CSBL600 指令一覽表

CSBL600 指令一覽表

運動指令

FA x	(X Fast move Absolutely)	高速移動至絕對座標 x 處。
FR x	(X Fast move Relatively)	高速移動至相對座標 x 處。
MA x	(X Move Absolutely)	以工作速度移動至絕對座標 x 處。
MR x	(X Move Relatively)	以工作速度移動至相對座標 x 處。
PZ	(Pause)	暫停。
REDO	(Redo)	繼續未完成動作。
STOP	(Emergency Stop)	緊急停止

JOG 運動指令

JGF	(Jog Forward)	馬達持續正轉。
JGR	(Jog Reverse)	馬達持續反轉。
JGA	(Jog by Analog Input)	馬達由輸入電壓控制旋轉。
JG0	(Jog stop)	結束 JOG 運動，馬達減速停止。

回原點指令

H m, n	(X Home)	回原點。m 為回原點模式，n 為做為原點信號的輸入埠($1 \leq n \leq 8$)。
--------	----------	--

座標設定指令

CS x	(Coordinate Set)	設定現在位置座標。
------	------------------	-----------

馬達控制指令

HON	(Hold On)	啓動控制
HOFF	(Hold Off)	暫停馬達控制
RSV	(Reset Space Vector)	重置馬達相位
RESET	(Reset)	重置馬達控制
FT	(Fix Torque)	設定馬達輸出固定扭力

輸出介面控制指令

SET Pn	(Set Port)	設定第 n 輸出埠為 On ($1 \leq n \leq 4$)
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 4 個輸出埠的狀態。 ($0 \leq n \leq 15$)
ISSET Pn (Immediately Set Port)	立即設定第 n 輸出埠為 On ($1 \leq n \leq 4$)
ICLR Pn (Immediately Clear Port)	立即清除第 n 輸出埠為 Off ($1 \leq n \leq 4$)
PWM Pn,duty,f (PWM Port)	由第 n 輸出埠輸出 PWM 脈波($n=1,3$)

運動同步輸出控制指令

M1SET Pn,x	在移動到座標 x 時，設定 Pn 輸出埠為 On。
M2SET Pn,x	($1 \leq n \leq 4$)
M1CLR Pn,x	在移動到座標 x 時，清除 Pn 輸出埠為 Off。
M2CLR Pn,x	($1 \leq n \leq 4$)
M1RP Pn,x,d1,d2,n	在移動到座標 x 時，連續改變 Pn 輸出埠狀態。
M2RP Pn,x,d1,d2,n	
CLR M1	清除之前設定的同步輸出指令
CLR M2	

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。
SG addr (Step Go)	程式無條件跳躍至指定位址。
JP addr (Jump)	當指定的輸入埠為 On 時，程式跳躍至指定位址。 ($1 \leq i \leq 8$)
JJ i,addr	當指定的輸入埠為 Off 時，程式跳躍至指定位址。 ($1 \leq i \leq 8$)
JNI i,addr	當指定的變數為 0 時，程式跳躍至指定位址。
JZ #int,add	當指定的變數不為 0 時，程式跳躍至指定位址。
JNZ #int,addr	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。
JE c,#var,addr	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。
JNE c,#var,addr	當變數#var 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。
JG c,#var,addr	當變數#var 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。
JNG c,#var,addr	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
JTI i,tm,addr	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。
JNTI i,tm,addr	

JS s,addr		當指定的指撥開關為 On 時，程式跳躍至指定位址。 (1 ≤ s ≤ 6)
JNS s,addr		當指定的指撥開關為 Off 時，程式跳躍至指定位址。 (1 ≤ s ≤ 6)
CALL addr	(Call)	呼叫副程式
RET	(Return)	由副程式返回。若原來已在程式最上層程式，則結束程式執行
WI i	(Wait Input)	若輸入埠 i 為 Off，則本指令之後的馬達運動指令會先做好準備，但暫不執行。等待輸入埠 i 轉為 On 的觸發信號才執行
WNI i	(Wait No Input)	若輸入埠 i 為 On，則本指令之後的馬達運動指令會先做好準備，但暫不執行。等待輸入埠 i 轉為 Off 的觸發信號才執行
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
CLR BUF	(Clear Buffer)	清除未執行指令
SET EV	(Set Event)	定義中斷事件觸發條件
On EV	(On Event)	設定中斷事件副程式位址
CLR EV	(Clear Event)	清除之前設定的中斷事件

速度控制指令

AVM	(Analog set VM)	由類比輸入控制馬達轉速
AVJ	(Analog set VJ)	由類比輸入控制馬達 JOG 轉速

程式管理指令

PG	(Program Generate)	進入程式編輯模式
PA	(Program Apend)	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
LD P	(Load Program)	取用存於 Flash Rom 中的程式

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

資料庫管理指令

DATA idx,data1,data2...	儲存資料 r 至 idx 指定的資料庫位址 ($0 \leq \text{idx} \leq 2048$)
@idx=data	
SAVE D	(Save Data) 儲存全部資料庫資料至 Flash Rom
LD D	(Load Data) 取用存於 Flash Rom 中的資料庫資料

變數指令

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
CLR VAR	清除全部變數
CLR #var	清除單一變數

系統參數管理指令

參數名稱=data	設定系統參數
DF	(Default) 將全部系統參數重置成預設值
SAVE C	(Save Config) 儲存全部系統參數至 Flash Rom
LD C	(Load Config) 取用存於 Flash Rom 中的系統參數

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

?IN n	(Input) 讀取輸入埠的狀態 ($1 \leq n \leq 8$)
?AIN n	(Analog Input) 讀取類比輸入埠的狀態 ($1 \leq n \leq 3$)
?SW s	(Analog Input) 讀取指撥開關的狀態 ($1 \leq s \leq 6$)
?ST	(Status) 讀取系統狀況

?ID	(Identify Number)	讀取 ID
?VER	(Version Number)	讀取軟體編號
?DUTY		讀取負載狀況
?RPM		讀取馬達轉速
?FLE	(Follow Error)	讀取現在位置誤差
?AC		讀取現在電源電壓
?@idx		讀取 idx 位址的資料庫資料
?#var		讀取變數
?參數		
?數學函數		

4.3 指令詳細說明

指令： AVJ (Set VJby Analog Input)

語法： AVJ n;

參數範圍： $1 \leq n \leq 3$

模式： 立即，程式

功能： 由類比輸入控制馬達 JOG 轉速

說明：

範例：

指令： AVM (Set VM by Analog Input)

語法： AVM n;

參數範圍： $1 \leq n \leq 3$

模式： 立即，程式

功能： 由類比輸入控制馬達轉速

說明：

範例：

指令： CALL (Call subroutine)

語法： CALL @addr;

參數範圍：

模式： 程式

功能： 呼叫副程式

說明：

範例：

指令： CHG P (Change output)
語法： CHG Pn;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 改變輸出埠為開關(On/Off)狀態。
說明：

CSBL-600 控制器共有 4 個供一般用途的輸出埠，每一輸出埠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”。

指令： CLR BUF (Clear Buffer)

語法： CLR BUF ;

參數範圍：

模式： 立即，程式

功能：

說明：

範例：

指令： CLR P (CLear Port)
語法： CLR Pn;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 設定輸出埠為關(Off)狀態。
說明：

CSBL-600 控制器共有 4 個供一般用途的輸出埠，每一輸出埠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「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： CLR VAR (Clear Variable)

語法： CLR VAR;
CLR #var;

參數說明：

參數範圍：

模式： 程式，立即

功能： 清除變數

說明：

本指令用以清除變數。所謂清除，即釋出變數所佔用的記憶體空間及名稱。

CLR #var 只清除單一變數。CLR VAR 指令則用以同時清除所有之前宣告過的變數。

範例：

指令： CS (Coordinate Set)
語法： CS x;
參數範圍： $-32,767.9999 < x < +32,767.9999$
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 定為現在位置的座標。若參數 x=0，執行本指令，即將現在位置設為座標原點。

座標系統單位距離，可由系統參數 SC 設定。

範例：

指令： DATA (Set DATA)
語法： DATA idx, r;
參數範圍： $0 \leq \text{idx} \leq 2048$
模式： 立即，程式
功能：
說明： 儲存資料 r 至 idx 指定的資料庫位址

範例：

指令： DF (Set Default)
語法： DF ;
參數範圍：
模式： 立即，程式
功能： 將所有系統參數重置成預設值
說明：

範例：

指令： DL (Down Load program)
語法： DL;
參數範圍：
模式： 立即
功能： 經由 RS232 埠，由 PC 下載程式
說明：

範例：

指令： DN (Done)

語法： DN ;

參數範圍：

模式： 程式

功能： 確定先前所下指令已全部執行完畢再繼續

說明：

範例：

指令： FA (Fast move Absolute)

語法： FA x；

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

參數 x 另受到系統參數 CLP(座標正極限)及 CLM(座標負極限)所設定的範圍限制。

模式： 立即，程式

功能： 快速移動被馬達帶動的滑座至指定位置，此位置由絕對座標設定。

說明：

CSBL-600 將速度分為高速及工作速度兩種模式。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。而在切削時，則應使用工作速度模式。

本指令的功能就是以系統設定的高速，移動被馬達帶動的滑座至指定位置。系統的高速，是由系統參數 VF 所設定（請參考系統參數 VF 的說明）。

x 參數即為終點位置的絕對座標。此座標可以有正負號，也可以使用小數表示。座標所用的單位是由系統參數 SC 所設定（請參考系統參數 SC 的說明）。

執行本指令後，會將系統的座標模式，設成絕對座標模式。以後只要執行“F”指令，就與“FA”指令有同樣的功能。

本指令與 MA 指令類似，只是使用的速度不同。本指令也與 FR 指令類似，只是使用的座標模式不同。

範例：

指令： FR (Fast move Relative)

語法： FR x；

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

參數 x 另受到系統參數 CLP(座標正極限)及 CLM(座標負極限)所設定的範圍限制。

模式： 立即，程式

功能： 快速移動被馬達帶動的滑座至指定位置，此位置由相對於現在位置的座標設定。

說明：

CSBL-600 將速度分為高速及工作速度兩種模式。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。而在切削時，則應使用工作速度模式。

本指令的功能就是以系統設定的高速，移動被馬達帶動的滑座至指定位置。系統的高速，是由系統參數 VF 所設定（請參考系統參數 VF 的說明）。

x 參數即為終點位置相對於現在位置的座標。此座標可以有正負號，也可以使用小數表示。座標所用的單位是由系統參數 SC 所設定（請參考系統參數 SC 的說明）。

執行本指令後，會將系統的座標模式，設成相對座標模式。以後只要執行“F”指令，就與“FR”指令有同樣的功能。

本指令與 MR 指令類似，只是使用的速度不同。本指令也與 FA 指令類似，只是使用的座標模式不同。

範例：

指令： FT (Fix Torque)

語法： FT n;

FT 0;

參數範圍：

模式： 立即，程式

功能： 使馬達輸出固定的扭力

說明：

範例：

指令： G (Go)
語法： G;
G @addr;
G \$label;
G Ln;
參數說明： @addr 直接位址
\$label 標籤位址
Ln 行號位址
參數範圍： 0 <addr < 8192
模式： 立即
功能： 開始執行程式
說明：

使用者必須先編輯好所要執行的程式，然後執行本指令”G”，CSBL-600 即立即進入程式模式，從位址參數指定的位址開始執行程式。若無位址參數，程式將從位址 0 開始執行。

進入程式模式後，CSBL-600 即依程式命令，逐行執行。直至執行到”RET”指令，或全部程式均已執行完畢，才會結束程式模式，回到立即命令模式。

本指令可以附加位址參數。讓使用者可以選擇只執行程式中的某一段副程式，並不一定要執行全部程式。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

相關指令：

指令： H (Home)
語法： H m, n;
模式： 立即，程式
功能： 尋找機械原點。
說明：

一般尋找機械原點的目的，是爲了要定義軟體的座標，使其能與系統的機構位置配合。所謂機械原點，是指在機構上的一個可以感測且固定的位置點，一般就是利用這固定的位置來定義座標系統的原點。取得機械原點的方法很多，但基本上應是一滑台到位的感測器裝置，當滑台移到感測器的位置時，能送出一信號。若沒有到位感測器裝置，也可以利用碰到機構極限的位置做爲機械原點。

爲了提高對機械原點感測的精度，CSBL-600 將尋找機械原點的程序分成進原點與出原點兩段進行。所謂進原點就是先是以一比較高的速度（此速度由系統參數 VH 設定），將滑台往機械原點的方向移動，待收到原點到位信號後即減速停止。然後再進行出原點的動作，就是以一比較慢的速度（此速度由系統參數 VH 設定）往離開原點的方向移動，待原點到位信號結束即立即停止，此時的位置即機械原點。

以上說明僅是就一般系統的狀況說明尋找機械原點的程序。實務上，可能每一使用者的應用狀況都不一樣。CSBL600 爲了因應各種不同的應用狀況，設有很多系統參數，來規畫及設定尋找機械原點動作的詳細程序。這些系統參數包括 HM、HN、HC、VH、HEL 等，請參考這些參數的使用說明。

本指令可以有兩個參數 m、n，也可以不附參數。若附有 m、n 參數，其功能就是系統參數 HM 與 HN。若未附參數，執行“H”指令，就會以系統參數 HM 及 HN 來代替 m、n 參數。

範例：

指令： HOFF (Hold Off)
語法： HOFF ;
模式： 立即，程式
功能： 暫停馬達控制
說明：

指令： HON (Hold ON)
語法： HON;
模式： 立即，程式
功能： 重新啓動馬達控制
說明：

指令： ICLR P (Immediately Clear Port)
語法： ICLR Pn;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 立即設定輸出埠為關(Off)狀態。
說明：

CSBL-600 控制器共有 4 個供一般用途的輸出埠，每一輸出埠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 指定的輸出埠立即設定為「關」的狀態，也就是將其輸出光偶合器的電晶體設定為 Off 的狀態。

本指令與”CLR P”指令類似，但”CLR P”是按順序執行的指令，必須等在其之前下達的指令均已執行完畢，才會被執行。而本指令則是要立即執行的指令，不論在其之前是否還有其它未執行完畢的指令，本指令都會立即先執行。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

範例：

指令： ISET P (Immediately Set Port)
語法： ISET Pn;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 立即設定輸出埠為開(On)狀態。
說明：

CSBL-600 控制器共有 4 個供一般用途的輸出埠，每一輸出埠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 指定的輸出埠立即設定為「開」的狀態，也就是將其輸出光偶合器的電晶體設定為 On 的狀態。

本指令與”SET P”指令類似，但”CLR P”是按順序執行的指令，必須等在其之前下達的指令均已執行完畢，才會被執行。而本指令則是要立即執行的指令，不論在其之前是否還有其它未執行完畢的指令，本指令都會立即先執行。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

範例：

指令： JE (Jump on Equal)

語法： JE #var, c, @addr;

JE #var, c, \$label;

JE #var, c, Ln;

參數說明： #var 變數

c 可為常數，變數或函數，但必須與#var 同類型

@addr 直接位址

\$label 址標位址

Ln 行號位址

參數範圍： 0 < addr < 8192

模式： 程式

功能： 當變數#var 等於 c 時，程式跳躍至指定位址。

說明：

本指令依變數是否等於一設定值，決定程式流程方向。當變數#var 等於 c 時，程式跳躍至指定位址。否則即繼續執行下一指令。其中變數#var 可為整數變數，也可為實數變數。c 可為常數，變數或函數，但必須與#var 同類型。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JNE 指令類似，僅判定的狀態條件相反。

相關指令：

範例：

指令： JG (Jump on Greater than)

語法： JG #var, c, @addr;

JG #var, c, \$label;

JG #var, c, Ln;

參數說明： #var 變數

c 可為常數，變數或函數，但必須與#var 同類型

@addr 直接位址

\$label 址標位址

Ln 行號位址

參數範圍： 0 < addr < 8192

模式： 程式

功能： 當變數#var 大於 c 時，程式跳躍至指定位址。

說明：

本指令依變數是否大於一設定值，決定程式流程方向。當變數#var 大於 c 時，程式跳躍至指定位址。否則即繼續執行下一指令。其中變數#var 可為整數變數，也可為實數變數。c 可為常數，變數或函數，但必須與#var 同類型。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。址標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JNG 指令類似，僅判定的狀態條件相反。

相關指令：

範例：

指令： JGA (Jog by Analog input)

語法： JGA;

模式： 立即

功能： 依類比輸入埠的輸入電壓(JOG 輸入)，設定馬達運轉的方向及速度。

說明：

相關指令： JG0 停止 JOG 運轉指令

範例：

指令： JGF (Jog Forward)
JGR (Jog Reverse)
語法： JGF ;
JGR ;
模式： 立即
功能： 模擬 JOG 輸入，JGF 設定馬達持續正轉
JGR 設定馬達持續反轉。
說明：

本指令模擬搖桿(JOG)或按鈕輸入，設定馬達持續運轉。JGF 設定馬達正轉，JGR 設定馬達反轉。

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。

收到 JGF 或 JGR 指令後，被馬達帶動的滑台，即開始往正座標方向或負座標方向移動。滑台會以系統參數 VA 設定的加速度加速，待加至系統參數 VJ 設定的速度，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 JG0 指令，滑台才會減速停止。

在實務上，本指令一般是配合在 PC 電腦上的應用程式使用。例如以 VB 在視窗上設計一命令按鈕，當按鈕被按下時，即送出 JGF 或 JGR 指令。當按鈕被放開時，再送出“JG0”指令。這樣就可以模擬以按鈕的方式，來控制馬達運動。

相關指令： JG0 停止 JOG 運轉指令
範例：

指令： JG0 (Jog Stop)

語法： JG0;

模式： 立即

功能： 停止正在執行的 JGF 或 JGR 指令。

說明：

控制器正在執行 JGF 或 JGR 指令時，執行本指令才有意義。

收到本指令後，正在執行的 JGF 或 JGR 指令即開始減速，減速度由系統參數 VA 設定，當速度減至系統參數 VB 以下時即停止。

相關指令： JGF 設定馬達持續正轉

JGR 設定馬達持續反轉

範例：

指令： JI (Jump at Input)

語法： JI n,@addr;
JI n,\$label;
JI Ln;

參數說明： n 輸入埠 No.
@addr 直接位址
\$label 址標位址
Ln 行號位址

參數範圍： $1 < n < 8$
 $0 < \text{addr} < 8192$

模式： 程式

功能： 參數 n 指定的輸入為 On 時，程式跳躍至參數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 指定的輸入埠為 On (輸入光耦合器的 LED 有電流致能) 時，程式跳躍至參數所指定的位址。反之，參數 n 指定的輸入埠為 Off (輸入光耦合器的 LED 沒有電流致能)，則程式不跳躍，繼續執行下一指令。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。址標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JNI 指令類似，僅輸入埠的狀態條件相反。

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： JNE (Jump on Not Equal)

語法： JNE #var, c, @addr;

JNE #var, c, \$label;

JNE #var, c, Ln;

參數說明： #var 變數
c 可為常數，變數或函數，但必須與#var 同類型
@addr 直接位址
\$label 址標位址
Ln 行號位址

參數範圍： $0 < \text{addr} < 8192$

模式： 程式

功能： 當變數#var 不等於 c 時，程式跳躍至指定位址。

說明：

本指令依變數是否等於一設定值，決定程式流程方向。當變數#var 不等於 c 時，程式跳躍至指定位址。否則即繼續執行下一指令。其中變數#var 可為整數變數，也可為實數變數。c 可為常數，變數或函數，但必須與#var 同類型。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JE 指令類似，僅判定的狀態條件相反。

相關指令：

範例：

指令： JNG (Jump on Not Greater than)

語法： JNG #var, c, @addr;
JNG #var, c, \$label;
JNG #var, c, Ln;

參數說明： #var 變數
c 可為常數，變數或函數，但必須與#var 同類型
@addr 直接位址
\$label 址標位址
Ln 行號位址

參數範圍： $0 < \text{addr} < 8192$

模式： 程式

功能： 當變數#var 不大於 c 時，程式跳躍至指定位址。

說明：

本指令依變數是否大於一設定值，決定程式流程方向。當變數#var 不大於 c 時，程式跳躍至指定位址。否則即繼續執行下一指令。其中變數#var 可為整數變數，也可為實數變數。c 可為常數，變數或函數，但必須與#var 同類型。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。址標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JG 指令類似，僅判定的狀態條件相反。

相關指令：

範例：

指令： JNl (Jump at Input)

語法： JNl n, @addr ;
JNl n, \$label ;
JNl Ln ;

參數說明： n 輸入埠 No.
@addr 直接位址
\$label 標籤位址
Ln 行號位址

參數範圍： $1 < n < 8$
 $0 < \text{addr} < 8192$

模式： 程式

功能： 參數 n 指定的輸入為 Off 時，程式跳躍至參數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 指定的輸入埠為 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 指定的輸入埠為 On (輸入光耦合器的 LED 有電流致能)，則程式不跳躍，繼續執行下一指令。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JI 指令類似，僅輸入埠的狀態條件相反。

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 NoInput)

語法： JNTI n, @addr;
JNTI n, \$label;
JNTI Ln;

參數說明： n 輸入埠 No.
@addr 直接位址
\$label 標籤位址
Ln 行號位址

參數範圍： $1 < n < 16$
 $0 < \text{addr} < 8192$

模式： 程式

功能： 參數 n 指定的輸入為 Off，或在 tm 設定的時間內，由 On 轉為 Off，程式跳躍至參數 addr 所指定的位址繼續執行

說明：

本指令與 JNI 相似，可由輸入埠 On/Off 的狀態，決定程式流程方向，但可再加上時間限制。

若參數 n 指定的輸入埠為 Off (輸入光耦合器沒有電流致能) 時，或在 tm 設定的時間內，由 On 轉為 Off，程式跳躍至參數 addr 所指定的位址。反之，若參數 n 指定的輸入埠一直維持為 On (輸入光耦合器有電流致能)，則程式不跳躍，繼續執行下一指令。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JTI 指令類似，僅輸入埠的狀態條件相反。

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： JP (Jump to)

語法： JP @addr

JP \$label

JP Ln

參數說明： @addr 直接位址

\$label 址標位址

Ln 行號位址

參數範圍： 0 <addr < 8192

模式： 程式

功能： 將程式無條件跳躍至位址參數所指定的位址繼續執行

說明：

本指令只用在程式模式，可改變程式流程，無條件跳躍至參數所指定的位址。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

相關指令：

範例：

指令： JTI (Jump depend on Timer and Input)

語法： JTI n,tm,@addr;
JTI n,tm,\$label;
JTI n,tm,Ln;

參數說明： n 輸入埠 No.
@addr 直接位址
\$label 標籤位址
Ln 行號位址

參數範圍： $1 < n < 8$
 $0 < \text{addr} < 8192$

模式： 程式

功能： 參數 n 指定的輸入為 On，或在 tm 設定的時間內，由 Off 轉為 On，程式跳躍至參數 addr 所指定的位址繼續執行

說明：

本指令與 JI 相似，可由輸入埠 On/Off 的狀態，決定程式流程方向，但可再加上時間限制。

若參數 n 指定的輸入埠為 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 指定的輸入埠一直維持為 Off (輸入光耦合器沒有電流致能)，則程式不跳躍，繼續執行下一指令。

代表位址的參數，可以為直接位址，也可以用址標或行號表示。直接位址的前面應加一位址符號@。指標位址是一組以\$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 10 個字(或 5 個中文字)。使用行號位址則應在行號數字之前加一 L 以為識別。

本指令與 JI 指令類似，僅輸入埠的狀態條件相反。

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： MA (Move Absolute)

語法： MA x;

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

參數 x 另受到系統參數 CLP(座標正極限)及 CLM(座標負極限)所設定的範圍限制。

模式： 立即，程式

功能： 以工作速度移動被馬達帶動的滑台至指定位置，此位置由絕對座標設定。

說明：

CSBL-600 將速度分為高速及工作速度兩種模式。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。而在切削時，則應使用工作速度模式。

本指令的功能就是以系統設定的工作速度，移動被馬達帶動的滑台至指定位置。系統的工作速度，是由系統參數 VM 所設定（請參考系統參數 VM 的說明）。

x 參數即為終點位置的絕對座標。此座標可以有正負號，也可以使用小數表示。座標所用的單位是由系統參數 SC 所設定（請參考系統參數 SC 的說明）。

執行本指令後，會將系統的座標模式，設成絕對座標模式。以後只要執行“M”指令，就與“MA”指令有同樣的功能。

本指令與 FA 指令類似，只是使用的速度不同。本指令也與 MR 指令類似，只是使用的座標模式不同。

範例：

指令： MR (Move Relative)

語法： MR x;

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

參數 x 另受到系統參數 CLP(座標正極限)及 CLM(座標負極限)所設定的範圍限制。

模式： 立即，程式

功能： 以工作速度移動被馬達帶動的滑台至指定位置，此位置由相對於現在位置的座標設定。

說明：

CSBL-600 將速度分為高速及工作速度兩種模式。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。而在切削時，則應使用工作速度模式。

本指令的功能就是以系統設定的工作速度，移動被馬達帶動的滑台至指定位置。系統的工作速度，是由系統參數 VM 所設定（請參考系統參數 VM 的說明）。

x 參數即為終點位置相對於現在位置的座標。此座標可以有正負號，也可以使用小數表示。座標所用的單位是由系統參數 SC 所設定（請參考系統參數 SC 的說明）。

執行本指令後，會將系統的座標模式，設成相對座標模式。以後只要執行“M”指令，就與“MR”指令有同樣的功能。

本指令與 FR 指令類似，只是使用的速度不同。本指令也與 MA 指令類似，只是使用的座標模式不同。

範例：

指令： ON EV (On Event)

語法： ON EVn, @addr ;

ON EVn, \$label ;

ON EVn, Ln ;

參數範圍： $1 \leq n \leq 4$

$0 < \text{addr} < 8192$

模式： 程式

功能：

說明：

範例：

指令： OUT (Output to RS232)

語法： OUT #var;

OUT "string";

參數範圍：

模式： 立即，程式

功能：

說明：

範例：

指令： **OUTP** (Output Port)

語法： **OUTP** *n* ; ;

參數範圍： $0 \leq n \leq 15$

模式： 立即，程式

功能：

說明：

範例：

指令： PA (Program Apend)

語法： PA;

參數範圍：

模式： 立即

功能：

說明：

範例：

指令： PD (Program Delete)

語法： PD n;

參數範圍：

模式： 立即

功能： 刪除一行程式

說明：

範例：

指令： PE (Program Edit)

語法： PE n ;

參數範圍：

模式： 立即

功能： 修改一行程式

說明：

範例：

指令： PG (Program Generate)

語法： PG ;

參數範圍：

模式： 立即

功能： 進入編輯程式模式

說明：

範例：

指令： PI (Program Insert)

語法： PI n ;

參數範圍：

模式： 立即

功能：

說明：

範例：

指令： PL (Program List)

語法： PL ;

參數範圍：

模式： 立即

功能：

說明：

範例：

指令： PLS P (Pulse output)
語法： PLS Pn, tm;
參數範圍： $1 \leq n \leq 4$
 $4 \leq tm \leq 65535$
模式： 立即，程式
功能： 由指定的輸出埠輸出一脈波。
說明：

CSBL-600 控制器共有 4 個供一般用途的輸出埠，每一輸出埠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”。

指令： PWM P (Pulse Width Modulate Port)

語法： PWM Pn, f, duty;

參數範圍： $1 \leq n \leq 4$

$4 \leq tm \leq 65535$

模式： 立即，程式

功能： 由指定的輸出埠輸出 PWM 脈波。

說明：

指令： PZ (Pause)
語法： PZ;
參數範圍：
模式： 立即，程式
功能： 馬達停止
說明：

指令： P1CLR P (Position Clear port)
P2CLR P
語法： P1CLR P_n, x;
P2CLR P_n, x;
參數範圍： 1 ≤ n ≤ 4
模式： 立即，程式
功能： 當到達座標位置 x 時，設定輸出埠為關(Off)狀態。
說明：

CSBL-600 控制器有兩個位置模組(Position module)，即 P1 與 P2。利用這兩個位置模組，可以達成依位置直接控制輸出埠的目的，完全與馬達運轉狀態無關。

本指令下達後，並不立即改變輸出口的狀態。而是要等到馬達轉到座標位置 x 時，才關閉指定的輸出埠。關閉的動作與”CLR P”指令類似，就是將參數 n 指定的輸出埠設定為「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。

兩個位置模組 P1 與 P2，可以分開運用，不會有互相衝突問題。也就是說可以用 P1 設定一個位置，P2 設定另一個位置，分別對不同的輸出埠做不同的控制。

位置模組是完全獨立運作的元件，觸發動作是依據馬達編碼器(Encoder)的回授脈波定位，與馬達轉速或當時的控制狀態無關。所以即使馬達在高速狀態下不減速，觸發位置仍絕對準確。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

指令： P1SET P (Position SET port)
P2SET P
語法： P1SET Pn, x;
P2SET Pn, x;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 當到達座標位置 x 時，設定輸出埠為開(On)狀態。
說明：

CSBL-600 控制器有兩個位置模組(Position module)，即 P1 與 P2。利用這兩個位置模組，可以達成依位置直接控制輸出埠的目的，完全與馬達運轉狀態無關。

本指令下達後，並不立即改變輸出口的狀態。而是要等到馬達轉到座標位置 x 時，才設定指定指定的輸出埠。設定的動作與”SET P”指令類似，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。

兩個位置模組 P1 與 P2，可以分開運用，不會有互相衝突問題。也就是說可以用 P1 設定一個位置，P2 設定另一個位置，分別對不同的輸出埠做不同的控制。

位置模組是完全獨立運作的元件，觸發動作是依據馬達編碼器(Encoder)的回授脈波定位，與馬達轉速或當時的控制狀態無關。所以即使馬達在高速狀態下不減速，觸發位置仍絕對準確。

有關輸出電路的使用方法，請參照“輸出輸入電路”。

指令： REDO (Redo)
語法： REDO;
參數範圍：
模式： 立即，程式
功能： 馬達恢復運轉，繼續在”PZ”指令之前的動作
說明：

指令： **RESET** (Reset)
語法： **RESET;**
模式： 立即，程式
功能： 重置馬達及系統狀態
說明：

指令： RET (Return)
語法： RET;
參數範圍：
模式： 程式
功能： 由副程式返回
說明：

指令： RETI (Return from Interrupt)
語法： RETI ;
參數範圍：
模式： 程式
功能： 由中斷副程式返回
說明：

指令： RSV (Reset motor Space Vector)
語法： RSV;
參數範圍：
模式： 程式
功能： 重置馬達相位
說明：

指令： **SAVE C** (Save Configer)
語法： **SAVE C;**
模式： 立即，程式
功能： 儲存系統參數至系統記憶體
說明：

指令： SAVE D (Save Data)
語法： SAVE D;
模式： 立即，程式
功能： 儲存數據資料至系統記憶體
說明：

指令： SAVE P (Save Program)
語法： SAVE P;
模式： 立即
功能： 儲存程式至系統記憶體
說明：

指令： SET EV (Set Event)
語法： SET EVn, C;
SET EVn, In;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 設定事件發生的定義條件
說明：

指令： SET P (SET output)
語法： SET Pn;
參數範圍： $1 \leq n \leq 4$
模式： 立即，程式
功能： 設定輸出埠為開(On)狀態。
說明：

CSBL-600 控制器共有 4 個供一般用途的輸出埠，每一輸出埠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”。

指令： P1SET P (SET output by Position)

P2SET P

語法： P_tSET P_n, x;

參數說明： n 輸出埠 No.

x 座標位置

參數範圍： $1 \leq n \leq 4$

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

模式： 立即，程式

功能： 在滑台位置到達指定座標時，將指定的輸出埠設定為 On 狀態。

說明：

本指令與 SET P_n 指令類似。但執行本指令，並不會立即改變輸出埠的狀態。而是在滑台到達參數 x 指定的位置時，才將參數 n 指定的輸出埠設定為 On 狀態(輸出光耦合器的電晶體 On)。輸出狀態改變設定時，馬達及滑台的運動不受任何影響，並不需要做減速或停車配合。

本指令可配合各種運動控制指令使用，包括手動 (Hold off 狀態) 及 Jog 移動。

本指令只在下達後第一次滑台到位時執行，以後即使再重複到位，也不再動作。指令下達後若要取消 (第一次到位前)，可再下達沒有 x 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”。

指令： PCLR P (Clear output by Position)
語法： PCLR Pn, x;
參數說明： n 輸出埠 No.
x 座標位置
參數範圍： $1 \leq n \leq 4$
 $-32,767.9999 \leq x \leq +32,767.9999$
模式： 立即，程式
功能： 在滑台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埠的狀態。而是在滑台到達參數 x 指定的位置時，才將參數 n 指定的輸出埠設定為 Off 狀態(輸出光耦合器的電晶體 Off)。輸出狀態改變設定時，馬達及滑台的運動不受任何影響，並不需要做減速或停車配合。

本指令可配合各種運動控制指令使用，包括手動 (Hold off 狀態) 及 Jog 移動。

本指令只在下達後第一次濟到位時執行，以後即使再重複到位，也不再動作。指令下達後若要取消，可再下達沒有 x 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： SI (Set Integer 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：

本指令用以宣告一整數變數。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。變數一定要先經宣告才可以使用。

整數變數是用來儲存不帶小數的整數數字，數字大小的範圍為-2,147,483,648~2,147,483,647。若將一帶小數的數字存到整數變數，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，#參數 1，#Count100，等，都是可接受的變數名稱。

設定整數變數新值，有下列多種方法：

在宣告的同時設定其值

```
SI #AA=125;
```

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;
```

指令： SI (Set Integer 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：

本指令用以宣告一整數變數。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。變數一定要先經宣告才可以使用。

整數變數是用來儲存不帶小數的整數數字，數字大小的範圍為 -2,147,483,648~2,147,483,647。若將一帶小數的數字存到整數變數，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，#參數 1，#Count100，等，都是可接受的變數名稱。

設定整數變數新值，有下列多種方法：

在宣告的同時設定其值

```
SI #AA=125;
```

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;
```

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： 宣告一實數變數

說明：

本指令用以宣告一實數變數。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。變數一定要先經宣告才可以使用。

實數變數是用來儲存帶小數的數字，數字大小的範圍為-32767.9999~32767.9999。若將一整數常數存到實數變數，該常數會自動被轉成實數型式。但若將一整數變數存到實數變數，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，#參數 1，#Count100，等，都是可接受的變數名稱。

設定實數變數新值，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;
```

指令： STOP (Stop)
語法： STOP ;
模式： 程式，立即
功能： 馬達緊急停止
說明：

指令： UL (UpLoad)
語法： UL;
模式： 立即
功能： 上載程式
說明：

指令： WI (Wait Input)

語法： WI n;

參數範圍： $1 \leq n \leq 8$

模式： 立即命令，程式

功能： 等待參數 n 指定的輸入埠致能(On)，再繼續後續的程式

說明：

指令： WNI (Wait No Input)
語法： WNI n;
參數範圍： $1 \leq n \leq 8$
模式： 立即命令，程式
功能： 等待參數 n 指定的輸入埠信號關閉(Off)，再繼續後續的程式
說明：

本指令的功能是讓系統暫停一段時間，然後再繼續後續的程式。暫停的時間，由參數 tm 設定。時間單位為 msec。

有些立即執行指令，在解譯後即立即被執行，可能會不受本指令延遲時間的控制。要確保延遲時間的效果，可在緊接本指令後，再加一”DN”指令，即可暫停解譯後續程式，確定讓系統暫停所有動作。

指令： WT (Wait)
語法： WT tm;
參數範圍： $1 \leq tm \leq 65536$
模式： 程式
功能： 程式暫停，停止由參數 tm 設定的時間後，再繼續後續的程式
說明：

本指令的功能是讓系統暫停一段時間，然後再繼續後續的程式。暫停的時間，由參數 tm 設定。時間單位為 msec。

有些立即執行指令，在解譯後即立即被執行，可能會不受本指令延遲時間的控制。要確保延遲時間的效果，可在緊接本指令後，再加一”DN”指令，即可暫停解譯後續程式，確定讓系統暫停所有動作。

五、函數

5-1. 函數說明

CSBL600 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。

CSBL600 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

5-2. 函數使用方法例

在 CSBL600 要查詢函數值，只要執行“?函數”指令，CSBL600 就會立即以 ASCII 字串型式將函數值回應給終端機。

範例

在程式模式，使用者可以配合變數功能，將函數運算的結果，做為程式動作的依據。

範例

六、與 PC 連線

6-1. 啓用 Window “終端機” 軟體

CSBL600 附有一標準的 RS232 埠，可直接與 PC 或 PDA 連線。連線所需的軟體，我們建議使用 Window 作業系統內附的“終端機”或“超級終端機”即可。這是最標準的 RS232 通訊軟體。

在 Window 中要啓用“終端機”軟體程序如下：

步驟 1

在程式集→附屬應用程式→通訊→超級終端機 內開啓“超級終端機”

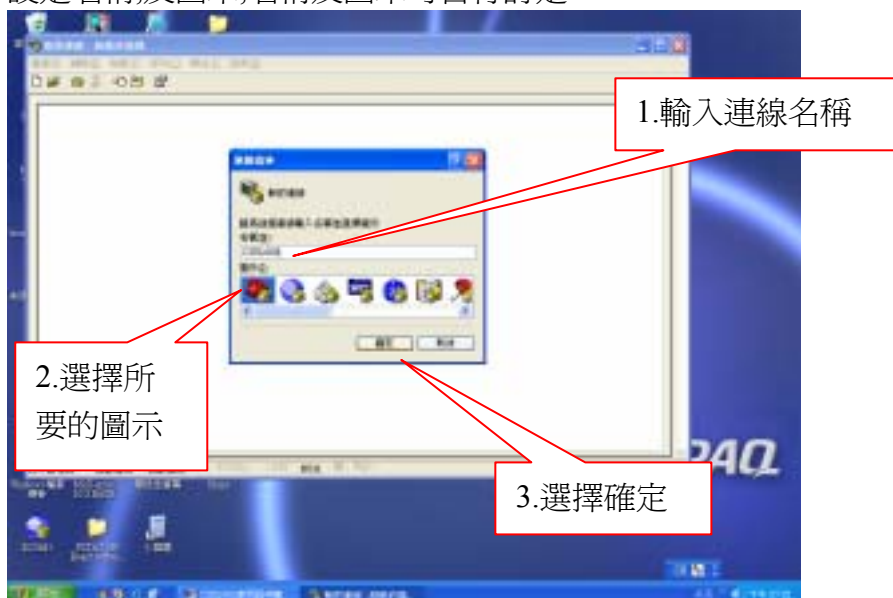


註：WIN98 只附有“終端機”軟體，其功能與“超級終端機”並無不同。

在上述畫面中若找不到“終端機”或“超級終端機”，可在控制台→新增/移除程式→WINDOWS 安裝程式內的通訊中開啓即可。

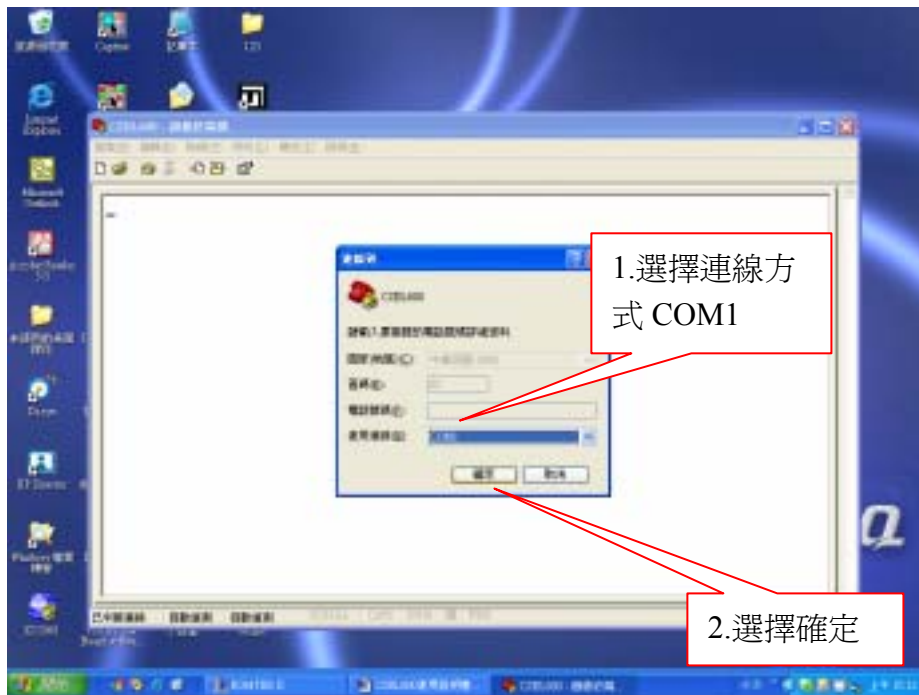
步驟 2

設定名稱,及圖示,名稱及圖示可自行訂定。



步驟 3

依據您電腦的硬體配置，選擇連線方式為 COM1 或其他 COM port。



步驟 4

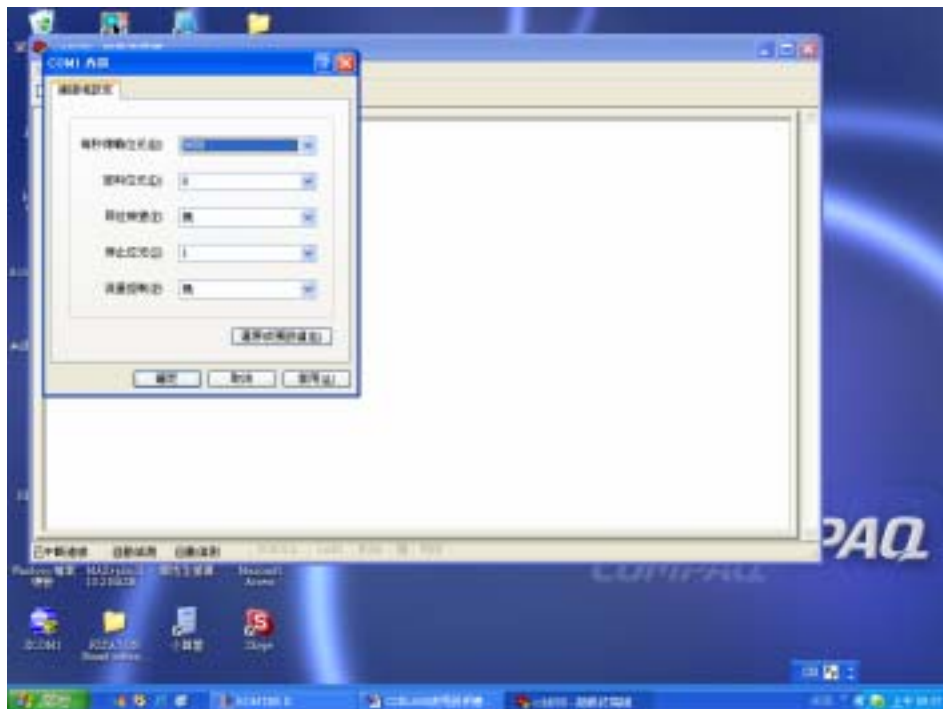
設定連接埠，每秒傳輸位元：9600，

資料位元：8，

同位檢查：無，

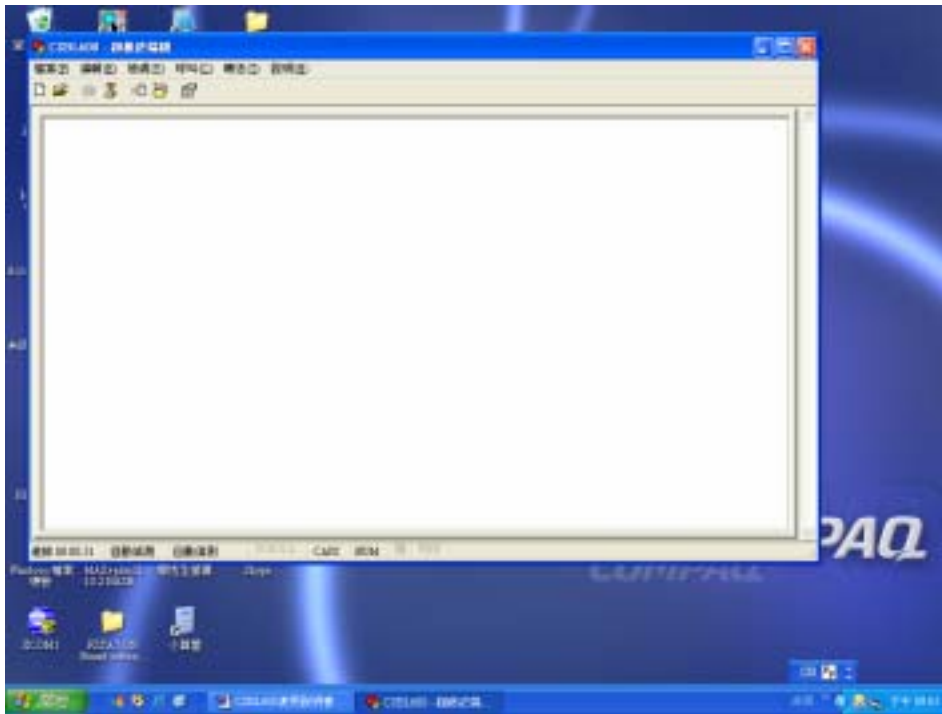
停止位元：1，

流量控制：無



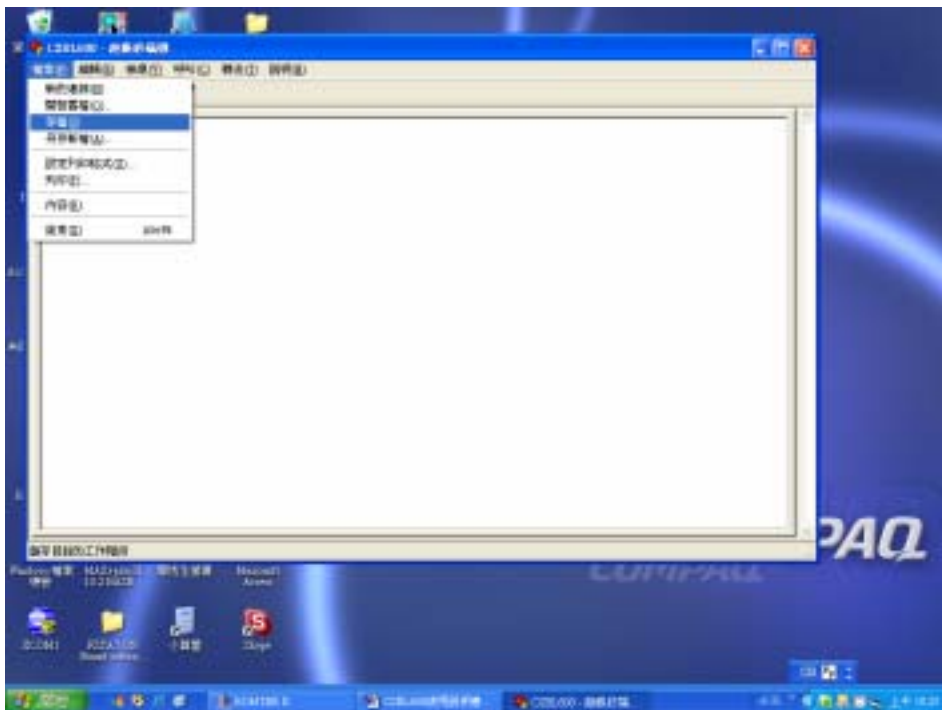
步驟 5

完成後電腦終端機即成待機模式

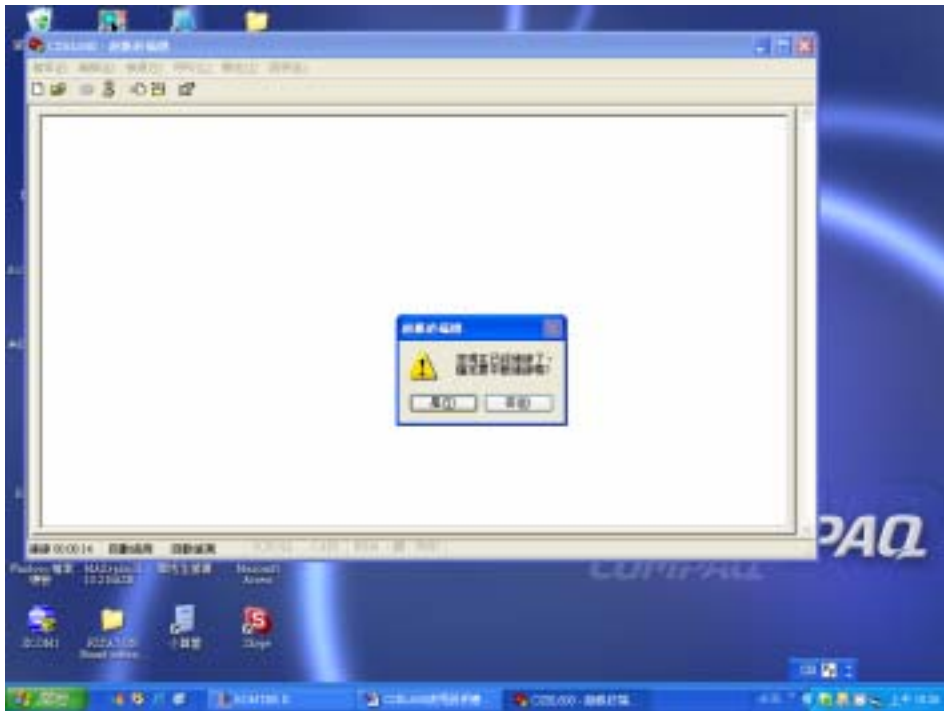


步驟 6

先選取存檔，然後即可結束



步驟 7



步驟 8

以後要再使用“終端機”軟體，只要在開始功能表中選取即可



6-2. 第一次在 PC 上使用“終端機”軟體與 CSBL600 連線

步驟 1

CSBL600 的指撥開關應設定如下

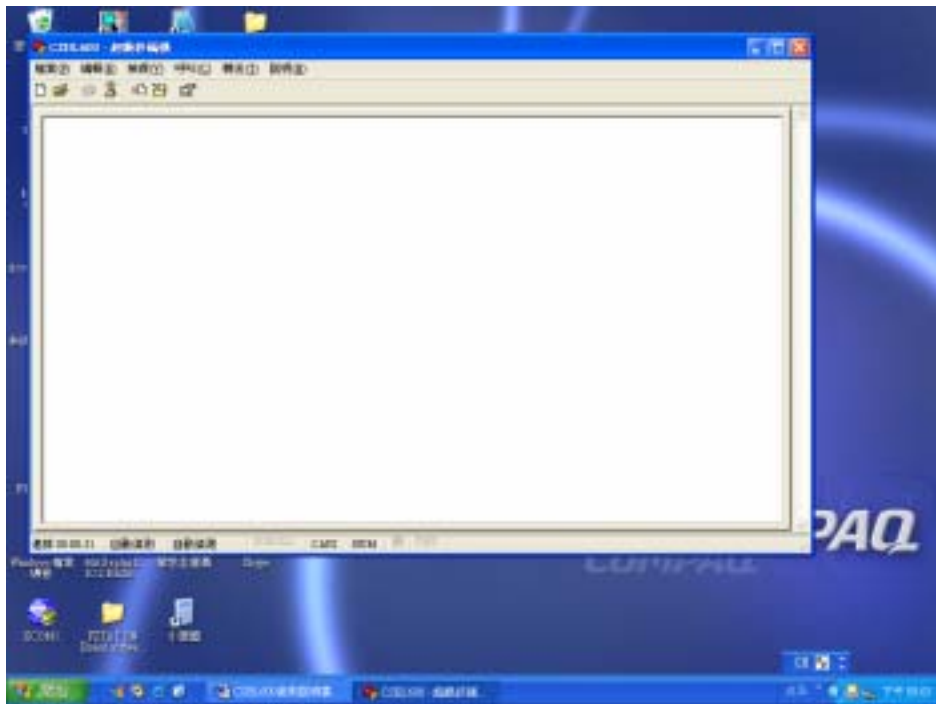
開關 No.	開關位置	功能說明
SW1	off	選擇控制器模式
SW2	off	選擇開機後不要自動進入程式執行模式
SW3	off	Echo On

步驟 2

先確定 CSBL600 的電源為關閉狀態，將馬達接上 CSBL600，包括馬達線及編碼器線兩組。再將 CSBL600 與 PC 以 RS232 線連接。

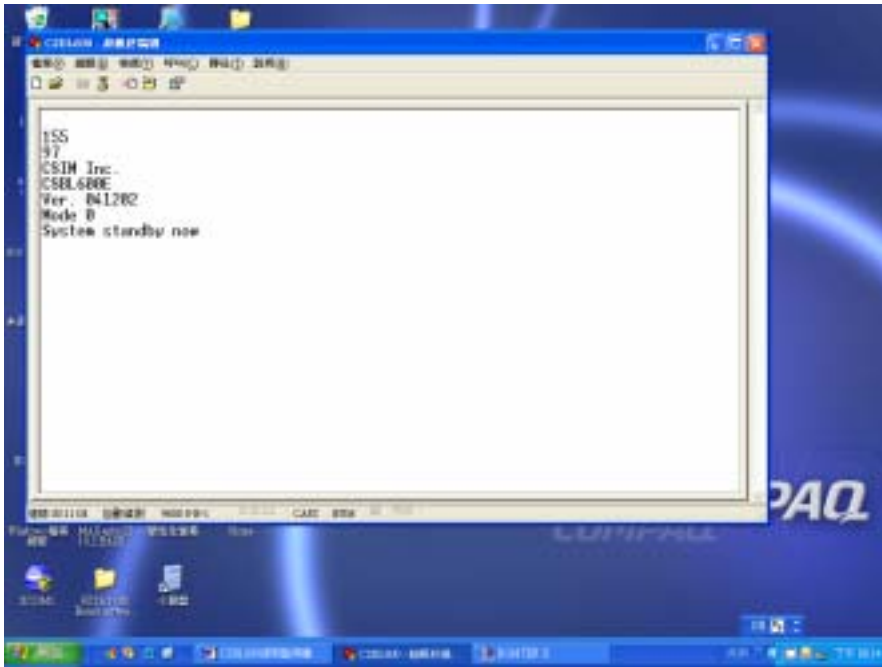
步驟 3

在 PC 上執行“終端機”軟體



步驟 4

打開 CSBL600 電源，此時在終端機上應看到下列回應。



步驟 5

在 PC 鍵入 "POL=4" 及 Enter 鍵，設定使用的馬達為 4 極馬達

CSBL600 應回應 "OK!"

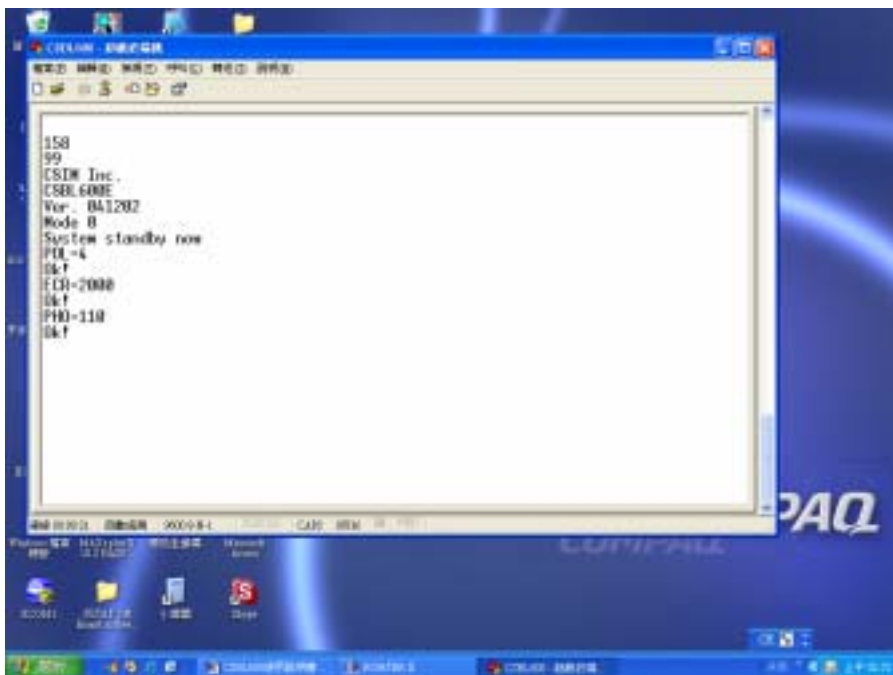
在 PC 鍵入 "ECR=2000" 及 Enter 鍵，設定使用的編碼器解析度為 2000Count/轉

CSBL600 應回應 "OK!"

在 PC 鍵入 "PHO=110" 及 Enter 鍵，

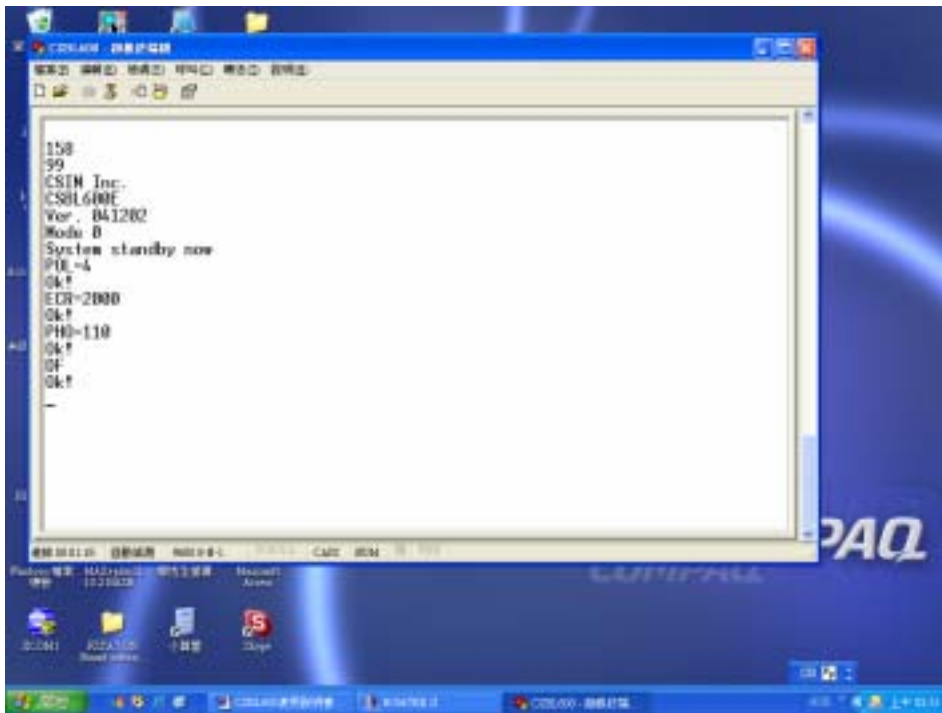
CSBL600 應回應 "OK!"

注意：以上參數僅適用於配合 CSBL600 的標準馬達及編碼器，若您使用的馬達或編碼器並非標準品，請向經銷商洽詢適用的參數。



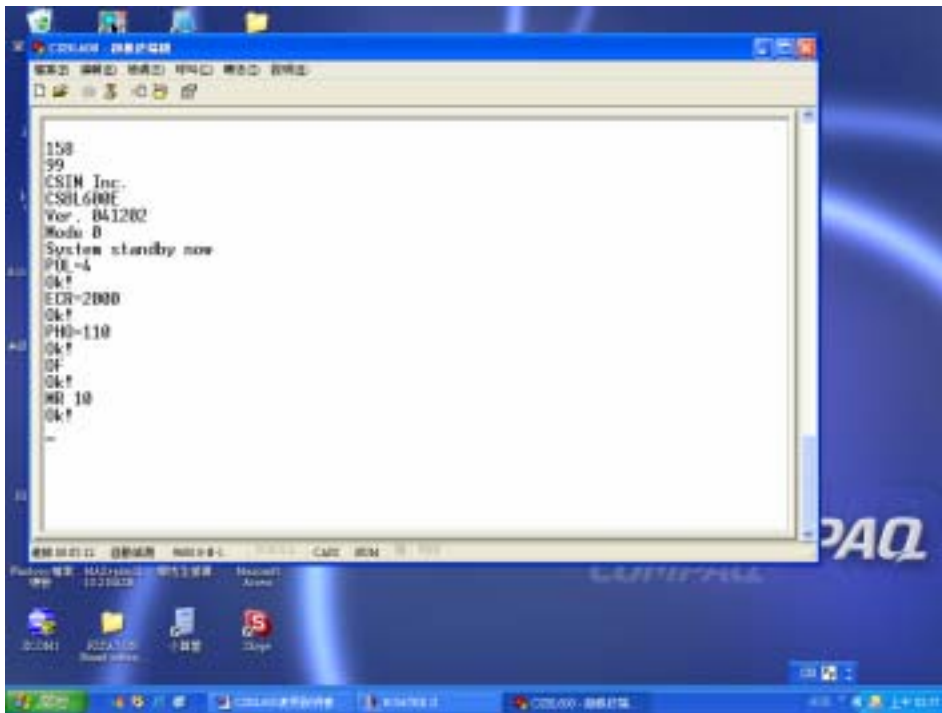
步驟 6

在 PC 鍵入”DF”及 Enter 鍵，將 CSBL600 的所有其他的系統參數設成預設值
CSBL600 應回應 ”OK!”



步驟 7

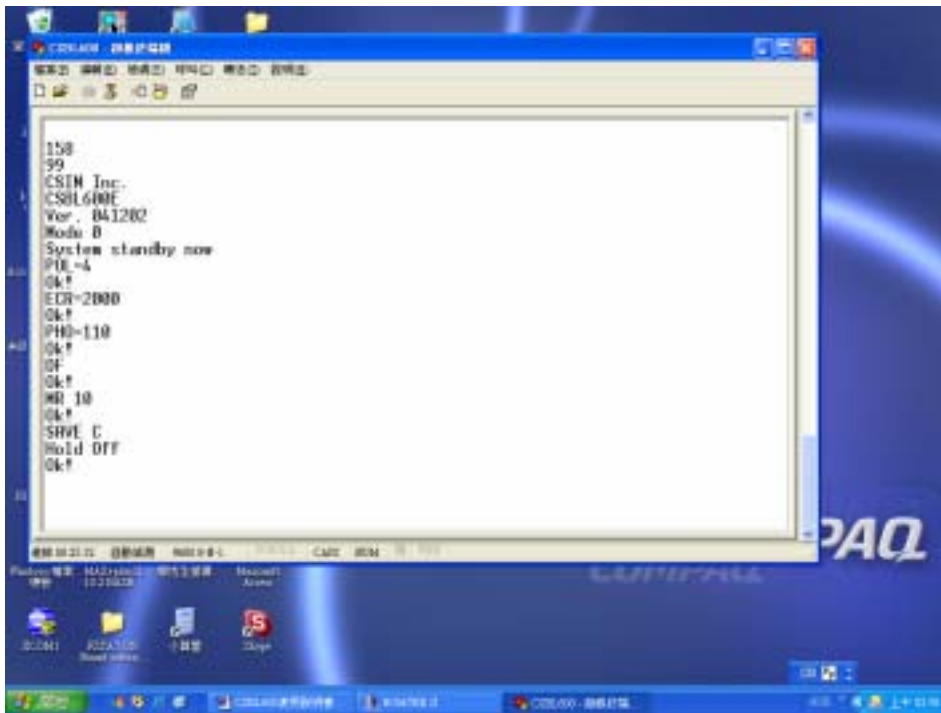
在 PC 鍵入”MR 10”及 Enter 鍵，命令 CSBL600 將馬達正轉一周。
CSBL600 應回應 ”OK!”，並立即將馬達正轉一周



步驟 8

檢查此時馬達的轉軸應已被鎖住〈以手去旋轉會有反抗扭力〉

若一切正常，在 PC 鍵入”SAVE C”及 Enter 鍵，命令 CSBL600 儲存現在的系統組態。
CSBL600 應回應 ”Hold Off”及”OK!”。



步驟 8

此時已可關機，而且以後開機後就可立即操作，不需再做上述的參數設定。

6-3. 編輯 CSBL600 程式

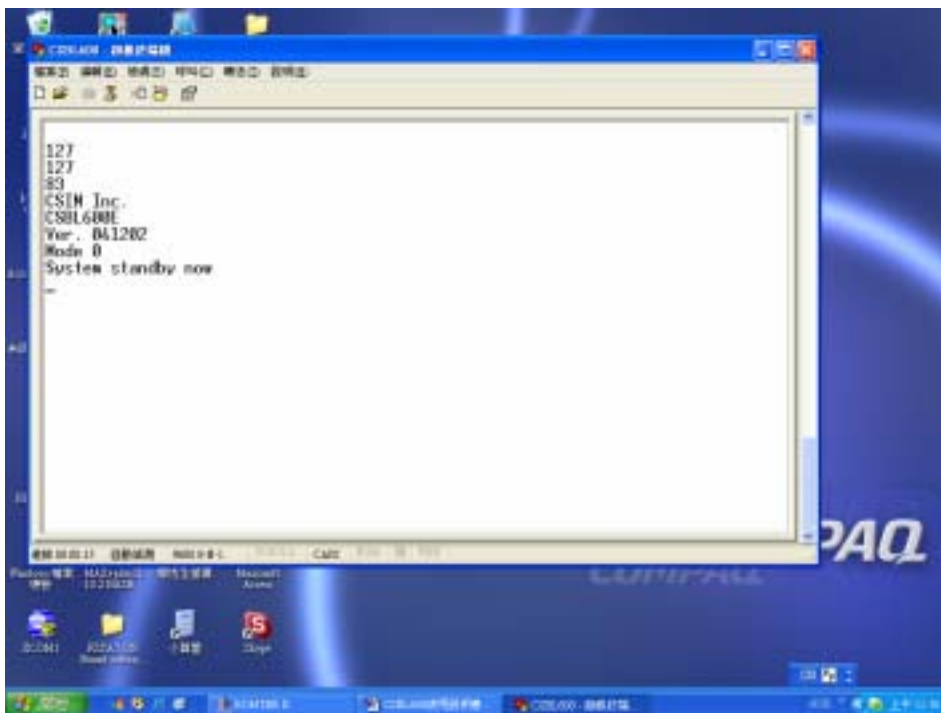
在 PC 上使用“終端機”軟體，與 CSBL 連線後，就可以直接下指令給 CSBL600 執行，或讀取 CSBL600 送回的訊息。但使用者若要做較複雜的自動化控制，一般而言會需要用執行程式的方式來完成。

要編輯 CSBL600 的程式，使用者有兩種方法。一是使用 CSBL600 內建的編譯器，配合 PC 的終端機功能，就可以直接在 上（On line）編輯程式。另一種方法是利用使用者自己所熟悉的任何文字處理程式，如視窗軟體所附的“Word Pad”或“記事本”等軟體編寫程式，寫好後再透過 RS232 埠下載（Download）至 CSBL600 執行。在本節中我們先介紹如何直接在 上（On line）編輯程式。利用外部文字處理程式編寫程式，然後再下載的方法，我們將在下一節中再做詳細說明。

註：為了便於說明，我們將從開機開始，將編寫一程式的全部過程，按步驟說明於後。實際編寫程式時，並不一定要完全遵循這些步驟。只要在控制器模式，使用者可以隨時進出程式編輯模式，也可以任意插入或修改一已有的程式。

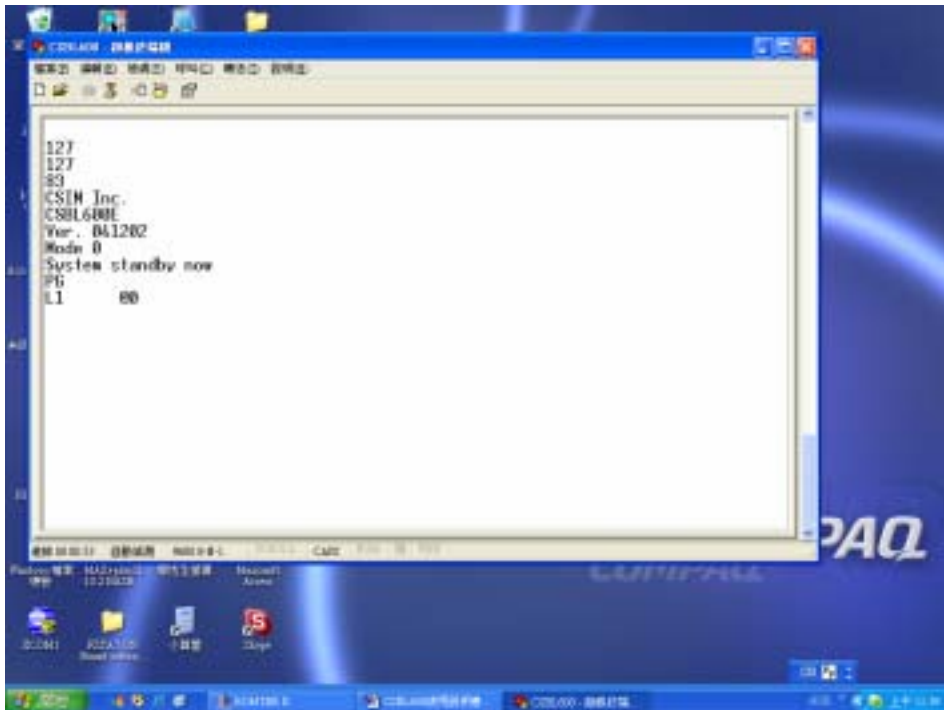
步驟 1

開機，“終端機”會顯示出開機畫面。



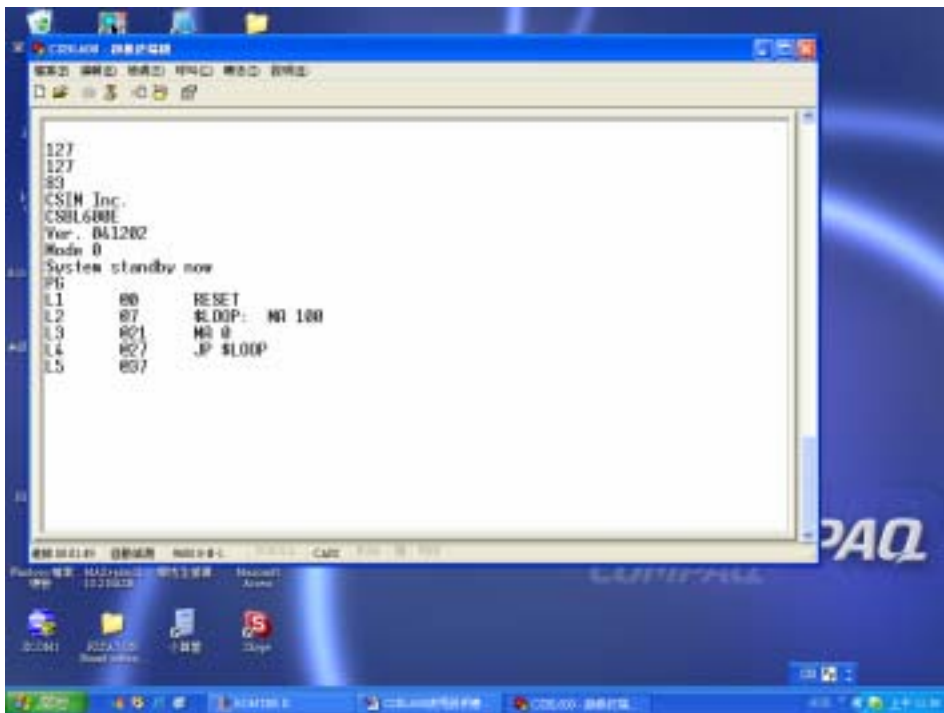
步驟 2

以“PG”指令進入程式編輯模式，此時在“終端機”的畫面上會顯示現在的行號“L1”及位址“@0”



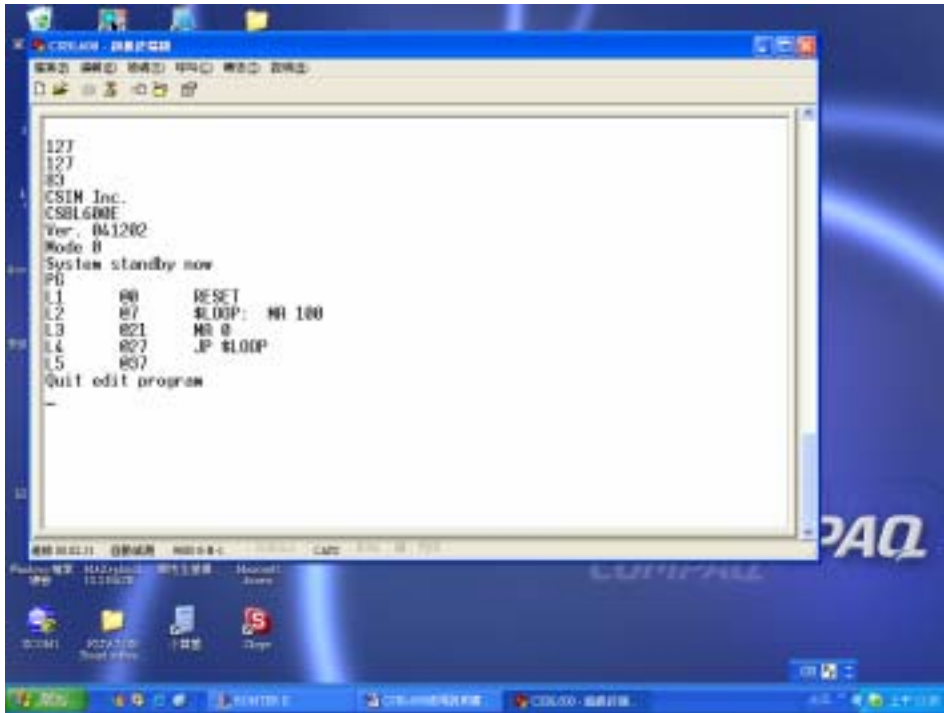
步驟 3

鍵入使用者程式。每鍵入一行程式，在結尾鍵入 Enter 鍵後，即會自動跳至下一行位置，並自動顯示新的行號及位址。



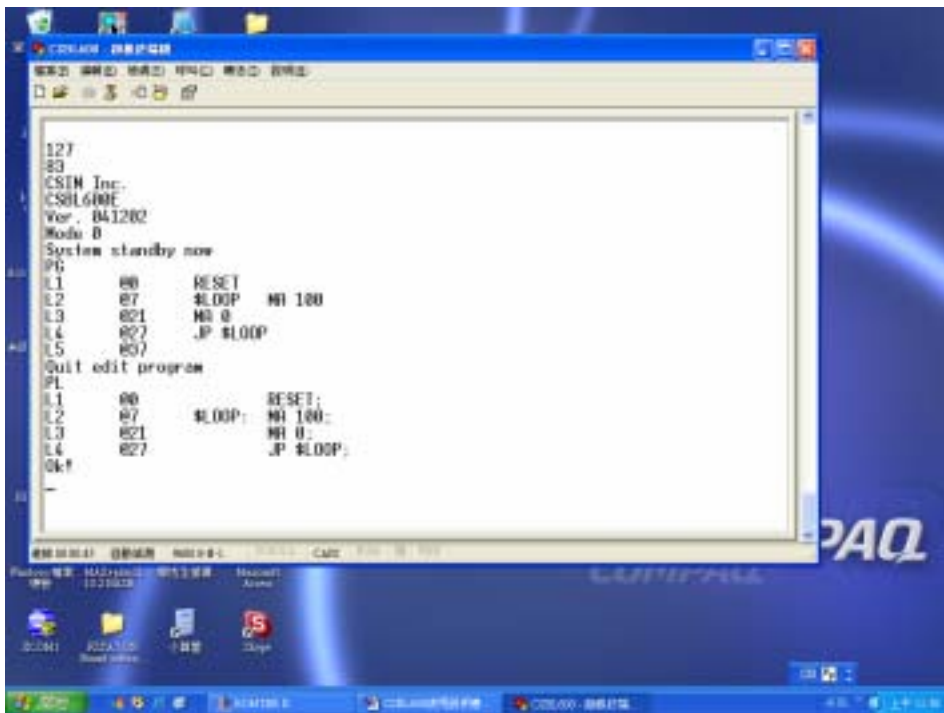
步驟 4

程式全部鍵入後，只要再重複鍵入一次 Enter 鍵，或鍵入 ESC 鍵，即可離開程式編輯模式，回到立即命令模式。



步驟 5

在立即命令模式下，鍵入“PL”指令（Program List）。在“終端機”上就會看到剛才鍵入的全部程式內容。並且已自動修正為正確的程式格式。

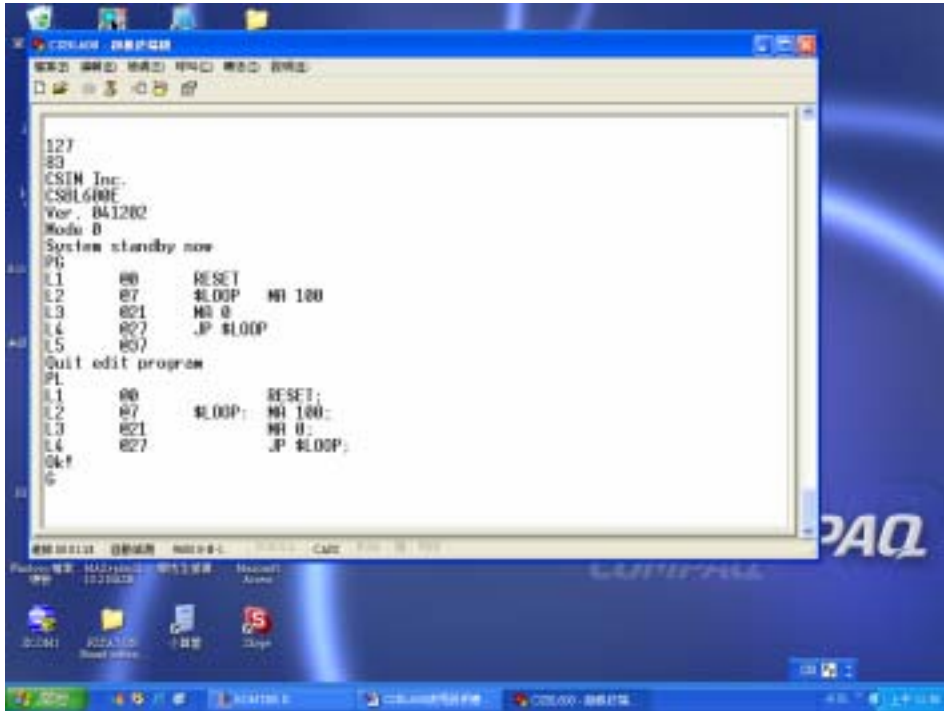


步驟 6

在立即命令模式下，要啟動程式，只要鍵入“G”指令（Go），程式即會開始執行。

另一啟動程式的指令是“SG”指令（Step Go），可以讓 CSBL600 進入單步執行模式。每鍵入一次 Enter 鍵，程式只執行一行。這種模式在程式驗證及除錯（Debug）時非常有用。

本程式範例並無太複雜的動作，只是讓馬達連續做正反方向運轉。



```
127
83
CSIM Inc.
CSBL600E
Ver. 041202
Mode 0
System standby now
PD
L1      00      RESET
L2      07      $LOOP: NI 100
L3      021     NI 0
L4      027     .JP $LOOP
L5      037
Quit edit program
P1
L1      00      RESET;
L2      07      $LOOP: NI 100;
L3      021     NI 0;
L4      027     .JP $LOOP;
Ok!
G
```

步驟 7

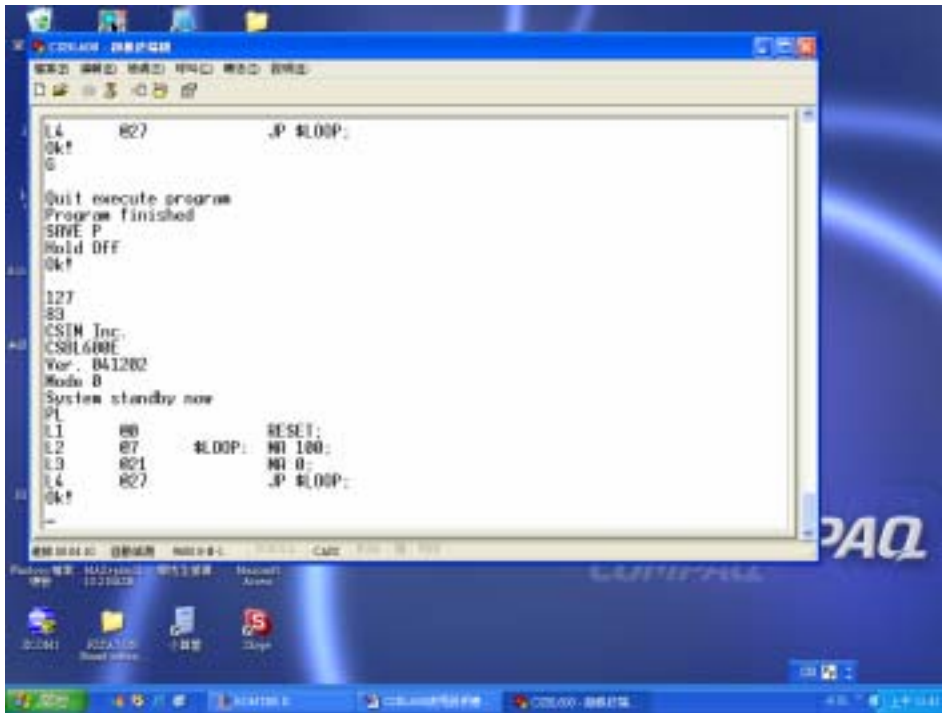
在本範例中，程式已形成一迴圈，所以一旦啟動永遠不會自動結束。要停止此程式，可按 ESC 鍵，CSLB600 即會停止程式執行，回到立即命令模式。



```
83
CSIM Inc.
CSBL600E
Ver. 041202
Mode 0
System standby now
PD
L1      00      RESET
L2      07      $LOOP: NI 100
L3      021     NI 0
L4      027     .JP $LOOP
L5      037
Quit edit program
P1
L1      00      RESET;
L2      07      $LOOP: NI 100;
L3      021     NI 0;
L4      027     .JP $LOOP;
Ok!
G
Quit execute program
Program finished
```

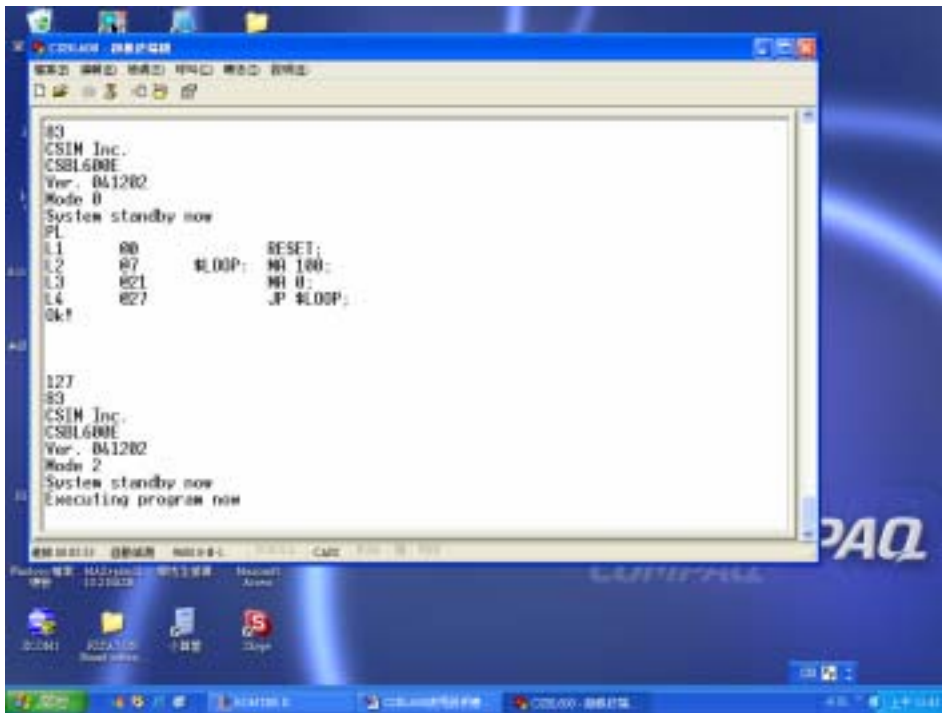
步驟 8

若程式功能已驗證完畢，則可以執行“SAVE P”指令，將編輯好的程式儲存到系統的 Flash ROM 中。儲存完成後，即使關機再重新開機，程式仍被保存。



步驟 9

程式若已編輯完成，並且已儲存在 CSBL600 的 Flash ROM 中。再將指撥開關設定在自動啟動（Auto Run）模式（指撥開關 SW1 Off，SW2 On）。則以後只要電源打開，CSBL600 就會自動啟動執行這些程式，完全不再需要 PC 或其它控制器裝置。



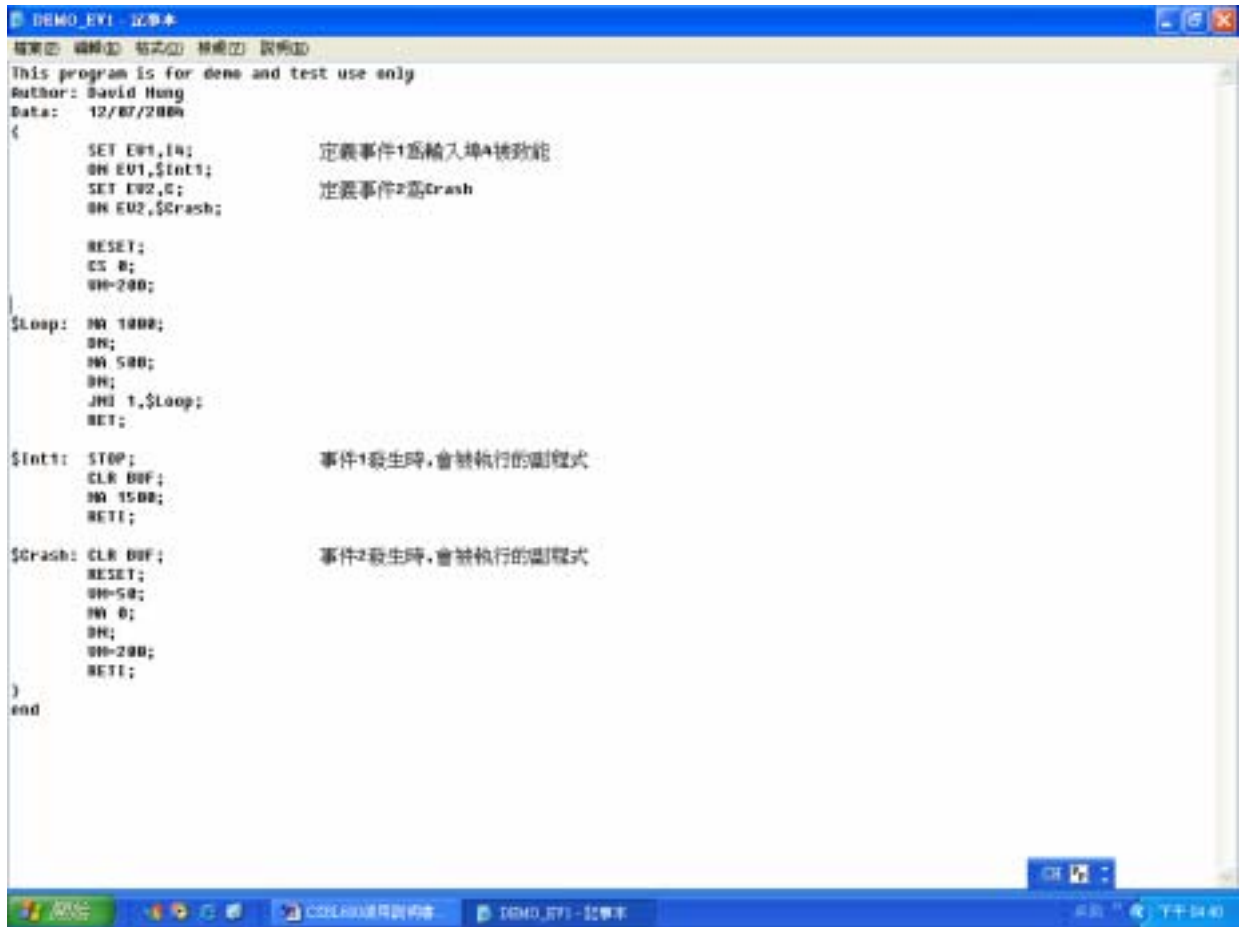
6-4. 利用外部文字處理程式編寫程式

使用者除了可以用 CSBL600 內建的編譯器，在 上（On line）編輯程式外。也可以用自己所熟悉的任何文字處理程式，如視窗軟體所附的“Word Pad”或“記事本”等軟體編寫程式，寫好後再透過 RS232 埠下載（Download）至 CSBL600 執行。

以下我們將以一實際範例，詳細說明每一步驟。

步驟 1

使用任何您所熟悉的文字處理程式（在本範例為“記事本”），編寫程式



```
DEMO_EV1 - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
This program is for demo and test use only
Author: David Hung
Date: 12/07/2009
{
    SET E01,I4;           定義事件1為輸入埠A轉功能
    ON E01,$Int1;
    SET E02,C;           定義事件2為Crash
    ON E02,$Crash;

    RESET;
    CS 0;
    ON=200;

$Loop: ON 1000;
    ON;
    ON 500;
    ON;
    JMI 1,$Loop;
    RET;

$Int1: STOP;           事件1發生時,會被執行的副程式
    CLR DIF;
    ON 1500;
    RETI;

$Crash: CLR DIF;       事件2發生時,會被執行的副程式
    RESET;
    ON=50;
    ON 0;
    ON;
    ON=200;
    RETI;
}
end
```

以上程式編寫完成後，請以文字文件格式儲存（請勿用 Rich text 格式）。

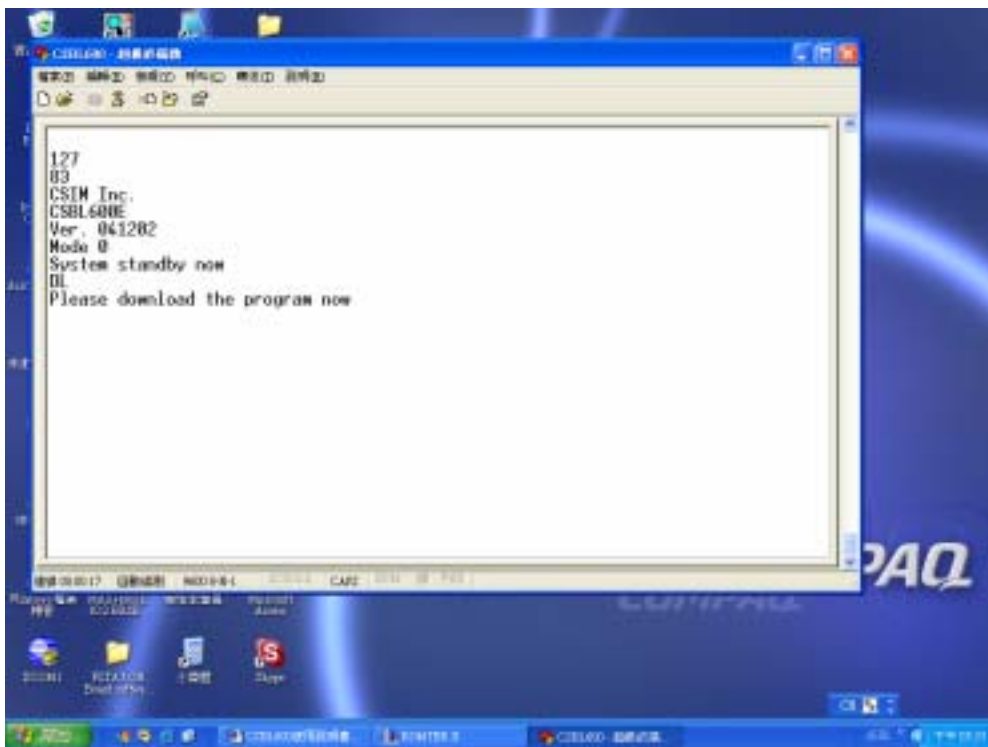
使用外部文字處理程式編寫程式時，由於 CSBL600 無法提供即時除錯功能，使用者需自行注意所有的程式格式及指令語法，必須完全無誤才能順利下載（Download）。

一般常易被人疏忽的有下列諸項

1. 所有的程式必須包含在“{”及“}”之間。程式下載時，在“{”之前及“}”之後的文字均會被視為註解而被忽略。
2. 每一行程式均應以“;”結束。在“;”之後的也會被視為註解而被忽略。
3. 所有址標應以“\$”開頭，以“:”結尾。
4. “{”及“}”必須各獨佔一行，所以在程式最後的“}”之後，不能直接結束。至少還需要加一換行（「CR」及「LF」）才能結束檔案。

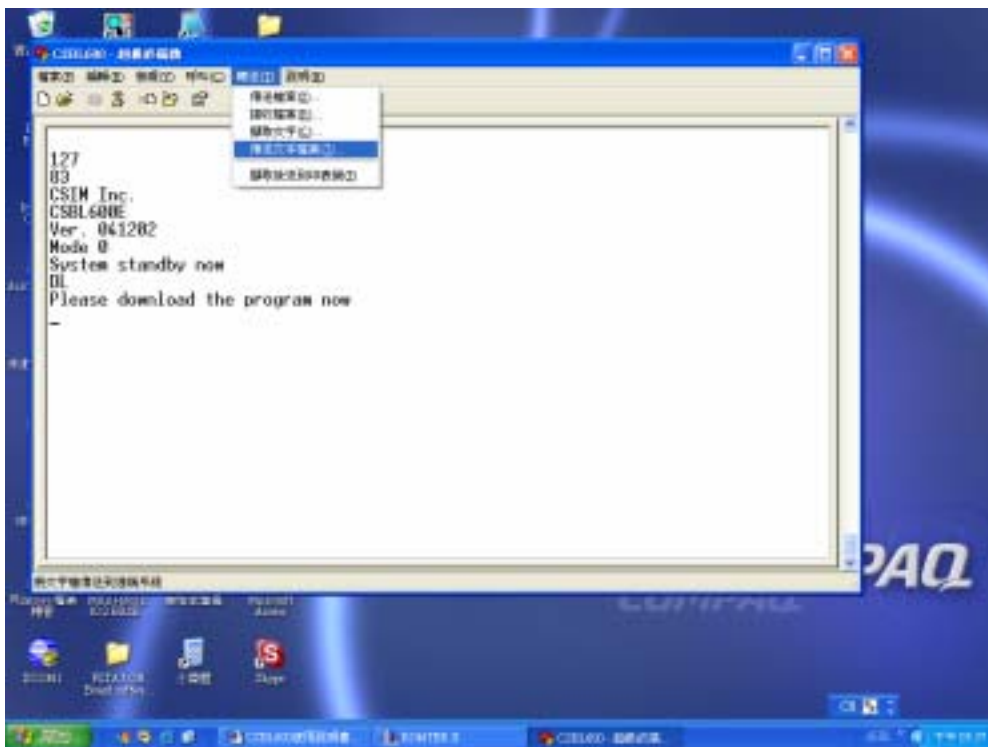
步驟 2

將 CSBL600 與 PC 以“終端機”軟體連線，在立即命令模式下鍵入“DL”指令，CSBL600 就會回應要求下載程式。



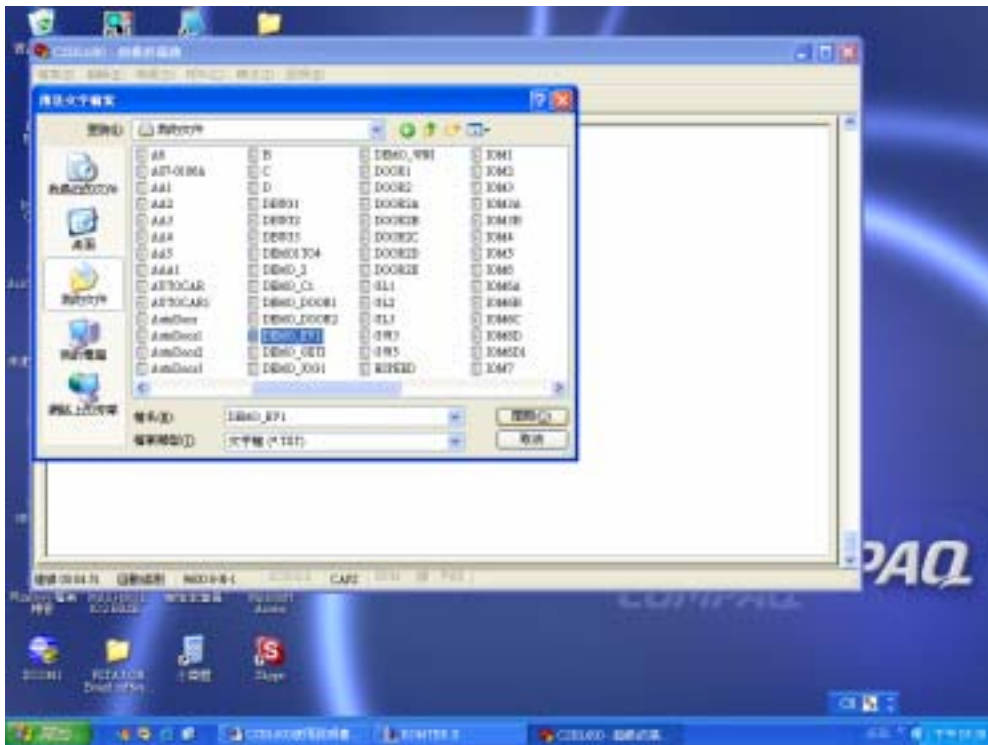
步驟 3

在“終端機”的功能表上點選“傳送”→“傳送文字檔案”



步驟 4

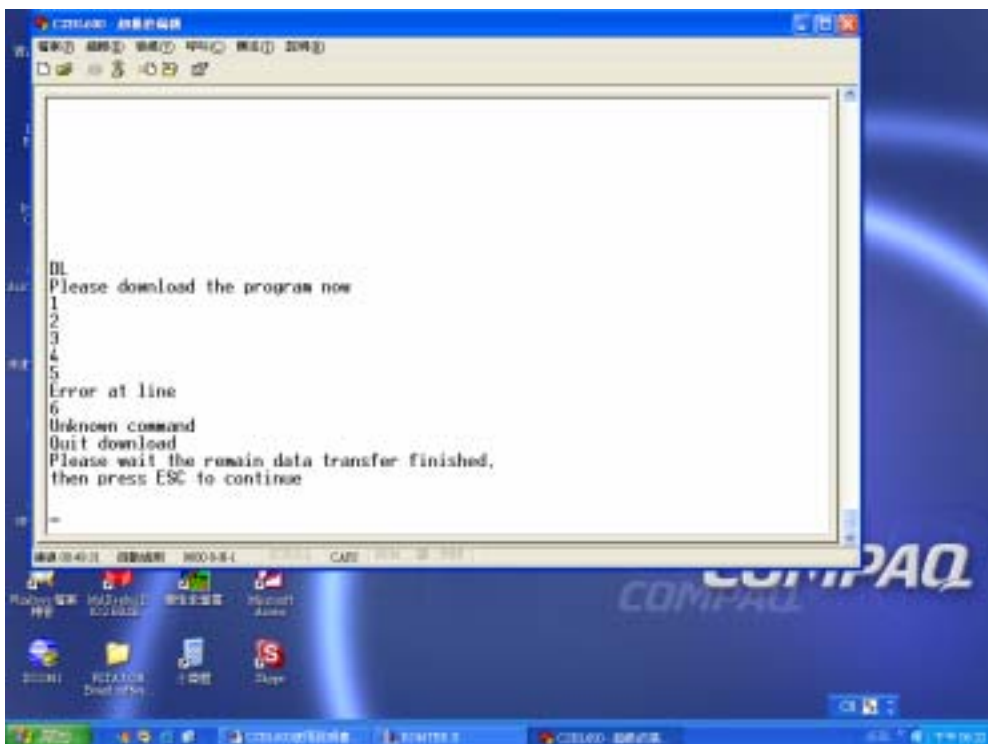
選擇所要下載的檔案，也就是之前由其它文字處理程式編寫好的程式，然後選按“開啓”。



步驟 5

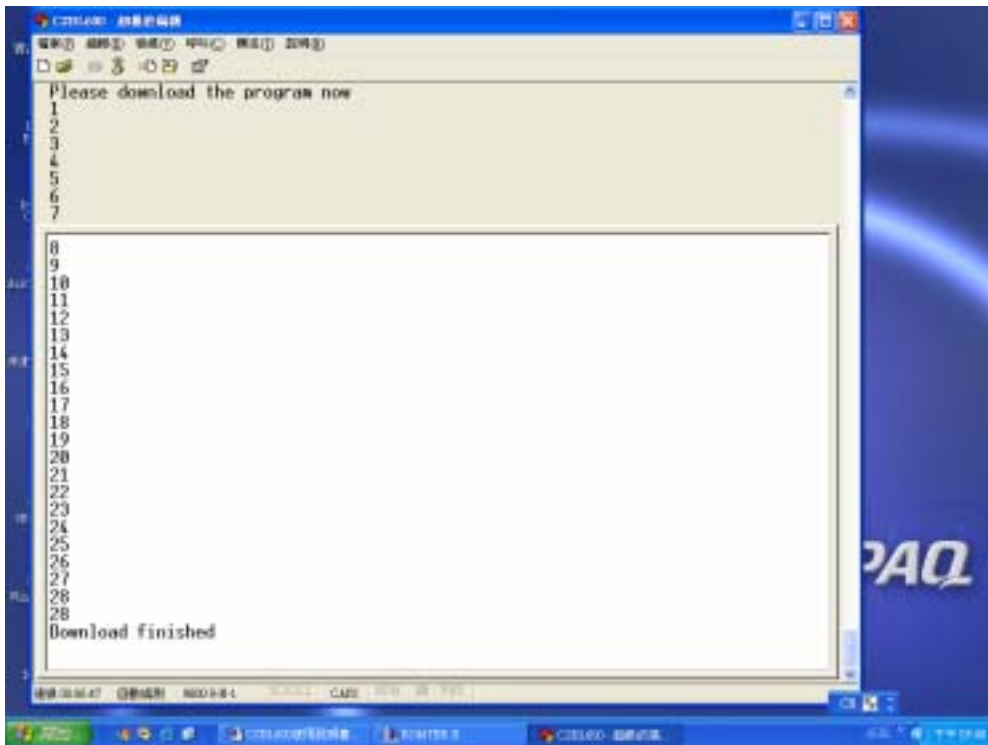
程式下載中，CSBL600 會顯示正在下載的程式行號。在程式下載過程中，若發現有程式或指令語法錯誤，也會顯示錯誤訊息。

在下面的例子中，就顯示出原始程式的第 6 行有錯誤。



步驟 6

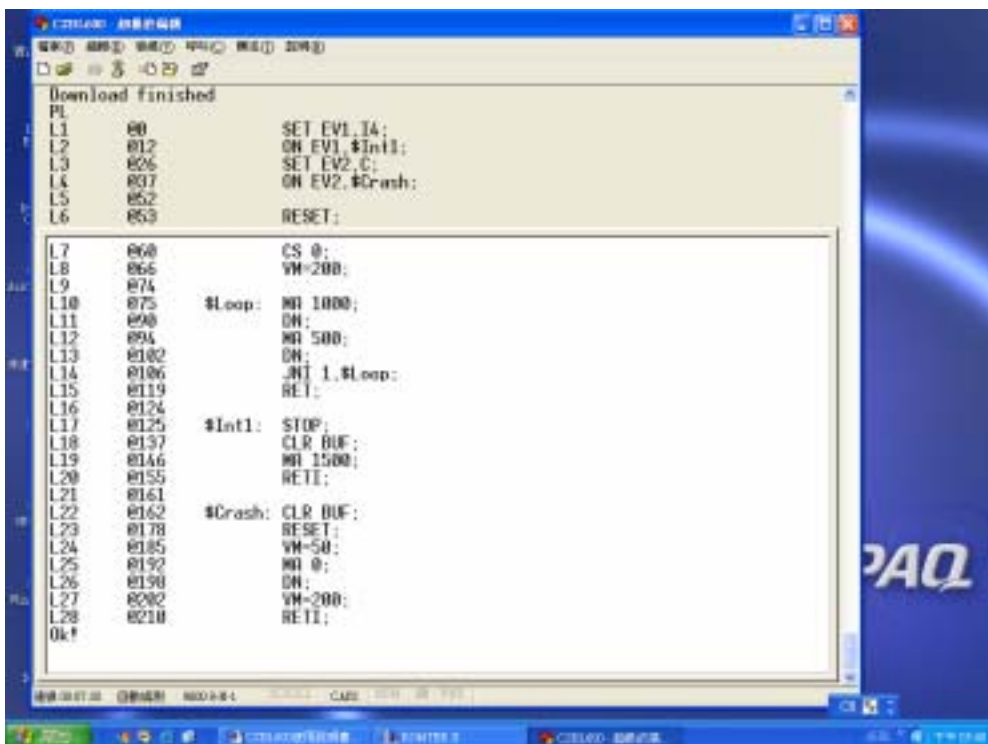
若沒有程式或指令語法的錯誤，最後當全部程式下載完畢時，CSBL600 會顯示 “Download finished”。



步驟 7

若想確認下載的程式，可在下載完畢後，以 “PL” 指令 (Program List) 要求 CSBL600 列示全部程式。

此時原程式中的註解或說明文字均已被濾除。



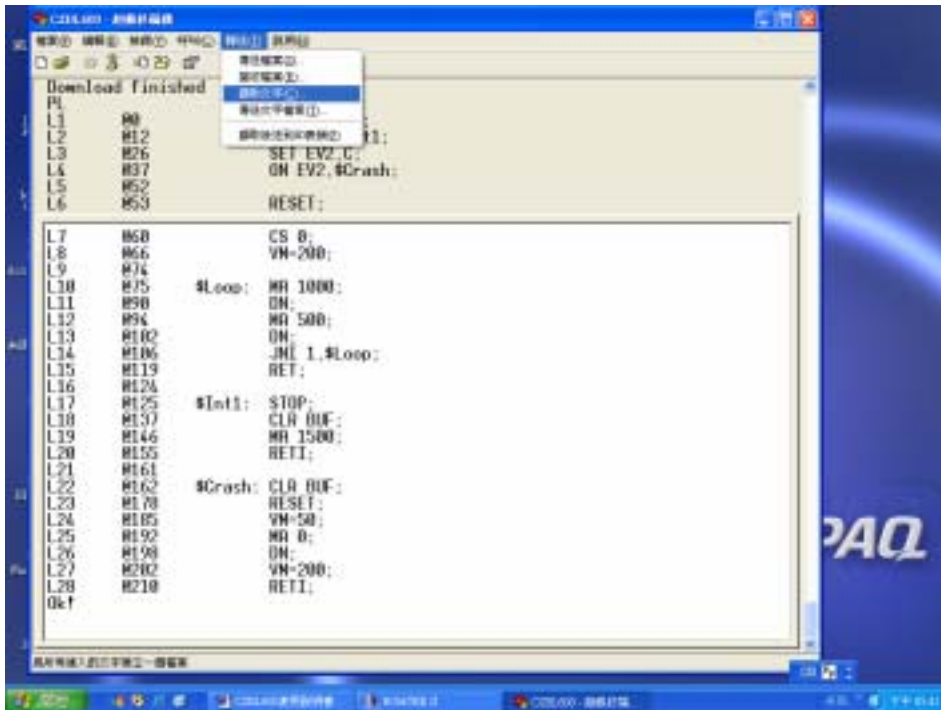
6-5. 上傳 CSBL600 的程式至 PC

利用 PC 的“終端機“，除了可以下載（Download）程式，將程式由 PC 下載列 CSBL600。也可以上傳（Upload）程式，也就是將 CSBL600 中的程式上傳至 PC，無論這程式是在 CSBL600 上直接編輯的或是之前由 PC 下載的。上傳至 PC 的程會以文字檔案型式儲存，可以再用其它文字處理程式(如視窗軟體所附的“Word Pad”或“記事本”等)整理，或存檔，或再下傳回 CSBL600 都很方便。

以下我們將逐步說明上傳程式的方法。

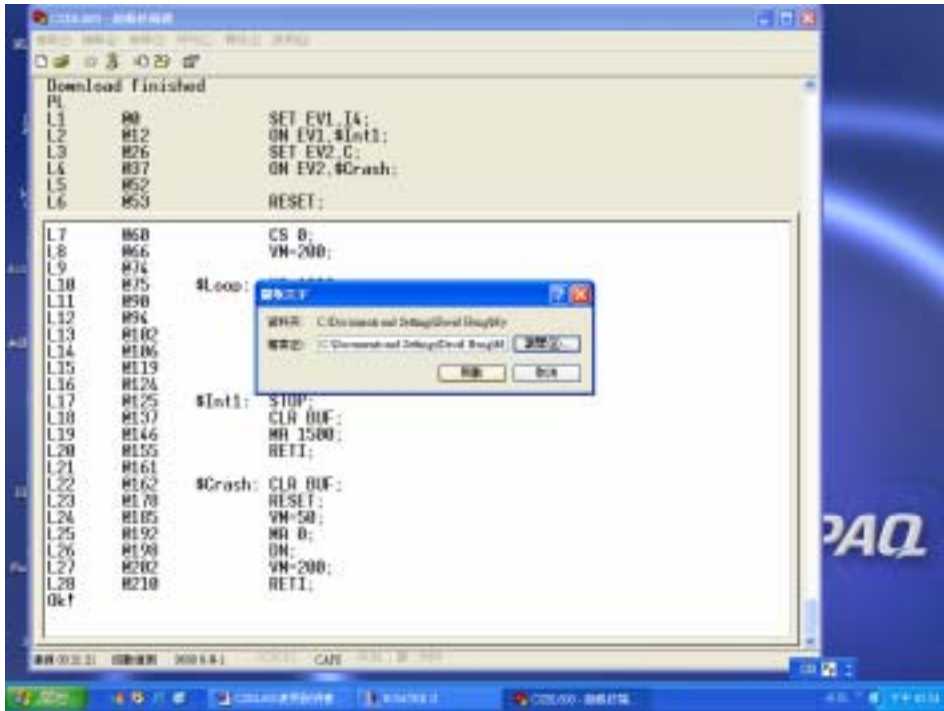
步驟 1

在“終端機“的功能表上選取“傳送” → “擷取文字”



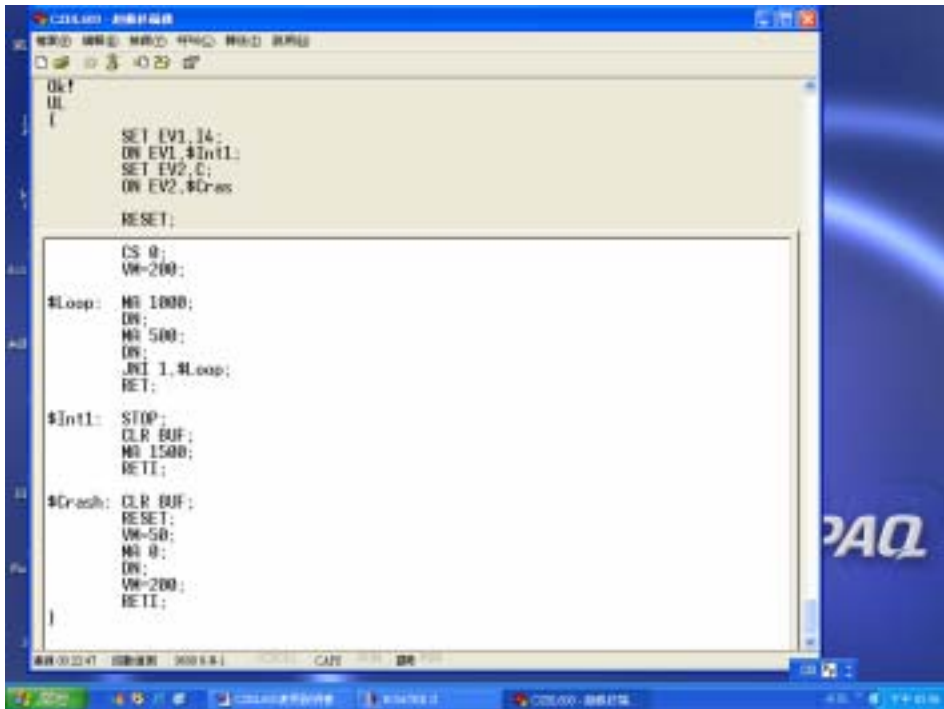
步驟 2

輸入儲存目錄及檔名，然後選按“啓動”。從這個時候開始，您在“終端機”上看到的任何訊息，都會被錄至您所選擇的檔案內。



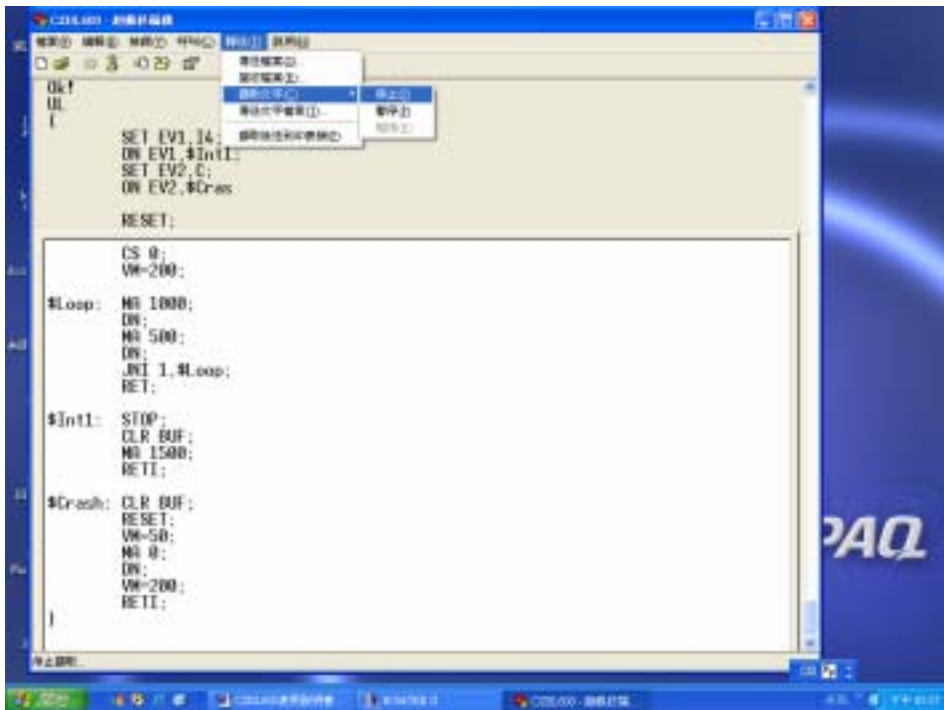
步驟 3

對 CSBL600 下“UL”指令 (Upload)，CSBL600 就會將全部指令以可以直接再下載的格式列示在“終端機”上。因之前我們已關啓了文字擷取功能，所以這些列示的程式也會同時被存進檔案。



步驟 4

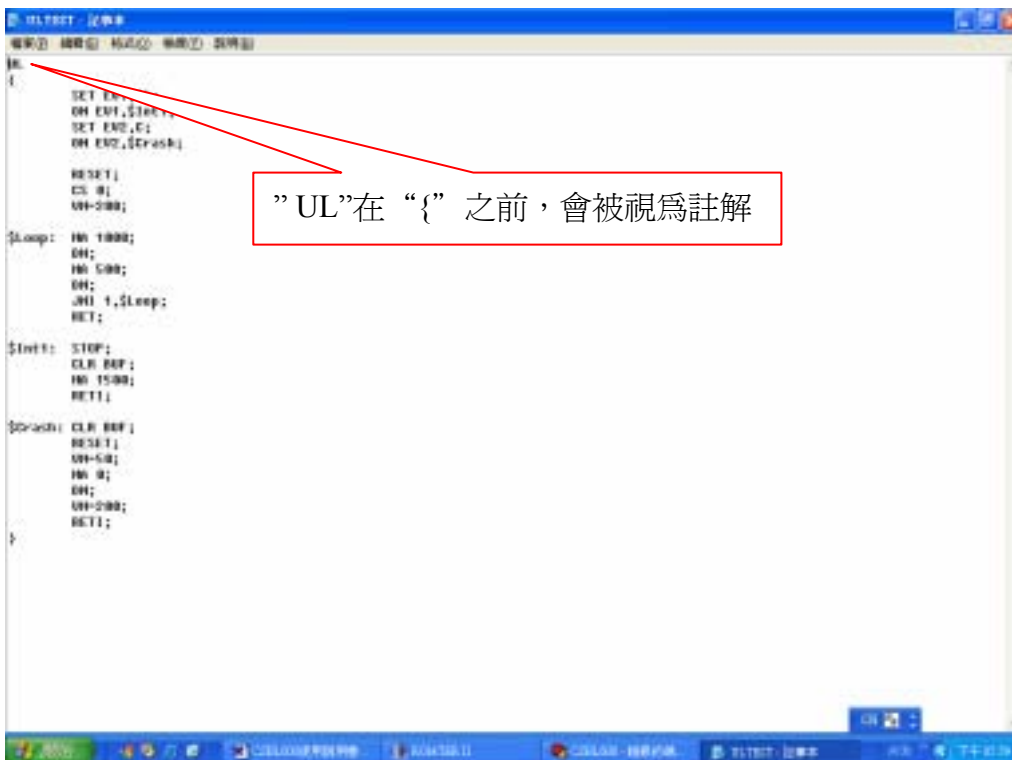
請立刻在“終端機“的功能表上選取“傳送” → “擷取文字” → “停止”，停止繼續擷取文字。否則在“終端機“上的任何訊息都還會繼續被存入檔案。



步驟 5

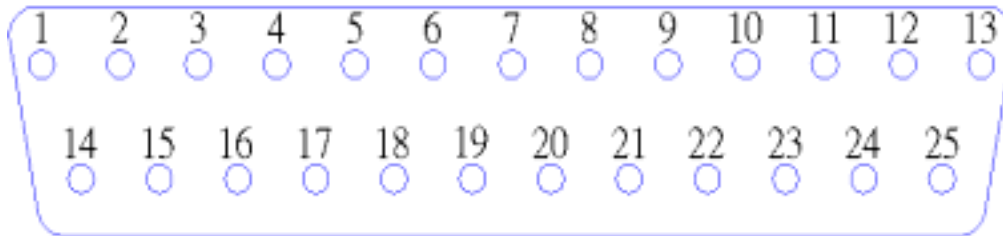
您可以用“記事本”（或其它文字處理軟體也都可以），打開剛才所選的目錄檔案，檢查程式是否已正確上傳。

在“記事本”上看到的程式格式，可以直接再下傳。因為在“{”之前及“}”之後的文字均會被視為註解而被忽略。



七、輸出入埠電路與配線

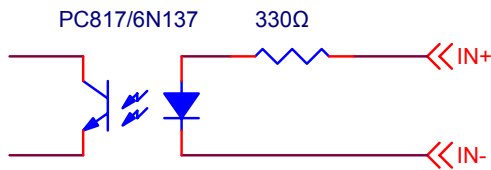
7-1. CSBL600 輸出入埠脚位定義



DB25 型接頭 (公)

Pin No.	信號名稱	功能				備註	
		控制器模式		驅動器模式			
1	IN1+	一般輸入埠 1		指撥開關	ON	脈波輸入	
14	IN1-			SW2	OFF	正轉脈波輸入	
2	IN2+	一般輸入埠 2		指撥開關	ON	方向輸入	
15	IN2-			SW2	OFF	逆轉脈波輸入	
3	IN3+	一般輸入埠 3		關閉馬達電流 (Hold Off)			
16	IN3-						
4	IN4+	一般輸入埠 4		重置(Reset)			
17	IN4-						
5	IN5+	一般輸入埠 5		強制關機 (Snap Off)			
18	IN5-						
6	IN6+	一般輸入埠 6		同控制器模式			
19	IN6-						
7	IN7+	EHC=0	一般輸入埠 7	同控制器模式			
20	IN7-	EHC=1	正極限輸入埠				
8	IN8+	EHC=0	一般輸入埠 8	同控制器模式			
21	IN8-	EHC=1	負極限輸入埠				
9	OUT0						
10	OUT1+	一般輸出埠 1		到位 (On Position)			
22	OUT1-						
11	OUT2+	一般輸出埠 2		跳機 (Crash)			
23	OUT2-						
12	OUT3+	一般輸出埠 3		備妥 (Ready)			
24	OUT3-						
13	OUT4+	PHZ=0	一般輸出埠 4	同控制器模式			
25	OUT4-	PHZ=1	PHZ 信號輸出				

7-2. 輸入埠電路



CSBL600 所有的輸入埠的內部電路均如上圖，使用光耦合器與外部隔離，並且在光耦合器的輸入電路上已串聯有一 330Ω 的電阻。由於光耦合器需要有 10~20mA 的電流才能致能 (Enable)，所以外部電路必須要有一 DC+5V 以上的電源，才能驅動 CSBL600 的輸入電路。

當外部電源電壓為 DC+5V 時，可直接驅動 CSBL600 的輸入電路。

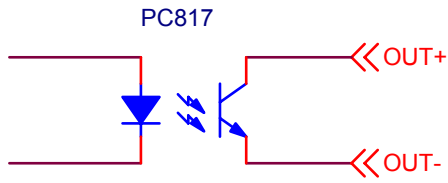
當外部電源電壓為 DC+12V 時，則必須串聯一 1KΩ 的電阻，才可接到 CSBL600 的輸入端 (請參考下節外部配綫範例)，否則電流超過 20mA 會將光耦合器燒毀。

當外部電源電壓為 DC+24V 時，則上述串聯電阻必須更加大為 2.2KΩ。

7-3. 輸入電路外部配綫範例

配合一般開關輸入配綫例	配合光遮開關(未含放大器)輸入配綫例
<p>控制器內部</p> <p>外部配綫範例</p>	<p>控制器內部</p> <p>外部配綫範例</p>
配合 TTL 輸入信號輸入配綫例	配合一般電晶體輸入信號輸入配綫例
<p>控制器內部</p> <p>外部配綫範例</p>	<p>控制器內部</p> <p>外部配綫範例</p>

7-4. 輸出埠電路

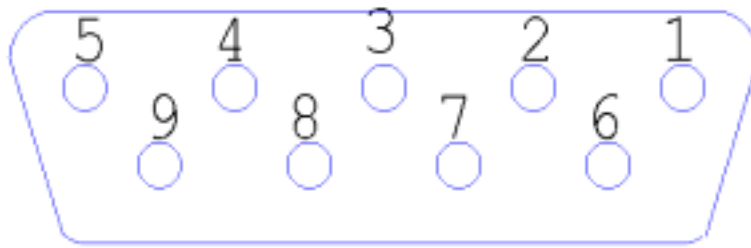


CSBL600 所有輸出埠的內部電路均如上圖，使用光耦合器與外部隔離。

7-5. 輸出電路外部配綫範例

配合 LED 或光耦合器輸出配綫例	配合小型繼電器 (<50mA 作動) 輸出配綫例
<p>控制器內部</p> <p>外部配綫範例</p>	<p>控制器內部</p> <p>外部配綫範例</p>
<p>配合一般繼電器輸出配綫例</p> <p>控制器內部</p> <p>外部配綫範例</p>	<p>配合 SSR 輸出配綫例</p> <p>控制器內部</p> <p>外部配綫範例</p>

7-6. RS232 接頭定義

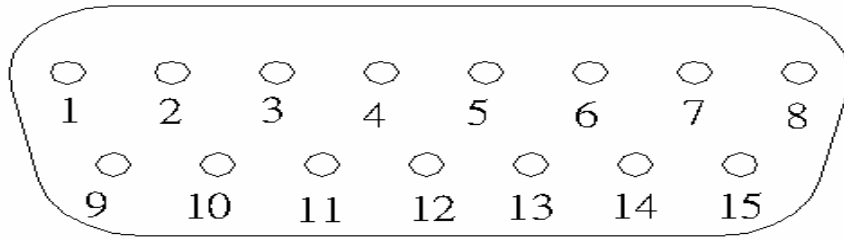


DB9 型接頭（母）

腳位	定義	資料方向	備註
Pin 1			
Pin 2	TXD	CSBL600→PC	傳送字元(Transmit)
Pin 3	RXD	PC→CSBL600	接收字元(Receive)
Pin 4	DSR	PC→CSBL600	資料備妥(Data Set Ready)
Pin 5	GND		地線(Ground)
Pin 6	DTR	CSBL600→PC	資料端備妥(Data Terminal Ready)
Pin 7			
Pin 8	RTS	CSBL600→PC	要求傳送(Request To Send)
Pin 9			

- CSBL600 與 PC 之 COM 埠相連使用 1 對 1 延長線即可。
- CSBL600 的 RS232 埠使用的參數如下
 每秒傳輸位元：9600
 資料位元：8
 同位檢查：無
 停止位元：1
- CSBL600 在正常狀態，DTR 及 RTS 均為高電位（+10V），表示此時 CSBL600 可以接收資料。但當 CSBL600 的 RS232 接收資料緩衝區(Data buufer)快要滿溢時，CSBL600 會將 DTR 及 RTS 改為低電位（-10V），表示這個時候 CSBL600 暫時已不能再接收資料。PC 側的軟體就必須立即停止傳送，否則資料就有可能因滿溢而發生錯誤。（一般而言，使用者若使用標準的“終端機”軟體與 CSBL600 連線，這些問題實際已被軟體自動處理，使用者並無需為此特別擔心。）

7-7 編碼器（ENCODER）接頭定義



DB15 型接頭（公）

腳位	定義	備註
Pin 1	+5V	
Pin 2	A	
Pin 3	B	
Pin 4	Z	
Pin 5	GND	
Pin 6	U	
Pin 7	V	
Pin 8	W	
Pin 9	A-	
Pin 10	B-	
Pin 11	GND	
Pin 12	Z-	
Pin 13	GND	
Pin 14	VCC	
Pin15	+V	未開放使用

八、使用 CSBL600 注意事項

8-1. 安裝場所請注意事項

- 室內
- 周圍溫度 0°C ~ 40°C
- 周圍溼度 85% 以下
- 不可有過度震動或衝擊
- 使用於通風、散熱良好的工作場所
- 避免在有水、油、腐蝕性氣體、塵埃等工作場所使用
- 當安裝使用在控制箱等密閉式場所，或接近發熱體的地方，請注意驅動器的溫昇。
- 請勿接近大的雜訊發生源（高週波溶接機、高壓機械、大型電器開關等）。
- 請勿讓有導電性的小碎片（碎屑、引線、電線渣、等....）跑進驅動器。

8-2. 配綫請注意事項

- 要插拔馬達或編碼器的接頭時，務必要先關閉 CSBL600 的電源。
- CSBL600 的外殼請勿接觸任何電器或帶電裝置。
- 馬達外殼（Connector 的第 4 綫）及 CSBL600 電源接頭的中性綫（Connector 的第 2 綫）需接地。（但馬達外殼與 CSBL600 的外殼請勿接觸）
- 馬達和 CSBL600 間的配綫長度不可超過 5 米。
- 馬達綫和編碼器信號綫，不可以互絞或長距離平行配置，應儘可能互相隔離。
- 編碼器綫及 RS232 信號綫應儘可能避開各種雜訊發生源，如變壓器、馬達、電磁鐵及這些裝置的電源綫。