如何使用 Wincon-8347 / 8747 或 uPAC-7186EG 或 iPAC-8437 / 8837 來連接一顆 i-7530 或多顆 i-7530, 來 讀取 或 控制 CAN 及 CANopen 設備 與 傳感器 ?

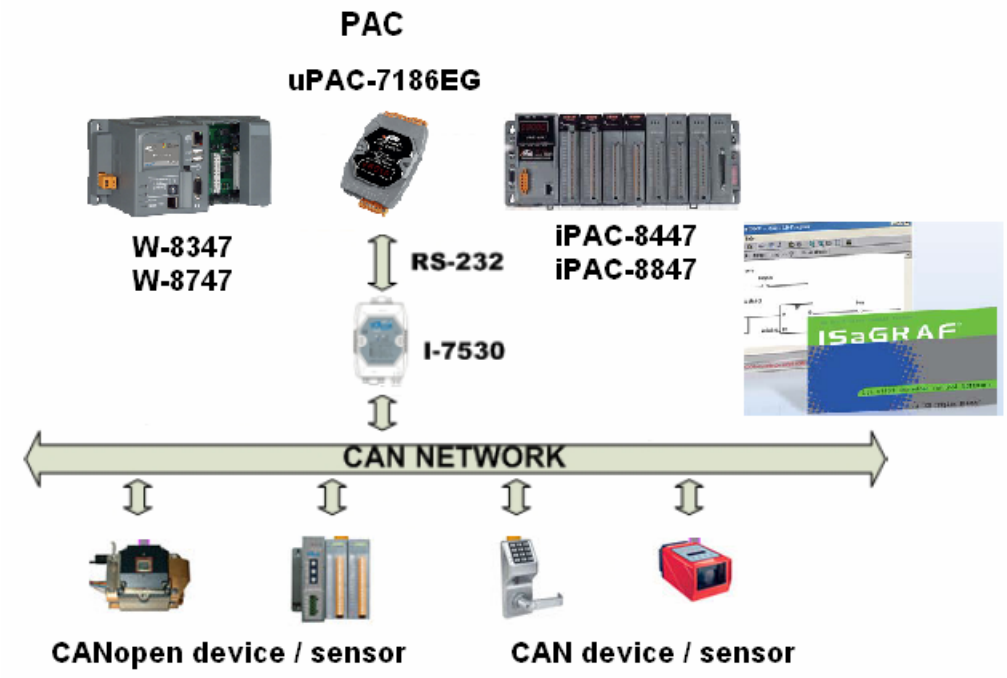
by chun@icpdas.com

以下的 ISaGRAF PAC 有支持 i-7530 : RS-232 to CAN converter . 可透過此 i-7530 converter 來連接 CAN device 或 CAN sensor 或 CAN open device 或 CAN open sensor.

W-8347 與 W-8747 :	從它的 driver 第 3.43 版 (2008.Feb 出版), 或更高版本.
uPAC-7186EG :	從產品可以取得之日起 (約 2008. Mar).
iPAC-8437 / iPAC-8837 :	從產品可以取得之日起 (約 2008. Q3).

優點:

1. 提供經濟型且有效率的 CAN solution , 支持 CAN 2.0A 與 CAN 2.0B .
2. 提供完整的 Demo Program, 以上的 3 類 PAC 全部都可使用 ISaGRAF 軟體來編程, 設計簡單又容易除錯 (有 Ladder, FBD, ST, IL, SFC 與 FC 共 6 種 PLC 語法可以使用)
3. 每台 PAC 可接 1 台 i-7530 或多台 i-7530 來連接各類 CAN / CANopen device 與 Sensor.
4. 除了連接的 CAN device 外,該 ISaGRAF PAC 支持的其它功能與應用也可整合(集成)進來.例如可熱插拔(Hot-Swap) RU-87P4/8 + I-87K 高卡 I/O, Modbus RTU (RS-232/485/422) Device, 連接其它 RS-232/485/422 設備, Data Logger, 發送夾帶資料檔附件的 Email, …等.



相關網頁資料: Or visit www.icpdas.com > Products

ISaGRAF FAQ : www.icpdas.com > FAQ > Software > ISaGRAF > 086
i-7530 : http://www.icpdas.com/products/Remote_IO/can_bus/i-7530.htm
W-8347 / 8747 : http://www.icpdas.com/products/PAC/wincon-8000/isagraf_wincon.htm
uPAC-7186EG : http://www.icpdas.com/products/PAC/i-7188_7186/uPAC-7186eg.htm
i-8112 / 8114 : http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm
X board : http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm

1.1: 相關 軟體 與 硬體設定 :

每台 ISaGRAF PAC 可以連接 i-7530 的 RS-232 串口編號, 與最多可連接 i-7530 的數量如下表.

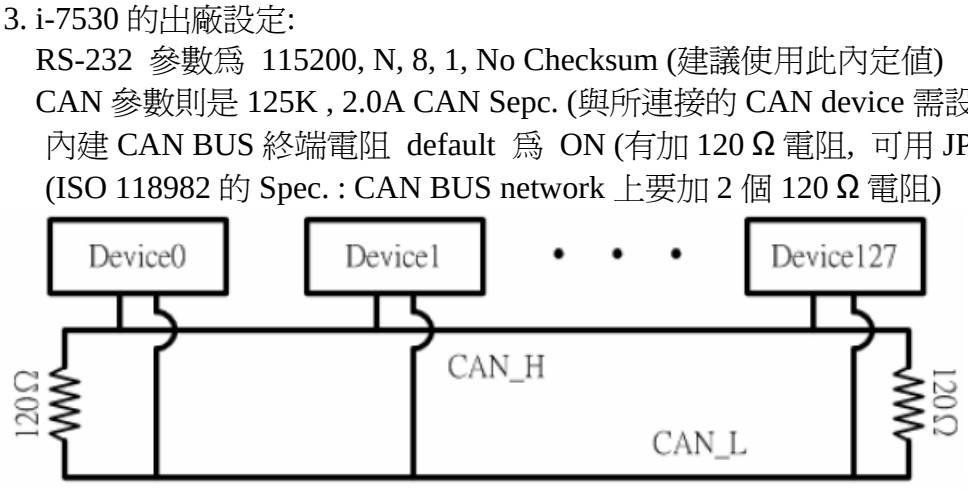
	W-8347 / 8747	**uPAC-7186EG (Mar. 2008)**	**iPAC-8447 / 8847 (Q3. 2008)**
可使用的 RS-232 串口編號	COM2 或 COM5 ~ COM14 (COM5 ~ COM14 位於 插在 Slot 1 ~ 5 上的 i-8112 / i-8114 擴充卡)	COM1 或 COM3 ~ COM8 (COM3 ~ COM8 位於插在 X-socket 上的 X-5xx : RS-232 擴充卡, 如 X-503, 504, 505, 506, 508)	COM1 或 COM3 ~ COM12 (COM5 ~ COM12 位於插在 Slot 0 及 Slot 1 上的 i-8112 / i-8114 擴充卡)
最多可連接的 i-7530 數量 (每個 RS-232 串口可連接一顆 i-7530)	最多可用 10 個 RS-232 串口連接 10 顆 i-7530	最多可用 3 個 RS-232 串口連接 3 顆 i-7530	最多可用 3 個 RS-232 串口連接 3 顆 i-7530
建議每顆 i-7530 不要連接超過多少個 CAN / CANopen device 或 Sensor	60 個 (連接的數量越多, Scan 的效率會降低, PLC Scan time 也大)	30 個 (連接的數量越多, Scan 的效率會降低, PLC Scan time 也大)	30 個 (連接的數量越多, Scan 的效率會降低, PLC Scan time 也大)

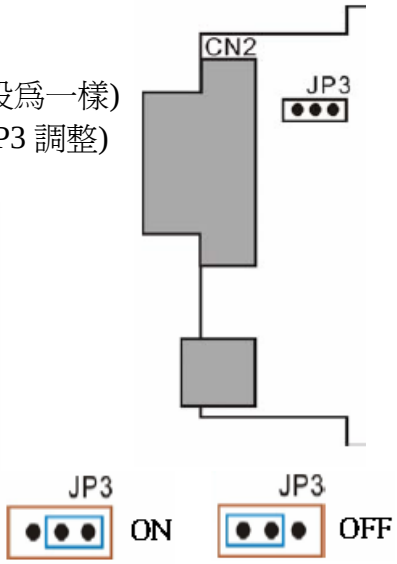
注意:

1. uPAC-7186EG 與 iPAC-8447/8847 的 COM1 串口出貨時內定爲 Modbus RTU slave 串口, 使用者若要使用 COM1 來連接 i-7530, 需先關閉 COM1 : Modbus RTU slave 的設定. (請參考 uPAC-7186EG 快速上手手冊 第 3.6 節)

2. W-8347 / 8747 , uPAC-7186EG , iPAC-8447 / 8847 等 ISaGRAF PAC 只支持使用 115200 , 57600 , 38400 , 19200 或 9600 等 5 種 RS-232 通訊 Baud rate 來連接 i-7530, 且其它的通訊參數需都設爲 No Parity , 8 bit size , 1 stop bit . 而 Checksum 可選擇使用 No 或 Yes.

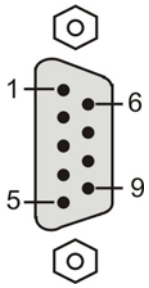
3. i-7530 的出廠設定:
   RS-232 參數爲 115200, N, 8, 1, No Checksum (建議使用此內定值)
   CAN 參數則是 125K , 2.0A CAN Sepc. (與所連接的 CAN device 需設爲一樣)
   內建 CAN BUS 終端電阻 default 爲 ON (有加 120 Ω 電阻, 可用 JP3 調整)
   (ISO 118982 的 Spec. : CAN BUS network 上要加 2 個 120 Ω 電阻)





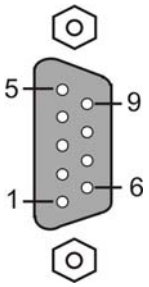
i-7530 的 RS-232 串口 接腳圖: RS-232 DB9 Female Connector (CN1)

Terminal	3-wire RS-232
1	Not Connected
2	TXD
3	RXD
4	Not Connected
5	GND
6	Not Connected
7	
8	
9	

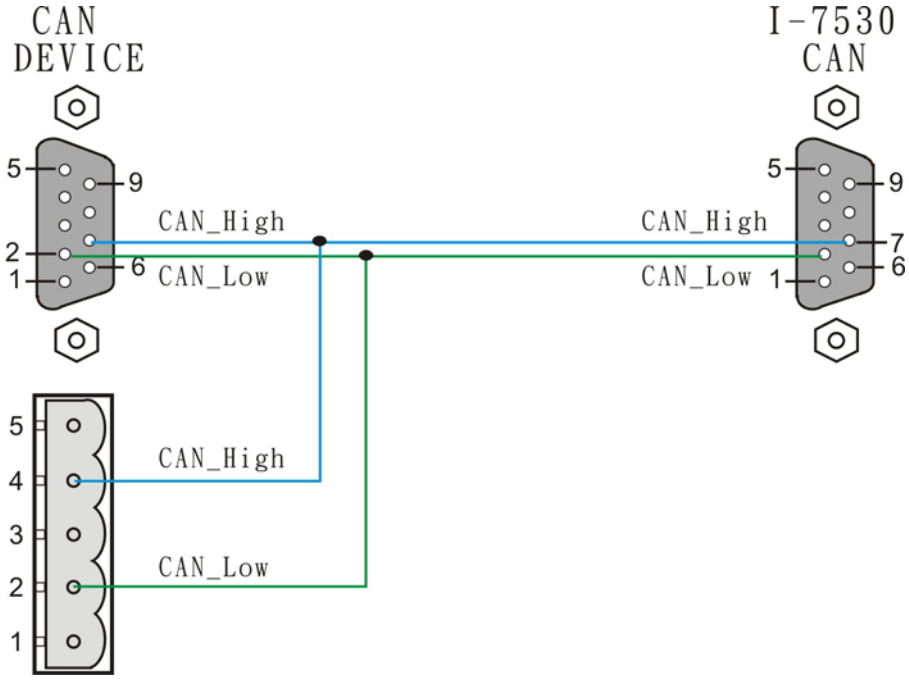


i-7530 的 CAN 串口 接腳圖: CAN DB9 Male Connector (CN2)

Terminal	2-wire CAN
1	Not Connected
2	CAN Low
3	Not Connected
4	
5	
6	
7	CAN High
8	Not Connected
9	



請使用以下方式連接 i-7530 的 CAN 串口 到其它 CAN 與 CANopen device:



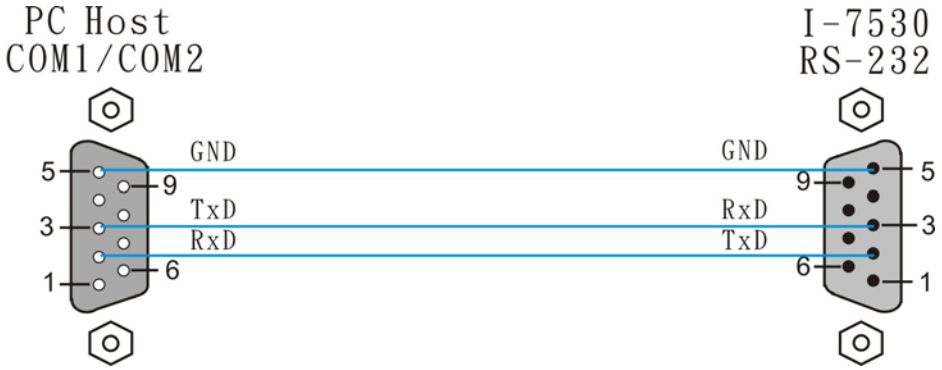
在使用 i-7530 連接 ISaGRAF PAC 與 CAN device 之前，請先確定

(1) CAN 串口的設定是否與要連接的 CAN device 一致 (i-7530 出廠初值是 125K , CAN 2.0A).

(2) RS-232 串口的設定要與連接的 ISaGRAF PAC 軟體設定相同 (i-7530 出廠初值是 115200, 8, N, 1, No checksum，此值也是 ISaGRAF PAC 的內定值，建議雙方都不要更改，直接套用).

若您要變更CAN或RS-232 的設定，您可使用 i-7530 提供的i7530.exe軟體工具來進行，該軟體工具放於 i-7530 出貨包裝盒的 CD-ROM 內 ，或可至 ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/converter/i-7530/utility/ 來下載.
關於更多 i-7530 的 軟體/硬體 說明，請參考它的使用手冊 – i-7530.pdf ，可於 CD-ROM內或 ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/converter/i-7530/manual/ 找到.

PC 與 i-7530 的 RS-232 接法如下:

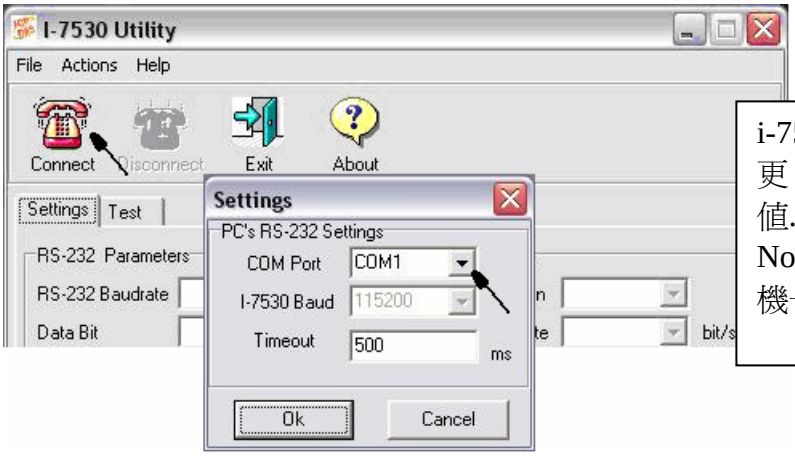


注意:

1. i-7530 背面有提供一個 Dip switch. 當要使用 PC 上的 i7530.exe 軟體工具來變更 i-7530 的通訊參數前，請 (1) 關閉 i-7530 的 24V 供電 (2) 將 Dip 撥到 Init 位置 (3) 開啓 24V 供電給 i-7530，之後就可操作 i7530.exe 來變更相關的 RS-232 或 CAN 串口設定. 切記變更完後要再把 Dip 撥到 Normal 位置, Reset i-7530 電源一次，它才會恢復正常 Normal 的模式 (Normal 模式下 i-7530 才能跟 其它 CAN / CANopen device 溝通).

2. 關於 i7530.exe 工具的詳細使用方法與命令格式，請參考 i-7530.pdf .

操作 i7530.exe 軟體工具:



i-7530 必需在 INIT 模式下才可變更 RS-232 與 CAN 串口的設定值. 設定完後請記得要恢復爲 Normal 模式,且重新對 i-7530 開機一次後才能恢復正常工作.

ISaGRAF PAC 到 i-7530 的 **連接線** 接腳圖:

(1) W-8347/8747 COM2 (與 iPAC-8447/8847 COM4) --- i-7530 的 連接線:

Wincon COM2 / iPAC-8447/8847 COM4 (RS-232) 9-Pin Dsub Female		i-7530 (RS-232) 9-Pin Dsub Male
2 RXD	———	2 TXD
3 TXD	———	3 RXD
5 GND	———	5 GND

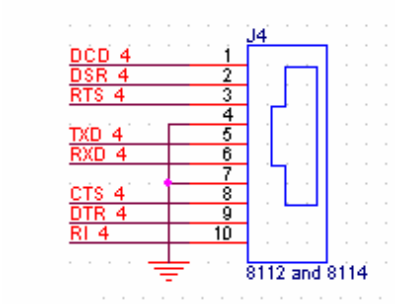
(2) uPAC-7186EG COM1 (與 iPAC-8447/8747 COM1) --- i-7530 的 連接線:

uPAC-7186EG / iPAC-8447/8847 COM1 (RS-232) 9-Pin Dsub Male		i-7530 (RS-232) 9-Pin Dsub Male
2 TXD	———	2 TXD
3 RXD	———	3 RXD
5 GND	———	5 GND

(3) iPAC-8447/8747 COM3 --- i-7530 的 連接線:

iPAC-8447/8847 COM3 (RS-232) 9-Pin Dsub Female		i-7530 (RS-232) 9-Pin Dsub Male
2 TXD	———	2 TXD
3 RXD	———	3 RXD
5 GND	———	5 GND

(4) i-8112 / 8114 擴充卡上的 RS-232 串口 --- i-7530 的 連接線:



i-8112 / 8114 (RS-232)		i-7530 (RS-232) 9-Pin Dsub Male
5 TXD	———	2 TXD
6 RXD	———	3 RXD
7 GND	———	5 GND

(若 i-8112 / 8114 是插在 W-8xx7 上, 需先使用 Wincon utility 先將 RS-232 串口設定過一次, 請參考 "Wincon Getting Started: ISaGRAF PAC" 手冊的 Appendix E)

(5) uPAC-7186EG 上的 X-5xx 擴充卡 --- i-7530 的 連接線:

X-5xx 擴充卡 (RS-232)		i-7530 (RS-232) 9-Pin Dsub Male
RXD	———	2 TXD
TXD	———	3 RXD
GND	———	5 GND

(X-5xx RS-232 擴充卡的接腳說明 請參考 "uPAC-7186EG 快速上手手冊" 的 3.14 節)

1.2: 認識 CAN 與 CANopen 通訊規約的差異

在進入 1.3 節編程之前, 這邊要先介紹 CAN bus 與 CANopen 通訊規約的差異.

CAN bus 是一種通訊介面, 可以比喻成類似 RS-485 通訊介面, 但通訊的硬體 與 實施方式 與 RS-485 不同. 在 CAN 之下又可分爲 CANopen 與 DeviceNet 等 2 種, 所以 CANopen 與 DeviceNet 符合 CAN bus 的規範. 爲了方便了解, user 可以把 CANopen 想像成是 RS-485 Modbus RTU 規約, 而它的底層是 RS-485. 就像 CANopen 底層是 CAN bus 的關係一樣.

每個 CAN bus 封包 (Frame) 格式如下

ID	RTR	DLC	8-byte Data

ID 爲一個識別編號, 若是採用 CAN 2.0A 規範, ID 爲 11 個 bit, 所以編號可以是 0 ~ 7FF (16 進位表示) , 若是採用 CAN 2.0B 規範, ID 爲 29 個 bit, 所以編號可以是 0 ~ 1FFFFFFF (16 進位表示).

CANopen 採用的是 2.0A 規範.

RTR 爲 1 個 bit, 若值是 1 表示 此封包 用於 “Remote-transmit requests”, 或簡稱 “Remote frame” , 它一般是用來詢問其它 CAN device “請把對應的資料發過來” , 此時 ”8-byte Data” 不使用.

RTR 值若是 0 表示 此封包 爲 “Standard frame” , 它一般是用來把 自己的 資料發出去給其它 CAN device. 此時 ”8-byte Data” 就要使用, 而 DLC 指的就是後面的資料有幾個 byte. 可以是 0 ~ 8 個 byte.

CANopen 的封包符合 CAN bus 封包的規定, 但它把 ID 的 11 個 bit 再區分如下.

Function Code, 4 bit	Node ID, 7 bit

Bit 1 ~ 7 指的是 CAN OPEN “Node-ID” (或者稱呼爲 CANopen 站號, 或 CANopen Station No.), 此值可以是 1 ~ 7F (16 進位表示, 若以 10 進位表示爲 1 ~ 127) , 值 0 用於一些特殊用途. (比如設定某個 CANopen 站號切換到 “operational state” , 就是採用 Node ID = 0) . 所以 一條 CANopen 網路 最多可連接 127 個 CANopen device.

Function Code 有 4 個 bit (爲 bit 8 ~ 11), 它定義了 CANopen 封包的用途, 比如有些用途是要求傳 Application Data 過來, 有些是設定 Application data 出去, … 等. 在此無法仔細介紹各種 Function code 的用途, 請參考各別 CANopen device 的使用手冊.

使用 ISaGRAF PAC 來連接 CAN 或 CANopen device 時, User 應該要了解有使用到的 CAN device 的規定, 比如 CANopen 站號是幾號, Application data 是要用那個 Function Code 來傳出, 資料格式, …等.

注意: 請不要把 CAN device 與 CANopen device 混合在同一個 CAN bus 網路上使用, 因爲 ID 的定義很有可能會互相衝突. 若該網路上連接的都是 CANopen device, 那只要 CANopen 站號不同 (1~127), 就不會衝突.

1.3: 使用 ISaGRAF 軟體 來 編寫 CAN 的控制程式

在開始編程之前, 請先確定你使用的 ISaGRAF PAC 的 drver 版本是否正確.

W-8347 與 W-8747 :	從它的 driver 第 3.43 版 (2008.Feb 出版), 或更高版本.
uPAC-7186EG :	從產品可以取得之日起 (約 2008. Mar).
iPAC-8437 / iPAC-8837 :	從產品可以取得之日起 (約 2008. Q3).

然後確定是否您PC上安裝的ISaGRAF 軟件有已經有包含以下的 ISaGRAF IO library (2008.Feb 出版). 若沒有可至 http://www.icpdas.com/products/PAC/i-8000/isagraf.htm 下載 "ICP DAS Utilities For ISaGRAF" , 然後運行其內的 setup.exe 來先 Remove 之後再 Re-install一次.

關於以下的 CAN_R , CAN_BY_W , CANSTR_W 與 CANOP_ST 的使用說明請參考 1.4 節.

CAN7530 :	I/O complex equipment (IO connection 視窗內 IO 複合板)
CAN_R:	ISaGRAF C function-block
CAN_BY_W:	ISaGRAF C function
CANSTR_W:	ISaGRAF C function
CANOP_ST:	ISaGRAF C function

www.icpdas.com > FAQ > Software > ISaGRAF > 中文 > 086 內可以取得 ISaGRAF連接CAN的範例程式 wdmo_71a ~ wdmo_71e與 CAN7530, CAN_R, CAN_BY_W, CANSTR_W, CANOP_ST 等ISaGRAF IO library (給各別安裝部份IO library使用 "即非重新安裝整個ISaGRAF IO libary", 請參考 "ISaGRAF進階使用手冊-附錄" Appendix - A.2). 若不知道如何安裝範例程式於 ISaGRAF內, 請參考 "ISaGRAF進階使用手冊" 第 9.5 節.

範例程式:

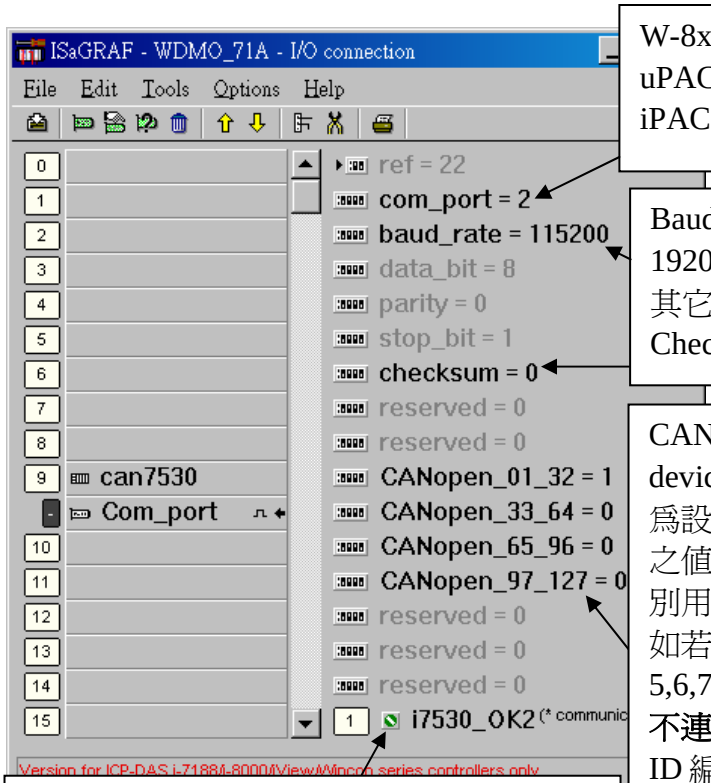
wdmo_71a :	W-8xx7 COM2 連接一顆 i-7530 再連接一個 CANopen device
wdmo_71b :	W-8xx7 + i-8112 使用 COM5 與 COM6 兩個 RS-232 串口, 分別都連接一顆 i-7530, 再分別都連接一個 CANopen device
wdmo_71c :	W-8xx7 COM2 連接一顆 i-7530 再連接一個 CAN device
wdmo_71d :	W-8xx7 + i-8112 使用 COM5 與 COM6 兩個 RS-232 串口, 分別都連接一顆 i-7530, 再分別都連接一個 CAN device
wdmo_71e :	W-8xx7 + i-8112 使用 COM5 與 COM6 兩個 RS-232 串口, 分別都連接一顆 i-7530, 之後 COM5 的那個 i-7530 連接一個 CANopen device, 而 COM6 的那個 i-7530 連接一個 CAN device

注意:

1. 範例 wdmo_71a 與 wdmo_71c 若要使用在 uPAC-7186EG COM1 或 iPAC-8447/8847 COM1 上, 需更改程式內 (1) IO connection 視窗內 "can7530" 內的 "com_port" 之值爲 1 , (2) Integer 變數宣告 "Port2" 之初值爲 1, 然後重新 Compile 後才能使用. (uPAC-7186EG 與 iPAC-8447/8847 的 COM1 串口出貨時內定爲 Modbus RTU slave 串口, 使用者若要使用 COM1 來連接 i-7530, 需先關閉 COM1 : Modbus RTU slave 的設定. 請參考 uPAC-7186EG 快速上手手冊 第 3.6 節)
2. 範例 wdmo_71b , wdmo_71d 與 wdmo_71e 若要使用在 uPAC-7186EG 或 iPAC-8447/8847 的其它 RS-232 串口上, 需更改程式內 (1) IO connection 視窗內 "can7530" 內的 "com_port" 之值 , (2) Integer 變數宣告 "Port5" 與 "Port6" 之初值, 然後重新 Compile 後才能使用.

以下以 wdmo_71a 爲例來說明如何編寫 ISaGRAF 程式 (只列出主要的程式).

Step 1 : 在 IO connection 視窗內連接 “can7530”



W-8xx7 可使用 COM2, COM5 ~ 14 (最多 10 個)
uPAC-7186EG 則是 COM1, COM3 ~ 8 (最多 3 個)
iPAC-8xx7 則是 COM1, COM3 ~ 12 (最多 3 個)

Baud_rate 只能設爲 115200 , 57600 , 38400 , 19200 , 9600. (建議使用 115200)
其它參數固定爲 8 , 0 , 1 不可修改
Checksum 可以是 0: No 或 1:Yes

CANopen_01_32 用來設定是否要連接 CANopen device ID 編號 1 ~ 32,採用 16 進位 32-bit 表示法. Bit 1 爲設定是否要連接 CAN open device ID 編號 1,若該 bit 之值爲 0 表示不連接, 1 表示要連接. Bit 2 ~ 32 則是分別用來設定是否要連接 CAN open ID 編號 2 ~ 32. 比如若值設爲 F0 則表示要連接 CANopen ID 編號 5,6,7,8. **若值設爲 0,表示只連接一般的 CAN device, 不連接 CAN open device**, 若值設爲 1C 則表示要連接 ID 編號 3,4,5. 若值設爲 3 則表示要連接 ID 編號 1 與 2. 若值設爲 FFFFFFFF 則表示要連接 ID 編號 1~32. 其它 ID 33 ~ 127 的參數定義與 CANopen_01_32 類似.

此 D/I 傳回值表示該 i-7530 是否正常連線.
True: 正常 , False: 斷線

Step 2 : 編寫 ST 程式 – “Scan1”

```
if  INIT  then
(* INIT 初值在 dictionary 內是 True,所以此區內的程序只在第一個 PLC scan 內運行一次而已 *)
  INIT := False ;
  TMR2 := T#0s ;
  Tstart(TMR2) ;    (* 開始啓動 Timer 變數 TMR2 來計時 *)
  Max_Step2 := Period2 / Interval2 ;
  (*設定一個變數來取得 CAN open ID 編號 1 的連線狀態,超過 5 秒沒回應,值就設爲 False *)
  TMP := CanOp_st( Port2 , CAN_OPEN2_ID1 , 1 , 5 ) ;   (* 變數爲 CAN_OPEN2_ID1 *)
  (* 若 Port2 還有連接其它 CANopen ID, 請加入.
  如 TMP := CanOp_st( Port2 , CAN_OPEN2_ID2 , 2 , 5 ) ;          編號 2 , 5 秒
  如 TMP := CanOp_st( Port2 , CAN_OPEN2_ID3 , 3 , 10 ) ;         編號 3 , 10 秒
  注意: 要連接 CANopen device 要先在 Step1 – “can7530” 內做好設定, 不然 ” CanOp_st( )”
      會沒法作用, 若不連接 CAN open device 而只連接一般的 CAN device, 則不必使用
      CanOp_st( )函式 *)
end_if ;
```

Step 3 : 編寫 ST 程式 – “Can2_r” (處理 從 i-7530 讀到的 CAN frame)

```
num_frame := 0 ;     (*  一開始先設初值爲  0 *)

While    num_frame < 10    Do    (*  設定每次進入此 ST 程式,最多只可讀 10 個 CAN frame *)

   (*  測試是否有收到 CAN frame,   此例  “CanR”  是宣告爲  “CAN_R”  的 FB instance *)
   CanR(Port2) ;

   (*  此值若爲 True,  表示有讀到 CAN frame,  若爲 False,  則以下的值無意義  *)
   Can_Coming := CanR.Q_ ;

   Can_Mode := CanR.MODE_ ;    (* CAN frame 的 MODE 值, 0 : 2.0A  或  1 : 2.0B *)
   Can_RTR    := CanR.RTR_ ;     (* CAN frame 的 RTR 值, 0: Standard frame  或  1: remote *)
   Can_ID      := CanR.ID_ ;          (* CAN frame 的 ID *)
   Can_DLC    := CanR.DLC_ ;     (* CAN frame 的資料長度, 0 ~ 8 *)
   Can_By1    := CanR.BY1_ ;       (*  讀到的 Byte 資料,  最多 8 個  Byte *)
   Can_By2    := CanR.BY2_ ;
   Can_By3    := CanR.BY3_ ;
   Can_By4    := CanR.BY4_ ;
   Can_By5    := CanR.BY5_ ;
   Can_By6    := CanR.BY6_ ;
   Can_By7    := CanR.BY7_ ;
   Can_By8    := CanR.BY8_ ;
   Can_str    := CanR.MSG_ ;       (*  讀到的 Byte 資料的 String 格式  *)

   If    Can_Coming = False    then
     return ;          (*  未讀到  CAN frame,  離開此 ST 程式去 Run 下一個程式  *)
   end_if ;

   num_frame := num_frame + 1 ;     (*  有讀到  CAN frame,  數量  + 1 *)

   if    Can_Mode = 0    then       (*  若是  CAN 2.0A frame *)
     if    Can_RTR = 0    then       (*  若是  Standard frame *)
         Case    Can_ID    Of
           16#181 :     (* D/I 資料: Function code 16#180 + CAN OPEN ID 1 = 16#181 *)
             If    Can_DLC > 0    then                  (*  資料 byte 數量必需  > 0 *)
               DI_01 := Byte_Bit( Can_By1, 1) ;     (*  取出 Ch.1 的 D/I 值*)
               DI_02 := Byte_Bit( Can_By1, 2) ;
               DI_03 := Byte_Bit( Can_By1, 3) ;
               DI_04 := Byte_Bit( Can_By1, 4) ;
               DI_05 := Byte_Bit( Can_By1, 5) ;
               DI_06 := Byte_Bit( Can_By1, 6) ;
               DI_07 := Byte_Bit( Can_By1, 7) ;
               DI_08 := Byte_Bit( Can_By1, 8) ;     (* Ch.8 *)
             End_if ;
```

```
            16#281 :     (* A/I 資料: Function code 16#280 + CAN OPEN ID 1 = 16#281 *)
              If   Can_DLC >= 8   then     (*  本例假定 16#281 內含有 4 個 Ch.的  A/I 資料  *)
                AI_01 := Byte_sWD( Can_By1 , Can_By2 ) ;     (*每 Ch. A/I 由 2 個 byte 組成*)
                AI_02 := Byte_sWD( Can_By3 , Can_By4 ) ;     (* LoByte , HiByte *)
                AI_03 := Byte_sWD( Can_By5 , Can_By6 ) ;
                AI_04 := Byte_sWD( Can_By7 , Can_By8 ) ;
              End_if ;

            16#381 :     (* A/I 資料: Function code 16#380 + CAN OPEN ID 1 = 16#381 *)
              If   Can_DLC >= 8   then     (*  本例假定  16#381 內含有 4 個 Ch.的  A/I 資料  *)
                AI_05 := Byte_sWD( Can_By1 , Can_By2 ) ;     (*每 Ch. A/I 由 2 個 byte 組成*)
                AI_06 := Byte_sWD( Can_By3 , Can_By4 ) ;     (* LoByte , HiByte *)
                AI_07 := Byte_sWD( Can_By5 , Can_By6 ) ;
                AI_08 := Byte_sWD( Can_By7 , Can_By8 ) ;
              End_if ;

             (*  若有其它  CANopen ID 資料請加於此處.
                 比如 Function code 16#280 + CAN OPEN ID 2 = 16#282 *)
             (*
             16#282 :
               If   Can_DLC >= 4   then
                 AI_09 := Byte_sWD( Can_By1 , Can_By2 ) ;
                 AI_10 := Byte_sWD( Can_By3 , Can_By4 ) ;
               End_if ;
             *)

          End_case ;

       Else     (* Can_RTR = 1 :   收到的 CAN frame  爲  "Remote"   frame              *)

       end_if ;

    else     (* Can_Mode=1 :  收到的 CAN frame  爲  CAN 2.0B   frame *)

    end_if ;

End_While ;
```

Step 4 : 編寫 ST 程式 – “Can2_W” (傳送 CAN frame 資料給 i-7530)

```
(* wdmo_71a 使用 Period2 = 200 (ms) , Interval2 = 20 (ms). 表示每 0.2 秒爲一個週期, 每一個週
期內最多可傳送 (200 / 20) - 1 = 9 個 CAN frame, Frame 與 Frame 的時間 間隔爲 0.02 秒
Step2=1 (於 0 ms) : 傳 Remote frame 到 ID = 16#181 要求 CANopen device 1 傳 D/I 資料過來
Step2=2 (於 20 ms) : 傳 Remote frame 到 ID = 16#281 要求 CANopen device 1 傳 A/I 資料過來
Step2=3 (於 40 ms) : 傳 Remote frame 到 ID = 16#381 要求 CANopen device 1 傳 A/I 資料過來
Step2=4 (於 60 ms) : 傳 Standard frame 到 ID = 16#201 要求 CANopen device 1 對 D/O 輸出
Step2=5 (於 80 ms) : 傳 Standard frame 到 ID = 16#301 要求 CANopen device 1 對 A/O 輸出
Step2 = 0, 6 ~ 10 : (於每個週期的 80 到 200 ms 時間) : 不傳送任何 CAN frame
Step2 在本例只能用到 9, 若需要使用更多 Step 值,需更改 Period2 與 Interval2 的宣告初值
Interval 之值最小需爲 10 (ms) , 更不可爲 0 或 負值.
Period2 之值至少要是 Interval 值的 2 倍, 且不可小於 100 (ms) *)

TMR2_val := ANA( TMR2 ) ;     (* 轉換 Timer 爲整數值, 單位 ms *)
TMR2_val := MOD( TMR2_val , Period2 ) ;   (* 取餘數 *)

Send2 := False ;     (* 先設爲 False: 表示 不要求 傳送資料 *)

(* 此處將 Step 區分爲 200/20 = 10 個 Step. 間隔爲 20 ms *)
(*   Max_Step2 之值是在 Scan1 程式內算出來的, 本例爲 200/20 = 10 *)
if   Step2 >= 0   and   Step2 <= Max_Step2-1   then
  if   TMR2_val >= Interval2 * Step2   then
    Step2 := Step2 + 1 ;
    Send2 := True ;       (* 已抵達每個 Step 的時間點, 設爲 True 來要求傳出 CAN frame *)
  end_if ;

else     (* Step 值已經抵達 Max_Step 值時 *)

  if   TMR2 >= TMR(Period2)   then     (* 若 Timer 值已進行完一個週期 *)
    Step2 := 0 ;            (* reset Step 爲 0 *)
    TMR2 := T#0s ;      (* reset Timer 值爲 0 *)
  end_if ;

end_if ;

(* 以下的 Code 爲處理每個 Step 要傳送的 CAN frame *)
If   Send2   then

  Send2 := False ;     (* reset 爲 False *)

  CASE   Step2   Of

    0 :       (* Step 值爲 0: 不可傳送資料 *)
```

```
1 :   (*Step 值爲 1:  傳出詢問 CAN open device ID=1 的 D/I 值的 frame ,  詢問 1 個  byte*)
        TMP := CAN_BY_W( Port2 , 0 , 1 , 16#181 , 1 ,    0,0,0,0,    0,0,0,0 ) ;

2 :   (*Step 值爲 2:  傳出詢問 CAN open device ID=1 的 A/I 值的 frame,  詢問 8 個 byte *)
        TMP := CAN_BY_W( Port2 , 0 , 1 , 16#281 , 8 ,    0,0,0,0,    0,0,0,0 ) ;

3 :   (*Step 值爲 3:  傳出詢問 CAN open device ID=1 的 A/I 值的 frame,  詢問 8 個 byte *)
        TMP := CAN_BY_W( Port2 , 0 , 1 , 16#381 , 8 ,    0,0,0,0,    0,0,0,0 ) ;

4 :   (*Step 值爲 4:  傳出輸出 CAN open device ID=1 的 D/O 值的 frame,  輸出 1 個 byte *)

        (*  先將 8 個  Boolean 值組合成  1  個  byte 值  *)
        Tmp_val := Bit_WD( DO_01, DO_02, DO_03, DO_04, DO_05, DO_06, DO_07,
                           DO_08, False, False, False, False, False, False, False, False ) ;

        (*  再傳出去  *)
        TMP := CAN_BY_W( Port2 , 0 , 0 , 16#201 , 1 ,    Tmp_val ,    0,0,0,    0,0,0,0 ) ;

5 :   (*Step 值爲 5:  傳出輸出 CAN open device ID=1 的 A/O 值的 frame,  輸出 8 個 byte *)
        TMP := CAN_BY_W( Port2 , 0 , 0 , 16#301 , 8 ,
                           MOD(AO_01,256) , AO_01/256 ,     (* Lo_byte , Hi_byte *)
                           MOD(AO_02,256) , AO_02/256 ,
                           MOD(AO_03,256) , AO_03/256 ,
                           MOD(AO_04,256) , AO_04/256      ) ;

  (*  若還有其它 Frame 要送出,  請加入
    Step2 在本例只能用到 9,若需要使用更多 Step 值,需更改 Period2 與 Interval2 的宣告初值
    Interval 之值最小需爲 10 (ms) ,  更不可爲  0 或負值.
    Period2 之值至少要是  Interval 值的 2 倍,  且不可小於  100 (ms)
  *)
  (*  以下爲詢問  CAN open ID=2  的  A/I 資料,  詢問 4 個 byte
  6 :
        TMP := CAN_BY_W( Port2 , 0 , 1 , 16#282 , 4 ,    0,0,0,0,    0,0,0,0 ) ;
  *)

  (*本例由於  (Period2 , Interval2)  爲  (200, 20) ,  所以 Step 最多只能使用到(200/20)-1= 9 *)

 End_case ;

End_if ;
```

其它關於 wdmo_71b ~ wdmo_71e 的詳細說明, 請參考 ”ISaGRAF 進階使用手冊” 第 9.5 節 先安裝這些範例程式於 ISaGRAF 內, 然後開啓 ISaGRAF 內各別的程式, 程式內有詳細的註解說明.

1.4: 關於 CAN 函式的使用說明

CAN_R(PORT_) ： 爲 c-function block (使用 ST 設計時要宣告使用 FB instance)

測試是否有收到 CAN frame

輸入參數:

PORT_ Integer 可以使用以下的值.
W-8xx7 : 可使用 2 , 5 ~ 14 (最多可使用 10 個 RS-232 串口)
uPAC-7186EG: 可使用 1 , 3 ~ 8 (最多可使用 3 個 RS-232 串口)
iPAC-8xx7: 可使用 1 , 3 ~ 12 (最多可使用 3 個 RS-232 串口)

傳回值:

Q_ Boolean True : 有收到 CAN frame . False : 未收到 CAN frame.

只有 Q_ 爲 True 時, 以下的 傳回值 才有意義

MODE_ Integer 0: frame 爲 2.0A frame , (ID 爲 11 個 bit)
1: frame 爲 2.0B frame , (ID 爲 29 個 bit)

RTR_ Integer 0: frame 爲 “Standard” frame (有含 0 ~ 8 個 byte 資料)
1: frame 爲 “Remote” frame, (不含 byte 資料)

ID_ Integer frame 的 CAN ID .

DLC_ Integer frame 內含的資料 Byte 數量, 0 ~ 8 .

BY1_ ~ BY8_ Integer frame 內含的 Byte 資料.
(只有 ”Standard” frame 的 BY1_ ~ BY8_ 才有意義)

MSG_ Message 收到的 Byte 資料的 String 格式. 注意若 BY1_ ~ BY8_ 內有值爲 0, 會被當成是 Strine 結尾. 比如若收到 8 個 Byte 分別以 16 進位值來表示 41 , 42 , 43 , 4A , 0 , 4B , 4C , 4D , 則 MSG_ 會是 'ABCJ'
(只有 ”Standard” frame 的 MSG_才有意義)

CAN_BY_W(PORT_ , MODE_ , RTR_ , ID_ , DLC_ , BY1_ , BY2_ , BY3_ , BY4_ , BY5_ , BY6_ , BY7_ , BY8)　：　爲 c-function

傳送 CAN frame 出去給 CAN / CANopen device

輸入參數:

PORT_　Integer　可以使用以下的值.
W-8xx7 :　可使用 2 , 5 ~ 14　(最多可使用 10 個 RS-232 串口)
uPAC-7186EG:　可使用 1 , 3 ~ 8　(最多可使用 3 個 RS-232 串口)
iPAC-8xx7:　可使用 1 , 3 ~ 12　(最多可使用 3 個 RS-232 串口)

MODE_　Integer　0: frame 爲 2.0A frame , (ID 爲 11 個 bit)
1: frame 爲 2.0B frame , (ID 爲 29 個 bit)

RTR_　Integer　0: frame 爲 “Standard” frame (有含 0 ~ 8 個 byte 資料)
1: frame 爲 “Remote” frame, (不含 byte 資料, BY1_ ~ BY8_請都給 0)

ID_　Integer　frame 的 CAN ID .

DLC_　Integer　frame 內含的資料 Byte 數量, 0 ~ 8 .

BY1_ ~ BY8_　Integer　frame 內含的 Byte 資料.
(只有 ”Standard” frame 的 BY1_ ~ BY8_ 才有意義.
若 RTR_是 1, 請將 BY1_ ~ BY8_都設爲 0)

傳回值:

Q_　Boolean　True : 成功 .　False : 失敗.
(失敗的原因 有可能是 傳入參數值是錯的, 或 該 CAN PORT_ 未成功開啓 或 其它)

CANSTR_W(PORT_ , MODE_ , ID_ , MSG_)　：　爲 c-function

傳送 CAN frame 出去給 CAN / CANopen device　(傳送的資料爲 String)

*** 使用 CANSTR_W( ) 傳出的 CAN frame 一定都是 “Standard” frame, 若要傳送 “Remote” frame 請改使用 CAN_BY_W( )

輸入參數:

PORT_　Integer　可以使用以下的值.
W-8xx7 :　可使用 2 , 5 ~ 14　(最多可使用 10 個 RS-232 串口)
uPAC-7186EG: 可使用 1 , 3 ~ 8　(最多可使用 3 個 RS-232 串口)
iPAC-8xx7:　可使用 1 , 3 ~ 12　(最多可使用 3 個 RS-232 串口)

MODE_　Integer　0: frame 爲 2.0A frame , (ID 爲 11 個 bit)
1: frame 爲 2.0B frame , (ID 爲 29 個 bit)

ID_　Integer　frame 的 CAN ID .

MSG_　Message　要傳出去的 String, 最長只能是 8 個 byte .
(第 9 個 byte 及 以上的 byte 不會採用)

傳回值:

Q_　Boolean　True : 成功 .　False : 失敗.
(失敗的原因 有可能是 傳入參數值是錯的, 或 該 CAN PORT_ 未成功開啓, 或 其它)

CANOP_ST(PORT_ , BOO_ , ID_ , TOUT_)　：　爲 c-function

設定一個 Boolean 變數來取得 CAN open device 各別 ID 編號 的 連線狀態

*** 只給連接 CAN open device 時使用, 一般其它的 CAN device 不需使用它.
*** 此 CANOP_ST 函式只能在第一個 PLC scan 內使用

輸入參數:

PORT_　Integer　可以使用以下的值.
W-8xx7 :　可使用 2 , 5 ~ 14　(最多可使用 10 個 RS-232 串口)
uPAC-7186EG: 可使用 1 , 3 ~ 8　(最多可使用 3 個 RS-232 串口)
iPAC-8xx7:　可使用 1 , 3 ~ 12　(最多可使用 3 個 RS-232 串口)

MODE_　Integer　0: frame 爲 2.0A frame , (ID 爲 11 個 bit)
1: frame 爲 2.0B frame , (ID 爲 29 個 bit)

BOO_　Boolean　需爲一個 Boolean 變數的名稱, 不可使用 Constant 或 True, False 等值

ID_　Integer　CAN open device 的 ID 編號 , 值爲 1 ~ 127 .

TOUT_　Integer　單位爲秒, 可以是 3 ~ 120. 即爲超過多久沒收到該 CANopen device 傳過來的 CAN frame, 就會設爲 False 來表示該 CAN open device 斷線.
若值爲 True, 表示該 CANopen device 一直有資料在指定的時間內傳過來.

傳回值:

Q_　Boolean　True : 設定成功 .　False : 失敗.
(失敗的原因 有可能是 (1)傳入參數值是錯的, 或 (2)該 CANopen ID 沒有在 IO connection 視窗內的 “can7530” 內設爲啓用該 CANopen ID
或 (3) 此 CANOP_ST 函式不在第一個 PLC scan 內使用 或 (4) 其它)