ViewX HMI

進階 Macro 功能函數

2014-08-18

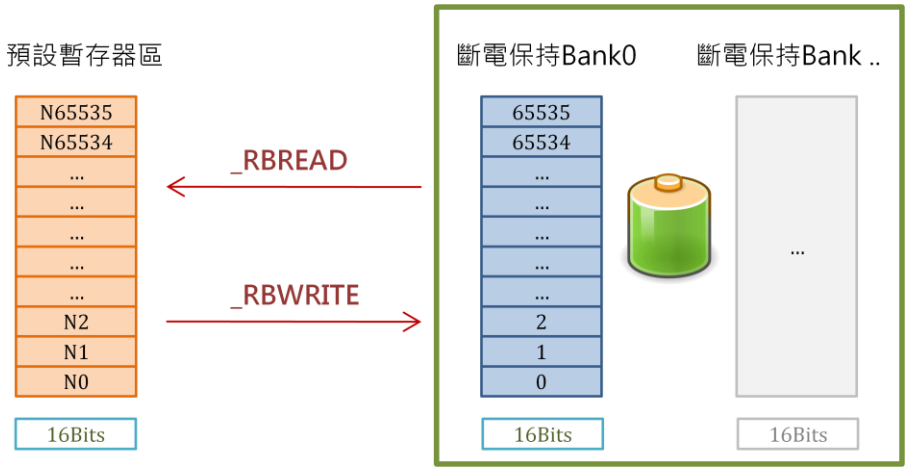
目錄

1. 暫存器/記憶體相關函式

函數名稱	函數功能	版本需求
_MEMCPY	記憶體複製	1.33
_MEMCMP	記憶體比較	1.33
_MEMSET	記憶體設定	1.33
_REGCPY	變數區域複製	1.33
_REGCMP	變數區域比較	1.33
_REGSET	變數區域設定	1.33
_MWORD2BYTE	多筆字組資料轉換至多筆位元組	1.33
_MBYTE2WORD	多筆位元組資料轉換至多筆字組	1.33
_RBREAD	讀取斷電保持 Bank 記憶體區塊的資料	1.33
_RBWRITE	寫入斷電保持 Bank 記憶體區塊的資料	1.33

斷電保持 Bank 記憶體區塊說明

斷電保持 Bank 記憶體區塊是由多個斷電保持 Bank 記憶體組成,編號由 0 開始遞增. (目前只開放 0, Bank0),一個斷電保持 Bank 記憶體容量為 65536 個字組(16bits)資料可供存放,編號由 0 遞增至 65535.透過_RBREAD 及_RBWRITE 巨函式來提供讀取及寫入功能.



相關函數

_RBREAD, _RBWRITE

_MEMCPY

記憶體複製

INT32S _MEMCPY(INT32S destRegNum, INT32S srcRegNum, INT32S len)

將資料從變數記憶體區複製到另外一個變數記憶體區

destRegNum 可以使用範圍 0~64511
srcRegNum 可以使用範圍 0~64511
len 可以使用範圍 1~256
回傳值表示複製的位元組數 0:表示記憶體範圍錯誤, 1~256 複製的位元組數

範例

N100 = _MEMCPY(200, 250, 10)

將 N250 變數區域起 10 個 BYTE(5 個變數,N250-N254)複製到 N200-N204,複製位元組(BYTE)數量 10 放置 N100.

範例

N100=1000
N101=1500
N102=20
N500 = _MEMCPY(N100, N101, N102)

將 N1000 變數區域起 20 個 BYTE(10 個變數,N1000-N1009)複製到 N1500-N1509,複製位元組(BYTE)數量 20 放置 N500.

相關函數

_MEMCMP, _MEMSET,_REGCPY,_REGCMP, _REGSET

_MEMCMP

記憶體比較

INT32S _MEMCMP(INT32S destRegNum, INT32S srcRegNum, INT32S len)

比較 2 個變數記憶體區內一定數目的位元組(BYTE)

destRegNum 可以使用範圍 0~65535

srcRegNum 可以使用範圍 0~65535

len 可以使用範圍 1~256

回傳值表示比較結果 -1:表示記憶體範圍錯誤, 0: 變數內容相同, 1: 變數內容不同

範例

N100 = _MEMCMP(1000, 2000, 10)

比較 N1000-N1004 與 N2000-N2004 變數數值是否相同.

相關函數

_MEMCPY, _MEMSET,_REGCPY,_REGCMP, _REGSET

_MEMSET

記憶體設定

INT32S _MEMSET(INT32U regNum, INT32S value, INT32S len)

將某個變數記憶體區所有位元組設定成某個值

regNum 可以使用範圍 0~64511

len 可以使用範圍 1~256

回傳值表示設定的位元組數 0:表示記憶體範圍錯誤, 1~256 設定的位元組數

範例

N100 = _MEMSET(1000, 55H, 10)

設定 N1000-N1004 數值為 5555H.(設定預設為位元組(8Bits)為單位, N 變數為字組(16Bits))

相關函數

_MEMCPY, _MEMCMP,_REGCPY,_REGCMP, _REGSET

_REGCPY

變數區域複製

INT32S _REGCPY(INT32S destRegNum, INT32S srcRegNum, INT32S len)

將資料從變數記憶體區複製到另外一個變數記憶體區

destRegNum 可以使用範圍 0~64511

srcRegNum 可以使用範圍 0~64511

len 可以使用範圍 1~256

回傳值表示複製的變數數量 0:表示記憶體範圍錯誤, 1~256 複製的變數數量

範例

```
N100 = _REGCPY( 200, 250, 10)
```

將 N250-N259 變數複製到 N200-N209 變數.

範例

```
N100=1000
N101=1500
N102=20
N500 = _REGCPY( N100, N101, N102)
```

將 N1500-N1519 變數複製到 N1000-N1019 變數.

相關函數

_MEMCPY, _MEMCMP,_MEMSET,_REGCMP, _REGSET

_REGCMP

變數區域比較

INT32S _REGCMP(INT32S destRegNum, INT32S srcRegNum, INT32S len)

比較 2 個變數記憶體區內一定數目的變數(16Bits)

destRegNum 可以使用範圍 0~65535

srcRegNum 可以使用範圍 0~65535

len 可以使用範圍 1~256

回傳值表示比較結果-1:表示記憶體範圍錯誤, 0: 變數內容相同, 1: 變數內容不同

範例

```
N100 = _REGCMP( 1000, 2000, 10)
```

比較 N1000-N1009 與 N2000-N2009 變數數值是否相同.

相關函數

_MEMCPY, _MEMCMP,_MEMSET,_REGCPY, _REGSET

_REGSET

變數區域設定

INT32S _REGSET(INT32U regNum, INT32S value, INT32S len)

將某個變數記憶體區所有變數設定成某個值

regNum 可以使用範圍 0~64511

len 可以使用範圍 1~256

回傳值表示設定的變數數量 0:表示記憶體範圍錯誤, 1~256 設定的變數數量

範例

N100 = _REGSET(1000, 1234, 10)

設定 N1000-N1009 數值為 1234.

相關函數

_MEMCPY, _MEMCMP,_MEMSET,_REGCPY, _REGCMP

_MWORD2BYTE

多筆字組資料轉換至多筆位元組

INT32U _MWORD2BYTE(INT32S destByteRegNum, INT32S srcWordRegNum, INT32S len)

回傳值表示轉換的筆數

destByteRegNum 可以使用範圍 0~64511
srcWordRegNum 可以使用範圍 0~64511
len 可以使用範圍 1~256
回傳值表示轉換的筆數 0:表示變數範圍錯誤, 1~256 轉換的筆數

範例

```
_STRW( N200, "ABCD")
_MWORD2BYTE( 100, 200, 4)
```

執行結果

```
N100 = 4241H
N101 = 4443H
N200 = 0041H
N201 = 0042H
N202 = 0043H
N203 = 0044H
```

相關函數

_MBYTE2WORD

_MBYTE2WORD

多筆位元組資料轉換至多筆字組

INT32U _MBYTE2WORD(INT32S destWordRegNum, INT32S srcByteRegNum, INT32S len)

回傳值表示轉換的筆數

destWordRegNum 可以使用範圍 0~64511
srcByteRegNum 可以使用範圍 0~64511
len 可以使用範圍 1~256
回傳值表示轉換的筆數 0:表示變數範圍錯誤, 1~256 轉換的筆數

範例

```
_STR( N200, "ABCD")
_MBYTE2WORD( 100, 200, 4)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N100 = 0043H
N101 = 0044H
N200 = 4241H
N201 = 4443H
```

相關函數

_MWORD2BYTE

_RBREAD

讀取斷電保持 Bank 記憶體區塊的資料

INT32S _RBREAD(INT32S bank, INT32S destRegNum, INT32S scrAddr, INT32S cnt)

回傳值表示讀取的暫存器數量

bank 可以使用範圍 0~6(目前只開放 0:Bank0)

destRegNum 讀取資料的目的暫存器位置的編號

scrAddr 讀取 Bank 的來源位置的編號

cnt 可以使用範圍 1~8192

回傳值 0:表示變數範圍錯誤或運行機型沒有提供斷電保持 Bank 記憶體,其他值為讀取的暫存器數量

注意

此巨集函式只支援有提供 斷電保持 Bank 記憶體的機型運作.

範例

```
N100 = _RBREAD( 0,1000,3000,100)
```

執行結果

```
N100 = 100
N1000~N1099 = RBank0.3000~ RBank0.3099
```

相關函數

_RBWRITE

_RBWRITE

寫入斷電保持 Bank 記憶體區塊的資料

INT32S _ RBWRITE(INT32S bank, INT32S scrRegNum, INT32S destAddr, INT32S cnt)

回傳值表示寫入的暫存器數量

bank 可以使用範圍 0~6(目前只開放 0, Bank0)
scrRegNum 資料來源暫存器位置的編號
destAddr 寫入 Bank 內目的位置的編號
cnt 可以使用範圍 1~8192
回傳值 0:表示變數範圍錯誤或運行機型沒有提供斷電保持 Bank 記憶體,其他值為寫入的暫存器數量

注意:

此巨集函式只支援有提供 斷電保持 Bank 記憶體的機型運作.

範例:

```
N100 = _RBWRITE( 0,1000,3000,100)
```

執行結果

```
N100 = 100
RBank0.3000~ RBank0.3099 = N1000~N1099
```

相關函數

_RBREAD

2. 字串相關函式

函數名稱	函數功能	版本需求
_STR	設定 ASCII(位元組)字串至變數記憶體	1.33
_STRW	設定 Unicode(字組)字串至變數記憶體	1.33
_STRLEN	傳回 ASCII(位元組)字串長度	1.33
_STRWLEN	傳回 Unicode(字組)字串長度	1.33
_STR2DEC	10 進制字串轉換成數值	1.33
_STR2HEX	16 進制字串轉換成數值	1.33
_STR2BIN	2 進制字串轉換成數值	1.33
_DECSTR	數值轉換成 10 進制字串	1.33
_HEXSTR	數值轉換成 16 進制字串	1.33
_BINSTR	數值轉換成 2 進制字串	1.33

_STR

設定 ASCII 字串至變數記憶體

INT32S _STR(REG reg, STRING str)

將 ASCII(8Bits)字串寫入變數記憶體區

reg 可以使用範圍 N0~N64511

str 為 ASCII 字串,字串長度為 1~256

回傳值表示位元組的長度 0:表示記憶體範圍錯誤, 1~256 設定的位元組的長度

字串格式裡除了可以使用一般文字外還可以利用"\"跳脫符號加入非字串文字

跳脫符號	字元值
""	22H=34
\'	27H=39
\"	22H=34
\?	3FH=63
\\	5CH=92
\r	0DH=13
\n	0AH=10
\000 ~ \777	8 進制碼字元
\x00 ~ \xFF	16 進制碼字元

範例

```
N100 = _STR( N1000, "ABCD")
```

執行結果

```
N100 = 5
N1000 = 4241H      //'BA'
N1001 = 4443H      //'DC'
N1002 = 0000H      //字串結尾
```

相關函數

_STRW

_STRW

設定 Unicode 字串至變數記憶體

INT32S _STRW(INT32U regNum, STRING str)

將 Unicode(16Bits)字串寫入變數記憶體區

regNum 可以使用範圍 0~64511

str 為 ASCII 字串,字串長度為 1~256

回傳值表示位元組的長度 0:表示記憶體範圍錯誤, 1~256 設定的位元組的長度

注意

字串使用 Unicode 編碼,在 HMI 顯示字串時,內部編碼只有低 7 位元(0-127 英文字體)與 Unicode 相同,所以字串中出現 Unicode 編碼超出 127 時,HMI 顯示會不正確或無法顯示.

範例

```
N100 = _STRW( N1000, "ABCD")
```

執行結果

```
N100 = 10
N1000 = 0065H      //'A'
N1001 = 0066H      //'B'
N1002 = 0067H      //'C'
N1003 = 0068H      //'D'
N1004 = 0000H      //字串結尾
```

相關函數

_STR, _STRLEN, _STRWLEN

_STRLEN

傳回 ASCII(位元組)字串長度

INT32U _STRLEN(INT32S regNum)

回傳值表示 ASCII 字串的長度

regNum 可以使用範圍 0~64383

回傳值表示字串長度 0:表示變數範圍錯誤或長度超過 256 或字串長度為 0,1~256 字串長度

範例

```
_STR( N200, "ABCD")
N100 = _STRLEN( 200)
```

執行結果

```
N100 = 4
N200 = 4241H
N201 = 4443H
```

相關函數

_STR, _STRW, _STRWLEN

_STRWLEN

傳回 Unicode(字組)字串長度

INT32U _STRWLEN(INT32S regNum)

回傳值表示 Unicode 字串的長度

regNum 可以使用範圍 0~64255

回傳值表示字串長度 0:表示變數範圍錯誤或長度超過 256 或字串長度為 0,1~256 字串長度

範例

```
_STRW( N200, "ABCD")
N100 = _STRWLEN( 200)
```

執行結果

```
N100 = 4
N200 = 0041H
N201 = 0042H
N202 = 0043H
N203 = 0044H
```

相關函數

_STR, _STRW, _STRLEN

_STR2DEC

10 進制字串轉換成數值

INT32S _STR2DEC(INT32S regNum, INT32S cnt)

回傳值表示轉換後的數值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256(當轉換超出數值範圍時,會發生溢位錯誤數值會不正確)

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為轉換的數值

範例

```
_STRW( N100, "12345")
N200 = _STRWLEN( 100)
N500 = _STR2DEC( 100, N200)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0045H
N200 = 5
N500 = 12345
```

範例

```
_STRW( N100, "1234567")
N200 = _STRWLEN( 100)
N500 = _STR2DEC( 100, N200)(DW)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0045H
N105 = 0046H
N106 = 0047H
N200 = 7
N500 = -10617
N501 = 18
雙字組 N500 = 1234567
```

相關函數

_STR2HEX, _STR2BIN

_STR2HEX

16 進制字串轉換成數值

INT32S _STR2HEX(INT32S regNum, INT32S cnt)

回傳值表示轉換後的數值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256(當轉換超出數值範圍時,會發生溢位錯誤數值會不正確)

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為轉換的數值

範例

```
_STRW( N100, "55AA")
N200 = _STRWLEN( 100)
N500 = _STR2HEX( 100, N200)
```

執行結果

```
N100 = 0035H
N101 = 0035H
N102 = 0041H
N103 = 0041H
N200 = 4
N500 = 55AAH
```

範例

```
_STRW( N100, "55AA")
N200 = _STRWLEN( 100)
N500 = _STR2HEX( 100, 2)
```

執行結果

```
N100 = 0035H
N101 = 0035H
N102 = 0041H
N103 = 0041H
N200 = 4
N500 = 0055H
```

相關函數 _STR2DEC, _STR2BIN

_STR2BIN

2 進制字串轉換成數值

INT32S _STR2BIN(INT32S regNum, INT32S cnt)

回傳值表示轉換後的數值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256(當轉換超出數值範圍時,會發生溢位錯誤數值會不正確)

2 進位字串可以使用 'H' 或 '1' 表示 1 的值

2 進位字串可以使用 'L' 或 '0' 表示 0 的值

2 進位字串可以使用 ':' 及 '_' 當分隔符號

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為轉換的數值

範例

```
_STRW( N100, "10100000_00000001")
N200 = _STRWLEN( 100)
N500 = _STR2BIN( 100, N200)
```

執行結果

```
N100~N116 = "10100000_00000001"
N200 =17
N500 = A051H
```

範例

```
_STRW( N100, "HLHL:LHLH:HHHH:LLLL")
N200 = _STRWLEN( 100)
N500 = _STR2BIN( 100, N200)
```

執行結果

```
N100~N118 = "HLHL:LHLH:HHHH:LLLL"
N200 =19
N500 = A5F0H
```

相關函數

_STR2DEC, _STR2HEX

_DECSTR

數值轉換成 10 進制字串

INT32S _DECSTR(INT32S destRegNum, INT32S data, INT32S format)

回傳值表示轉換後的字串長度

destRegNum 可以使用範圍 0~64255

data 輸入轉換的數值

format 轉換數值的格式

0 : 8 位元有號數

1 : 8 位元無號數

2 : 16 位元有號數

3 : 16 位元無號數

4 : 32 位元有號數

5 : 32 位元無號數

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為轉換後的字串長度

範例

```
N200 = _DECSTR( 100, 12345, 2)
```

執行結果

```
N100~N104 = "12345"
N200 =5
```

範例

```
N200 = _DECSTR( 100, -12345, 2)
```

執行結果

```
N100~N105 = "-12345"
N200 =6
```

相關函數

_HEXSTR, _BINSTR

_HEXSTR

數值轉換成 16 進制字串

INT32S _HEXSTR(INT32S destRegNum, INT32S data, INT32S format)

回傳值表示轉換後的字串長度

destRegNum 可以使用範圍 0~64255

data 輸入轉換的數值

format 轉換數值的格式

0 : 8 位元無號數,轉換字串小寫 0-9,a-f

1 : 8 位元無號數,轉換字串大寫 0-9,A-F

2 : 16 位元無號數,轉換字串小寫 0-9,a-f

3 : 16 位元無號數,轉換字串大寫 0-9,A-F

4 : 32 位元無號數,轉換字串小寫 0-9,a-f

5 : 32 位元無號數,轉換字串大寫 0-9,A-F

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為轉換後的字串長度

範例

```
N200 = _HEXSTR( 100, 0AA55H, 2)
```

執行結果

```
N100~N103 = "aa55"
N200 =4
```

範例

```
N200 = _HEXSTR( 100, 55AAFF00H, 5)
```

執行結果

```
N100~N107 = "55AAFF00"
N200 =8
```

相關函數

_DECSTR, _BINSTR

_BINSTR

數值轉換成 2 進制字串

INT32S _BINSTR(INT32S destRegNum, INT32S data, INT32S format)

回傳值表示轉換後的字串長度

destRegNum 可以使用範圍 0~64255

data 輸入轉換的數值

format 轉換數值的格式

0 : 8 位元無號數,轉換字串 0,1

1 : 8 位元無號數,轉換字串 H,L

2 : 16 位元無號數,轉換字串 0,1

3 : 16 位元無號數,轉換字串 H,L

4 : 32 位元無號數,轉換字串 0,1

5 : 32 位元無號數,轉換字串 H,L

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為轉換後的字串長度

範例

```
N200 = _BINSTR( 100, 0AA55H, 2)
```

執行結果

```
N100~N115= "1010101001010101"
N200 =16
```

範例

```
N200 = _BINSTR( 100, 55AAFF00H, 5)
```

執行結果

```
N100~N131 = "LHLHLHLHHLHLHLHLHHHHHHHHHLLLLLLLL"
N200 =32
```

相關函數

_DECSTR, _HEXSTR

3. 數學計算相關函式

函數名稱	函數功能	版本需求
_MIN	取多筆 16 位元有號數資料的最小值	1.33
_MAX	取多筆 16 位元有號數資料的最大值	1.33
_AVE	取多筆 16 位元有號數資料的平均值	1.33
_MIND	取多筆 32 位元有號數資料的最小值	1.33
_MAXD	取多筆 32 位元有號數資料的最大值	1.33
_AVED	取多筆 32 位元有號數資料的平均值	1.33
_MINF	取多筆 32 位元浮點數資料的最小值	1.33
_MAXF	取多筆 32 位元浮點數資料的最大值	1.33
_AVEF	取多筆 32 位元浮點數資料的平均值	1.33

_MIN

取多筆 16 位元有號數資料的最小值

INT32S _MIN(INT32S regNum, INT32S cnt)

回傳值表示最小值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或最小值為 0,其他值為最小值

範例

```
N100 = 156
N101 = 6578
N102 = -578
N103 = 4568
N200 = _MIN( 100, 4)
```

執行結果

N200 =-578

相關函數

_MAX, _AVE

_MAX

取多筆 16 位元有號數資料的最大值

INT32S _MAX(INT32S regNum, INT32S cnt)

回傳值表示最大值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或最大值為 0,其他值為最大值

範例

```
N100 = 156
N101 = 6578
N102 = -578
N103 = 4568
N200 = _MAX( 100, 4)
```

執行結果

N200 =6578

相關函數

_MIN, _AVE

_AVE

取多筆 16 位元有號數資料的平均值

INT32S _AVE(INT32S regNum, INT32S cnt)

回傳值表示平均值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或平均值為 0,其他值為平均值

範例

```
N100 = 156
N101 = 6578
N102 = -578
N103 = 4568
N200 = _AVE( 100, 4)
```

執行結果

N200 =2681

相關函數

_MIN, _MAX

_MIND

取多筆 32 位元有號數資料的最小值

INT32S _MIND(INT32S regNum, INT32S cnt)

回傳值表示最小值

regNum 可以使用範圍 0~65534

cnt 可以使用範圍 1~128

回傳值 0:表示變數範圍錯誤或最小值為 0,其他值為最小值

範例

```
N100 = 1564(DW)
N102 = 6573458(DW)
N104 = -574358(DW)
N106 = 4568(DW)
N200 = _MIND( 100, 4)(DW)
```

執行結果

雙字組 N200 = -574358

相關函數

_MAXD, _AVED

_MAXD

取多筆 32 位元有號數資料的最大值

INT32S _MAXD(INT32S regNum, INT32S cnt)

回傳值表示最大值

regNum 可以使用範圍 0~65534

cnt 可以使用範圍 1~128

回傳值 0:表示變數範圍錯誤或最大值為 0,其他值為最大值

範例

N100 = 1564(DW)
N102 = 6573458(DW)
N104 = -574358(DW)
N106 = 4568(DW)
N200 = _MAXD(100, 4)(DW)

執行結果

雙字組 N200 = 6573458

相關函數

_MIND, _AVED

_AVED

取多筆 32 位元有號數資料的平均值

INT32S _AVED(INT32S regNum, INT32S cnt)

回傳值表示平均值

regNum 可以使用範圍 0~65534

cnt 可以使用範圍 1~128

計算平均值總值時使用 32 位元有號數計算,請注意總值是否發生溢位

回傳值 0:表示變數範圍錯誤或平均值為 0,其他值為平均值

範例

```
N100 = 1564(DW)
N102 = 6573458(DW)
N104 = -574358(DW)
N106 = 4568(DW)
N200 = _AVED( 100, 4)(DW)
```

執行結果

雙字組 N200 = 1501308

相關函數

_MIND, _MAXD

_MINF

取多筆 32 位元浮點數資料的最小值

FLOAT32 _MIND(INT32S regNum, INT32S cnt)

回傳值表示最小值

regNum 可以使用範圍 0~65534

cnt 可以使用範圍 1~128

回傳值 0.0:表示變數範圍錯誤或最小值為 0.0,其他值為最小值

範例:

```
N100 = 15.64(FLOAT)
N102 = 6573.458(FLOAT)
N104 = -5743.58(FLOAT)
N106 = 45.68(FLOAT)
N200 = _MINF( 100, 4)( FLOAT)
```

執行結果

浮點數 N200 = -5743.58

相關函數

_MAXF, _AVEF

_MAXF

取多筆 32 位元浮點數資料的最大值

FLOAT32 _MAXD(INT32S regNum, INT32S cnt)

回傳值表示最大值

regNum 可以使用範圍 0~65534

cnt 可以使用範圍 1~128

回傳值 0.0:表示變數範圍錯誤或最大值為 0.0,其他值為最大值

範例

N100 = 15.64(FLOAT)

N102 = 6573.458(FLOAT)

N104 = -5743.58(FLOAT)

N106 = 45.68(FLOAT)

N200 = _MAXF(100, 4)(FLOAT)

執行結果

浮點數 N200 = -574358

相關函數

_MINF, _AVEF

_AVEF

取多筆 32 位元浮點數資料的平均值

FLOAT32 _AVEF(INT32S regNum, INT32S cnt)

回傳值表示平均值

regNum 可以使用範圍 0~65534

cnt 可以使用範圍 1~128

計算平均值總值時使用 32 位元浮點數計算,請注意總值是否發生溢位

回傳值 0.0:表示變數範圍錯誤或平均值為 0.0,其他值為平均值

範例

```
N100 = 15.64(FLOAT)
N102 = 6573.458(FLOAT)
N104 = -5743.58(FLOAT)
N106 = 45.68(FLOAT)
N200 = _AVEF( 100, 4)( FLOAT)
```

執行結果

浮點數 N200 = 222.8

相關函數

_MINF, _MAXF

4. 時間相關函式

函數名稱	函數功能	版本需求
_TICK	取得現在的時間滴答值	1.33
_TICK2TIME	時間滴答值轉換至日期時間資料	1.33
_TIME2TICK	日期時間資料轉換至間滴答值	1.33

_TICK

取得現在的時間滴答值

INT32U _TICK()

回傳值表示現在時間滴答值

參數:無

範例

```
N100 = _TICK()(DW)
N102 = N65448(DW)      // (DW)N65448 與 _TICK()功能相同
```

執行結果

```
雙字組 N100 = 1287635383    //數值會依時間點不同而數值也會不同.
雙字組 N102 = 1287635383
```

相關函數

_TICK2TIME, _TIME2TICK

_TICK2TIME

時間滴答值轉換至日期時間資料

INT32S _TICK2TIME(INT32S destRegNum, INT32U tick)

回傳值表示轉換是否成功

destRegNum 可以使用範圍 0~64255, 轉換資料需要 7 個字組暫存器
tick 輸入的 Tick 值
回傳值 0:表示變數範圍錯誤或轉換失敗, 1:表示轉換成功

轉換暫存器資料配置

0: sec [0~59]
1: min [0~59]
2: hour [0~23]
3: day [1~31]
4: month [1~12]
5: year [1970~2069]
6: week [0~6]

範例

```
N100 = _TICK()(INT32U)
_TICK2TIME( 200, (INT32U)N100)
```

執行結果

```
雙字組 N100 = 1287639390    //數值會依時間點不同而數值也會不同.
N200 = 30      //30 秒
N201 = 36      //36 分
N202 = 5       //5 時
N203 =21       //21 日
N204 = 10      //10 月
N205 = 2010    //2010 年
N206 = 4       //星期四
```

相關函數

_TICK, _TIME2TICK

_TIME2TICK

日期時間資料轉換至時間滴答值

INT32S _TIME2TICK(INT32S destRegNum)

回傳值表示為時間滴答值

destRegNum 可以使用範圍 0~64255, 轉換資料需要 7 個字組暫存器
回傳值 0:表示變數範圍錯誤或轉換失敗,其他值為時間滴答值

轉換暫存器資料配置

0: sec [0~59]
1: min [0~59]
2: hour [0~23]
3:day [1~31]
4: month [1~12]
5: year [1970~2069]
6: week [0~6]

範例

```
N200 = 30      //30 秒
N201 = 36      //36 分
N202 = 5       //5 時
N203 = 21      //21 日
N204 = 10      //10 月
N205 = 2010    //2010 年
N206 = 4       //星期四
N100 = _TIME2TICK( 200)(INT32U)
```

執行結果

雙字組 N100 = 1287639390 //數值會依時間點不同而數值也會不同.

相關函數

_TICK, _TICK2TIME

5. 資料轉換相關函式

函數名稱	函數功能	版本需求
_HIBYTE	變數取高位元組資料	1.33
_LOBYTE	變數取低位元組資料	1.33
_HIWORD	變數取高字組資料	1.33
_LOWORD	變數取低字組資料	1.33
_SWAPBYTE	變數高低位元組互換	1.33
_SWAPWORD	變數高低字組互換	1.33
_BIN2GRAY	二進制資料轉換成格雷碼	1.33
_GRAY2BIN	格雷碼資料轉換二進制資料	1.33
_MAKEWORD	二個位元組組合成一個字組	1.33
_MAKEDWORD	二個字組組合成一個雙字組	1.33

_HIBYTE

變數取高位元組資料

INT32U _HIBYTE(INT32U data)

回傳輸入值的高位元組資料

範例

N100 = _HIBYTE(1234H)

執行結果

N100 = 0012H

相關函數

_LOBYTE, _HIWORD, _LOWORD, _MAKEWORD, _MAKEDWORD

_LOBYTE

變數取低位元組資料

INT32U _LOBYTE(INT32U data)

回傳輸入值的低位元組資料

範例:

N100 = _LOBYTE(1234H)

執行結果

N100 = 0034H

相關函數

_HIBYTE, _HIWORD, _LOWORD, _MAKEWORD, _MAKEDWORD

_HIWORD

變數取高字組資料

INT32U _HIWORD(INT32U data)

回傳輸入值的高字組資料

範例

N100 = _HIWORD(12345678H)

執行結果

N100 = 1234H

相關函數

_HIBYTE ,_LOBYTE, _LOWORD, _MAKEWORD, _MAKEDWORD

_LOWORD

變數取低字組資料

INT32U _LOWORD(INT32U data)

回傳輸入值的低字組資料

範例

N100 = _LOWORD(12345678H)

執行結果

N100 = 5678H

相關函數

_HIBYTE ,_LOBYTE, _HIWORD, _MAKEWORD, _MAKEDWORD

_SWAPBYTE

變數高低位元組互換

INT32U _SWAPBYTE(INT32U data)

回傳輸入值的高低位元組交換後的資料

範例

N100 = _SWAPBYTE(1234H)

執行結果

N100 = 3412H

相關函數

_HIBYTE ,_LOBYTE, _HIWORD, _LOWORD, _MAKEWORD, _MAKEDWORD, _SWAPWORD

_SWAPWORD

變數高低字組互換

INT32U _SWAPWORD(INT32U data)

回傳輸入值的高低字組交換後的資料

範例

N100 = _SWAPWORD(12345678H)

執行結果

N100 = 1234H
N101 = 5678H

相關函數

_HIBYTE ,_LOBYTE, _HIWORD, _LOWORD, _MAKEWORD, _MAKEDWORD, _SWAPBYTE

_BIN2GRAY

二進制資料轉換成格雷碼

INT32U _BIN2GRAY(INT32U data)

回傳輸入值的二進制資料轉換成格雷碼後的資料

範例

N100 = _BIN2GRAY(55555555H)(DW)

執行結果

N100 = FFFFH
N101 = 7FFFH

相關函數

_GRAY2BIN

_GRAY2BIN

格雷碼資料轉換二進制資料

INT32U _GRAY2BIN(INT32U data)

回傳輸入值格雷碼資料轉換成二進制後的資料

範例

N100 = _GRAY2BIN(7FFFFFFFH)(DW)

執行結果

N100 = 5555H
N101 = 5555H

相關函數

_BIN2GRAY

_MAKEWORD

二個位元組組合成一個字組

INT32U _MAKEWORD(INT32U hiData, INT32U loData)

回傳組合成一個字組的資料

範例

N100 = _MAKEWORD(12H, 34H)

執行結果

N100 = 1234H

相關函數

_MAKEDWORD

_MAKEDWORD

二個字組組合成一個雙字組

INT32U _MAKEDWORD(INT32U hiData, INT32U loData)

回傳組合成一個雙字組的資料

範例

N100 = _MAKEDWORD(1234H,5678H)(DW)

執行結果

N100 = 1234H
N101 = 5678H

相關函數

_MAKEWORD

6. 計算檢查碼相關函式

函數名稱	函數功能	版本需求
_REGLRC	使用 LRC 方法計算變數區域檢查碼	1.33
_REGCRC16	使用 CRC16 方法計算變數區域檢查碼	1.33
_REGCRC32	使用 CRC32 方法計算變數區域檢查碼	1.33
_REGCCITT	使用 CCITT 方法計算變數區域檢查碼	1.33
_REGBCC	使用 BCC 方法計算變數區域檢查碼	1.33
_REGBCC2	使用 BCC 的補數方法計算變數區域檢查碼	1.33

_REGLRC

使用 LRC 方法計算變數區域檢查碼

INT32S _REGLRC(INT32S regNum, INT32S cnt)

回傳值表示 LRC 檢查碼的數值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為 LRC 檢查碼的數值

範例

```
_STRW( N100, "1234")
N200 = _REGLRC( 100, 4)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0000H
N200 = 202    //202=0CAH
```

相關函數

_REGCRC16, _REGCRC32, _REGCCITT, _REGBCC, _REGBCC2

_REGCRC16

使用 CRC16 方法計算變數區域檢查碼

INT32S _REGCRC16(INT32S regNum, INT32S cnt)

回傳值表示 CRC16 檢查碼的數值

CRC16 algorithm ($x^{16} + x^{15} + x^2 + 1$)

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為 CRC16 檢查碼的數值

範例

```
_STRW( N100, "1234")
N200 = _REGCRC16( 100, 4)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0000H
N200 = 30BAH
```

相關函數

_REGLRC, _REGCRC32, _REGCCITT, _REGBCC, _REGBCC2

_REGCRC32

使用 CRC32 方法計算變數區域檢查碼

INT32S _REGCRC32(INT32S regNum, INT32S cnt)

回傳值表示 CRC32 檢查碼的數值

CRC32 algorithm

(x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x1 + 1)

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為 CRC32 檢查碼的數值

範例

```
_STRW( N100, "1234")
N200 = _REGCRC32( 100, 4)(DW)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0000H
雙字組 N200 = 09BE3E0A3H
```

相關函數

_REGLRC, _REGCRC16, _REGCCITT, _REGBCC, _REGBCC2

_REGCCITT

使用 CCITT 方法計算變數區域檢查碼

INT32S _REGCCITT(INT32S regNum, INT32S cnt)

回傳值表示 CCITT 檢查碼的數值

CRC-CCITT algorithm (X^16 + X^12 + X^5 + 1)

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為 CCITT 檢查碼的數值

範例

```
_STRW( N100, "1234")
N200 = _REGCCITT( 100, 4)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0000H
N200 = 9741H
```

相關函數

_REGLRC, _REGCRC16, _REGCRC32, _REGBCC, _REGBCC2

_REGBCC

使用 BCC 方法計算變數區域檢查碼

INT32S _REGBCC(INT32S regNum, INT32S cnt)

回傳值表示 BCC 檢查碼的數值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為 BCC 檢查碼的數值

範例

```
_STRW( N100, "1234")
N200 = _REGBCC( 100, 4)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0000H
N200 = 0004H
```

相關函數

_REGLRC, _REGCRC16, _REGCRC32, _REGCCITT, _REGBCC2

_REGBCC2

使用 BCC 補數方法計算變數區域檢查碼

INT32S _REGBCC2(INT32S regNum, INT32S cnt)

回傳值表示 BCC 補數檢查碼的數值

regNum 可以使用範圍 0~65535

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或數值為 0,其他值為 BCC 補數檢查碼的數值

範例

```
_STRW( N100, "1234")
N200 = _REGBCC2( 100, 4)
```

執行結果

```
N100 = 0041H
N101 = 0042H
N102 = 0043H
N103 = 0044H
N104 = 0000H
N200 = 00FBH
```

相關函數

_REGLRC, _REGCRC16, _REGCRC32, _REGCCITT, _REGBCC

7. 通訊相關函式

函數名稱	函數功能	版本需求
_LWRITE	資料透過通訊埠送出	1.33
_LWRITEBUFCNT	查詢通訊埠可以寫入型態封包數量	1.37.01

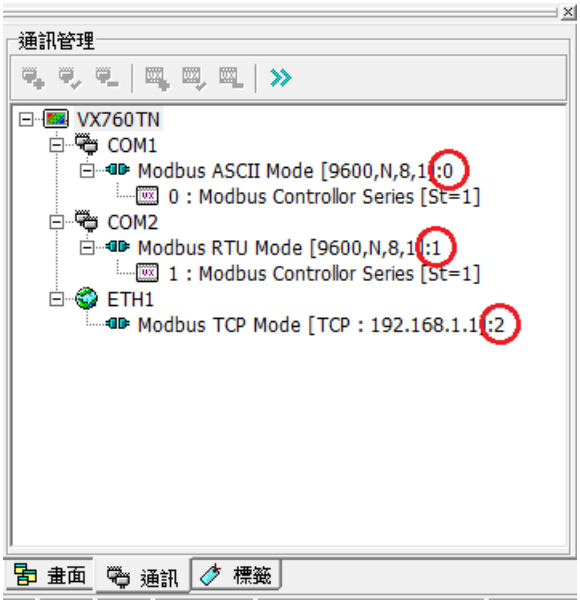
_LWRITE

資料透過通訊埠送出

INT32S _ LWRITE(INT32U handle, INT32S regNum, INT32S cnt)

回傳值表示傳送是否完成

handle 通訊連接的操作值(Handle 顯示在通訊管理視窗[通訊協定]參數後)



regNum 資料來源暫存器位置的編號

cnt 可以使用範圍 1~256

回傳值 0:表示變數範圍錯誤或 handle 錯誤或傳送發生問題,1:表示傳送完成

範例

```
_STR( N1000, "A0=1234\r\n")
N100 = _STRLEN( 1000)
N101= _LWRITE( 0, 1000, N100)
```

執行結果

字串透過通訊埠送出 "A0=1234\r\n"

N100 = 9

N101 = 1

相關函數

_ LWRITEBUFCNT

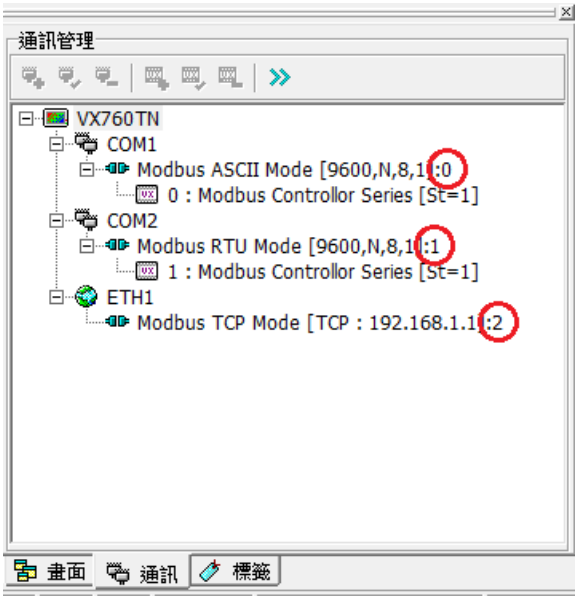
_LWRITEBUFCNT

查詢通訊埠可以寫入型態封包數量

INT32S _ LWRITEBUFCNT(INT32U handle)

回傳值表示傳送是否完成

handle 通訊連接的操作值(Handle 顯示在通訊管理視窗[通訊協定]參數後)



回傳值-1:表示 handle 錯誤,正數:表示回傳寫入型態封包可用空間的數量

範例

N100 = _LWRITEBUFCNT(0)

執行結果

N100 寫入回傳寫入型態封包可用空間的數量.

相關函數

_LWRITE

8. 聲音相關函式

函數名稱	函數功能	版本需求
_SOUND	發出聲音	1.33

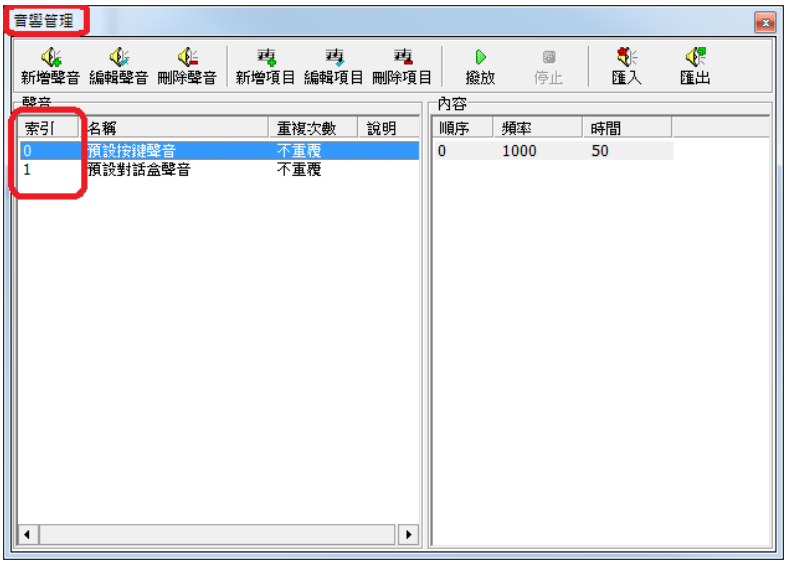
_SOUND

發出已編排的頻率聲音

INT32S _ SOUND(INT32U index)

回傳值表示聲音開關開起及聲音索引值正確

index 發出頻率聲音的索引值



回傳值 0:表示聲音關閉或索引值錯誤,1:表示聲音已開始發聲

範例

_SOUND(0)

執行結果

蜂鳴器或發出索引值 0 所編排的頻率聲音.