SMART Motor

快速上手

SMARTMOTOR 伺服馬達/控制器

應用說明書

MONTROL 敏石系統有限公司

www.montrol.com.tw

330桃園市同德11街58號10樓之2　tel:03.3586008　fax:03.3586009

e-mail:info@montrol.com.tw

單軸RS232通訊

SmartMotor Combo接頭　　電腦RS232接頭

7 Pin Combo D-Sub connection

A1 +20V to +48V DC※
A2 Power Ground
1 Sync or I/O
2 +5V Out
3 RS232 Transmit (Tx)
4 RS232 Receive (Rx)
5 RS232 Ground (Gnd)

9PIN RS232 connection

2 RS232 Receive (Rx)
3 RS232 Transmit (Tx)
5 RS232 Ground (Gnd)

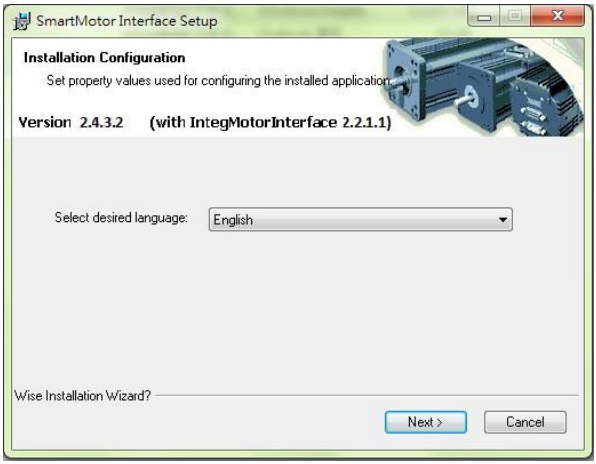
※注意電源正負極不可接反
※高速運轉之應用需使用48V 電壓輸入

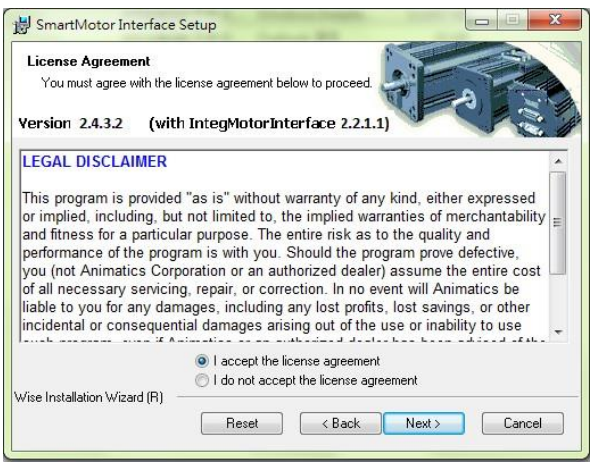
安裝程式

SMIInstall.exe

請執行**SMI**安裝檔並依照選單依序
執行下一步即可完成安裝

※**SMI**安裝檔可於敬石系統網站下載

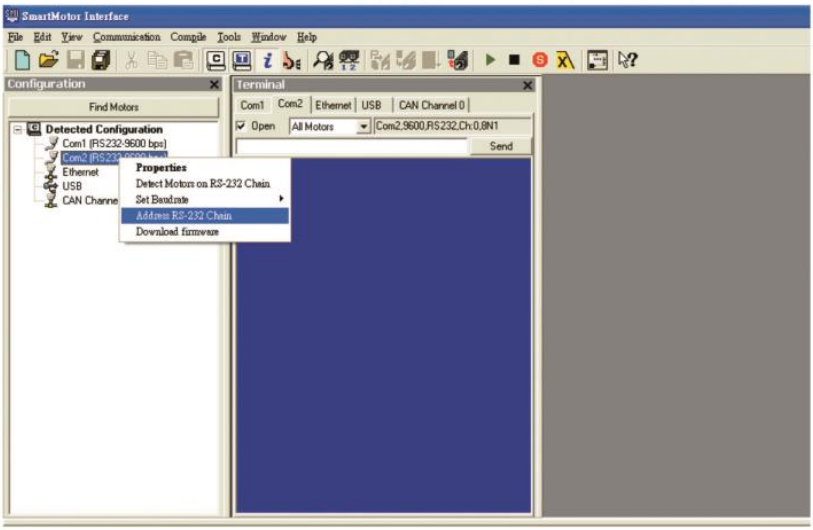




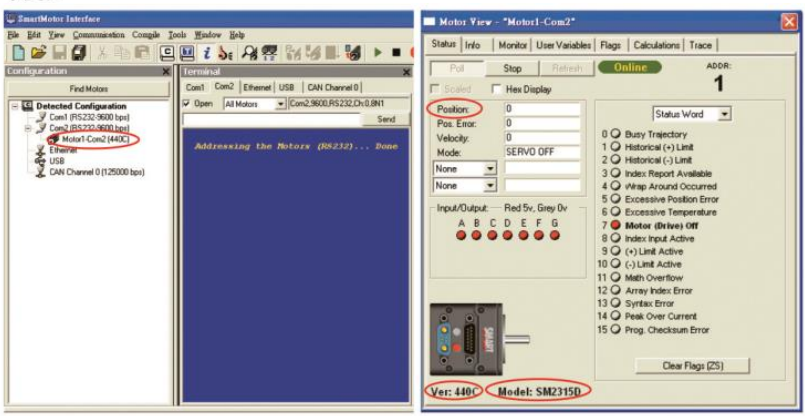
SMI : Smart Motor Interface

建立通訊

建立PC與SmartMotor通訊為使用SMI軟體的第一步驟



在Configuration視窗內找到已連接的ComPort按右鍵>再按Address RS-232 Chain

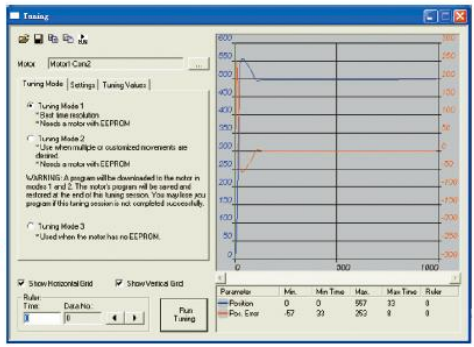


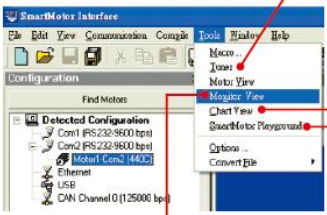
若通訊成功將出現馬達圖示
此時表示硬體接線及軟體安裝
皆為正確

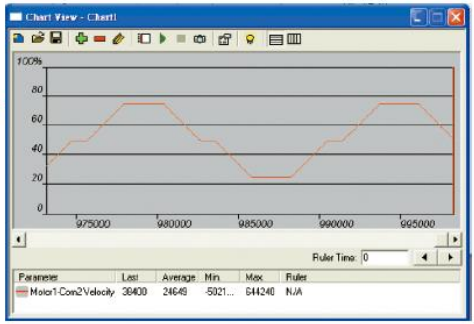
連按馬達圖示兩次，將顯示出
SmartMotor主要資訊，包括
位置、型號及韌體版本

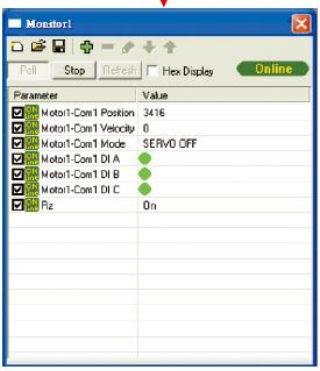
使用工具

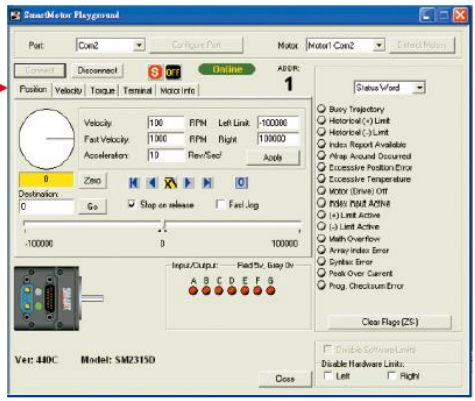
SMI Tools工具程式
具有多樣及實用功能
使用者可依需求應用
進階操作說明請參考
SMI HELP....



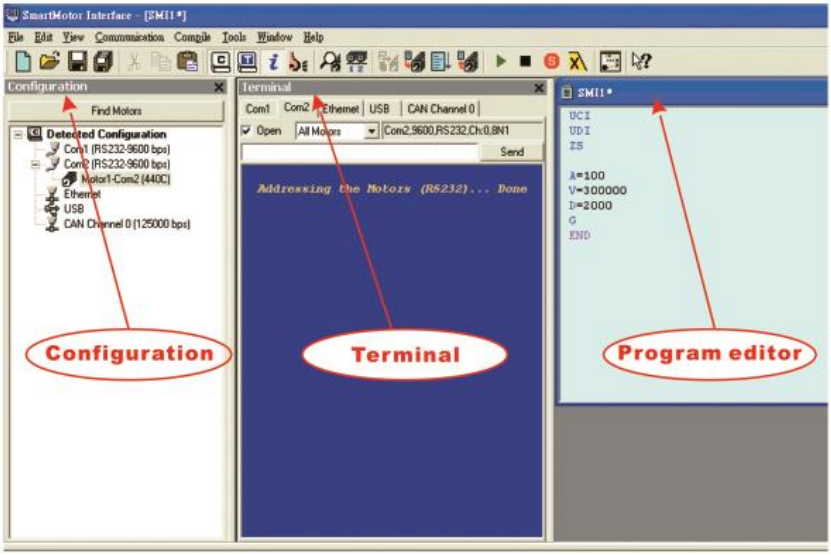








SMI標準控制介面



Configuration

經由自動搜尋後，由此狀態欄可得知所有與電腦連接的馬達狀況。

Terminal

此視窗讓使用者透過即時且直接的指令控制馬達。

Program editor

此視窗讓使用者編輯將傳送到馬達的程式。

基本指令操作

將指令傳送至馬達

--請於 Terminal 視窗下執行--

傳送控制指令至馬達

```
EIGN(2)          ‘取消正極限
EIGN(3)          ‘取消負極限
ZS               ‘清除異常
MP               ‘設定為位置模式(程式預設模式，也可以不輸入)
ADT=400          ‘設定加減速為400（1rev/sec^2=>4，預設值0)
VT=32768         ‘設定速度為32768（1rps=60rpm=>32768，預設值0)
PRT=4000         ‘設定相對位置為4000（1rev=>4000，預設值0)
G                ‘馬達動作開始
RPA              ‘回報馬達目前實際位置
O=0              ‘將目前位置設為0
RPA
PT=12345         ‘設定絕對位置為12345
VT=327680        ‘設定速度為327680(10rps)
G
RPA
MV               ‘設定為速度模式
G
VT=VT/2          ‘設定目標速度為目前速度的一半
G
VT= -VT          ‘設定馬達反轉
G
X                ‘馬達減速到停止
MT               ‘設定為扭力模式
T=1600           ‘設定扭力為1600（馬達最大扭力為±32767)
G
T=1800
G
T= -T
G
T=1300
G
X                ‘馬達減速到停止
a=123            ‘設定變數a數值為123
Ra               ‘回報變數a的數值
b=a+2            ‘設定變數b數值為a加2
Rb               ‘回報變數b的數值
OFF              ‘關閉馬達伺服控制
Z                ‘RESET,相當於關閉電源後再開機
```

--請於 Program editor 視窗下執行--

編輯控制程式並下載於馬達

```
EIGN(2)  EIGN(3)  ZS     '取消正負極限功能
AT=1000                  '設定目標加速度值
DT=600                   '設定目標減速度值
VT=30*32768              '設定目標速度值為30rps=1800rpm
PRT=20*4000              '設定目標(相對)位置值為20rev

a=0                      '設變數a為0
WHILE a<3                'WHILE之後判斷若為真,則執行與LOOP間指令
a=a+1                    '設定a值增加1
G                        '執行馬達動作
TWAIT                    '等待馬達動作停止,再繼續下一指令
WAIT=1000                '等待時間1秒=1000ms
LOOP                     '回到相對應WHILE指令

PRT=1000
b=0
WHILE b<6
b=b+1
G
TWAIT
WAIT=150
PRT=-PRT                  '設定目標位置反轉
LOOP                      '以下...可用空白鍵區隔指令
WAIT=2000  VT=VT*2  PT=0  G  TWAIT  WAIT=1000  RPA
END                       '程式結束
```

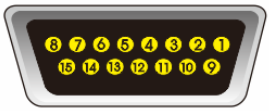
將程式下載於馬達

傳送至馬達後，如右圖所示，執行Run，或是重新啓動電源 或按Reset，馬達就會依程式自動執行。

I/O 控制與應用

I/O的配置說明

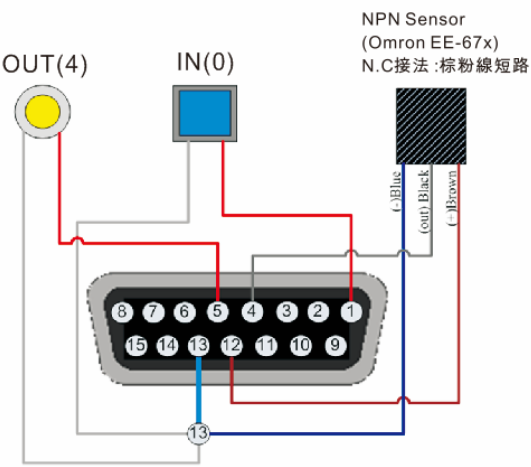
SmartMotor具有7個接點(I/O 0~6)
提供使用者作數位輸入(DI)、輸出(DO)
以及類比輸入(AI)的應用，以下是
SmartMotor 15 PIN接點的介紹：



15 PIN D-Sub I/O	
1 I/O 0	9 Encoder B Out
2 I/O 1	10 SM RS-232 Transmit
3 I/O 2 ※	11 SM RS-232 Receive
4 I/O 3 ※	12 +5V Out
5 I/O 4	13 Ground
6 I/O 5	14 Power Ground(A2)
7 I/O 6 ※	15 Main Power(A1)
8 Encoder A Out	

※ I/O 2 **預設值為正極限接點** (面向馬達軸心CW方向)
※ I/O 3 **預設值為負極限接點** (面向馬達軸心CCW方向)
※ I/O 6 **預設值為執行動作接點** (相當於G指令)
※ **警告**: SmartMotor**內部** I/O **為**5V , **請勿輸入**24V**電壓電源**

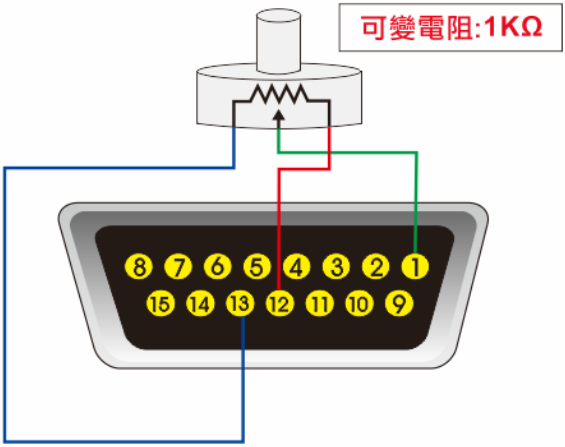
接線範例如右圖



I / O 的應用介紹

```
EIGN(2) EIGN(3) ZS
GOSUB0                          '執行副程式0
WHILE 1                         '無窮迴圈
GOSUB1                          '執行副程式1
LOOP
END                             '程式結束
----------副程式0 / 原點覆歸----------
C0
SLD                             '取消韌體極限功能
MT T=-1800 G                    '執行扭力控制尋找機械邊緣點
e=100
WHILE e>10
q=PA
WAIT=100
w=PA
WAIT=100
e=q-w
LOOP
MP ADT=500 VT=327680 PRT=4000 G TWAIT WAIT=500
O=0 SLN=-1000  SLP=41000  SLE   '設原點及韌體極限範圍
RETURN
  ----------副程式1 / 主動作程式----------
C1
PRT=8000
b=0
WHILE b<5
IF IN(0)==0                      ‘判斷Input(0)是否為0(與Pin13Gnd接觸)
b=b+1                            ‘若為真則執行與ENDIF間指令
G TWAIT
GOSUB2
ENDIF
LOOP
WHILE IN(0)==1 LOOP              ‘執行WHILE..LOOP迴圈直到條件不成立
VT=VT*2 PT=0 G TWAIT
GOSUB2
GOSUB2
RETURN
  ----------副程式2 / 到位輸出----------
C2
OS(4)                            ‘Output(4)輸出5V(燈亮)
WAIT=500                         ‘等待時間500ms=0.5sec
OR(4)                            ‘Output(4)=>0V(燈滅)
WAIT=500
RETURN
```

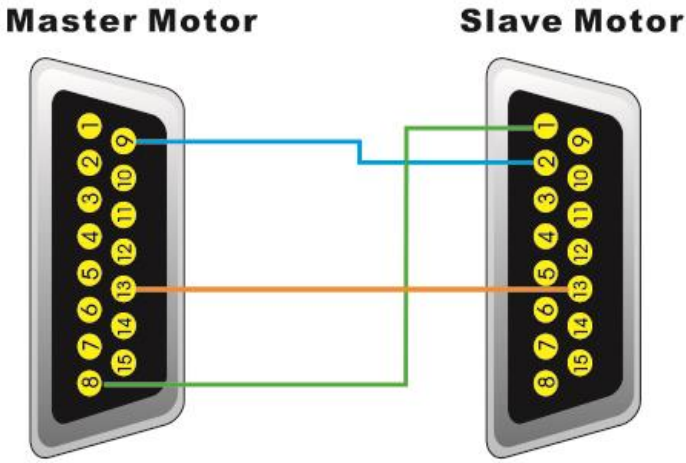
類比輸入應用範例



運用類比輸入改變馬達位置

```
EIGN(2)  EIGN(3)  ZS
VT=100000                 ‘設定目標速度
ADT=1000                  ‘設定目標加減速
WHILE 1                   ‘無窮迴圈
  b=INA( A, 0)            ‘設定Input(0)為類比型態,
                           並將值存為變數b
  PT=b                    ‘設定目標位置為變數b
  G                       ‘執行動作
LOOP                      ‘回到WHILE指令
END                       ‘程式結束
```

馬達跟隨運動與電子齒輪的應用



Master 馬達內部 Encoder 輸出接腳為 8 與 9，
Slave 馬達接收外部 Encoder 的接腳是 1 與 2。

運用Encoder執行馬達跟隨運動

```
EIGN(2) EIGN(3) ZS
MFR                 ‘設定為跟隨模式
MFMUL=2             ‘設定跟隨比例:分子
MFDIV=21            ‘設定跟隨比例:分母
G                   ‘執行馬達跟隨運動

END                 ‘程式結束
```

基本程式流範例

IF ELSEIF ELSE ENDIF structure:

```
IF a<b
    PRINT (“a is less than b”,#13)
ELSEIF q==123
    PRINT (“q equals 123”,#13)
ELSE    ‘if no condition above was true
   PRINT (“nothing above was true”,#13)
ENDIF
```

WHILE LOOP structure:

```
a=0
 WHILE a<10
    a=a+1
  LOOP
  PRINT (“loop code executed 10 times”,#13)
```

GOTO, GOSUB structure:

```
 C1
IF a>b
    GOTO (1)
ELSEIF b>c
    GOSUB (5)
ENDIF
    GOTO (6)
C5
 PRINT(“b is greater than c”,#13)
RETURN
C6
END
```

SWITCH CASE BREAK structure:

```
 SWITCH a
    CASE 1
        PRINT(“A=1”,#13)
    BRAKE
    CASE 2
       PRINT(“a=2”,13)
    BRAKE
    DEFAULT
   PRINT(“A DOES NOT EQUAL 1 OR 2 “,#13)
    BRAK
 END
```

運用扭力控制模式

```
EIGN(2)
EIGN(3)
SLD
ZS
WHILE 1
   WHILE IN(1)==0
      ADT=500
      VT=327680
      PT=80000
      G
      TWAIT
      WAIT=1000
      OUT(4)=0
      MT
      T=1600
      q=100
      WHILE q>4
         w=PA
         WAIT=100
         e=PA
         q=e-w
      LOOP
      MP
      WAIT=3000
      OUT(4)=1
      ADT=250
      VT=6103
      PT=0
      G
      TWAIT
      WAIT=1000
   LOOP
LOOP
END
```

定位控制完成後，切換為
扭力模式，當馬達運動受到
阻礙(Encoder值不變)時
停住並施以固定扭力及
特定時間，再轉為定位模式
回到原點

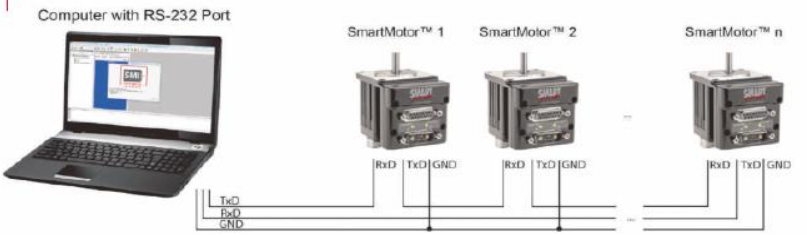
類比控制方向速度及位置教導

```
EIGN(2)
EIGN(3)
SLD
ZS
ADT=400
i=1
WHILE 1
   GOSUB(1)
   GOSUB(2)
   GOSUB(3)
LOOP
END
C1
IF IN(6)==0
   MV
   WHILE IN(6)==0
      h=INA(A,2)-16384
      VT=h*10
      G
   LOOP
   OFF
ENDIF
PETURN
C2
IF IN(5)==0
   CLK=0
   WHILE IN(5)==0
   LOOP
   IF CLK>2000
      IF i==1
         a=PA
      ELSE
         b=PA
      ENDIF
      i=-1
   ENDIF
ENDIF
RETURN
C3
IF IN(3)==0
   MP
   VT=600000
   IF i==1
      PT=a
   ELSE
      PT=b
   ENDIF
   i=-i
   G
   TWAIT
ENDIF
RETURN
```

C1:讓馬達產生位移，亦可用手推動
C2:將位置點儲存於a,b變數
C3:執行a,b點來回運動

多軸控制接線及定址

RS232 串接



9Pin RS232
2 RS232 Receive (Rx)
3 RS232 Transmit (Tx)
5 RS232 Ground (Gnd)

7 Pin Combo D-Sub
A1 +20V to +48V DC
A2 Power Ground
1 Sync or I/O
2 +5V Out
3 RS232 Transmit (Tx)
4 RS232 Receive (Rx)
5 RS232 Ground (Gnd)

RS485 並接

RS232485轉接器	
1	+5V
2	GND
3	RS-485A
4	RS-485B

15 Pin D-Sub	
12	+5V out
13	Ground
5	I/O E
6	I/O F

※ RS485A連接 I/O#4，RS485B連接 I/O#5

※ RS485B與Ground之間需連接pull-down電阻(500Ω)

※ 如果最後一顆SmartMotor距離很遠，可能必需於RS485A與B之間連接shunt電阻(250Ω)

※ 採用雙絞線金屬隔離網，並且僅一端與Hoset Gnd接地，有較佳的防干擾作用

※ RS232/485轉接器之5V電源，最好由外部(非馬達)電源供應

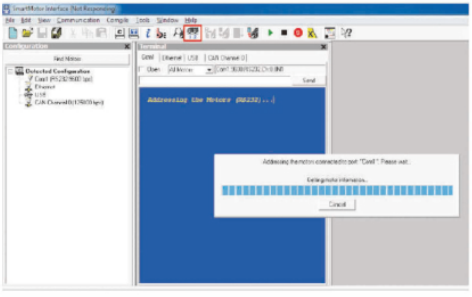
定址 (Addressing)

目的：為了能夠將通訊的指令傳達到指定的馬達，所以必須先定義每個馬達在整個系統中的位址。

方法： SmartMotor **提供兩種方式來定義馬達位址：**

1) **自動定址**(Auto-Addressing)：**每次開機後,系統依照接線順序自動做定址**
2) **預設定址**(Pre-Addressing) ：**預先將位址儲存至每一個馬達程式內**

自動定址(Auto-Addressing)

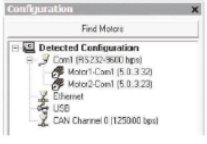


執行SMI功能表上的按鈕

(Addressing)

馬達位址以馬達和電腦之間的接線順序自動決定，最靠近電腦的為馬達1，其次為馬達2，依此類推。

當完成後，系統將自動偵測，馬達狀態將會顯示如右圖



預設定址(Pre-Addressing)

我們依序將下列指令儲存在馬達內：

```
SADDRn   '預先設定馬達位址為n (n=1,2,3,..)
ECHO     開啓回應※
END
```

※ 使用RS-232多軸通訊，全部的馬達必須開啓回應(ECHO)

※ 使用RS-485多軸通訊，全部的馬達必須關閉回應(ECHO OFF)

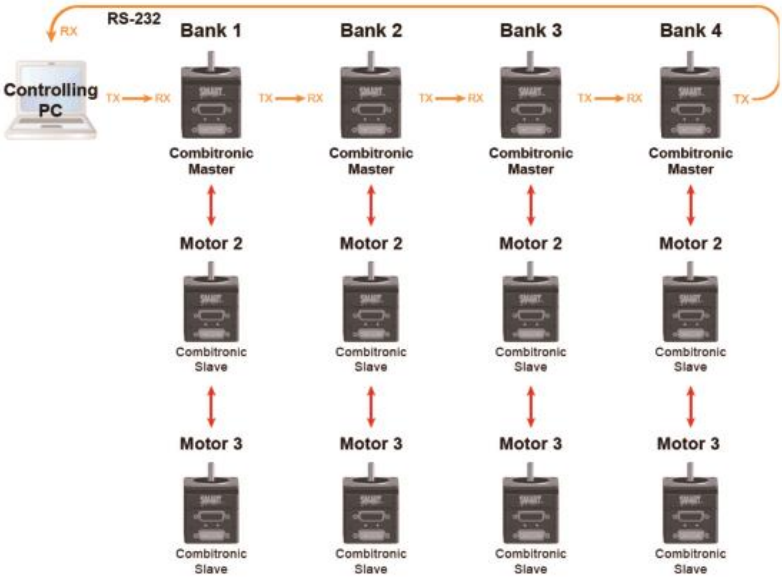
重新啓動電源 (Reset)後，不需再經由自動定址的步驟，即可完成定址的動作。

多軸指令

CANBUS多軸指令

```
2PT:3=1234          'Bank2中的Motor3目標位置設定  為1234
3PT:0=0             'Bank3全部的目標位置設定為0
4PT=345             '只Bank4的第一顆馬達目標位置  設定為345
0G                  '所有Bank的第一顆馬達執行G指令
0G:0                所有的馬達執行G指令

VTS=100000          設定多軸補間速度
ATS=1000            設定多軸補間加速度
DTS=100             設定多軸補間減速度
PTS(123000;1,20000;2)  設定多軸補間位置
GS                  執行多軸補間
```



多軸控制範例

```
0EIGN(2)
0EIGN(3)
0ZS
0MP              ‘設定所有馬達為位置模式(程式預設模式)，也可不輸入
0ADT = 500       ‘設定所有馬達加速度為500(4 = 1rev/sec2)
0VT = 32768      設定所有馬達速度為32768(32768 = 1rps = 60 rpm)
0PRT = 4000      ‘設定所有馬達相對位置為4000(4000 = 1rev)
0G               所有的馬達動作開始
1RPA             ‘回報馬達1的位置
2RPA             ‘回報馬達2的位置
0O = 0           ‘將所有的馬達目前位置設為原點
1MV
1VT = 300000
1ADT = 25
1G
1X
1VT = -100000
1G
1X
2MT
2T = 1600
2G
2X
2T = 1280
2G
```

※更多的範例程式可於原廠網站
www.SmartMotor.com下載

※圓弧補間運動須以PC通過.dll 檔執行,請參考 SMIEngine 說明檔內容

SmartMotor常用指令

Reference Key:

- is the IO Bit Number

m - is the mask value of which bits are affected

W - defines it as a word (16 bits)

expression - an expression must contain no more than a total maximum of 32 operators, values, and parenthesis.

value - a number, variable or math expression with one operand

constant - means a fixed integer

gen# Trajectory generator number: 1 or 2

i - Interrupt number , valid values are from 0 to 7

Communication Commands:

ADDR=expression Set motor's serial communications address. Applies for both RS232 and RS485

BAUD(x)=y This allows for COM0 or COM1 to be changed, x is the channel (0 or 1) and y is baud rate

CADDR=expression Set CAN address, can be different from serial address, default is 63

CBAUD=expression Set CAN baud rate, default is 125000

CCHN(RS2,0) Close communication channel command

ECHO Must be used to insure all data received in one motor will be echoed to next motor

ECHO_OFF Default, turn communication's echo off

GETCHR Get the next character from channel 0

GETCHR1 Get the next character from channel 1

LEN Number of characters in channel 0 buffer

LEN1 Number of characters in channel 1 buffer

OCHN(RS2,0,N,9600,1,8,C,1000) Default: (RS232,chan=0, no parity, 9600 baud,1 stopbit, 8 databits, command,1000 ms timeout)

PRINT("Hello World",#13) Print command to say "Hello World", see print section for more detailed examples

PRINT1("Hello World",#13) Print command to say "Hello World" on channel 1, see print section for more detailed example

RCADDR Reports CAN address

RCBAUD Reports CAN baud rate

SILENT Ignore print commands to channel 0 from user program

SILENT1 Ignore print commands to channel 1 from user program

SLEEP Ignore commands for channel 0 except the WAKE command

SLEEP1 Ignore commands for channel 1 except the WAKE command

TALK Enable prints for channel 0 from user program

TALK1 Enable prints for channel 1 from user program

WAKE Wake for channel 0

WAKE1 Wake for channel 1

Program Flow Commands:

CASE expression Switch case statement

C constant Subroutine label, e.g. C10 for subroutine 10, must have a RETURN for each C label

DEFAULT Default action for switch case statement

DITR(i) Individual interrupt disable

EITR(i) Individual interrupt enable

ELSEIF expression Used for IF statements to test another condition, if expression is true, then execute code

END End program execution

ENDIF End statement for IF code structures

ENDS Command for end of switch case statement

GOSUB(value) Call a subroutine, value up to 999

GOTO(value) Jump program execution to a label, value up to 999

IF expression Conditional Test, expression can be multiple math operations

ITR(i, status_wrd#, bit#, s ,label#) Interrupt setup

ITRD Global interrupt scanner disable

ITRE Global interrupt scanner enable

LOOP Loop command for while loops

PAUSE Pause program execution, used for interrupts

RESUME Resume program execution

RETURN Return from subroutine

RETURNI Return from interrupt

RUN Start program execution

RUN? Wait at this point for RUN command before program starts to execute

STACK Resets all GOSUB stack returns and Interrupts

SWITCH expression Switch case statement

TWAIT Wait for trajectory to complete, only used in program

TWAIT(gen#) Wait for trajectory generator (gen#) to complete it's move

TSWAIT Wait for synchronized trajectory to complete, down loaded program only *COMBITRONIC*

WAIT=expression Set wait time in milliseconds

WHILE expression

I/O Commands:

EIGN(#) Assign a single I/O point as general use input

EIGN(W,0) Assign all local I/O as general use inputs

EIGN(W,0,m) Assign a masked word-sized set of local I/O as general use inputs at once

EILN Set port C (I/O-2) as negative over travel limit

EILP Set port D (I/O-3) as positive over travel limit

EIRE Set I/O 6 to capture external encoder's current value

EIRI Set I/O 6 to capture internal encoder's current value

EOBK(#) Configure a given output to control an external brake

IN(#) x=IN(#), assign the state of a specific I/O to a variable (x in this case)

IN(W,0) x=IN(W,0), assign the state of the first word of local I/O to the variable x

INA(A,#) x=INA(A,#), raw analog reading: 10 bit resolution spanned over signed 16 bit range

INA(V,#) x=INA(V,#), input voltage in millivolts of analog input value for a given I/O defined by #

INA(V1,#) x=INA(V1,#), scaled 0-5 VDC reading in millivolts directly, 3456 would be 3.456 VDC

OR(value) Reset output (turn off)

OS(value) Set output (turn on)

OUT(#)=expression if expression LSB = 1, then it's true(1), otherwise it's false (0)

Math Commands:

- Subtract

!| Bitwise exclusive OR

!= Not equal to

% Modulo (remainder) division

& Bitwise AND

***** Multiply

/ Divide

^ Power limited to 4th power and below, integers only

| Bitwise inclusive OR

+ Add

< Less than

<= Less than or equal to

== Equal to

> Greater than

>= Greater than or equal to

ABS(value) Absolute Value

ACOS(value) Arc Cosine

ASIN(value) Arc Sine

ATAN(value) Arc Tangent

COS(value) Cosine

FABS(value) Floating point absolute value

FSQRT(value) Floating point square root

RANDOM=expression Set the random seed value 0 to 2^31 - 1

RRANDOM Report the next available random number in the range 0 to 2^31 -1

SIN(value) Sine

SQRT(value) Square Root

TAN(value) Tangent

TMR(x,t) Sets timer x for t milliseconds

Motion Commands:

ADT=expression Set the accel/decel at once for a move *COMBITRONIC*

ADTS=expression Set sync accel/decel at once for a move

Ai(0) Arm index rising edge of internal encoder

Ai(1) Arm index rising edge of external encoder

Aij(0) Arm index rising edge then falling edge internal encoder

Aij(1) Arm index rising edge then falling edge external encoder

Aj(0) Arm index falling edge of internal encoder

Aj(1) Arm index falling edge of external encoder

Aji(0) Arm index falling edge then rising edge internal encoder

Aji(1) Arm index falling edge then rising edge external encoder

AMPS=expression Current limit value. 0-1023

AT=expression Set the acceleration target for a move

ATS=expression Set sync acceleration target for a move *COMBITRONIC*

BREAK Break out of while loop

BRKENG Manually Engage the brake

BRKRLS Manually Release the brake

BRKSRV Brake Servo, engage the brake when the drive is not active (default)

BRKTRJ Brake Trajectory

CTR(0) Present value of internal encoder

CTR(1) Present value of external encoder

DEL=expression Set maximum allowable derivative error limit

DT=expression Set the deceleration target for a move

DTS=expression Set sync deceleration for

COMBITRONIC **These commands require Combitronic with -C or -DN product configuration option to execute.**

a move *COMBITRONIC*

EL=expression Set maximum allowable following error limit

ENC1 Enable external encoder for servo

ENC0 Enable internal encoder for servo

F Set tuning values

G Go, initiates all buffered modes of operation

G(gen#) Go, initiate motion in trajectory generator (gen#)

GS Go synchronized, initiates linear interpolated moves *COMBITRONIC*

KA=expression Feed forward gain

KD=expression Derivative gain coefficient

KG=expression Gravity offset

KI=expression PID integral gain

KL=expression PID integral limit

KP=expression PID proportional gain

KS=expression Differential sample rate

KV=expression Velocity feed forward gain

MC Initiate electronic camming

MC(2) Set Trajectory Generator 2 to run in electronic camming

MDB Enable TOB when in one of the 2 trapezoidal modes

MDE Set motor to enhanced trapezoidal mode communtation by using encoder

MDS Set motor to sine mode commutation

MDT Set motor to trapezoidal mode communtation using hall sensors (default mode)

MFA(value) Accel over value master distance. Default is zero (off)

MFD(value) Decel over value master distance. Default is zero (off)

MFDIV=expression Assign Incoming counts Divisor

MFMUL=expression Assign Incoming counts Multiplier

MF0 Initiate and zero counter, but do not follow

MFR Select follow mode using quadrature encoder input.

MFR(2) Set Trajectory Generator 2 to run in Mode Follow Ratio (electronic Gearing)

MFSLEW(value) Stay at slew for value distance, then decel

MINV(0) Default motor commutation direction

MINV(1) Invert commutation, shaft rotates opposite direction

MP Initiate Position Mode

MP(1) Set Trajectory Generator 1 to run in Position Mode

MS0 Initiate and zero counter, but do not follow

MSR Calculate Mode Step Ratio and prepare to follow

MT Initiate Torque Mode (Open Loop)

MTB Enable mode torque brake

MV Initiate Velocity Mode

MV(1) Set Trajectory Generator 1 to run in Velocity Mode

O=expression Set origin, set present position to some value

O(gen#)=expression Set origin for move gen# to some value

OFF Turn the amplifier off

OSH=expression Origin shift of position counter on the fly

OSH(gen#)=expression Shift origin for move gen# by some value

PML=expression Sets the position modulo limit wrap value

PMT=expression Set the position modulo target

PRT=expression Set the relative target position

PRTS=(dist1;axis1,dist2;axis2,dist3;axis3) Set synchronized relative target position *COMBITRONIC*

PRTSS=(dis1;axis) Set supplemental synchronized relative target position *COMBITRONIC*

PT=expression Set the absolute target position

PTS=(dist1;axis1,dist2;axis2,dist3;axis3) Set synchronized absolute target position *COMBITRONIC*

PTSS=(dis1;axis) Set supplemental synchronized absolute target position *COMBITRONIC*

S Instantly stop motor

S(gen#) Instantly stop trajectory generator (gen#)

T=expression Set the commanded torque while in MT mode

TH=expression Set maximum allowable thermal limit (degrees C)

VT=expression Set the velocity target for a move

VTS=expression Set synchronized velocity target for a move *COMBITRONIC*

X Decelerate to a stop at present deceleration rate

X(gen#) Decelerate to a stop , trajectory generate (gen#)

Status Commands:

Ba Over current bit, status word 0, bit 4 status word 1, bit 3

Be Excessive position error, status word 0, bit 6

Bh Excessive temperature occurred, status word 0, bit 5

Bl Left (-) over travel limit, status word 0, bit 13

Bm Left (-) over travel limit active, status word 0, bit 15

Bo Motor is off, status word 0, bit 1

Bp Right (+) over travel limit active, status word 0, bit 14

Br Right (+) over travel limit, status word 0, bit 12

Bt Trajectory in progress, status word 0, bit 2

Bv Velocity limit, status word 0, bit 7

CLK=expression System Clock value in milliseconds

RAC Report commanded acceleration

RAT Report target acceleration

Ra Report value of variable 'a'

Rab[0] Report value of ab[0]

Raf[0] Report floating point value of af[0]

Ral[0] Report value of al[0]

Raw[0] Report value of aw[0]

RCKS Report Checksum

RB(sw,b) Report status bit, b, from status word, sw

RCLK Report system clock in milliseconds

RCTR(0) Report present value of internal encoder

RCTR(1) Report present value of external encoder

RDEA Report actual derivative error

RDEL Report commanded derivative error limit

RDT Report target deceleration

REA Report actual following error

REL Report commanded following error limit

RI (0) Report where the rising edge of the internal index was detected

RI (1) Report where the rising edge of the external index was detected

RIN(#) Report the state of a I/O

RIN(W,0) Report the first word of local I/O

RINA(V,#) Reports voltage level (scaled from supply) of analog input value for a given I/O defined by #

RINA(V1,#) Reports voltage level (scaled 0-5 VDC) of analog input value for a given I/O defined by #

RJ(0) Report where the falling edge of the internal index was detected

RJ(1) Report where the falling edge of the external index was detected

RMFDIV Report Divisor

RMFMUL Report Multiplier

RPA Report present actual position

RPC Report present commanded position

RPC(gen#) Report commanded position for trajectory generator (gen#)

RPMA Report the current modulo counter

RPML Report position modulo limit

RPMT Report the most recent setting of PMT (position modulo target)

RPRA Report actual relative position

RPRC Report commanded relative position

RPRT Report present relative target position

RPT Report present target position

RRES Report encoder resolution of motor

RSP Report sampling rate and firmware version

RTMR(x) Report timer x (present time left in milliseconds)

RT Report commanded torque

RVC Report commanded velocity

RVT Report target velocity

RUIA Reports current (Amps=UIA/1000)

RUJA Reports bus voltage (Volts=UJA/1000)

RW(value) Report status word

Z(sw,b) Clears/zeros status word bits

Za Reset over current bit

Ze Reset position error bit

Zh Reset over temperature bit

Zl Reset left(-) historical limit bit

Zr Reset right(+) historical limit bit

ZS Clear all errors, reset system latches to power up state

Zw Reset wraparound bit

Variable Commands:

a=expression Variable, 32 bit signed integers, a-z, aa-zz, aaa-zzz, 78 total variables

ab[x]=expression Array variables, 8 bit byte arrays, x can be 0-203

af[x]=expression Floating point array variables, x can be 0-7

al[x]=expression Array variables, 32 bit long arrays, x can be 0-50

aw[x]=expression Array variables, 16 bit word arrays, x can be 0-101

EPTR=expression EEPROM pointer, non-volatile memory, use before VLD and VST commands

VLD(variable,quantity) Load values from EEPROM to variables starting at EPTR location

VST(variable,quantity) Store values to EEPROM from variables starting at EPTR location

Note: See users guide for complete list of commands and full syntax. Many commands such as Cam mode and dual trajectory mode commands are not fully explained here.

COMBITRONIC **These commands require Combitronic with -C or -DN product configuration option to execute.**