I-7188XBD-CAN/µPAC-7186EXD-CAN

使用者手冊

產品保固

凡泓格科技股份有限公司產品從購買即日起，若無任何材料性缺損保固一年。

免責聲明

凡使用本系列產品除品質量所造成的損害，泓格科技股份有限公司不承擔任何法律責任。泓格科技股份有限公司有義務提供本系列產品可靠而詳盡的資料，但保留修改權利，且不承擔使用者非法利用資料對第三方所造成侵害構成的法律責任。

註冊商標

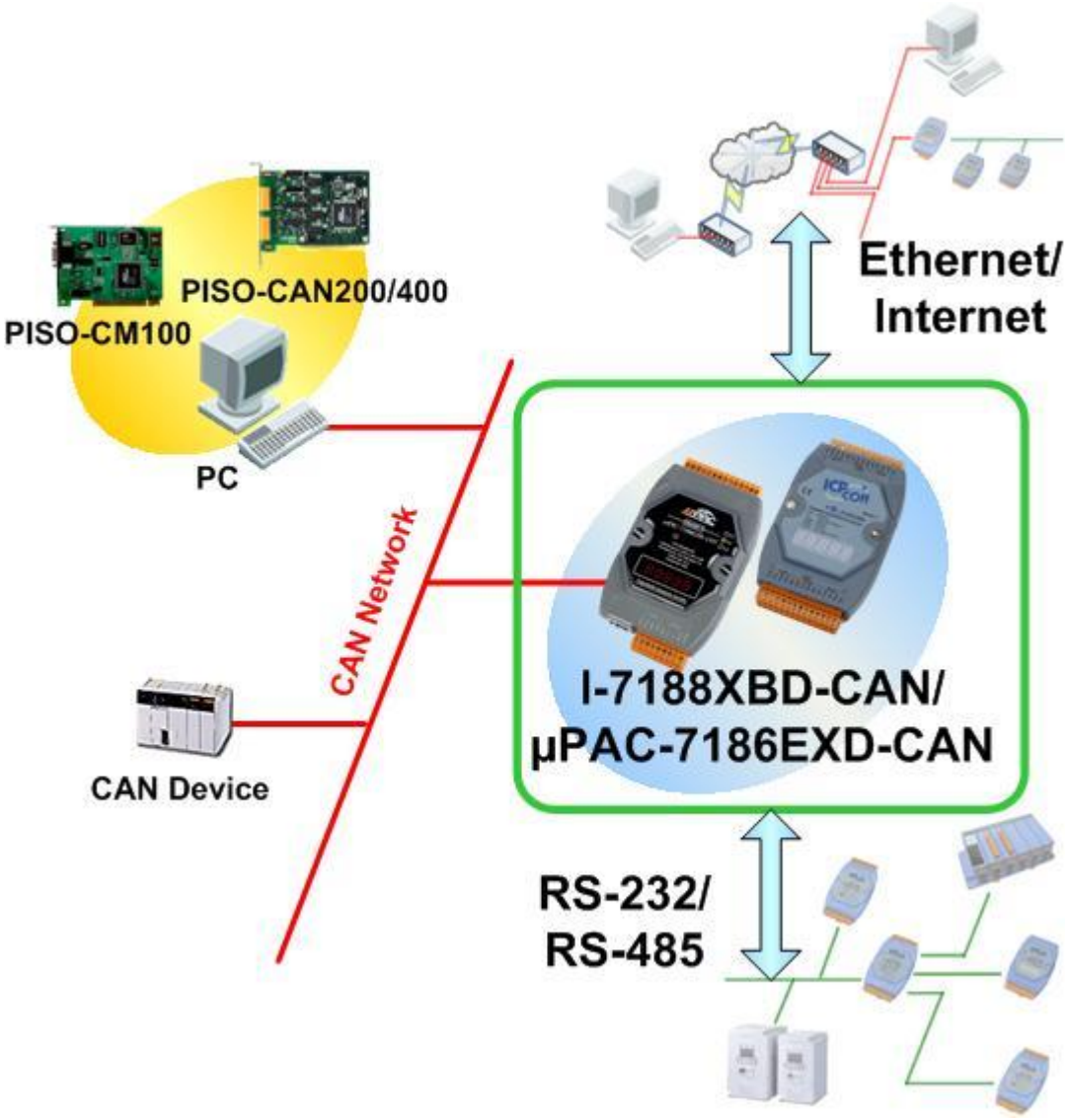
手冊中所涉及所有公司商標，商標名稱以及產品名稱分別屬於該商標或名稱的擁有者所有。

目錄

1 簡介

1.1 概要

CAN (Controller Area Network；控制器區域網路) 是一種串列式通訊協定，特別適合使用在主系統或子系統下提供更完整的智慧型網路設備如感應器及驅動器。它提供高安全等級及有效率的分散式即時控制，更具備了偵錯和優先權判別的機制。 I-7188XBD-CAN/µPAC-7186EXD-CAN 使用獨立的 CAN 控制器，且由一個 XC100 子卡及 I-7188XBD/µPAC-7186EXD 板卡所組成，並提供一個 5-Pin 螺釘接線端子的 CAN 通訊埠。 除此之外，I-7188XBD-CAN/ µPAC-7186EXD-CAN 使用了 Phillips SJA1000T CAN 控制器與 82C250 收發器， 相容於 CAN 2.0A 及 2.0B 規範，支援錯誤重傳功能、仲裁機制及錯誤偵測。I-7188XBD-CAN/ µPAC-7186EXD-CAN 可用應於數種工業通訊介面，如 RS-232、RS-485 和 Ethernet。因此，使用者可以廣泛的應用在不同的通訊協定之間。


PISO-CAN200/400
PISO-CM100
PC
Ethernet/
Internet
CAN Network
I-7188XBD-CAN/
µPAC-7186EXD-CAN
CAN Device
RS-232/
RS-485

1.2 硬體特性

- 1000V_{DC} 隔離電壓保護
- 相容於 CAN 2.0A/B 的規範
- 可程式傳輸速率高達 1 M(bps)
- 可跳線選擇 120Ω(歐姆)終端電阻
- 可程式 XC100 函式庫
- 可程式化
- 支援多種通訊介面
- 支援看門狗機制
- 可設定中斷服務程序與定時器(timers)
- 支援可程式的 C/C++ 語言

1.3 硬體規格

系統

- CPU: 80186, 80MHz (適用於 µPAC-7186EXD-CAN)
- CPU: 80188, 40MHz (適用於 I-7188XBD-CAN)
- SRAM: 512K bytes
- 內建快取記憶體(flash)、EEPROM、NVSRAM、RTC(實時時鐘)
- 內建看門狗定時器
- 16-bit 定時器

快閃記憶體

- 512K bytes
- 最小抹除單位為一個扇區(sector) (64K bytes)
- 100,000 次抹/寫週期

EEPROM

- 16K bytes (64 個區塊，每一區塊為 256 bytes)
- 資料保存長達 100 年以上
- 可重覆讀寫 1,000,000 次以上

RTC(實時時鐘)

- 支援時間格式為秒、分、時、日、月、年(有效期限 1980～2079)
- NVSRAM: 31 bytes、電池備份、資料有效期限達 10 年

CAN port

- Philip SJA1000 CAN 控制器
- Philip 82C250 CAN 收發器
- CAN 端 1000V 電壓保護
- 120Ω 終端電阻跳線選擇
- 16M Hz 時脈

COM1

- RS-232 或 RS-485 介面
- RS-232: TXD, RXD, RTS, CTS, GND
- 通訊速度: 最快 115200(bps)
- 程式下載通訊埠

COM2

- RS-485: D2+, D2-

- 通訊速度: 最快 115200(bps)
- 可連接到 DCON 的 IO 模組

顯示

- 5 位 7 段顯示器
- 4 個 LED 燈 (L1, L2, L3 及 round LED)

數位輸入 (僅適用於 I-7188XBD-CAN)

- 1 個 DI 數位輸入通道
- 乾接點: 邏輯準位 0: 接近 GND、邏輯準位 1: open
- 濕接點: 邏輯準位 1: 3.5V~30V、邏輯準位 0: 0~1V

數位輸出 (僅適用於 I-7188XBD-CAN)

- 1 個 DO 數位輸出通道
- 輸出電流與電壓：100 mA, 最大 30V
- 開集極輸出

電源需求

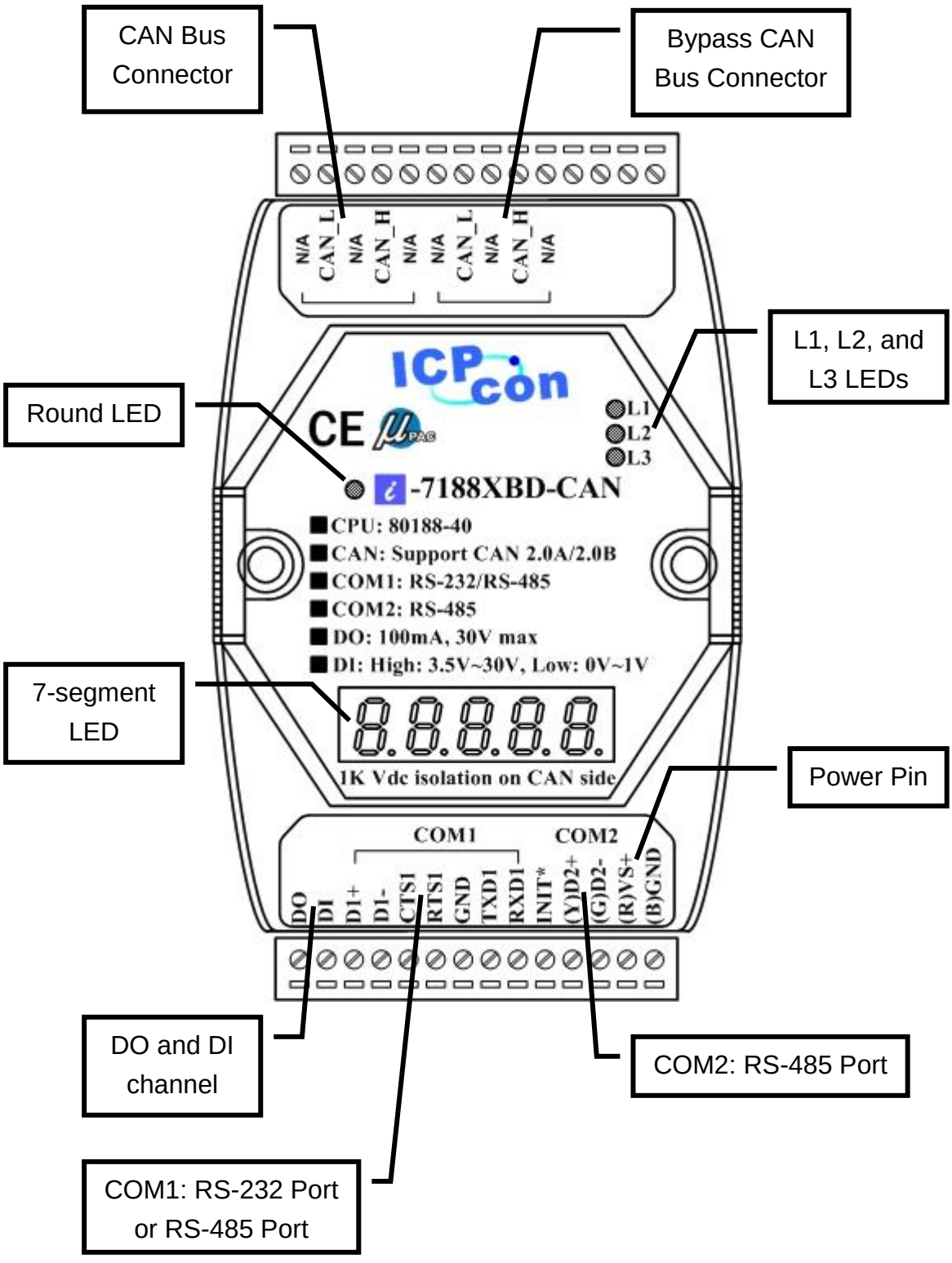
- 電源供應：+10~+30 V_{DC} (non-regulated)
- 電源消耗: 3.0W

應用環境

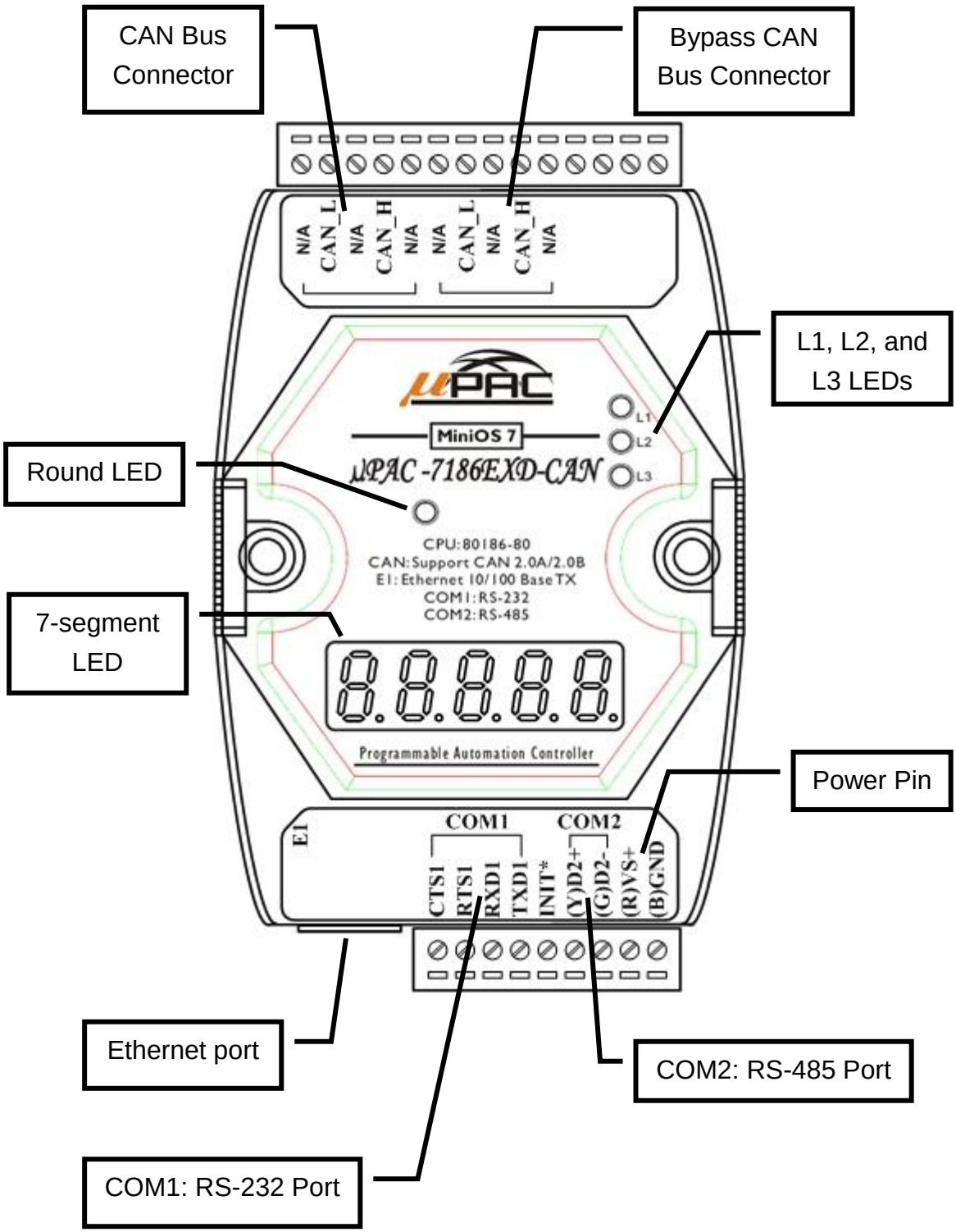
- 工作溫度: -25°C to +75°C
  保存溫度: -30°C to +85°C
- 相對濕度: 5~95%
- 尺寸: 123mm*64.5mm*19.6mm

2 硬體配置

2.1 I-7188XBD-CAN 硬體結構

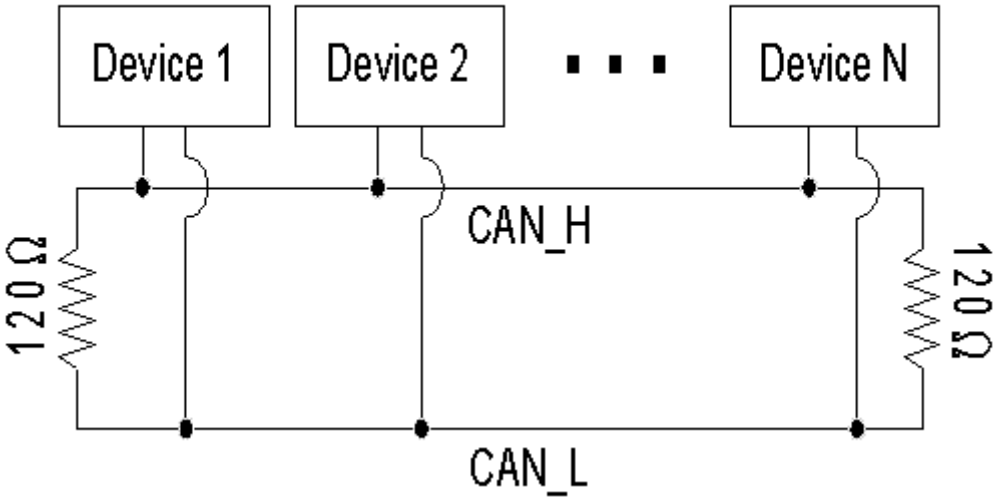


2.2 µPAC-7186EXD-CAN 硬體結構



2.3 CAN 網路線路連接

為了儘量減少反射效應，CAN bus 線路需要透過兩個終端電阻來終止運作，如下圖所示。根據 ISO 11898-2 規範，每一個終端電阻為 120Ω(歐姆) (或介於 108Ω～132Ω 之間)。CAN bus 線路的阻抗與長度有關，約為 70 mΩ/m。使用者在安裝 CAN 網路之前，應先確認 CAN bus 線路的阻抗。



而且，為了儘量減少長距離的電壓降，終端電阻應高於 ISO 11898-2 所制定的值，參考如下表所示。

Bus 長度 (公尺)	Bus Cable 參數		終端電阻 (Ω)
	阻抗 (mΩ/m)	橫切面 (Type)	
0~40	70	0.25(23AWG)~ 0.34mm^2(22AWG)	124 (0.1%)
40~300	< 60	0.34(22AWG)~ 0.6mm^2(20AWG)	127 (0.1%)
300~600	< 40	0.5~0.6mm^2 (20AWG)	150~300
600~1K	< 20	0.75~0.mm^2 (18AWG)	150~300

CAN bus 的鮑率與 bus 的長度有很高的關聯性，下表指出每一種鮑率相對應的 bus 長度。

鮑率 (bit/s)	最大長度 (m)
1 M	25
800 K	50
500 K	100
250 K	250
125 K	500
50 K	1000
20 K	2500
10 K	5000

注意: 當 bus 長度大於 1000m，需使用橋接器(bridge)或重置器(repeater)。

2.4 終端電阻跳線選擇

移除 I-7188XBD-CAN/µPAC-7186EXD-CAN 的上蓋，可看見其內部的部份結構，如圖 2.1 所示。 XC100 提供使用者一個終端電阻跳線選擇(J3)，它的位置如圖 2.1 所示。

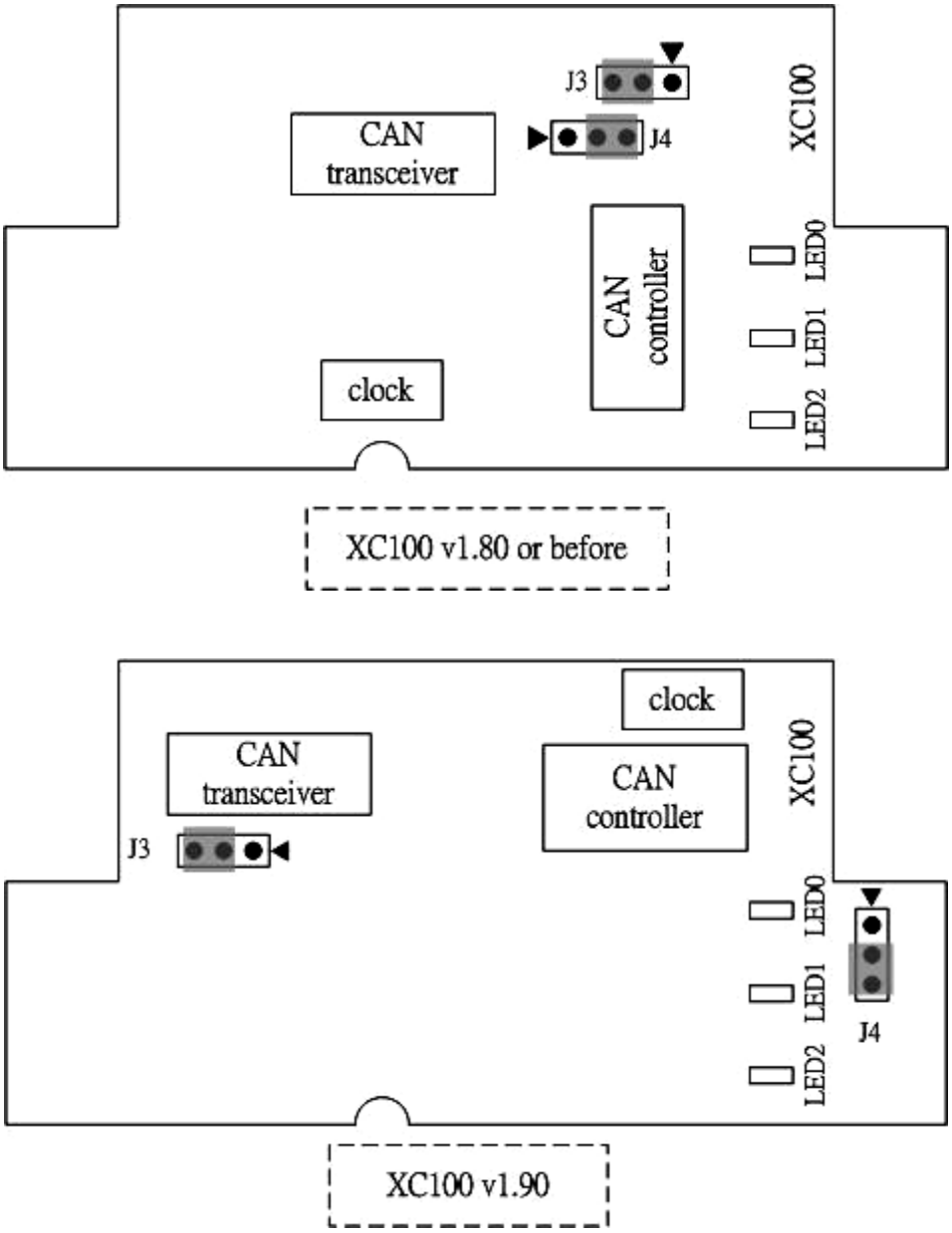


圖 2.1　XC100 I/O 擴充板卡元件分佈圖

J3 跳線器用來判斷 CAN 網路上，I-7188XBD-CAN/ µPAC-7186EXD-CAN 的終端電阻選擇與否，而 J3 跳線器相關設定，請參考表 2.1。

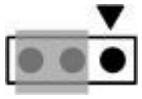
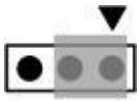
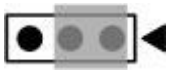
使用終端電阻(120Ω)		不使用終端電阻	
版本 1.80(含以前)	版本 1.90	版本 1.80(含以前)	版本 1.90
			

表 2.1　J3 跳線選擇

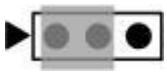
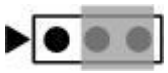
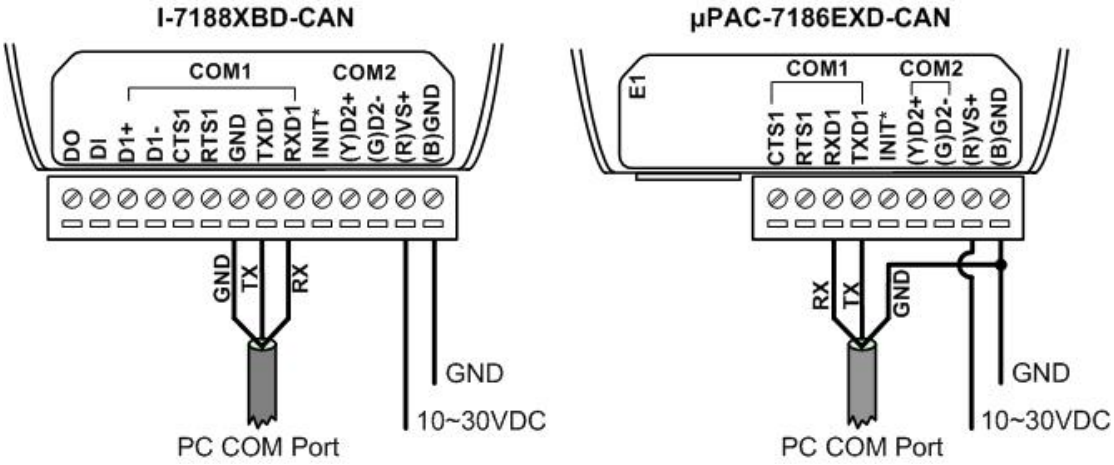
適用於 I-7188XBD-CAN		適用於 µPAC-7186EXD-CAN	
版本 1.80(含以前)	版本 1.90	版本 1.80(含以前)	版本 1.90
			

表 2.2　J4 跳線選擇

2.5 不同應用的接線圖

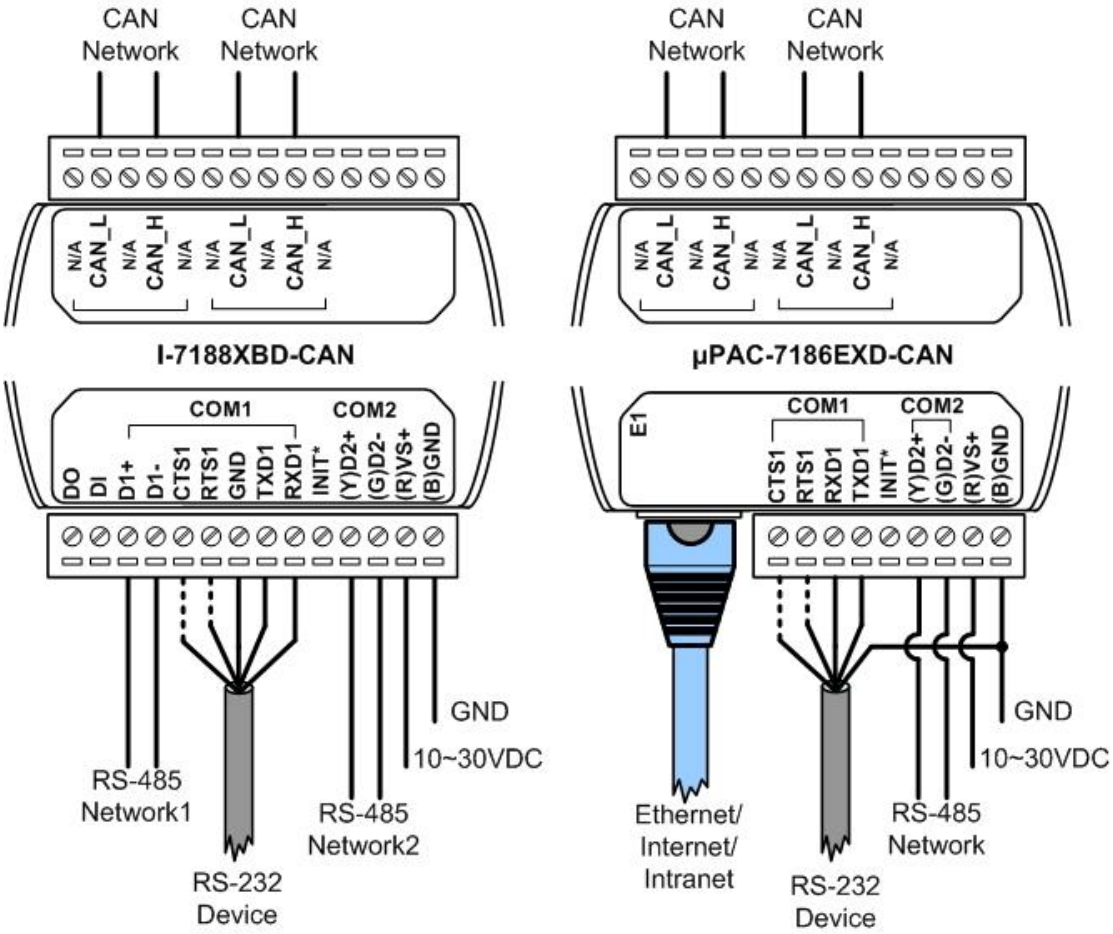
2.5.1 程式下載

如果想要載入使用者程式到 I-7188XBD-CAN/µPAC-7186EXD-CAN 上，使用者必須先利用下載線(I-7188XBD-CAN/µPAC-7186EXD-CAN 的配件)將 I-7188XBD-CAN/ µPAC-7186EXD-CAN COM1 與 PC 上可用的 COM 埠連接。接著利用 7188xw.exe 工具軟體從 OSImage 資料夾下載使用者程式，更詳細的操作設定，請參考第四章。

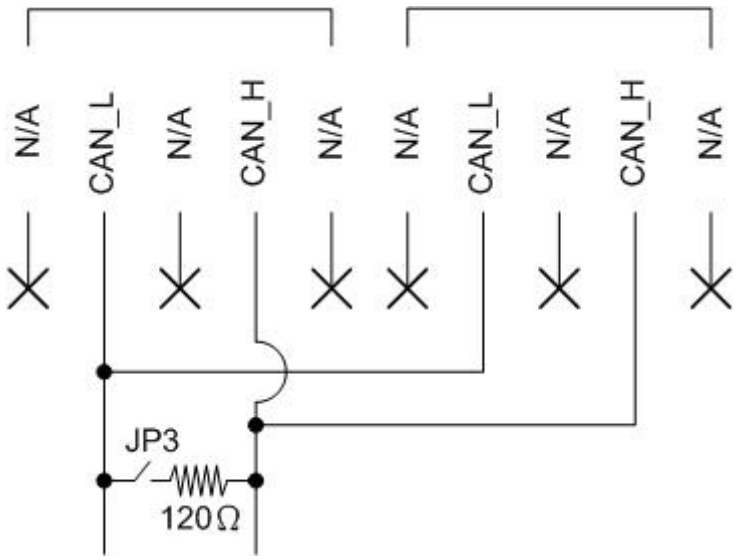


2.5.2 一般應用

下圖所示為一般應用接線圖。當資料經由 RS-232 與 RS-485 傳送到 I-7188XBD-CAN 的 COM1 時，將從 COM1 的 RS-232 或 RS-485 其中一個通訊埠接收資料。因此，不建議同時使用 COM1 的 RS-232 及 RS-485 功能。如果使用者選擇 COM1 的 RS-232 功能，其 RTS1 和 CTS1 腳位不是必要的，但必須檢查連結目標機器是採用 3 線或 5 線的 RS-232 來通訊。

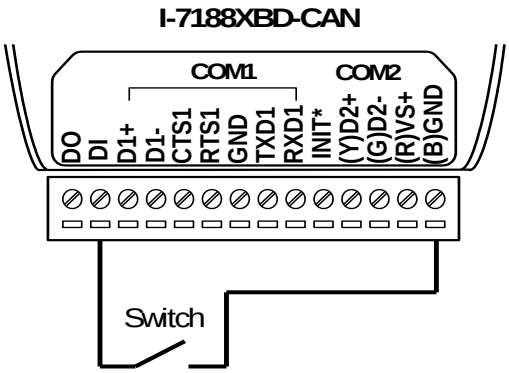


除此之外，為了接線方便，I-7188XBD-CAN/ µPAC-7186EXD-CAN 不只提供一個 CAN 埠，而且還有另一個旁路 CAN 埠。這兩個 CAN 埠是同一個，所以旁路 CAN 埠只是為了與另一個 CAN 設備方便接線，不具任何其他功能。

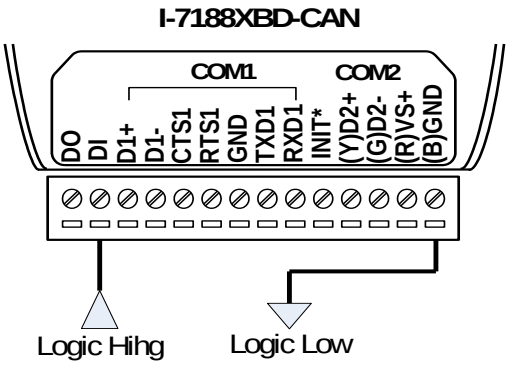


2.5.3 I-7188XBD-CAN DI & DO 通道接線圖

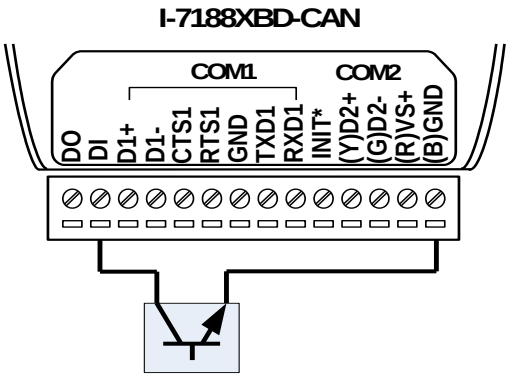
下圖所示為 I-7188XBD-CAN 中 DI 通道的一般線路連接方法。



Dry Contact Signal Input

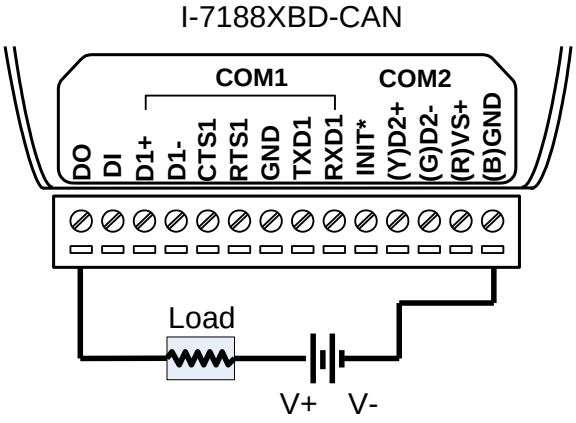


TTL/CMOS Signal Input



Open-Collector Signal Input

I-7188XBD-CAN 中 DO 通道的線路連接如下所示。



3 XC100 函式庫

XC100 子卡的函式庫“XC100L.lib”用來協助使用者設計不同的 CAN 設備。在這裡提供 TC、BC 及 MSC 的函式庫來接收、發送 CAN 訊息及配置 CAN 控制器。在這一章節，將告訴你函式庫有什麼功能及如何使用它。對於開發一個程式而言，可以參考圖 3.1 與圖 3.2。XC100L.lib 僅用於 XC100 子卡硬體，適用於 C/C++編譯器的 Large Mode。 有關程式程序逐步地操作資訊，請參考 4.1 節。

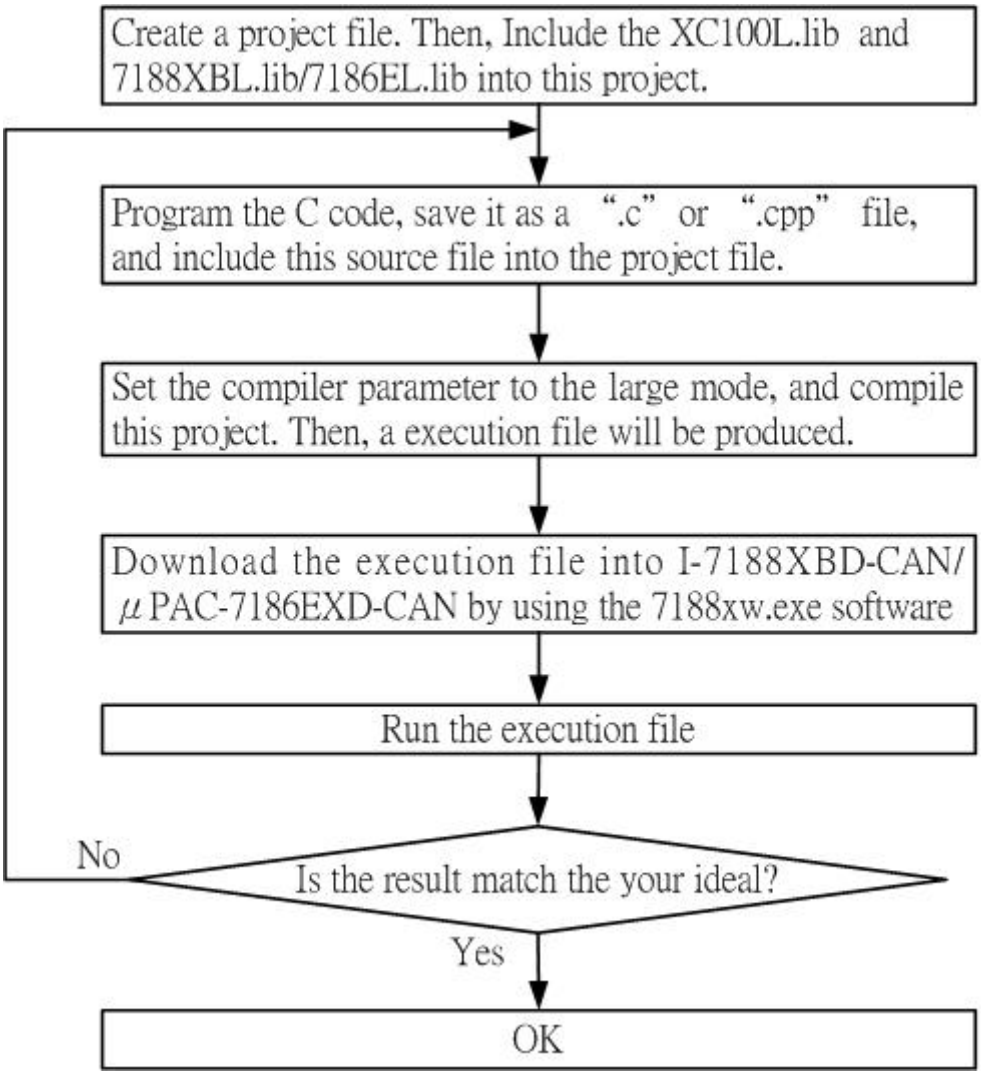


圖 3.1 程式程序

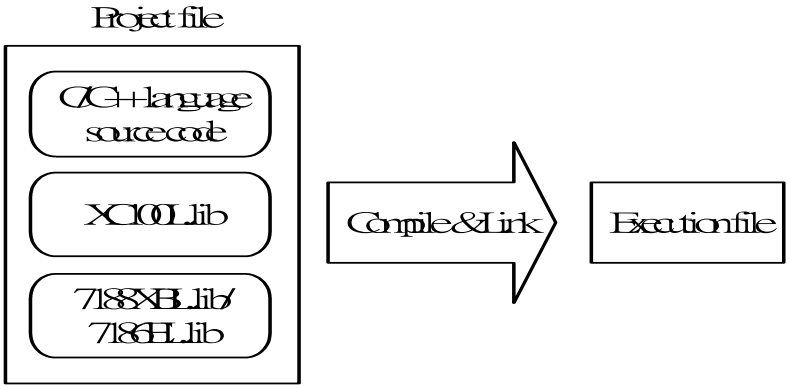


圖 3.2 編譯概念

由於本手冊是特別為 CAN 功能做說明，若使用者想了解其他功能與 I-7188XBD-CAN/µPAC-7186EXD-CAN 的範例(例如 EEPROM 功能、7 段顯示器功能、快閃記憶體功能功能、RTC(實時時鐘)功能、看門狗功能、COM 埠功能、等等)，請參考 I-7188XB(D)/I-7188EX(D) 的使用者手冊，或參考 7000/7188/8000 產品光碟中的線上說明。在線上說明裡，使用者可以找到許多關於 MiniOS7 操作命令、I-7188XBD-CAN/ µPAC-7186EXD-CAN 的其他功能、及下載工具軟體“7188xw.exe”的功能。MiniOS7 是 I-7188XBD-CAN/ µPAC-7186EXD-CAN 的作業系統，是一個類似 DOS 的作業系統。“7188xw.exe”是一個下載工具，提供使用者下載使用者程式到 I-7188XBD-CAN/ µPAC-7186EXD-CAN。使用者可於 CAN 光碟中的 uPAC-7186EXD-CAN-OS-Image 資料夾內取得下載工具“7188xw.exe”，它的路徑為“CAN/PAC/uPAC-7186EXD-CAN”。 **(注意: I-7188EX(D)範例可以在 µPAC-7186EXD-CAN 上使用。使用者只需用 7186EL.lib 再一次編譯 I-7188EX(D)範例。I-7188XBD-CAN 與 µPAC-7186EXD-CAN 的功能明說，可參考線上說明，如下所示。)**

關於線上說明：

8000cd /napdos/7188xabc/7188xb/document/ (適用於 I-7188XBD-CAN)
產品光碟/can/pac/i-7188xbd-can/document
8000cd /napdos/7186e/document/ (適用於 uPAC-7186EXD-CAN)
產品光碟/can/pac/upac-7186exd-can/document/

關於範例：

8000cd /napdos/7188xabc/7188xb/demo/ (適用於 I-7188XBD-CAN)
產品光碟/can/pac/i-7188xbd-can/demo/
8000cd /napdos/7186e/demo/ (適用於 uPAC-7186EXD-CAN)
產品光碟/can/pac/upac-7186exd-can/document

3.1 函式庫定義與說明

表 3.1 介紹了 XC100 函式庫的所有函式，它們提供使用者來建構特有的 CAN 設備。對於每一個函式的詳細說明，請參考接下來的章節。

函式定義	說明	頁碼
CAN_Reset	CAN 控制器硬體重置	21
XC100Init XC100Init_Listen	XC100 硬體初始化並進入正常模式或監聽模式	22
SetCANBaud	更改 CAN 的鮑率	25
SetCANMask	更改 CAN 信息的濾波器	26
CAN_InstallIrq	啟用嵌入式控制器中斷	27
CAN_RemoveIrq	停用嵌入式控制器中斷	28
CAN_Restore	釋放資源並停用嵌入式控制器中斷程序	29
CAN_CreateBuffer	更改接收與傳輸緩衝區大小	30
SendCANMsg SendCANMsg_NonBlock	使用 Blocking 或 Non-Blocking 方式發送 CAN 信息到 CAN 網路	31
GetNonBlockTxBufferLockedCount	於 Non-Blocking 傳送訊息時，使用此功能以取得傳輸緩衝區閉鎖錯誤並重試的次數。	33
GetNonBlockTxIncompleteCount	於 Non-Blocking 傳送訊息時，使用此功能以取得傳輸未完成並重試的次數。	34
GetCANMsg	接收 CAN 訊息	35
GetStatus	取得 CAN 控制器動狀態並傳送接收/傳輸緩衝器狀態	37
ClearStatus	重置接收與傳輸緩衝區狀態	38
L1Off	關閉 LED0	39
L2Off	關閉 LED1	40
L3Off	關閉 LED2	41
L1On	開啟 LED0	42
L2On	開啟 LED1	43
L3On	開啟 LED2	44
UserCANInt	設計使用者自定中斷程序	45
CAN_SearchBaud	搜尋必要的 CAN Bus 鮑率	47
CAN_BusOff_Recovery	呼叫此功能以恢復 CAN 總線關閉(Bus Off)狀態	48

表 3.1 XC100 函式庫清單

3.1.1 CAN_Reset

- **說明:**

  藉由硬體電路重新啟動 CAN 控制器。在執行此函式後，CAN 控制器設定為初始狀態。更多關於此功能的資訊，請參考 SJA1000 控制器資料表，網址如下。

  http://www.semiconductors.philips.com/pip/SJA1000.html#datasheet

  .

- **語法:**

  void CAN_Reset(void)

- **參數:**

  無

- **回傳:**

  無

- **相關函式:**

  無

3.1.2 XC100Init / XC100Init_Listen

■ **說明:**

➢XC100Init:

初始化軟體緩衝區及 XC100 硬體，其中包含 CAN 控制器、L1(LED)、L2(LED)及 L3(LED)。

➢XC100Init_Listen:

初始化軟體緩衝區及 XC100 硬體以進入監聽模式(Listen only mode)，其中包含 CAN 控制器、L1(LED)、L2(LED)及 L3(LED)。

此功能於 XC100 函式庫版本 1.80(含以後)支援。

■ **語法:**

```
int XC100Init(int TypeOf7188,char IntMode, unsigned long CANBaud,
              char BT0, char BT1,
              unsigned long AccCode, unsigned long AccMask)

int XC100Init_Listen(
              int TypeOf7188,char IntMode, unsigned long CANBaud,
              char BT0, char BT1,
              unsigned long AccCode, unsigned long AccMask)
```

■ **參數:**

➢TypeOf7188: 定義使用何種模組。

value	TypeOf7188
0	代表 I-7188XBD-CAN
1	代表 µPAC-7186EXD-CAN

➢IntMode: 設定 CAN 控制器中斷模式。IntMode 參數的每一位元表示不同的功能，如下所示。

中斷種類	IntMode 值
啟用接收中斷	0x01
啟用傳輸中斷	0x02
啟用錯誤警告中斷	0x04
啟用資料溢位中斷	0x08
啟用喚醒中斷	0x10
啟用 Error Passive 中斷	0x20
啟用仲裁遺失中斷	0x40
啟用 Bus 錯誤中斷	0x80

中斷類型	意義
Receive Interrupt	當一個信息正確無誤的被接收時,將觸發接收中斷。
Transmit Interrupt	當一個信息成功傳送或傳輸緩衝器是可再存取的狀態時，將觸發傳輸中斷。
Error Warning Interrupt	如果錯誤或 bus 狀態被設定或清除時，將觸發錯誤中斷。
Data Overrun Interrupt	如果在 FIFO (FIFO 有 64 bytes)中沒有足夠的空間，而造成信息遺失時，將觸發溢位中斷。
Wake-up Interrupt	當 CAN 控制器處於睡眠狀態，且偵測到 bus 活動時，將觸發喚醒(Wake-up)中斷。
Error Passive Interrupt	如果CAN控制器至少有一個錯誤計數器超過協議定義的 127 階，或 CAN 控制器處於 error passive 狀態時，將觸發 Error Passive 中斷。
Arbitration Lost Interrupt	當 CAN 控制器遺失一個仲裁，並且成為接收器時，將觸發仲裁遺失中斷。
Bus Error Interrupt	當 CAN 控制器在 CAN bus 上偵測到錯誤時，將觸發 Bus 錯誤中斷。

利用一個字元的值來實現中斷。例如，如果在 BasicCAN(CAN 2.0A)模式中，是需要接收與溢位中斷時，則設定 IntMode 值為 0x09(意謂著 0x01+0x08)。

- CANBaud: 使用一個長整數(long int)來設定此參數。例如，如果使用者欲設定 CAN 的鮑率為 125K bps 時,則設此參數值為 125000UL。

- BT0, BT1: 自定鮑率設定。使用者可藉由此參數任意的設定鮑率，但前提是需具有 SJA1000 CAN 控制器與 82C251 CAN 收發器，且自行運算 BT0 與 BT1 的值(CAN 控制器的時脈頻率為 16MHz)。

- AccCode, AccMask: AccCode 用來決定 CAN 控制器將接受何種 ID 類型。 而 AccMask 用來決定 ID 信息的那些位元需使用 AccCode 來檢查。如果 AccMask 的任一位元被設定為 0 時，這意謂著 ID 信息相同位元值需檢查，並且需要與 AccCode 的相同位元匹配。

關於 11-bit ID 訊息:

暫存器	暫存器位元	濾波器目標
AccCode[0] and AccMask[0]	bit7~bit0	ID 的 bit10 ~ bit3
AccCode[1] and AccMask[1]	bit7~bit5	ID 的 bit2 ~ bit0
AccCode[1] and AccMask[1]	bit4	RTR
AccCode[1] and AccMask[1]	bit3~bit0	未使用
AccCode[2] and AccMask[2]	bit7~bit0	第一個字元資料的 bit7 ~ bit0
AccCode[3] and AccMask[3]	bit7~bit0	第二個字元資料的 bit7 ~ bit0

關於 29-bit ID 訊息:

暫存器	暫存器位元	濾波器目標
AccCode[0] and AccMask[0]	bit7~bit0	ID 的 bit28 ~ bit21
AccCode[1] and AccMask[1]	bit7~bit0	ID 的 bit20 ~ bit13
AccCode[2] and AccMask[2]	bit7~bit0	ID 的 bit12 ~ bit5
AccCode[3] and AccMask[3]	bit7~bit3	ID 的 bit4 ~ bit0
AccCode[3] and AccMask[3]	bit2	RTR
AccCode[3] and AccMask[3]	bit1~bit0	未使用

注意: 1. AccCode[0]指 AccCode 的最高有效位元而
AccCode[3]指 AccCode 的最低有效位元。
2. AccMask[0]指 AccMask 的最高有效位元而
AccMask[3]指 AccMask 的最低有效位元。
3. Bit10 為最高有效字元而 Bit0 為最低有效位元。

例如 (在 29 bit ID 訊息):

AccCode : 00h 00h 00h A0h
AccMask : FFh FFh FFh 1Fh
ID Value : ?? ?? ?? Ah and Bh 將被接受。(??:忽略)

(注意: "h" 表示該值為十六進制格式)

- **回傳:**

  CAN_NoError: 正常。

  CAN_BaudNotSupport: CAN 控制器不支援此鮑率。

  CAN_ResetError: 重新啟動 CAN 控制器失敗。

  CAN_ConfigError: CAN 控制器暫存器配置失敗。

  CAN_SetACRError: AccCode 暫存器設定失敗。

  CAN_SetAMRError: AccMask 暫存器設定失敗。

  CAN_NotEnoughMemory:建立 CAN 信息的接收/傳輸軟體緩衝器失敗。

  CAN_TypeOf7188Error: I-7188 型號無此函式庫的定義。

- **相關函式:**

  無

3.1.3 SetCANBaud

■ **說明:**

在呼叫 XC100init 函式後，藉由此函式來更改 CAN 的鮑率。

■ **語法:**

int SetCANBaud(unsigned long CANBaud, char BT0, char BT1)

■ **參數:**

➢ CANBaud, BT0, BT1: 請參考 3.1.2 節的 XC100Init 函式的參數設明。

■ **回傳:**

CAN_NoError: 正常。

CAN_BaudNotSupport: CAN 控制器不支援此鮑率。

CAN_ResetError: CAN 控制器無法進入重置模式。因此，無法正常的設定所有參數。

■ **相關函式:**

3.1.2 XC100Init / XC100Init_Listen

3.1.4 SetCANMask

- **說明:**

  在使用 XC100init 函式後,可藉由此函式來更改 CAN 信息濾波器的設定。

- **語法:**

  int SetCANMask(unsigned long AccCode, unsigned long AccMask)

- **參數:**

  - AccCode, AccMask: 請參考 3.1.2 節 XC100Init 函式的參數說明。

- **回傳:**

  CAN_NoError: 正常。

  CAN_ResetError: CAN 控制器無法重置。

  CAN_SetACRError: AccCode 暫存器設定失敗。

  CAN_SetAMRError: AccMask 暫存器設定失敗。

- **相關函式:**

  3.1.2 XC100Init / XC100Init_Listen

3.1.5 CAN_InstallIrq

- **說明:**

  啟用 IRQ 中斷功能。然後，I-7188 系列的 CPU 嵌入的控制器可以接收 CAN 控制器發送的中斷信息。

- **語法:**

  void CAN_InstallIrq(void)

- **參數:**

  無

- **回傳:**

  無

- **相關函式:**

  3.1.6 CAN_RemoveIrq

3.1.6 CAN_RemoveIrq

- **說明:**
  停用 IRQ 中斷功能。然後，I-7188 系列的 CPU 嵌入的控制器不能接收 CAN 控制器發送的中斷信息。

- **語法:**
  void CAN_RemoveIrq(void)

- **參數:**
  無

- **回傳:**
  無

- **相關函式:**
  3.1.5 CAN_InstallIrq

3.1.7 CAN_Restore

- **說明:**

  停用中斷功能，釋放所有軟體緩衝區，並且重置 CAN 晶片。在程式中止前，必須呼叫此函式來釋放系統資源。

- **語法:**

  void CAN_Restore(void)

- **參數:**

  無

- **回傳:**

  無

- **相關函式:**

  無

3.1.8 CAN_CreateBuffer

- **說明:**

  此函式為改變接收與傳輸軟體緩衝區大小。如果使用者不使用此函式，其緩衝區預設值皆為 256 筆記錄。

- **語法:**

  int CAN_CreateBuffer(int BufMode, unsigned int BufferSize)

- **參數:**

  - BufMode: “0” 表示為改變接收軟體緩衝區大小，其餘則表示為改變傳輸軟體緩衝區大小。
  - BufferSize: 軟體緩衝區的新空間大小。

- **回傳:**

  CAN_NoError: 正常。

  CAN_NotEnoughMemory: 建立 CAN 信息的接收/傳輸軟體緩衝器失敗。

- **相關函式:**

  3.1.2 XC100Init / XC100Init_Listen

3.1.9 SendCANMsg / SendCANMsg_NonBlock

■ **說明:**

➢ SendCANMsg():

如果傳輸緩衝區被停用時，此函式將發送信息到 CAN 網路。然而，若是傳輸緩衝區被啟用，此函式將發送儲存在緩衝區內的所有信息到 CAN 網路。

➢ SendCANMsg_NonBlock():

如果傳輸緩衝區被停用時，此函式將發送信息到 CAN 網路。然而，若是傳輸緩衝區被啟用，此函式將回覆錯誤並通知使用者傳輸緩衝區被啟用。這意謂著使用此函式時不能使用傳輸緩衝區。此功能於 XC100 含式庫版本 1.80(含以後)支援。

■ **語法:**

```
int SendCANMsg(unsigned char Mode, unsigned long MsgID,
               unsigned char RTR, unsigned char DataLen,
               unsigned char *Data)

int SendCANMsg_NonBlock(
               unsigned char Mode, unsigned long MsgID,
               unsigned char RTR, unsigned char DataLen,
               unsigned char *Data)
```

■ **參數:**

➢ Mode: 此參數用來表示 CAN ID 的類型。

Mode 值	意義
0	發送 11-bit ID CAN 信息。
others	發送 29-bit ID CAN 信息。

➢ MsgID: CAN 信息的 ID。ID 值可能為 11-bit 或 29-bit。

➢ RTR: 遠端傳輸要求位元。

RTR 值	意義
0	此 CAN 信息非遠端傳輸要求信息。
1	此 CAN 信息為遠端傳輸要求信息。

➢ DataLen: CAN 信息的純資料長度，此值的範圍介於 0~8。

➢ *Data: CAN 信息的資料字元，其字元數需與"DataLen"所設定的相同。

■ **回傳:**

CAN_NoError: 正常。

CAN_DataLengthError: CAN 信息的資料長度超過 8 個字元。

CAN_TransmitBufferLocked: CAN 控制器的傳輸緩衝區被鎖住。

CAN_TransmitIncomplete: CAN 無法成功發送信息。

■ **相關函式:**

3.1.2 XC100Init / XC100Init_Listen

3.1.10 GetNonBlockTxBufferLockedCount

- **說明:**

  此函式為在 non-blocking 模式下傳送 CAN 訊息並取得傳輸緩衝區閉鎖次數。在使用 SendCANMsg_NonBlock 後，使用者可使用此功能以取得傳輸緩衝區閉鎖錯誤的重試次數。此功能於 XC100 含式庫版本 1.80(含以後)支援。

- **語法:**

  unsigned long GetNonBlockTxBufferLockedCount(void) 。

- **參數:**

  無。

- **回傳:**

  於 non-blocking 模式下，取得傳輸緩衝區閉鎖錯誤的重試次數。

- **相關函式:**

  3.1.9 SendCANMsg / SendCANMsg_NonBlock。

3.1.11 GetNonBlockTxIncompleteCount

- **說明:**

  此函式為在 non-blocking 模式下傳送 CAN 訊息並取得傳輸未完成狀態的次數。在使用 SendCANMsg_NonBlock 後，使用者可使用此功能以取得傳輸未完成狀態的重試次數。此功能於 XC100 含式庫版本 1.80(含以後)支援。

- **語法:**

  unsigned long GetNonBlockTxIncompleteCount(void)。

- **參數:**

  None。

- **回傳:**

  於 non-blocking 模式下，取得傳輸未完成狀態的次數。

- **相關函式:**

  3.1.9 SendCANMsg / SendCANMsg_NonBlock。

3.1.12 GetCANMsg

■ **說明:**

從接收緩衝區或直接從 CAN bus 接收 CAN 信息。如果在 XC100Init 函式的 IntMode 參數設定接收中斷為啟用，則此函式將自動讀回儲存在軟體緩衝區的 CAN 信息。反之，若接收中斷設為停用，此函式使用輪循方法檢查 CAN 晶片緩衝區內，是否有任何 CAN 信息，若有則將該信息回傳。

■ **語法:**

int GetCANMsg(unsigned char *Mode, unsigned long *MsgID
, unsigned char *RTR, unsigned char *DataLen
, unsigned char *Data, unsigned long *UpperTime
, unsigned long *LowerTime)

■ **參數:**

- *Mode: 此參數用來取得 CAN 信息的 ID 類型(11-bit ID 或 29-bit ID)。
- *MsgID: 取得 CAN 信息的 ID。
- *RTR: 取得 CAN 信息的 RTR。

RTR 值	意義
0	此 CAN 信息非遠端傳輸要求信息。
1	此 CAN 信息為遠端傳輸要求信息。

- *DataLen: 取得 CAN 信息的資料長度。
- *Data: 取得 CAN 信息的資料。資料緩衝區的大小必須是 8 個字元。
- *UpperTime: 取得 CAN 信息的時間標記，單位為 us (micro second)。此參數只顯示上部分的時間標記。

  即時標記(Real time stamp) = 上半部分(upper part)* 0x1000000UL+下半部分(lower part)

- *LowerTime: 取得 CAN 信息的下半部分的時間標記。

■ **回傳:**

CAN_NoError: 正常。

CAN_ReceiveBufferEmpty: CAN 接收緩衝區無信息。

CAN_SoftBufferIsEmpty: 軟體接收緩衝區無信息。

CAN_DataLengthError: 接收的信息長度超過 8 個字元。

■ **相關函式:**

3.1.2 XC100Init / XC100Init_Listen

3.1.13 GetStatus

■ **說明:**

讀取 CAN 控制器狀態與軟體緩衝區溢位旗標信息。

■ **語法:**

void GetStatus(unsigned char *CANReg, unsigned char *OverflowFlag)

■ **參數:**

➢ *CANReg: 此指標為取得目前 CAN 控制器狀態。有關於 CANReg 值所代表的意義，請參考如下表格。

Bit NO.	說明
7 (MSB)	Bus 狀態。“1”為 bus off；“0”為 bus on。
6	錯誤狀態。“1”為至少一個錯誤；“0”為 OK。
5	傳送狀態。“1”為傳送中；“0”為閒置。
4	接收狀態。“1”為接收中；“0”為閒置。
3	傳送完成狀態。“1”為完成；“0”為未完成。
2	傳送緩衝區狀態。“1”為釋放；“0”為鎖住。
1	資料溢位狀態。“1”為接收緩衝區溢位；“0”為正常。
0 (LSB)	接收緩衝區狀態。“1”為至少一個信息儲存在接收緩衝區；“0”為空的。

➢ *OverflowFlag: CAN 接收與傳輸溢位旗標資訊。有關於 OverflowFlag 值所代表的意義，請參考如下表格。

Bit NO.	說明
Others	保留
1	“1”為接收軟體緩衝區溢位；“0”為正常的。
0 (LSB)	“1”為傳輸緩衝器溢位；“0”為正常的。

■ **回傳:**

無

■ **相關函式:**

3.1.14 ClearStatus

3.1.14 ClearStatus

- **說明:**

  此函式用來清除 CAN 接收或傳輸軟體緩衝區溢位旗標。當任一個緩衝區滿了，其對應的溢位旗標被設為“1”。在此情形下，使用者需使用此函式來清除溢位旗標來回應錯誤資訊。

- **語法:**

  void ClearStatus(void)

- **參數:**

  無

- **回傳:**

  無

- **相關函式:**

  3.1.13 GetStatus

3.1.15 L1Off

- **說明:**
  關閉 L1 LED 燈。有關於 L1 LED 燈的位置，請參考 2.1 節中的圖 2.1。

- **語法:**
  void L1Off(void)

- **參數:**
  無

- **回傳:**
  無

- **相關函式:**
  3.1.18 L1On

3.1.16 L2Off

- **說明:**
  關閉 L2 LED 燈。有關於 L2 LED 燈的位置，請參考 2.1 節中的圖 2.1。

- **語法:**
  void L2Off(void)

- **參數:**
  無

- **回傳:**
  無

- **相關函式:**
  3.1.19 L2On

3.1.17 L3Off

- **說明:**
  關閉 L3 LED 燈。有關於 L3 LED 燈的位置，請參考 2.1 節中的圖 2.1。

- **語法:**
  void L3Off(void)

- **參數:**
  無

- **回傳:**
  無

- **相關函式:**
  3.1.20 L3On

3.1.18 L1On

- **說明:**
  開啟 L1 LED 燈。有關於 L1 LED 燈的位置，請參考 2.1 節中的圖 2.1。

- **語法:**
  void L1On(void)

- **參數:**
  無

- **回傳:**
  無

- **相關函式:**
  3.1.15 L1Off

3.1.19 L2On

- **說明:**
  開啟 L2 LED 燈。有關於 L2 LED 燈的位置，請參考 2.1 節中的圖 2.1。

- **語法:**
  void L2On(void)

- **參數:**
  無

- **回傳:**
  無

- **相關函式:**
  3.1.16 L2Off

3.1.20 L3On

- **說明:**
  開啟 L3 LED 燈。有關於 L3 LED 燈的位置，請參考 2.1 節中的圖 2.1。

- **語法:**
  void L3On(void)

- **參數:**
  無

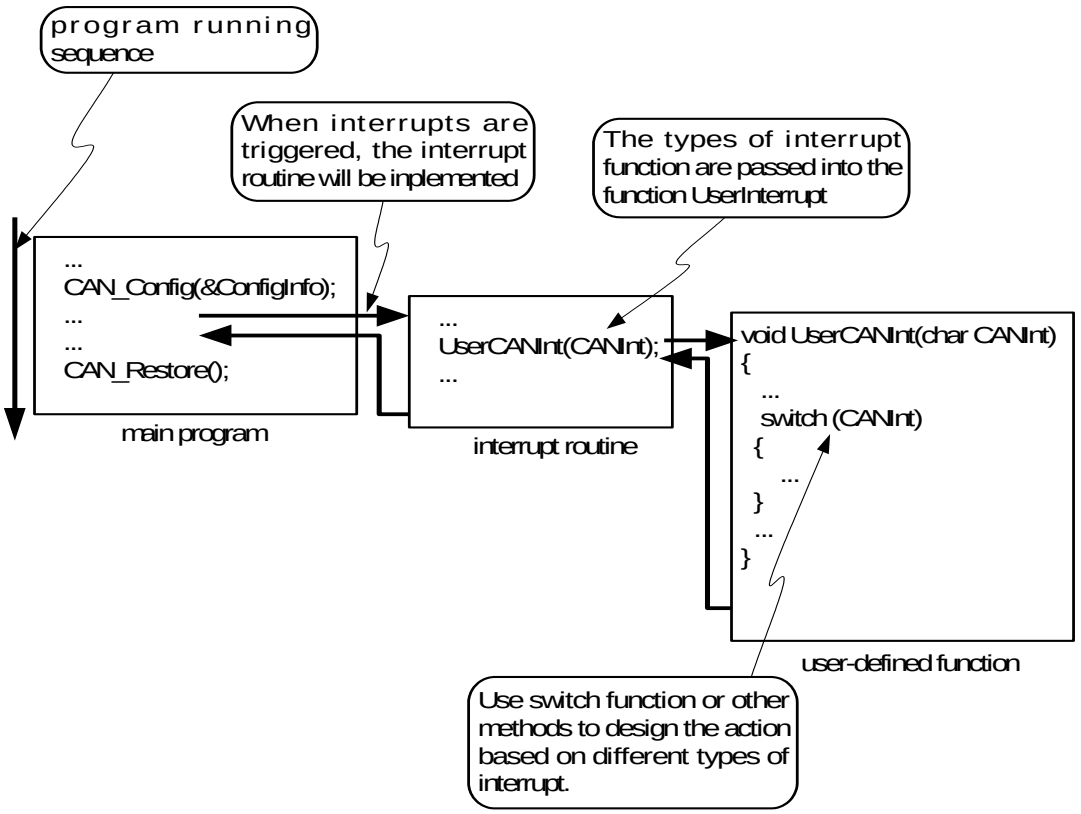
- **回傳:**
  無

- **相關函式:**
  3.1.17 L3Off

3.1.21 UserCANInt

■ **說明:**

此函式由使用者建立，並用於中斷服務程序的設計。當中斷功能被觸發時，自動地傳送參數 CANInt。這指出那一種 CAN 控制器中斷的種類被啟動了。因此，使用者只需根據不同的中斷功能設計它們的中斷程序。如果不使用此函式，請於使用者的.C 檔中註銷此函式，以避免編譯錯誤。下圖是 UserCANInt 函式的一般概念。



■ **語法:**

void UserCANInt(char CANInt)

■ **參數:**

➢ CANInt: 中斷服務程式將透過 CANInt 參數向使用者說明何種中斷被觸發。有關於 CANInt 參數的意義，請參考下表。

CANIntMode 值 (Hex)	意義
0x01	成功接收信息
0x02	成功傳送信息
0x04	錯誤警告
0x08	資料溢位
0x10	喚醒(wake-up)CAN 控制器
0x20	Bus Passive
0x40	仲裁遺失
0x80	Bus 錯誤

■ **回傳:**

無

■ **相關函式:**

3.1.2 XC100Init / XC100Init_Listen

3.1.22 CAN_SearchBaud

■ **說明:**

進入“監聽模式(Listen Only Mode)”，並且啟動接收與錯誤中斷來檢測 CAN bus 的正確位元率。當信息成功的接收時，將回傳“CAN_NoError”信息，否則回傳“CAN_AutoBaudTimeout”信息。

■ **語法:**

int CAN_SearchBaud(unsigned long CANBaud, char BT0,
char BT1,unsigned int Timeout)

■ **參數:**

- CANBaud: 使用長整數(long int)來設定此參數。例如，假設使用者欲設定 CAN 的鮑率為 125 K bps，此參數值設為 125000UL。

- BT0, BT1: 使用者自定鮑率設定。使用者可藉由此參數任意的設定鮑率，但前提是需具有 SJA1000 CAN 控制器與 82C251 CAN 收發器，且自行運算 BT0 與 BT1 的值(CAN 控制器的時脈頻率為 16MHz)。

- Timeout: 設定 timer 是為了搜尋必要的 CAN bus 鮑率。

■ **回傳:**

CAN_NoError: 正常。
CAN_ResetError: CAN 控制器重置失敗。
CAN_ConfigError: CAN 控制器暫存器配置失敗。
CAN_SetBaudRateError: CAN 鮑率設定失敗。
CAN_BaudNotSupport: 不支援此鮑率。
CAN_AutoBaudTimeout: 找不到必要的 CAN bus 鮑率。

■ **相關函式:**

無

3.1.23 CAN_BusOff_Recovery

- **說明:**

  當 CAN 總線進入總線關閉(Bus Off)狀態時，使用者可使用此函式以恢復總線狀態為主動狀態。此功能於 XC100 含式庫版本 1.80(含以後)支援。

- **語法:**

  void CAN_BusOff_Recovery(void)。

- **參數:**

  無。

- **回傳:**

  無。

- **相關函式:**

  3.1.9 SendCANMsg/ SendCANMsg_NonBlock。

3.2 回傳碼

回傳碼	錯誤 ID	註釋
0	CAN_NoError	正常
5	CAN_ResetError	進入重置模式錯誤
8	CAN_ConfigError	CAN 晶片配置錯誤
9	CAN_SetACRError	接受碼暫存器設定錯誤
10	CAN_SetAMRError	接受遮罩暫存器錯誤
11	CAN_SetBaudRateError	鮑率設定錯誤
14	CAN_InstallIrqFailure	中斷功能啟動失敗
15	CAN_RemoveIrqFailure	中斷功能停用失敗
16	CAN_TransmitIncomplete	無法成功地傳送資料
17	CAN_TransmitBufferLocked	資料傳輸尚未完成
18	CAN_ReceiveBufferEmpty	接收緩衝區內無信息
19	CAN_DataOverrun	軟體緩衝區無足夠空間，導致資料遺失
20	CAN_ReceiveError	資料接收未完成
21	CAN_SoftBufferIsFull	軟體傳輸緩衝區已滿
22	CAN_SoftBufferIsEmpty	軟體緩衝區內無信息
23	CAN_BaudNotSupport	不支援此鮑率
24	CAN_DataLengthError	資料長度與總資料字元不一致
25	CAN_NotEnoughMemory	記憶體空間不足，無法建立接收或傳輸軟體緩衝區
26	CAN_TypeOf7188Error	7188 的類型在此函式庫未定義
50	CAN_AutoBaudTimeout	找不到 CAN bus 鮑率
51	CAN_TxBufferNotZero	於傳輸緩衝未被關閉的情況下，呼叫 SendCANMsg_NonBlock 後，會回應此錯誤。
52	CAN_BusOff	當總線關閉時，呼叫 SendCANMsg 函式後，會回應此錯誤。

4 範例程式

I-7188XBD-CAN / uPAC-7186EXD-CAN 資料夾結構如下表所示。

```
|--\document                → Users manual
|--\OSimage                 → OS image used for testing demo
|--\demo                    → demo folier
   |--\LIB100               → BC++3.1 library folder
   |--\BCPP31               → BC++3.1 demo folder
      |--\AC_AM             → BC++3.1 AC_AM demo folder
      |--\All_Demo          → BC++3.1 All_Demo demo folder
      |--\L1_L2_L3          → BC++3.1 L1_L2_L3 demo folder
      |--\RxInt             → BC++3.1 RxInt demo folder
      |--\RxPoll            → BC++3.1 RxPoll demo folder
      |--\TxInt             → BC++3.1 TxInt demo folder
      |--\TxPoll            → BC++3.1 TxPoll demo folder
      |--\UserInt           → BC++3.1 UserInt demo folder
      |--\SearchCANBaud     → BC++3.1 SCH_Baud demo folder
   |--\TCPP31               → TC++1.01 demo folder
      |--\AC_AM             → TC++1.01 AC_AM demo folder
      |--\All_Demo          → TC++1.01 All_Demo demo folder
      |--\L1_L2_L3          → TC++1.01 L1_L2_L3 demo folder
      |--\RxInt             → TC++1.01 RxInt demo folder
      |--\RxPoll            → TC++1.01 RxPoll demo folder
      |--\TxInt             → TC++1.01 TxInt demo folder
      |--\TxPoll            → TC++1.01 TxPoll demo folder
      |--\UserInt           → TC++1.01 UserInt demo folder
      |--\SearchCANBaud     → TC++1.01 SCH_Baud demo folder
   |--\MSC                  → MSC 1.52 demo folder
      |--\AC_AM             → MSC 1.52 AC_AM demo folder
      |--\All_Demo          → MSC 1.52 All_Demo demo folder
      |--\L1_L2_L3          → MSC 1.52 L1_L2_L3 demo folder
      |--\RxInt             → MSC 1.52 RxInt demo folder
      |--\RxPoll            → MSC 1.52 RxPoll demo folder
      |--\TxInt             → MSC 1.52 TxInt demo folder
      |--\TxPoll            → MSC 1.52 TxPoll demo folder
      |--\UserInt           → MSC 1.52 UserInt demo folder
      |--\SearchCANBaud     → MSC 1.52 SCH_Baud demo folder
```

在這裡，提供分別使用 BC++3.1、TC++1.01 和 MSC 1.52 的 XC100 函式庫的範例程式。每一個範例程式的內容如下表所示。當使用者欲編譯範例程式，請複制範例資料夾到新的資料夾，且最多以 8 個字母來命名。BC++3.1/TC++1.01/MSC6 編譯器是 16 位元編譯器，且可能有一個長檔名的問題。µPAC-7186EXD-CAN 資料夾結構類似 I-7188XBD-CAN 資料夾結構，因此其結構如上所述。

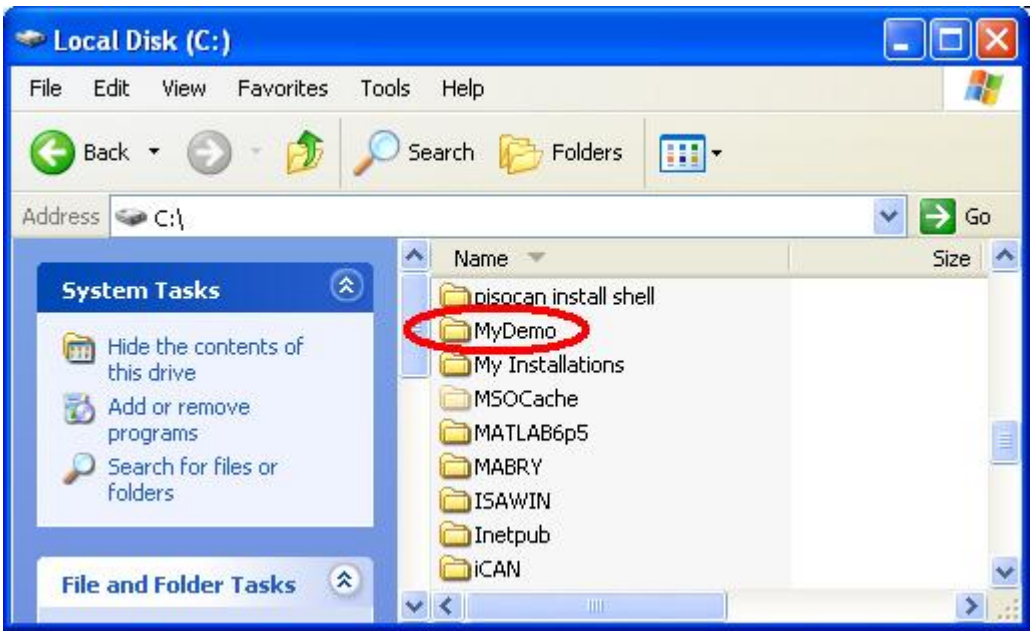
範例	內容
AC_AM	使用接受碼 AccCode 與接受遮罩 AccMask
All_Demo	藉由 XC100L.lib 提供所有函數的使用範例
L1_L2_L3	使用 L1、 L2、及 L3 LED 顯示器
RxInt	藉由中斷模式接收 CAN 信息
RxPoll	藉由輪循模式接收 CAN 信息
TxInt	藉由中斷模式發送 CAN 信息到 CAN 網路
TxPoll	藉由輪循模式發送 CAN 信息到 CAN 網路
UserInt	利用 “UserCANInt” 函式來實施使用者的 CAN 中斷服務程序
SCH_Baud	搜尋 CAN bus 鮑率的範例

為了清楚的介紹使用者如何使用 XC100 函式庫，我們將在接下來的章節逐步的介紹操作程序。使用者可以知道如何用 XC100L.lib 來建置一個執行檔、載入使用者程式及在 I-7188XBD-CAN/µPAC-7186EXD-CAN 上執行程式。

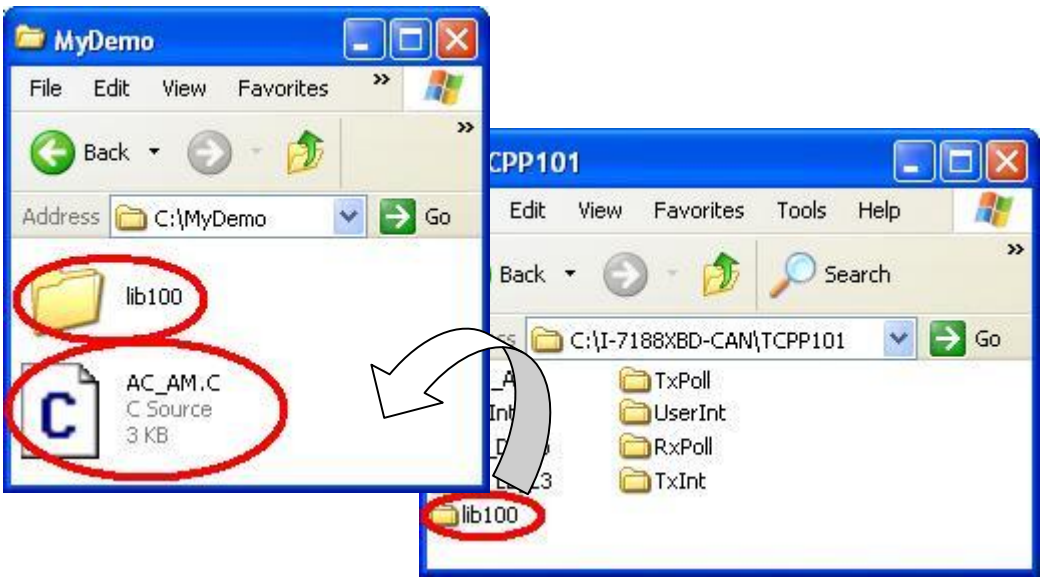
4.1 Program Download Procedure

在這裡，將說明如何使用 XC100L.lib 建立一個執行檔(.exe)，並且如何在 I-7188XBD-CAN/µPAC-7186EXD-CAN 上執行程式。

步驟 1: 在 C 槽(C:\)建立一個資料夾，並命名為“MyDemo”。

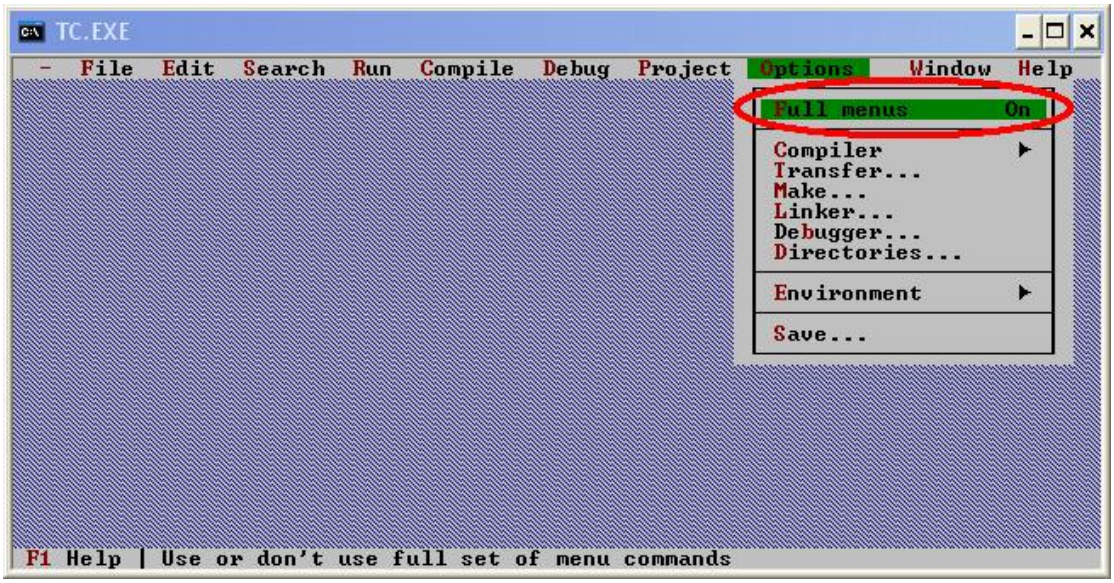


步驟 2: 複制“lib100”資料夾及使用者程式到“MyDemo”資料夾。

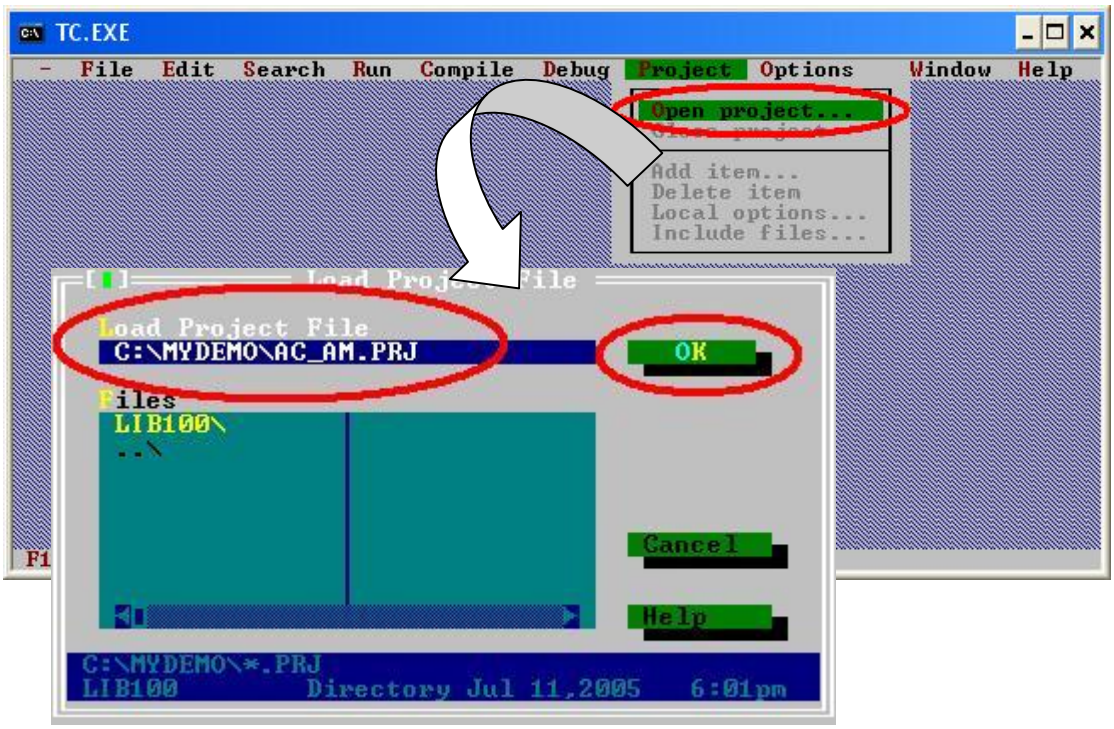


步驟 3: 開啟 TC++1.01 開發環境，點擊 “Options\Full menus” 展開所有功能項目。使用者可以免費下載 TC++1.01 開發軟體，網址如下：

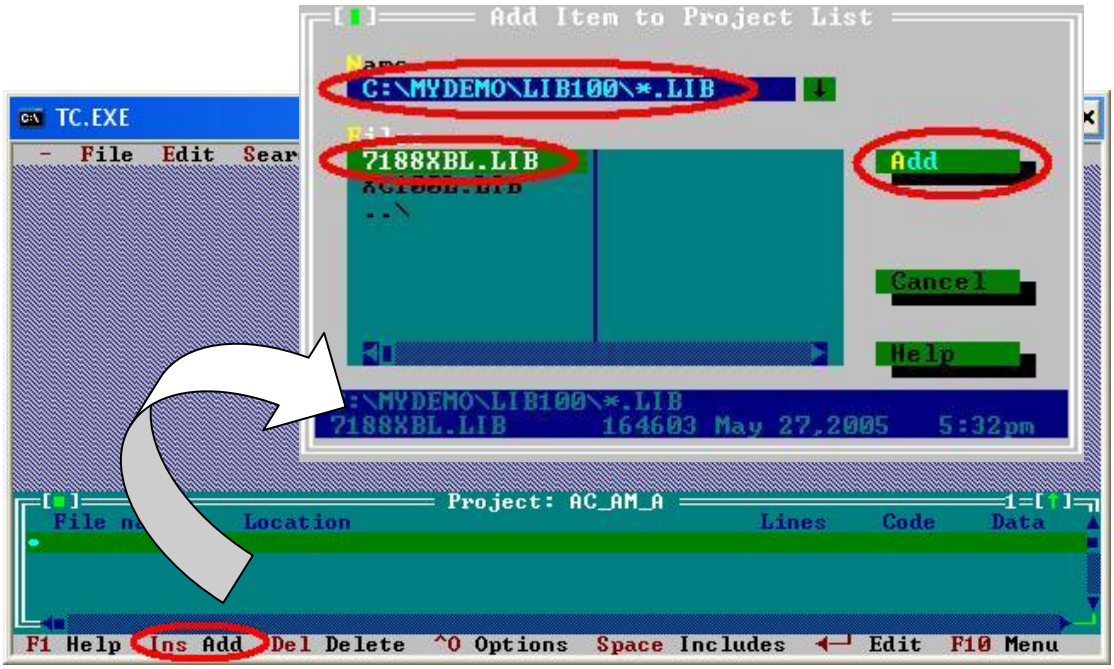
http://comsmunity.borland.com/museum



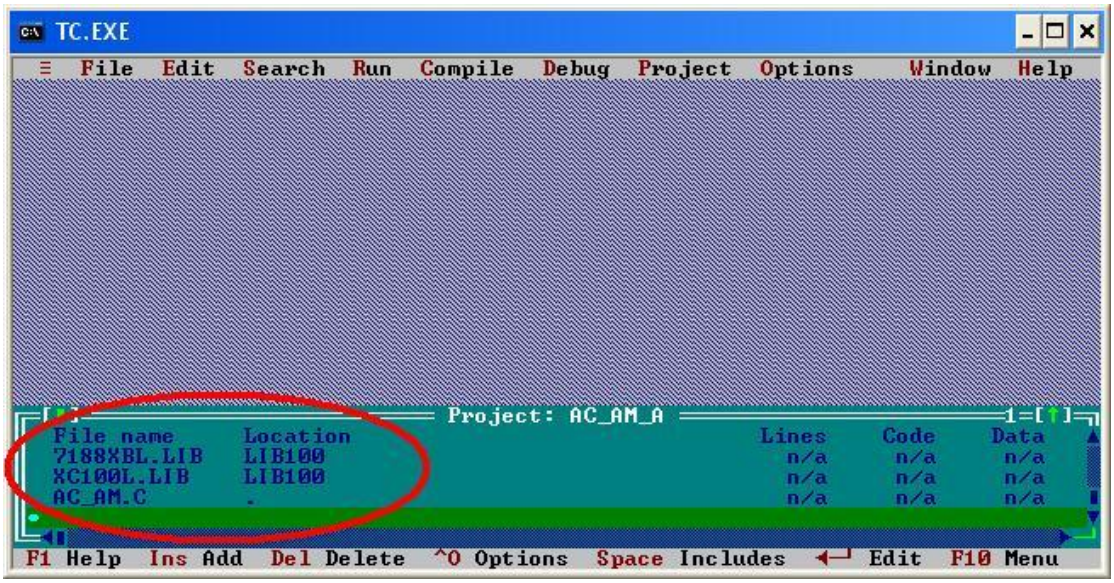
步驟 4: 點擊 “Project\Open project…” 建立一個新的專案，並命名為 “AC_AM.PRJ” 。



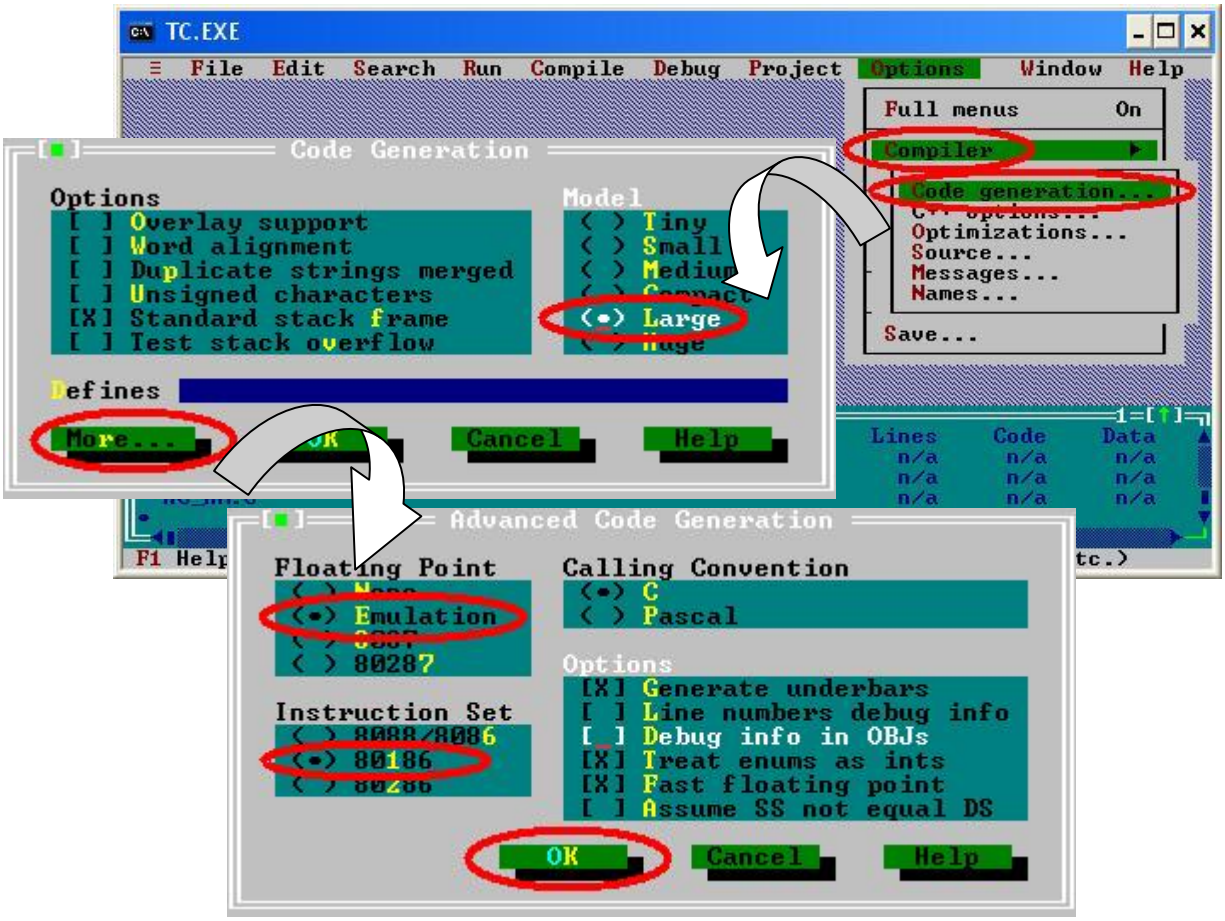
步驟 5: 於檔案名稱欄”Name”以 “ *.lib” 搜尋所有的函式庫檔案接著，點擊 “Add” 功能新增函式庫 “XC100L.lib” 到 “MyDemo” 專案。



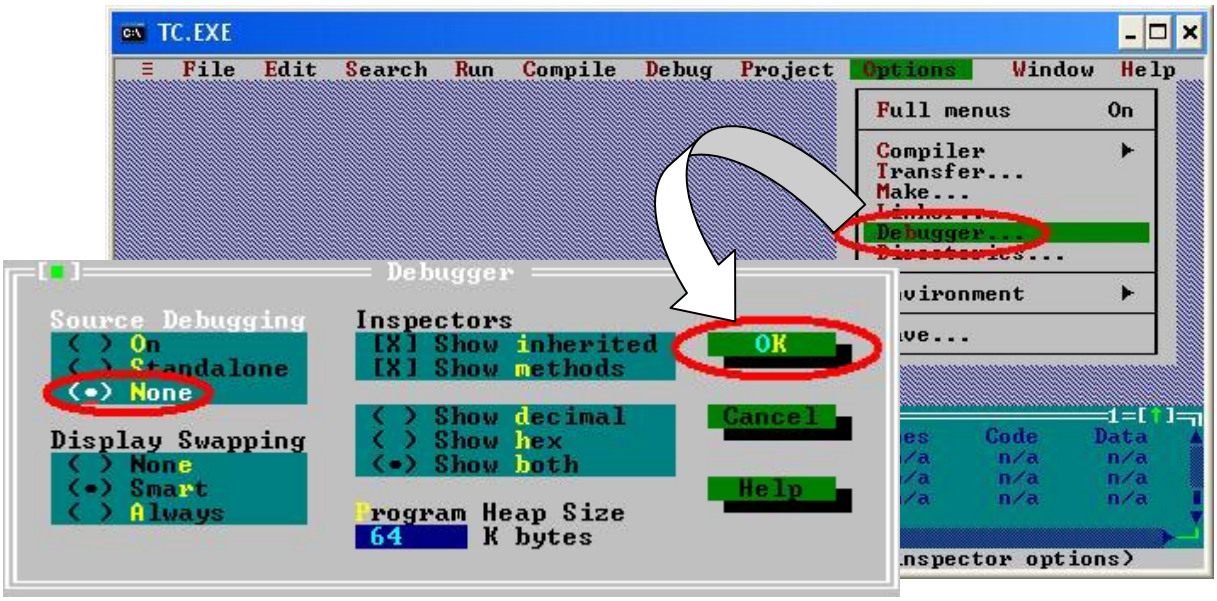
步驟 6: 以步驟 5 方法再新增另外兩個檔案。一個是 “7188XBL.lib” ，如果是使用 µPAC-7186EXD-CAN，則函式庫為 “7188XBL.lib” 。另一個是使用者的 C 源碼檔，在這裡我們使用 “AC_AM.c” 。



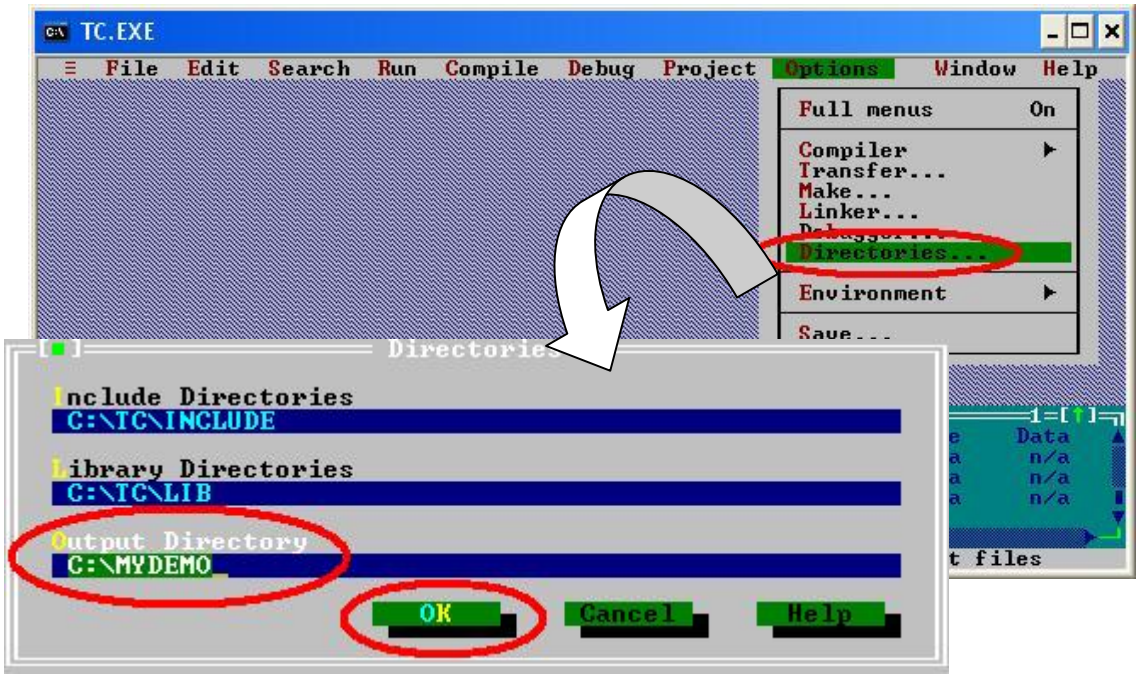
步驟 7: 點擊 “Options/Compiler/Code generation…” 來設定編譯模式為 “large” 。 然後，點擊 “More…” ，並分別設定 “Floating point” 與 “Instruction Set” 參數為 “Emulation” 及 “80186” 。接著點擊 “OK” 儲存組態設定。



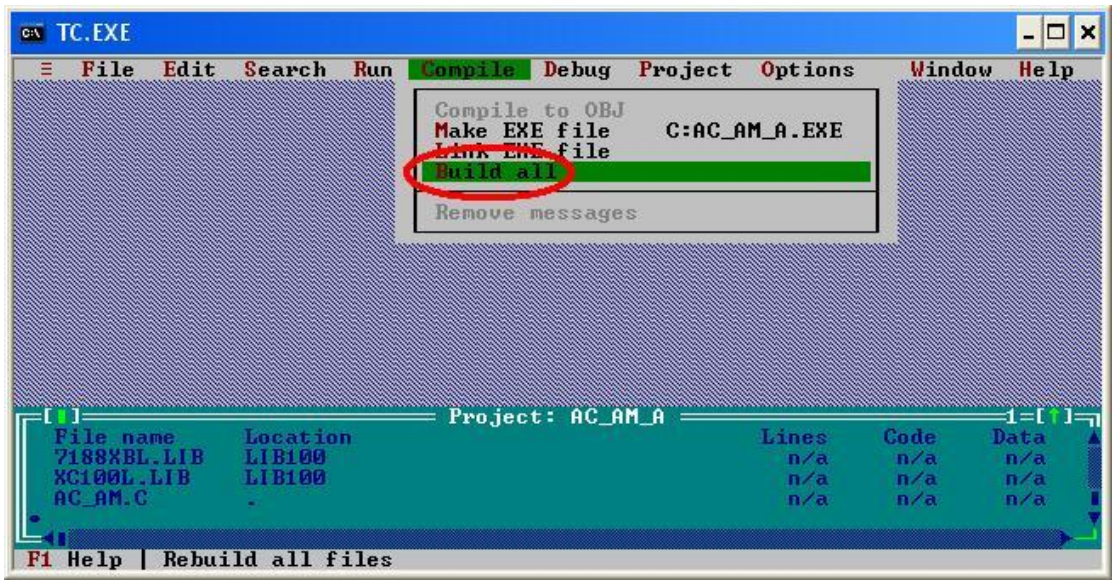
步驟 8: 點擊 “Option/Debugger...” 來設定 “Source Debugging” 的參數為 “None” 。



步驟 9: 點擊 “Option/Directories...” 來設定輸出目錄參數 “Output Directory” 。在此，設定輸出目錄參數 “Output Directory” 為 “C:\MyDemo” 。

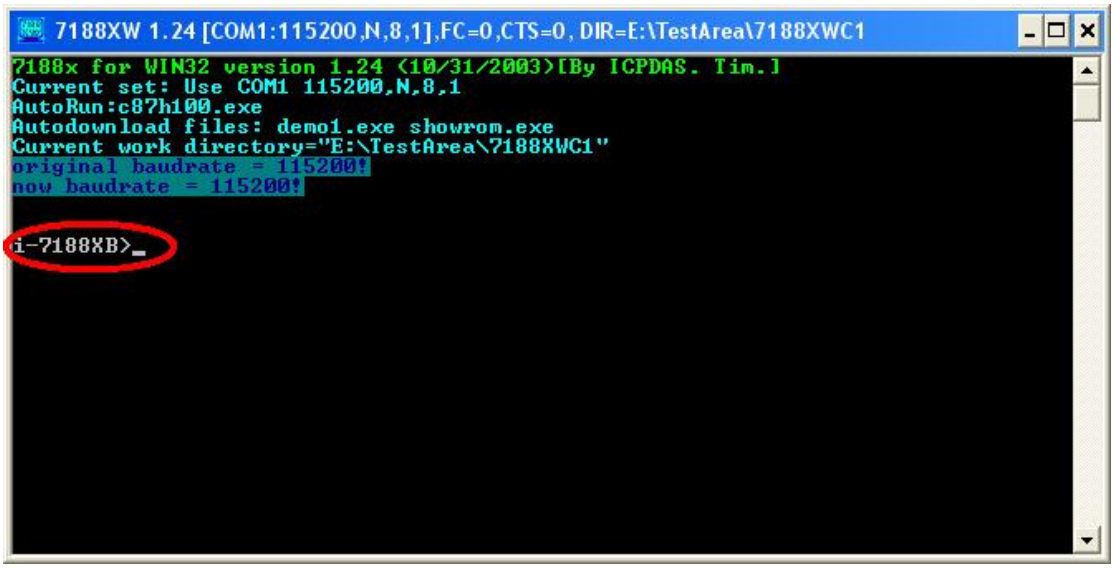


步驟 10: 在完成所有參數設定後，點擊 “Compile/build all” 來產生執行檔並命名為 “AC_AM.exe” 。

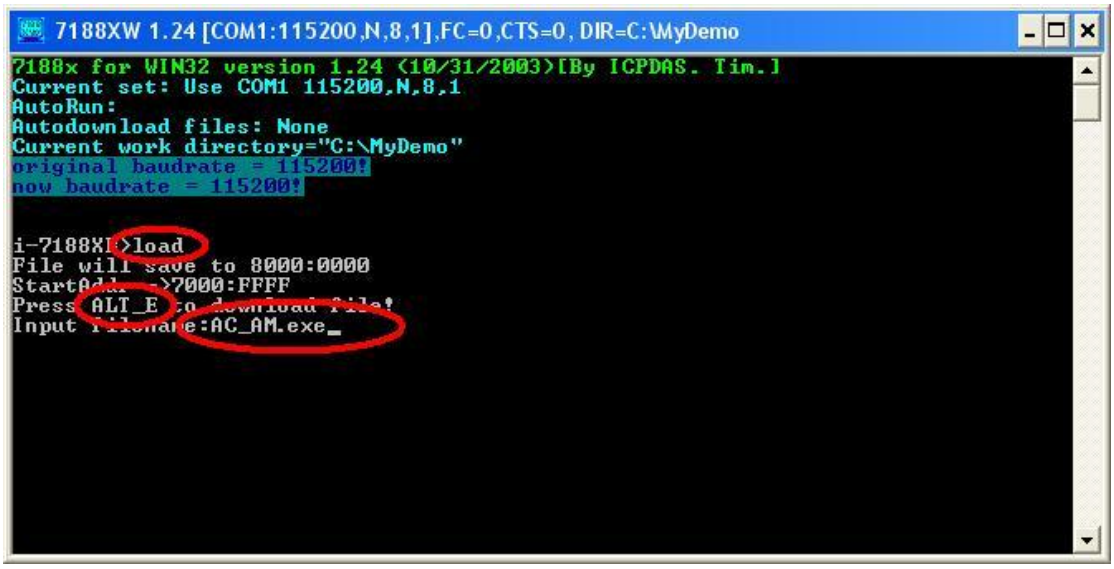


步驟 11: 複制執行檔“7188xw.exe”到“MyDemo”資料夾內，接著雙擊“7188xw.exe”。執行檔“7188xw.exe”可於“OSimage”資料夾內找到。

步驟 12: 如果 I-7188XBD-CAN 的 COM1 連接到 PC 的 COM1 時，於“7188xw.exe”程式視窗中按下“Enter”，便會出現提示標記“I-7188XB>”。若是使用 µPAC-7186EXD-CAN 的 COM1 連接到 PC 的 COM1，則提示標記顯示“uPAC-7186EXD_UDP>”。



步驟 13: 在“7188xw.exe”程式中輸入 “load” 指令，接著依提示指令按下“Alt+E”，並且輸入檔名“AC_AM.exe”來載入執行檔。



步驟 14: 在完成載入程序後，輸入 “run” 指令來執行“AC_AM.exe”。