
附錄的目錄

附錄 A : ISAGRAF 提供的函式及函式方塊.....	A-1
附錄 A.1: 標準 ISAGRAF 函式方塊.....	A-1
附錄 A.2: 增加新的功能方塊到 ISAGRAF 的 I/O 函式庫.....	A-2
附錄 A.3: 七段式顯示 LED 的定義表格.....	A-4
附錄 A.4: 給 ICP DAS 的 ISAGRAF 控制器使用的功能方塊.....	A-5
A4_20_TO.....	A-5
ANA.....	A-6
ARCREATE.....	A-7
ARRAY_R.....	A-8
ARRAY_W.....	A-9
ARREAD.....	A-10
ARWRITE.....	A-10
ARY_F_R.....	A-11
ARY_F_W.....	A-11
ARY_N_R.....	A-12
ARY_N_W.....	A-12
ARY_W_R.....	A-13
ARY_W_W.....	A-13
BCD_V.....	A-14
BIN2ENG.....	A-14
BIT_WD.....	A-14
BLINK.....	A-15
BOO.....	A-17
CJC.....	A-18
CJC_STS.....	A-19
CJC2.....	A-20
COM_STS.....	A-21
COMARY_R.....	A-22
COMARY_W.....	A-22
COMAY_NW.....	A-23
COMAY_WW.....	A-24
COMCLEAR.....	A-25
COMCLOSE.....	A-25
COMOPEN.....	A-26
COMOPEN2.....	A-27
COMREAD.....	A-28
COMREADY.....	A-29
COMSTR_W.....	A-30
COMWRITE.....	A-31
CRC_16.....	A-32
DI_CNT.....	A-33
DT2MESAG.....	A-33
EBUS_B_R.....	A-34
EBUS_B_W.....	A-34
EBUS_F_R.....	A-35
EBUS_F_W.....	A-35
EBUS_N_R.....	A-36
EBUS_N_W.....	A-36
EBUS_STS.....	A-36
EEP_B_R.....	A-37
EEP_B_W.....	A-37
EEP_BY_R.....	A-38
EEP_BY_W.....	A-38
EEP_EN.....	A-38
EEP_F_R.....	A-39
EEP_F_W.....	A-40

<i>EEP_N_R</i>	A-41
<i>EEP_N_W</i>	A-41
<i>EEP_PR</i>	A-42
<i>EEP_WD_R</i>	A-43
<i>EEP_WD_W</i>	A-43
<i>F_APPEND</i>	A-44
<i>F_CLOSE</i>	A-44
<i>F_COPY</i>	A-45
<i>F_CREAT</i>	A-45
<i>F_DELETE</i>	A-46
<i>F_DIR</i>	A-46
<i>F_END</i>	A-46
<i>F_EOF</i>	A-47
<i>F_READ_B</i>	A-47
<i>F_READ_F</i>	A-47
<i>F_READ_W</i>	A-48
<i>F_ROPEN</i>	A-48
<i>F_SEEK</i>	A-49
<i>F_TRIG</i>	A-49
<i>F_WOPEN</i>	A-50
<i>F_WRIT_B</i>	A-51
<i>F_WRIT_F</i>	A-51
<i>F_WRIT_S</i>	A-52
<i>F_WRIT_W</i>	A-52
<i>FA_READ</i>	A-53
<i>FA_WRITE</i>	A-54
<i>FBUS_B_R</i>	A-55
<i>FBUS_B_W</i>	A-55
<i>FBUS_F_R</i>	A-56
<i>FBUS_F_W</i>	A-56
<i>FBUS_N_R</i>	A-57
<i>FBUS_N_W</i>	A-57
<i>FBUS_STS</i>	A-57
<i>FM_READ</i>	A-58
<i>FM_WRITE</i>	A-59
<i>FR_B</i>	A-60
<i>FR_B_A</i>	A-61
<i>GET_INFO</i>	A-62
<i>GET_SN</i>	A-62
<i>GET_VER</i>	A-62
<i>GETCTS</i>	A-63
<i>I_DICNT</i>	A-64
<i>I_DICNT2</i>	A-65
<i>I_RESET</i>	A-66
<i>I7000_EN</i>	A-66
<i>I8KE_B</i>	A-67
<i>I8KE_B_A</i>	A-68
<i>I8KE_F</i>	A-69
<i>I8KE_F_A</i>	A-70
<i>I8KE_N</i>	A-71
<i>I8KE_N_A</i>	A-72
<i>INP10LED</i>	A-73
<i>INP16LED</i>	A-74
<i>INT_REAL</i>	A-75
<i>INT_STR3</i>	A-75
<i>LONG_WD</i>	A-75
<i>MBUS_B_R</i>	A-76
<i>MBUS_B_W</i>	A-77
<i>MBUS_BRI</i>	A-78
<i>MBUS_N_R</i>	A-79
<i>MBUS_N_W</i>	A-80
<i>MBUS_NRI</i>	A-81
<i>MBUS_R</i>	A-82

<i>MBUS_RI</i>	A-83
<i>MBUS_WB</i>	A-84
<i>MI_BOO</i>	A-85
<i>MI_INP_N</i>	A-85
<i>MI_INP_S</i>	A-86
<i>MI_INT</i>	A-86
<i>MI_REAL</i>	A-87
<i>MI_STR</i>	A-87
<i>MSG_F</i>	A-88
<i>MSG_N</i>	A-89
<i>MSGARY_R</i>	A-90
<i>MSGARY_W</i>	A-90
<i>PID_AL</i>	A-91
<i>PLC_MODE</i>	A-93
<i>PWM_DIS</i>	A-94
<i>PWM_EN</i>	A-94
<i>PWM_EN2</i>	A-94
<i>PWM_OFF</i>	A-94
<i>PWM_ON</i>	A-94
<i>PWM_SET</i>	A-94
<i>PWM_STS</i>	A-94
<i>PWM_STS2</i>	A-94
<i>R_MB_ADR</i>	A-95
<i>R_MB_REL</i>	A-95
<i>R_TRIG</i>	A-96
<i>RDN_A</i>	A-97
<i>RDN_B</i>	A-97
<i>RDN_F</i>	A-97
<i>RDN_N</i>	A-97
<i>RDN_T</i>	A-97
<i>REAL</i>	A-98
<i>REA_STR2</i>	A-99
<i>REAL_INT</i>	A-99
<i>REAL_STR</i>	A-99
<i>RETAIN_A</i>	A-100
<i>RETAIN_B</i>	A-101
<i>RETAIN_F</i>	A-102
<i>RETAIN_N</i>	A-103
<i>RETAIN_T</i>	A-104
<i>RETAIN_X</i>	A-105
<i>S_B_R</i>	A-106
<i>S_B_W</i>	A-106
<i>S_BY_R</i>	A-107
<i>S_BY_W</i>	A-107
<i>S_DL_DIS</i>	A-108
<i>S_DL_EN</i>	A-108
<i>S_DL_RST</i>	A-108
<i>S_DL_STS</i>	A-108
<i>S_FL_AVL</i>	A-109
<i>S_FL_INI</i>	A-110
<i>S_FL_RST</i>	A-110
<i>S_FL_STS</i>	A-111
<i>S_M_R</i>	A-112
<i>S_M_W</i>	A-112
<i>S_MB_ADR</i>	A-113
<i>S_MV</i>	A-114
<i>S_N_R</i>	A-115
<i>S_N_W</i>	A-115
<i>S_R_R</i>	A-116
<i>S_R_W</i>	A-116
<i>S_WD_R</i>	A-117
<i>S_WD_W</i>	A-117
<i>SET_LED</i>	A-118

<i>SETRTS</i>	A-119
<i>SMS_GET</i>	A-120
<i>SMS_GETS</i>	A-120
<i>SMS_SEND</i>	A-121
<i>SMS_STS</i>	A-121
<i>SMS_TEST</i>	A-122
<i>STR_REAL</i>	A-122
<i>SYSDAT_R</i>	A-123
<i>SYSDAT_W</i>	A-124
<i>SYSTEM_R</i>	A-125
<i>SYSTEM_W</i>	A-126
<i>TCP_RECV</i>	A-127
<i>TCP_SEND</i>	A-127
<i>TIME_STR</i>	A-128
<i>TMR</i>	A-128
<i>TO_A4_20</i>	A-129
<i>TO_V0_10</i>	A-130
<i>TOF</i>	A-131
<i>TON</i>	A-132
<i>TP</i>	A-133
<i>TWIN_LED</i>	A-133
<i>UDP_RECV</i>	A-134
<i>UDP_SEND</i>	A-134
<i>V_BCD</i>	A-135
<i>V0_10_TO</i>	A-135
<i>VAL_HEX</i>	A-136
<i>VAL10LED</i>	A-137
<i>VAL16LED</i>	A-138
<i>W_MB_ADR</i>	A-139
<i>W_MB_REL</i>	A-139
<i>WD_BIT</i>	A-140
<i>WD_LONG</i>	A-140

附錄 B : 設定 I-8437/8837, IP-8X47, I-7188EG & μPAC-7186EG 的 IP, MASK, GATEWAY.....B-1

附錄 C : 更新 I-8417/8817/8437/8837 的驅動程式.....C-1

附錄 C.1: 設定 I-8xx7 & I-7188EG 的 COM1 為非 MODBUS-SLAVE PORT.....C-4

附錄 D : 類比 I/O 數值對照表.....D-1

I-87013, I-7013, I-7033, I-7015, M-7015, M-7033, I-87015.....	D-1
I-8017HW(8/16-CH), I-8017H(8-CH), I-8017HS(16-CH).....	D-4
I-87017W, I-87017, I-87017R, I-87017RW, I-7017, I-7017R, M-7017, M-7017R.....	D-5
I-87017ZW, I-87017DW.....	D-6
I-7017RC, M-7017RC, I-87017RC.....	D-7
I-87018Z, I-87018R, I-87018, I-87019R, I-7018Z, I-7018R, I-7018, M-7018, M-7018R, I-7019R, M-7019R (1) ...	D-8
I-87018Z, I-87018R, I-87018, I-87019R, I-7018Z, I-7018R, I-7018, M-7018, M-7018R, I-7019R, M-7019R (2) ...	D-9
I-7021.....	D-11
I-7022.....	D-11
I-7005, M-7005, I-87005.....	D-12
I-8024.....	D-14
I-87024, I-7024.....	D-14
I-87022, I-87026.....	D-15

附錄 E : ISAGRAF 語法參考.....E-1

附錄 F : 如何 ENABLE/DISABLE WP-8XX7 的 LAN2.....F-1

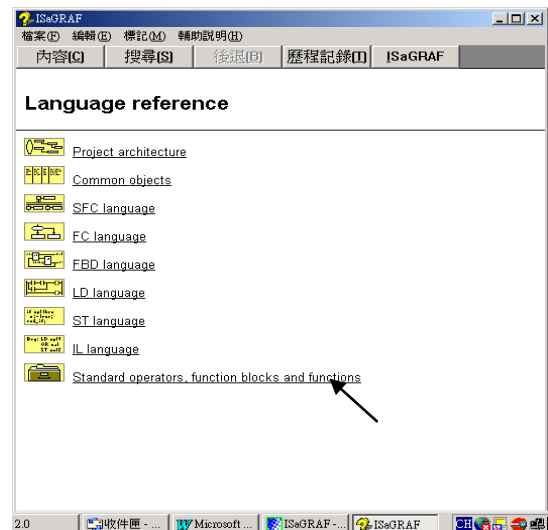
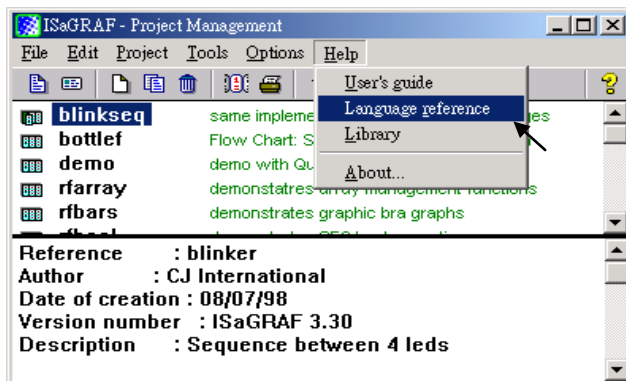
附錄 A : ISaGRAF 提供的函式及函式方塊

附錄 A.1: 標準 ISaGRAF 函式方塊

下面介紹支持 iP-8xx7, I-8xx7, I-7188EG/XG , WP-8xx7, XP-8xx7-CE6 與 VP-2xW7 控制器的 ISaGRAF 標準功能方塊，若有 “*” 或 ”#” 的表示不被 iP-8xx7, I-8xx7 , uPAC-7186EG, I-7188EG/XG 控制器支持，但 WP-8xx7, XP-8xx7-CE6 與 VP-25W7 / 23W7 只有 “#” 不支持。

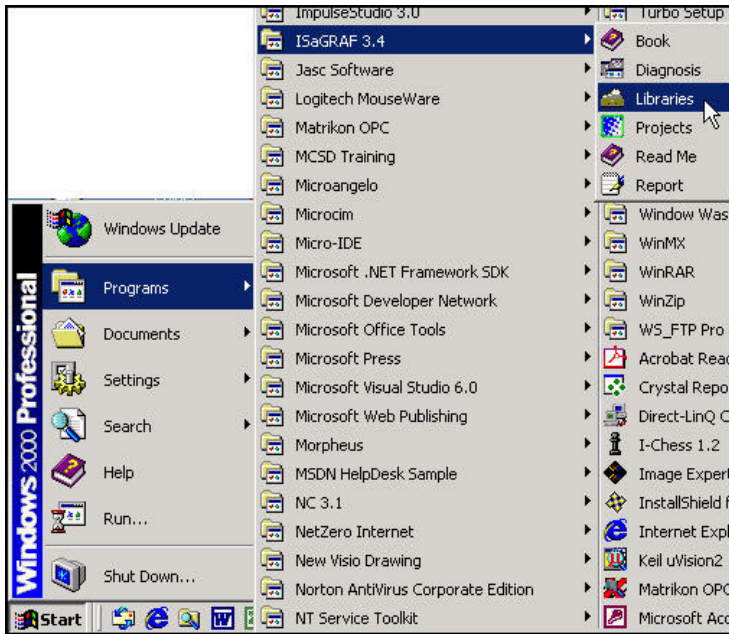
-	ARWRITE	*F_ROPEN	MSG	SHR
& (AND)	ASCII	F_TRIG	MUX4	SIG_GEN
*	ASIN	*F_WOPEN	MUX8	SIN
/	ATAN	*FA_READ	Neg	SQRT
+	AVERAGE	*FA_WRITE	NOT_MASK	SR
<	BLINK	FIND	ODD	STACKINT
<=	BOO	*FM_READ	#OPERATE	#SYSTEM
<>	CAT	*FM_WRITE	OR_MASK	TAN
=	CHAR	HYSTER	POW	TMR
=1 (XOR)	CMP	INSERT	R_TRIG	TOF
>	COS	INTEGRAL	RAND	TON
>=	CTD	LEFT	REAL	TP
>=1 (OR)	CTU	LIM_ALRM	REPLACE	TRUNC
1 gain	CTUD	LIMIT	RIGHT	XOR_MASK
ABS	#DAY_TIME	LOG	ROL	
ACOS	DELETE	MAX	ROR	
ANA	DERIVATE	MID	RS	
AND_MASK	EXPT	MIN	SEL	
ARCREATE	*F_CLOSE	MLEN	SEMA	
ARREAD	*F_EOF	MOD	SHL	

請參考 ISaGRAF 內的線上說明

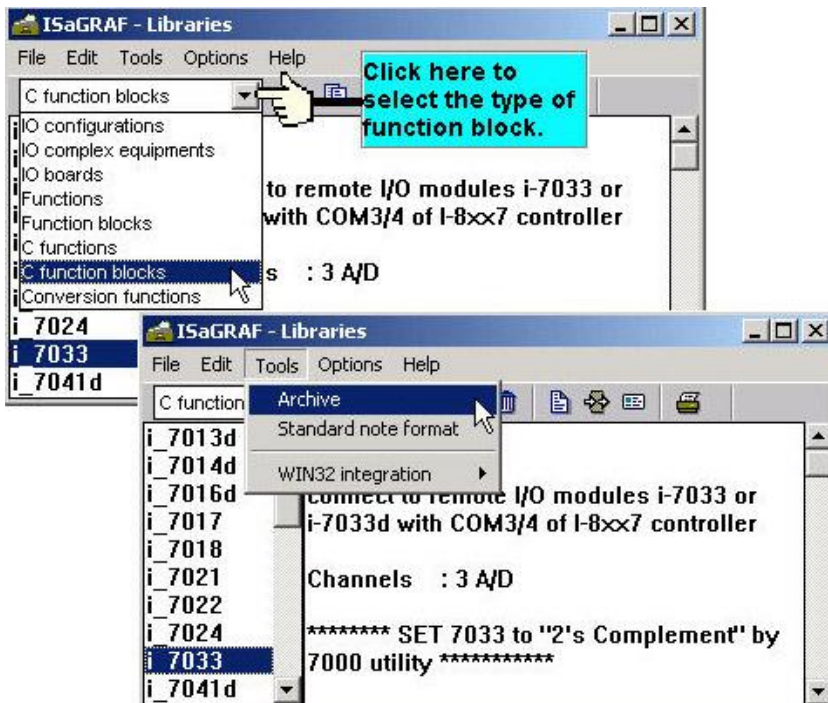


附錄 A.2: 增加新的功能方塊到 ISaGRAF 的 I/O 函式庫

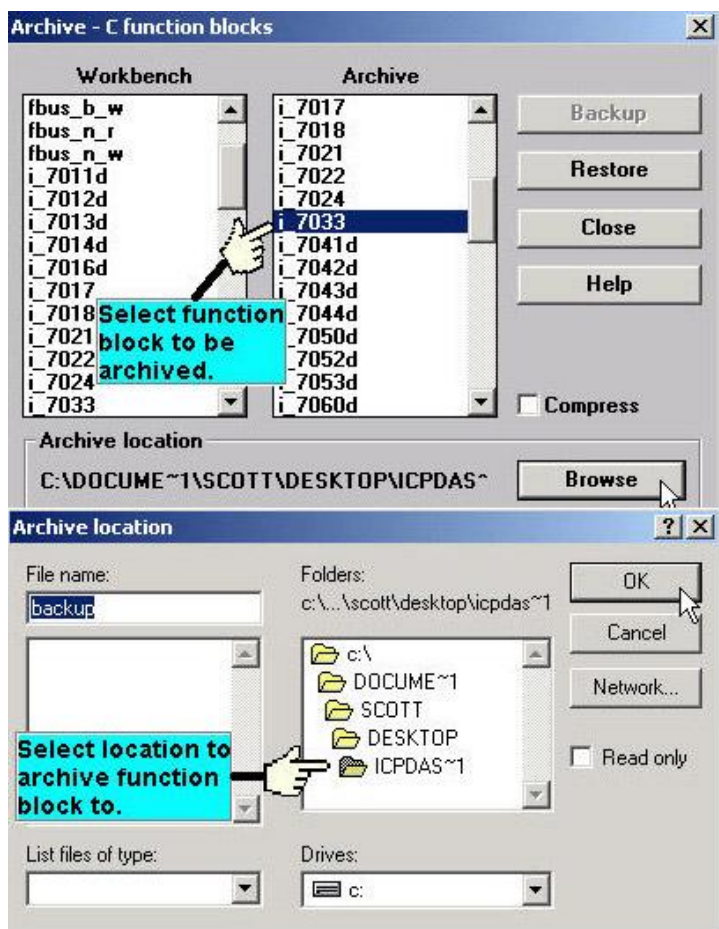
按[Start]→[Program]→[ISaGRAF3.4]的“Libraries”來開始安裝或更新 ISaGRAF 函式或功能方塊。



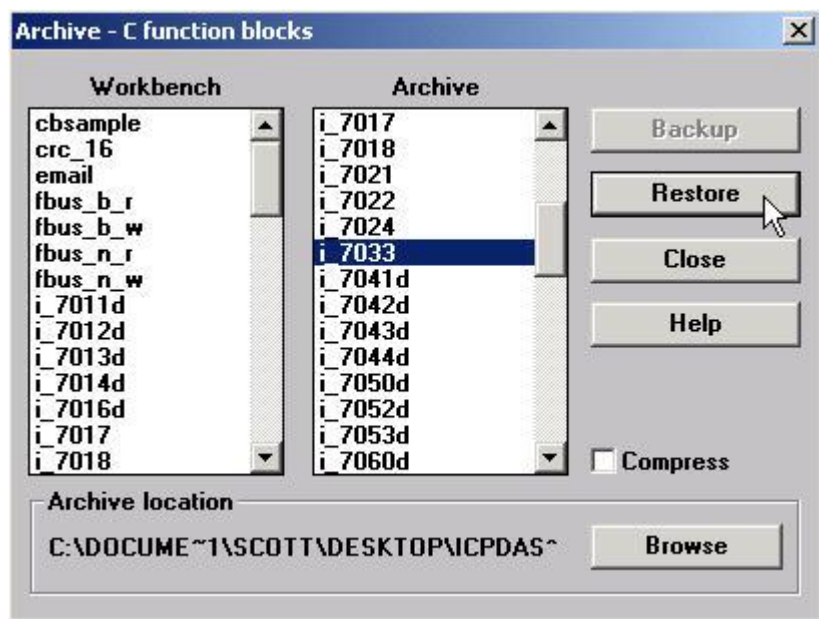
當您按下“Libraries”，“ISaGRAF Libraries”視窗將會開啓。選擇“Tools”裡的“Archive”選項來增加新的 C 函式方塊 (C function Block)。



按下你所要“Archive”的檔名，然後按“Browse”去選擇您所要增加的 C 功能方塊，檔案放置於 CD_ROM 內的 \Napdos\ISaGRAF\ARK\ 內

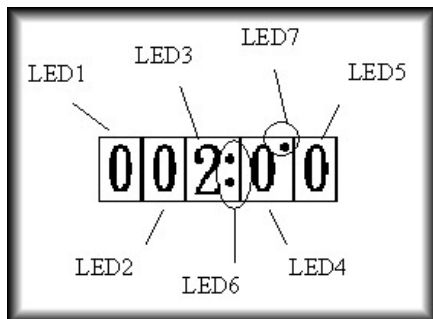


在“Archive”視窗中，選擇您所要增加的 C 功能方塊，按下“Restore”鍵，當您按下“Restore”鍵後方塊即已加入到 ISaGRAF workbench 中了。



附錄 A.3: 七段式顯示 LED 的定義表格

在下面的表格中，我們提供了 iP-8xx7, I-8xx7, uPAC-7186EGD, I-7188EGD & I-7188XGD 控制器中七段式顯示 LED 的定義表格。



LED 6: 若設為 TURE 則顯示":" (冒號):

LED 7: 若設為 TURE 則顯示"." (點)

顯示表: LED 1 至 LED 5

顯示的字元	設的值	顯示的字元	設的值	顯示的字元	設的值
0	0	4.	20	r	40
1	1	5.	21	L	41
2	2	6.	22	n	42
3	3	7.	23	y	43
4	4	8.	24	U	44
5	5	9.	25	P	45
6	6	A.	26	o	46
7	7	b.	27	r.	47
8	8	C.	28	n.	48
9	9	d.	29	y.	49
A	10	E.	30	h.	50
b	11	F.	31	L.	51
C	12		32	U.	52
d	13		33	P.	53
E	14	-	34	o.	54
F	15	_	35	.	55
0.	16	H	36	~.	56
1.	17	h	37	~.	57
2.	18	H.	38	r	其他
3.	19	.	39		

附錄 A.4: 給 ICP DAS 的 ISaGRAF 控制器使用的功能方塊

下面的函式方塊是專門為 iP-8xx7, I-8xx7, I-7188EG/XG, μPAC-7186EG, VP-2117, WP-8xx7, VP-25W7 / 23W7, XP-8xx7-CE6 與 Wincon-8xx7 控制器設計的. 型態若為 Standard_Function 或 Standard_Function Block 表示為 ISaGRAF 提供的標準 Function 與 Function Block. 若為 C_Function 或 C_Function Block 則為 ICP DAS Controller 提供的 Function 與 Function Block.

A4_20_TO

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

轉換 Analog Input 值從 4 – 20 mA 變為 User 自定的工程數值

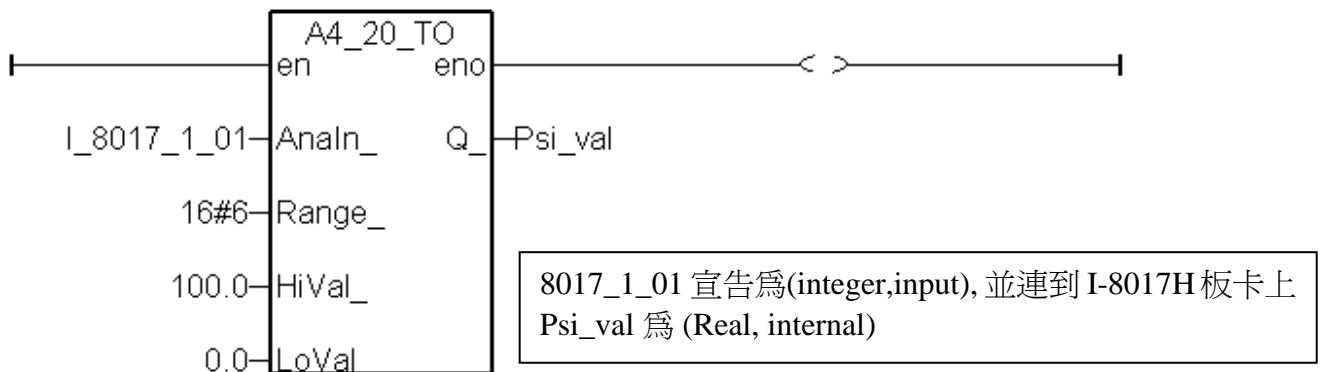
輸入參數 :

AnaIn_	Integer	對應於 Analog Input 的整數變數. 此變數之值通常為-32768 ~ +32767. 與 IO 板卡的設定有關.
Range_	Integer	Analog Input 板卡的 Range 設定值 16#6 : -20 to +20 mA, 16#7 : 4 to +20 mA 16#D : -20 to +20 mA, 16#1A : 0 to +20 mA,
HiVal_	Real	User 自定的工程數值 上限值 (Analog input 為 20 mA 時的值)
LoVal_	Real	User 自定的工程數值 下限值 (Analog input 為 4 mA 時的值) 例如,轉換 I-8017H 的值由 4 ~ 20 mA 變為 0 ~ 100 psi. 請設 HiVal_ 為 100.0, LoVal_ 為 0.0, Range_ 為 16#6 (參照板卡的 range 設定)

傳回值 :

Q_	Real	轉換後的 User 工程數值. 若發生錯誤, 如 Range_ 設錯, 會回傳 1.23E-20
-----------	------	--

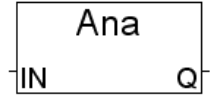
範例 : 轉換 I-8017H 的電流輸入, Range=6: (-20 to +20 mA), 由 (4mA, +20mA)轉到 (0, 100 psi).
4 mA 代表 0 psi, 20 mA 代表 100 psi



- 注意:** 1. 請參考類似的 function: to_A4_20, to_V0_10, A4_20_to, V0_10_to.
2. 使用 A4_20_to, To_A4_20, To_V0_10, V0_10_to 等 function, 需將 driver 更新為 I-7188EG: 2.16 版, I-7188XG: 2.14 版, I-8xx7: 3.18 版, 或更高的版本, 程式才不會有問題 (較舊的 driver 會發生程式 run 一段時間後會停止的現象).

ANA

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

轉換任何資料型態的數值 成爲 Integer

輸入參數 :

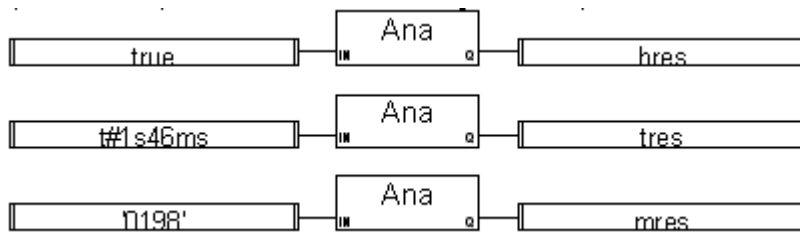
IN Any 任何非整數的數值

傳回值 :

Q Integer 若 IN 爲 FALSE 則傳回 0 / 若 IN 爲 TRUE 則傳回 1
Timer 則傳回毫秒數值
Real 值則傳回整數
String 則傳回對應的十進位數值

範例 :

(* FBD 範例*)



(* ST 相等式 : *)

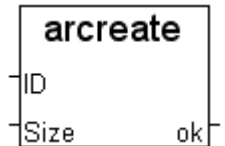
```
bres := ANA (true); (* bres = 1 *)
tres := ANA (t#1s46ms); (* tres = 1046 *)
mres := ANA ('0198'); (* mres = 198 *)
r1 := ANA (3.27); (* r1 = 3 *)
```

(* IL 相等式 : *)

```
LD true
ANA
ST bres
LD t#1s46ms
ANA
ST tres
LD '0198'
ANA
ST mres
```

ARCREATE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

配置一塊整數(或實數)記憶區, 給 ISaGRAF 程式暫存資料用

輸入參數 :

ID	Integer	WinCon 只支援設為 1, 不可設成其他數字
SIZE	Integer	可以是 (1 ~ 3,000,000) WinCon 最多可配置 3,000,000 個 32-bit Integer 記憶空間. (換算為 byte 共 3,000,000 x 4 = 12,000,000 bytes)

傳回值 :

OK	Integer	1: 配置成功; 其他: 失敗.
-----------	---------	------------------

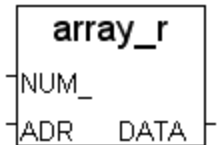
注意 :

1. W-8xx7/8xx6 只能使用 ARcreate 一次, 如下:
在一開機後的第一個 PLC Scan 配置此記憶區:

```
IF INIT THEN
  INIT := False;
  TMP_v := ARcreate(1, 2000000);
END_IF;
(* INIT 初值為 True, TMP_v 型態為 Internal Integer *)
```
2. 請參考第 11.3.10 節的範例說明
3. W-8XX7 的驅動程式, 需為 3.36 版起才有支援 ARCREATE, ARREAD 與 ARWRITE.

ARRAY_R

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

從 byte 陣列內讀出 1 個 byte (unsigned 8-bit)

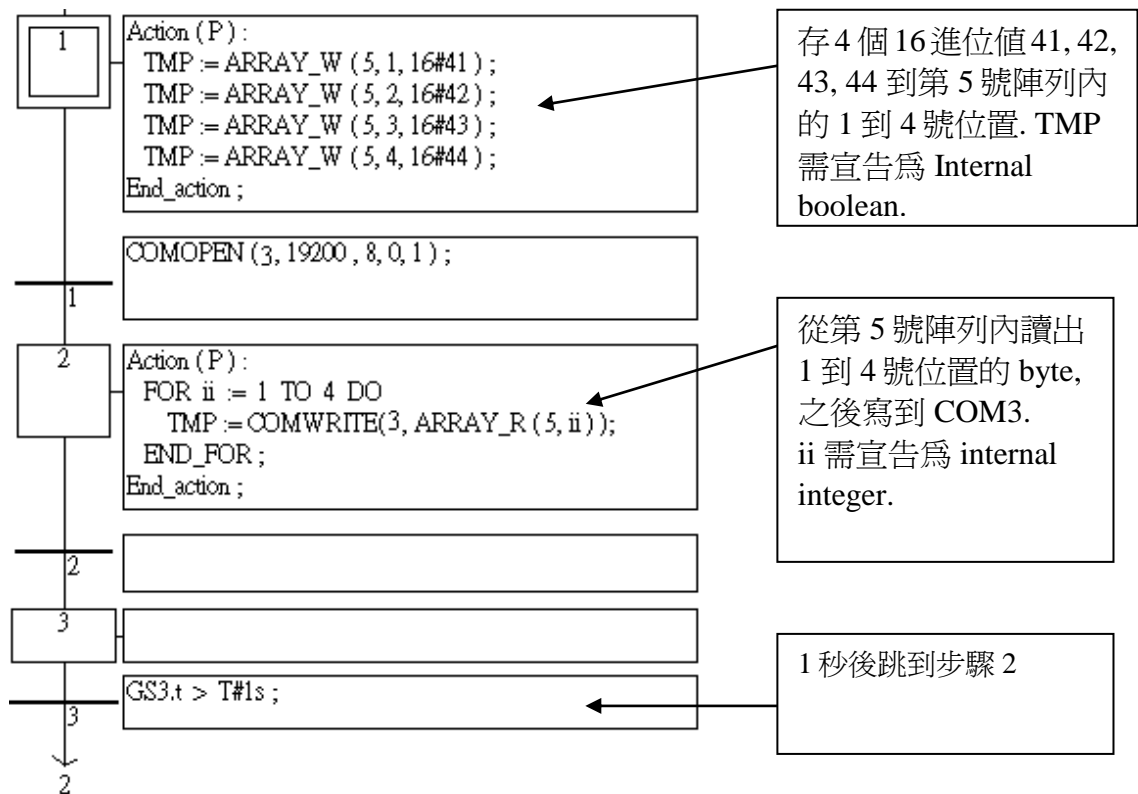
輸入參數：

NUM_	Integer	所使用的陣列編號. 範圍為 I-8xx7 & I-7188EG/XG: 1 到 24 , Wincon-8xx7: 1 到 48
ADR_	Integer	所使用的陣列內的位置編號. 範圍為 I-8xx7 & I-7188EG/XG: 1 到 256 , Wincon-8xx7: 1 到 512

傳回值：

DATA_	Integer	取得的 byte 值 (0~255)
-------	---------	--------------------

範例：



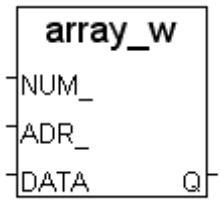
注意: 存到陣列內的資料關電後會消失.

ARRAY_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

存 1 個 byte(unsigned 8-bit) 到 byte 陣列



輸入參數：

NUM_	Integer	所使用的陣列編號, 範圍為 I-8xx7 & I-7188EG/XG: 1 到 24 , Wincon-8xx7: 1 到 48
ADR_	Integer	所使用的陣列內的位置編號, 範圍為 I-8xx7 & I-7188EG/XG: 1 到 256 , Wincon-8xx7: 1 到 512
DATA_	Integer	要存進去的 byte 值, 範圍為 0 到 255.

傳回值：

Q_	Boolean	成功 回傳 TRUE, 失敗回傳 FALSE
-----------	---------	------------------------

範例：請參考“ARRAY_R”的範例

注意：存到陣列內的資料關電後會消失.

ARREAD

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

從 User 配置的整數記憶區讀取一個整數 (32-bit signed)

輸入參數 :

ID	Integer	WinCon 只支援設為 1, 不可設成其他數字
POS	Integer	要讀取哪個位址的整數, 可以是 (1 ~ 3,000,000) 若 POS 超出 ARcreate 配置的數量, 資料會不正確

傳回值 :

Q	Integer	讀到的整數.
----------	---------	--------

ARWRITE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

寫一個整數到 User 配置的記憶區

輸入參數 :

ID	Integer	WinCon 只支援設為 1, 不可設成其他數字
POS	Integer	要寫到哪個位址, 可以是 (1 ~ 3,000,000) 若 POS 超出 ARcreate 配置的數量, 資料不會寫入
IN	Integer	要寫入的整數.

傳回值 :

OK	Integer	1: 成功; 其他值為失敗.
-----------	---------	----------------

注意 :

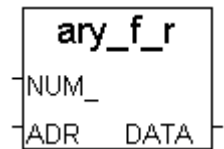
1. 若未使用過 ARcreate 先配置記憶區, ARread 與 ARwrite 的傳回值都是錯的
2. 若要在該記憶區內讀/寫實數, 請配合使用 Real_Int 與 Int_Real 函式 (請參考第 11.3.10 節的範例)
3. W-8XX7 的驅動程式, 需為 3.36 版起才有支援 ARCREATE, ARREAD 與 ARWRITE.

ARY_F_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

從 Float 陣列內讀出 1 個實數值 (32-bit float)



輸入參數 :

NUM_	Integer	所使用的陣列編號. W-8xx7/8xx6 範圍為 1 到 18. 7188EG/XG, I-8xx7 範圍為 1 到 6.
ADR_	Integer	所使用的陣列內的位置編號. 範圍為 1 到 256

傳回值 :

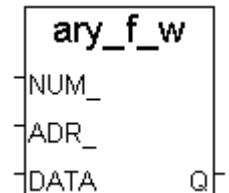
DATA_	Real	取得的實數值 (32-bit float)
-------	------	-----------------------

ARY_F_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

存 1 個實數值(32-bit float) 到 float 陣列



輸入參數 :

NUM_	Integer	所使用的陣列編號. W-8xx7/8xx6 範圍為 1 到 18. 7188EG/XG, I-8xx7 範圍為 1 到 6.
ADR_	Integer	所使用的陣列內的位置編號. 範圍為 1 到 256
DATA_	Real	要存進去的實數值 (32-bit float)

傳回值 :

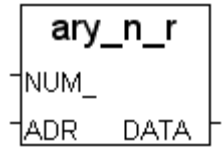
Q_	Boolean	成功 回傳 TRUE, 失敗回傳 FALSE
----	---------	------------------------

注意:

1. 存到陣列內的資料關電後會消失.
2. I-7188EG/XG 與 I-8XX7 的 Float 陣列跟 Integer 陣列使用相同的記憶區, 請小心使用, 當使用 ARY_F_R 讀 REAL 值卻發現該記憶位址內存的是 Integer 資料時, 值會錯誤.
3. 從以下版本起的驅動程式才有支援 ARY_F_R 與 ARY_F_W:
 - I-7188EG: 2.17 版起
 - I-7188XG: 2.15 版起
 - I-8XX7 : 3.19 版起
 - W-8XX7 : 建議更新至 3.36 版或更高版

ARY_N_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 integer 陣列讀出 1 個長整數 (signed 32-bit)

輸入參數 :

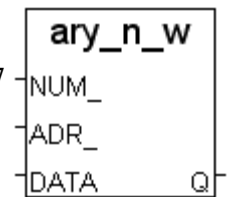
NUM_	Integer	所使用的陣列編號, 範圍為 I-8xx7 & I-7188EG/XG: 1 到 6, Wincon-8xx7: 1 到 18
ADR_	Integer	所使用的陣列內的位置編號, 範圍為 1 到 256

傳回值 :

DATA_	Integer	取得的長整數值
-------	---------	---------

ARY_N_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

存 1 個長整數(signed 32-bit) 到 integer 陣列

輸入參數 :

NUM_	Integer	所使用的陣列編號, 範圍為 I-8xx7 & I-7188EG/XG: 1 到 6, Wincon-8xx7: 1 到 18
ADR_	Integer	所使用的陣列內的位置編號, 範圍為 1 到 256
DATA_	Integer	要存進去的長整數值

傳回值 :

Q_	Boolean	成功 回傳 TRUE, 失敗回傳 FALSE
----	---------	------------------------

注意: 1. Integer 陣列和 word 陣列使用同一塊記憶區. 請小心安排使用.

word 陣列(編號,位置)	Integer 陣列(編號,位置)
(1,1)	(1,1)
(1,2)	
(1,3)	(1,2)
(1,4)	
...	...
...	
(12,255)	(6,256)
(12,256)	
...	...
...	

2. 存到陣列內的資料關電後會消失.

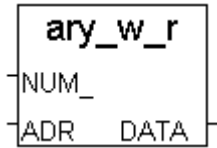
範例: 請參考“ARRAY_R”的範例

ARY_W_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

從 word 陣列讀出 1 個 word (signed 16-bit)



輸入參數：

NUM_ Integer 所使用的陣列編號, 範圍為
I-8xx7 & I-7188EG/XG: 1 到 12, Wincon-8xx7: 1 到 36

ADR_ Integer 所使用的陣列內的位置編號, 範圍為 1 到 256

傳回值：

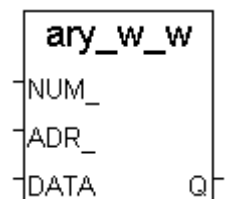
DATA_ Integer 取得的 word 值, 範圍為-32768 到 +32767

ARY_W_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

存 1 個 word (signed 16-bit) 到 word 陣列



輸入參數：

NUM_ Integer 所使用的陣列編號, 範圍為
I-8xx7 & I-7188EG/XG: 1 到 12, Wincon-8xx7: 1 到 36

ADR_ Integer 所使用的陣列內的位置編號, 範圍為 1 到 256

DATA_ Integer 要存進去的 word 值 (-32768 到 +32767)

傳回值：

Q_ Boolean 成功回傳 TRUE, 失敗回傳 FALSE

注意： 1. Integer 陣列和 word 陣列使用同一塊記憶區. 請小心安排使用.

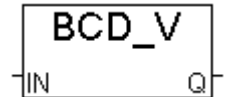
word 陣列(編號,位置)	Integer 陣列(編號,位置)
(1,1)	(1,1)
(1,2)	
(1,3)	(1,2)
(1,4)	
...	...
...	
(12,255)	(6,256)
(12,256)	
...	...
...	

2. 存到陣列內的資料關電後會消失.

範例： 請參考“ARRAY_R”的範例

BCD_V

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

轉換 BCD 值到 1 個 整數值

輸入參數 :

IN_ Integer 要被轉換的 BCD 值

傳回值 :

Q_ Integer 轉換後得到的值,例如,
16#12345 → 12345
18 → 12

BIN2ENG

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

轉換 IO 板卡的 2 的補數值 到 1 個 整數值

輸入參數 :

IN_ Integer 要被轉換的 2's complement 值

HI_2S_ Integer 2's complement 的上限值

LO_2S_ Integer 2's complement 的下限值

HI_EN_ Integer 轉換後的值 的上限值

LO_EN_ Integer 轉換後的值 的下限值

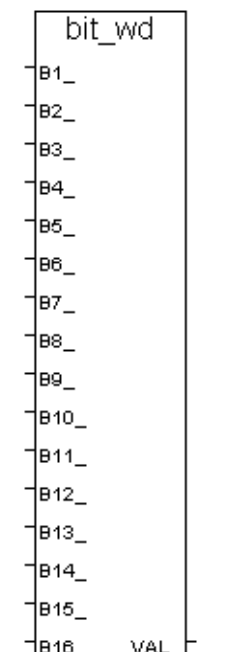
傳回值 :

OUT_ Integer 轉換後得到的值,例如,
HI_2s_ = 32767 , LO_2s_ = -32768, HI_EN_ = 1000, LO_EN_ = -1000
IN_ = 16383 → OUT_ = 500
IN_ = -12345 → OUT_ = -377

注意: HI_2S_ 不可跟 LO_2S_ 值相同, 且二者之值需在 (-32768 到 +32767)區間內

BIT_WD

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

轉換 16 個 Boolean 值到 1 個 word 值 (signed 16-bit)

輸入參數 :

B1_ ~ B16_ Boolean 要被轉換的 16 個 boolean 值

傳回值 :

VAL_ Integer 轉換後得到的 word 值
例如, B1_ 和 B2_ 為 TRUE, 而其它皆為 FALSE,
得到 VAL_ 為 3. 如果 B4_ 為 TRUE, 而其它皆為
FALSE, 得到 VAL_ 為 8

BLINK

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function Block

產生一個 ON_OFF 方波閃爍的訊號

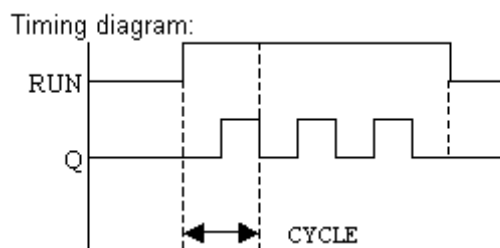
輸入參數 :

RUN	Boolean	模式: TRUE=閃爍; FALSE=不閃爍, 並會將 Q 輸出為 FALSE
CYCLE	Timer	閃爍週期

傳回值 :

Q	Boolean	輸出的 ON_OFF 方波閃爍訊號
----------	---------	-------------------

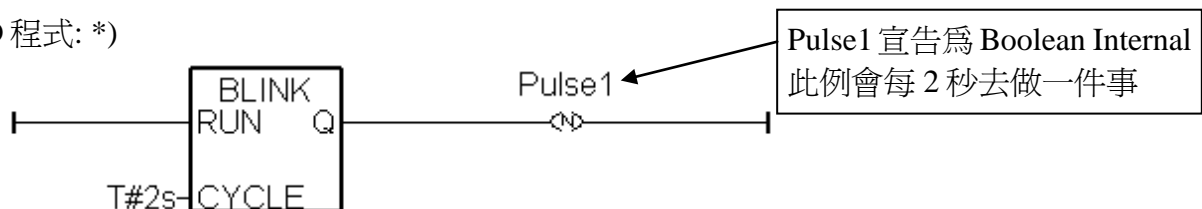
時序圖 :



應用說明:

BLINK 方塊可用來每隔固定一段時間就產生一個 Pulse True, 所以可以使用在每固定一段時間就做一件事的應用上. 如下:

(* LD 程式: *)



(* ST 程式: *)

```
IF Pulse1 THEN
  (* 做一件事 *)
  (* ..... *)
END_IF;
```

但以上的程式在時間間隔較短, 比如小於 200ms 或 PLC Scan Time 較大時會變得不精確. 例如每 50ms 做一件事, 因為 50ms 是比較小的間隔, 跟 PLC Scan Time 比較接近, 就會不準確, 所以可以改成以下方法, 就可以提高準確性.

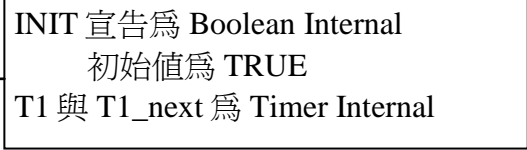
(見下頁)

ST 程式:

```
IF INIT THEN
  INIT := False;
  T1 := T#0s;
  T1_next := T1 + T#50ms;
  Tstart (T1);
END_IF;

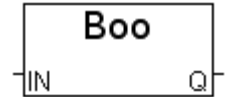
IF T1 >= T1_next THEN
  IF T1 > T#22h THEN
    T1 := T#0s;
    T1_next := T#0s;
  END_IF;
  T1_next := T1_next + T#50ms;
  (* 做一件事 *)
  (* ..... *)
END_IF;
```

INIT 宣告為 Boolean Internal
初始值為 TRUE
T1 與 T1_next 為 Timer Internal



BOO

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

轉換任何資料型態的變數為布林型態

輸入參數 :

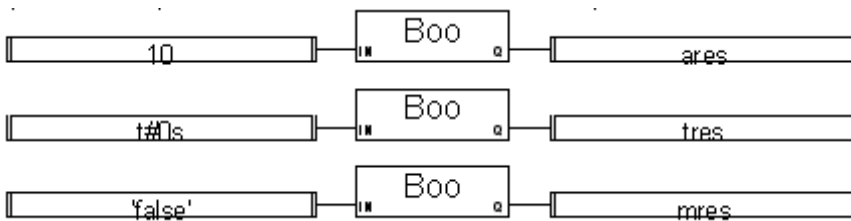
IN Any 任一非布林的值

傳回值 :

Q Boolean 非 0 的數值傳回 TRUE, 0 則傳回 FALSE
'TRUE' message 傳回 TRUE,
'FALSE' message 傳回 FALSE

範例 :

(* FBD 範例 "Convert to Boolean" blocks *)



(* ST 相等式: *)

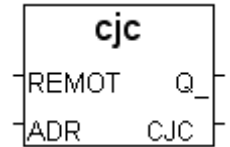
```
ares := BOO (10); (* ares is TRUE *)
tres := BOO (t#0s); (* tres is FALSE *)
mres := BOO ('false'); (* mres is FALSE *)
```

(* IL 相等式: *)

```
LD 10
BOO
ST ares
LD t#0s
BOO
ST tres
LD 'false'
BOO
ST mres
```

CJC

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function Block

讀取 I-7018/7019 和 I-87018/87019 模組的冷端補償值
(Cold-Junction Compensation, 簡稱 CJC)

輸入參數 :

REMOTE_ : Boolean

必需是常數, 不能為變數

若 I-7018/7019 和 I-87018/87019 為遠端 I/O 模組
則 REMOTE_ 為 TRUE.

若 I-87018/87019 插在主要控制器上
則 REMOTE_ 為 FALSE.

ADR_ : Integer

必需是常數, 不能為變數

若 REMOTE_ 為 TRUE,

ADR_ 為遠端 I/O 模組的位址 (1 ~ 255).

若 REMOTE_ 為 FALSE,

ADR_ 為插槽編號 0~7 (W-8xx7/8xx6 則為插槽
1~7).

傳回值 :

Q_ : Boolean

若正常運作, 傳回 TRUE

若 Q_ 為 FALSE, 表示通訊不良, 則下列回傳值無意
義.

CJC_ : Integer

傳回類比輸入值 (2 的補數格式)

D3B4 ---> 0000 ---> 7FFF (十六進位制)

代表值 : -11340 ---> 0 ---> 32767 (十進位制)

溫度範圍 : -45 ---> 0 ---> +130 (攝氏度)

注意 : (非常重要)

1. 若 I-87018 / 87019 是插在主控制器上, 請先連接 87018 / 87019 I/O 卡, 否則設定的 "CJC" 函式方塊會無法運作.
2. 若 I-7018 / 7019 和 I-87018 / 87019 做為 RS-485 遠端 I/O, 請先以 DCON utility 將格式設定為 2 的補數.
3. W-8xx7/8xx6 適用於他的驅動程式 3.21 版本以上

CJC_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function Block

啓動/停止 I-7018/7019 和 I-87018/87019 模組的冷端補償
(Cold-Junction Compensation, 簡稱 CJC)

輸入參數 :

REMOTE_ : Boolean

必需是常數, 不能為變數

若 I-7018/7019 和 I-87018/87019 作為遠端 I/O 模組
則 REMOTE_ 為 TRUE.

若 I-87018/87019 插在主要控制器上
則 REMOTE_ 為 FALSE.

ADR_ : Integer

必需是常數, 不能為變數

若 REMOTE_ 為 TRUE,

ADR_ 為遠端 I/O 模組的位址 (1 ~ 255).

若 REMOTE_ 為 FALSE,

ADR_ 為插槽編號 0~7.

SET_STS_ : Boolean

若 SET_STS_ 為 FALSE, 可停止 CJC 補償

若 SET_STS_ 為 TRUE, 可啓動 CJC 補償

傳回值 :

Q_ : Boolean

若正常運作, 傳回 TRUE

若 Q_ 為 FALSE, 表示通訊不良, 下列回傳值無意義.

NOW_STS_ : Boolean

若 NOW_STS_ 為 FALSE, CJC 補償目前狀態為停止

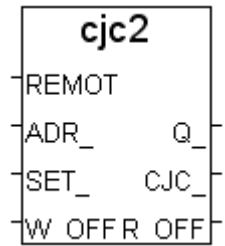
若 NOW_STS_ 為 TRUE, CJC 補償目前狀態為啓動

注意 (非常重要):

1. 若 I-87018 / 87019 是插在主控制器上, 請先連接 87018 / 87019 I/O 卡, "CJC_STS" 函式方塊才會運作.
2. 若是使用為 RS-485 Remote I/O, 連接之前請注意下列各項:
 - A. 在 I/O 模組端, 請使用 "DCON Utility" 來設定:
 - *1. 設定 "address" 為獨一無二的位址編號. (1~255)
 - *2. 類比輸入模塊需設為 "2's complement" 格式, 類比輸出模塊需設為 "Engineering" 格式
 - *3. 設定通訊參數的 "baud rate" 及 "8,N,1"
 - *4. 若為類比板卡要設定為 "Range Type"
 - B. 在 I-8xx7/I-7188EG/I-7188XG /W-8xx7 主控端:
 - *1. 連結 I/O 請選擇 "complex equipment" 及 "bus7000B", 並設定其 "baud rate" 與 I/O 模組的 "baud rate" 相同, Checksum 也需相同.
 - *2. I-8xx7, 7188EG/XG 最多可連接 64 個遠程 I/O 模組, W-8xx7 最多 255 個
 - *3. 程式內連接 RS-485 Remote I/O 模組請選用 i_7*** function block 或 i_87*** Function Block.

CJC2

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function Block

讀取 I-7018/7019 和 I-87018/87019 模組偏移的冷端補償值

輸入參數 :

REMOTE_ :	Boolean	必需是常數, 不能為變數 若 I-7018/7019 和 I-87018/87019 作為遠端 I/O 模組 則 REMOTE_ 為 TRUE. 若 I-87018/87019 插在主要控制器上 則 REMOTE_ 為 FALSE.
ADR_ :	Integer	必需是常數, 不能為變數 若 REMOTE_ 為 TRUE, ADR_ 為遠端 I/O 模組的位址 (1 ~ 255). 若 REMOTE_ 為 FALSE, ADR_ 為插槽編號 0~7.
SET_ :	Boolean	若為 TRUE 則開始設定 CJC 溫度的偏移
W_OFFSET_ :	Integer	CJC 溫度的偏移量, 十進位值, -4096 ~ +4096, 以 0.01 攝氏 度為一單位

傳回值 :

Q_ :	Boolean	若正常運作, 傳回 TRUE 若 Q_ 為 FALSE, 表示通訊不良, 下列回傳值無意義.
CJC_ :	Integer	傳回類比輸入值 (2 的補數格式) D3B4 ---> 0000 ---> 7FFF (十六進位制) 代表值 : -11340 ---> 0 ---> 32767 (十進位制) 溫度範圍 : -45 ---> 0 ---> +130 (攝氏度)
R_OFFSET_ :	Integer	讀取偏移的 CJC 溫度值

注意 (非常重要):

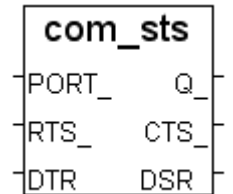
1. 若 I-87018 / 87019 是插在主控制器上, 請先連接 87018 / 87019 I/O 卡, "CJC2" 函式方塊才會運作.
2. 若是使用為 RS-485 Remote I/O, 連接之前請注意下列各項:
 - A. 在 I/O 模組端, 請使用 "DCON Utility" 來設定:
 - *1. 設定 "address" 為獨一無二的位址編號. (1~255)
 - *2. 類比輸入模塊需設為 "2's complement" 格式, 類比輸出模塊需設為 "Engineering" 格式
 - *3. 設定通訊參數的 "baud rate" 及 "8,N,1"
 - *4. 若為類比板卡要設定為 "Range Type"
 - B. 在 I-8xx7/I-7188EG/I-7188XG /W-8xx7 主控端:
 - *1. 連結 I/O 請選擇 "complex equipment" 及 "bus7000B", 並設定其 "baud rate" 與 I/O 模組的 "baud rate" 相同, Checksum 也需相同.
 - *2. I-8xx7, 7188EG/XG 最多可連接 64 個遠程 I/O 模組, W-8xx7 最多 255 個
 - *3. 程式內連接 RS-485 Remote I/O 模組請選用 i_7*** function block 或 i_87*** Function Block.

COM_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

設定COM埠的 RTS, DTR 並取得 CTS, DSR 狀態



輸入參數 :

PORT_ :	Integer	COM埠號. 3:COM3, 4:COM4, 5:COM5... iP-8xx7/8437/8837 有效埠號: 3, 4 & 5 I-7188XG & I-7188EG 有效埠號: 4
RTS_ :	Boolean	TRUE : 設定 RTS 啟用, FALSE : 設定 RTS 閒置
DTR_ :	Boolean	TRUE : 設定 DTR 啟用, FALSE : 設定 DTR 閒置

傳回值 :

Q_ :	Boolean	TRUE 表示 OK , FALSE 表示沒有成功
CTS_ :	Boolean	取得 CTS 狀態
DSR_ :	Boolean	取得 DSR 狀態

注意 :

1. 使用 Comopen() 開啓 COM 埠.

COMARY_R

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

從 COM 埠讀取全部已經收到的 byte (unsigned 8-bit), 並存入 byte 陣列內

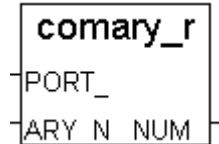
輸入參數 :

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...
ARY_NO_	Integer	Byte 陣列編號(I-8xx7 & I-7188EG/XG:1-24, W-8xx7:1-48), 該陣列會儲存從 Com 收到的 byte

傳回值 :

NUM_	Integer	從 Com 埠讀到的 byte 的數量 (0~256)
-------------	---------	-----------------------------

重要 : 串列通訊資料需要時間來傳遞, COMARY_R 被呼叫當時若已經有多少 byte 傳進來, 就會存入同樣數量的 byte 到指定的陣列內, 還未傳進來的資料, 當然不會在陣列內 (因為還沒收到).



COMARY_W

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

寫出 byte 陣列內的 數個 byte (unsigned 8-bit) 到 1 個 COM 埠

輸入參數 :

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...
ARY_NO_	Integer	要寫出的 Byte 陣列編號(I-8xx7 & I-7188EG/XG:1 - 24) , (W-8xx7:1 - 48)
NUM_	Integer	從該 Byte 陣列的第 1 號位置起寫出多少個 byte

傳回值 :

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

注意:

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

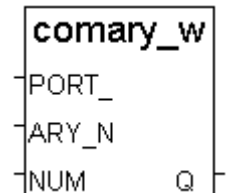
* I-8xx7:

- ComPort No. on slot 0: Com5 ~ Com8
- ComPort No. on slot 1: Com9 ~ Com12
- ComPort No. on slot 2: Com13 ~ Com16
- ComPort No. on slot 3: Com17 ~ Com20
- ComPort No. on slot 4 ~ 7 不支援

範例:

請參考第 11 章 - Demo_21, 22 & 23.

請參考附錄 A.4 : “ARRAY_R” & “ARRAY_W” 的說明



COMAY_NW

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

寫出 Integer 陣列內的 數個長整數 (signed 32 bit) 到 1 個 COM 埠

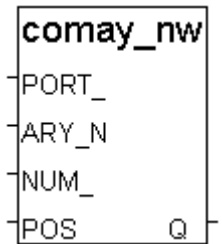
每個長整數包含 4 個 byte. 而且是有正負號的.

寫出長整數到 Com 會以 INTEL 格式內的長整數排法來寫出.

[lowest byte] [] [] [highest byte]

例: 假如有 3 個長整數要寫出, 第 1 個為 16#04030201 (67,305,985), 第 2 個為 16#08070605 (134,678,021), 第 3 個為 16#FFFFFFFE (-2).

則寫出的 12 個 byte 依序為 [01] [02] [03] [04] [05] [06] [07] [08] [FE] [FF] [FF] [FF]



輸入參數 :

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20, I-7188EG:1~8, I-7188XG:2~8 W-8xx7:2,3, or ...
ARY_NO_	Integer	要寫出的 integer 陣列編號(I-8xx7 & I-7188EG/XG:1 - 6), (W-8xx7:1 - 18)
NUM_	Integer	要寫出多少個 integer
POS_	Integer	從該 integer 陣列內的第幾號位置起開始寫出 (1-256) 假如 POS_+NUM_ > 257, 則只有 (257-POS_) 個長整數會寫出 例. POS_=255, NUM_=3, 則只有 2 個長整數會寫出, 分別為第 255 及 256 號位置.

傳回值 :

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

注意:

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

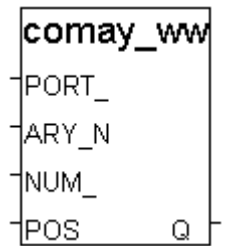
* I-8xx7:

- ComPort No. on slot 0: Com5 ~ Com8
- ComPort No. on slot 1: Com9 ~ Com12
- ComPort No. on slot 2: Com13 ~ Com16
- ComPort No. on slot 3: Com17 ~ Com20
- ComPort No. on slot 4 ~ 7 不支援

* Integer 陣列和 word 陣列使用同一塊記憶區. 請小心安排使用 (參考 Ary_N_W)

COMAY_WW

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫出 Word 陣列內的 數個 Word (signed 16 bit) 到 1 個 COM 埠

每個 Word 包含 2 個 byte. 而且是有正負號的 (-32768 ~ +32767).

寫出 Word 到 Com 會以 INTEL 格式內的 word 排法來寫出.

[low byte] [high byte]

例: 假如有 3 個 word 要寫出, 第 1 個為 16#0403 (1,027), 第 2 個為 16#0807 (2,055), 第 3 個為 16#FFFE (-2).

則寫出的 6 個 byte 依序為 [03] [04] [07] [08] [FE] [FF]

輸入參數 :

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20, I-7188EG:1~8, I-7188XG:2~8 W-8xx7:2,3, or ...
ARY_NO_	Integer	要寫出的 word 陣列編號(I-8xx7 & I-7188EG/XG:1 - 12), (W-8xx7:1 - 36)
NUM_	Integer	要寫出多少個 word
POS_	Integer	從該 word 陣列內的第幾號位置起開始寫出 (1-256) 假如 POS_+NUM_ > 257, 則只有 (257-POS_) 個 word 會寫出 例. POS_=255, NUM_=3, 則只有 2 個 word 會寫出, 分別為第 255 及 256 號位置.

傳回值 :

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

注意:

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

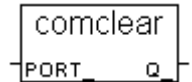
* I-8xx7:

- ComPort No. on slot 0: Com5 ~ Com8
- ComPort No. on slot 1: Com9 ~ Com12
- ComPort No. on slot 2: Com13 ~ Com16
- ComPort No. on slot 3: Com17 ~ Com20
- ComPort No. on slot 4 ~ 7 不支援

* Integer 陣列和 word 陣列使用同一塊記憶區. 請小心安排使用 (參考 Ary_W_W)

COMCLEAR

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

清空 Com埠的 接收 buffer

輸入參數 :

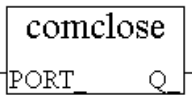
PORT_ Integer port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...

傳回值 :

Q_ Boolean 成功回傳 TRUE

COMCLOSE

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

關閉 COM 埠

輸入參數 :

PORT_ Integer port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...

傳回值 :

Q_ Boolean 成功回傳 TRUE

注意:

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為 非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7的COM2或COM3, 請先設定他們為 非 Modbus-RTU port (參考W-8xx7的“快速上手手冊”)

* I-8xx7:

- ComPort No. on slot 0: Com5 ~ Com8
- ComPort No. on slot 1: Com9 ~ Com12
- ComPort No. on slot 2: Com13 ~ Com16
- ComPort No. on slot 3: Com17 ~ Com20
- ComPort No. on slot 4 ~ 7 不支援

範例:

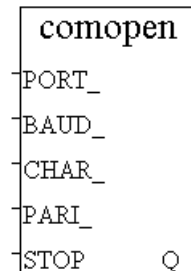
請參考 COMOPEN 的範例

COMOPEN

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

開啓 COM 埠



輸入參數：

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20, I-7188EG:1~8, I-7188XG:2~8 W-8xx7:2,3, or ...
BAUD_	Integer	通訊速率, 可設成 300, 600, 1200, 2400,4800, 9600, 19200, 38400, 57600, 115200
CHAR_	Integer	字元大小, 7 或 8
PARI_	Integer	同位檢查, 0: 沒有, 1: 偶數, 2: 奇數, 3: 標記, 4: 空白字元 3 和 4 只適用於 I-8xx7: COM3~20; I-7188EG/XG:

COM3~8;

Wincon-8xx7: COM2, 或, ...

STOP_	Integer	停止位元大小, 1 或 2
--------------	---------	---------------

傳回值：

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

注意：

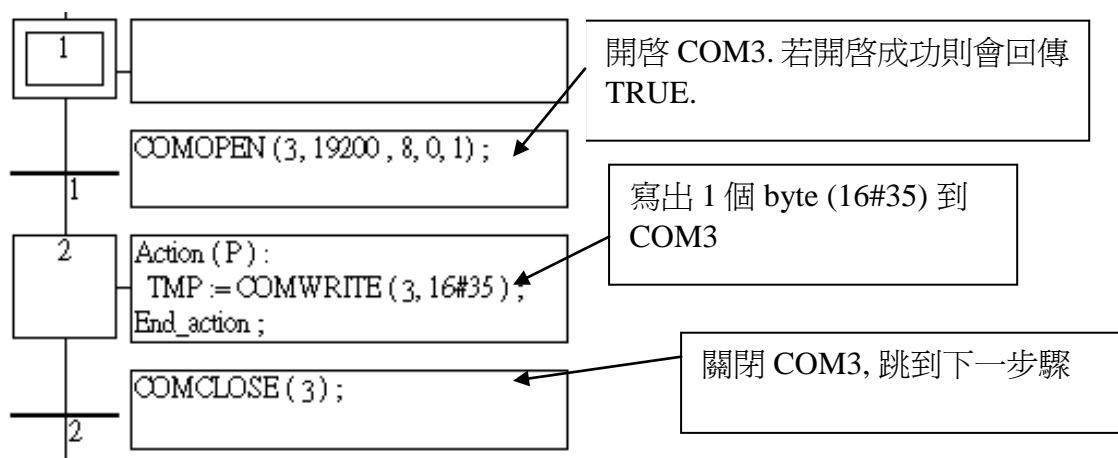
- * 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)
- * 若使用 W-8xx7的COM2或COM3, 請先設定他們為非 Modbus-RTU port (參考W-8xx7的“快速上手手冊”)

* I-8xx7:

- ComPort No. on slot 0: Com5 ~ Com8
- ComPort No. on slot 1: Com9 ~ Com12
- ComPort No. on slot 2: Com13 ~ Com16
- ComPort No. on slot 3: Com17 ~ Com20
- ComPort No. on slot 4 ~ 7 不支援

範例：

請參考第 11 章 - Demo_21, 22 & 23.



COMOPEN2

□ I-8x17 □ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

開啓可做 Flow Control 的 RS232 埠

輸入參數：

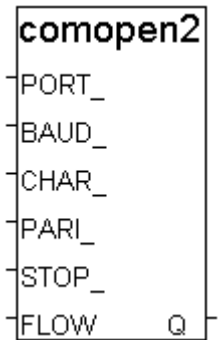
PORT_	Integer	port 編號, I-7188EG/XG:3~8; W-8xx7:2, 或 ...
BAUD_	Integer	通訊速率, 可設成 300, 600, 1200, 2400,4800, 9600, 19200, 38400, 57600, 115200
CHAR_	Integer	字元大小, 7 或 8
PARI_	Integer	同位檢查, 0: 沒有, 1: 偶數, 2: 奇數, 3: 標記, 4: 空白字元 3 和 4 只適用於 I-8xx7: COM3 ~ 20; I-7188EG/XG: COM3 ~8; Wincon-8xx7: COM2, 或, ...
STOP_	Integer	停止位元大小, 1 或 2
FLOW_	Boolean	True: 硬體 flow control (CTS / RTS) (7188EG/XG 3 ~ 5), False: 軟體 flow control (XON / XOF) (7188EG/XG 3 ~ 8)

傳回值：

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

注意：

若使用 W-8xx7 的 COM2, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)



COMREAD

comread

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW

PORT DATA

型態：C_Function

從 Com 埠讀取 1 個 byte (unsigned 8-bit)

輸入參數：

PORT_ Integer port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...

傳回值：

Q_ Integer 讀到的 byte 值 (0~255)

注意：

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

* I-8xx7:

ComPort No. on slot 0: Com5 ~ Com8

ComPort No. on slot 1: Com9 ~ Com12

ComPort No. on slot 2: Com13 ~ Com16

ComPort No. on slot 3: Com17 ~ Com20

ComPort No. on slot 4 ~ 7 不支援

*** 需先使用 COMREADY 去測試有無資料從 COM 進來. 有, 才能使用, 不然使用本 Function 可能會造成通訊鎖死的狀況(Dead Lock).**

範例：

請參考“COMREADY”的範例

COMREADY

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

測試 COM 埠有無資料進來

輸入參數 :

PORT_ Integer port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...

傳回值 :

Q_ Boolean 只要有資料進來就回傳 TRUE. (即使只進來一個 byte)

注意:

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

* I-8xx7:

ComPort No. on slot 0: Com5 ~ Com8

ComPort No. on slot 1: Com9 ~ Com12

ComPort No. on slot 2: Com13 ~ Com16

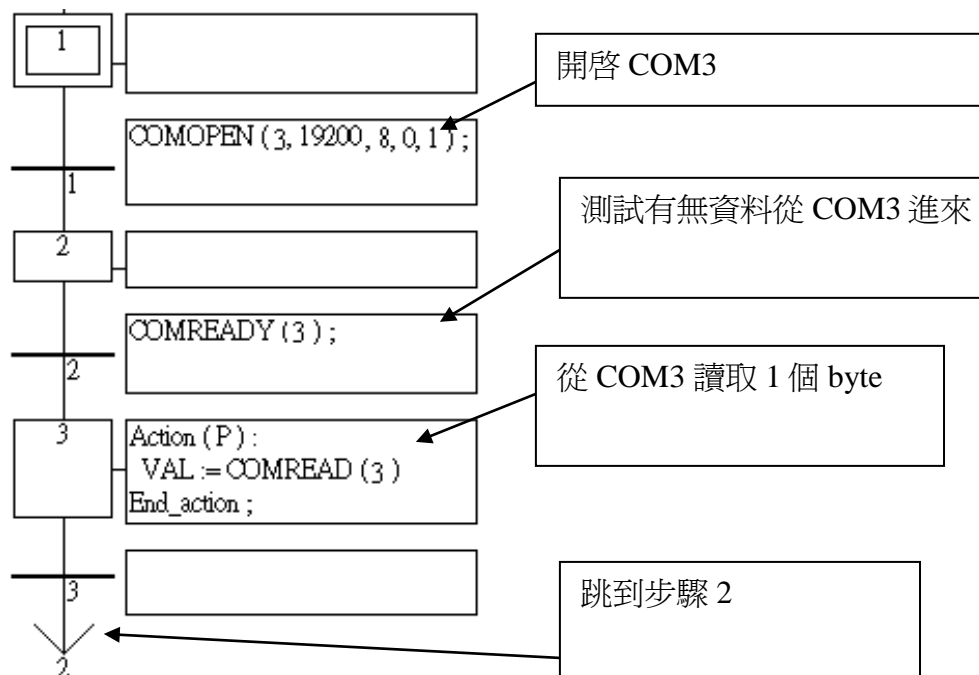
ComPort No. on slot 3: Com17 ~ Com20

ComPort No. on slot 4 ~ 7 不支援

*** 使用 COMREAD 之前要先使用 COMREADY 去測試有無資料從 COM 進來. 有, 才能使用, 不然使用 COMREAD 可能會照成通訊鎖死的狀況(Dead Lock)**

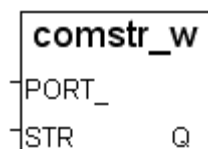
範例:

請參考第 11 章 - Demo_21, 22 & 23.



COMSTR_W

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

寫 1 個字串到 COM 埠

輸入參數：

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20; I-7188EG:1~8; I-7188XG:2~8; W-8xx7:2,3, 或 ...
STR_	Message	要寫出的字串(最大長度為 255).

傳回值：

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

注意：

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)

* 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

* I-8xx7:

ComPort No. on slot 0: Com5 ~ Com8

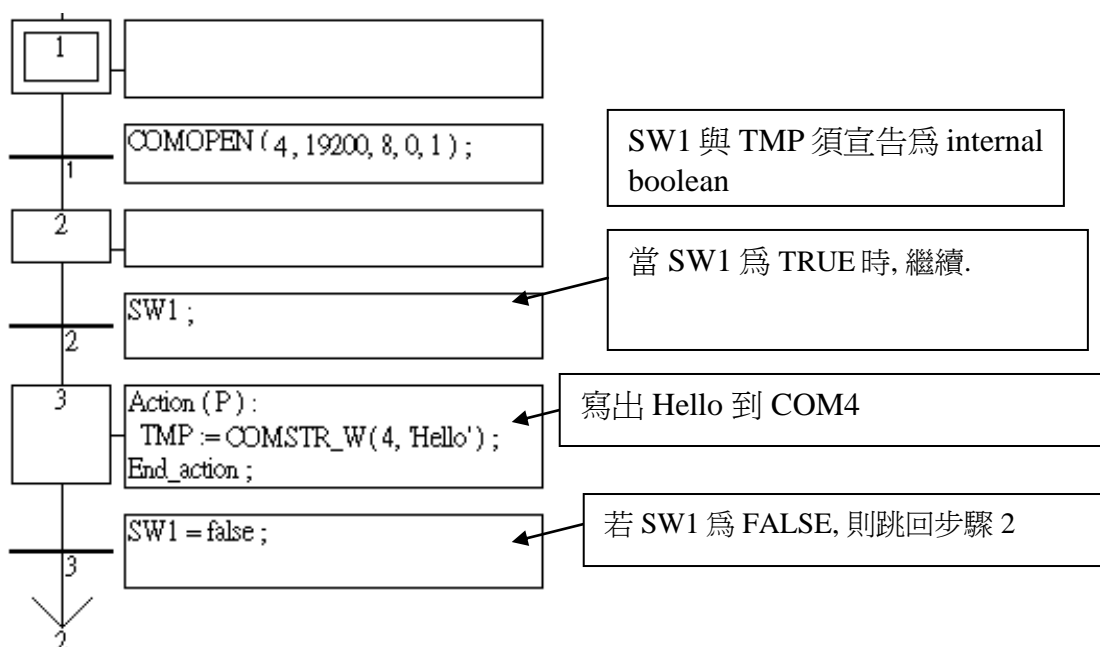
ComPort No. on slot 1: Com9 ~ Com12

ComPort No. on slot 2: Com13 ~ Com16

ComPort No. on slot 3: Com17 ~ Com20

ComPort No. on slot 4 ~ 7 不支援

範例：



COMWRITE

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

寫 1 個 byte (unsigned 8-bit) 到 COM 埠

輸入參數：

PORT_	Integer	port 編號, I-8xx7:1, 3 ~ 20, I-7188EG:1~8, I-7188XG:2~8 W-8xx7:2,3, or ...
DATA_	Integer	要寫出的 byte (0 ~ 255)

傳回值：

Q_	Boolean	成功回傳 TRUE
-----------	---------	-----------

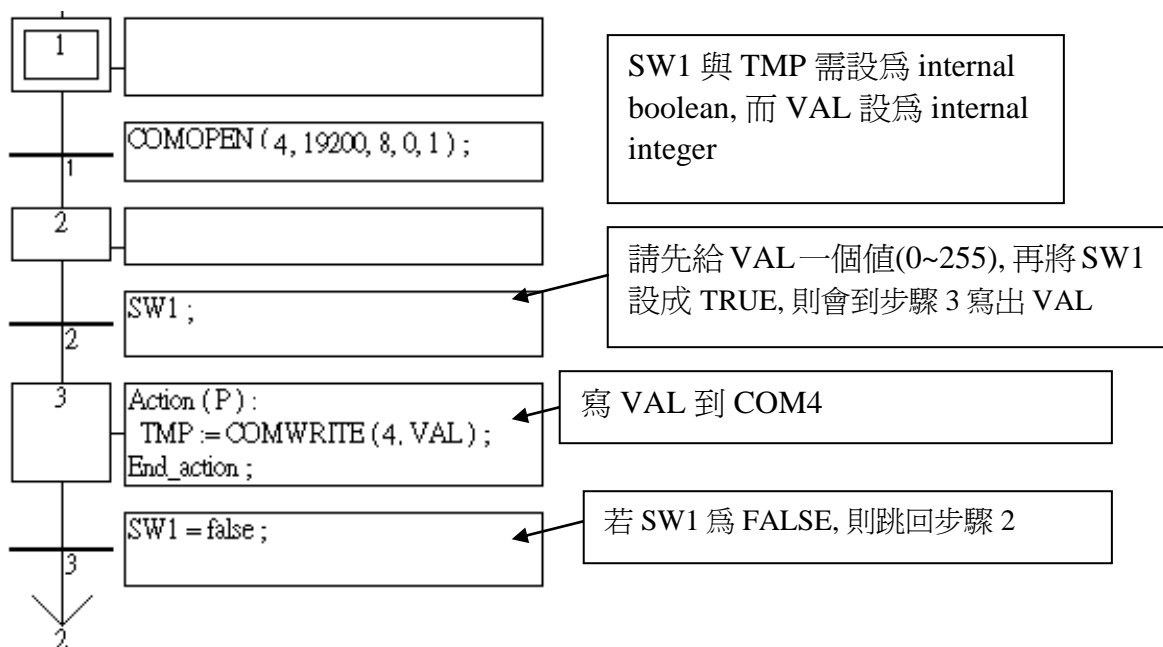
注意：

* 若使用 I-8xx7 & I-7188EG 的 COM1, 請先設定 COM1 為非 Modbus-RTU port(參考附錄 C.1)
 * 若使用 W-8xx7 的 COM2 或 COM3, 請先設定他們為非 Modbus-RTU port (參考 W-8xx7 的“快速上手手冊”)

* I-8xx7:

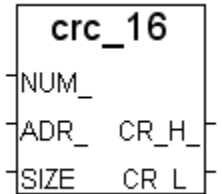
ComPort No. on slot 0: Com5 ~ Com8
 ComPort No. on slot 1: Com9 ~ Com12
 ComPort No. on slot 2: Com13 ~ Com16
 ComPort No. on slot 3: Com17 ~ Com20
 ComPort No. on slot 4 ~ 7 不支援

範例：



CRC_16

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
採用第 5.3 節的演算法。



型態 : C_Function Block

計算 CRC-16 檢查碼

輸入參數 :

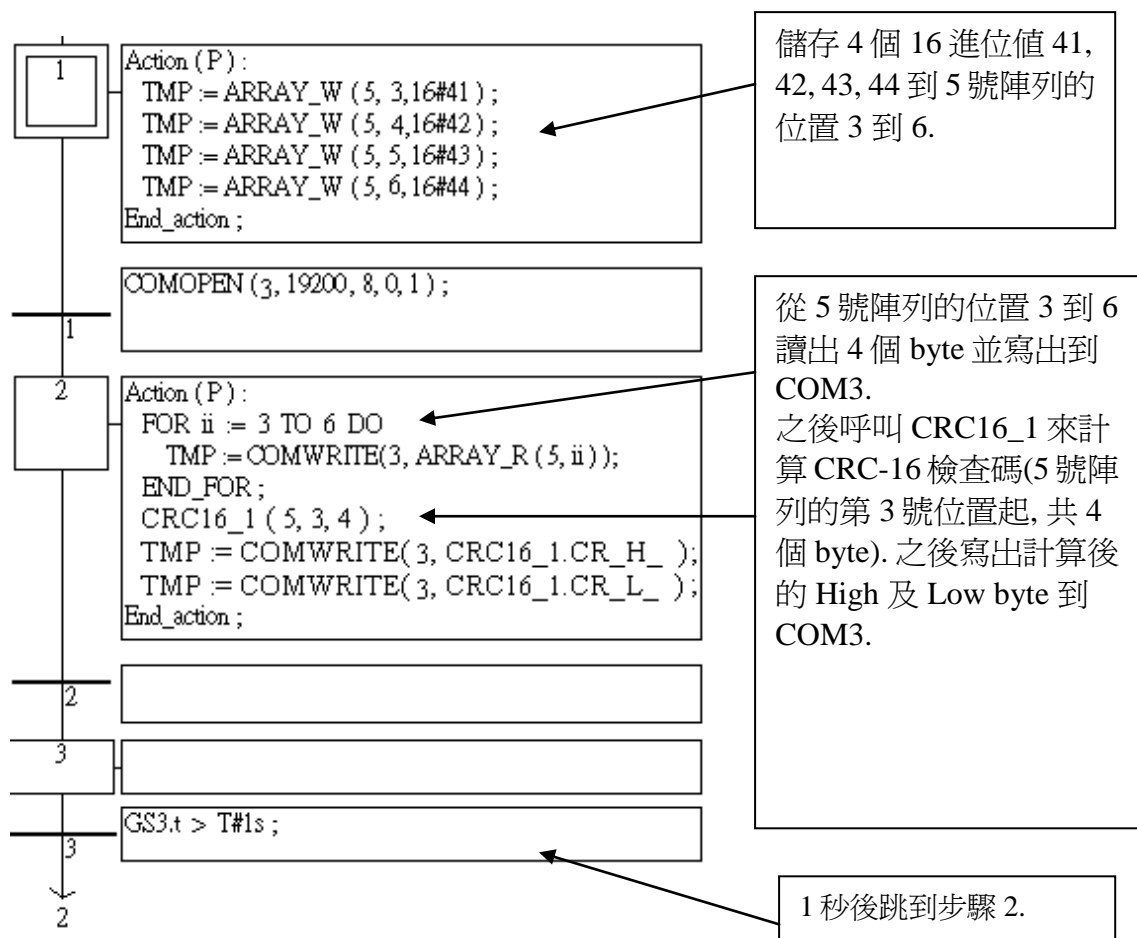
NUM_	Integer	要計算的 byte 陣列編號, 範圍為 I-8xx7 & I-7188EG/XG: 1 到 24; Wincon-8xx7: 1 到 48;
ADR_	Integer	從陣列內的那個位置起開始計算(1~256)
SIZE_	Integer	共有幾個 byte 要計算

傳回值 :

CR_H_	Integer	計算結果, high byte
CR_L_	Integer	計算結果, low byte

範例:

TMP 宣告為 internal boolean. ii, CR_H_ 及 CR_L_ 為 internal integer, CRC16_1 宣告為 FB instance 其型態為 CRC_16.



DI_CNT

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

讀取 Slot 0 的 DI Counter 值. Wincon 則為 Slot 1, 請參閱第 3.8 節

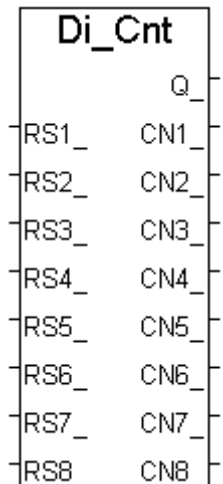
輸入參數:

RS1_ ~ RS8_ Boolean 當由 FALSE 上升到 TRUE 時重置該點的 counter 值為 0.

傳回值:

Q_ Boolean 正確: TRUE. 如果 Q_ 傳回 FALSE, 表示 "slot 0 找不到並列輸入點 "

CN1_ ~ CN8_ Integer 第 1 到 8 點的 counter 值. 範圍介於 0 到 2,147,483,647. 假如值超過 2,147,483,647, 會從 0 開始.



注意:

1. 只有插在 slot 0 的並列輸入(parallel D/I)卡可使用 "Di_Cnt", 其它 slot 不行.
2. WinCon W-8xx7 必需插在 slot 1, 其它 slot 不行.
3. 只有前 8 個 D/I 點可使用 "Di_Cnt".
4. I-7188EG/XG 必須在 slot 0 連結 Xxxx 卡, "Di_Cnt" 才能使用.
5. 每台 I-8xx7 & I-7188EG/XG 控制器最多只能使用 8 個並列 D/I counter. Counter 的輸入頻率最高為 500Hz. 最小脈波長度需大於 1 ms.
6. 每台 W-8xx7 控制器最多只能使用 8 個並列 D/I counter. Counter 的輸入頻率最高為 250Hz. 最小脈波長度需大於 2 ms.

範例: W-8xx7: Wdemo_22 , I-8xx7: demo_63

DT2MESAG

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

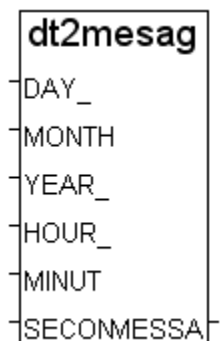
轉換 Date 和 Time 成爲 Message 資料型態

輸入參數:

DAY_ : Integer 日期 (1~31)
MONTH_ Integer 月份 (1~12)
YEAR_ : Integer 年, 例如. 01,02
HOUR_ : Integer 時 (0~23)
MINUTE_ : Integer 分 (0~59)
SECOND_ : Integer 秒 (0~59)

傳回值:

MESSAGE_: Message "日.月.年 時:分:秒", 例如: "20.01.07 11:05:40"



注意:

若輸入的參數不正確, 傳回的 Message 爲 '' (空訊息 NULL).

EBUS_B_R

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

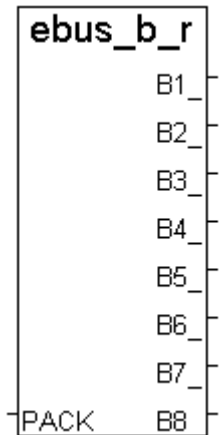
從 Ebus 上讀取 boolean 封包

輸入參數:

PACK_ Integer 要讀哪個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

傳回值 :

B1_ ~ B8_ Boolean 讀到該封包內的 8 個 boolean 值



注意: 請參考第 7.5 節

EBUS_B_W

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

寫出 boolean 封包到 Ebus 上

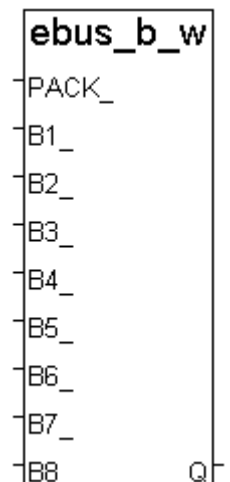
輸入參數:

PACK_ Integer 寫出哪個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

B1_ ~ B8_ Boolean 要寫出的 8 個 boolean 值

傳回值 :

Q Boolean 永遠回傳 TRUE.



注意: 請參考第 7.5 節

EBUS_F_R

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

從 Ebus 上讀取 Real 封包

輸入參數:

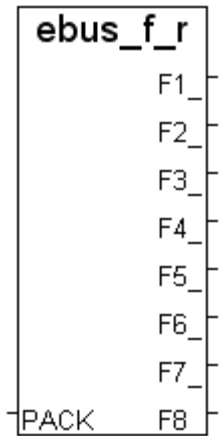
PACK_NO_ Integer 要讀哪個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

傳回值 :

F1_ ~ F8_ Real 讀到該封包內的 8 個 Real 值
如有錯誤發生, 傳回 1.23E-20

注意:

1. "EBUS_F_R" 與 "EBUS_N_R" 使用相同記憶體
2. Integer 和 REAL 不要使用同一個封包號碼. 否則資料會錯誤, 有時並會產生錯誤: "ERROR 115: EBUS_F_R float error"



EBUS_F_W

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

寫出 Real 封包到 Ebus 上

輸入參數:

PACK_ Integer 寫出哪個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

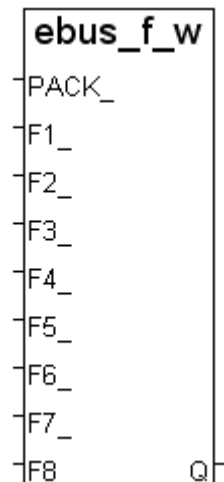
N1_ ~ N8_ Boolean 要寫出的 8 個 Real 值

傳回值 :

Q Boolean 永遠回傳 TRUE

注意:

1. "EBUS_F_W" 與 "EBUS_N_W" 使用相同記憶體區
2. Integer 和 REAL 不要使用同一個封包號碼. 否則資料會錯誤, 有時並會產生錯誤: "ERROR 115: EBUS_F_R float error"
3. 從以下版本起的驅動程式才有支援 EBUS_F_R 與 EBUS_F_W:
 - I-7188EG: 2.17 版起
 - I-7188XG: 2.15 版起
 - I-8XX7 : 3.19 版起
 - W-8XX7 : 建議更新至 3.36 版或更高版



EBUS_N_R

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

從 Ebus 上讀取 integer 封包

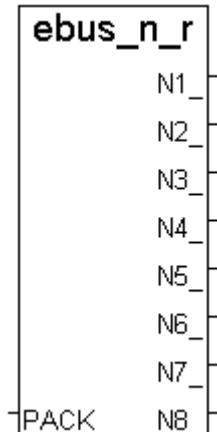
輸入參數:

PACK_ Integer 要讀哪個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

傳回值:

N1_ ~ N8_ Integer 讀到該封包內的 8 個 integer 值

注意: 請參考第 7.5 節



EBUS_N_W

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

寫出 integer 封包到 Ebus 上

輸入參數:

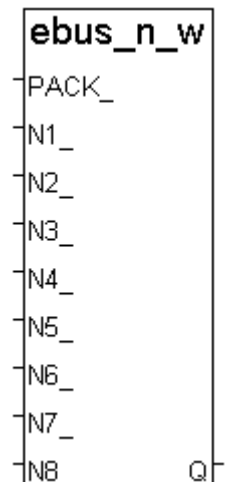
PACK_ Integer 寫出哪個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

N1_ ~ N8_ Boolean 要寫出的 8 個 integer 值

傳回值:

Q Boolean 永遠回傳 TRUE

注意: 請參考第 7.5 節



EBUS_STS

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

取得 Ebus 封包的傳送狀態

輸入參數:

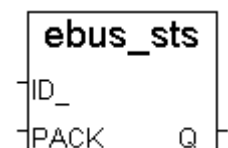
ID_ Integer 取得什麼? (0 ~ 1), 0: Boolean 封包, 1: Integer 封包

PACK_ Integer 取得那個編號的封包,
I-8xx7, I-7188EG/XG : 1 ~ 128
W-8xx7 : 1~256

傳回值:

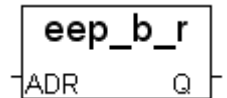
Q_ Boolean TRUE: 該封包通訊正常, FALSE: 該封包通訊異常
封包通訊異常的原因可能是, Ebus_m 控制器沒有啓動該封包
編號, 通訊線中斷, 發送該封包的工控器死機了, Ebus_m 控制 器

死機了, 或其它



EEP_B_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 EEPROM 讀出 1 個 boolean

輸入參數 :

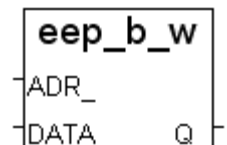
ADR_	Integer	讀哪個位置 I-8xx7 & I-7188EG/XG: (1 ~ 256) , WP-8xx7: (1 ~ 1024)
-------------	---------	--

傳回值 :

Q_	Boolean	讀到的 boolean 值
-----------	---------	---------------

EEP_B_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫 1 個 boolean 值到 EEPROM

輸入參數 :

ADRES_	Integer	寫到哪個位置 I-8xx7 & I-7188EG/XG: (1 ~ 256); WP-8xx7: (1 ~ 1024).
---------------	---------	---

DATA_	Boolean	要寫的 boolean 值
--------------	---------	---------------

傳回值 :

Q_	Boolean	正確回傳 TRUE.
-----------	---------	------------

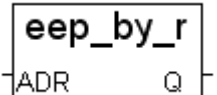
注意:

- * 讀 EEPROM 沒有次數限制
- * 寫 EEPROM 有次數限制 (請參考 10.2 節)
- * 需用 EEP_EN 開啓 EEPROM, 才寫的進去
- * 讀/寫 EEPROM 很耗 CPU 時間, 會造成 Scan Time 時間大幅增加, 請小心使用.

範例: 請參考 demo_17

EEP_BY_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 EEPROM 讀出 1 個 byte (unsigned 8-bit)

輸入參數 :

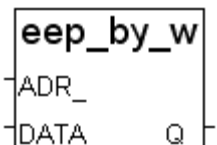
ADR_ Integer 讀哪個位置
I-8xx7 & I-7188EG/XG: (1 ~ 1512), WP-8xx7: (1 ~ 6144)

傳回值 :

Q_ Integer 讀到的 byte 值 (0~255)

EEP_BY_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫 1 個 byte (unsigned 8-bit) 值到 EEPROM

輸入參數 :

ADR_ Integer 寫到哪個位置
I-8xx7 & I-7188EG/XG: (1 ~ 1512), WP-8xx7: (1 ~ 6144)

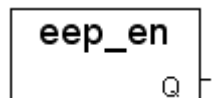
DATA_ Integer 要寫的 byte 值 (0 ~ 255)

傳回值 :

Q_ Boolean 正確回傳 TRUE.

EEP_EN

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

開啟 EEPROM 以便可以寫入

傳回值 :

Q_ Boolean 成功回傳 TRUE.

注意 :

* EEP_BY_R, EEP_BY_W, EEP_WD_R, EEP_WD_W, EEP_N_R, EEP_N_W, EEP_F_R 與 EEP_F_W 等 functions 使用相同的記憶體區, 請小心配置使用. 下面的位址編號使用相同的記憶體位址。

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... *)
Word	2n-1, 2n	
Integer, Real	n	

* 寫 EEPROM 有次數限制 (請參考 10.2 節)

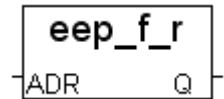
* 需用 EEP_EN 開啟 EEPROM, 才寫的進去

* 讀/寫 EEPROM 很耗 CPU 時間, 會造成 Scan Time 時間大幅增加, 請小心使用.

範例: 請參考第 11 章 demo_17

EEP_F_R

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 EEPROM 讀取 1 個 Real 值

輸入參數 :

ADR_ Integer 讀哪個位置 (與 EEP_N_R 使用相同 EEPROM 位址)
I-8xx7 & I-7188EG/XG: (1 ~ 378), WP-8xx7: (1 ~ 1536)

傳回值 :

Q_ Real 讀到的 Real 值. 若 ADR_ 超出有效範圍, 則 Q_ = 1.23E-20.
若儲存在 EEPROM 的不是 REAL, 則 Q_ 的值會是錯的,
有時並會發生錯誤: "ERROR 114: EEP_F_R float error".

重要 :

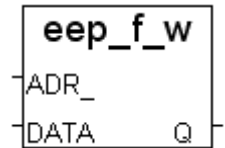
1. "EEP_F_R" 與 "EEP_N_R" 使用相同的 EEPROM 位址
2. Integer 和 REAL **不要**讀/寫同一個 EEPROM 位址. 否則有可能會產生錯誤: "ERROR 114: EEP_F_R float error".
3. 每次讀/寫 EEPROM 皆耗費許多 CPU 時間, 尤其是寫的動作, 請小心使用.

注意 :

1. 這個函式即使沒有呼叫 EEP_EN 也可以使用.
2. EEP_BY_R, EEP_BY_W, EEP_WD_R, EEP_WD_W, EEP_N_R, EEP_N_W, EEP_F_R 與 EEP_F_W 等 functions 使用相同的記憶區, 請小心配置使用.
例如: EEP_N_R 的 ADR_2 使用 4 bytes, 和 EEP_WD_R 的 ADR_3, 4 及 EEP_BY_R 的 ADR_5, 6, 7, 8 使用相同的記憶區.
3. WP-8xx7/8xx6 使用 EEPROM 的 16 - 31 區來儲存 Boolean (每區有 64 bytes), 第 32 - 254 區用來儲存 byte, word 和 long. 第 0 - 15 則沒有使用. 第 255 區保留.
4. 從以下版本起的驅動程式才有支援 EEP_F_R 與 EEP_F_W:
 - I-7188EG: 2.17 版起
 - I-7188XG: 2.15 版起
 - I-8XX7 : 3.19 版起
 - WP-8XX7 : 建議更新至 3.36 版或更高版

EEP_F_W

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫 1 個 Real 值到 EEPROM

輸入參數 :

ADR_	Integer	寫到哪個位置 (與 EEP_N_W 使用相同 EEPROM 位址) I-8xx7 & I-7188EG/XG: (1 ~ 378), WP-8xx7: (1 ~ 1536)
DATA_ :	Real	要寫入的 REAL 值

傳回值 :

Q_	Boolean	TRUE : 成功, FALSE: 失敗.
-----------	---------	-----------------------

重要 :

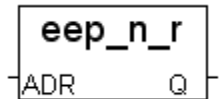
1. "EEP_F_W" 與 "EEP_N_W" 使用相同的 EEPROM 位址
2. Integer 和 REAL **不要**讀/寫同一個 EEPROM 位址. 否則有可能會產生錯誤: "ERROR 114: EEP_F_R float error".
3. 每次讀/寫 EEPROM 皆耗費許多 CPU 時間, 尤其是寫的動作, 請小心使用.

注意 :

1. 需使用 EEP_EN 開啓 EEPROM, 才寫的進去.
2. EEP_BY_R, EEP_BY_W, EEP_WD_R, EEP_WD_W, EEP_N_R, EEP_N_W, EEP_F_R 與 EEP_F_W 等 functions 使用相同的記憶區, 請小心配置使用.
例如: EEP_N_R 的 ADR_2 使用 4 bytes, 和 EEP_WD_R 的 ADR_3, 4 及 EEP_BY_R 的 ADR_5, 6, 7, 8 使用相同的記憶區.
3. 小心使用此函式! 當寫入超過 100,000 次, EEPROM 將會損毀.
4. 寫入之前須先呼叫 EEP_EN() 來解除 EEPROM 的寫入保護, 如此之後才能寫入.
5. 寫入之後, 需呼叫 EEP_PR() 來保護 EEPROM
6. WP-8xx7/8xx6 使用 EEPROM 的 16 - 31 區來儲存 Boolean (每區有 64 bytes), 第 32 - 254 區用來儲存 byte, word 和 long. 第 0 - 15 則沒有使用. 第 255 區保留.
7. 從以下版本起的驅動程式才有支援 EEP_F_R 與 EEP_F_W:
 - I-7188EG: 2.17 版起
 - I-7188XG: 2.15 版起
 - I-8XX7 : 3.19 版起
 - WP-8XX7 : 建議更新至 3.36 版或更高版

EEP_N_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 EEPROM 讀出 1 個 長整數 (signed 32-bit)

輸入參數 :

ADR_ Integer 讀哪個位置
I-8xx7 & I-7188EG/XG: (1 ~ 378) , WP-8xx7: (1 ~ 1536)

傳回值 :

Q_ Integer 讀到的長整數值

* EEP_BY_R, EEP_BY_W, EEP_WD_R, EEP_WD_W, EEP_N_R, EEP_N_W, EEP_F_R 與 EEP_F_W 等 functions 使用相同的記憶區, 請小心配置使用. 下面的位址編號使用相同的記憶位址。

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... *)
Word	2n-1, 2n	
Integer, Real	n	

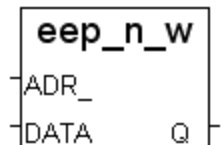
* 讀 EEPROM 沒有次數限制, 不管有無使用 EEP_EN 開啓 EEPROM, 都可讀

* 寫 EEPROM 有次數限制 (請參考 10.2 節)

範例: 請參考第 11 章 demo_17

EEP_N_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫 1 個 長整數(signed 32-bit) 值到 EEPROM

輸入參數 :

ADR_ Integer 寫到哪個位置
I-8xx7 & I-7188EG/XG: (1 ~ 378) , WP-8xx7: (1 ~ 1536)

DATA_ Integer 要寫的長整數值

傳回值 :

Q_ Boolean 正確回傳 TRUE.

* EEP_BY_R, EEP_BY_W, EEP_WD_R, EEP_WD_W, EEP_N_R, EEP_N_W, EEP_F_R 與 EEP_F_W 等 functions 使用相同的記憶區, 請小心配置使用. 下面的位址編號使用相同的記憶位址。

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... *)
Word	2n-1, 2n	
Integer, Real	n	

* 寫 EEPROM 有次數限制 (請參考 10.2 節)

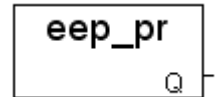
* 需用 EEP_EN 開啓 EEPROM, 才寫的進去

* 讀/寫 EEPROM 很耗 CPU 時間, 會造成 Scan Time 時間大幅增加, 請小心使用.

範例: 請參考第 11 章 demo_17

EEP_PR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

關閉 EEPROM 以防止寫入

傳回值：

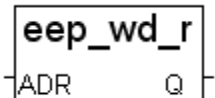
Q_ Boolean 成功回傳 TRUE

- * 寫 EEPROM 有次數限制 (請參考 10.2 節)
- * 需用 EEP_EN 開啓 EEPROM, 才寫的進去

範例： 請參考第 11 章 demo_17

EEP_WD_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 EEPROM 讀出 1 個 word (signed 16-bit)

輸入參數 :

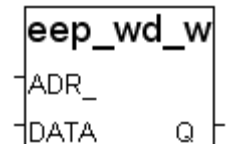
ADR_	Integer	讀哪個位置 I-8xx7 & I-7188EG/XG: (1 ~ 756) , WP-8xx7: (1 ~ 3072)
-------------	---------	--

傳回值 :

Q_	Integer	讀到的 word 值 (-32768 ~ +32767)
-----------	---------	------------------------------

EEP_WD_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫 1 個 word (signed 16-bit) 值到 EEPROM

輸入參數 :

ADR_	Integer	寫到哪個位置 I-8xx7 & I-7188EG/XG: (1 ~ 756) , WP-8xx7: (1 ~ 3072)
DATA_	Integer	要寫的 word 值 (-32768 ~ 32767)

傳回值 :

Q_	Boolean	正確回傳 TRUE.
-----------	---------	------------

* EEP_BY_R, EEP_BY_W, EEP_WD_R, EEP_WD_W, EEP_N_R, EEP_N_W, EEP_F_R 與 EEP_F_W 等 functions 使用相同的記憶區, 請小心配置使用. 下面的位址編號使用相同的記憶位址。

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... *)
Word	2n-1, 2n	
Integer	n	

* 寫 EEPROM 有次數限制 (請參考 10.2 節)

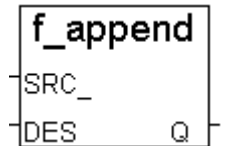
* 需用 EEP_EN 開啓 EEPROM, 才寫的進去

* 讀/寫 EEPROM 很耗 CPU 時間, 會造成 Scan Time 時間大幅增加, 請小心使用.

範例: 請參考第 11 章 demo_17

F_APPEND

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

將 1 個檔案之內容 附加到 1 個檔案 的後方

輸入參數 :

SRC_	Message	來源檔名. 例如 '\System_Disk\data.txt'
DES_	Message	目地檔名. 例如 '\System_Disk\data1.txt'

傳回值 :

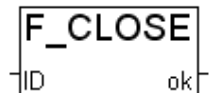
Q_	Boolean	True: Ok, False: 失敗
-----------	---------	---------------------

注意:

1. 如果有任一個檔案不存在, 回傳 False.
2. 來源檔 與 目地檔的狀態 必需是 Close 的. 未 Open
3. 可參考其它 ISaGRAF 的標準函數 – F_wopen, F_ropen, F_close, F_eof, Fa_read, Fa_write
4. 可參考 ICP DAS 加入的函數 – F_creat, F_copy, F_append, F_dir, F_end, F_seek, F_writ_b, F_writ_f, F_writ_s, F_writ_w
5. 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

F_CLOSE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

關閉以 F_ROPEN, F_WOPEN 或 F_CREAT 開啓的二進位檔案
ISaGRAF 的 simulator 不包含此函式, 無法模擬(仿真)

輸入參數 :

ID	Integer	F_ROPEN, F_WOPEN 或 F_CREAT 傳回的檔案編號: .
-----------	---------	---------------------------------------

傳回值 :

ok	Boolean	狀態回傳 TRUE :檔案關閉 ok; FALSE : 失敗
-----------	---------	-----------------------------------

範例 :

(* ST 程式: *)

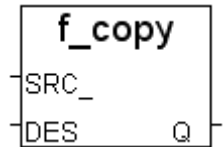
```
file_id := F_ROPEN('\System_Disk\data.bin');
ok := F_CLOSE(file_id);
```

(* IL 相等式: *)

```
LD '\System_Disk\data.bin'
F_ROPEN
ST file_id
F_CLOSE (* file_id 已經在 IL 程式結果中 *)
ST ok
```


F_COPY

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

複製 1 個檔案

輸入參數：

SRC_	Message	來源檔名. 例如 '\System_Disk\data.txt'
DES_	Message	目的地檔名. 例如 '\System_Disk\data1.txt'

傳回值：

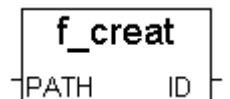
Q_	Boolean	True: Ok, False: 失敗
-----------	---------	---------------------

注意：

1. 複製 來源檔 到目的地檔.
2. 來源檔 與 目的地檔 的狀態 必需是 Close 的. 未 Open
3. 如果目的地檔已經存在, 會被整個替換掉.

F_CREAT

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

建立 1 個新的 空檔案 以便日後 讀 或 寫

輸入參數：

Path_	Message	檔名. 例如 '\System_Disk\data.txt'
--------------	---------	--------------------------------

傳回值：

ID_	Integer	檔案代碼, 若為 0, 表示 建立檔案 失敗
------------	---------	------------------------

注意：

1. 如果檔案已經存在, 呼叫此函數會 清空 原先檔案內的資料.
2. 若要 讀取 已經存在的檔案, 請用 ISaGRAF 的標準函數 – “F_ROPEN()”
3. 若要 寫入 已經存在的檔案, 請用 ISaGRAF 的標準函數 – “F_WOPEN()”
4. 可參考其它 ISaGRAF 的標準函數 – F_wopen, F_ropen, F_close, F_eof, Fa_read, Fa_write
5. 可參考 ICP DAS 加入的函數 – F_creat, F_copy, F_append, F_dir, F_end, F_seek, F_writ_b, F_writ_f, F_writ_s, F_writ_w
6. 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內 則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

範例： WP-8xx7 CD-ROM : \napdos\isagraf\WP-8xx7\demo\ 內的 Wpdmo_11 & Wpdmo_12

F_DELETE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

刪除 1 個檔案

輸入參數 :

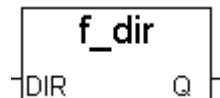
Name_ Message 檔名. 例如 '\System_Disk\data.txt'

傳回值 :

Q_ Boolean True: Ok, False: 失敗

F_DIR

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

建立一個新的目錄 (Directory)

輸入參數 :

Dir_ Message Directory 的名稱. 例如 '\DATA21'

傳回值 :

Q_ Boolean True: Ok, False: 失敗 (例如 Directory 已存在)

範例:

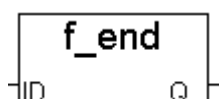
(* INIT 宣告為 Internal Boolean, 初值為 True*)

(* TMP 宣告為 Internal Boolean*)

```
if INIT then
  INIT := False ;
  TMP := f_dir('\DATA21') ;
End_if ;
```

F_END

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

移動檔案目前位置 到檔案的結尾

輸入參數 :

ID_ Integer 檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)

傳回值 :

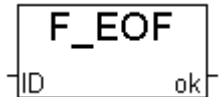
Q_ Boolean True: Ok, False: 失敗

注意:

1. 請參考 F_seek 來移動檔案目前位置 到一個指定的位置
2. 可參考其它 ISaGRAF 的標準函數 – F_wopen, F_ropen, F_close, F_eof, Fa_read, Fa_write
3. 可參考 ICP DAS 加入的函數 – F_creat, F_copy, F_append, F_dir, F_end, F_seek, F_writ_b, F_writ_f, F_writ_s, F_writ_w
4. 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

F_EOF

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

測試是否 檔案目前位置 已抵達 檔案的結尾

輸入參數 :

ID_ Integer 檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)

傳回值 :

Ok Boolean True: 已抵達 檔案的結尾, False: 還未抵達

F_READ_B

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從檔案內的目前位置讀出 1 個 byte 值

輸入參數 :

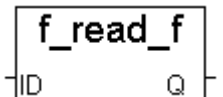
ID_ Integer 檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)

傳回值 :

Q_ Integer 讀到的 byte 值 (0 ~ 255)

F_READ_F

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從檔案內的目前位置讀出 1 個 實數值 (32-bit float)

輸入參數 :

ID_ Integer 檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)

傳回值 :

Q_ Real 讀到的 實數值 (32-bit float), 若該位置存的不是實數值, 會傳回錯誤的值, 有時還會發生 "ERROR 117:F_READ_error".

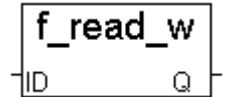
注意:

1. 請使用 ISaGRAF 標準函式 –“FA_READ” & “FA_WRITE” 來讀/寫 長整數 (signed 32-bit)
2. 請使用 ISaGRAF 標準函式 –“FM_READ” & “FM_WRITE” 或 “F_writ_s” 來讀/寫 字串 (string)
3. 可參考其它 ISaGRAF 的標準函數 – F_wopen, F_ropen, F_close, F_eof, Fa_read, Fa_write
4. 可參考 ICP DAS 加入的函數 – F_creat, F_copy, F_append, F_dir, F_end, F_seek, F_writ_b, F_writ_f, F_writ_s, F_writ_w
5. 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失。

範例: 請參考 WP-8xx7 CD:\napdos\isagraf\wp-8xx7\demo\ “wpdmo_01” & “wpdmo_02”

F_READ_W

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從檔案內的目前位置讀出 1 個 Word 值 (signed 16-bit)

輸入參數 :

ID_ Integer 檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)

傳回值 :

Q_ Integer 讀到的 Word 值 (-32768 ~ +32767)

F_ROPEN

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

以 Read 模式開啓一個二進位檔案. 之後該檔案只能做讀的動作.

輸入參數 :

Path_ Message 檔案名稱
可包含執行路徑, 使用 \ 或 / (兩者相同) 符號指定路徑.

傳回值 :

ID_ Integer 檔案號碼
0: 有錯誤, 檔案不存在.

範例 :

```
(* ST 程式: *)  
file_id := F_ROPEN('\System_Disk\ISaGRAF\data.bin ');  
error := (file_id=0);
```

```
(* IL 相等式: *)  
LD '\System_Disk\ISaGRAF\data.bin'  
F_ROPEN  
ST file_id  
EQ 0  
ST error
```

注意: 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

F_SEEK

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

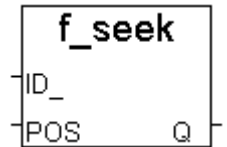
移動檔案內的 目前位置 到 指定位置...

輸入參數 :

ID_ Integer 檔案代碼, (使用 F_ROPEN , F_WOPEN 或 F_CREAT 的回傳值)
POS_ Integer 移到那個位置, 單位為 byte (1 ~ ...)

傳回值 :

Q_ Boolean True: 成功. False: 失敗



F_TRIG

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : Standard_Function

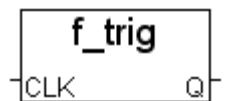
偵測 boolean 變數的 falling edge (下降邊緣)...

輸入參數 :

CLK_ Boolean 任何布林變數

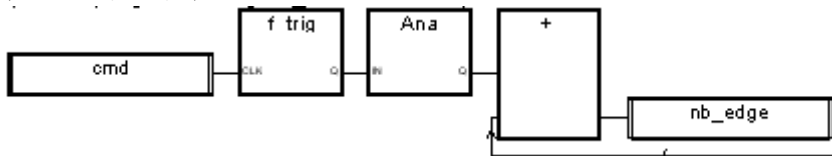
傳回值 :

Q_ Boolean TRUE : 若 CLK 由 TRUE 變為 FALSE
FALSE : 其他狀態



範例 :

(* FBD 程式 *)



(* ST 相等式: 假設 F_TRIG1 為 F_TRIG 函式方塊 *)

```
F_TRIG1(cmd);
```

```
nb_edge := ANA(F_TRIG1.Q) + nb_edge;
```

(* IL 相等式: *)

```
LD cmd
```

```
ST F_TRIG1.clk
```

```
CAL F_TRIG1
```

```
LD F_TRIG1.Q
```

```
ANA
```

```
ADD nb_edge
```

```
ST nb_edge
```

F_WOPEN

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

F_WOPEN	
Path	ID

型態 : Standard_Function

以 Write 模式開啓一個二進位檔案. 之後該檔案可作讀/寫動作.
ISaGRAF 的 simulator 不包含此函式

輸入參數 :

Path	Message	檔案名稱 可包含執行路徑, 使用 \ 或 / (兩者相同) 符號指定路徑.
-------------	---------	--

傳回值 :

ID	Integer	檔案號碼 0 : 表示有錯誤. 若檔案已存在, 將被覆蓋.
-----------	---------	----------------------------------

範例 :

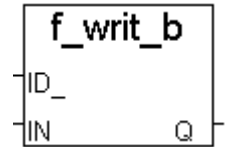
```
(* ST 程式: *)  
file_id := F_WOPEN('\System_Disk\hello.dat');  
error := (file_id=0);
```

```
(* IL 相等式: *)  
LD '\System_Disk\hello.dat'  
F_WOPEN  
ST file_id  
EQ 0  
ST error
```

注意: 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

F_WRIT_B

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 1 個 Byte 值到檔案內...

輸入參數 :

ID_ Integer 檔案代碼, (使用 F_ROPEN , F_WOPEN 或 F_CREAT 那個回傳值)
IN_ Integer 要寫入的 Byte 值 (0 ~ 255), 如果 > 255 或 < 0, 最低的 byte 值被寫入

傳回值 :

Q_ Boolean True: 成功. False: 失敗

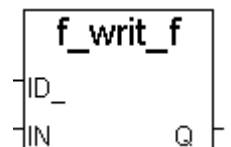
注意:

1. 請使用 ISaGRAF 標準函式 –“FA_READ” & “FA_WRITE” 來讀/寫 長整數 (signed 32-bit)
2. 請使用 ISaGRAF 標準函式 –“FM_READ” & “FM_WRITE” 或 “F_writ_s” 來讀/寫 字串 (string)
3. 可參考其它 ISaGRAF 的標準函數 – F_wopen, F_ropen, F_close, F_eof, Fa_read, Fa_write
4. 可參考 ICP DAS 加入的函數 – F_creat, F_copy, F_append, F_dir, F_end, F_seek, F_writ_b, F_writ_f, F_writ_s, F_writ_w
5. 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

範例: 請參考 WP-8xx7 CD:\napdos\isagraf\wp-8xx7\demo\ “wpdmo_01” & “wpdmo_02”

F_WRIT_F

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 1 個 實數值 (32-bit float) 到檔案內...

輸入參數 :

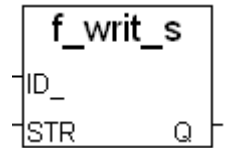
ID_ Integer 檔案代碼, (使用 F_ROPEN , F_WOPEN 或 F_CREAT 的回傳值)
IN_ Real 要寫入的 實數值

傳回值 :

Q_ Boolean True: 成功. False: 失敗

F_WRIT_S

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 1 個 字串 到檔案內 (字串結尾不含 <CR> <LF>)

輸入參數 :

ID_	Integer	檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)
STR_	Message	要寫入的字串

傳回值 :

Q_	Boolean	True: 成功. False: 失敗
-----------	---------	---------------------

注意:

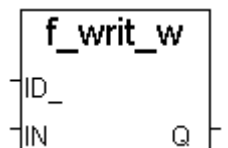
1. F_writ_s 不會在 字串結尾加入 <CR> <LF>, FM_write 則會在 字串結尾加入 <CR> <LF>
2. 請使用 ISaGRAF 標準函式 –“FA_READ” & “FA_WRITE” 來 讀/寫 長整數 (signed 32-bit)
3. 請使用 ISaGRAF 標準函式 –“FM_READ” & “FM_WRITE” 或 “F_writ_s” 來 讀/寫 字串 (string)
4. 可參考其它 ISaGRAF 的標準函數 – F_wopen, F_ropen, F_close, F_eof, Fa_read, Fa_write
5. 可參考 ICP DAS 加入的函數 – F_creat, F_copy, F_append, F_dir, F_end, F_seek, F_writ_b, F_writ_f, F_writ_s, F_writ_w
6. 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內 則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

範例:

請參考 WP-8xx7 CD:\napdos\isagraf\wp-8xx7\demo\ “wpdmo_01” & “wpdmo_02”

F_WRIT_W

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 1 個 Word (signed 16-bit) 到檔案內...

輸入參數 :

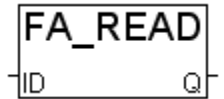
ID_	Integer	檔案代碼, (使用 F_ROPEN, F_WOPEN 或 F_CREAT 的回傳值)
IN_	Integer	要寫入的 Word 值 (-32768 ~ +32767)

傳回值 :

Q_	Boolean	True: 成功. False: 失敗
-----------	---------	---------------------

FA_READ

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

讀取檔案內的一個長整數 (32-bit signed)

ISaGRAF 的 simulator 不包含此函式

輸入參數 :

ID Integer F_ROPEN, F_WOPEN 或 F_CREAT 傳回的檔案編號.

傳回值 :

Q Integer 由檔案讀出的整數值

範例 :

(* ST 程式: *)

```
file_id := F_ROPEN('\System_Disk\data.dat');
```

```
vinc := FA_READ(file_id);
```

```
delta_tim := tmr(FA_READ(file_id));
```

```
ok := F_CLOSE(file_id);
```

(* IL 相等式: *)

```
LD '\System_Disk\data.dat'
```

```
F_ROPEN
```

```
ST file_id
```

```
LD file_id
```

```
FA_READ (* 讀取 vinc *)
```

```
ST vinc
```

```
LD file_id
```

```
FA_READ (* 讀取 timer: delta_tim *)
```

```
TMR (* 轉換格式為 timer *)
```

```
ST delta_tim
```

```
LD file_id
```

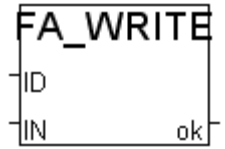
```
F_CLOSE
```

```
ST ok
```

注意: 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

FA_WRITE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

寫一個長整數(32-bit signed) 到檔案內
ISaGRAF 的 simulator 不包含此函式

輸入參數 :

ID	Integer	F_WOPEN, F_WOPEN 或 F_CREAT 傳回的檔案編號.
IN	Integer	要寫入檔案的整數值

傳回值 :

OK	Boolean	執行狀態: TRUE 表示 ok
-----------	---------	------------------

範例 :

```
(* ST 程式: *)
file_id := F_WOPEN('\System_Disk\data.dat');
nb_written := 0;
nb_written := nb_written + ana(FA_WRITE(file_id,vstart));
ok := F_CLOSE(file_id);
IF ( nb_written <> 4) THEN
    ERROR := ERR_FILE;
END_IF;
```

(* IL 相等式: 請參考 ISaGRAF Projects 的 Help/ language Reference*)

注意: 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

FBUS_B_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

從 Fbus 上讀取 boolean 封包

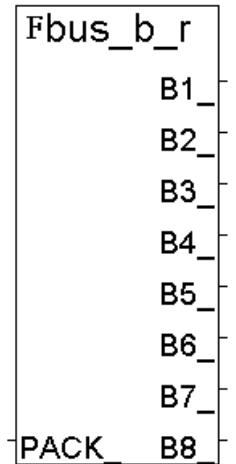
輸入參數：

PACK_ Integer 封包編號, 1 ~ 128

傳回值：

B1_ ~ B8_ Boolean 讀到的封包內的 8 個 boolean 值

範例： 請參考第 7 章或 demo_11a & demo_11b



FBUS_B_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

寫出 Boolean 封包到 Fbus 上

輸入參數：

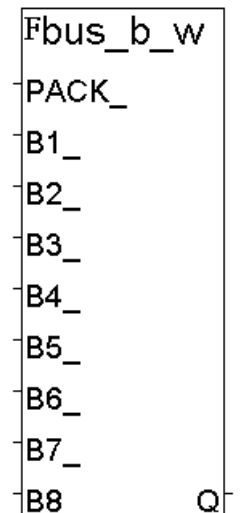
PACK_ Integer 封包編號, 1 ~ 128

B1_ ~ B8_ Boolean 要寫出的 8 個 boolean 值

傳回值：

Q_ Boolean 只回傳 TRUE.

範例： 請參考第 7 章或 demo_11a & demo_11b



FBUS_F_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

從 Fbus 上讀取 Real 封包

輸入參數：

PACK_NO_ Integer 封包編號, 1 ~ 128

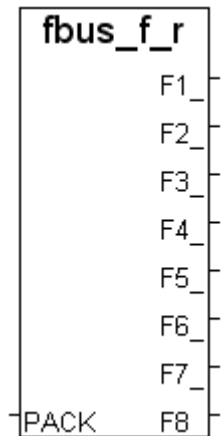
傳回值：

F1_ ~ F8_ Real 讀到的封包內的 8 個 Real 值
錯誤則傳回 1.23E-20

重要：

1. "FBUS_F_R" 與 "FBUS_N_R" 使用相同的記憶體區。
2. Integer 和 Real 資料型態請**不要**使用同一個封包號碼。否則資料會錯誤, 有時並會產生:
"ERROR 116: FBUS_F_R float error"
3. 請使用 "FBUS_N_R" 和 "FBUS_N_W" 來傳送 Integer 值

範例: 請參考第 7 章或 demo_11a & demo_11b



FBUS_F_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

寫出 Real 封包到 Fbus 上

輸入參數：

PACK_NO_ Integer 封包編號, 1 ~ 128

F1_ ~ F8_ Real 要寫出的 8 個 Real 值

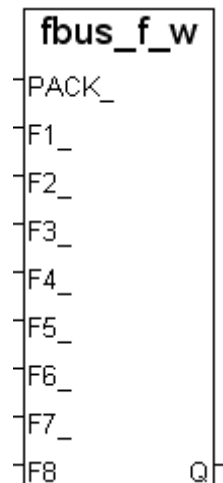
傳回值：

Q_ Boolean 只回傳 TRUE.

重要：

1. "FBUS_F_W" 與 "FBUS_N_W" 使用相同的記憶體區。
2. Integer 和 Real 資料型態請**不要**使用同一個封包號碼。否則資料會錯誤, 有時並會產生:
"ERROR 116: FBUS_F_R float error"
3. 請使用 "FBUS_N_R" 和 "FBUS_N_W" 來傳送 Integer 值
4. 從以下版本起的驅動程式才有支援 FBUS_F_R 與 FBUS_F_W:
 - I-7188EG: 2.17 版起
 - I-7188XG: 2.15 版起
 - I-8XX7 : 3.19 版起
 - WP-8XX7 : 建議更新至 3.36 版或更高版

範例: 請參考第 7 章或 demo_11a & demo_11b



FBUS_N_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

從 Fbus 上讀取 Integer 封包

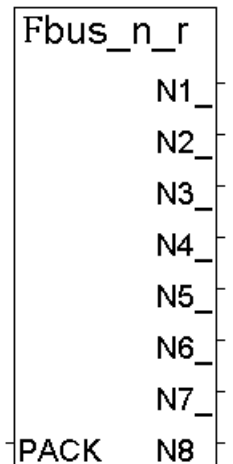
輸入參數 :

PACK_ Integer 封包編號, 1 ~ 128

傳回值 :

N1_ ~ N8_ Integer 讀到的封包內的 8 個 integer 值

範例: 請參考第 7 章或 demo_11a & demo_11b



FBUS_N_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

寫出 Integer 封包到 Fbus 上

輸入參數 :

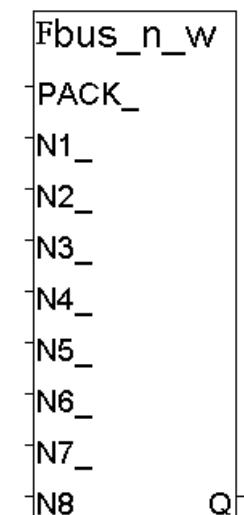
PACK_ Integer 封包編號, 1 ~ 128

N1_ ~ N8_ Boolean 要寫出的 8 個 Boolean 值

傳回值 :

Q_ Boolean 只回傳 TRUE.

範例: 請參考第 7 章或 demo_11a & demo_11b



FBUS_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

取得 Fbus 封包的通訊狀態

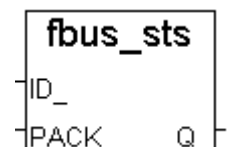
輸入參數 :

ID_ Integer 取得什麼? (0 ~ 1), 0: Boolean 封包, 1: Integer 封包

PACK_ Integer 取得那個編號的封包. 1 ~ 128

傳回值 :

Q_ Boolean TRUE: 該封包通訊正常, FALSE: 該封包通訊異常
封包通訊異常的原因可能是, Fbus_m 控制器沒有啟動該封包編號, 通訊線中斷, 發送該封包的控制器死機了, Fbus_m 控制器死機了, 或其它 ...



FM_READ

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

從二進位檔讀取 MESSAGE 變數.

ISaGRAF 的 simulator 不包含此函式

輸入參數 :

ID Integer F_ROPEN, F_WOPEN 或 F_CREAT 傳回的檔案編號.

傳回值 :

Q Message 讀出的 message 值

範例 :

(* ST 程式: *)

```
file_id := F_ROPEN('\System_Disk\m1.txt');
```

```
status1 := FM_READ(file_id);
```

```
ok := F_CLOSE(file_id);
```

(* IL 相等式: *)

```
LD '\System_Disk\m1.txt'
```

```
F_ROPEN
```

```
ST file_id
```

```
FM_READ
```

(* 讀取 status1 *)

```
ST status1
```

```
LD file_id
```

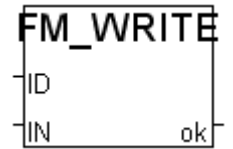
```
F_CLOSE
```

```
ST ok
```

注意: 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

FM_WRITE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

將 MESSAGE 寫入二進位檔.

ISaGRAF 的 simulator 不包含此函式

輸入參數 :

ID	Integer	F_WOPEN 傳回的檔案號碼.
IN	Message	要寫入檔案的訊息

傳回值 :

OK	Boolean	執行狀態 : TRUE 表示成功
-----------	---------	------------------

範例 :

(* ST 程式: *)

```
file_id := F_WOPEN('\System_Disk\m1.txt ');
ok := FM_WRITE(file_id,'First message');
ok := FM_WRITE(file_id,'Last message');
ok := F_CLOSE(file_id);
```

(* IL 相等式: *)

```
LD 'trace.txt'
F_WOPEN
ST file_id
FM_WRITE      'First message'      (*寫第一個訊息*)
ST ok
LD file_id
FM_WRITE      'Last message'      (*寫第二個訊息*)
ST ok
LD file_id
F_CLOSE
ST ok
```

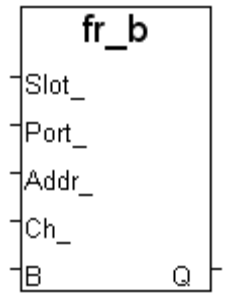
注意: 在 WP-8xx7 的 \System_Disk\ 路徑內操作 File 會消耗很多 CPU 時間, 若是在 RAM Disk 內則不會, 比如 \Temp\ 內, 但關機後, File 就會消失.

FR_B

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Boolean 變數對應到一個 FRNET I/O 點
請參考第 23 章來使用 FR_16Di 與 FR_16Do 與 FR_B_A



輸入參數 :

Slot_ :	Integer	所插入的 I-8172 的插槽編號 (1 - 7)
Port_ :	Integer	使用那個 I-8072 的埠號 (0 或 1)
Addr_ :	Integer	模組位址, DO (0 - 7), DI (8 - 15)
Ch_ :	Integer	使用的 Channel 編號 (1 - 16)
B_ :	Boolean	boolean 變數名稱

傳回值 :

Q_ :	Boolean	True: Ok. False: 參數錯誤.
------	---------	------------------------

注意 :

1. I-8172 請連接到 Wincon 的 slot 1 到 7 之中, 然後再以其 Port0 或 Port1 連接 FRNET I/O 模組
2. FR_B , FR_B_A 必須在第一次 PLC scan 時呼叫. 第二次以後的 PLC scan 則無法呼叫成功.
3. FRNET D/O 模組不支援通訊狀態偵測, 而 FRNET D/I 模組則支援通訊狀態偵測.
4. 每個 FRNET 輸出模組的 Dip switch 上有一個 'RESET' dip 或特殊 Jumper. 需先將 'RESET' dip 或 Jumper 設到 'ON' 的位置 (或 enable), 如此一來, 當 I-8172 與 FRNET D/O 模組通訊中斷時,便可重置輸出 channel 為 OFF 狀態.
例如: 將 FR-2057 的第 8 個 Dip 切到 'ON' 則可重新啓動.
5. 請參考下列網址 取得更多資訊:

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ048' 及
http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm

範例: 請參考網址 <http://www.icpdas.com/faq/isagraf.htm> 的 'FAQ048', 範例檔 Wdemo_39

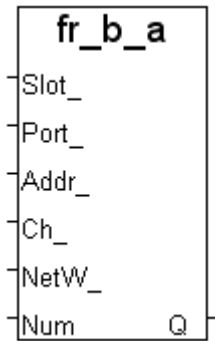
FR_B_A

□ I-8x17 ■ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Boolean 變數陣列對應到數個 FRNET I/O 點

(請參考 2.6 節關於"變數陣列"的說明, 與參考第 23 章來使用 FR_16Di 與 FR_16Do 與 FR_B_A)



輸入參數 :

Slot_ :	Integer	所插入的 I-8172 的插槽編號 (1 - 7)
Port_ :	Integer	使用那個 I-8072 的埠號 (0 或 1)
Addr_ :	Integer	起始模組位址, DO (0 - 7), DI (8 - 15)
Ch_ :	Integer	起始 Channel 編號 (1 - 16)
NetW_ :	Integer	對應 "變數陣列" 的網路位址編號 : 1~ 8191
Num_ :	Integer	使用 FRNet I/O 時, boolean "變數陣列" 的數量 : 1 - 255. 例如: Bi[0..15] 的大小為 16, 可設定 NUM_ 為 1 到 16. ABC[0..127] 的大小為 128, 可設定 NUM_ 為 1 到 128

傳回值 :

Q : Boolean True: Ok. False: 參數錯誤.

注意 :

1. I-8172 請連接到 Wincon 的 slot 1 到 7 之中, 然後再以其 Port0 或 Port1 連接 FRNET I/O 模組
2. FR_B , FR_B_A 必須在第一次 PLC scan 時呼叫. 第二次以後的 PLC scan 則無法呼叫成功.
3. FRNET D/O 模組不支援通訊狀態偵測, 而 FRNET D/I 模組則支援通訊狀態偵測.
4. 每個 FRNET 輸出模組的 Dip switch 上有一個 'RESET' dip 或特殊 Jumper. 需先將 'RESET' dip 或 Jumper 設到 'ON' 的位置 (或 enable), 如此一來, 當 I-8172 與 FRNET D/O 模組通訊中斷時, 便可重置輸出 channel 為 OFF 狀態.
例如: 將 FR-2057 的第 8 個 Dip 切到 'ON' 則可重新啓動.
5. 宣告 ISaGRAF 版本 3.4 (或 3.5) 的 "變數陣列", 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 下的 "isa.ini" 檔案頂端加入 2 行. 加入後, 開啓 ISaGRAF 工作平台, 在 Dictionary 宣告視窗內增加的 "DIM" 欄位中設定.

*請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行 :

```
[DEBUG]
arrays=1
```

6. 請參考下列網址 取得更多資訊:

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ048' 及
http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm

範例: 請參考網址 <http://www.icpdas.com/faq/isagraf.htm> 的 'FAQ048', 範例檔 Wdemo_39

GET_INFO

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

取得控制器資訊

輸入參數 :

TYPE_ : Integer 取得資訊種類

傳回值 :

DATA_ : Integer 傳回資訊的值.

"TYPE_" V.S. "DATA_" 對照表

TYPE_		DATA_
1	取得控制器的 ID (slave No.)	1 - 255
2	保留給未來設定	0
3	保留給未來設定	0
4	保留給未來設定	0

GET_SN

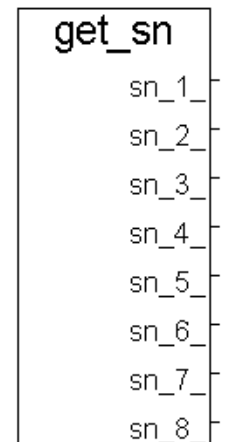
■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

取得 硬體唯一的 serial No. 共 8 個整數

傳回值 :

SN_1_ ~ SN_8_ : Integer 硬體唯一的 serial No



GET_VER

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

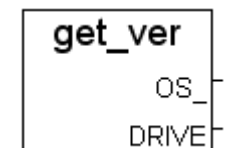
取得硬體驅動程式版本

(適用 I-8xx7: v2.19 , I-7188EG: v1.10 , I-7188XG: v1.08 或更新版本,)

傳回值 :

OS_ : Message 應該使用的 OS 版本 (長度: 48)
例如: "Must use 8n020704.img" 應更新為 8n020704.img

DRIVER_ : Message 目前的驅動程式版本 (長度: 48)
例如: "I-8xx7 : isa.exe - 2.19 , Dec.09,2002"



GETCTS

□ I-8x17 □ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function Block

取得 COM 埠的 CTS , 有效 COM 埠為 3 ~ 5

輸入參數 :

PORT_ : Integer 3:COM3 , 4:COM4, 5:COM5

傳回值 :

Q_ : Boolean 成功.: TRUE , 失敗 : FALSE

I_DICNT

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

取得與 I-8xx7 的 COM3/4 或與 I-7188XG/EG 的 COM2 或與 W-8xx7 的 COM3 連接的外接 RS-485 remote DI 模組的 4 個 DI counter 值。

輸入參數 :

ADR_ : Integer I/O 模組的位址 (1-255), 需為常數, 不能為變數
ST_CN_ : Integer 起始 Channel 編號. 需為常數, 不能為變數.
有效值 1 到 13.

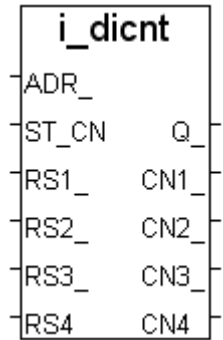
Ex: I-87052 有 8 個 DI.

若 ST_CN_ 設定為 5, 則 CN1_ 取得 Ch.5 的 counter 值,
CN2_ 取得 Ch.6, CN3_ 取得 Ch.7, CN4_ 取得 Ch.8 的值

RS1_ ~ RS4_ : Boolean 當 False 上升為 True, 重置相對應的 D/I counter 為 0.

傳回值 :

Q_ : Boolean 成功: TRUE, 失敗: FALSE
若 Q_ 傳回 FALSE, 表示通訊失敗, 下列回傳值無意義
CN1_ ~ CN4_ : Integer 由 ST_CN_ 起始的 channel 編號的 DI Counter 值為 0.



**下列模組有 DI counters (最大 100 Hz) :

"I_DICNT" 支援 : I-87051, 87052, 87053, 87054, 87055, 87058, 87063
I-7041, 7044, 7050, 7051, 7052, 7053, 7055, 7058, 7060, 7063, 7065

"I_DICNT2" 支援 : I-87040

Counter input channels: 4, 有效值 0 到 65535 (最大 100 Hz)

Reset Counter channels: 4, 當 False 上升為 True, 重置相對應 channel 的 D/I counter 為 0.

注意 :

- I-DiCnt 函式方塊適用版本 : I-8xx7:2.18, I-7188EG:1.10, I-7188XG:1.08, W-8x37:3.20C 或更新版本
- 遠程 I-87041 模組(32 D/I) 請使用 "i_DiCnt2" 函式方塊

特別重要 :

使用 RS-485 Remote I/O, 連接之前請注意下列各項 :

A. 在 I/O 模組端, 請使用 "DCON Utility" 來設定 :

- *1. 設定 "address" 為獨一無二的位址編號. (1~255)
- *2. 類比輸入模塊需設為 "2's complement" 格式, 類比輸出模塊需設為 "Engineering" 格式
- *3. 設定通訊參數的 "baud rate" 及 "8,N,1"
- *4. 若為類比板卡要設定為 "Range Type"

B. 在 I-8xx7/I-7188EG/I-7188XG/W-8xx7 主控端 :

- *1. 連結 I/O 請選擇 "complex equipment" 及 "bus7000B", 並設定其 "baud rate" 與 I/O 模組的 "baud rate" 相同, Checksum 也需相同.
- *2. I-8xx7, 7188EG/XG 最多可連接 64 個遠程 I/O 模組, W-8xx7 最多 255 個
- *3. 程式內連接 RS-485 Remote I/O 模組請選用 i_7*** function block 或 i_87*** Function Block.

I_DICNT2

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

取得與 I-8xx7 的 COM3/4 或與 I-7188XG/EG 的 COM2 或與 W-8xx7 的 COM3 連接的外接 RS-485 remote DI counter 值。

輸入參數 :

ADR_ : Integer I/O 模組的位址 (1-255), 需為常數, 不能為變數
ST_CN_ : Integer 起始 Channel 編號. 需為常數, 不能為變數.
有效值 1 到 29.

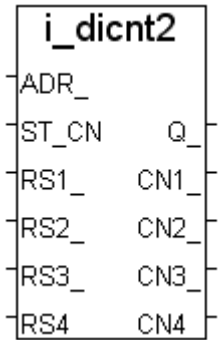
Ex: I-87040 有 32 個 DI.

若 ST_CN_ 設定為 15, 則 CN1_ 取得 Ch.15 的 counter 值,
CN2_ 取得 Ch.16, CN3_ 取得 Ch.17, CN4_ 取得 Ch.18 的值

RS1_ ~ RS4_ : Boolean 當 False 上升為 True, 重置相對應的 D/I counter 為 0.

傳回值 :

Q_ : Boolean 成功.: TRUE ,
若 Q_ 傳回 FALSE, 表示通訊失敗, 下列回傳值無意義
CN1_ ~ CN4_ : Integer 由 ST_CN_ 起始的 channel 編號的 DI Counter 值為 0.



**下列模組有 DI counters (最大 100 Hz) :

"I_DICNT" 支援 : I-87051 , 87052 , 87053 , 87054 , 87055 , 87058 , 87063

I-7041, 7044, 7050 , 7051, 7052 , 7053 , 7055, 7058, 7060 , 7063 , 7065

"I_DICNT2" 支援 : I-87040,

Counter input channels: 4 , 有效值 0 到 65535 (最大 100 Hz)

Reset Counter channels: 4 , 當 False 上升為 True, 重置相對應 channel 的 D/I counter

注意 :

- i-DiCnt 函式方塊適用版本 : I-8xx7:2.18 , I-7188EG:1.10 , I-7188XG:1.08 , W-8x37:3.20C 或更新版本
- 遠程 I-87041 模組(32 D/I) 請使用 "i_DiCnt2" 函式方塊

特別重要 :

使用 RS-485 Remote I/O, 連接之前請注意下列各項 :

A. 在 I/O 模組端, 請使用 "DCON Utility" 來設定 :

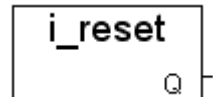
- *1. 設定 "address" 為獨一無二的位址編號. (1~255)
- *2. 類比輸入模塊需設為 "2's complement" 格式, 類比輸出模塊需設為 "Engineering" 格式
- *3. 設定通訊參數的 "baud rate" 及 "8,N,1"
- *4. 若為類比板卡要設定為 "Range Type"

B. 在 I-8xx7/I-7188EG/I-7188XG /W-8xx7 主控端 :

- *1. 連結 I/O 請選擇 "complex equipment" 及 "bus7000B", 並設定其 "baud rate" 與 I/O 模組的 "baud rate" 相同, Checksum 也需相同.
- *2. I-8xx7, 7188EG/XG 最多可連接 64 個遠程 I/O 模組, W-8xx7 最多 255 個
- *3. 程式內連接 RS-485 Remote I/O 模組請選用 i_7*** function block 或 i_87*** Function Block.

I_RESET

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

重新開機

傳回值 :

Q_ : Boolean 無作用

注意：使用本函式須非常小心謹慎。

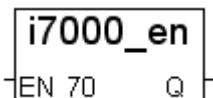
- 對於 iP-8xx7, I-8xx7, I-7188EG, I-7188XG :
若控制器一直重置, 請參考 "User's Manual Of The iP-8xx7" 使用手冊 第 1.3.7 節的說明 刪除 控制器內的專案.
- 對於 W-8xx7/8xx6 (Wincon ISaGRAF 版本) :
若控制器一直重置, 請暫時拔除 CF 卡然後重新啓動 WinCon 電源, 接著再次插上 CF 卡然後刪除以下檔案 - "\CompactFlash\ISaGRAF\ISA11". 之後, 再次重新啓動 WinCon 電源.
- 對於 WP-8xx7, VP-2xW7, XP-8xx7-CE6 :
若控制器一直重置, 請暫時將 PAC 關機, 將主機面板上的 Rotary Switch 轉到 1: Safe Mode 開機, 開好機後 將 \System_Disk\ISaGRAF\ 內的 ISA11 刪除, 之後關機, 切換 Rotary Switch 到 0: Normal 再開機.

範例 :

```
(* OK1 宣告爲 boolean input, TMP 宣告爲 boolean internal *)  
if OK1=TRUE then  
    TMP := i_reset();  
end_if;
```

I7000_EN

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

啓動 / 停止 "Bus7000"

輸入參數 :

EN_7000_ : Boolean TRUE: 啓動, FALSE: 停止

傳回值 :

Q_ : Boolean 永遠傳回 TRUE.

注意 :

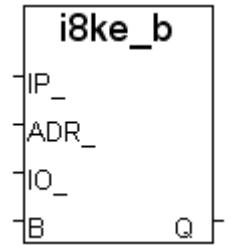
- 預設值是 啓動.
- 只在 IO 複合設備 "Bus7000" 與 "Bus7000B" 連接狀態下才有效.

I8KE_B

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Boolean 變數對應到一個 i8KE4/8-MTCP 的 Boolean I/O 點



輸入參數 :

IP_ :	Message	相關 i8KE4/8-MTCP 的 IP 位址, 例如: '192.168.100.123'
ADR_ :	Integer	i8KE4/8-MTCP 內 DI 或 DO 的 Modbus 位址, 0 到 267
IO_ :	Boolean	True: 輸入, False: 輸出
B_ :	Boolean	布林變數名稱

傳回值 :

Q_ :	Boolean	True: Ok. False: 參數錯誤.
-------------	---------	------------------------

注意 :

1. 請參考下列網址取得更詳細資料

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ042' 及

http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm

2. i8KE_B, i8KE_N, i8KE_F, i8KE_B_A, i8KE_N_A, i8KE_F_A 需在第一次 PLC scan 時呼叫. 第二次以後呼叫無效.

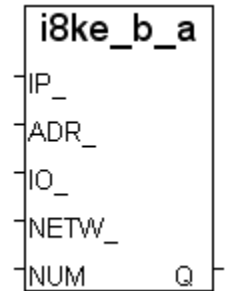
範例程式: Wdemo_30 & Wdemo_31 at <http://www.icpdas.com/faq/isagraf.htm> 'FAQ042'

I8KE_B_A

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Boolean 變數陣列對應到數個 i8KE4/8-MTCP 的 Boolean I/O 點. (請參考 2.6 節 關於變數陣列的說明)



輸入參數 :

- IP_ :** Message 相關 i8KE4/8-MTCP 的 IP 位址, 例如: '192.168.100.123'
- ADR_ :** Integer i8KE4/8-MTCP 內 DI 或 DO 的 Modbus 位址, 0 到 267
- IO_ :** Boolean True: 輸入, False: 輸出
- NetW_ :** Integer "變數陣列"第一個元素的網路位址編號. 1 ~ 8191
- Num_ :** Integer 使用乙太網路 IO 時設定變數陣列布林的數量, 有效範圍: 1 ~ 255. (ADR_ + Num_) 不能大於 264.
Ex: Bi[0..15] 大小為 16, NUM_ 可設為 1 ~ 16.
ABC[0..7] 大小為 8, NUM_ 可設為 1 ~ 8.

傳回值 :

- Q_ :** Boolean True: Ok. False: 參數錯誤.

注意 :

- 請參考下列網址取得更詳細資料
<http://www.icpdas.com/faq/isagraf.htm> 'FAQ042' 及
http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm
- i8KE_B, i8KE_N, i8KE_F, i8KE_B_A, i8KE_N_A, i8KE_F_A 需在第一次 PLC scan 時呼叫. 第二次以後呼叫無效.
- ISaGRAF 版本 3.4 (或 3.5) 的 "變數陣列" 宣告方式, 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 之下 "isa.ini" 檔案的最頂端加 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗裡會增加一個 "DIM" 欄位, 在此設定陣列.

請在 c:\isawin\exe\isa.ini 檔案最頂端, 加進 2 行程式碼 :

```
[DEBUG]
arrays=1
```

範例程式 : Wdemo_30 和 Wdemo_31 請參考 <http://www.icpdas.com/faq/isagraf.htm> 'FAQ042'

I8KE_F

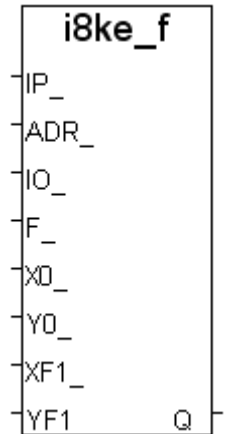
□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Real 變數對應到一個 i8KE4/8-MTCP 的類比 I/O 點. 並轉換為 Real 格式

輸入參數 :

IP_ :	Message	相關 i8KE4/8-MTCP 的 IP 位址, Ex: '192.168.100.123'
ADR_ :	Integer	i8KE4/8-MTCP 的 AI/AO Modbus 位址: 0~127
IO_ :	Boolean	True: 輸入 , False: 輸出
F_ :	REAL	REAL 變數名稱



----- 下列參數供數值轉換之用, 若不需轉換, 參數請設為 (0 , 0 , 0.0 , .0) -----

X0_ :	Integer	類比輸入/輸出卡的原始值. X0_ 不能等於 Y0_. 有效範圍: $-32768 \leq X0_ \leq +32767$
Y0_ :	Integer	類比輸入/輸出卡的原始值.. X0_ 不能等於 Y0_. 有效範圍: $-32768 \leq Y0_ \leq +32767$
XF1_ :	REAL	轉換後的工程值. XF1_ 不能等於 YF1_ .
YF1_ :	REAL	轉換後的工程值. XF1_ 不能等於 YF1_ .

傳回值 :

Q_ :	Boolean	True: Ok. False: 參數錯誤.
------	---------	------------------------

範例 :

Ex 1: 若 I-8017h 設定的 range_type 為 '+/- 10 V' (I-8017h's input value is -32768 to +32767). 使用者要將 (0 , 10 V) 轉換為工程值 (0 , 1000 Psi). 請設定 (X0_ , Y0_) = (0 , +32767) , (XF1_ , YF1_) = (0.0 , 1000.0)

Ex 2: 若 I-8024 設定 range_type 為 '0 to 20 mA' (I-8024's output value is 0 to +32767). 使用者要將 (4 , 20 mA) 轉換為工程值 (0 , 3000 rpm). 請設定 (X0_ , Y0_) = (6553 , +32767) , (XF1_ , YF1_) = (0.0 , 3000.0)

注意 :

1. 請參考下列網址取得更詳細資料

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ042' 及

http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm

2. i8KE_B , i8KE_N , i8KE_F , i8KE_B_A , i8KE_N_A , i8KE_F_A 需在第一次 PLC scan 時呼叫. 第二次以後呼叫無效.

範例程式: Wdemo_30 和 Wdemo_31 請參考 <http://www.icpdas.com/faq/isagraf.htm> 'FAQ042'

I8KE_F_A

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Real 變數陣列對應到數個 i8KE4/8-MTCP 的類比 I/O 點。
並轉換為 REAL 格式 (請參考 2.6 節 關於變數陣列的說明)

輸入參數 :

IP_ :	Message	對應 i8KE4/8-MTCP 的 IP 位址,ex: '192.168.100.123'
ADR_ :	Integer	i8KE4/8-MTCP 內 AI 或 AO 的 Modbus 位址: 0~127
IO_ :	Boolean	True: 輸入 , False: 輸出
NetW_ :	Integer	“變數陣列” 第一個元素的網路位址. 1~8191.
Num_ :	Integer	使用乙太網路 IO 時設定變數陣列 REAL 的數量, 有效範圍: 1 ~ 255. (ADR_ + Num_) 不能大於 128. Ex: R1[0..31] 大小為 32, NUM_ 可設為 1 ~ 32. R3[0..7] 大小為 8, NUM_ 可設為 1 ~ 8.

----- 下列參數供數值轉換之用, 若不需轉換, 參數請設為 (0 , 0 , 0.0 , .0) -----

X0_ :	Integer	類比輸入/輸出卡的原始值. X0_ 不能等於 Y0_. 有效範圍: $-32768 \leq X0_ \leq +32767$
Y0_ :	Integer	類比輸入/輸出卡的原始值.. X0_ 不能等於 Y0_. 有效範圍: $-32768 \leq Y0_ \leq +32767$
XF1_ :	REAL	轉換後的工程值. XF1_ 不能等於 YF1_ .
YF1_ :	REAL	轉換後的工程值. XF1_ 不能等於 YF1_ .

傳回值 :

Q_ :	Boolean	True: Ok. False: 參數錯誤.
------	---------	------------------------

範例 :

Ex : 若 I-87024 設定 range_type 為 '4 to 20 mA' (I-87024 輸出值為 0 ~ +32767). 使用者要將 (4 , 20 mA) 轉換為工程值 (0 , 5000 rpm). 請設定 (X0_ , Y0_) = (0 , +32767) , (XF1_ , YF1_) = (0.0 , 5000.0)

注意 :

1. 請參考下列網址取得更詳細資料

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ042' 及

http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm

2. i8KE_B , i8KE_N , i8KE_F , i8KE_B_A , i8KE_N_A , i8KE_F_A 需在第一次 PLC scan 時呼叫. 第二次以後呼叫無效.

3. ISaGRAF 版本 3.4 (或 3.5) 的 "變數陣列" 宣告方式, 請在 ISaGRAF 子目錄

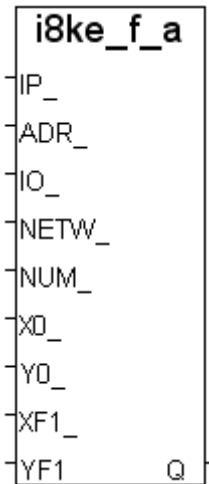
"C:\ISAWIN\EXE\" 之下 "isa.ini" 檔案的最頂端加 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗裡會增加一個 "DIM" 欄位, 在此設定陣列.

請在 c:\isawin\exe\isa.ini 檔案最頂端, 加進 2 行程式碼 :

```
[DEBUG]
```

```
arrays=1
```

範例程式 : Wdemo_30 及 Wdemo_31 請參考 <http://www.icpdas.com/faq/isagraf.htm> 'FAQ042'

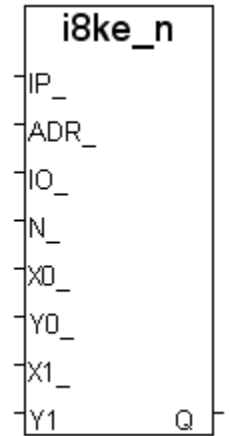


I8KE_N

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Integer 變數對應到一個 i8KE4/8-MTCP 的類比 I/O 點。
並轉換為 Integer 格式



輸入參數 :

IP_ :	Message	對應 i8KE4/8-MTCP 的 IP 位址, Ex: '192.168.100.123'
ADR_ :	Integer	i8KE4/8-MTCP 的 AI/AO Modbus 位址: 0~127
IO_ :	Boolean	True: 輸入 , False: 輸出
N_ :	Integer	Integer 變數名稱

----- 下列參數供數值轉換之用, 若不需轉換, 參數請設為 (0 , 0 , 0.0 , .0) -----

X0_ :	Integer	類比輸入/輸出卡的原始值. X0_ 不能等於 Y0_. 有效範圍: $-32768 \leq X0_ \leq +32767$
Y0_ :	Integer	類比輸入/輸出卡的原始值.. X0_ 不能等於 Y0_. 有效範圍: $-32768 \leq Y0_ \leq +32767$
X1_ :	Integer	比例後的工程值. X1_ 不能等於 Y1_. 有效範圍: $-30000 \leq X1_ \leq +30000$
Y1_ :	Integer	比例後的工程值. X1_ 不能等於 Y1_. 有效範圍: $-30000 \leq Y1_ \leq +30000$

傳回值 :

Q_ :	Boolean	True: Ok. False: 參數錯誤.
------	---------	------------------------

範例 :

Ex 1: I-8017h 設定的 range_type 為 '+/- 10 V' (I-8017h 輸入值為 -32768 ~ +32767). 使用者要將 (0 , 10 V) 轉換為工程值 (0 , 1000 Psi). 請設定 (X0_ , Y0_) = (0 , +32767) , (X1_ , Y1_) = (0 , 1000)

Ex 2: I-8024 設定 range_type 為 '0 to 20 mA' (I-8024 輸出值為 0 ~ +32767). 使用者要將 (4 , 20 mA) 轉換為工程值 (0 , 3000 rpm). 請設定 (X0_ , Y0_) = (6553 , +32767) , (X1_ , Y1_) = (0 , 3000)

注意 :

1. 請參考下列網址取得更詳細資料

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ042' 及

http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm

2. i8KE_B , i8KE_N , i8KE_F , i8KE_B_A , i8KE_N_A , i8KE_F_A 需在第一次 PLC scan 時呼叫. 第二次以後呼叫無效.

範例程式: Wdemo_30 及 Wdemo_31 請參考 <http://www.icpdas.com/faq/isagraf.htm> 'FAQ042'

I8KE_N_A

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 ISaGRAF Integer 變數陣列對應到數個 i8KE4/8-MTCP 的類比 I/O 點。
並轉換為 Integer 格式 (請參考 2.6 節 關於變數陣列的說明)

輸入參數 :

IP_ : Message 對應 i8KE4/8-MTCP 的 IP 位址,ex: '192.168.100.123'
ADR_ : Integer i8KE4/8-MTCP 內 AI 或 AO 的 Modbus 位址 . 0~127
IO_ : Boolean True: 輸入 , False: 輸出
NetW_ : Integer “變數陣列” 第一個元素的網路位址. 1~8191.
Num_ : Integer 使用乙太網路 IO 時設定變數陣列 Integer 的數量,
有效範圍: 1 ~ 255. (ADR_ + Num_) 不能大於 128.
Ex: ENG1[0..63] 大小為 64, NUM_ 可設為 1 ~ 64.
Ai[0..7] 大小為 8, NUM_ 可設為 1 ~ 8.

----- 下列參數供數值轉換之用, 若不需轉換, 參數請設為 (0 , 0 , 0.0 , .0) -----

X0_ : Integer 類比輸入/輸出卡的原始值. X0_ 不能等於 Y0_.
有效範圍: $-32768 \leq X0_ \leq +32767$
Y0_ : Integer 類比輸入/輸出卡的原始值.. X0_ 不能等於 Y0_.
有效範圍: $-32768 \leq Y0_ \leq +32767$
X1_ : Integer 轉換後的工程值. X1_ 不能等於 Y1_.
有效範圍: $-30000 \leq X1_ \leq +30000$
Y1_ : Integer 轉換後的工程值. X1_ 不能等於 Y1_.
有效範圍: $-30000 \leq Y1_ \leq +30000$

傳回值 :

Q_ : Boolean True: Ok. False: 參數錯誤.

範例 :

若 I-87018R 設定的 range_type 為 'Thermo-Couple K-type: -270 to +1372 degree celsius'
(I-87018R 輸入值為 -6448 ~ +32767). 使用者要將(-270 , +1372 degree) 轉換為工程值(-2700 , +13720). 請設定 (X0_ , Y0_) = (-6448 , +32767) , (X1_ , Y1_) = (-2700 , +13720)

注意 :

1. 請參考下列網址取得更詳細資料

<http://www.icpdas.com/faq/isagraf.htm> 'FAQ042' 及

http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm

2. i8KE_B , i8KE_N , i8KE_F , i8KE_B_A , i8KE_N_A , i8KE_F_A 需在第一次 PLC scan 時呼叫. 第二次以後呼叫無效.

3. ISaGRAF 版本 3.4 (或 3.5) 的 "變數陣列" 宣告方式, 請在 ISaGRAF 子目錄

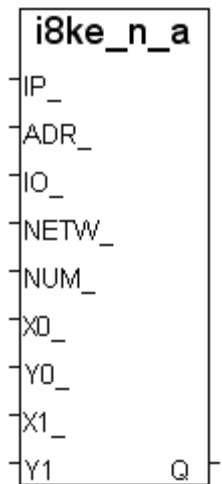
"C:\ISAWIN\EXE\" 之下 "isa.ini" 檔案的最頂端加 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗裡會增加一個 "DIM" 欄位, 在此設定陣列.

請在 c:\isawin\exe\isa.ini 檔案最頂端, 加進 2 行程式碼:

```
[DEBUG]
```

```
arrays=1
```

範例程式 : Wdemo_30 及 Wdemo_31 請參考 <http://www.icpdas.com/faq/isagraf.htm> 'FAQ042'



INP10LED

■ I-8x17 ■ iP-8xx7 □ 718xEG □ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

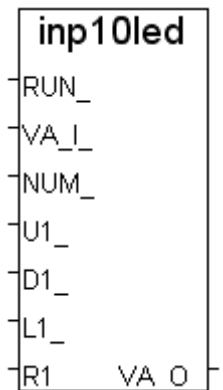
從 S-MMI 輸入 1 個 10 進位的整數

輸入參數：

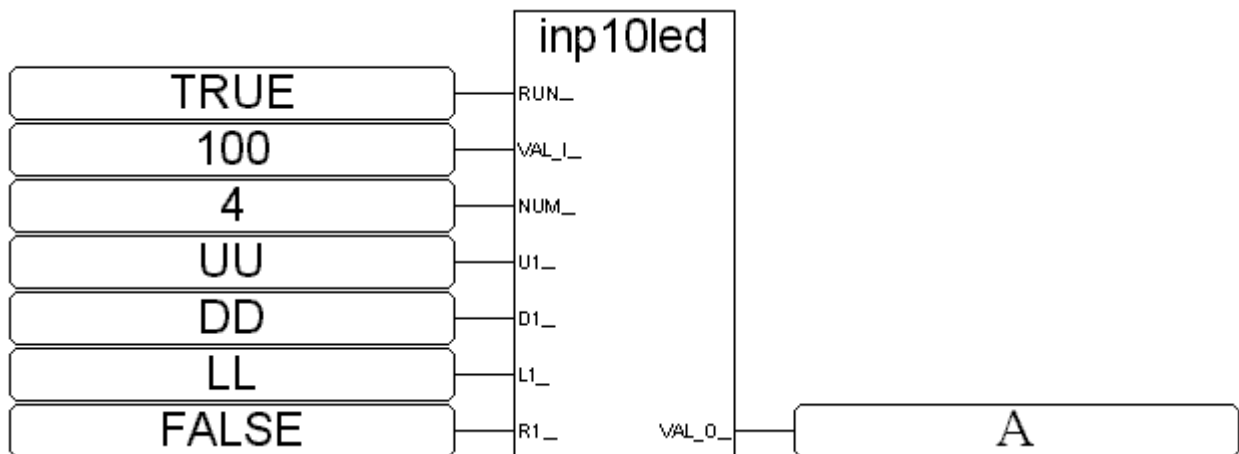
RUN_	Boolean	當為 TRUE 時才動作
VAL_I_	Integer	顯示在 S-MMI 的初值 (0 ~ 99999)
NUM_	Integer	要顯示幾位數 (1 ~ 5)
U1_	Boolean	由 FALSE 升到 TRUE 時,目前位置的值加 1
D1_	Boolean	由 FALSE 升到 TRUE 時,目前位置的值減 1
L1_	Boolean	由 FALSE 升到 TRUE 時,目前位置往左移
R1_	Boolean	由 FALSE 升到 TRUE 時,目前位置往右移

傳回值：

VAL_O_	Integer	經過操作後的整數值
---------------	---------	-----------



範例：請參考 demo_08 及 demo_11a.



ST 相等式:

```
A := INP10LED(TRUE,100,4,UU,DD,LL,FALSE);
(* A 需宣告 integer, 屬性為 internal *)
(* UU,DD,LL 可宣告為 boolean, 屬性為 input, 可連接至 push4key *)
```

INP16LED

■ I-8x17 ■ iP-8xx7 □ 718xEG □ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

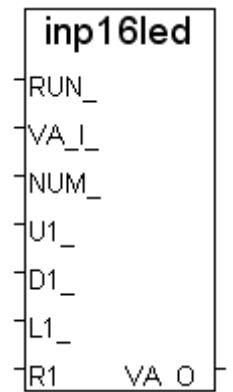
從 S-MMI 輸入 1 個 16 進位的整數

輸入參數：

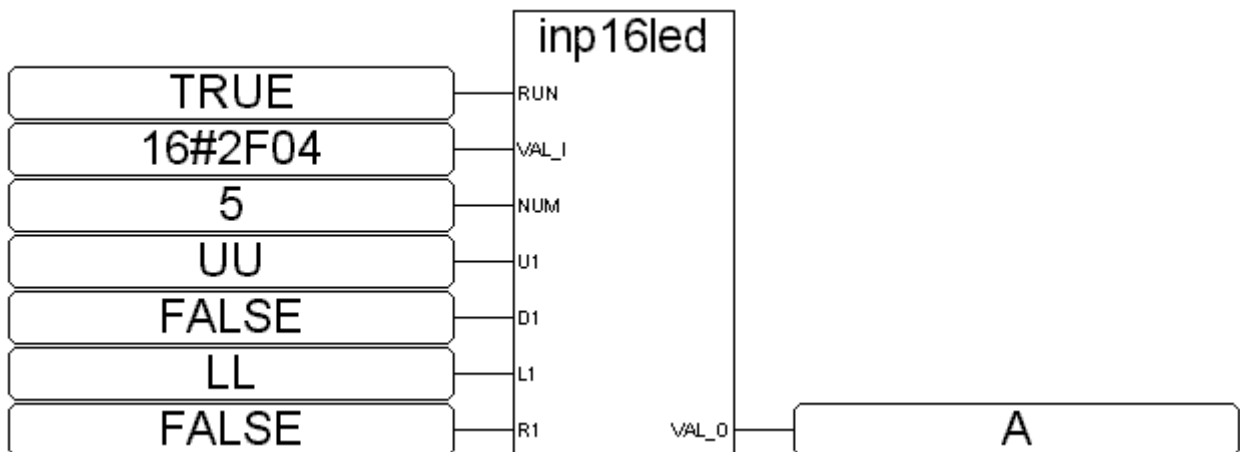
RUN_	Boolean	當為 TRUE 時才動作
VAL_I_	Integer	顯示在 S-MMI 的初值 (0 ~ 16#FFFFFF)
NUM_	Integer	要顯示幾位數 (1 ~ 5)
U1_	Boolean	由 FALSE 升到 TRUE 時,目前位置的值加 1
D1_	Boolean	由 FALSE 升到 TRUE 時,目前位置的值減 1
L1_	Boolean	由 FALSE 升到 TRUE 時,目前位置往左移
R1_	Boolean	由 FALSE 升到 TRUE 時,目前位置往右移

傳回值：

VAL_O_	integer	經過操作後的整數值
---------------	---------	-----------



範例：



ST 相等式：

```
A := INP16LED(TRUE,16#2F04,4,UU,FALSE,LL,FALSE);
(* A 需宣告 integer, 屬性為 internal *)
(* UU,LL 可宣告為 boolean, 屬性為 input, 可連接至 push4key *)
```

INT_REAL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

對應 1 個長整數成爲 1 個實數. 以 C 語言的語法表示爲 `Real_ = *((float *)&Long_)`

輸入參數 :

Long_ Integer 要被對應的 integer 值

傳回值 :

Real_ Real 對應完後的 real 值

注意:

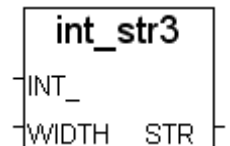
1. "Real_Int" 可用來對應 real 成 integer
2. 如果只是要轉換一個整數爲實數, 比如 `32 → 32.0`, 請用 `Real()`

非常重要:

誤用 "int_real(L1)" 可能會使控制器產生 Float Error, 請參考 ISaGRAF 手冊 10.6 節.

INT_STR3

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

將 Integer 轉換爲 String, 有長度限制

輸入參數 :

INT_ : Integer 要轉換的 integer 值.

WIDTH_ : Integer 指定最大顯示位數, 1 ~ 13

傳回值 :

STR_ : Message 傳回的字串 (最大長度: 13).

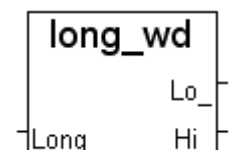
Ex. 若 `WIDTH_ = 4`

`12 ---> ' 12'`

`123456 ---> '****'`

LONG_WD

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function Block

將 1 個長整數 (signed 32-bit) 轉換成 2 個 word (signed 16-bit)

輸入參數 :

LONG_ : Integer 要轉換的 32-bit integer 值.

傳回值 :

LO_ Integer 轉換後的 low word (-32768 ~ +32767)

HI_ Integer 轉換後的高 word (-32768 ~ +32767)

MBUS_B_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

從 Modbus 設備一次讀取 8 個 Boolean 值

使用 Modbus function 1

輸入參數 :

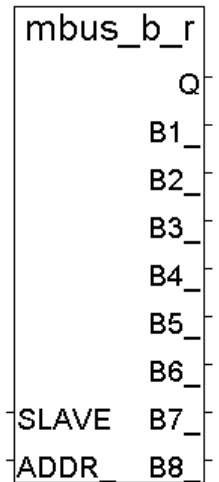
SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 位址開始讀(0~65535). , 需為常數值, 非變數

傳回值 :

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
B1_ ~ B8_	Boolean	讀到的 8 個 Boolean 值

注意: 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_B_R + MBUS_BR1).
W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例: 請參考第 8 章



MBUS_B_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

寫出 1 到 4 個 Boolean 值到 Modbus 設備

當 NUM_W=1 時, 使用 Modbus function 5

當 NUM_W=2 到 4 時, 使用 Modbus function 15

輸入參數 :

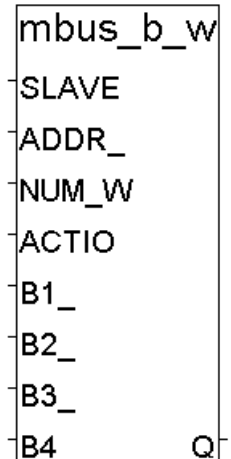
SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 資料位址開始寫(0~65535) 需為常數值, 非變數
NUM_W_	Integer	要寫幾個 boolean 值 (1 ~ 4), 需為常數值, 非變數
ACTION_	Boolean	設為 TRUE 才寫
B1_ ~ B4_	Boolean	要寫出的 boolean 值

傳回值 :

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
-----------	---------	-----------------------

注意: 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_B_W + MBUS_WB).
W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例: 請參考第 8 章



MBUS_BR1

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

每隔一段時間從 Modbus 設備一次讀取 8 個 Boolean 值

使用 Modbus function 1

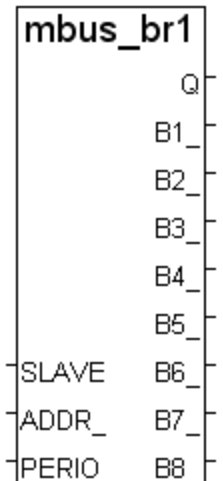
輸入參數：

SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 資料位址開始讀(0~65535) 需為常數值, 非變數
PERIOD_	Integer	每隔多久去讀 (1~600), 單位為 秒

傳回值：

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
B1_ ~ B8_	Boolean	讀到的 8 個 Boolean 值

注意：同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_B_R + MBUS_BR1).
W-8xx7 控制器的每一個 COM Port 則最多使用 256 個



MBUS_N_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

從 Modbus 設備一次讀取 8 個 Word (signed 16-bit)

使用 Modbus function 3

輸入參數 :

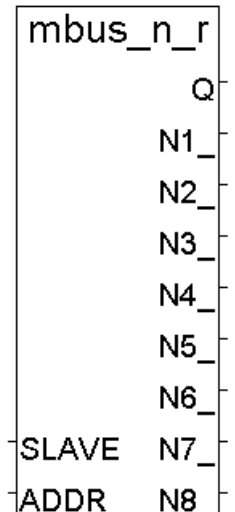
SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 資料位址開始讀(0~65535), 需為常數值, 非變數

傳回值 :

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
N1_ ~ N8_	Integer	讀到的 8 個 Word 值 (-32768 ~ 32767)

注意: 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_N_R + MBUS_R + MBUS_NR1 + MBUS_R1).
W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例: 請參考第 8 章



MBUS_N_W

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

寫出 1 到 4 個 word (signed 16-bit) 值到 Modbus 設備

當 NUM_W=1 時, 使用 Modbus function 6

當 NUM_W=2 到 4 時, 使用 Modbus function 16

輸入參數：

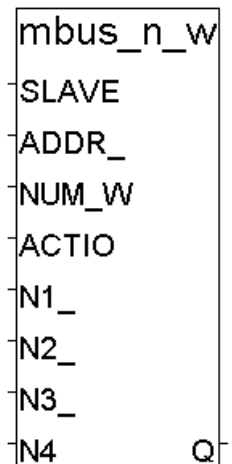
SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 資料位址開始讀(0~65535), 需為常數值, 非變數
NUM_W_	Integer	要寫幾個 word 值 (1 ~ 4), 需為常數值, 非變數
ACTION_	Boolean	設為 TRUE 才寫
N1_ ~ N4_	Integer	要寫出的 word 值 (-32768 ~ 32767)

傳回值：

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
-----------	---------	-----------------------

注意： 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 MBUS_N_W。
W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例： 請參考第 8 章



MBUS_NR1

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

每隔一段時間從 Modbus 設備一次讀取 8 個 Word (signed 16-bit)

使用 Modbus function 3

輸入參數：

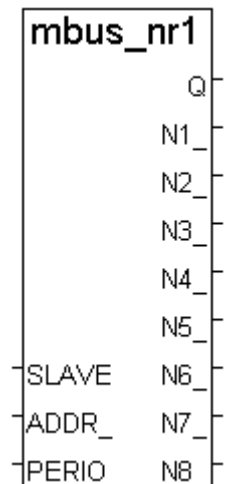
SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 資料位址開始讀(0~65535), 需為常數值, 非變數
PERIOD_	Integer	每隔多久去讀 (1~600), 單位為 秒

傳回值：

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
N1_ ~ N8_	Integer	讀到的 8 個 Word 值 (-32768 ~ 32767)

注意：同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_N_R + MBUS_NR1 + MBUS_R + MBUS_R1)
W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例： 請參考第 8 章



MBUS_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

從 Modbus 設備 讀取資料

* 可選擇使用 Modbus function 1 或 2 或 3 或 4

輸入參數 :

SLAVE_	Integer	<p>值需為常數值, 非變數</p> <p>Modbus 設備的站號 = SLAVE_ / 1000 的餘數</p> <p>ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數</p> <p>I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5)</p> <p>I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3)</p> <p>I-7188XG : (2:COM2, 3:COM3)</p> <p>W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14)</p> <p>例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.</p>
ADDR_	Integer	<p>從那個 Modbus 資料位址開始讀(0~65535),</p> <p>需為常數值, 非變數</p>
CODE_	Integer	<p>使用那個 Modbus function 編號, 1 - 4</p> <p>需為常數值, 非變數</p>
NUM_	Integer	<p>如果 CODE_=1 或 2 表示詢問幾個 bit ? 1-192</p> <p>如果 CODE_=3 或 4 表示詢問幾個 word ? 1- 12</p> <p>需為常數值, 非變數</p>

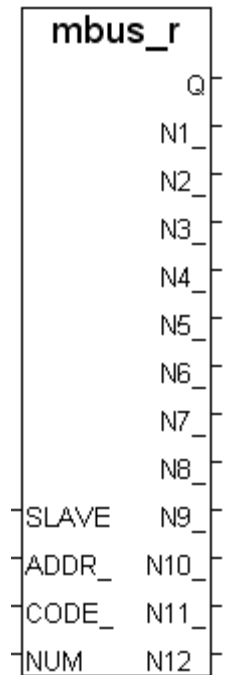
傳回值 :

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
N1_ ~ N12_	Integer	<p>讀到的 Bit 或 Word 值</p> <p>如果 CODE_=1 或 2, N1_傳回 bit 1 到 16, N2_ 傳回 bit 17 到 32, ...</p> <p>N12_傳回 bit 177 到 192</p> <p>如果 CODE_=3 或 4, N1_ 到 N12_ 傳回 Word 1 到 12</p> <p>(值介於-32768 到 32767).</p>

注意: 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_N_R + MBUS_NR1 + MBUS_R + MBUS_R1)

W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例: 請參考第 8 章



MBUS_R1

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

每隔一段時間從 Modbus 設備讀取資料

* 可選擇使用 Modbus function 1 或 2 或 3 或 4

輸入參數 :

SLAVE_	Integer	值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.
ADDR_	Integer	從那個 Modbus 資料位址開始讀(0~65535), 需為常數值, 非變數
CODE_	Integer	使用那個 Modbus function 編號, 1 - 4 需為常數值, 非變數
NUM_	Integer	如果 CODE_=1 或 2 表示詢問幾個 bit ? 1-192 如果 CODE_=3 或 4 表示詢問幾個 word ? 1- 12 需為常數值, 非變數
PERIOD_	Integer	每隔多久去讀 (1~600), 單位為 秒

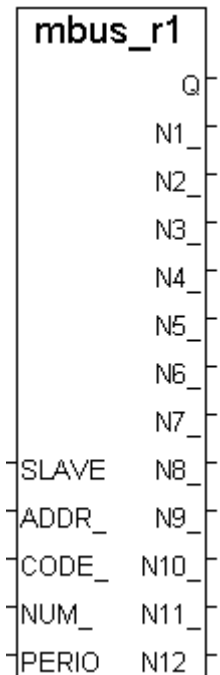
傳回值 :

Q_	Boolean	正確傳回 TRUE, 失敗傳回 FALSE
N1_ ~ N12_	Integer	讀到的 Bit 或 Word 值 如果 CODE_ =1 或 2, N1_傳回 bit 1 到 16, N2_ 傳回 bit 17 到 32, ... N12_傳回 bit 177 到 192 如果 CODE_ =3 或 4, N1_ 到 N12_ 傳回 word 1 到 12 (值介於 -32768 ~ 32767).

注意: 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_N_R + MBUS_NR1 + MBUS_R + MBUS_R1)

W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例: 請參考第 8 章



MBUS_WB

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

寫出 1~16 個 boolean 值到 Modbus 設備

使用 Modbus function 15

輸入參數 :

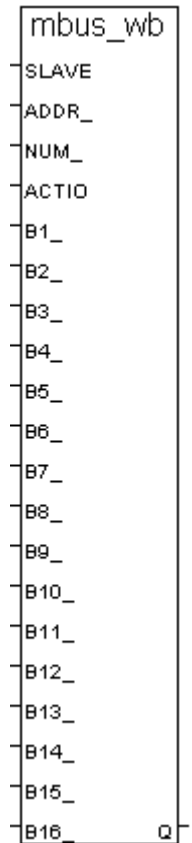
SLAVE_	Integer	<p>值需為常數值, 非變數 Modbus 設備的站號 = SLAVE_ / 1000 的餘數 ISaGRAF Controller 使用的 COM port = SLAVE_ / 1000 的商數 I-8xx7 : (1:COM1, 3:COM3, 4:COM4, 5:COM5) I-7188EG/7186EG : (1:COM1, 2:COM2, 3:COM3) I-7188XG : (2:COM2, 3:COM3) W-8xx7/8xx6 : (2:COM2, 3:COM3, 4:COM5 到 14:COM14) 例如: SLAVE_ = 3001, 則使用 COM3, 設備的站號是 1.</p>
ADDR_	Integer	<p>從那個 Modbus 資料位址開始讀(0~65535), 需為常數值, 非變數</p>
NUM_W_	Integer	<p>要寫幾個 boolean 值 (1 ~ 16), 需為常數值, 非變數</p>
ACTION_	Boolean	<p>設為 TRUE 才寫</p>
B1_ ~ B16_	Boolean	<p>要寫出的 boolean 值</p>

傳回值 :

Q_	Boolean	<p>正確傳回 TRUE, 失敗傳回 FALSE</p>
-----------	---------	------------------------------

注意: 同一台 I-8xx7 或 I-7188EG/XG 控制器最多只能使用 64 個 (MBUS_B_W + MBUS_WB)
 W-8xx7 控制器的每一個 COM Port 則最多使用 256 個

範例: 請參考第 8 章



MI_BOO

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 MMICON 上顯示 1 個 Boolean 值



輸入參數 :

X_	Integer	位置 X, 1-30
Y_	Integer	位置 Y, 1-8
BOO_	Boolean	要顯示的 boolean 值. TRUE 顯示“ON”, FALSE 顯示 “OFF”

傳回值 :

Q_	Boolean	Ok. 傳回 TRUE, 失敗為 FALSE
-----------	---------	------------------------

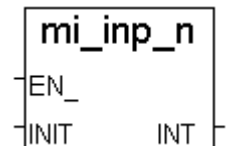
範例: 請參考第 16 章 & demo_38, demo_39

MI_INP_N

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 MMICON 上輸入 1 個 integer 值



輸入參數 :

EN_	Boolean	TRUE: 才動作
INIT_	Integer	初值

傳回值 :

INT_	Integer	傳回的 integer 輸入值. 如果 EN_ 設為 FALSE , 傳回 0
-------------	---------	---

注意 :

MI_INP_N 及 MI_INP_S 只可在 1 個 ISaGRAF Project 內使用 1 次, 同時在同個 Project 內的兩個或以上的地方出現會無法正常工作.

範例: 請參考第 16 章 & demo_38, demo_39

MI_INP_S

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 MMICON 上輸入 1 個字串



輸入參數 :

EN_ Boolean TRUE: 才動作
INIT_ Message 字串初值

傳回值 :

STR_ Message 傳回的字串輸入值. 如果 EN_ 設為 FALSE , 傳回 '' (空字串)

注意:

MI_INP_N 及 MI_INP_S 只可在 1 個 ISaGRAF Project 內使用 1 次, 同時在同個 Project 內的兩個或以上的地方出現會無法正常工作.

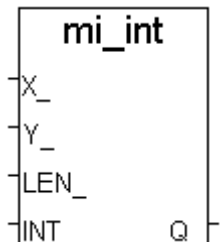
範例: 請參考第 16 章 & demo_38, demo_39

MI_INT

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 MMICON 上顯示 1 個 integer 值



輸入參數 :

X_ Integer 位置 X, 1-30
Y_ Integer 位置 Y, 1-8
LEN_ Integer 最多可顯示幾個字元, 1-11
INT_ Integer 要顯示的 integer 值, 長度超過 LEN_ 會顯示 '*****'

傳回值 :

Q_ Boolean Ok. 傳回 TRUE, 失敗為 FALSE

範例: 請參考第 16 章 & demo_38, demo_39

MI_REAL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 MMICON 上顯示 1 個 real 值

輸入參數 :

X_	Integer	位置 X, 1-30
Y_	Integer	位置 Y, 1-8
LEN_	Integer	最多可顯示幾個字元, 1-13
LEN1_	Integer	小數點後可顯示幾個字元, 0~4 且值需比 LEN_ 小. 例, 若 LEN_=7, LEN1_=2, "123.4567" 會顯示成 " 123.45"
REAL_	Real	要顯示的 Real 值. 若位數超過 LEN_, 會顯示 '*****'

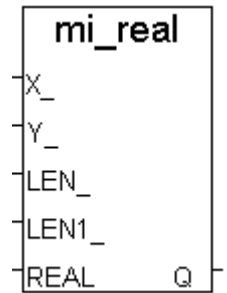
傳回值 :

Q_	Boolean	Ok. 傳回 TRUE, 失敗為 FALSE
-----------	---------	------------------------

注意:

假如要顯示的實數的絕對值 ($\geq 1,000,000$) 或 (不等於 0 且 < 0.0001), 請設定 LEN_ 為 13. 例, -123,456,789, 請設 LEN_ 為 13, 它會顯示為 -1.23457e+008. 又如 0.0000123456, 請設 LEN_ 為 13, 它會顯示為 1.23456e-005

範例: 請參考第 16 章 & demo_38, demo_39



MI_STR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 MMICON 上顯示 1 個字串

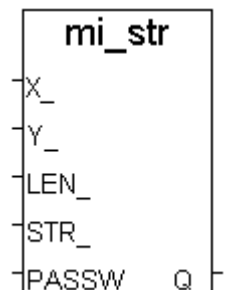
輸入參數 :

X_	Integer	位置 X, 1-30
Y_	Integer	位置 Y, 1-8
LEN_	Integer	最多可顯示幾個字元, 1-240
STR_	Message	要顯示的字串. 假如字元數量超過 LEN_, 超過的部份不顯示
PASSWD_	Boolean	設為 TRUE 表示要顯示成密碼, 所有字元以 * 代替, 設為 FALSE 則為正常顯示.

傳回值 :

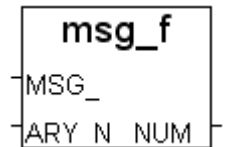
Q_	Boolean	Ok. 傳回 TRUE, 失敗為 FALSE
-----------	---------	------------------------

範例: 請參考第 16 章 & demo_38, demo_39



MSG_F

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從字串(Message) 讀出一些 REAL 值(浮點值), 並儲存在浮點陣列

輸入參數 :

MSG_	Message	要讀出的 Message(字串), 1~255 bytes (characters)
ARY_NO_	Integer	存放讀出的 REAL 值的浮點陣列編號 I-8xx7, I-718xEG/XG : 1 ~ 6 (其 Float 陣列和 Integer 陣列使用同一個的記憶體, 請小心使用) W-8xx7/8xx6 : 1 ~ 18

傳回值 :

NUM_	Integer	讀到的 REAL 值的總數 (0 ~ 128), 若傳回 -1, 表示格式錯誤
-------------	---------	--

注意:

1. REAL 值可以使用 "ARY_F_R" 和 "ARY_F_W" 函式讀出和寫入。
2. 字串的 REAL 值之間隔開需用"空格"或"逗號"或"TAB" 或 "ENTER" 或 "NEW LINE" 字元
3. 若字串中對應的值不是正確的 REAL 格式, 例如: '1.23 , 2.45A , 3.0 , 2+3 ' 中的第二和第四個值不是正確的 REAL 格式, 函式傳回的值就會是 -1.
4. 從以下版本起的驅動程式才有支援 MSG_F 與 MSG_N:

I-7188EG: 2.17 版起

I-7188XG: 2.15 版起

I-8XX7 : 3.19 版起

W-8XX7 : 建議更新至 3.36 版或更高版

範例:

- (1.) 若指定 MSG_ = '2.3 , -567.002 , +0.0025 , 1 , 1.4E-5', 而 ARY_NO_ = 1, 則傳回 NUM_ 為 5. 浮點陣列中儲存的 5 個 REAL 值, 由編號 1 到 5 的位址分別是:
addr.1 = 2.3 , addr.2 = -567.002 , addr.3 = 0.0025 , addr.4 = 1.0 , addr.5 = 1.4E-5

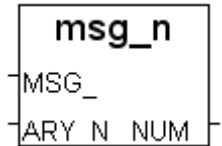
- (2.) 若指定 MSG_ = '4.01 , -8.09 , +-3.45', 而 ARY_NO_ = 2, 則會傳回 -1, 因為第 3 個值 +-3.45 是不正確的 REAL 格式.

範例程式 :

1. wdemo_52.pia 放置於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\
2. wdemo_52.pia 放在 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

MSG_N

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從字串(Message) 讀出一些 Integer 值, 並儲存在 Integer 陣列

輸入參數 :

MSG_	Message	要讀出的 Message(字串), 1~255 bytes (characters)
ARY_NO_	Integer	存放讀出的 Integer 值的 Integer 陣列編號 I-8xx7, I-718xEG/XG : 1 ~ 6 (其 Float 陣列和 Integer 陣列使用同一個的記憶體, 請小心使用) W-8xx7/8xx6 : 1 ~ 18

傳回值 :

NUM_	Integer	讀到的 Integer 值的總數 (0 ~ 128), 若傳回 -1, 表示格式錯誤
-------------	---------	---

注意:

- Integer 值可以使用 "ARY_N_R" 和 "ARY_N_W" 函式讀出和寫入。
- 字串的 Integer 值之間隔開需用"空格"或"逗號"或"TAB" 或 "ENTER" 或 "NEW LINE" 字元
- 若字串中對應的值不是正確的 Integer 格式, 例如: '123 , -8G , 3' 中的第二個值不是正確的 Integer 格式, 函式傳回的值就會是 -1.
- 從以下版本起的驅動程式才有支援 MSG_F 與 MSG_N:
 - I-7188EG: 2.17 版起
 - I-7188XG: 2.15 版起
 - I-8XX7 : 3.19 版起
 - W-8XX7 : 建議更新至 3.36 版或更高版

範例:

(1.) 若指定 MSG_ = '3 , -567 +2', 而 ARY_NO_ = 1, 則傳回 NUM_ 為 3.

Integer 陣列中儲存的 3 個 Integer 值, 由編號 1 到 3 的位址分別是:

addr.1 = 3 , addr.2 = -567 , addr.3 = 2

(2.) 若指定 MSG_ = '401 , 3A , +-345', 而 ARY_NO_ = 2, 則會傳回 -1, 因為第 2 個值 '3A' 與第 3 個值 '+-345' 是不正確的 Integer 格式.

範例程式 :

- wdemo_53.pia 放置於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\
2. wdemo_53.pia 放在 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

MSGARY_R

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從 Message 陣列讀出 1 個字串

輸入參數 :

ADDR_ Integer 位置, 1 - 1024

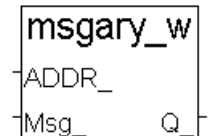
傳回值 :

Msg_ Message 讀到的字串 (字串長度範圍 0 ~ 255)

範例: 請參考第 11 章 Wincon-8xx7 的 wdemo_06

MSGARY_W

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 1 個字串到 Message 陣列

輸入參數 :

ADDR_ Integer 位置, 1 - 1024

MSG_ Message 要寫入的的字串 (字串長度範圍 0 ~ 255)

傳回值 :

Q_ Boolean True: 成功, False: 失敗

範例: 請參考第 11 章 Wincon-8xx7 的 wdemo_06

PID_AL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

範例：

請參考第 11 章 - Demo_18, 與 ICP DAS 光碟：

\Napdos\ISaGRAF\8000\English_Manu\PID_AL.Complex PID algorithm implementation.htm

PID_AL 原是由 CJ Internal 公司提供的 function, 以下為 CJ 公司的說明 (部分譯為中文)

Author : EDS

Product : ISaGRAF V3

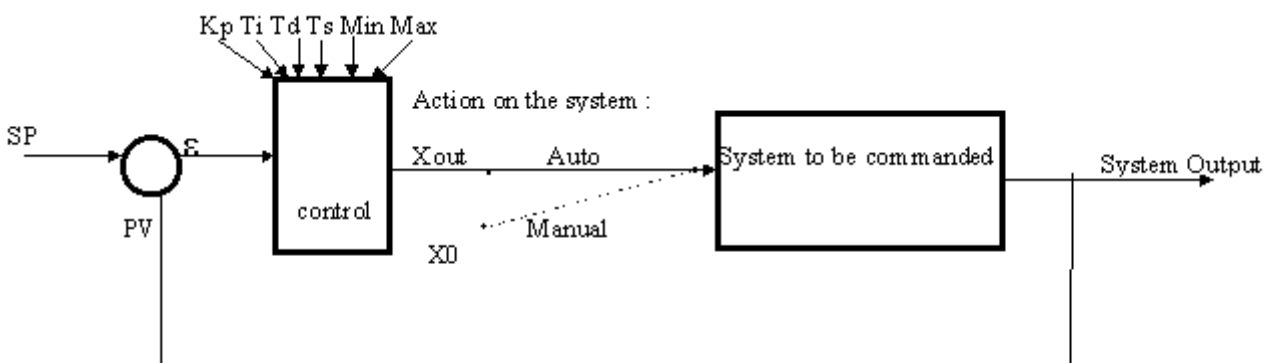
Date : 26 aug 96

File : PID_AL.Complex PID algorithm implementation.htm

Subject : Complex PID algorithm implementation

Keywords: PID - PID_AL

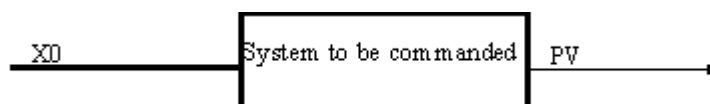
PID 是一個進程的調節校正。以反饋的觀念 將輸出(output) 在實際數值與期望數值的差距間做調節。



PV 是輸出(Output)。入口處的 **e** 是 **SP-PV** 間計算而得的差距。**Control** 則計算 系統維持進程調節應做的動作。

System to command : 在模擬(仿真)作業中, 是個第二順位系統。

相對的, 在沒有模擬的系統, 則是如下圖:



PID 的主要各點描述如下:

SP : 是設定點, 設定期望的輸出值。

X0 (手動模式中, 開啓迴圈的 case) 是進入系統的未調節值。

Xout (已調節用以結束迴圈的 case) 是進入系統的已調節值。

CJ PID-A1 說明:

輸入參數:

Auto: 自動或手動模式

Pv : 程序的輸出值

Sp : 設定為設定點的輸出期望值

X0 : 調節值: 手動模式時, 輸出 PID 控制元等於 X0

Kp : 比例性常數

Ti : 整體性 (積分式) 時間常數 (Integral time constant)

Td : 衍生性時間常數 (derivative time constant)

Ts : 採樣週期

Min, Max : Xout 值接受的範圍

傳回值:

Xout : 指令

原型: `PID(Auto, Pv, Sp, X0, Kp, Ti, Td, Ts, Min, Max);`

`Command:= PID.Xout;`

注意: 自動模式的初始狀態必須設為 false

- 完成式運算原型 :

$$u(t) = K_p(A(t)) + \frac{1}{T_i} \int_0^t A(t) dt + T_d \frac{dA(t)}{dt} \quad \text{Theoretical Continue (理論持續) 方程式}$$

$$u(k) = K_p(A(k)) + \frac{T_s}{T_i} I(k) + \frac{T_d}{T_s} [A(k) - A(k-1)] \quad \text{Implemented Discret (完成離散) 方程式}$$

$$I(k) = I(k-1) + A(k)T_s$$

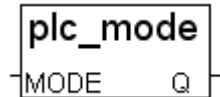
Copyright CJ International 1996.

PLC_MODE

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

改變 PAC 的 ISaGRAF 程式速度



輸入參數 :

MODE_ : Integer 可以是 0, 1, 2, 或 3
0: 快速模式, 預設值為此模式, 最小 PLC scan 時間是約 3 ms
1: 慢速模式, 最小 PLC scan 時間是約 6 到 7 ms
2: 更慢速模式, 最小 PLC scan 時間是約 9 到 11 ms
3 或其他值: 最慢速模式, 最小 PLC scan 時間是約 19 到 21 ms

傳回值 :

Q : Boolean 永遠傳回 TRUE

注意 :

1. Winon-8xx7 從 3.24B 版本才開始支援 "PLC_mode" 函式
2. 預設的模式是 "快速模式"
3. 使用者可以在第一次 PLC scan 就呼叫 "PLC_mode()" function 來改變 PLC 速度.
4. 通常降慢 PLC 速度的原因是爲了增進其他與 ISaGRAF 程式同時執行的 HMI 程式的執行效率. 例如: 一台 WinCon 中同時執行 Indusoft 和 ISaGRAF 程式的狀況.

範例 :

```
(* TMP 宣告爲布林內部變數 *)
(* INIT 宣告爲布林內部變數, 初始值 TRUE *)
if INIT then
  INIT := False ; (* 只在第一回 PLC scan 中執行*)
  TMP := PLC_mode(2); (* 設定 PLC 速度爲 2: 更慢速模式*)
end_if ;
```

下列各 PWM 函式皆為 C_Function, 詳細內容與範例 請參考手冊 第 3.7 節

PWM_DIS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 關閉 PWM 輸出

PWM_EN

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 開啓 PWM 輸出.

PWM_EN2

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 開啓 PWM 輸出一給定數量的脈波

PWM_OFF

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 立刻將 parallel D/O 輸出為 FALSE

PWM_ON

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 立刻將 parallel D/O 輸出為 TRUE

PWM_SET

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 動態更改 ON_, OFF_ & NUM_ 的設定

PWM_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 測試 PWM 的狀態

PWM_STS2

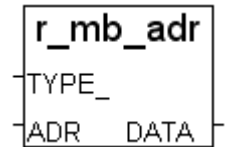
■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
Function 取得”pwm_en2”與”pwm_en”已輸出的 pulse 數量

R_MB_ADR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

使用 Modbus 位址讀取 boolean 或 integer 變數的值



輸入參數：

TYPE_ : Integer 0: 布林變數 , 1: 整數變數
ADR_ : Integer 讀取的 Modbus 資料位址 ,
有效範圍: I-8xx7 & I-7188xG 是 1~4095, Wincon 是 1~8191

傳回值：

DATA_ : Integer 整數值 (若 TYPE 是 boolean, 1 代表 True, 0 代表 False)

注意：

1. 請使用 R_MB_REL function 讀取 "REAL" 變數.
2. 若沒有該 Modbus 位址定義的變數, 則傳回 0.
3. 若給予的 TYPE_ 為整數 而對應的變數是 "Boolean" 型態, 則傳回值: 0:表 False , 1:表 True
4. 若給予的 TYPE_ 為整數 而對應的變數是 "Real", 則對應的 32-bit 會複製到 DATA_ 中.
您可使用 "int_real" 函式將 32-bit 整數對應到實數值. (最好使用 "R_MB_REL" 來讀取 "Real" 變數)
5. 若給予的 TYPE_ 為布林 而對應的變數不是 "Boolean" 型態, 則傳回值: 0 .
6. 若長整數 (32-bit 整數) 透過 Modbus 通訊協定傳遞到 HMI, 則必需佔用 2 個 Modbus 位址, 請參考 ISaGRAF 進階使用手冊 第 4.2 節.

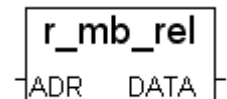
範例：請參考 ISaGRAF Projects/ Tools/ Libraries 說明

R_MB_REL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

使用 Modbus 位址讀取 Real 變數的值



輸入參數：

ADR_ : Integer 讀取的 Modbus 資料位址 ,
有效範圍: I-8xx7 & I-7188xG 是 1~4095, Wincon 是 1~8191

傳回值：

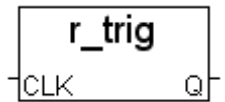
DATA_ : Real 讀取的實數值

注意：

1. 請確認對應的變數型態為 "Real". 若為 "Integer", 請使用 "R_MB_ADR" function.
2. 若對應的變數型態不是 "Analog" (Real 或 integer), 則傳回 1.23E-20
3. 若沒有以該 Modbus 位址定義的變數, 則傳回 1.23E-20
 4. 若實數值透過 Modbus 通訊協定傳遞到 HMI, 則必需佔用 2 個 Modbus 位址, 請參考 ISaGRAF 進階使用手冊 第 4.2 節.
 5. 若該 Modbus 位址變數不是實數, 有可能會產生 Local Fault 編號 103, 請參考 10.6 節.

R_TRIG

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

偵測布林變數的上昇變化

輸入參數 :

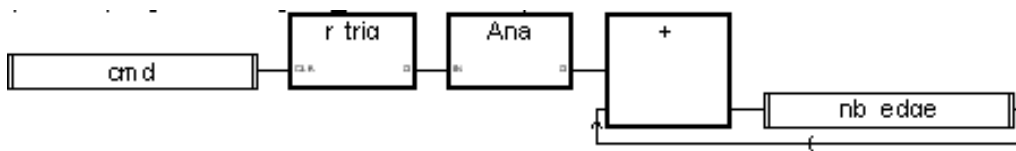
CLK Boolean 任何布林變數

傳回值 :

Q Boolean TRUE : CLK 由 FALSE 上昇為 TRUE
FALSE : 其他所有狀況

範例 :

(* FBD 程式 *)



(* ST 相等式 : 假設 R_TRIG1 相當於 R_TRIG 功能方塊 *)

```
R_TRIG1(cmd);  
nb_edge := ANA(R_TRIG1.Q) + nb_edge;
```

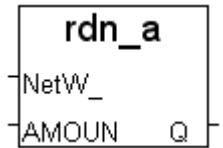
(* IL 相等式 : *)

```
LD cmd  
ST R_TRIG1.clk  
CAL R_TRIG1  
LD R_TRIG1.Q  
ANA  
ADD nb_edge  
ST nb_edge
```

下列各 RDN 函式皆為 Redundant 相關函式，詳細內容與範例 請參考 ISaGRAF 手冊第 20 章

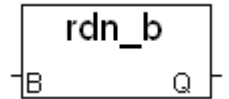
RDN_A

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
C_Function 設定 1 個變數陣列為 Redundant 同步資料



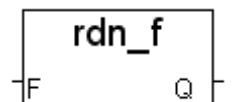
RDN_B

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
C_Function 設定 1 個 Boolean 變數為 Redundant 同步資料



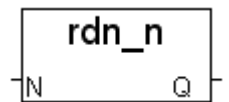
RDN_F

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
C_Function 設定 1 個 Real 變數為 Redundant 同步資料



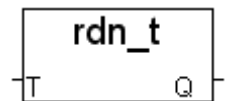
RDN_N

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
C_Function 設定 1 個 Integer 變數為 Redundant 同步資料



RDN_T

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7
C_Function 設定 1 個 Timer 變數為 Redundant 同步資料



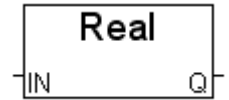
注意:

1. Redundant 系統的那 2 台 Wincon 內 Run 的 ISaGRAF project 是一模一樣的 project
2. 所有 I-7K 與 I-87K Remote IO 的功能塊必需放在 ISaGRAF project 的最上方
3. 使用以上的 RDN_function 之前，需在 IO connection 內連接“rdn”
4. RDN_B, RDN_F, RDN_N, RDN_T, RDN_A 必需使用在一開機後的第 1 個 PLC Scan 內
5. 最大可傳遞的同步資料量為 6000 byte. 每個 Boolean 佔 1 byte, 每個 Real & Integer & Timer 各佔 4 byte.
6. 更多的訊息請參考第 20 章

範例: Wincon CD-ROM : \napdos\isagraf\wincon\demo\ 內的 Wdemo_18a & Wdemo_18b

REAL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard_Function

轉換任何 1 個變數成爲 1 個實數

輸入參數 :

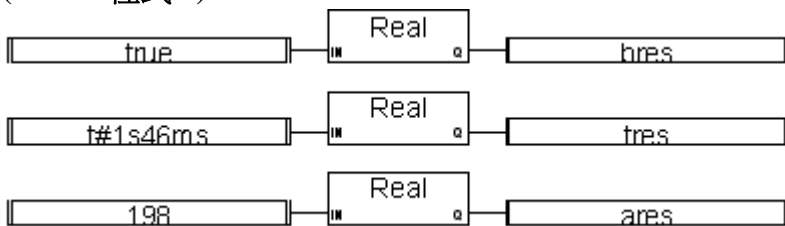
IN Any 任何一個非實數的類比值 (非訊息)

傳回值 :

Q Real 當 IN 爲 FALSE 傳回 0.0 / 當 IN 爲 TRUE 傳回 1.0
計時器型態則傳回毫秒的數據
類比整數則傳回等值的實數

範例 :

(* FBD 程式 *)



(* ST 相等式: *)

bres := REAL (true);

(* bres 爲 1.0 *)

tres := REAL (t#1s46ms);

(* tres 爲 1046.0 *)

ares := REAL (198);

(* ares 爲 198.0 *)

(* IL 相等式: *)

LD true

REAL

ST bres

LD t#1s46ms

REAL

ST tres

LD 198

REAL

ST ares

REA_STR2

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

轉換 Real 為 1 個字串, 小數點後的位數固定

輸入參數 :

REAL_	Real	要被轉換的 Real 值
DEC_	Integer	小數點後的位數, 0 ~ 5

傳回值 :

STR_	Message	轉換後的字串(最長為 13 個字元).
-------------	---------	---------------------

比如 DEC_ = 2

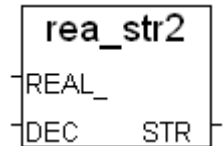
1.234 ---> '1.23'

123456.0 ---> '1.23456.00'

0.00001234 ---> '0.00'

注意: "STR_REAL" 可用來轉換字串為 Real 值.

範例: 請參考第 16 章 & demo_38, demo_39



REAL_INT

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

對應 1 個實數成為 1 個長整數

輸入參數 :

Real_	Real	要被對應的 real 值
--------------	------	--------------

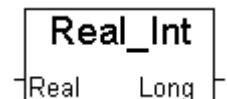
傳回值 :

Long_	Integer	對應完後的 integer 值
--------------	---------	-----------------

注意:

1. "Int_Real" 可用來對應 integer 成 real.

2. 如果只是要轉換一個實數為整數, 比如 -76.345 → -76, 請用 ANA()



REAL_STR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

轉換 Real 為 1 個字串

輸入參數 :

REAL_	Real	要被轉換的 Real 值
--------------	------	--------------

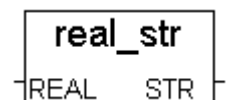
傳回值 :

STR_	Message	轉換後的字串(最長為 13 個字元).
-------------	---------	---------------------

1.234 ---> '1.234'

123456789.0 ---> '1.23457E+008'

0.00001234 ---> '1.234E-005'

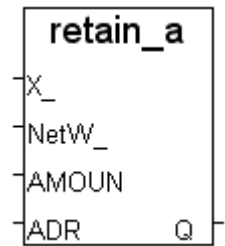


RETAIN_A

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定“變數陣列”為“可保留變數” - 以第 1 個元素的網路位址編號來設定



* 請參考第 10 章與第 2.6 節

* 各種 ISaGRAF 控制器適用此“可保留變數”的設定方式:

* Target 1: I-8xx7 + S-256 或 S-512 (保留變數), iP-8xx7, VP-2117

* Target 2: (µPAC-7186/I-7188EG) + X607 或 X608 (保留變數)

* Target 3: I-7188XG + X607 或 X608 (保留變數)

* Target 4: WP-8xx7/8xx6, XP-8xx7-CE6, VP-8xW7

輸入參數 :

X_ : Message 'B' 或 'b': 布林, 'N' 或 'n': 整數, 'F' 或 'f': 實數, 'T' 或 't': 計時器

NetW_ : Integer “變數陣列”第一個元素的網路位址編號.

Target 1/2/3 : 1~4095 , Target 4: 1~8191

Amount_ : Integer “變數陣列”的大小, 有效範圍: 1~255,

例如: CNT[0..15] 大小為 16, ABC[0..7] 大小為 8.

ADR_ : Integer 設定為變數要儲存的位址.

Target 1/2/3 : 布林和計時器為 1~256, 整數和實數為

1~1024 .

Target 4 : 布林和計時器為 1~1024 , 整數和實數為 1~4096 .

傳回值 :

Q_ : Boolean Ok 傳回 True, 錯誤則傳回 False (例如: 找不到保留記憶體)

注意 :

1. 使用 Retain_X, Retain_A, Retain_N, Retain_B, Retain_F 和 Retain_T 函式之前, 請確認 ISaGRAF IO 連結視窗中 "IO complex equipment" 欄位的下列各設備已正確連接.

a. I-7188EG/XG : "X607_608"

b. I-8417/8817/8437/8837 和 W-8037/8337/8737 : "S256_512"

2. 要在 ISaGRAF 3.4 (或 3.5) 版本中宣告 "變數陣列", 請在 ISaGRAF 子目錄

"C:\ISAWIN\EXE\" 下 "isa.ini" 檔案的最頂端加入 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗新增的 "DIM" 欄位內指定陣列個數. 請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行:

```
[DEBUG]
```

```
arrays=1
```

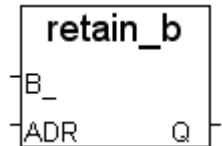
3. 使用 "Retain_A" 之前, 請在 ISaGRAF dictionary 視窗中 設定"變數陣列"的 "Network

Address" 欄位內 指定一個網路位址編號. 請務必不要勾選 "Retain" 欄位. 例如, 指定 整數 "變數陣列" CNT[0..7] 的 "Network Address".= 1A (16 進位制. 十進位制是 26) .

4. 請務必確認可保留記憶體 (7188EG/XG: X607/X608, I-8xx7 & W-8xx7: S-256/S-512) 正確成功的安插在控制器上.

RETAIN_B

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

設定布林變數為“可保留變數”

* 請參考第 10 章

* 各種 ISaGRAF 控制器適用此“可保留變數”的設定方式:

* Target 1: I-8xx7 + S-256 或 S-512 (保留變數), iP-8xx7, VP-2117

* Target 2: (µPAC-7186/I-7188EG) + X607 或 X608 (保留變數)

* Target 3: I-7188XG + X607 或 X608 (保留變數)

* Target 4: WP-8xx7/8xx6, XP-8xx7-CE6, VP-8xW7

輸入參數 :

B_ :	Boolean	B_ 必須為布林變數, 不能是常數.
ADR_ :	Integer	設定此布林變數的保留位址. Target 1 和 2 和 3 : 1 ~ 256; Target 4: 1 ~ 1024

傳回值 :

Q_ :	Boolean	Ok 傳回 True, 錯誤則傳回 False (例如 : 找不到保留記憶體)
-------------	---------	---

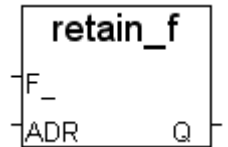
注意 :

1. 使用 Retain_X, Retain_A, Retain_N, Retain_B, Retain_F 和 Retain_T 函式之前, 請確認 ISaGRAF IO 連結視窗中 "IO complex equipment" 欄位的下列各設備已正確連接.
 - a. I-7188EG/XG : "X607_608"
 - b. I-8417/8817/8437/8837 和 W-8037/8337/8737 : "S256_512"
2. 要在 ISaGRAF 3.4 (或 3.5) 版本中宣告 "變數陣列", 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 下 "isa.ini" 檔案的最頂端加入 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗新增的 "DIM" 欄位內指定陣列個數. 請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行:

```
[DEBUG]
arrays=1
```
3. 使用 "Retain_A" 之前, 請在 ISaGRAF dictionary 視窗中 設定"變數陣列"的 "Network Address" 欄位內 指定一個網路位址編號. 請務必不要勾選 "Retain" 欄位. 例如, 指定 整數 "變數陣列" CNT[0..7] 的 "Network Address".= 1A (16 進位制. 十進位制是 26) .
4. 請務必確認可保留記憶體 (7188EG/XG: X607/X608, I-8xx7 & W-8xx7: S-256/S-512) 正確成功的安插在控制器上.

RETAIN_F

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

設定實數變數為“可保留變數”

* 請參考第 10 章

* 各種 ISaGRAF 控制器適用此“可保留變數”的設定方式:

* Target 1: I-8xx7 + S-256 或 S-512 (保留變數), iP-8xx7, VP-2117

* Target 2: (µPAC-7186/I-7188EG) + X607 或 X608 (保留變數)

* Target 3: I-7188XG + X607 或 X608 (保留變數)

* Target 4: WP-8xx7/8xx6, XP-8xx7-CE6, VP-8xW7

輸入參數 :

F_ :	Real	F_ 必須為實數變數, 不能是常數.
ADR_ :	Integer	設定此實數變數的保留位址. Target 1 和 2 和 3 : 1 ~ 1024; Target 4: 1 ~ 4096

傳回值 :

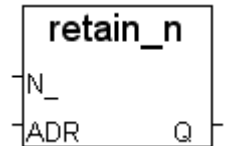
Q_ :	Boolean	Ok 傳回 True, 錯誤則傳回 False (例如 : 找不到保留記憶體)
-------------	---------	---

注意 :

1. 使用 Retain_X, Retain_A, Retain_N, Retain_B, Retain_F 和 Retain_T 函式之前, 請確認 ISaGRAF IO 連結視窗中 "IO complex equipment" 欄位的下列各設備已正確連接.
 - a. I-7188EG/XG : "X607_608"
 - b. I-8417/8817/8437/8837 和 W-8037/8337/8737 : "S256_512"
2. 要在 ISaGRAF 3.4 (或 3.5) 版本中宣告 "變數陣列", 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 下 "isa.ini" 檔案的最頂端加入 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗新增的 "DIM" 欄位內指定陣列個數. 請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行:
[DEBUG]
arrays=1
3. 使用 "Retain_A" 之前, 請在 ISaGRAF dictionary 視窗中 設定"變數陣列"的 "Network Address" 欄位內 指定一個網路位址編號. 請務必不要勾選 "Retain" 欄位. 例如, 指定 整數 "變數陣列" CNT[0..7] 的 "Network Address".= 1A (16 進位制. 十進位制是 26) .
4. 請務必確認可保留記憶體 (7188EG/XG: X607/X608, I-8xx7 & W-8xx7: S-256/S-512) 正確成功的安插在控制器上.

RETAIN_N

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

設定整數變數為“可保留變數”

* 請參考第 10 章

* 各種 ISaGRAF 控制器適用此“可保留變數”的設定方式:

* Target 1: I-8xx7 + S-256 或 S-512 (保留變數), iP-8xx7, VP-2117

* Target 2: (µPAC-7186/I-7188EG) + X607 或 X608 (保留變數)

* Target 3: I-7188XG + X607 或 X608 (保留變數)

* Target 4: WP-8xx7/8xx6, XP-8xx7-CE6, VP-8xW7

輸入參數 :

N_ :	Integer	N_ 必須為整數變數, 不能是常數.
ADR_ :	Integer	設定此整數變數的保留位址. Target 1 和 2 和 3 : 1 ~ 1024; Target 4: 1 ~ 4096

傳回值 :

Q_ :	Boolean	Ok 傳回 True, 錯誤則傳回 False (例如 : 找不到保留記憶體)
-------------	---------	---

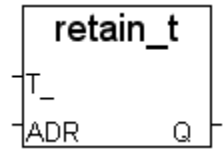
注意 :

1. 使用 Retain_X, Retain_A, Retain_N, Retain_B, Retain_F 和 Retain_T 函式之前, 請確認 ISaGRAF IO 連結視窗中 "IO complex equipment" 欄位的下列各設備已正確連接.
 - a. I-7188EG/XG : "X607_608"
 - b. I-8417/8817/8437/8837 和 W-8037/8337/8737 : "S256_512"
2. 要在 ISaGRAF 3.4 (或 3.5) 版本中宣告 "變數陣列", 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 下 "isa.ini" 檔案的最頂端加入 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗新增的 "DIM" 欄位內指定陣列個數. 請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行:

```
[DEBUG]
arrays=1
```
3. 使用 "Retain_A" 之前, 請在 ISaGRAF dictionary 視窗中 設定"變數陣列"的 "Network Address" 欄位內 指定一個網路位址編號. 請務必**不要**勾選 "Retain" 欄位. 例如, 指定 整數 "變數陣列" CNT[0..7] 的 "Network Address".= 1A (16 進位制. 十進位制是 26) .
4. 請務必確認可保留記憶體 (7188EG/XG: X607/X608, I-8xx7 & W-8xx7: S-256/S-512) 正確成功的安插在控制器上.

RETAIN_T

■ I-8xx7 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

設定計時器變數為“可保留變數”

* 請參考第 10 章

* 各種 ISaGRAF 控制器適用此“可保留變數”的設定方式:

* Target 1: I-8xx7 + S-256 或 S-512 (保留變數), iP-8xx7, VP-2117

* Target 2: (µPAC-7186/I-7188EG) + X607 或 X608 (保留變數)

* Target 3: I-7188XG + X607 或 X608 (保留變數)

* Target 4: WP-8xx7/8xx6, XP-8xx7-CE6, VP-8xW7

輸入參數 :

T_ :	Timer	T_ 必須為計時器變數, 不能是常數.
ADR_ :	Integer	設定此計時器變數的保留位址. Target 1 和 2 和 3 : 1 ~ 256; Target 4: 1 ~ 1024

傳回值 :

Q_ :	Boolean	Ok 傳回 True, 錯誤則傳回 False (例如 : 找不到保留記憶體)
-------------	---------	---

注意 :

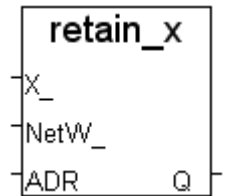
1. 使用 Retain_X, Retain_A, Retain_N, Retain_B, Retain_F 和 Retain_T 函式之前, 請確認 ISaGRAF IO 連結視窗中 "IO complex equipment" 欄位的下列各設備已正確連接.
 - a. I-7188EG/XG : "X607_608"
 - b. I-8417/8817/8437/8837 和 W-8037/8337/8737 : "S256_512"
2. 要在 ISaGRAF 3.4 (或 3.5) 版本中宣告 "變數陣列", 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 下 "isa.ini" 檔案的最頂端加入 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗新增的 "DIM" 欄位內指定陣列個數. 請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行:
[DEBUG]
arrays=1
3. 使用 "Retain_A" 之前, 請在 ISaGRAF dictionary 視窗中 設定"變數陣列"的 "Network Address" 欄位內 指定一個網路位址編號. 請務必**不要**勾選 "Retain" 欄位. 例如, 指定 整數 "變數陣列" CNT[0..7] 的 "Network Address".= 1A (16 進位制. 十進位制是 26) .
4. 請務必確認可保留記憶體 (7188EG/XG: X607/X608, I-8xx7 & W-8xx7: S-256/S-512) 正確成功的安插在控制器上.

RETAIN_X

■ I-8xx17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定一個變數為“可保留變數” – 以變數的網路位址編號來設定



* 請參考第 10 章

* 各種 ISaGRAF 控制器適用此“可保留變數”的設定方式:

* Target 1: I-8xx7 + S-256 或 S-512 (保留變數), iP-8xx7, VP-2117

* Target 2: (μPAC-7186/I-7188EG) + X607 或 X608 (保留變數)

* Target 3: I-7188XG + X607 或 X608 (保留變數)

* Target 4: WP-8xx7/8xx6, XP-8xx7-CE6, VP-8xW7

輸入參數 :

X_ : Message 'B' 或 'b': 布林, 'N' 或 'n': 整數, 'F' 或 'f': 實數, 'T' 或 't': 計時器
對應變數的網路位址編號.

NetW_ : Integer Target 1/2/3 : 1 ~ 4095; Target 4: 1 ~ 8191

ADR_ : Integer 設定為變數保留的位址..

Target 1/2/3 : 1 ~ 1024; Target 4: 1 ~ 4096

傳回值 :

Q : Boolean Ok 傳回 True, 錯誤則傳回 False (例如 : 找不到保留記憶體)

注意 :

1. 使用 Retain_X, Retain_A, Retain_N, Retain_B, Retain_F 和 Retain_T 函式之前, 請確認 ISaGRAF IO 連結視窗中 "IO complex equipment" 欄位的下列各設備已正確連接.

a. I-7188EG/XG : "X607_608"

b. I-8417/8817/8437/8837 和 W-8037/8337/8737 : "S256_512"

2. 要在 ISaGRAF 3.4 (或 3.5) 版本中宣告 "變數陣列", 請在 ISaGRAF 子目錄

"C:\ISAWIN\EXE\" 下 "isa.ini" 檔案的最頂端加入 2 行程式. 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 的宣告視窗新增的 "DIM" 欄位內指定陣列個數. 請在 c:\isawin\exe\isa.ini 檔案頂端, 加入下列 2 行 :

```
[DEBUG]
```

```
arrays=1
```

3. 使用 "Retain_A" 之前, 請在 ISaGRAF dictionary 視窗中 設定"變數陣列"的 "Network

Address" 欄位內 指定一個網路位址編號. 請務必不要勾選 "Retain" 欄位. 例如, 指定 整數 "變數陣列" CNT[0..7] 的 "Network Address".= 1A (16 進位制. 十進位制是 26) .

4. 請務必確認可保留記憶體 (7188EG/XG: X607/X608, I-8xx7 & W-8xx7: S-256/S-512) 正確成功的安插在控制器上.

S_B_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從電池保持式 SRAM 讀出 1 個 Boolean 值

輸入參數 :

ADR_ Integer 從那個位置讀出, 1 個 Boolean 佔有 1 個 Byte.

S256: 1 ~ 249,856 (1 ~ 16#3D000)

S512: 1 ~ 512,000 (1 ~ 16#7D000)

X607: 1 ~ 118,784 (1 ~ 16#1D000)

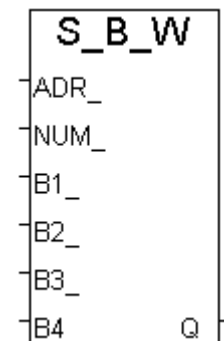
X608: 1 ~ 512,000 (1 ~ 16#7D000)

傳回值 :

BOO_ Boolean 讀到的 Boolean 值, 0=FALSE, 非 0 = TRUE

S_B_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 Boolean 值到電池保持式 SRAM

輸入參數 :

ADR_ : Integer 從那個位置開始寫, 1 個 Boolean 佔有 1 個 byte.

S256: 1 ~ 249,856 (1 ~ 16#3D000)

S512: 1 ~ 512,000 (1 ~ 16#7D000)

X607: 1 ~ 118,784 (1 ~ 16#1D000)

X608: 1 ~ 512,000 (1 ~ 16#7D000)

NUM_ : Integer 要寫幾個 Boolean, 0 ~ 4

B1_~B4_ : Boolean 要寫入的 Boolean 值

傳回值 :

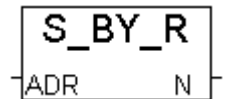
Q : Boolean 正確: TRUE, 失敗: FALSE
寫入的值為 FALSE=0, TRUE=1

範例 : demo_40, demo_41, demo_42, demo_44

* 請參考第 10.3 節.

S_BY_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從電池保持式 SRAM 讀出 1 個 Byte 值

輸入參數 :

ADR_ : Integer 從那個位置讀出, 1 個 Byte 佔有 1 個 Byte.

S256: 1 ~ 249,856 (1 ~ 16#3D000)

S512: 1 ~ 512,000 (1 ~ 16#7D000)

X607: 1 ~ 118,784 (1 ~ 16#1D000)

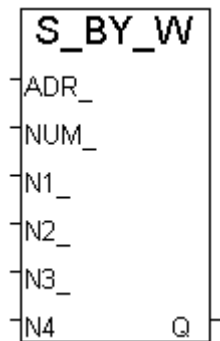
X608: 1 ~ 512,000 (1 ~ 16#7D000)

傳回值 :

N_ : Integer 讀到的 Byte 值, 0 ~ 255

S_BY_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 Byte 值到電池保持式 SRAM

輸入參數 :

ADR_ : Integer 從那個位置開始寫, 1 個 Byte 佔有 1 個 byte.

S256: 1 ~ 249,856 (1 ~ 16#3D000)

S512: 1 ~ 512,000 (1 ~ 16#7D000)

X607: 1 ~ 118,784 (1 ~ 16#1D000)

X608: 1 ~ 512,000 (1 ~ 16#7D000)

NUM_ : Integer 要寫幾個 Byte, 0 ~ 4

N1_~N4_ : Boolean 要寫入的 Byte 值, 0~255

傳回值 :

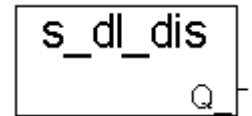
Q_ : Boolean 正確: TRUE, 失敗: FALSE

範例 : demo_40, demo_41, demo_42, demo_44

* 請參考第 10.3 節.

S_DL_DIS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

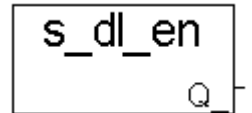
關閉下載的授權, 如此 PC 無法下載資料到 電池保持式 SRAM 內

傳回值 :

Q_ : Boolean 正確: TRUE, 失敗: FALSE

S_DL_EN

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

開啓下載的授權, 如此 PC 可以下載資料到 電池保持式 SRAM 內

傳回值 :

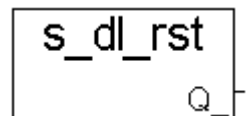
Q_ : Boolean 正確: TRUE, 失敗: FALSE

注意:

控制器開機後的預設值為“關閉”, 程式必須呼叫“S_DL_EN”, PC 才能下載資料進去 SRAM 內

S_DL_RST

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

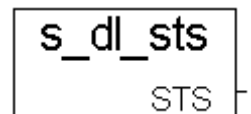
重置 SRAM 下載狀態 為“-1: 無動作”

傳回值 :

Q_ : Boolean 正確: TRUE, 失敗: FALSE

S_DL_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

取得 PC 對 SRAM 的下載狀態

傳回值 :

STS_ : Integer

- 1: 無動作
- 1: PC 正在下載資料到 SRAM
- 2: 下載完畢
- 3: PC 有要求下載資料, 但資料從未下載完成 (可能通訊出了問題)

可呼叫 S_DL_RST 將 下載狀態 設為 -1 (無動作)

範例 : demo_40, demo_41, demo_42, demo_44

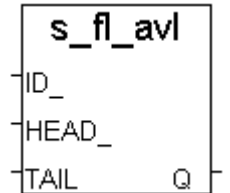
* 請參考第 10.3 節.

S_FL_AVL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

設定 SRAM 檔案的目前可取得資料的所在位置



傳入參數 :

ID_ :	Integer	檔案編號 (1 ~ 8)
HEAD_ :	Integer	起頭位置的 Byte 編號
TAIL_ :	Integer	結尾位置的 Byte 編號

(HEAD_, TAIL_) 必需坐落於 檔案配置區 內(請參考 "S_FL_INI"), 否則 Q_ 回傳 FALSE

S256: 1 - 249856 (1 - 16#3D000)

S512: 1 - 512000 (1 - 16#7D000)

X607: 1 - 118784 (1 - 16#1D000)

X608: 1 - 512000 (1 - 16#7D000)

例如 :

某 SRAM 檔案的 檔案配置區 為 1 ~ 20000 , 表示此檔最多可存 20000 個 byte.

1. 若任何一個 HEAD_ 及 TAIL_ 為 -1, 表示此檔目前無資料.
2. 若 HEAD_=1, TAIL_=1000, 表示此檔目前的資料 位於 1 ~ 1000. 共 1000 個 byte
3. 若 HEAD_=10001, TAIL_=5000, 表示此檔目前的資料 開始於 10001 ~ 20000, 緊接著由 1 ~ 5000, 共 15000 個 byte.
4. 若 HEAD_=1000, TAIL_=1000, 表示此檔目前無資料.

回傳值:

Q_ :	Boolean	TRUE: 正確 , FALSE: 失敗
-------------	---------	----------------------

注意: S_FL_INI 需在使用 S_FL_AVL 之前先被呼叫

範例 : demo_40, demo_41, demo_42, demo_44

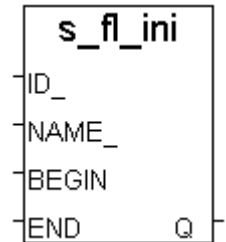
* 請參考第 10.3 節.

S_FL_INI

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

設定 SRAM 檔案的 檔案配置區



傳入參數：

ID_ :	Integer	檔案編號 (1 ~ 8)
NAME_ :	Message	檔案名稱, 名稱最多 8 個字+延伸檔名 3 個字. 例如, "data1.txt", "A1234567.bin". 合法的字為: A ~ Z, a ~ z, _, 0 ~ 9, 且第 1 個字需為 A ~ Z 或 a ~ z
BEGIN_ :	Integer	檔案配置區的起頭位置的Byte編號. BEGIN_ 需小於END_
END_ :	Integer	檔案配置區的結尾位置的Byte編號. BEGIN_ 需小於END_

S256: 1 ~ 249,856
S512: 1 ~ 512,000
X607: 1 ~ 118,784
X608: 1 ~ 512,000
例, BEGIN_=101, END_=5000 :

此 SRAM 檔案的檔案配置區位於 Byte 編號 101 ~ 5000

傳回值：

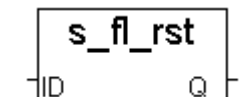
Q_ :	Boolean	TRUE: 正確, FALSE: 失敗
-------------	---------	---------------------

S_FL_RST

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

重置 SRAM 檔案狀態為 "PC 未上載資料過"



傳入參數：

ID_ :	Integer	檔案編號 (1 ~ 8)
--------------	---------	--------------

回傳值：

Q_ :	Boolean	TRUE: 正確, FALSE: 失敗
-------------	---------	---------------------

注意：

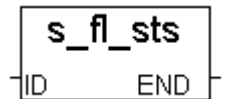
1. S_FL_INI 需先被呼叫過.
2. S_FL_RST 可用來重置 狀態 為 -1 ("PC 未上載資料過")

範例：demo_40, demo_41, demo_42, demo_44

* 請參考第 10.3 節.

S_FL_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

取得 SRAM 檔案的狀態, PC 上載資料的結尾 byte 編號

傳入參數：

ID_ : Integer 檔案編號 (1 ~ 8)

回傳值：

END_ : Integer PC 上載資料的結尾 Byte 編號
若 PC 未上載資料過： -1
S256: 1 ~ 249,856
S512: 1 ~ 512,000
X607: 1 ~ 118,784
X608: 1 ~ 512,000

例如,

某 SRAM 檔案的檔案配置區為 1~20000, 而此檔案目前可取得資料位於 1001~10000

1. 若 END_ 為 -1, 表示 PC 未上載資料
2. 若 END_ 為 10000, (通常此值會等於檔案目前可取得資料的結尾位置的 Byte 編號), 表示 PC 已經上載取得 1001 ~ 10000 內的資料
3. 若 END_ 為 8000, 表示 PC 已經上載取得 1001 ~ 8000 內的資料

注意：

1. S_FL_INI 需先被呼叫過.
2. S_FL_RST 可用來重置 狀態 為 -1 ("PC 未上載資料過")

範例：demo_40, demo_41, demo_42, demo_44

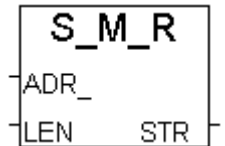
* 請參考第 10.3 節.

S_M_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

從電池保持式 SRAM 讀出 1 個字串



傳入參數 :

ADDR_ : Integer 從那個位置讀出.
S256: 1 ~ 249,856 (1 ~ 16#3D000)
S512: 1 ~ 512,000 (1 ~ 16#7D000)
X607: 1 ~ 118,784 (1 ~ 16#1D000)
X608: 1 ~ 512,000 (1 ~ 16#7D000)

LEN_ : Integer 要讀取的字串的長度, 0 ~ 255

傳回值 :

STR_ : Message 讀到的字串

例如 : 若存於 SRAM 內的資料為 16#31, 16#32, 16#33, 16#34, 16#35, 0, 16#37, 16#38, ...

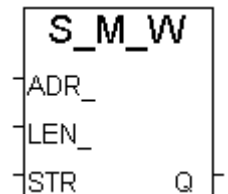
LEN_=0 ----> STR_ = " (空字串)
LEN_=3 ----> STR_ = '123'
LEN_=5 ----> STR_ = '12345'
LEN_=6 ----> STR_ = '12345'
LEN_=7 ----> STR_ = '12345'
LEN_=100 ----> STR_ = '12345'

S_M_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

寫入字串到電池保持式 SRAM



傳入參數 :

ADDR_ : Integer 從那個位置開始寫.
S256: 1 ~ 249,856 (1 ~ 16#3D000)
S512: 1 ~ 512,000 (1 ~ 16#7D000)
X607: 1 ~ 118,784 (1 ~ 16#1D000)
X608: 1 ~ 512,000 (1 ~ 16#7D000)

LEN_ : Integer 要寫的字串的長度, 0 ~ 255

STR_ : Message 要寫的字串.

例如 :

LEN_=0, STR='12345' ----> 沒有資料寫入
LEN_=1, STR='12345' ----> 16#31 (寫入 1 個 byte)
LEN_=3, STR='12345' ----> 16#31, 16#32, 16#33 (寫入 3 個 byte)
LEN_=7, STR='12345' ----> 16#31, 16#32, 16#33, 16#34, 16#35, 0, 0 (寫入 7 個 byte)
LEN_=100, STR='12345' --> 16#31, 16#32, 16#33, 16#34, 16#35, 0, 0, 0, ... (寫入 100 個

byte)

傳回值 :

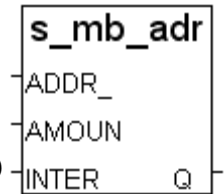
Q_ : Boolean 正確: TRUE, 失敗: FALSE

範例 : demo_40, demo_41, demo_42, demo_44

* 請參考第 10.3 節.

S_MB_ADR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

指定連續的或跳一號的 Modbus 網路位址編號給“變數陣列”(參閱 2.6.1 節)

控制器分類 :

Target 1: I-8417/8817/8437/8837, I-7188EG, I-7188XG

Target 2: Wincon-8xx7

傳入參數 :

ADDR_ : Integer Target 1: 1-4095, Target 2: 1-8191. 指定到 Dictionary 視窗 "變數陣列" 欄位的 Modbus 起始網路位址編號。

AMOUNT_ : Integer 有效範圍 : 1- 255, "變數陣列" 的大小, 例如: CNT[0..15] 大小為 16, ABC[0..7] 大小為 8.

INTERVAL_ : Integer 0 : 連續的 modbus 位址. (例如, 10, 11, 12, ...) 1 : 跳號的 modbus 位址. (例如, 10, 12, 14, ...)

傳回值 :

Q_ : Boolean True: Ok, False: ADDR_ 或 AMOUNT_ 或 INTERVAL_ 錯誤

注意:

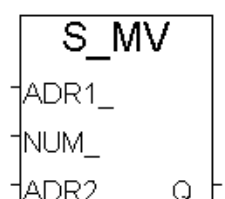
1. 此函式須在第一個 PLC scan 內呼叫, 第二次之後呼叫無效.
2. Modbus 起始網路位址編號必須在 ISaGRAF Dictionary 視窗 "變數陣列" 欄位裡指定. 例如: 指定起始網路位址編號 21 到 L[0..9] 的 "變數陣列".
3. 宣告 ISaGRAF 3.4 (或 3.5) 版本的 "變數陣列", 請在 ISaGRAF 子目錄 "C:\ISAWIN\EXE\" 下的 "isa.ini" 檔案頂端加入 2 行程式, 之後, 開啓 ISaGRAF 工作平台, 在 Dictionary 宣告視窗新增的 "DIM" 欄位指定. 請在 c:\isawin\exe\isa.ini 檔案頂端加入下列 2 行,

```
[DEBUG]
arrays=1
```

4. 當您在幫控制器程式除錯時, 要觀察或控制 "變數陣列" 的值, 請使用 "Spy list" 並插入陣列變數的名稱. 例如: 插入 L[0], L[1]. 請參考 ISaGRAF 進階使用手冊 第 9.12 節 "Spy list" 部份
5. 請務必小心宣告與使用 "變數陣列". 僅宣告需要的陣列大小, 宣告太大會無謂的佔據記憶體. 並注意不要使用大於您陣列大小的指標, 例如: 若您宣告變數陣列為 CNT[0..9], 千萬不可使用超過您宣告大小的指標 (如下列程式碼), 否則將導致程式執行錯誤. Ex:

```
For index := 0 to 10 do
  A := A + CNT[index] ;
End_For ;
(* 本範例 CNT[10] 並不存在 *)
```

6. 請小心指定 Modbus 網路位址, 不可與其他變數相衝突.
 7. 手冊: 有關控制器 冗於備援 與 變數陣列 的詳細資料, 請參考下列手冊或網址中的相關文章 \redundancy.pdf (冗於備援) 及 Variable Array (變數陣列).
1. Wincon-8xx7 CD-ROM: \napdos\isagraf\wincon\english_manu\
 2. ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/english_manu/



S_MV

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

在 SRAM 內複製資料

輸入參數：

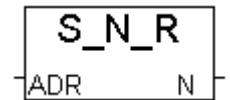
ADR1_	Integer	目的地的起頭位置 S256: 1 - 249856 (1 - 16#3D000) S512: 1 - 512000 (1 - 16#7D000) X607: 1 - 118784 (1 - 16#1D000) X608: 1 - 512000 (1 - 16#7D000)
NUM_	Integer	要複製多少個 byte, 0 - 512,000
ADR2_	Integer	從那個位置複製過來

傳回值：

Q_	Boolean	正確: TRUE, 失敗: FALSE
-----------	---------	---------------------

S_N_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從電池保持式 SRAM 讀出 1 個 Integer 值

輸入參數 :

ADR_	Integer	從那個位置讀出, 1 個 Integer 佔有 4 個 Byte. S256: 1 ~ 249,856 (1 ~ 16#3D000) S512: 1 ~ 512,000 (1 ~ 16#7D000) X607: 1 ~ 118,784 (1 ~ 16#1D000) X608: 1 ~ 512,000 (1 ~ 16#7D000)
------	---------	---

傳回值 :

N_	Integer	讀到的 Integer 值
----	---------	---------------

注意 : 儲存在電池保持式 SRAM 內的 Integer 格式為 [Lowest byte] [2nd byte] [3rd byte] [High byte],

例如 16#01020304, 會存入為 [04] [03] [02] [01]

S_N_W

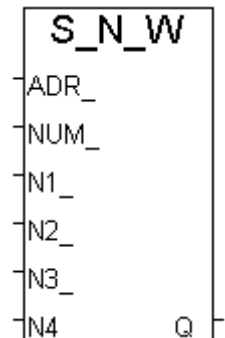
■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

寫入 Integer 值到電池保持式 SRAM

輸入參數 :

ADR_	Integer	從那個位置開始寫, 1 個 Integer 佔有 4 個 byte. S256: 1 ~ 249,856 (1 ~ 16#3D000) S512: 1 ~ 512,000 (1 ~ 16#7D000) X607: 1 ~ 118,784 (1 ~ 16#1D000) X608: 1 ~ 512,000 (1 ~ 16#7D000)
NUM_	Integer	要寫幾個 Integer, 0 ~ 4
N1_~N4_	Integer	要寫入的 Integer 值



注意 : 儲存在電池保持式 SRAM 內的 Integer 格式為 [Lowest byte] [2nd byte] [3rd byte] [High byte],

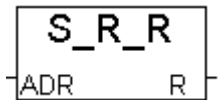
例如 16#01020304, 會存入為 [04] [03] [02] [01]

傳回值 :

Q_	Boolean	正確: TRUE, 失敗: FALSE
----	---------	---------------------

S_R_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從電池保持式 SRAM 讀出 1 個 Real 值

輸入參數 :

ADR_	Integer	讀那個位置, 1 個 Real 佔有 4 個 byte. S256: 1 ~ 249,856 (1 ~ 16#3D000) S512: 1 ~ 512,000 (1 ~ 16#7D000) X607: 1 ~ 118,784 (1 ~ 16#1D000) X608: 1 ~ 512,000 (1 ~ 16#7D000)
-------------	---------	--

傳回值 :

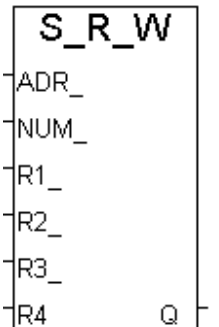
R_	Real	讀到的 Real 值
-----------	------	------------

注意 :

- 儲存在電池保持式 SRAM 內的 Real 格式為 [Lowest byte] [2nd byte] [3rd byte] [High byte], 例如: 1.23, 會存入為 16#A4, 16#70, 16#9D, 16#3F
- 若存放於電池保持式 SRAM 內的資料不是 Real 型態, 使用 S_R_R 讀出時, 有可能會產生 Local Fault 編號 102 (請參閱第 10.6 節)

S_R_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 Real 值到電池保持式 SRAM

輸入參數 :

ADR_	Integer	從那個位置開始寫, 1 個 Real 佔有 4 個 byte. S256: 1 ~ 249,856 (1 ~ 16#3D000) S512: 1 ~ 512,000 (1 ~ 16#7D000) X607: 1 ~ 118,784 (1 ~ 16#1D000) X608: 1 ~ 512,000 (1 ~ 16#7D000)
-------------	---------	---

NUM_	Integer	要寫幾個 Real, 0 ~ 4
-------------	---------	------------------

R1_~R4_	Real	要寫入的 Real 值
----------------	------	-------------

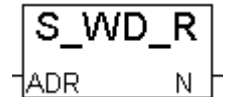
注意 : 儲存在電池保持式 SRAM 內的 Real 格式為 [Lowest byte] [2nd byte] [3rd byte] [High byte], 例如 1.23, 會存入為 16#A4, 16#70, 16#9D, 16#3F

傳回值 :

Q_	Boolean	正確: TRUE, 失敗: FALSE
-----------	---------	---------------------

S_WD_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從電池保持式 SRAM 讀出 1 個 Word 值

輸入參數 :

ADR_	Integer	從那個位置讀出, 1 個 Word 佔有 2 個 Byte. S256: 1 ~ 249,856 (1 ~ 16#3D000) S512: 1 ~ 512,000 (1 ~ 16#7D000) X607: 1 ~ 118,784 (1 ~ 16#1D000) X608: 1 ~ 512,000 (1 ~ 16#7D000)
-------------	---------	--

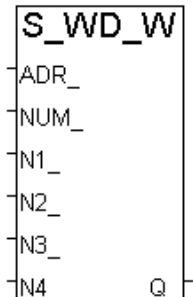
傳回值 :

N_	Integer	讀到的 Word 值, -32768 ~ +32767
-----------	---------	-----------------------------

注意 : 儲存在電池保持式 SRAM 內的 Word 格式為 [Low byte] [High byte],
例如 16#0102, 會存入為 [02] [01]

S_WD_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

寫入 Word 值到電池保持式 SRAM

輸入參數 :

ADR_	Integer	從那個位置開始寫, 1 個 Word 佔有 2 個 byte. S256: 1 ~ 249,856 (1 ~ 16#3D000) S512: 1 ~ 512,000 (1 ~ 16#7D000) X607: 1 ~ 118,784 (1 ~ 16#1D000) X608: 1 ~ 512,000 (1 ~ 16#7D000)
-------------	---------	---

NUM_	Integer	要寫幾個 Word, 0 ~ 4
-------------	---------	------------------

N1_~N4_	Integer	要寫入的 Word 值, -32768 ~ 32767
----------------	---------	-----------------------------

注意 : 儲存在電池保持式 SRAM 內的寫入的 Word 格式為 [Low byte] [High byte],
例如 16#0102, 會存入為 [02] [01]

傳回值 :

Q_	Boolean	正確: TRUE, 失敗: FALSE
-----------	---------	---------------------

SET_LED

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

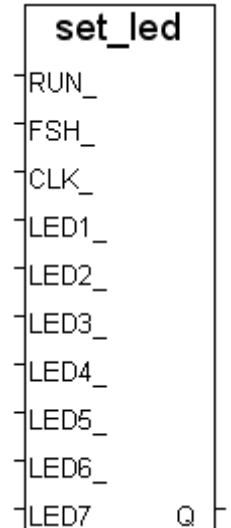
顯示訊息到 S-MMI

輸入參數 :

RUN_	Boolean	設為 TRUE 才動作
FLASH_	Integer	相對應的位置設為 1 則閃爍。 例如, 設為 11 (000011), 則第 6 及第 7 的位置會閃爍. 若設為 100001 (0100001), 則第 2 及第 7 的位置會閃爍
CLK_	Timer	閃爍的週期時間
LED1_	Integer	位置 1 要顯示的符號
LED2_	Integer	位置 2 要顯示的符號
LED3_	Integer	位置 3 要顯示的符號
LED4_	Integer	位置 4 要顯示的符號
LED5_	Integer	位置 5 要顯示的符號
LED6_	Boolean	設為 TRUE 則顯示位置 6
LED7_	Boolean	設為 TRUE 則顯示位置 7

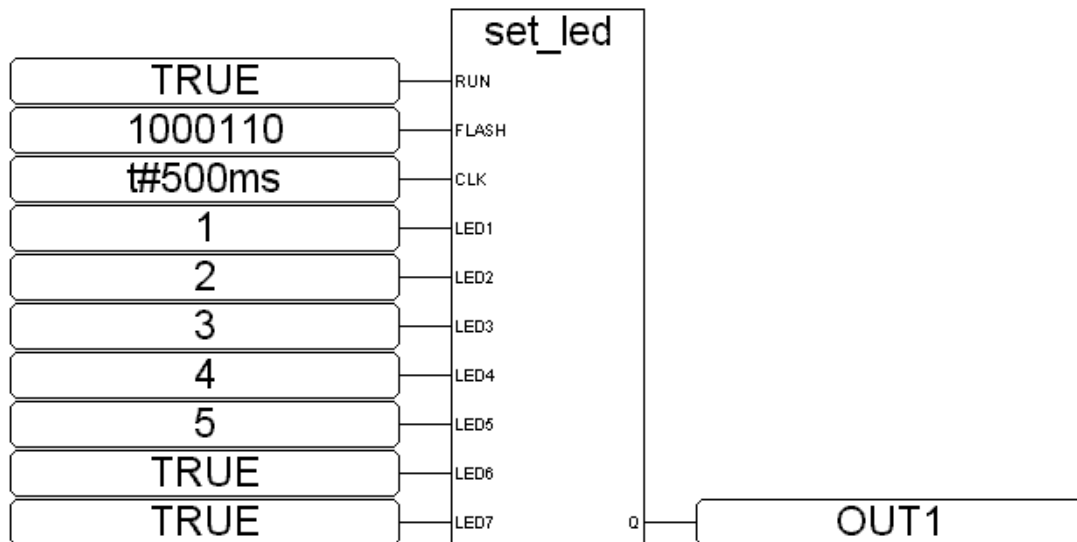
傳回值 :

Q_	Boolean	無作用
-----------	---------	-----



* 請參考附錄 A.3 來查閱符號表

範例:

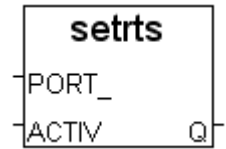


ST 相等式:

```
TMP := SET_LED(TRUE,1000110,t#500ms,1,2,3,4,5,TRUE,TRUE);
(* TMP 需宣告為 boolean 型態 *)
```

SETRTS

□ I-8x17 □ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

設定 連接埠的 RTS 訊號, 適用 COM 號 : 3 ~ 5

輸入參數 :

PORT_ : Integer 要設定的 COM 號. 3:COM3, 4:COM4, 5:COM5

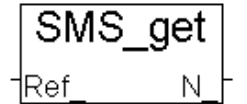
ACTIVE_ : Boolean TRUE : 設定 RTS 作用中, FALSE : 設定 RTS 不作用

傳回值 :

Q_ : Boolean TRUE : OK , FALSE : 失敗

SMS_GET

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

取得收到簡訊的日期與時間 (請參閱第 17 章)

輸入參數：

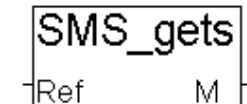
REF_	Integer	取得什麼？ 1 ~ 7
		1: 年, (N_ = 2000 ~ 2099)
		2: 月, (N_ = 1 ~ 12)
		3: 日, (N_ = 1 ~ 31)
		4: 星期幾, (N_ = 1 ~ 7, 星期天為 7)
		5: 時, (N_ = 0 ~ 23)
		6: 分, (N_ = 0 ~ 59)
		7: 秒, (N_ = 0 ~ 59)
		其他: 回傳 N_=-1

傳回值：

N_	Integer	回傳對應的資料, 若回傳-1, 可能是 沒有收到簡訊 或 REF_不在 1~7 的範圍內
-----------	---------	---

SMS_GETS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態：C_Function

取得簡訊內容及其它訊息 (請參閱第 17 章)

輸入參數：

REF_	Integer	取得什麼？ 1 ~ 3
		1: 簡訊內容
		2: 發送者的電話號碼
		3: 日期與時間 (字串格式)
		其它: 回傳 M_ = 'error'

傳回值：

M_	Message	回傳對應的資料, 若回傳 'error', 可能是沒有收到簡訊 或 REF_不在 1~3 的範圍內
-----------	---------	---

注意：

1. SMS_gets & SMS_get 可用來取得簡訊的相關資料
2. 當呼叫過 SMS_gets(1) 之後 (取得簡訊內容), 簡訊的 buffer 會清除為 "無簡訊". 因此請在使用 SMS_gets(1)之前先將其它需要的資料先讀出, 如 SMS_get(1~7) 及 SMS_gets(2) & SMS_gets(3)

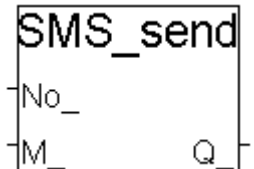
範例: demo_43, demo_43a

SMS_SEND

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

驅動 工控器 發送 1 封簡訊 (請參閱第 17 章)



輸入參數 :

No_	Message	發送給誰, 例如 '+886920119135', 最大長度為 31 字
M_	Message	要發送的簡訊內容

傳回值 :

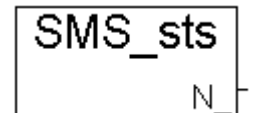
Q_	Boolean	True: 正確, False: 錯誤的號碼或發送失敗
----	---------	-----------------------------

SMS_STS

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

取得 簡訊發送狀態 (請參閱第 17 章)



傳回值 :

N_	Integer	狀態碼 0: 等待中, 尚未有發送命令進來 1: 忙碌中. (有 1 封簡訊正在發) 21: 簡訊發送成功 -1: SMS 系統錯誤 (請檢查 GSM Modem 與 SIM 卡) -2: Timeout 或 不明的原因
----	---------	---

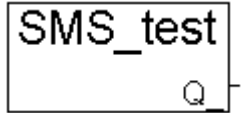
注意:

1. 請先使用 SMS_sts 取得“簡訊發送狀態”後,再使用 SMS_send. 只有在狀態不為”1: 忙碌中”時才有辦法發送
2. 呼叫 SMS_send 後會設定 "簡訊發送狀態" 為 "1: 忙碌中", 之後,隨著時間過去, 會設定成適當的值. 如 21 表示 簡訊發送成功, 或-1 或 -2 表示 有錯誤發生

範例: demo_43, demo_43a

SMS_TEST

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

測試有無收到簡訊 (請參閱第 17 章)

傳回值 :

Q_ Boolean TRUE: 有收到簡訊, FALSE: 無

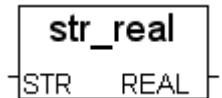
注意:

1. SMS_gets & SMS_get 可用來取得簡訊的相關資料
2. 當呼叫過 SMS_gets(1) 之後 (取得簡訊內容), 簡訊的 buffer 會清除為 "無簡訊". 因此請在使用 SMS_gets(1) 之前先將其它需要的資料先讀出, 如 SMS_get(1~7) 及 SMS_gets(2) & SMS_gets(3)

範例 : demo_43 , demo_43a

STR_REAL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

轉換字串為 1 個 Real 值

輸入參數 :

STR_ Message 要被轉換的字串, 例, '-0.2345', '+2.13E10', '15.2345E-2'

傳回值 :

REAL_ Real 轉換後的 Real 值. 若該值為 1.23E-20, 表示字串格式錯誤
例, 若 STR_='123.AB' 或 '23-45.17' 或 '1.2.345'
則 REAL_ 為 1.23E-20

注意:

"REAL_STR" 與 "REA_STR2" 可用來轉換 Real 值為字串.

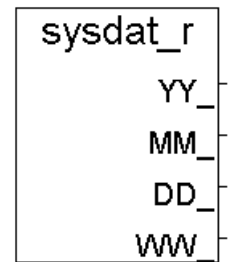
範例: 請參考第 16 章 & demo_38, demo_39

SYSDAT_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

讀取年, 月, 日 及 星期幾

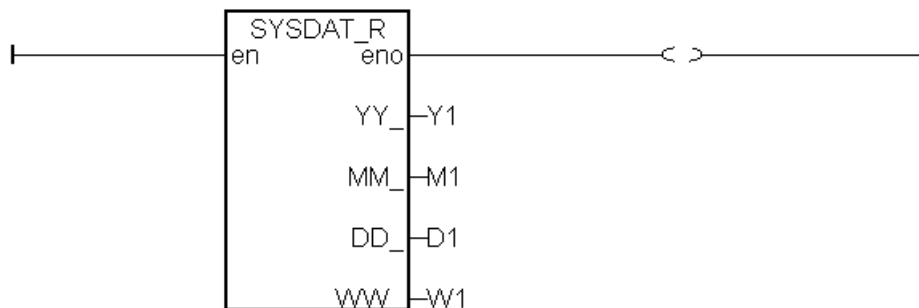


傳回值 :

YY_	Integer	年, 如 2002, 2003, 2010
MM_	Integer	月, 1 ~ 12
DD_	Integer	日, 1 ~ 31
WW_	Integer	星期幾, 1 ~ 7, 7 為星期日

範例: 請參考 demo_03.

Y1, M1, D1 及 W1 需宣告為 integer



ST 相等式:

```

DAT_R1();                (* 呼叫 DAT_R1 *)
Y1 := DAT_R1.YY_ ;      (* 取得年 *)
M1 := DAT_R1.MM_ ;     (* 取得月 *)
D1 := DAT_R1.DD_ ;     (* 取得日 *)
W1 := DAT_R1.WW_ ;     (* 取得星期幾 *)
(* DAT_R1 需宣告為 SYSDAT_R 的 FB instance *)
```

SYSDAT_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function Block

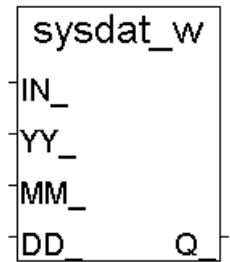
設定年,月,日

輸入參數：

IN_	Boolean	當由 FALSE 上升為 TRUE 時才設定一次
YY_	Integer	年, 如 2002, 2003, 2010
MM_	Integer	月, 1 ~ 12
DD_	Integer	日, 1 ~ 31

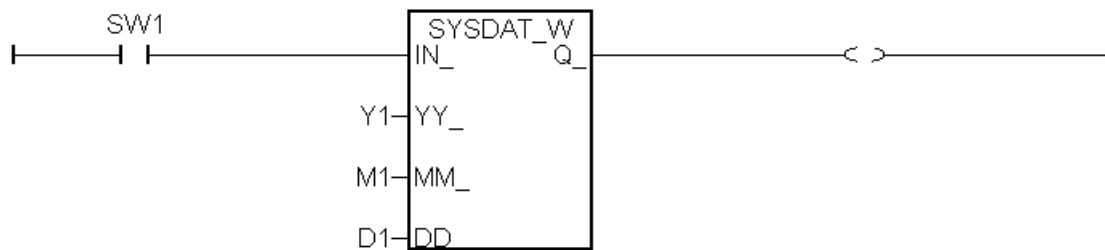
傳回值：

Q_	Boolean	正確回傳 TRUE
-----------	---------	-----------



範例：請參考 demo_03.

SW1 宣告為 boolean. Y1, M1, D1 宣告為 integer.



ST 相等式:

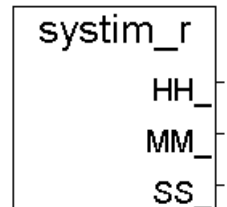
```
DAT_W1( SW1, Y1, M1, D1);    (* 呼叫 DAT_W1 *)  
OUT1 := DAT_W1.Q_ ;        (* 取得回傳值 Q_* )  
(* DAT_W1 需宣告為 SYSDAT_W 的 FB instance *)  
(* OUT1 宣告為 boolean *)
```


SYSTIM_R

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

讀取時,分及秒

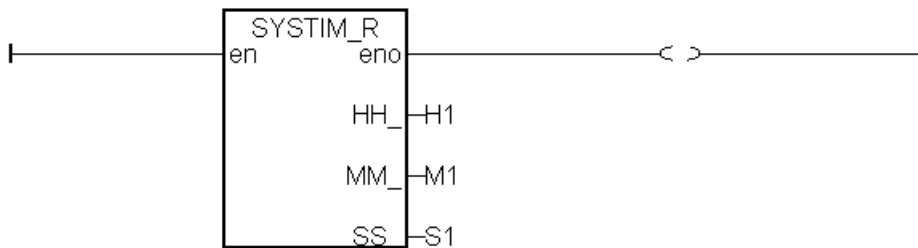


傳回值 :

HH_	Integer	時, 0 ~ 23
MM_	Integer	分, 0 ~ 59
SS_	Integer	秒, 0 ~ 59

範例: 請參考 demo_03 及 demo_15b.

H1, M1 及 S1 宣告為 integer



ST 相等式:

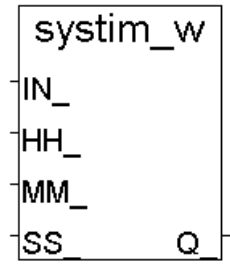
```
(* TIM_R1 宣告為 SYSTIM_R 的 FB instance *)
TIM_R1(); (* 呼叫 TIM_R1 *)
H1 := TIM_R1.HH_; (* 取得時 *)
M1 := TIM_R1.MM_; (* 取得分 *)
S1 := TIM_R1.SS_; (* 取得秒 *)
```

SYSTIM_W

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

設定時,分及秒



輸入參數 :

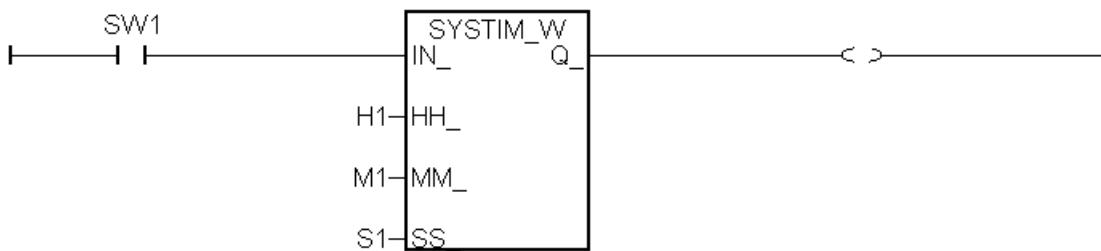
IN_	Boolean	當由 FALSE 上升為 TRUE 時才設定一次
HH_	Integer	時, 0 ~ 23
MM_	Integer	分, 0 ~ 59
SS_	Integer	秒, 0 ~ 59

傳回值 :

Q_	Boolean	正確回傳 TRUE
-----------	---------	-----------

範例: 請參考 demo_03

SW1 宣告為 boolean. H1, M1, S1 宣告為 integer.



ST 相等式:

```
TIM_W1( Sw1,2000,7,5);          (* 呼叫 TIM_W1 *)  
OUT1 := TIM_W1.Q_;            (* 取得回傳值 Q_ *)  
(* TIM_W1 宣告為 SYSTIM_W 的 FB instance *)  
(* OUT1 宣告為 boolean *)
```

TCP_RECV

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

TCP Client 端接收遠端 PC 或 TCP/IP Server 傳來的 message (透過乙太網路)
(請參閱第 19.3 節)

輸入參數 :

ID_ : Integer 對應的 "Tcp_Clie" 的連結編號, 可為 1 到 4. 對應的 "IP 位址" 和 "埠號" 請在 IO 連結設定視窗選 "設備" 的 "Tcp_clie" 來定義

傳回值 :

Msg_ : Message 接收的訊息. 若 Msg_ = " (空訊息), 表示沒有訊息進來.

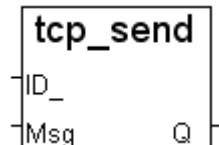
注意 :

1. 在使用 "tcp_recv" 和 "tcp_send" 之前, 請先連接 IO connection 視窗中的 "tcp_clie" (TCP_Client)
2. 接收的暫存器大小為 4096 bytes. 若暫存器滿了, 最先接收到的訊息會被清除, 把空間留給新接收的訊息.

範例 : Wdemo_32 和 Wdemo_33 (Wincon CD_ROM:\napdos\isagraf\wincon\demo\)

TCP_SEND

□ I-8x17 □ iP-8xx7 □ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

TCP Client 端傳送 message 到遠端 PC 或 TCP/IP Server (透過乙太網路)
(請參閱第 19.3 節)

輸入參數 :

ID_ : Integer 對應的 "Tcp_Clie" 的連結編號, 可為 1 到 4. 對應的 "IP 位址" 和 "埠號" 請在 IO 連結設定視窗選 "設備" 的 "Tcp_clie" 來定義

Msg_ : Message 傳送的訊息

傳回值 :

Q_ : Boolean True: 傳送 OK, False: 參數錯誤(如, ID_ 設成 8) 或 IO connection 視窗中對應連結的 "Tcp_clie" 沒有設定成功.

注意 :

1. 在使用 "tcp_send" 和 "tcp_recv" 之前, 請先連接 IO connection 視窗中的 "tcp_clie" (TCP_Client)
2. 傳送的暫存器大小為 4096 byte, 這表示一次 PLC scan 最多可傳送 4096 byte 到遠端 IP. 若暫存器滿了, 最先接收到的訊息會被清除, 把空間留給新的 "tcp_send()" 傳送的訊息.
3. 當傳送暫存器裡有訊息, 控制器只會在每次 PLC scan 傳送出一個訊息. 例如: 傳送暫存器裡有 100 個訊息, 則需要 100 次的 PLC scan 來傳送完這 100 個訊息. 然而, 如果把 "Tcp_clie" 的 "Send_Time_Gap" 設定為較大的值, 例如 100(ms), 則可以每隔 100 毫秒才傳送一個訊息.

範例: Wdemo_32 和 Wdemo_33 (Wincon CD_ROM:\napdos\isagraf\wincon\demo\)

TIME_STR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

轉換日期 & 時間 為字串格式

輸入參數 :

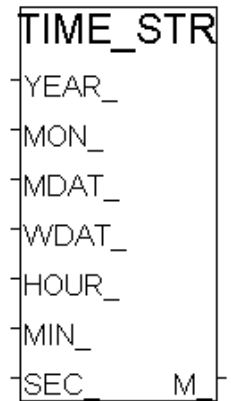
YEAR_ : Integer 年, 2000 ~
MON_ : Integer 月, 1 ~ 12
MDAY_ : Integer 日, 1 ~ 31
WDAY_ : Integer 星期幾, 1 ~ 7 (星期一 ~ 星期日)
HOUR_ : Integer 時, 0 ~ 23
MIN_ : Integer 分, 0 ~ 59
SEC_ : Integer 秒, 0 ~ 59

如果輸入參數有錯會回傳 M_ = " (空字串). 如 MON_=14

傳回值 :

M_ : Message 字串長度為 24, 如 'Feb/18/2003,13:25:45,Tue'

注意: 請使用 sysdat_r & systim_r 來取得控制器的日期 & 時間



TMR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : Standard Function

轉換任一變數為Timer格式

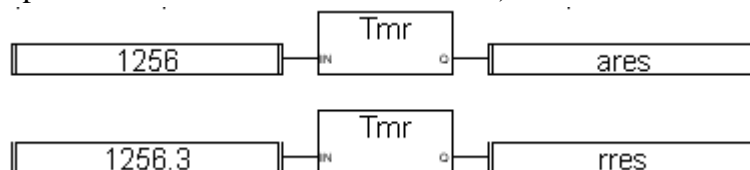
輸入參數 :

IN_ : Int-real 任何非計時器格式的值
 IN (若為實數, 則指整數部分) 單位為毫秒

傳回值 :

Q_ : Timer IN 的計時器格式值

範例: (* FBD example with "Convert to Timer" blocks *)

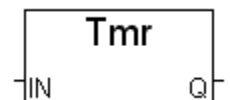


(* ST 相等式: *)

ares := TMR (1256); (* ares := t#1s256ms *)
 res := TMR (1256.3); (* rres := t#1s256ms *)

(* IL equivalence: *)

```
LD 1256
TMR
ST ares
LD 1256.3
TMR
ST rres
```

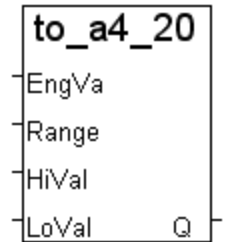


TO_A4_20

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

將使用者工程單位的值 ("實數" 格式) 轉換為 IO 板卡類比輸出訊號的值 (類比輸出訊號為 4 ~ 20mA, 整數格式)



輸入參數 :

- | | | |
|------------------|---------|---|
| EngVal_ : | Real | 要轉換的工程單位值. |
| Range_ : | Integer | 類比訊號輸出卡或模組的 Range 設定.
16#0 : 0 ~ 20 mA
16#1 : 4 ~ 20 mA
16#30 : 0 ~ 20 mA
16#31 : 4 ~ 20 mA |
| HiVal_ : | Real | 當類比輸出訊號為 20 mA 時, 使用者工程單位相對的高對應值 |
| LoVal_ : | Real | 當類比輸出訊號為 4 mA 時, 使用者工程單位相對的低對應值 |

EX: 將 0 - 100 psi 轉換成 I-8024 的 AO 值, 請設定

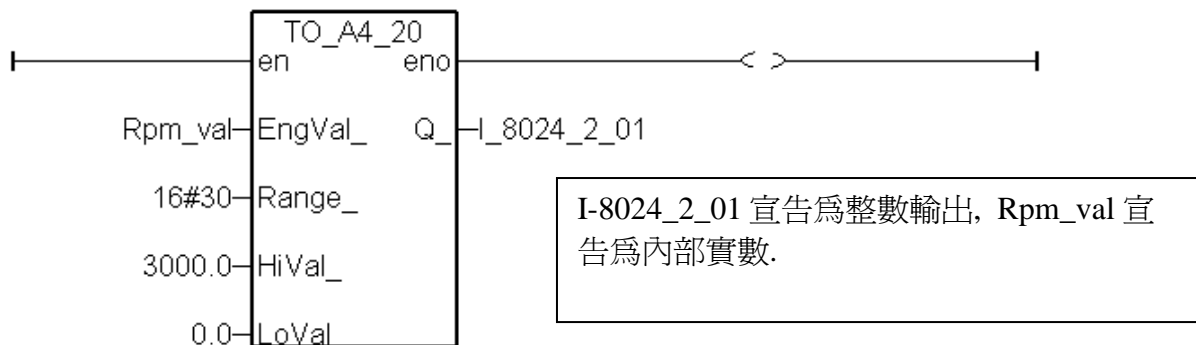
HiVal_ = 100.0, LoVal_ = 0.0 及 Range_ = 16#30 (請依據相關 IO 卡的 range 設定)

傳回值 :

- | | | |
|-------------|---------|--|
| Q_ : | Integer | 傳回轉換後的 AO 值
(通常在 0 ~ +32767 之間, 依據 IO 卡的 Range 設定而異).
若輸入不正確的 Range_ 或 (HiVal_ = LoVal_), 則回傳 -1 |
|-------------|---------|--|

範例 :

- 將 (0 ~ 100) psi 轉換為 (6554 ~ 32767) (若類比輸出卡的 range 設定為 16#30 : 0 - 20mA)
- 將 (0 ~ 100) psi 轉換為 (0 ~ 32767) (若類比輸出卡的 range 設定為 16#31 : 4 - 20mA)
- 將 (0 ~ 3000) rpm 轉換輸出到 I-8024 (range 設定為 30: 0 ~ 20 mA). 0 rpm 要輸出為 4 mA, 3000 rpm 輸出為 20 mA.



注意: 1. 類比輸出卡或模組的相關 range 需設定為 mA .

例如, (-20, +20mA), (0, 20mA), (4, 20mA) 等 mA

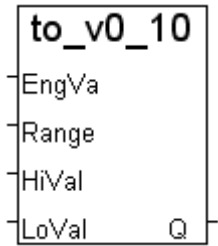
- 使用 A4_20_to, To_A4_20, To_V0_10, V0_10_to 等 function, 需將 driver 更新為 I-7188EG: 2.16 版, I-7188XG: 2.14 版, I-8xx7: 3.18 版, 或更高的版本, 程式才不會有問題 (較舊的 driver 會發生程式 run 一段時間後會停止的現象).

TO_V0_10

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

將使用者工程單位的值 ("實數" 格式) 轉換為 IO 板卡類比輸出訊號的值 (類比輸出訊號為 0 ~ 10 V, 整數格式)



輸入參數 :

EngVal_ :	Real	要被轉換的使用者工程單位值.
Range_ :	Integer	類比輸出卡或模組的 Range 設定. 16#2 : 0 ~ 10 V 16#32 : 0 ~ 10 V 16#33 : -10 ~ 10 V 16#34 : 0 ~ 5 V 16#35 : -5 ~ +5 V
HiVal_ :	Real	當類比輸出訊號為 10 V 時, 使用者工程單位相對的高對應值
LoVal_ :	Real	當類比輸出訊號為 0 V 時, 使用者工程單位相對的高對應值

Ex: 將 0 - 100 psi 轉換成 I-8024 的 AO 值, 請設定

HiVal_ = 100.0, LoVal_ = 0.0 及 Range_ = 16#33 (請依據相關 IO 卡的 range 設定)

傳回值 :

Q_ :	Integer	傳回轉換後的 AO 值 (通常在 0 ~ +32767 之間, 依據 IO 卡的 Range 設定而異). 若輸入不正確的 Range_ 或 (HiVal_ = LoVal_), 則回傳 -1
------	---------	--

範例 :

1. 將 (0 - 100) psi 轉換為 (0 - 32767) (若類比輸出卡的 range 設定為 16#32 : 0 ~ 10V)
2. 將 (0 - 100) psi 轉換為 (0 - 32767) (若類比輸出卡的 range 設定為 16#33 : -10 ~ +10V)

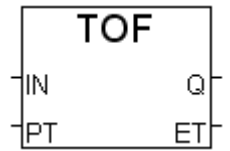
注意: 1. 類比輸出卡或模組的相關 range 需設定為 Voltage range.

例如, (0, 10 V), (-10, 10 V) 等

2. 使用 A4_20_to, To_A4_20, To_V0_10, V0_10_to 等 function, 需將 driver 更新為 I-7188EG: 2.16 版, I-7188XG: 2.14 版, I-8xx7: 3.18 版, 或更高的版本, 程式才不會有問題 (較舊的 driver 會發生程式 run 一段時間後會停止的現象).

TOF

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard Function

OFF-Delay (延遲關閉) 的控制.

輸入參數 :

IN : Boolean 控制 off-delay 的輸入, 若為上升, 則 Q 馬上輸出為 ON
若為下降, 則啟動 ET 的 off-delay 開始計時

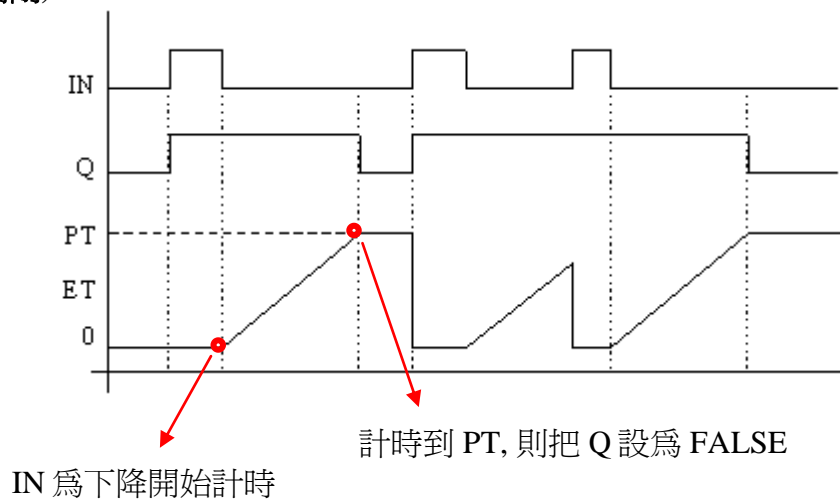
PT : Timer 設定的 delay (延遲) 時間,
當 ET 計時到 PT 時, 就將 Q 關閉(設為 False)

傳回值 :

Q : Boolean 輸出

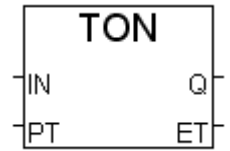
ET : Timer TOF 開始動作時 經過的時間

時序圖: (延遲關閉)



TON

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : Standard Function

ON-Delay (延遲啟動) 的控制.

輸入參數 :

IN : Boolean 控制 on-delay 的輸入,
若為上升, 則啟動 ET 的 on-delay 開始計時
若為下降, 則停止計時並重設 Q 為 False

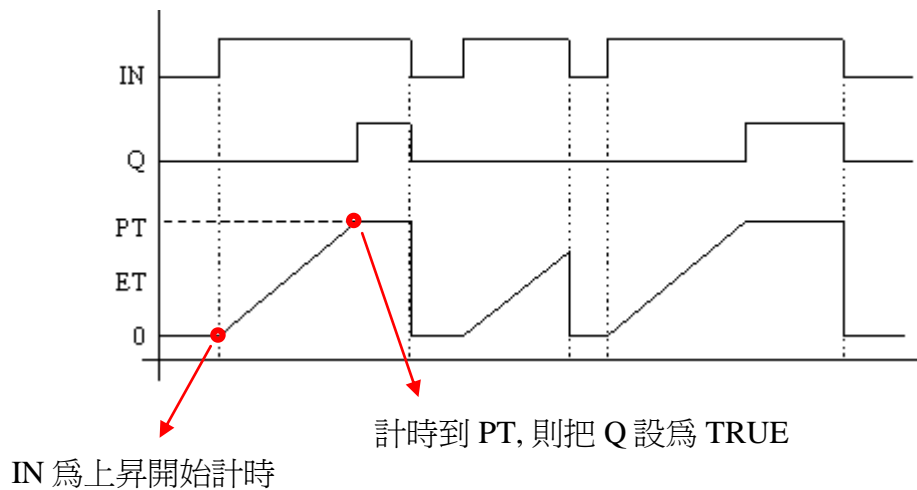
PT : Timer 設定的 delay (延遲) 時間

傳回值 :

Q : Boolean 輸出

ET : Timer TON 開始動作時 經過的時間

時序圖: (延遲啟動)

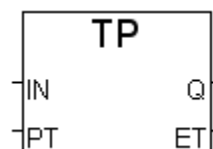


TP

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : Standard Function

輸出指定的一段時間為 ON (Pulse timer).



輸入參數 :

IN : Boolean 控制 TP 的輸入, 當 IN 上升, 則 Q 輸出為 ON, 並啟動 ET 計時, 若為下降, 則停止計時並重設計時器 (IN 的上升下降只在 Q 為 off 時有效, 且計時期間 IN 的改變無效)

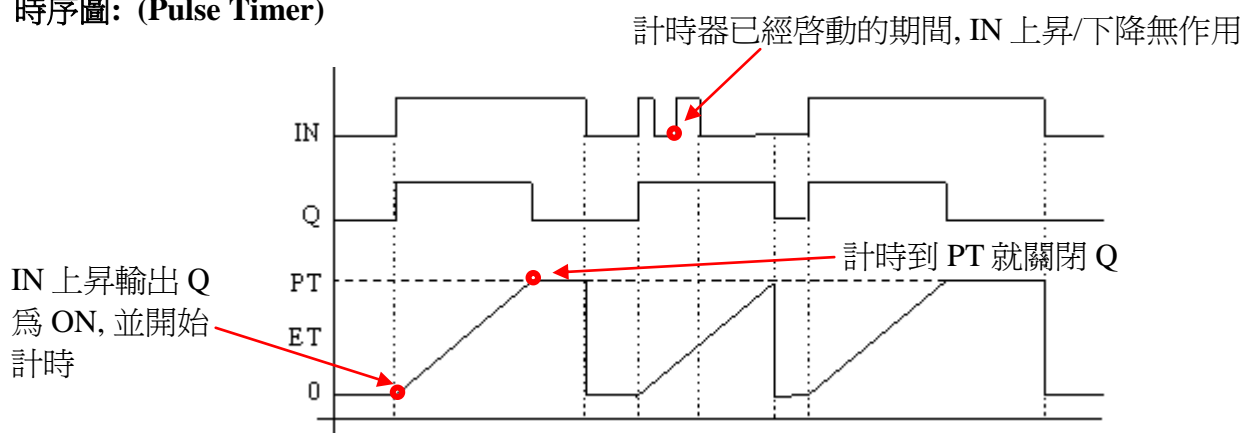
PT : Timer 設定 Q 要 ON 的時間長度

傳回值 :

Q : Boolean 輸出

ET : Timer TP 開始動作時 經過的時間

時序圖: (Pulse Timer)

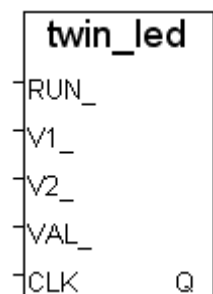


TWIN_LED

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG □ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 S-MMI 上顯示 2 個數值畫面.



輸入參數 :

RUN_ Boolean 設為 TRUE 才有動作

V1_ Integer 要顯示在第 1 個畫面左側 2 個位置的值, 0 ~ 99

V2_ Integer 要顯示在第 1 個畫面右側 2 個位置的值, 0 ~ 99

VAL_ Integer 要顯示在第 2 個畫面的值, -99999 ~ 99999

CLK_ Timer 畫面切換的週期時間

傳回值 :

Q_ Boolean 只傳回 TRUE

範例: 請參考 demo_10.

UDP_RECV

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

從遠端的 UDP/IP 連結 接收訊息(透過乙太網路). (請參考 19.2 節)

傳回值 :

Msg : Message 接收的訊息. 若 **Msg_ = "**(空訊息), 表示沒有訊息進來

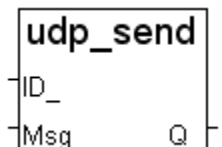
注意 :

1. W-8xx7 的 UDP 功能的接收的 buffer 大小為 8192 byte , 包含每個字串的結尾符號 1 byte 而 I-7188EG 與 I-8437/8837 的接收的 buffer 大小則為 2048 byte.
2. 如果一次進來太多個字串 導致 接收的 buffer 無法及時消化掉, 新進來的字串會取代掉 最早進來的那筆 字串.

範例 : Wdmo_19 和 Wdmo_19a (Wincon CD_ROM:\napdos\isagraf\wincon\demo\)

UDP_SEND

□ I-8x17 ■ iP-8xx7 ■ 718xEG □ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

傳送訊息到 UDP/IP 連結的遠端 PC 或控制器 (透過乙太網路). (請參考 19.2 節)

輸入參數 :

ID_ : Integer 對應的連結, 可為 1 到 4. 對應的 "IP 位址" 和 "埠號" 請在 "udp_ip" 裡設定

Msg_ : Message 要傳送的訊息

傳回值 :

Q_ :: Boolean True: 傳送 OK, False: 傳送 buffer 已滿或參數錯誤 (如, ID_ = 8).

注意 :

1. W-8xx7 的 UDP 功能的傳送的 buffer 大小為 2048 byte , 包含每個字串的結尾符號 1 byte 表示 每個 ISaGRAF PLC Scan 最多只能傳送 2048 個 byte .
2. 請不要很頻繁的 傳送大量資料出去, 跟 "udp_ip" 內的 "Send_Time_Gap" 參數有關, 若資料太多了, 消化不掉, 會積在傳送的 buffer 內, Buffer 若滿了, 使用 udp_send() 會傳回 False, 表示滿了, 無法再塞入更多的 Message. 每個 PLC Scan 對每個 UDP 連線, 只能消化 1 筆 Message (傳出 1 筆)

範例 : Wdmo_19 和 Wdmo_19a (Wincon CD_ROM:\napdos\isagraf\wincon\demo\)

V_BCD

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

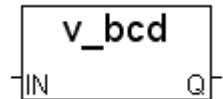
轉換十進位值為 BCD 值.

輸入參數 :

IN_ : Integer 要轉換的十進位值, 有效範圍 : 0 ~ 99999999

傳回值 :

Q : Integer BCD 值 ex: 12345 → 16#12345
16 → 22 (16#16)



V0_10_TO

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

將類比輸入訊號的值從 0 - 10V 單位轉換成使用者工程單位("實數" 格式).

如: 將 I-8017H 的類比輸入值轉換成 0 - 100 psi. 或 0 - 3000 rpm

輸入參數 :

AnaIn_ : Integer 整數變數, 與類比輸入卡或模組有關. 在 -32768 ~ +32767 之間, 依據 IO 卡的 range 設定而異.

Range_ : Integer 類比輸入 IO 卡或模組的 range 設定.

16#0 : -15mV ~ +15 mV
 16#1 : -50mV ~ +50 mV
 16#2 : -100mV ~ +100 mV
 16#3 : -500mV ~ +500 mV
 16#4 : -1 ~ +1 V
 16#5 : -2.5 ~ +2.5 V
 16#7 : -1.25 ~ +1.25 V
 16#8 : -10 ~ +10 V
 16#9 : -5 ~ +5 V
 16#A : -1 ~ +1 V
 16#B : -500mV ~ +500 mV
 16#C : -150mV ~ +150 mV

HiVal_ : Real 當類比輸入訊號為 10 V 時, 使用者工程單位相對的高對應值

LoVal_ : Real 當類比輸入訊號為 0 V 時, 使用者工程單位相對的低對應值

Ex: 將 I-8017H 的輸入訊號由 0 - 10 V 轉換成 0 - 100 psi, 請設定 HiVal_ = 100.0, LoVal_ = 0.0 及 Range_ = 16#5 或 16#7 或 16#8 或 16#9
(請依據相關 IO 卡的 range 設定)

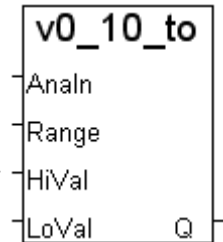
傳回值 :

Q : Real 轉換後使用者工程單位的值.
若輸入不正確的 Range_, 則回傳 1.23E-2

注意: 1. 類比輸入卡或模組的相關 range 需設定為 Voltage.

例如, (-10, +10V), (-5, +5V), (-1, 1 V), 等 Voltage range.

2. 使用 A4_20_to, To_A4_20, To_V0_10, V0_10_to 等 function, 需將 driver 更新為 I-7188EG: 2.16 版, I-7188XG: 2.14 版, I-8xx7: 3.18 版, 或更高的版本, 程式才不會有問題 (較舊的 driver 會發生程式 run 一段時間後會停止的現象).



VAL_HEX

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7



型態 : C_Function

轉換整數為 16 進位字串.

輸入參數 :

VAL_	Integer	要被轉換的值
DIGIT_	Integer	要轉換成幾個位數, 1 ~ 8, 超出範圍傳回 '' (空字串)

傳回值 :

HEX_	Message	轉換後的字串
-------------	---------	--------

範例:

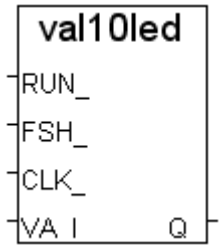
val_hex(100,3)	--->	'064'
val_hex(192,4)	--->	'00C0'
val_hex(4589,2)	--->	'ED' ('11ED', 位數為 2, 造成 '11' 被刪掉)
val_hex(4589,9)	--->	'' (位數 > 8, 傳回空字串)
val_hex(-2,8)	--->	'FFFFFFFE'

VAL10LED

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 S-MMI 上顯示 1 個 10 進位數值



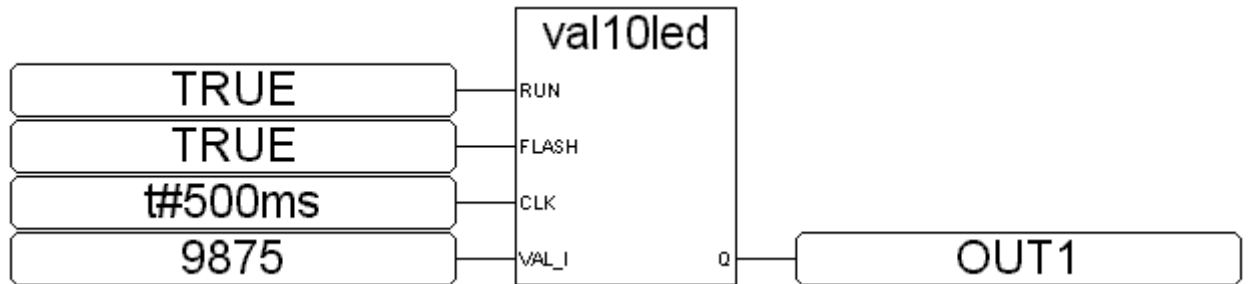
輸入參數 :

RUN_	Boolean	設為 TRUE 才動作
FLASH_	Boolean	設為 TRUE 則閃爍
CLK_	Timer	閃爍的週期時間
VAL_I_	Integer	要顯示的 integer, -9999 ~ +99999

傳回值 :

Q_	Boolean	只傳回 TRUE
-----------	---------	----------

範例: 請參考 demo_07 及 demo_11b



ST 相等式:

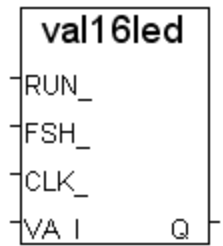
```
TMP := VAL10LED(TRUE,TRUE,t#500ms,9875);
(* TMP 宣告為 boolean *)
```

VAL16LED

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

在 S-MMI 上顯示 1 個 16 進位數值



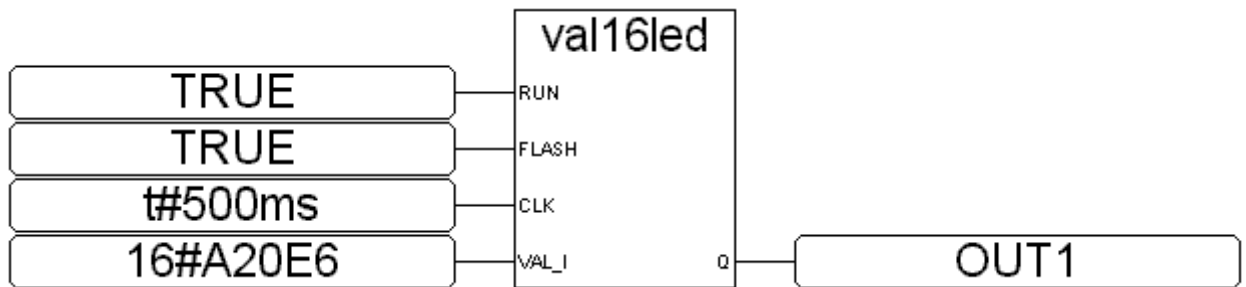
輸入參數 :

RUN_	Boolean	設為 TRUE 才動作
FLASH_	Boolean	設為 TRUE 則閃爍
CLK_	Timer	閃爍的週期時間
VAL_I_	Integer	要顯示的值, 16#0 ~ 16#FFFFFF

傳回值 :

Q_	Boolean	只傳回 TRUE
-----------	---------	----------

範例:



ST 相等式:

```
TMP := VAL10LED(TRUE,FALSE,t#500ms,16#A20E6);  
(*TMP 宣告為 boolean *)
```

W_MB_ADR

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

使用 Modbus 位址, 寫入值到布林或整數變數

輸入參數：

TYPE_ :	Integer	0: 布林變數, 1: 整數變數
ADR_ :	Integer	要寫入的 Modbus 位址, 有效範圍 Wincon: 1~8191, 其餘: 1~4095
DATA_ :	Integer	要寫入的整數 (或 布林, 0:False, 1:True)

傳回值：

Q_ :	Boolean	TRUE : 成功., FALSE : 失敗
------	---------	------------------------



注意：

1. “實數” 變數請使用 W_MB_REL function 來寫入.
2. 若指定的 Modbus 位址沒有定義對應的變數, 則不做寫入動作.
3. 若 TYPE_ 為整數, 而對應的變數是 "實數" 型態, 則寫入相對應的 32-bit. 最好使用 "W_MB_REL" 來寫入實數變數.
4. 若 TYPE_ 為整數, 而對應的變數是 "布林" 型態, 則不做寫入動作.
5. 若 TYPE_ 為布林, 而對應的變數不是 "布林" 型態, 則不做寫入動作.
6. 若長整數 (32-bit 整數) 要藉由 Modbus 通訊協定傳遞到 HMI, 需佔用 2 個 Modbus 位址編號. 詳細資料請參考 ISaGRAF 使用手冊第 4.2 章.

W_MB_REL

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態：C_Function

使用 Modbus 位址寫入值到實數變數

輸入參數：

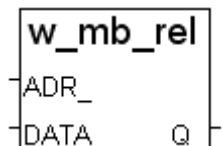
ADR_ :	Integer	要寫入的 Modbus 位址, 有效範圍 Wincon: 1~8191, 其餘: 1~4095
DATA_ :	Real	要寫入的實數

傳回值：

Q_ :	Boolean	TRUE : 成功., FALSE : 失敗
------	---------	------------------------

注意：

1. 請確認對應變數的型態為 "實數". 若為 "整數", 請使用 "W_MB_ADR" function.
2. 若對應的變數型態不是 "類比 Analog" (即實數或整數), 則不做寫入動作.
3. 若指定的 Modbus 位址沒有定義對應的變數, 則不做寫入動作.
4. 若實數藉由 Modbus 通訊協定傳遞到 HMI, 需佔用 2 個 Modbus 位址編號. 詳細資料請參考 ISaGRAF 使用手冊第 4.2 章.



WD_BIT

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function Block

轉換 1 個 word (signed 16-bit) 值成 16 boolean 值

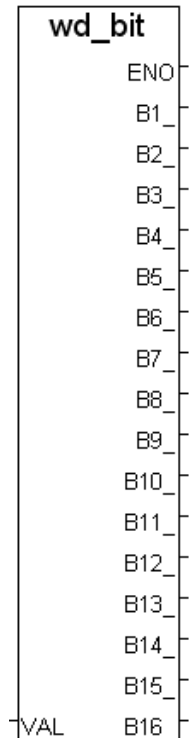
輸入參數 :

VAL_ Integer 要被轉換的 word (只有最低的 16-bit 有用)

傳回值 :

ENO_ Boolean 保留, 無作用.

B1_ ~ B16_ Boolean 轉換後的 16 個 boolean 值, 例如 VAL_ 為 4, 則 B3_ 為 TRUE, 其它為 FALSE.



WD_LONG

■ I-8x17 ■ iP-8xx7 ■ 718xEG ■ 7188XG ■ WP-8xx7, XP-8xx7-CE6, VP-2xW7

型態 : C_Function

合併 2 個 word (signed 16-bit) 為 1 個長整數 (signed 32-bit)

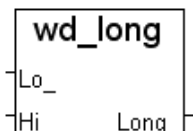
輸入參數 :

Lo_ Integer 要被轉換的 Low word (只有最低的 16-bit 有用)

Hi_ Integer 要被轉換的 High word (只有最低的 16-bit 有用)

傳回值 :

Long_ Integer 轉換後的 integer



範例:

Lo_	Hi_	---	Long_
-32768 (8000)	-1 (FFFF)	---	-32768 (FFFF 8000)
-1 (FFFF)	-1 (FFFF)	---	-1 (FFFF FFFF)
-32768 (8000)	0 (0000)	---	+32768 (0000 8000)
100 (0064)	4103 (1007)	---	+ 268 894 308 (1007 0064)

附錄 B：設定 I-8437/8837, iP-8x47, I-7188EG & μ PAC-7186EG 的 IP, Mask, Gateway

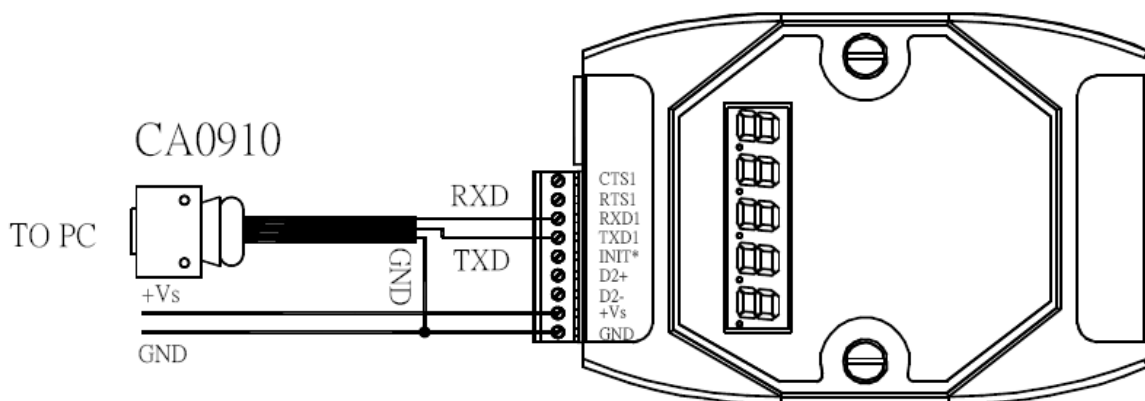
在這個章節裡,我們描述如何設定 I-8437 /8837 , iP-8x47, I-7188EG & μ PAC-7186EG 控制器的 IP, Mask 及 Gateway 位址. 有關 WP-8xx7, XP-8xx7-CE6 與 VP-25W7 控制器的設定方式, 請參考其“快速上手手冊”, 放置在

http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm .

每一台 I-8437/8837, iP-8x47, I-7188EG & μ PAC-7186EG 控制器可以使用通訊埠號碼 502 來與人機界面程式和 ISaGRAF Workbench 溝通。且最多有 4 台 PC 可以透過 Modbus TCP/IP 通訊協定與 I-8437 /8837, iP-8x47, I-7188EG & μ PAC-7186EG 控制器溝通。

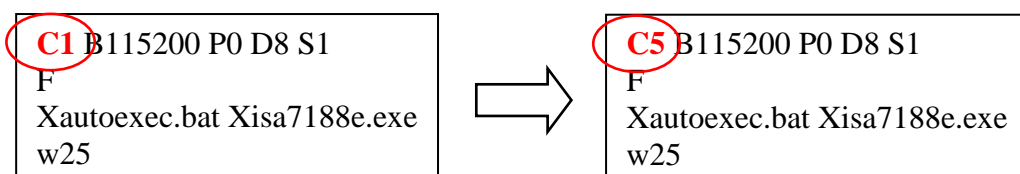
在 I-7188EG & μ PAC-7186EG 控制器的設定步驟:

1. 在您的 PC 硬碟建立一個名為 "7188" 的資料夾. 例如: "c:\7188".
2. 複製 CD 中的 \Napdos\ISaGRAF\7188EG\Driver\2.xx\7188xw.exe, 7188xw.ini, 檔案到您建立的“7188”資料夾:
3. 執行 "\7188\7188xw.exe" (Windows NT, Windows 2000 & Windows XP) 或執行
4. 藉由 RS232 傳輸線 (CA0910), 連接您 PC 的 COM1 或 COM2 到控制器的 COM1



若您的電腦沒有 COM1/COM2 或您要使用其他 COM 埠 (如 COM5)來連接 I-7188, 可以變更“7188xw.ini”檔案第一行的“C 編號”。

例如: 使用電腦的 COM5 連接 I-7188, 則更改 C1 為 C5, 如下



5. 關閉 I-7188 EG/XG 電源, 連接 "INIT" 和 "GND", 重新開啓電源.

6. 假如連接成功，畫面上將會出現“i7188E>”的訊息，如圖.
7. 輸入 "ip" 可查看目前的 IP 位址設定值
8. 輸入 "ip xxx.xxx.xxx.xxx" 可設定新的 IP 位址.
Ex: i7188E> ip 192.168.1.200
9. 輸入 "mask" 可查看目前位址的遮罩.
10. 輸入 "mask xxx.xxx.xxx.xxx" 可設定新的位址遮罩(mask).
Ex: i7188E> mask 255.255.255.0
11. 輸入 "gateway" 可查看目前的 gateway 位址.
i7188E> gateway
12. 輸入 "gateway xxx.xxx.xxx.xxx" 可設定新的 gateway 位址.
i7188E> gateway 192.168.1.1
13. 按 ALT_X 來離開 exit "7188xw" 視窗，否則 COM1/COM2 會一直被佔用住.
14. 將“INIT”-“GND”的連接線移開，重新啓動控制器.

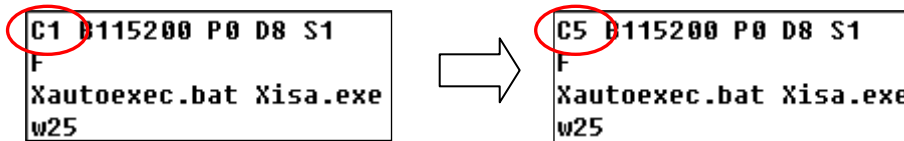
```
i7188E>ip
IP=192.168.255.1
i7188E>ip 192.168.1.200
Set IP=192.168.1.200
[ReadBack]IP=192.168.1.200
i7188E>mask
MASK=255.255.0.0
i7188E>mask 255.255.255.0
Set MASK=255.255.255.0
[ReadBack]MASK=255.255.255.0
i7188E>gateway
Gateway=192.168.0.1
i7188E>gateway 192.168.1.1
Set GATEWAY=192.168.1.1
[ReadBack]Gateway=192.168.1.1
i7188E>_
```

在 I-8437-80 / 8837-80 , iP-8x47 控制器的設定步驟:

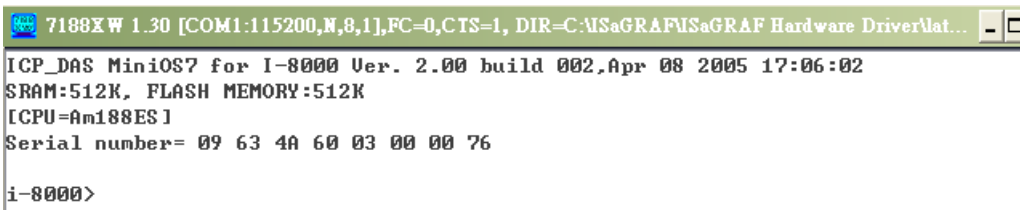
1. 在您的硬碟裡建立一個“8000”的檔案目錄. 例如: "c:\8000".
2. 從 CD_ROM 複製 \Napdos\ISaGRAF\8000\Driver\...\7188xw.exe, 7188xw.ini 到“8000”的目錄裡。
3. 執行\8000\7188xw.exe. 將會出現"7188xw " 視窗畫面。
4. 藉由 RS232 傳輸線，連接電腦上的 COM1 或 COM2 到 I-8437/8837 控制器的 COM1。

如果您想使用其他的 COM port(ex.COM5),請修正”7188xw.ini” 檔案第一行的“C 編號”。

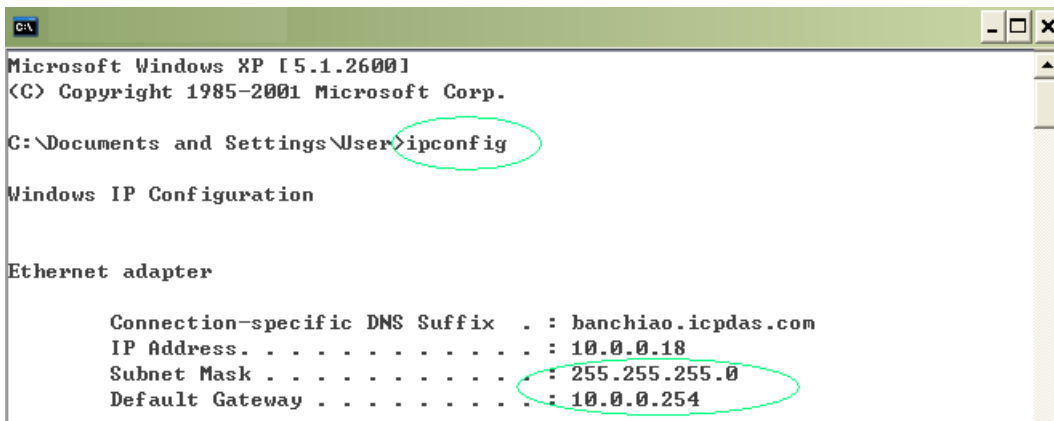
例如: 使用電腦的 COM5 連接, 則更改”7188xw.ini” C1 為 C5, 如下



5. 將 I-8437 / 8837 控制器的電源關掉，連接“INIT”和”INIT COM”，然後開起電源。
6. 假如連接成功，將會出現 7188xw 的畫面，如下圖所示。



7. 先使用電腦上的”命令提示字元”找出電腦的網路設定值。



★ 請根據電腦中的網路設定值,進行後續的設定. (IP/MASK/GATEWAY)

8. 在命令列輸入“ip”，可查看此 I-8437/8837 的 IP 位址。

輸入“ip 10.0.0.xxx”，可設定新 IP 位址。

```
7188x W 1.30 [COM1:115200,N,8,1],FC=0,CTS=1, DIR=C:\ISaGRAF\ISaGRAF Hardware Driver\lat...
7188x for WIN32 version 1.30 <2005/11/29> [By ICPDAS. Tim Tsai.]
[Begin Key Thread...]Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: autoexec.bat isa.exe
Current work directory="C:\ISaGRAF\ISaGRAF Hardware Driver\latest_8k\latest_8k\3
.16"
original baudrate = 115200!
now baudrate = 115200!

i-8000>ip
IP=10.0.0.123
i-8000>ip 10.0.0.123
Set IP=10.0.0.123
[ReadBack]IP=10.0.0.123
i-8000>
```

若是 iP-8x47 可以設定第 2 個 LAN 的 IP address, 可使用 ip2 來設定.
第 2 個 LAN 的 Mask 則是使用 Mask2
第 2 個 LAN 的 Gateway 則是使用 Gateway2

```
8000> ip
IP=192.168.255.255
8000>
```

9. 輸入“mask”可以查出目前 I-8437/8837 的網路遮罩(address mask)。

輸入”mask 255.255.255.0”設定新的網路遮罩。

```
i-8000>mask
MASK=255.255.255.0
i-8000>mask 255.255.255.0
Set MASK=255.255.255.0
[ReadBack]MASK=255.255.255.0
i-8000>
```

10. 輸入“gateway”可以查出目前的預設閘道(gateway)。

輸入“gateway 10.0.0.254”設定新的預設閘道。

```
i-8000>gateway
Gateway=10.0.0.254
i-8000>gateway 10.0.0.254
Set GATEWAY=10.0.0.254
[ReadBack]Gateway=10.0.0.254
i-8000>
```

11. 按 ALT_X 來離開 “7188x” 視窗, 否則 COM1 或 COM2 將會一直被佔用住。

12. 將 “INIT” –“INIT COM” 的连接線移開，重新開機 I-8437 /8837 控制器。

附錄 C : 更新 I-8417/8817/8437/8837 的驅動程式

ISaGRAF 的驅動程式屬於韌體，燒錄在 I-8417/8817/8437/8837, I-7188EG/XG, μ PAC-7186EG 及 Wincon-8xx7/8xx6 的 Flash 記憶體內。使用者可以很容易的自行更新驅動程式。

若要更新 Wincon-8xx7/8xx6, I-7188EG/XG & μ PAC-7186EG 的驅動程式, 更新方法請參考他們各自的“快速上手手冊”, 放置在 CD_ROM : `Napdos\ISaGRAF\` 下各自的子資料夾內 或 http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm .

我們最新版本的驅動程式放置在 (注意: 不同版本的檔案會是不同的檔名的 img 檔) <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 或 <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm> 下載後, 請解壓縮 (zip 檔)

警告: 這個 ISaGRAF 驅動程式的智慧財產權是屬於泓格科技股份有限公司。目前只有 I-8417,8817,8437,8837, I-7188EG/XG, μ PAC-7186EG 及 Wincon-8xx7/8xx6 已經註冊為合法的 ISaGRAF Target license, 若您燒錄此 ISaGRAF 驅動程式至其他的 I-8000 模組或其他相容產品是違法的。

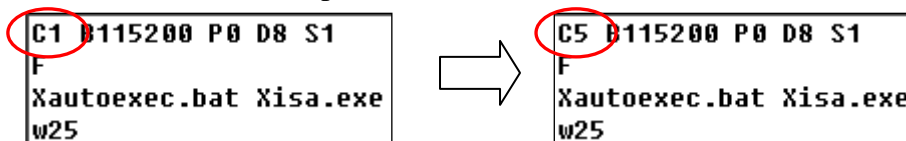
注意: 請先確定您目前的版本再更新驅動程式。

查看驅動程式版本 (此例,我們使用 3.16 版.)

1. 在您的硬碟裡建立一個“8000”的檔案目錄 (例如, "c:\8000")
2. 從 CD_ROM 複製 `Napdos\ISaGRAF\8000\Driver\40m\3.16\`
 - ①.7188xw.exe, ②.7188xw.f4, ③.7188xw.ini , ④.8k050408.img , ⑤.autoexec.bat , ⑥.isa.exe, ⑦.isa_data.exe 到“8000”的目錄裡。

(注意: 不同版本的檔案會是不同的檔名的 img 檔)

3. 執行 "c:\8000\7188xw.exe", 開啓 "7188xw" 視窗 (可按 F1 取得 Help).
4. 藉由 RS232 傳輸線, 連接電腦上的 COM1 或 COM2 到 I-8xx7 控制器的 COM1。如果您想使用其他的 COM port(ex.COM5), 請修正 "7188xw.ini". 的第 1 行。



5. 將 I-8xx7 控制器的電源關掉, 連接“INIT”和”INIT COM”, 然後開起電源。
6. 假如連接成功, 將會顯示”i-8000>”訊息於 7188xw 的畫面。

```
7188XW 1.30 [COM1:115200,N,8,1],FC=0,CTS=1, DIR=C:\ISaGRAF\ISaGRAF Hardware Driver\lat...
ICP_DAS MiniOS7 for I-8000 Ver. 2.00 build 002, Apr 08 2005 17:06:02
SRAM:512K, FLASH MEMORY:512K
[CPU=Am188ES]
Serial number= 09 63 4A 60 03 00 00 76
i-8000>
```

7. 輸入 "ver" 以查看控制器內的 OS 版本及日期.
8. 輸入 "isa *p=", 將會顯示目前安裝的驅動程式版本編號和日期

```

i-8000>ver
ICP_DAS MiniOS7 for I-8000 Ver. 2.00 build 002, Apr 08 2005 17:06:02
SRAM:512K, FLASH MEMORY:512K
[CPU=Am188ES]
Serial number= 09 63 4A 60 03 00 00 76

i-8000>isa *p=
Driver : I-8xx7 : isa.exe - 3.16, Oct.25,2006
MiniOS7 : Must use 8k050408.img
isa_data.exe - 1.8, Oct.25,2006
NED-ID : 1
COM1 is Modbus RTU slave port,19200,8,N,1
COM3 is Modbus RTU slave port,19200,8,N,1
Use 'isa *f=1' to free COM1, 'isa *f=0' to set COM1 as Modbus RTU

(C)Copyright:ICP DAS CO., LTD. Taiwan Id:84517297

```

更新 ISaGRAF 內嵌式驅動程式

9. 將 I-8xx7 控制器的電源關掉，連接“INIT”和”INIT COM”，然後開起電源。
10. 按 "F4" 將自動下載以下檔案並重新啓動系統. (約 60sec)
(isa_data.exe,autoexec.bat,isa.exe, 8k050408.img)

```

i-8000>del /y
Total File number is 2, do you really want to delete(y/n)?

i-8000>LOAD
File will save to 8000:0000
StartAddr-->7000:FFFF
Press ALT_E to download file!
Load file:isa_data.exe[crc=E70F,0000]
Send file info. total 287 blocks
Block 287
Transfer time is: 12.844000 seconds

```

⌚ 請等待約 60 秒讓 ISaGRAF 自動更新程式,期間請勿關閉電源!

```

i-8000>bios1
MiniOs7 for 8000 Ver 2.00.002, date=04/08/2005
Checking CRC-16...OK.
Update the OS code. Please wait the message <<Write Finished>>
Erase Flash [F000]
Write Flash
[FF]
<<Write Finished>>OK
Wait WDT reset system...
ICP_DAS MiniOS7 for I-8000 Ver. 2.00 build 002, Apr 08 2005 17:06:02
SRAM:512K, FLASH MEMORY:512K
[CPU=RDC 8820-D]
Serial number= 5A 5A 5A 5A 5A 5A 5A 5A

```

11. 輸入 " dir " 來確定 " autoexec.bat " 和 " isa.exe " 是否已下載成功

```
i-8000>DIR
0>autoexec.bat 05/21/2003 06:40:00 22[00016]8002:0000-8003:0006
1>isa.exe 10/25/2006 10:28:00 180678[2C1C6]8005:0006-AC21:000C
Total File number is 2 Free space=277956 bytes
```

12. 按 Alt_X 來離開 " 7188xw " 視窗
13. 將 "INIT" – "INIT COM" 的連接線移開，重新開機 I-8xx7 控制器。

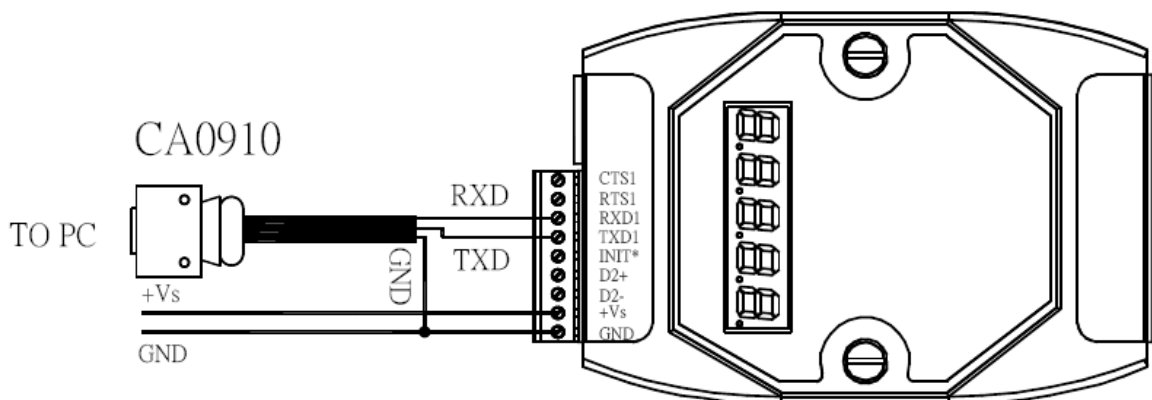
附錄 C.1: 設定 I-8xx7 & I-7188EG 的 COM1 為非 Modbus-Slave port

I-8417/8817/8437/8837, I-7188EG 及 μ PAC-7186EG 的 COM1 預設為支持 Modbus RTU Slave 通訊協議. 但是使用者可視需求關閉此功能, 將 COM1 變更為非 Modbus-Slave port, 如此便可利用 COM1 來作其他用途. 比如可用 "COMxxx" 等函式撰寫自訂的通訊協議. 或使用 COM1 來當 Modbus Master port.

注意: 若是 7188XG, COM1 固定支援 Modbus RTU Slave 協議, 不可更改.

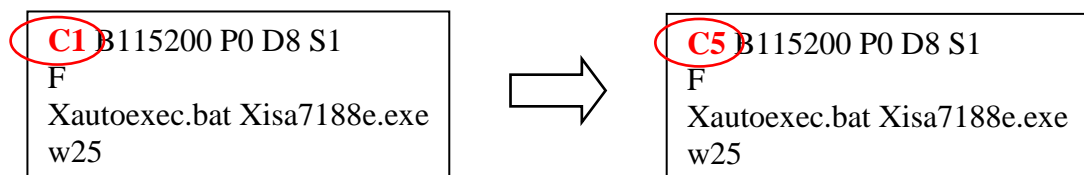
** 以下使用的控制器以 I-7188EG 為例: (I-8000 請參考括弧內說明)

- 步驟:**
1. 在您的 PC 硬碟建立一個名為 "7188" 的資料夾. 例如: "c:\7188". (I-8000 可用 "c:\8000")
 2. 複製 CD 中的 \Napdos\ISaGRAF\7188EG\Driver\2.xx\7188xw.exe, 7188xw.ini, 檔案到您建立的 "7188" 資料夾. (I-8000 請複製到 "8000" 資料夾)
 3. 執行該資料夾的 "7188xw.exe" (Windows NT, Windows 2000 & Windows XP 系統用此)
 4. 藉由 RS232 傳輸線 (如 CA0910), 連接您 PC 的 COM1 或 COM2 到控制器的 COM1



若您的電腦沒有 COM1/COM2 或您要使用其他 COM 埠 (如 COM5) 來連接控制器, 可以變更 "7188xw.ini" 檔案第一行的 "C 編號".

例如: 使用電腦的 COM5 連接控制器, 則更改 C1 為 C5, 如下



5. 關閉 I-7188 EG 電源, 連接 "INIT" 和 "GND", 重新開啓電源. (I-8000 則連接 "INIT" 和 "INIT COM")
6. 假如連接成功, 畫面上將會出現 "i7188E >" 的訊息, 如圖. (I-8000 則出現 "i-8000 >")
7. 輸入 "isa7188e *f=1" 來釋放控制器的 COM1 (即將 COM1 設定為非 Modbus-Slave 埠)


```
i7188E> isa7188e *f=1 (I-7188EG 使用)
i7188E> isa7186e *f=1 (μPAC-7186EG 使用)
i-8000> isa *f=1 (I-8000 使用此命令)
```

8. 按 ALT_X 來離開 "7188xw" 視窗, 否則 PC 的 COM1/COM2 會一直被佔用住.

9. 將 "INIT" –"GND" 的連接線移開, 重新啓動控制器.
(I-8000 則將 "INIT"–"INIT COM"的連接線移開)



```
i7188E>isa7188e *f=1
Com1 is free now
```

重要注意事項：

如果要恢復 COM1 爲 Modbus RTU Slave port, 請使用 "isa7188e *f=0" 指令, 如下:

```
Ex1: i7188E> isa7188e *f=0 (I-7188EG 使用)
Ex2: i7188E> isa7186e *f=0 (μPAC-7186EG 使用)
Ex3: i-8000> isa *f=0 (I-8000 使用此)
```

附錄 D : 類比 I/O 數值對照表

I-87013, I-7013, I-7033, I-7015, M-7015, M-7033, I-87015

Range Type Code (Hex)	RTD Type	Data Format	Max Value	Min Value
20 (Default)	Platinum 100 a = 0.00385 -100 ~ 100 °C	Temperature (攝氏度)	+100.0	-100.0
		Decimal Value	+32767	-32768
		2's complement HEX	7FFF	8000
		Ohms	+138.50	+060.60
21	Platinum 100 a = 0.00385 0 ~ 100 °C	Temperature (攝氏度)	+100.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000
		Ohms	+138.50	+100.00
22	Platinum 100 a = 0.00385 0 ~ 200 °C	Temperature (攝氏度)	+200.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000
		Ohms	+175.84	+100.00
23	Platinum 100 a = 0.00385 0 ~ 600 °C	Temperature (攝氏度)	+600.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000
		Ohms	+313.59	+100.00
24	Platinum 100 a = 0.003916 -100 ~ 100 °C	Temperature (攝氏度)	+100.0	-100.0
		Decimal Value	+32767	-32768
		2's complement HEX	7FFF	8000
		Ohms	+139.16	+060.60
25	Platinum 100 a = 0.003916 0 ~ 100 °C	Temperature (攝氏度)	+100.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000
		Ohms	+139.16	+100.00
26	Platinum 100 a = 0.003916 0 ~ 200 °C	Temperature (攝氏度)	+200.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000
		Ohms	+177.14	+100.00
27	Platinum 100 a = 0.003916 0 ~ 600 °C	Temperature (攝氏度)	+600.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000

Range Type Code (Hex)	RTD Type	Data Format	Max Value	Min Value
		Ohms	+317.28	+100.00
28	Nickel 120 -80 ~ 100 °C	Temperature (攝氏度)	+100.0	-80.0
		Decimal Value	+32767	-26214
		2's complement HEX	7FFF	999A
		Ohms	+200.64	+066.60
29	Nickel 120 0 ~ 100 °C	Temperature (攝氏度)	+100.0	+0.0
		Decimal Value	+32767	+0
		2's complement HEX	7FFF	0000
		Ohms	+200.64	+120.60
2A	Platinum 1000 a = 0.00385 -200 ~ 600 °C	Temperature (攝氏度)	+600.0	-200.0
		Decimal Value	+32767	-10922
		2's complement HEX	7FFF	D556
		Ohms	+3137.1	+0185.2
2B*	Cu 100 a = 0.00421 -20 ~ +150 °C	Temperature (攝氏度)	+150.0	-20.0
		Decimal Value	+32767	-4369
		2's complement HEX	7FFF	EEEE
		Ohms	+163.17	+091.56
2C*	Cu 100 a = 0.00427 0 ~ 200 °C	Temperature (攝氏度)	+200.0	0.0
		Decimal Value	+32767	0
		2's complement HEX	7FFF	0
		Ohms	+167.75	+090.34
2D*	Cu 1000 a = 0.00421 -20 ~ 150 °C	Temperature (攝氏度)	+150.0	-20.0
		Decimal Value	+32767	-4369
		2's complement HEX	7FFF	EEEE
		Ohms	+1631.7	+0915.6
2E	Platinum 100 a = 0.00385 -200 ~ 200 °C	Temperature (攝氏度)	+200.0	-200.0
		Decimal Value	+32767	-32768
		2's complement HEX	7FFF	8000
		Ohms	+175.84	+018.49
2F	Platinum 100 a = 0.003916 -200 ~ 200 °C	Temperature (攝氏度)	+200.0	-200.0
		Decimal Value	+32767	-32768
		2's complement HEX	7FFF	8000
		Ohms	+177.14	+017.14
80	Platinum 100 a = 0.00385 -200 ~ 600 °C	Temperature (攝氏度)	+600.0	-200.0
		Decimal Value	+32767	-10922
		2's complement HEX	7FFF	D556
		Ohms	+313.59	+018.49
81	Platinum 100 a = 0.003916	Temperature (攝氏度)	+600.0	-200.0
		Decimal Value	+32767	-10922

Range Type Code (Hex)	RTD Type	Data Format	Max Value	Min Value
	-200 ~ 600 °C	2's complement HEX	7FFF	D556
		Ohms	+317.28	+017.14

* Range Type Code 2B, 2C 與 2D 只有 I-7015, M-7015 及 I-87015 才有支援.

* I-87015, I-7015 與 M-7015 每點可設成不同的 Range Code.

I-8017HW(8/16-Ch), I-8017H(8-ch), I-8017HS(16-ch)

Range Type Code (Hex)	Data Format	Max value	Min value
05	Input Range	+2.5 V	-2.5 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
06*	Input Range	+20.0 mA	-20.0 mA
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
07	Input Range	+1.25 V	-1.25 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
08 (Default)	Input Range	+10.0 V	-10.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
09	Input Range	+5.0 V	-5.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000

* 每點可設成不同的 Range Code.

* 使用 Code 06 需外接 125Ω 電阻

I-87017W, I-87017, I-87017R, I-87017RW, I-7017, I-7017R, M-7017, M-7017R

Range Type Code (Hex)	Data Format	Max value	Min value
08 (Default)	Input Range	+10.0 V	-10.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
09	Input Range	+5.0 V	-5.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0A	Input Range	+1.0 V	-1.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0B	Input Range	+500.0 mV	-500.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0C	Input Range	+150.0 mV	-150.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0D*	Input Range (with 125 ohms resistor)	+20.0 mA	-20.0 mA
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000

* 使用 Code 0D 需外接 125Ω 電阻

I-87017ZW, I-87017DW

Range Type Code (Hex)	Data Format	Max value	Min value
08 (Default)	Input Range	+10.0 V	-10.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
09	Input Range	+5.0 V	-5.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0A	Input Range	+1.0 V	-1.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0B	Input Range	+500.0 mV	-500.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0C	Input Range	+150.0 mV	-150.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000

Range Type Code (Hex)	Data Format	Max value	Min value
7	Input Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	0
	2's Complement HEX	7FFF	0
D	Input Range	+20.0 mA	-20.0 mA
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
1A	Input Range	+20.0 mA	0 mA
	Decimal Value	+32767	0
	2's Complement HEX	7FFF	0

- * I-87017ZW 可用 Jumper 設定為 10-Ch. (differential input) 或 20-Ch. (single-end input). 設定為 differential input 時可以使用 Range type 為 7, D 與 1A, 但設為 20-Ch. 就不能使用 7, D 與 1A.
- * I-87017DW 可用 Jumper 設定為 8-Ch. (differential input) 或 16-Ch. (single-end input). 設定為 8-Ch. differential input 時可以使用 Range type 為 7, D 與 1A, 但設為 20-Ch. 就不能使用 7, D 與 1A.
- * 每個 I-87017ZW 與 I-87017DW 可每 Channel 規劃為不同的 Range type 值
- * I-87017ZW 量測電流時不需外接 125 ohm 電阻, 從板卡上的相對應 Channel 的 Jumper 來設定. (但 I-87017DW 量測電流時則需要外接 125 ohm 電阻)

I-7017RC, M-7017RC, I-87017RC

Range Type Code (Hex)	Data Format	Max value	Min value
7	Input Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	0
	2's Complement HEX	7FFF	0
D	Input Range	+20.0 mA	-20.0 mA
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
1A	Input Range	+20.0 mA	0 mA
	Decimal Value	+32767	0
	2's Complement HEX	7FFF	0

* 使用 I-7017RC, M-7017RC, I-87017RC 量電流不需外接電阻

I-87018Z, I-87018R, I-87018, I-87019R, I-7018Z, I-7018R, I-7018, M-7018, M-7018R, I-7019R, M-7019R (1)

Range Type Code (Hex)	Data Format	Max value	Min value
00	Input Range	-15.0 mV	-15.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
01	Input Range	+50.0 mV	-50.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
02	Input Range	+100.0 mV	-100.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
03	Input Range	+500.0 mV	-500.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
04	Input Range	+1.0 V	-1.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
05 (Default)	Input Range	+2.5V	-2.5V
	Decimal Value	+100.00	-100.00
	2's Complement HEX	7FFF	8000
06*	Input Range	+20.0 mA	-20.0 mA
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000

* 除了使用 I-7019, I-7019R, M-7019, M-7019R & I-87019R 量 Code 6 電流可用 jumper 來調整外, 其他 Module 量電流要外接 125Ω 電阻.

* 每塊 I-87018Z, I-7018Z 共有 10 個 Channel, 且每個 channel 可設為不同的 Type Code 值

● I-87018Z, I-7018Z 另外有支援

Range Type Code (Hex)	Data Format	Max value	Min value
7	Input Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	0
	2's Complement HEX	7FFF	0
1A	Input Range	+20.0 mA	0 mA
	Decimal Value	+32767	0
	2's Complement HEX	7FFF	0

I-87018Z, I-87018R, I-87018, I-87019R, I-7018Z, I-7018R, I-7018, M-7018, M-7018R, I-7019R, M-7019R (2)

Range Type Code (Hex)	Thermocouple Type	Data Format	Max Value	Min Value
0E	J Type -210 ~ 760 °C	Temperature (攝氏度)	+760.0	-210.0
		Decimal Value	+32767	-9054
		2's Complement HEX	7FFF	DCA2
0F	K Type -270 ~ 1372 °C	Temperature (攝氏度)	+1372.0	-270.0
		Decimal Value	+32767	-6448
		2's Complement HEX	7FFF	E6D0
10	T Type -270 ~ 400 °C	Temperature (攝氏度)	+400.0	-270.0
		Decimal Value	+32767	-22118
		2's Complement HEX	7FFF	A99A
11	E Type -270 ~ 1000 °C	Temperature (攝氏度)	+1000.0	-270.0
		Decimal Value	+32767	-8847
		2's Complement HEX	7FFF	DD71
12	R Type 0 ~ 1768 °C	Temperature (攝氏度)	+1768.0	+0.0
		Decimal Value	+32767	+0
		2's Complement HEX	7FFF	0000
13	S Type 0 ~ 1768 °C	Temperature (攝氏度)	+1768.0	+0.0
		Decimal Value	+32767	+0
		2's Complement HEX	7FFF	0000
14	B Type 0 ~ 1820 °C	Temperature (攝氏度)	+1820.0	+0.0
		Decimal Value	+32767	+0
		2's Complement HEX	7FFF	0000
15	N Type -270 ~ 1300 °C	Temperature (攝氏度)	+1300.0	-270.0
		Decimal Value	+32767	-6805
		2's Complement HEX	7FFF	E56B
16	C Type 0 ~ 2320 °C	Temperature (攝氏度)	+2320.0	+0.0
		Decimal Value	+32767	+0
		2's Complement HEX	7FFF	0000
17	L Type -200 ~ 800 °C	Temperature (攝氏度)	+800.0	-200.0
		Decimal Value	+32767	-8192
		2's Complement HEX	7FFF	E000
18	M Type -200 ~ 100 °C	Temperature (攝氏度)	+100.0	-200.0
		Decimal Value	+16384	-32768
		2's Complement HEX	4000	8000

- I-87018Z, I-7018Z 另外有支援

Range Type Code (Hex)	Thermocouple Type	Data Format	Max Value	Min Value
19	L Type DIN43710 -200 ~ 900°C	Temperature (攝氏度)	+900.0	-200.0
		Decimal Value	+32767	-7281
		2's Complement HEX	7FFF	E38F

- I-7019, I-7019R, M-7019, M-7019R & I-87019R 另外支持：

Range Type Code (Hex)	Data Format	Max value	Min value	
08 (Default)	Input Range	+10.0 V	-10.0 V	
	Decimal Value	+32767	-32768	
	2's Complement HEX	7FFF	8000	
09	Input Range	+5.0 V	-5.0 V	
	Decimal Value	+32767	-32768	
	2's Complement HEX	7FFF	8000	
0A	Input Range	+1.0 V	-1.0 V	
	Decimal Value	+32767	-32768	
	2's Complement HEX	7FFF	8000	
0B	Input Range	+500.0 mV	-500.0 mV	
	Decimal Value	+32767	-32768	
	2's Complement HEX	7FFF	8000	
0C	Input Range	+150.0 mV	-150.0 mV	
	Decimal Value	+32767	-32768	
	2's Complement HEX	7FFF	8000	
0D*	Input Range (with 125 ohms resistor)	+20.0 mA	-20.0 mA	
	Decimal Value	+32767	-32768	
	2's Complement HEX	7FFF	8000	
19	L Type DIN43710 -200 ~ 900°C	Temperature (攝氏度)	+900.0	-200.0
		Decimal Value	+32767	-7281
		2's Complement HEX	7FFF	E38F

* I-7019, I-7019R, M-7019, M-7019R & I-87019R 的 Range Code 可每 Channel 規劃為不同值

* I-7019, I-7019R, M-7019, M-7019R & I-87019R 使用 Code 0D 不需外接 125ohm 電阻, 只需用 Jumper 調整即可.

I-7021

Range Type Code (Hex)	Data Format	Max Value	Min Value
30	Output Range	+20.0 mA	+0.0 mA
	Decimal Value	+32767	+0
	2's complement HEX	7FFF	0000
31	Output Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	+0
	2's complement HEX	7FFF	0000
32 (Default)	Output Range	+10.0 V	+0.0 V
	Decimal Value	+32767	+0
	2's complement HEX	7FFF	0000

I-7022

Range Type Code (Hex)	Data Format	Max Value	Min Value
0	Output Range	+20.0 mA	+0.0 mA
	Decimal Value	+32767	+0
	2's complement HEX	7FFF	0000
1	Output Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	+0
	2's complement HEX	7FFF	0000
2 (Default)	Output Range	+10.0 V	+0.0 V
	Decimal Value	+32767	+0
	2's complement HEX	7FFF	0000

I-7005, M-7005, I-87005

Range Type Code (Hex)	Themistor Type	Data Format	Max Value	Min Value
60	PreCon Type III 10K @ 25 °C -35 ~ 115 °C	Temperature(華氏度)	+240.00 °F	-030.00 °F
		2's complement HEX	7FFF	E000
		Ohms	+000539.4	+173600.0
61	Fenwell U 2K @ 25 °C -50 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-050.00 °C
		2's complement HEX	7FFF	D556
		Ohms	+000037.2	+134020.0
62	Fenwell U 2K @ 25 °C 0 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	+000.00 °C
		2's complement HEX	7FFF	0000
		Ohms	+000037.2	+006530.0
63	YSI L Mix 100 @ 25 °C -80 ~ 100 °C	Temperature(攝氏度)	+100.00 °C	-080.00 °C
		2's complement HEX	7FFF	999A
		Ohms	+000014.3	+014470.0
64	YSI L Mix 300 @ 25 °C -80 ~ 100 °C	Temperature(攝氏度)	+100.00 °C	-080.00 °C
		2's complement HEX	7FFF	999A
		Ohms	+000035.8	+067660.0
65	YSI L Mix 1000 @ 25 °C -70 ~ 100 °C	Temperature(攝氏度)	+100.00 °C	-070.00 °C
		2's complement HEX	7FFF	A667
		Ohms	+000106.4	+132600.0
66	YSI B Mix 2252 @ 25 °C -50 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-050.00 °C
		2's complement HEX	7FFF	D556
		Ohms	+000041.8	+151000.0
67	YSI B Mix 3000 @ 25 °C -40 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-040.00 °C
		2's complement HEX	7FFF	DDDE
		Ohms	+000055.6	+101000.0
68	YSI B Mix 5000 @ 25 °C -40 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-040.00 °C
		2's complement HEX	7FFF	DDDE
		Ohms	+000092.7	+168300.0
69	YSI B Mix 6000 @ 25 °C -30 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-030.00 °C
		2's complement HEX	7FFF	E667
		Ohms	+000111.5	+106200.0
6A	YSI B Mix 10K @ 25 °C -30 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-030.00 °C
		2's complement HEX	7FFF	E667
		Ohms	+000185.9	+177000.0
6B	YSI H Mix 10K @ 25 °C	Temperature(攝氏度)	+150.00 °C	-030.00 °C
		2's complement HEX	7FFF	E667

Range Type Code (Hex)	Themistor Type	Data Format	Max Value	Min Value
	-30 ~ 150 °C	Ohms	+000237.0	+135200.0
6C	YSI H Mix 30K @ 25 °C -10 ~ 200 °C	Temperature(攝氏度)	+200.00 °C	-010.00 °C
		2's complement HEX	7FFF	F99A
		Ohms	+000186.7	+158000.0
70 ~ 77	User-defined -50 ~ 150 °C	Temperature(攝氏度)	+150.00 °C	-050.00 °C
		2's complement HEX	7FFF	D556
		Ohms	+000000.0	+000000.0

- * 使用者定義的 Type, 如果電阻大於 180000 ohms, 則被當作低於標準範圍.
- * 請參閱 I-7005/M-7005 User's Manual 第 1.11 節.

I-8024

Range Type Code (Hex)	Data Format	Max Value	Min Value
30	Output Range	+20.0 mA	+0.0 mA
	Decimal Value	+32767	+0
33	Output Range	+10.0 V	-10.0 V
	Decimal Value	+32767	-32768

* 每個 channel 可規劃成不同的 Range 值

I-87024, I-7024

Range Type Code (Hex)	Data Format	Max Value	Min Value
30	Output Range	+20.0 mA	+0.0 mA
	Decimal Value	+32767	+0
31	Output Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	+0
32	Output Range	+10.0 V	+0.0 V
	Decimal Value	+32767	+0
33 (Default)	Output Range	+10.0 V	-10.0 V
	Decimal Value	+32767	-32768
34	Output Range	+5.0 V	+0.0 V
	Decimal Value	+32767	+0
35	Output Range	+5.0 V	-5.0 V
	Decimal Value	+32767	-32768

I-87022, I-87026

Range Type Code (Hex)	Data Format	Max Value	Min Value
0	Output Range	+20.0 mA	+0.0 mA
	Decimal Value	+32767	+0
1	Output Range	+20.0 mA	+4.0 mA
	Decimal Value	+32767	+0
2	Output Range	+10.0 V	+0.0 V
	Decimal Value	+32767	0

* I-87022, I-87026 的每個 channel 可規劃成不同的 Range 值

附錄 E : ISaGRAF 語法參考

本節內容請參考 ISaGRAF 3.4x 或 3.5x 軟體版本 附的 CD-ROM 內的 ISaGRAF.pdf 檔案

附錄 F：如何 Enable/Disable WP-8xx7 的 LAN2

重要:

1. 建議使用 NS-205 或 NS-208 工業型乙太網轉換器。
2. ISaGRAF 應用時，LAN1 請務必使用固定 IP (若啓動 LAN2 也請使用固定 IP)。

WP-8xx7 的 LAN2 預設狀態為“停止”(disabled)，使用 LAN2 前需先設為“啓動”。

ISaGRAF 應用“Ebus” (請見 ISaGRAF 進階使用手冊 7.5 節) 與“新備援系統” (請見 www.icpdas.com > FAQ > Software > ISaGRAF > 093) 時，**必須**使用 WP-8xx7 的 LAN2；而 ISaGRAF 應用“以 UDP 或 TCP 傳送訊息”時也可以使用 LAN2 (請見 ISaGRAF 進階使用手冊 19.2 和 19.3 節)。

請點選 [Start] > [Setting] > [Control Panel]，開啓“Network and Dual-up Connections”，以滑鼠右鍵啓動 LAN2 為“Enable” (或設為停止“Disable”)。

再點選 [Start] > [Programs] > [WinPAC Utility]，選擇“Save and Reboot” 儲存設定值並重新啓動。

