

ViewX Macro 功能

目前提供巨集的下列幾種

INITIAL 巨集

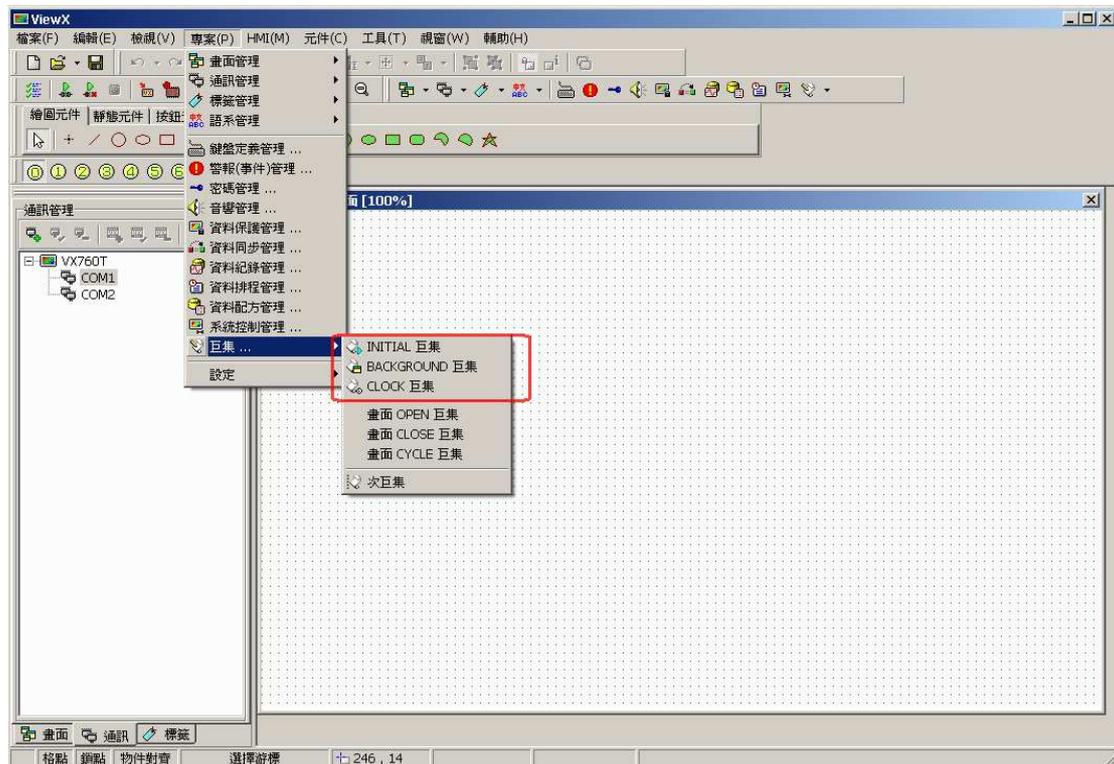
HMI 送電後或重置後,一開始啟動的巨集程式,此巨集只會執行一次,一般使用於初始值的設定,預設一次可以執行 4096 執行步階。

BACKGROUND 巨集

HMI 送電後或重置後,程式會一直執行,類似背景程式一樣,預設 50ms 可以執行 1024 執行步階。

CLOCK 巨集

HMI 送電後或重置後,程式會間隔一段時間執行,間隔時間預設 500ms 可以執行 4096 執行步階。



畫面 OPEN 巨集

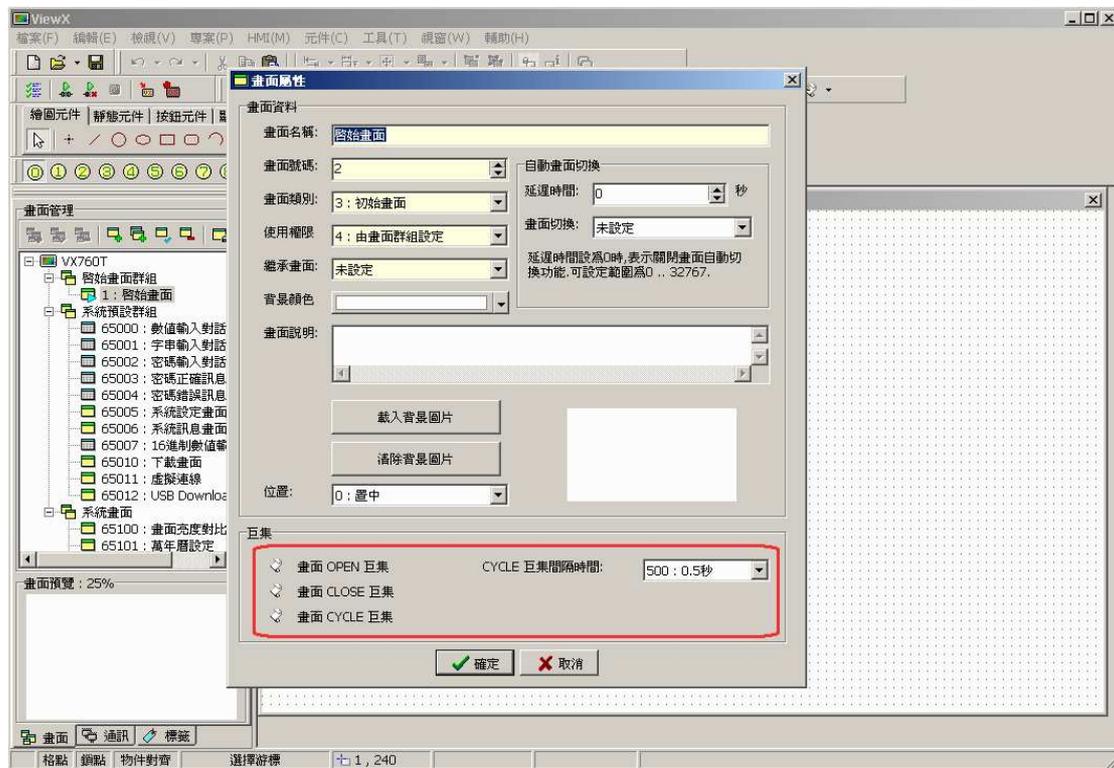
當畫面切換到目前設定的畫面時，會執行 OPEN 巨集程式，預設一次可以執行 4096 執行步階。

畫面 CLOSE 巨集

當目前畫面切換至其他畫面時，會執行 CLOSE 巨集程式，預設一次可以執行 4096 執行步階。

畫面 CYCLE 巨集

當畫面切換到目前設定的畫面時，會一直執行 CYCLE 巨集程式，間隔時間預設為 500ms，預設一次可以執行 4096 執行步階。



元件動作巨集 (按鈕元件及編輯元件)

當按鈕元件或編輯元件動作時, 會執行相對應的巨集程式, 預設一次可以執行 4096 執行步階。

設ON按鈕

按鈕 資料 狀態

資料

動作: 0 : 設ON

讀取位置: NO.0

寫入位置: NO.0

寫入通知:

On巨集: SETON

Off巨集: SETOFF

選項

使用權限: 4 : 由畫面設定

使用確認: 65535 : 按下時動作

鍵盤定義: 未設定

鍵盤視窗: 未設定

指定聲音: 未設定

致能條件:

隱藏物件

權限不足或除能時隱藏

權限不足時顯式提示符號

動作後, 設定為最低操作權限

確定 取消

數值編輯

邊框 資料 文字

資料

資料格式: 3 : [16Bit] 有號數

讀取位置:

最大值:

最小值:

寫入位置:

寫入通知:

動作前巨集: ActiveBefor

動作後巨集: ActiveAfter

選項

使用權限: 4 : 由畫面設定

使用確認: 65535 : 按下時動作

鍵盤定義: 未設定

鍵盤視窗: 65000 : 數值輸入對話盒

致能條件:

權限不足或除能時隱藏

權限不足時顯式提示符號

使用連續輸入

動作後, 設定為最低操作權限

工程轉換

偏移值: 0

倍率值: 1.0

輔助計算

顯示格式

整數位數: 4

小數位數: 0

十進位顯示

十六進位顯示

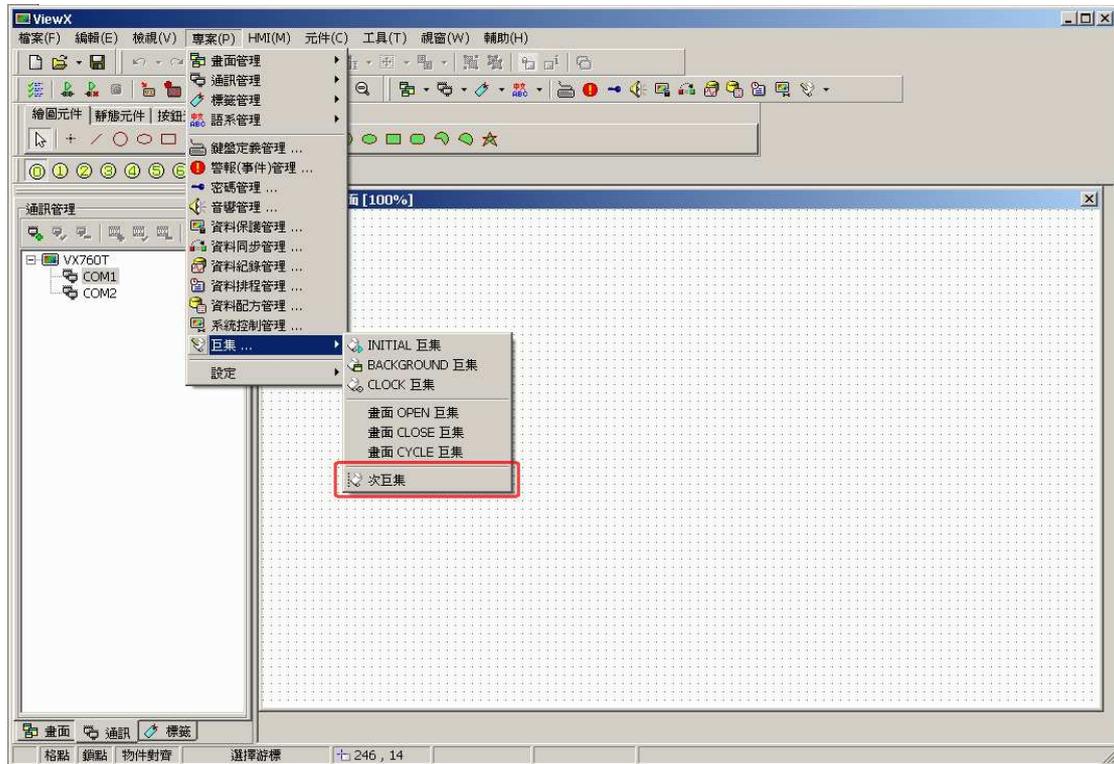
使用32位元浮點數顯示

高位數補零

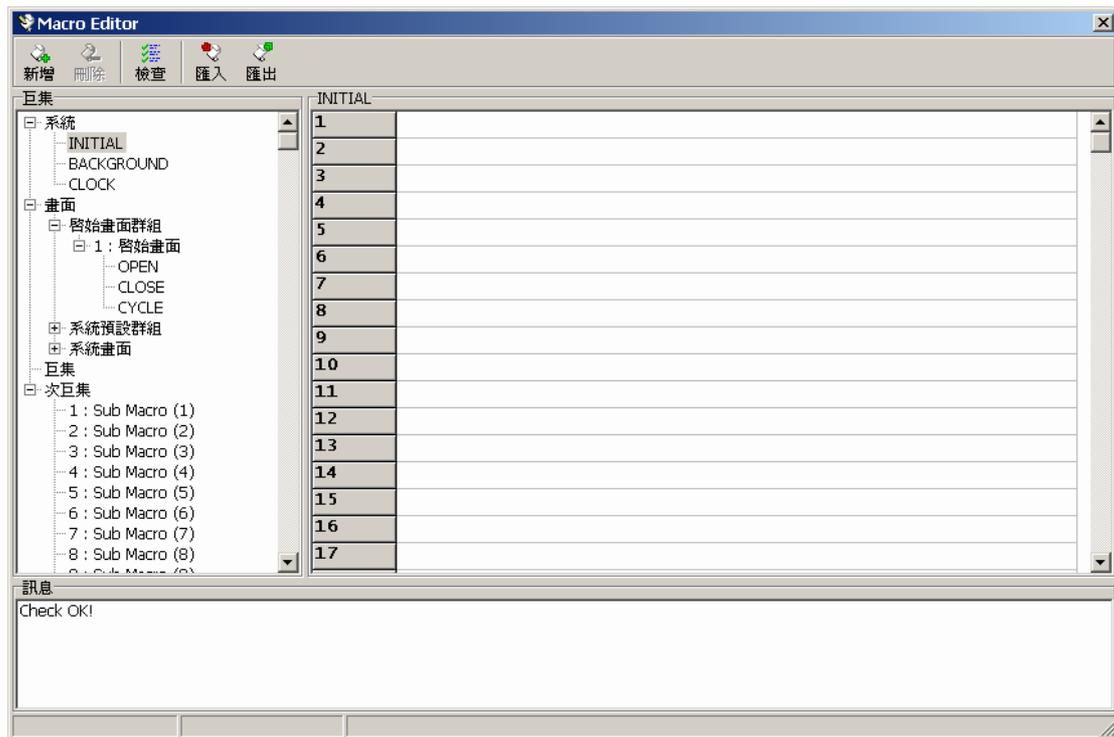
確定 取消

次巨集

為巨集程式中的副程式, 提供編號由 1 到 512 個副程式, 可以將常用及通用的功能寫在副程式中, 方便主程式呼叫。



巨集編輯畫面



編輯畫面上方提供工具列 新增, 刪除, 檢查, 匯入, 匯出

新增: 新增一個元件動作巨集.

刪除: 刪除一個元件動作巨集.

檢查: 檢查目前巨集程式, 是否有錯誤.

匯入: 從檔案匯入至目前編輯的巨集程式, 目前的編輯程式會被清除.

匯出: 將目前編輯的巨集程式, 儲存至檔案.

編輯畫面左邊提供巨集程式樹狀圖, 把所有的巨集顯示出來, 方便編輯及預覽巨集程式

編輯畫面右邊為編輯程式畫面主體.

編輯畫面下方為檢查程式訊息視窗, 當有執行檢查程序時, 才會顯示出來. 當檢查發現問題時, 會顯示在視窗內, 當滑鼠快按 2 下時, 編輯程式畫面會自動跳到對應程式位置.

巨集指令語法介紹

資料格式設定一欄表

格式	位元	8 位元 有號數	8 位元 無號數	16 位元 有號數	16 位元 無號數	32 位元 有號數	32 位元 無號數	32 位元 符點數
定義	BIT	INT8S	INT8U	INT16S 或 SIGNED	INT16U	INT32S 或 SIGNED DW	INT32U 或 DW	FLOAT32 或 FLOAT

運算元及設定運算元一欄表

運算元	說明
+	加法
-	減法
*	乘法
/	除法
%	整數餘數
!	邏輯 反 運算
~	位元 反 運算
==	等於
!=	不等於
<	小於
<=	小於等於
>	大於
>=	大於等於
 or "OR"	邏輯 或 運算
&& or "AND"	邏輯 及 運算
&	位元 及 運算
^	位元 互斥或 運算
	位元 或 運算
<<	左移運算
>>	右移運算

設定運算元	說明
=	設定
+=	加法設定
-=	減法設定
*=	乘法設定
/=	除法設定
%=	整數餘數設定
<<=	左移設定
>>=	右移設定
&=	位元 及設定
^=	位元 互斥或設定
=	位元 或設定

指令一欄表

指令	說明
LABEL <常數>	程式位置標籤(常數範圍為 0-65535)
GOTO <常數>	無條件跳躍到程式位置標籤(常數範圍為 0-65535)
CALL <運算式>	呼叫次巨集(次巨集編號 1-512)
RET	返回呼叫程式位置
IF <運算式> THEN GOTO LABEL <常數>	當變數條件成立(不等於 0)的時候, 跳躍到程式位置標籤, 否則繼續下一行的指令
IF <運算式> THEN DO	當變數條件成立(不等於 0)的時候, 繼續下一行的指令, 否則跳躍到下一個判斷式(EL IF)或(ELSE DO)繼續執行或執行 ENDIF 的下一行的指令.
EL IF <運算式> THEN DO	當變數條件成立(不等於 0)的時候, 繼續下一行的指令, 否則跳躍到下一個判斷式(EL IF)或(ELSE DO)繼續執行或執行 ENDIF 的下一行的指令.
ELSE DO	配合 IF 或 EL IF 指令, 標示當變數條件不成立(等於 0)的時候, 所要執行的位置
ENDIF	配合 IF 或 EL IF 指令, 標示當變數條件不成立(等於 0)的時候, 所要跳躍的位置
FOR <運算式>	程式迴圈, 以變數的數值為固定做幾次迴圈, 程式範圍為 FOR 到對應 NEXT 中間的程式.
NEXT	配合 FOR 指令, 標示程式迴圈的範圍
END	結束巨集程式
SETB <位元變數位址>	設定 位元變數
CLRB <位元變數位址>	清除 位元變數
INVB <位元變數位址>	反相 位元變數

系統變數 N0-N65535

為 HMI 內部已規劃有的變數, 數量為 65536 個 WORD(16Bits), 定義名稱為 N0-N65535.

當資料格式為 DWORD 時, 此時系統會自動讀取編號後面一個 WORD, 變成 Hi WORD 加上定義編號 Lo WORD 組成 DWORD. 例如 N0, N0 為 Lo WORD, N1 為 Hi WORD, 組成 DWORD.

當資料格式為 WORD 時, 就直接使用定義名稱, (N0-N65535).

當要使用位元定址時, 定義名稱後面加”.”符號在加位元位置 0~15. 例如 N0 的第 5 個位元可以寫成 “N0. 5”, N100 的第 11 個位元可以寫成 “N100. 11”.

指令說明

LABEL <常數>

程式位置標籤(常數範圍為 0-65535)

標籤的位置可以是在程式碼中的任何的一行.

N0=100

定義標籤編號 10

LABEL 10

N1=200

...

GOTO LABEL 10

...

GOTO LABEL 20

...

定義標籤編號 20

LABEL 20

GOTO <常數>

無條件跳躍到程式位置標籤(常數範圍為 0-65535)

N0=100

LABEL 10

N1=200

...

...

跳躍至標籤編號 10 的位置

GOTO LABEL 10

CALL <變數/常數/運算式>

呼叫次巨集(次巨集編號 1-512)

呼叫次巨集編號可以是變數或常數或運算式

```
N0=100
# CALL <變數>
CALL N0
...
# CALL <常數>
CALL 100
...
# CALL <運算式>
CALL 100+N0
```

RET

返回呼叫程式位置

IF <變數> THEN GOTO LABEL <常數>

當變數條件成立(不等於 0)的時候, 跳躍到程式位置標籤, 否則繼續下一行的指令

```
LABEL 10
...
#當變數 N0+N1 等於 200 時, 跳躍到標籤 10 的位置
IF (N0+N1)==200 THEN GOTO LABEL 10
```

IF <變數> THEN DO

當變數條件成立(不等於 0)的時候, 繼續下一行的指令, 否則跳躍到下一個判斷式(ELIF)或(ELSE DO)繼續執行或執行 ENDIF 的下一行的指令. IF, ENDIF 蜂巢結構最多 16 層.

```
#當變數 N0 不等於 100 時, 執行 N1=1000 的指令.
IF N0!=100 THEN DO
N1=1000
# IF .. THEN DO 必須與 ENDIF 配對
ENDIF
```

ELIF <變數> THEN DO

當變數條件成立(不等於 0)的時候,繼續下一行的指令,否則跳躍到下一個判斷式(ELIF)或(ELSE DO)繼續執行或執行 ENDF 的下一行的指令。

```
#當變數 N0 等於 100 時,執行 N1=1000 的指令
#當變數 N0 等於 200 時,執行 N1=2000 的指令
IF N0==100 THEN DO
N1=1000
# ELIF 必須在 IF THEN DO 語法後面
ELIF N0==200 THEN DO
N1=2000
ENDIF
```

ENDIF

配合 IF 或 ELIF 指令,標示當變數條件不成立(等於 0)的時候,所要跳躍的位置,IF, ENDF 蜂巢結構最多 16 層。

```
#當變數 N0 不等於 100 時,執行 N1=1000 的指令。
IF N0!=100 THEN DO
N1=1000
# IF THEN DO 必須與 ENDF 配對
ENDIF
```

FOR <變數>

程式迴圈,以變數的數值為固定做幾次迴圈,程式範圍為 FOR 到對應 NEXT 中間的程式。

FOR, NEXT 支援蜂巢結構最多 16 層

```
#計算 1 到 10 的總數放在 N0 的變數
N0=0
N1=1
FOR 10
N0+=N1
N1+=1
NEXT
```

NEXT

配合 FOR 指令, 標示程式迴圈的範圍, FOR, NEXT 支援蜂巢結構最多 16 層

#計算 1 到 50 的總數放在 N0 的變數

N0=0

N1=1

FOR 50

N0+=N1

N1+=1

NEXT

END

結束巨集程式

SETB <位元變數位址>

設定 位元變數

#將 N100 變數第五個位元設定為 1

SETB N100, 5

CLRB <位元變數位址>

清除 位元變數

#將 N10 變數第 12 個位元設定為 0

CLRB N10, 12

INVB <位元變數位址>

反相 位元變數

#將 N1 變數第 15 個位元反相 (0 變成 1, 1 變成 0).

INVB N1, 15

在 ViewX Macro 裡, 除了指令用法外, 有 2 個重要的語法, 運算式語法跟設定運算式。

運算式語法

提供變數間的數學運算或邏輯運算, 這些運算符號稱為運算元, 目前有提供的運算元如(表二), 總共有 20 種運算, 運算式主要搭配設定運算式語法, 及指令內的語法。

使用範例:

單純運算式

$(N0+N1)/2$

$(N0+100)$

搭配設定運算元

$N0 = (N1 << 2)$

$N0 += (N1+N2)/2$

搭配指令

IF $(N0 == 100)$ && $(N1 == 200)$ THEN GOTO LABEL 10

...

ENDIF

CALL $(N0+100)$

設定運算式語法

是巨集程式中最主要的設定變數的數值的語法, 這些設定符號稱為設定運算元, 目前有提供的設定運算元如(表二), 總共有 11 種運算。

使用範例:

$N0 = 100$

$N0 = (N1+N2)/2$

$N0 += 100$

$N0 *= 2$

$N0 <<= 2$

變數資料 及 設定運算 的資料格式設定

變數資料

目前提供巨集可以使用的變數有系統變數。系統變數預設資料格式為 16 位元無號數。如果需要其他資料格式運算時，必須加入資料格式設定，資料格式設定是在變數前加入左右括弧內填入資料格式保留字，例如 (INT32S)N0，表示使用 32 位元有號數運算，資料位址在 N0。

設定運算

要將設定值的變數設定資料格式，必須在一行指令最後面加入左右括弧內填入資料格式保留字，
例如 N0=(INT32S)N100(INT32S)，表示使用 32 位元有號數運算，資料位址在 N100，設定至 32 位元有號數，資料位址在 N0。

例如

#位元定址及設定

N0.3=(BIT)N10.4 && (BIT)N10.5(BIT)

#如果同一行變數資料格式都相同時，可以指設定 [設定運算] 資料格式

N0.3 = N10.4 && N10.5(BIT)

#32 位元有號數運算

N0=N2+N4(INT32S)

N0=N2+N4(SIGNED DW)

#32 位元浮點數運算

N0=N2+N4(FLOAT)

函數一覽表

函數名稱	函數功能	傳回值	參數 1	參數 2
_PI	傳回 π 浮點數值	FLOAT32		
_BCD	將 BIN 數值轉換為 BCD 格式的數值	INT32U	INT32U	
_BIN	將 BCD 數值轉換為 BIN 格式的數值	INT32U	INT32U	
_SQRT	計算平方根浮點數值	FLOAT32	FLOAT32	
_LOG	計算自然對數浮點數值	FLOAT32	FLOAT32	
_LOG10	計算以 10 為底的對數浮點數值	FLOAT32	FLOAT32	
_RADIAN	將度數轉換為弧度浮點數值	FLOAT32	FLOAT32	
_DEGREE	將弧度轉換為度數浮點數值	FLOAT32	FLOAT32	
_SIN	計算三角函數正弦值浮點數值	FLOAT32	FLOAT32	
_COS	計算三角函數餘弦值浮點數值	FLOAT32	FLOAT32	
_TAN	計算三角函數正切值浮點數值	FLOAT32	FLOAT32	
_ASIN	計算三角函數反正弦值浮點數值	FLOAT32	FLOAT32	
_ACOS	計算三角函數反餘弦值浮點數值	FLOAT32	FLOAT32	
_ATAN	計算三角函數反正切值浮點數值	FLOAT32	FLOAT32	
_ABS	計算絕對值浮點數值	FLOAT32	FLOAT32	

浮點數常數

直接設定浮點數常數至變數

`N0=1.2345 (FLOAT32)`

_BCD

將 BIN 數值轉換為 BCD 格式的數值

函數語法: `INT32U _BCD(INT32U)`

傳入資料格式: `INT32U` 32 位元無號數值

回傳資料格式: `INT32U` 32 位元無號數值

`N2=_BCD(N0) (INT32U)`

_BIN

將 BCD 數值轉換為 BIN 格式的數值

函數語法: INT32U _BIN(INT32U)

傳入資料格式: INT32U 32 位元無號數值

回傳資料格式: INT32U 32 位元無號數值

`N4= _BIN(N2) (INT32U)`

_SQRT

計算平方根浮點數值

函數語法: FLOAT32 _SQRT(FLOAT32)

傳入資料格式: FLOAT32 浮點數值

回傳資料格式: FLOAT32 平方根浮點數值

`N0= _SQRT(1, 5129) (FLOAT32) #1, 5129=1, 23*1, 23`

_LOG

計算自然對數浮點數值

函數語法: FLOAT32 _LOG(FLOAT32)

傳入資料格式: FLOAT32 浮點數值

回傳資料格式: FLOAT32 自然對數浮點數值

`N0= _LOG(N2) (FLOAT32)`

_LOG10

計算以 10 為底的對數浮點數值

函數語法: FLOAT32 _LOG10(FLOAT32)

傳入資料格式: FLOAT32 浮點數值

回傳資料格式: FLOAT32 以 10 為底的對數浮點數值

`N0= _LOG10(N2) (FLOAT32)`

_RADIAN

將度數轉換為弧度浮點數值

函數語法: **FLOAT32** **_RADIAN**(**FLOAT32**)

傳入資料格式: **FLOAT32** 度數浮點數值

回傳資料格式: **FLOAT32** 弧度浮點數值

$N0 = _RADIAN(N2)$ (**FLOAT32**)

_DEGREE

將弧度轉換為度數浮點數值

函數語法: **FLOAT32** **_DEGREE**(**FLOAT32**)

傳入資料格式: **FLOAT32** 弧度浮點數值

回傳資料格式: **FLOAT32** 度數浮點數值

$N0 = _DEGREE(N2)$ (**FLOAT32**)

_SIN

計算三角函數正弦值浮點數值

函數語法: **FLOAT32** **_SIN**(**FLOAT32**)

傳入資料格式: **FLOAT32** 弧度浮點數值

回傳資料格式: **FLOAT32** 正弦值浮點數值

$N0 = _SIN(N2)$ (**FLOAT32**)

_COS

計算三角函數餘弦值浮點數值

函數語法: **FLOAT32** **_COS**(**FLOAT32**)

傳入資料格式: **FLOAT32** 弧度浮點數值

回傳資料格式: **FLOAT32** 餘弦值浮點數值

$N0 = _COS(N2)$ (**FLOAT32**)

_ATAN

計算三角函數反正切值浮點數值

函數語法: FLOAT32 _ATAN(FLOAT32)

傳入資料格式: FLOAT32 弧度浮點數值

回傳資料格式: FLOAT32 反正切值浮點數值

$N0 = _ATAN(N2)$ (FLOAT32)

_ABS

計算絕對值浮點數值

函數語法: FLOAT32 _ABS(FLOAT32)

傳入資料格式: FLOAT32 輸入計算浮點數值

回傳資料格式: FLOAT32 絕對值浮點數值

$N0 = _ABS(-1, 23)$ (FLOAT32)

_PI

傳回 π 浮點數值

函數語法: FLOAT32 _PI ()

傳入資料格式: 無

回傳資料格式: FLOAT32 傳回 π 浮點數值

$N0 = _PI ()$ (FLOAT32)

巨集選項暫存器

N65152.0 (位元 0)

0=控制器變數設定時, 透過通訊修改數值

1=控制器變數設定時, 直接修改 HMI 記憶體資料並透過通訊修改數值.