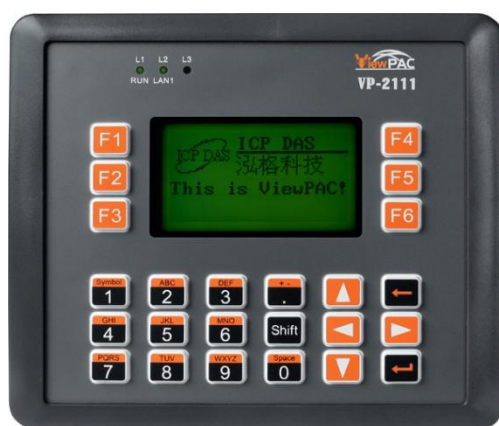


VP-2000/VH-2000 系列 (使用 C 語言，內建 MiniOS7) 使用手冊

版本: 1.0.3 · 2011 年 07 月



產品技術服務與使用資訊

VP-2111

VH-2110

保固說明

泓格科技股份有限公司 (ICP DAS) 所生產的產品，均保證原始購買者對於有瑕疵之材料，於交貨日起保有為期一年的保固。

免責聲明

泓格科技股份有限公司對於因為應用本產品所造成的損害並不負任何法律上的責任。本公司保留有任何時間未經通知即可變更與修改本文件內容之權利。本文所含資訊如有變更，恕不予另行通知。本公司盡可能地提供正確與可靠的資訊，但不保證此資訊的使用或其他團體在違反專利或權利下使用。此處包涵的技術或編輯錯誤、遺漏，概不負其法律責任。

版權所有

@ 2009 泓格科技股份有限公司保留所有權利。

商標識別

本文件提到的所有公司商標、商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

技術服務

如有任何問題，請與本公司客服聯絡，我們將盡速為您服務。

Email 信箱：service@icpdas.com

目錄

目錄	3
1. 簡介	6
1.1. ViewPAC 系列 (內建 MiniOS7)	7
1.2. 特色	8
1.3. 規格	9
1.4. 概述	11
1.5. 尺寸	13
1.6. 隨貨光碟	16
2. 快速上手	17
2.1. 硬體安裝	18
2.1.1. 架設硬體	19
2.1.2. 安裝 IP-65 防水等級的连接頭	25
2.2. 軟體安裝	26
2.3. 設定啟動模式	28
2.4. 上傳 ViewPAC 程式	29
2.4.1. 建立 PC 與 ViewPAC 之間的連線	30
2.4.2. 上傳並執行 ViewPAC 程式	39
2.4.3. 設定自動執行程式	40
2.5. 更新 ViewPAC 的 OS image	42

3. 第一個範例程式– “Hello World”	45
3.1. C 編譯器安裝	46
3.1.1. 安裝編譯器	47
3.1.2. 設定環境變數	51
3.2. ViewPAC 之應用程式介面 (API)	54
3.3. ViewPAC 中的第一個程式	55
4. API 與參考範例	66
4.1. COM Port 之 API	75
4.1.1. COM Port 函式類型	77
4.1.2. MiniOS7 COM Port 之 API	78
4.1.3. 標準 COM Port 之 API	81
4.1.4. COM Port 函式之對照	84
4.1.5. COM Port 請求/回應 的通訊協定定義	86
4.2. 用於 I/O 模組之 API	87
4.2.1. 於插槽上使用 I-8K 系列 I/O 模組 (適用型號: VP-2111)	89
4.2.2. 於插槽上使用 I-87K 系列 I/O 模組 (適用型號: VP-2111)	90
4.2.3. 使用連接於 COM Port 的 I-7K 與 I-87K 系列 I/O 模組	92
4.3. 用於 EEPROM 之 API	94
4.4. 用於快閃記憶體之 API	96
4.5. 用於 NVRAM 之 API	98
4.6. 用於計時器之 API	100
4.7. 用於看門狗計時器 (WDT) 之 API	102
4.8. MFS 之 API (適用型號: VP-2111)	104

附錄 A. 什麼是 MiniOS7?	110
附錄 B. 什麼是 MiniOS7 Utility?	111
附錄 C. 什麼是 MiniOS7 檔案系統 (MFS)? (適用型號: VP-2111)	112
附錄 D. I-8K 與 I-87K 系列模組 (適用型號: VP-2111)	116
附錄 E. RS-485 網路之應用.....	117
E.1. 基本的 RS-485 網路	118
E.2. 鏈狀 (Daisy Chain) RS-485 網路	119
E.3. 星狀 RS-485 網路.....	120
E.4. 隨機 RS-485 網路	122
E.5. Master/Slaves 設定	123
E.5.1. ViewPAC 作為 Master 設備 (預設).....	123
E.5.2. ViewPAC 作為 Slave 設備.....	125
附錄 F. 版本修訂紀錄.....	127

1. 簡介

ViewPAC 是一款結合了 iPAC、圖形顯示、按鍵功能為一體的控制器。它配備有一個 80186 處理器 (16 位元, 80 MHz) 並採用 MiniOS7 作業系統, 包含了數種通訊介面 (Ethernet, RS-232/485)、3 個用來擴充 I/O 的模組插槽、STN LCD 螢幕和橡膠按鍵。



MiniOS7 作業系統可以在極短的時間內 (0.4 ~ 0.8 秒) 完成啟動。它內建有硬體檢測功能, 並且提供完整的函式來存取 I-8K 和 I-87K 系列的 I/O 模組。例如: 數位輸入、數位輸出、數位輸入/輸出、類比輸入、類比輸出、頻率/計數、運動控制...等等的模組。

相較於傳統的 HMI + PLC 方案, ViewPAC 將為您降低整體系統的成本與空間, 並提供最佳的 HMI 和 PLC 功能。

1.1. ViewPAC 系列 (內建 MiniOS7)

ViewPAC 可依其功能區分為兩種類型：

- ▶ VP-2111
- ▶ VH-2110

ViewPAC 比較表

下列提供了 ViewPAC 型號之規格比較表：

	VP-2111	VH-2110
作業系統	MiniOS7	
處理器(CPU)	80 MHz	
Flash	512 KB	
RAM	768 KB	512 KB
雙電池備援 SRAM	512 KB	-
Flash Disk	64 MB	-
STN LCD 解析度	128 x 64	
Ethernet	1	
RS-232/RS-485	3	
I/O 插槽	3	-

1.2. 特色

硬體與軟體的主要特色。

軟體特色

- ▶ 內嵌式 MiniOS7 作業系統 (DOS-like)
- ▶ 基於 C 語言的軟體開發工具
- ▶ 提供 Modbus 函式庫
- ▶ 內建硬體診斷功能
- ▶ 可透過 RS-232 或 Ethernet 下載檔案

硬體特色

- ▶ 80186 處理器 (16 位元 · 80 MHz)
- ▶ IP65 面板保護等級
- ▶ STN LCD (含中文字型)
- ▶ 採用橡膠按鍵 (24 鍵)
- ▶ 一個 10/100M Ethernet 埠
- ▶ 64 位元硬體唯一序號 · 以保護您的程式
- ▶ 操作溫度： -15 ~ +55 °C

1.3. 規格

		VP-2111	VH-2110
系統軟體			
作業系統	MiniOS7 (類似 DOS 的嵌入式作業系統)		
程式下載介面	RS-232 (COM1) 或 Ethernet		
編程語言	C 語言		
編譯器建立執行檔	TC++ 1.01 / TC2.01 (免費軟體) · BC++ 3.1 ~ 5.2x / MSC 6.0 / MSC++ (1.5.2 之前版本)		
CPU 模組			
CPU	80186 或相容 (16-bit · 80 MHz)		
SRAM	768 KB	512 KB	
雙電池備援 SRAM	512 KB (資料可保存 5 年)	-	
Flash	512 KB (可重覆讀寫 100,000 次)		
Flash Disk	64 MB NAND Flash (可重覆讀寫 100,000 次)	-	
EEPROM	16 KB (資料可保持 40 年, 可重覆讀寫 1,000,000 次)		
NVRAM	31 bytes (電池保持, 斷電資料可保存 5 年)		
即時時鐘 (RTC)	可 讀/寫 年、月、日、時、分、秒, 並提供星期資訊		
64 位元硬體序號	有		
看門狗機制	有 (0.8 秒)		
通訊介面			
Ethernet	RJ-45 x 1 · 10/100 Base-TX (Auto-negotiating · Auto MDI/MDI-X · LED 指示燈)		
COM0	與插槽內的 I-87K 系列高卡, 內部通訊用		
COM1	RS-232 (用來更新韌體) (RXD · TXD 與 GND); 非隔離型		
COM2	RS-485	D2+ · D2-; 內建 self-tuner ASIC 晶片	
	隔離電壓	2500 Vdc	
COM3	RS-232/RS-485 ; 非隔離型 (<u>RS-232</u> : RxD · TxD · CTS · RTS 與 GND); (<u>RS-485</u> : Data+ , Data-);		RS-232 ; 非隔離型 (<u>RS-232</u> : RxD · TxD · CTS · RTS 與 GND);
人機介面 (MMI)			
LCD 螢幕	STN · 128 x 64 點陣式 LCD		
顯示模式	文字 + 圖形		

字型	英文 + 簡中/繁中	
橡膠按鍵	24 鍵	
蜂鳴器(Buzzer)	有	
LED 指示燈	3 個雙色 LED (PWR · RUN · LAN1 · L1 · L2 · L3 ; L1 ~ L3 供使用者自訂)	2 個雙色 LED (RUN · LAN1 · L1 · L2 ; L1 ~ L2 供使用者自訂)
I/O 擴充槽		
插槽數	3 個 (僅供 I-8K 與 I-87K 高卡)	-
熱插拔 (Hot Swap) (* 即將發行)	僅 I-87K 高卡	-
資料匯流排	8/16 位元	-
位址匯流排區間	2 K (每個插槽)	-
機構		
尺寸 (W x H x D)	182 mm x 158 mm x 125 mm	
安裝方式	面版安裝 (Panel mounting)	
防護等級	前方面板: 符合 IP 65 標準	
環境特性		
操作溫度	-15 ~ +55 °C	
儲存溫度	-30 ~ +80 °C	
相對溼度	10 ~ 90 % RH (非冷凝)	
電源		
輸入電源	+10 ~ +30 VDC	+12 ~ +48 VDC
靜電隔離	1 kV	-
電源推動功率	3 A · 5 V 供給 I/O 擴充插槽	-
消耗功率	6 W (0.25 A @ 24 V)	3.6 W (0.15 A @ 24 V)

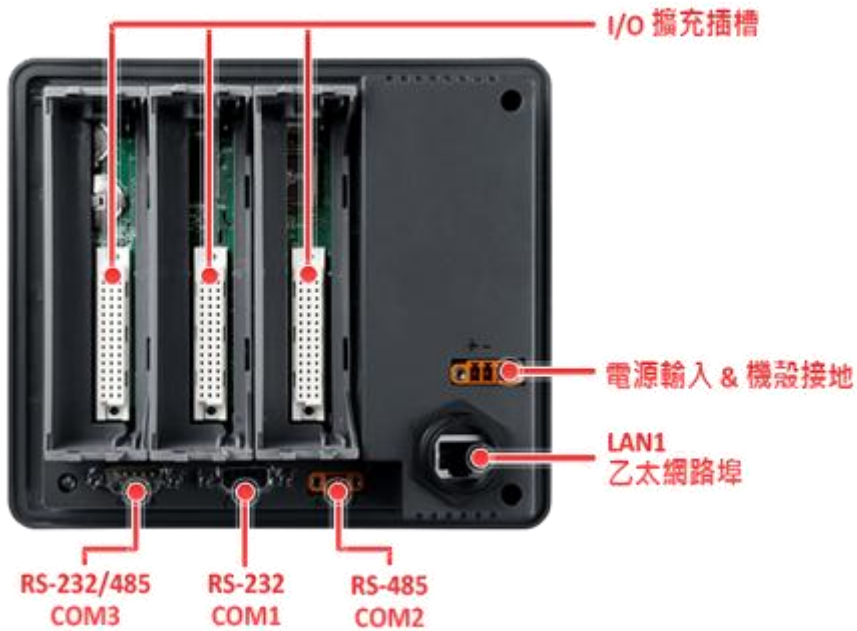
1.4. 概述

下列為各控制器之元件的簡要概述。

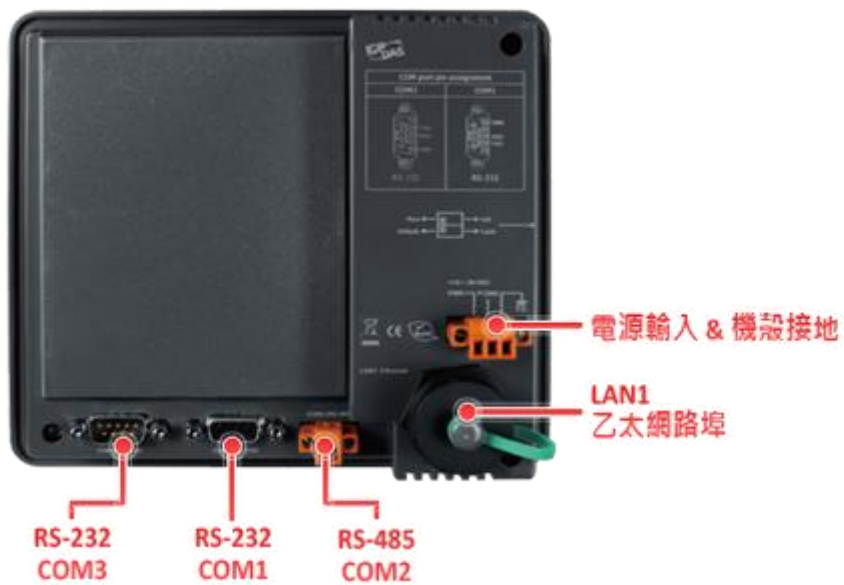
► VP-2000/VH-2000 系列



► VP-2111



► VH-2110

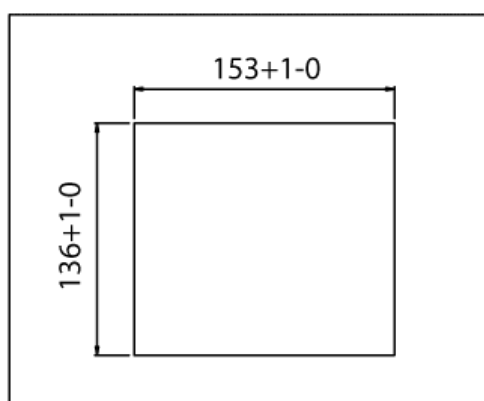


1.5. 尺寸

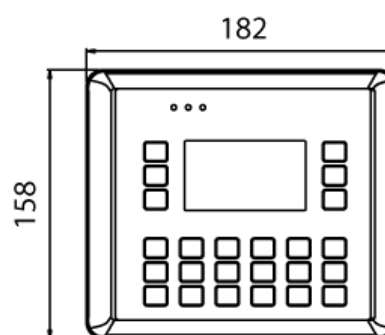
單位：“公厘 (mm)”。

► VP-2000/VH-2000 系列

建議面板開孔



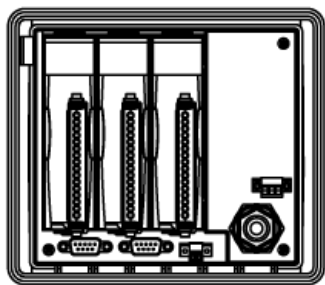
正面視圖



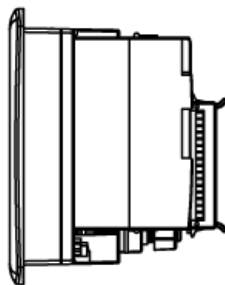
► VP-2000

單位：“公厘 (mm)”。

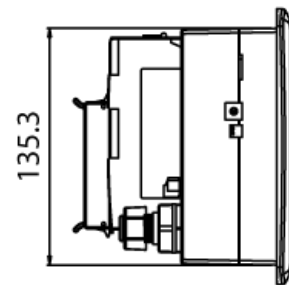
背面視圖



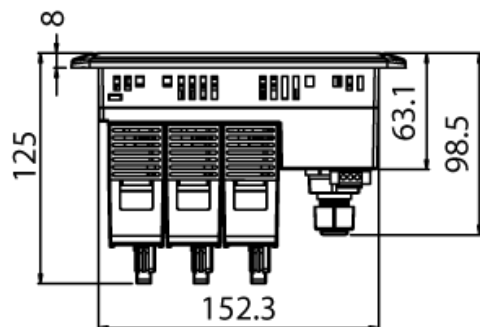
左側視圖



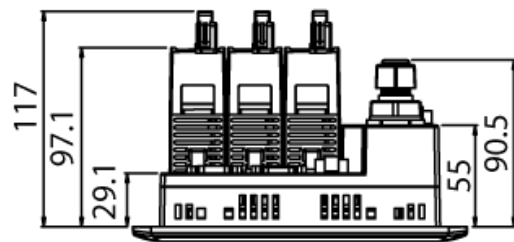
右側視圖



頂端視圖



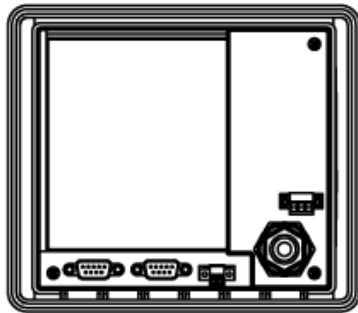
底部視圖



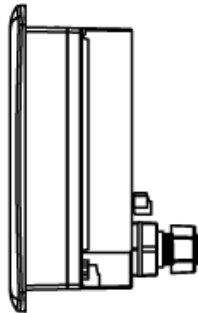
► VH-2000

單位：“公厘 (mm)”。

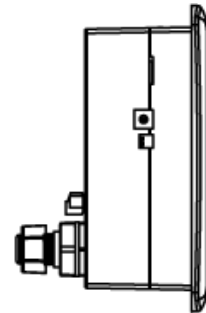
背面視圖



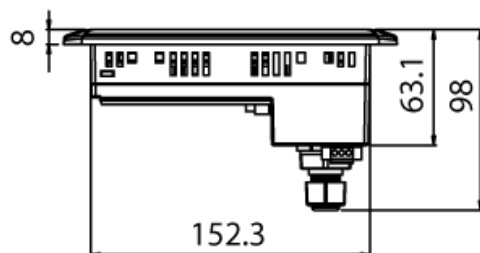
左側視圖



右側視圖



頂端視圖

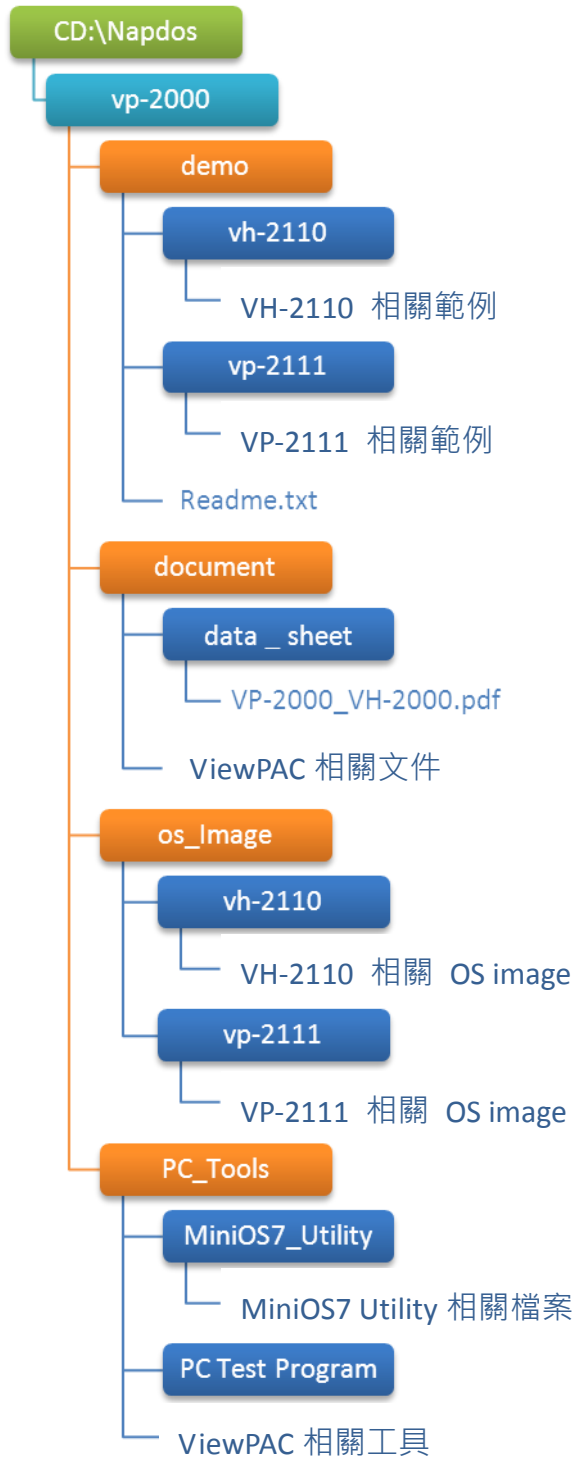


底部視圖



1.6. 隨貨光碟

隨貨光碟中提供了驅動程式、軟體設定工具、所有相關文件...等，如下圖所示。



2. 快速上手

若您是初次使用本產品，請由此章節著手。本章節提供了基本安裝與設定的導覽說明。

除了『快速安裝指南』，包裝中含有下列項目，如有任何品項損毀或遺失，請與我們聯繫。



VP-2111/VP-2111-TC
VH-2110/VH-2110-TC



RJ-45 防水套件



軟體工具光碟



CA-0915
RS-232 纜線



面板固定夾 *5



螺絲起子



螺絲 *5

2.1. 硬體安裝

在安裝硬體之前，您必須先對硬體規格有初步的了解，例如：硬碟容量、電源的可用電壓輸入範圍、通訊介面的類型。請參閱章節“1.3. 規格”以取得完整的硬體資訊。

此外，您還需要知道各種擴充功能，以便快速地選擇最佳擴充模組。請參閱下列網址，取得更多關於擴充模組與 I/O 擴充單元之詳細資訊：

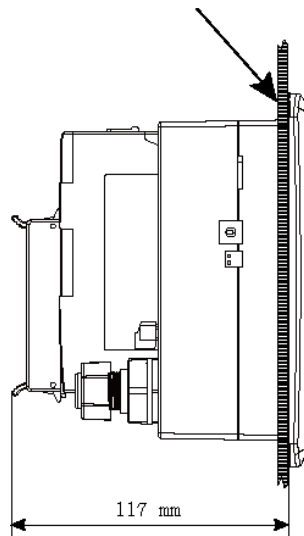
http://www.icpdas.com/products/PAC/viewpac/IO_Expansion.htm

2.1.1. 架設硬體

ViewPAC 可安裝在最大厚度 12mm 的面板中。

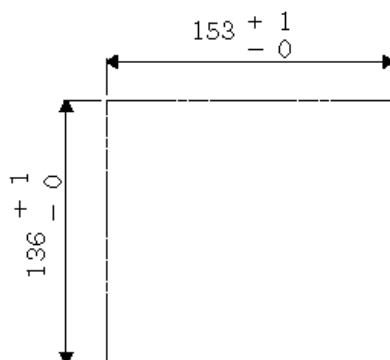
在控制器面板的後面可有充分的空間用來配線或維護設備。

面板厚度可達 12 mm



以下將一步步地引導您裝設 ViewPAC 。

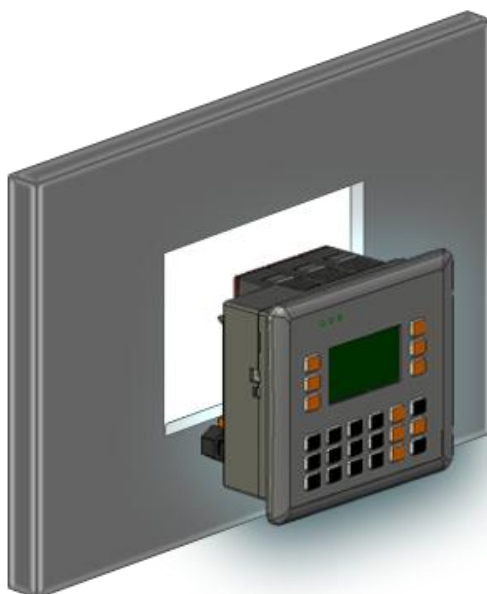
步驟 1: 準備如下圖大小的面板開孔。(單位: mm)



建議之面板開孔

請勿覆蓋到控制器頂端、底部與兩邊的通風孔。

步驟 2: 將 ViewPAC 裝入面板開孔。



步驟 3: 在 ViewPAC 的上方與下方面板安裝面板固定夾。



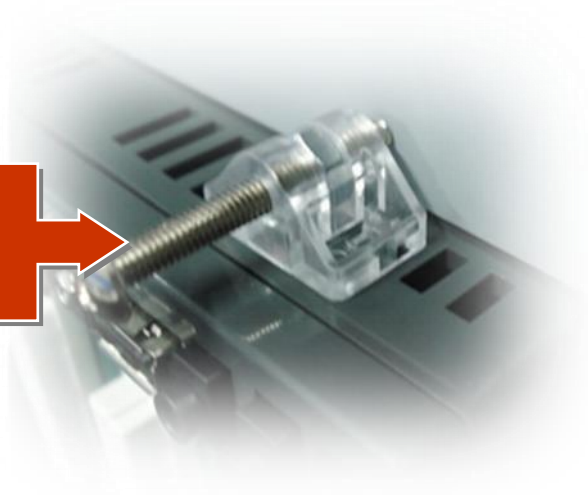
步驟 4: 使用螺絲鎖上面板固定夾。

小技巧與安全警告



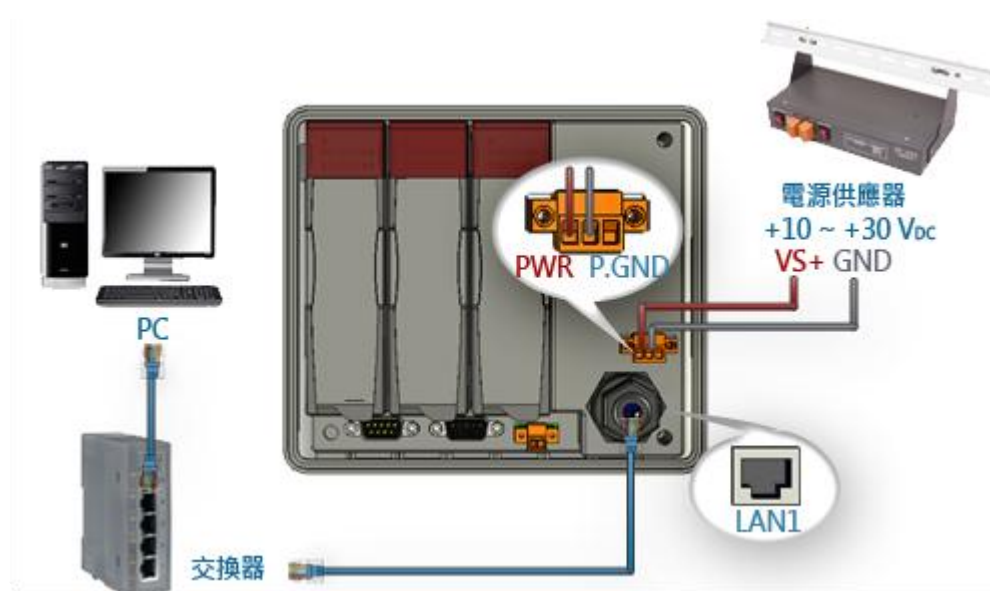
建議螺絲扭矩：3.4 ~ 4.5 kgf-cm。

鎖上螺絲：
M4 x 35L



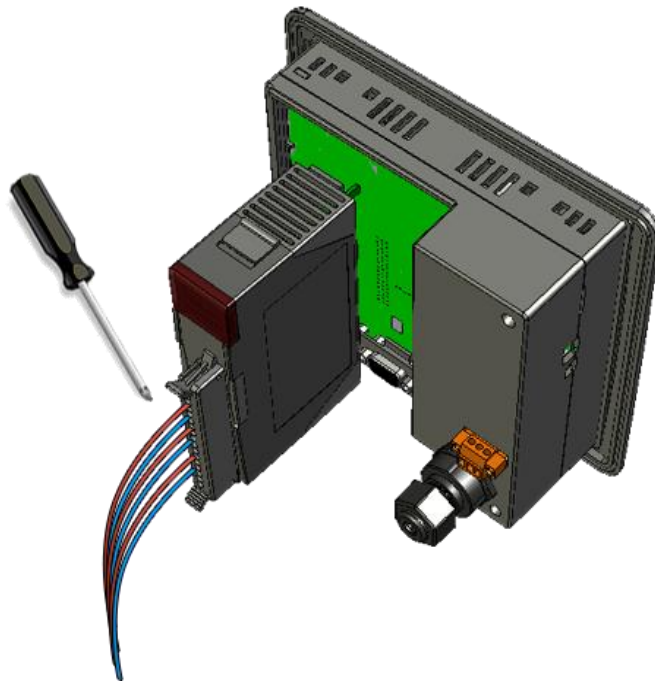
步驟 5: 將 ViewPAC 與 PC 相連接並設置電源供應器。

- i. 將 PC 連接至 ViewPAC 的 LAN Port。
- ii. 將電源供應器 (10 ~ 30 V) 連接至 ViewPAC 的 PWR1 與 GND。



步驟 6: 裝上 I/O 模組 (僅適用於型號 VP-2111 控制器)

在為控制器插槽上的 I/O 模組配線時，建議您先關閉 VP-2111 的電源。



請參閱下列網址，取得更多 ViewPAC 適用之擴充模組的詳細資訊：

http://www.icpdas.com/products/PAC/viewpac/IO_Expansion.htm

小技巧與安全警告



針對 I-8K 與 I-87K 系列擴充模組，ViewPAC 僅支援高卡 (High Profile) 版本。

2.1.2. 安裝 IP-65 防水等級的连接頭

ViewPAC 提供一個 IP-65 防水等級的连接頭，包含了下列裝入 RJ-45 连接線上的元件。

以下將一步步地引導您安裝 IP-65 防水等級的连接頭。



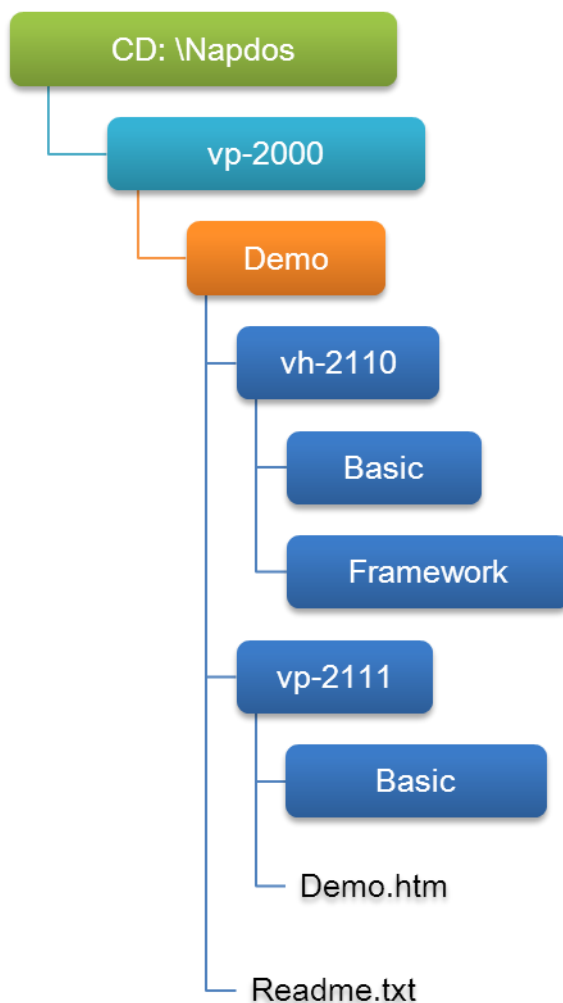
2.2. 軟體安裝

隨貨光碟中包括了一套完整的 API、範例程式與開發個人應用的發展工具。

以下將一步步地引導您安裝 ViewPAC 的 API、範例程式與軟體工具。

步驟 1: 由隨貨光碟中，複製 “Demo” 資料夾到您的 PC。

“Demo” 資料夾裡含有客戶開發個人應用所需的重要資源，包括：函式庫，標頭檔，範例程式與更多資訊，如下所示。



步驟 2: 安裝 MiniOS7 Utility。



MiniOS7_Utility_V321.exe
[MiniOS7 Utility Ver 3.21] Setup

MiniOS7 Utility 是一套工具軟體，用於管理 MiniOS7 設備 (例如 VP/VH-2000， μ PAC-5000，iPAC-8000， μ PAC-7186...等)。它區分為四個部份 – 系統偵測、通訊管理、檔案管理與 OS 載入程式。

MiniOS7 Utility 可由隨貨光碟或是 FTP 網站中取得：

CD:\Napdos\minios7\utility\minios7_utility\

ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/minios7_utility/

2.3. 設定啟動模式

上傳程式至 ViewPAC 之前，必須先進入初始 (Init) 模式並關閉防寫功能。

請確認 Unlock 的位置在 “ON”，且 Init 的位置在 “ON”。



2.4. 上傳 ViewPAC 程式

MiniOS7 Utility 是一套工具軟體，用於管理 MiniOS7 設備 (例如: VP/VH-2000， μ PAC-5000，iPAC-8000， μ PAC-7186...等)。它區分為四個部份 – 系統偵測、通訊管理、檔案管理與 OS 載入程式。

使用 MiniOS7 Utility 上傳程式之前，請確認 ViewPAC 已連接至 PC。

下列為主要的上傳步驟：

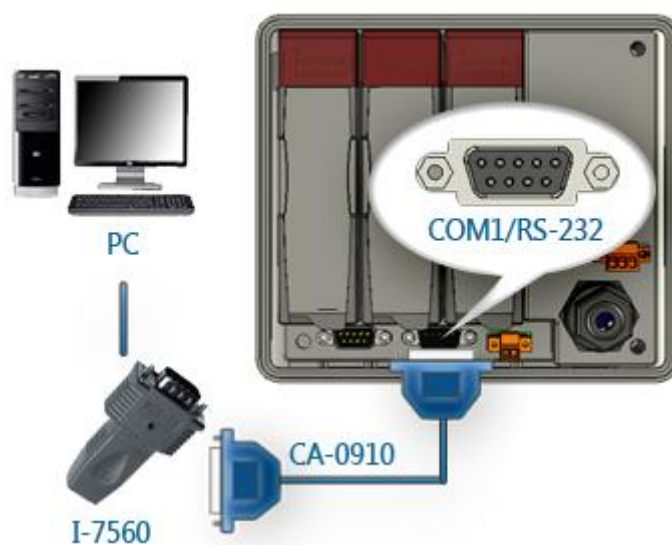
1. 建立 PC 與 ViewPAC 之間的連線。
2. 上傳程式至 ViewPAC 並執行程式。
3. 設定啟動後，自動執行程式。

後續將說明這些步驟的詳細內容。

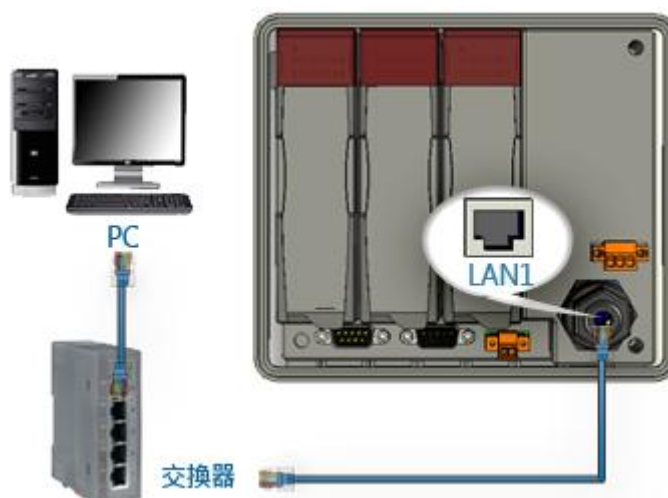
2.4.1. 建立 PC 與 ViewPAC 之間的連線

以下有兩種方式可建立 PC 與 ViewPAC 之間的連線。

► RS-232 連接方式



► Ethernet 連接方式



後續將說明此兩種連接類型。

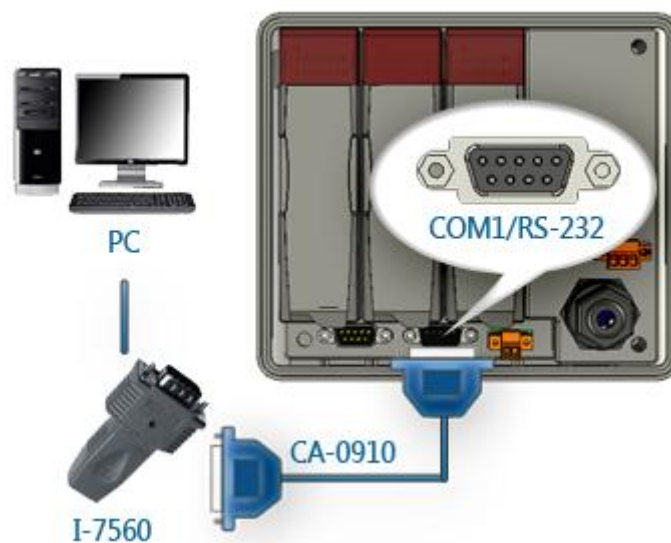
2.4.1.1. 採用 RS-232 連接方式

以下將一步步地引導您，如何使用 RS-232 連接至 PC。

步驟 1: 將開關 UnLock 切換至 “ON”，並將 Init 切換至 “ON”。

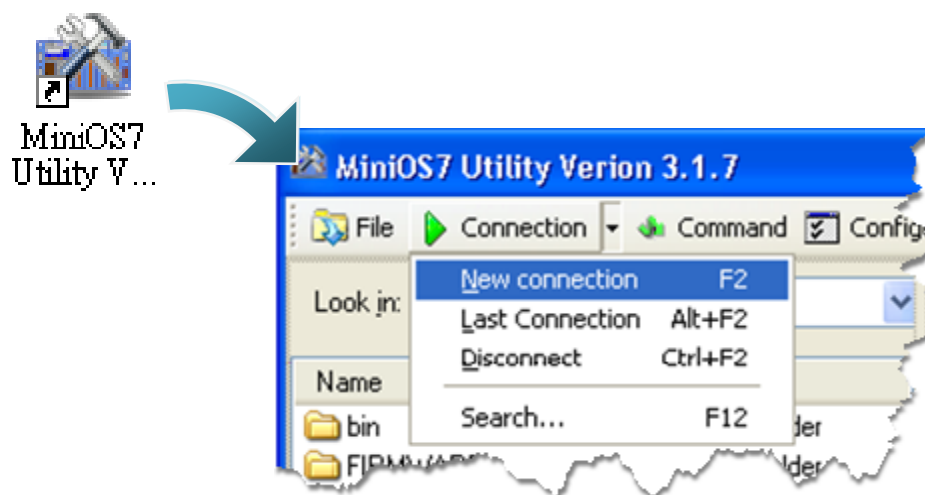


步驟 2: 將 RS-232 纜線 (CA-0910) 連接至 PC。

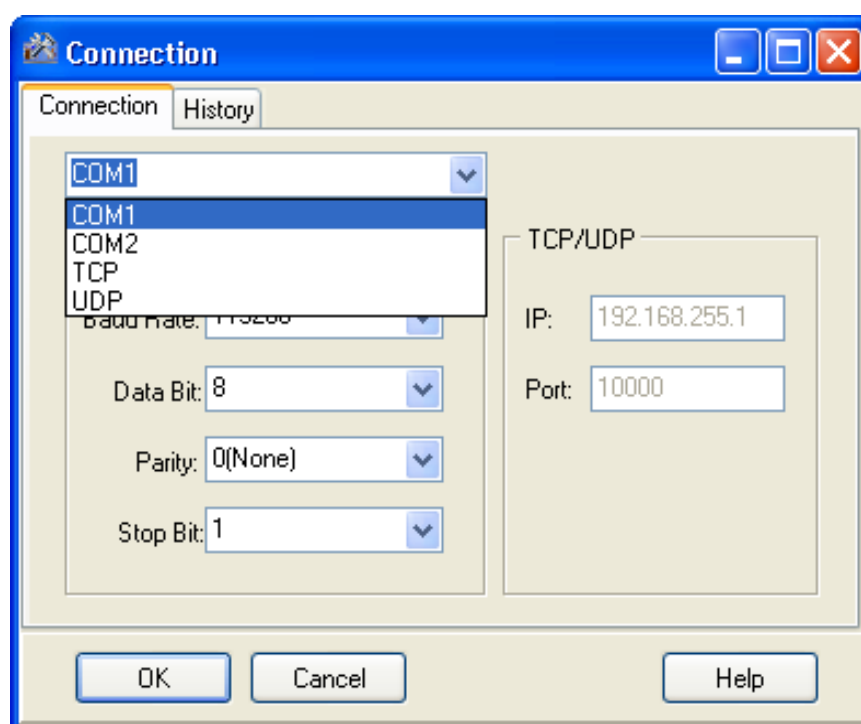


步驟 3: 執行 MiniOS7 Utility 。

步驟 4: 點選功能表 “Connection” > “New connection” 功能 。



步驟 5: (“Connection” 視窗 > “Connection” 頁籤 > 下拉選單)
選取 “COM1” 並點選 “OK” 。



步驟 6: 已建立連線。



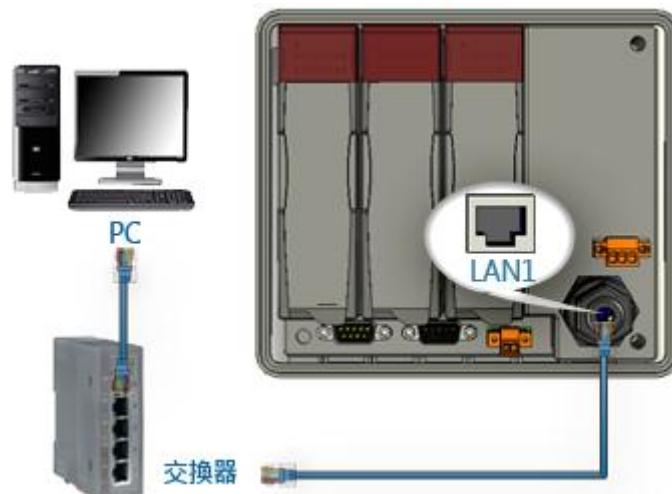
2.4.1.2. 採用 Ethernet 連接方式

下列將一步步地引導您，如何使用 Ethernet 方式與 PC 相連接。

步驟 1: 將開關 Unlock 切換至 “ON”，並將 Init 切換至 “ON”。



步驟 2: 將乙太網路線連接至 PC。



步驟 3: 執行 MiniOS7 Utility 。

步驟 4: 點選功能表 “Connection” > “Search” 功能 。

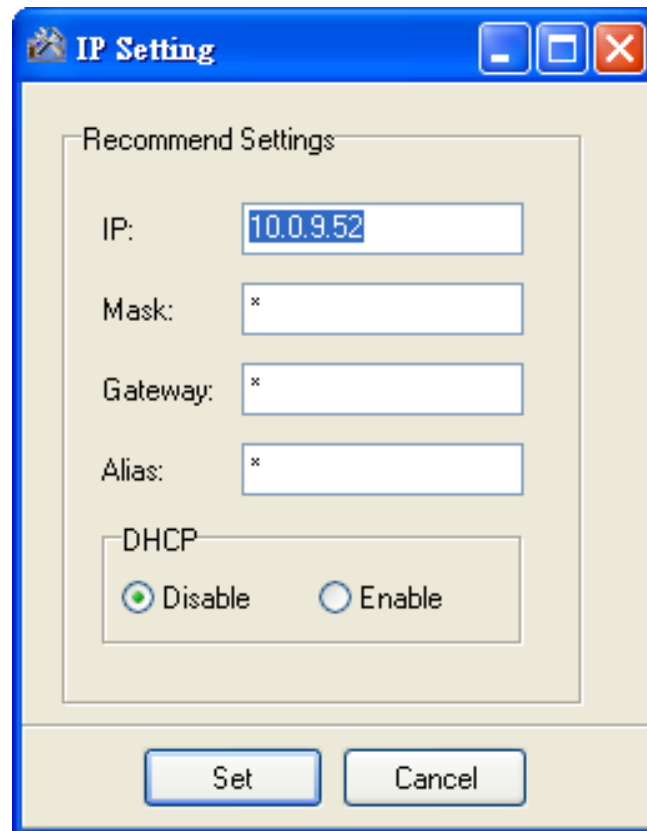


步驟 5: (“MiniOS7 Scan” 視窗)

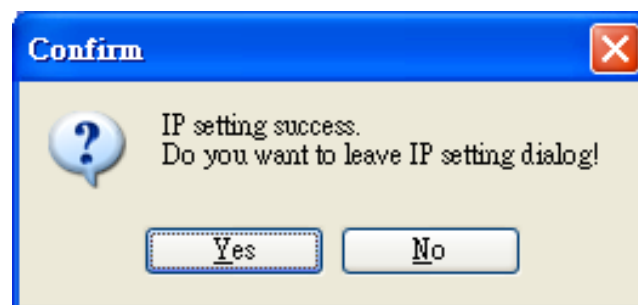
選取 模組名稱 並點選工具按鈕 “IP setting” 。



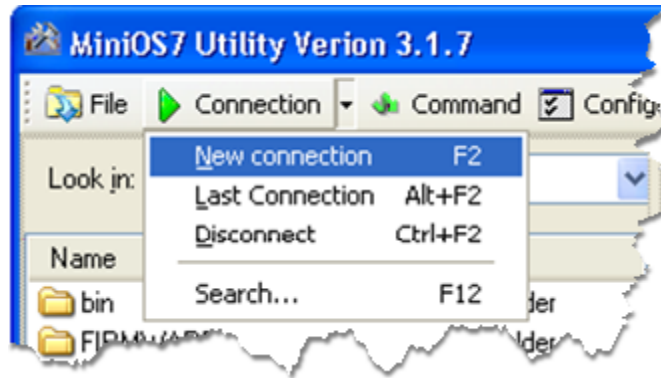
步驟 6: (“IP Setting” 視窗)
設定 “IP” 位址並點選 “Set” 。



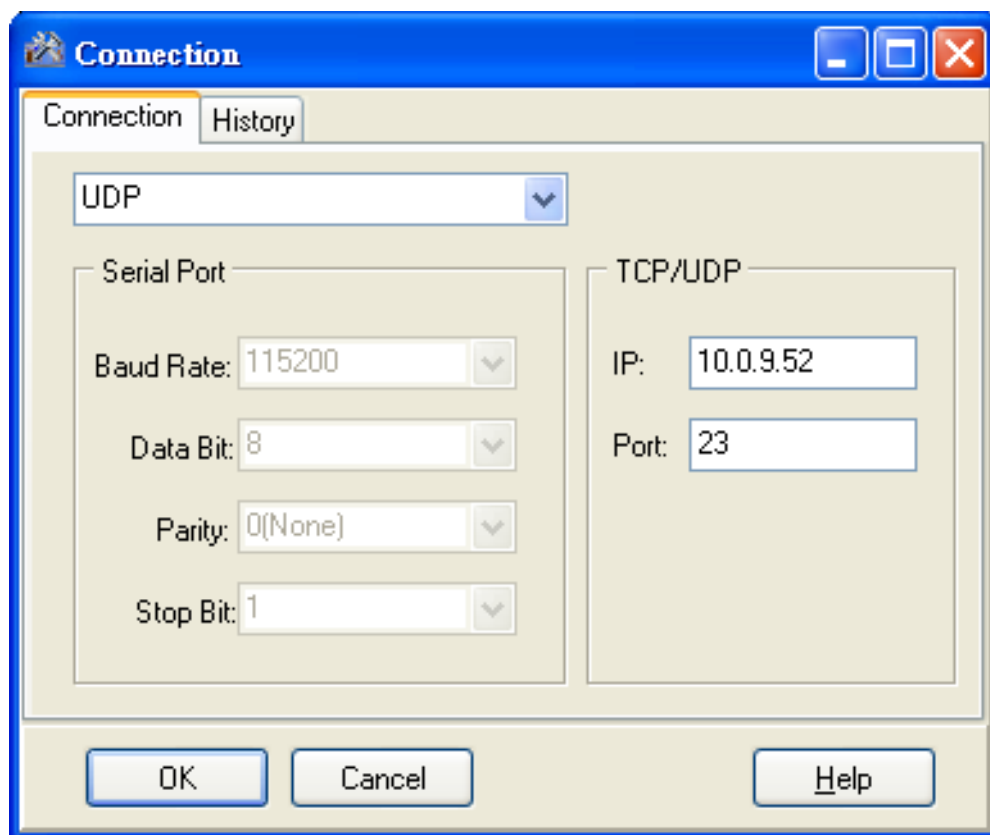
步驟 7: (“Confirm” 視窗) 點選 “Yes” 。



步驟 8: 點選功能表 “Connection” > “New connection” 。



步驟 9: (“Connection” 視窗 > “Connection” 頁籤 > 下拉選單)
選取 “UDP” 並指定 IP 位址，再點選 “OK” 。



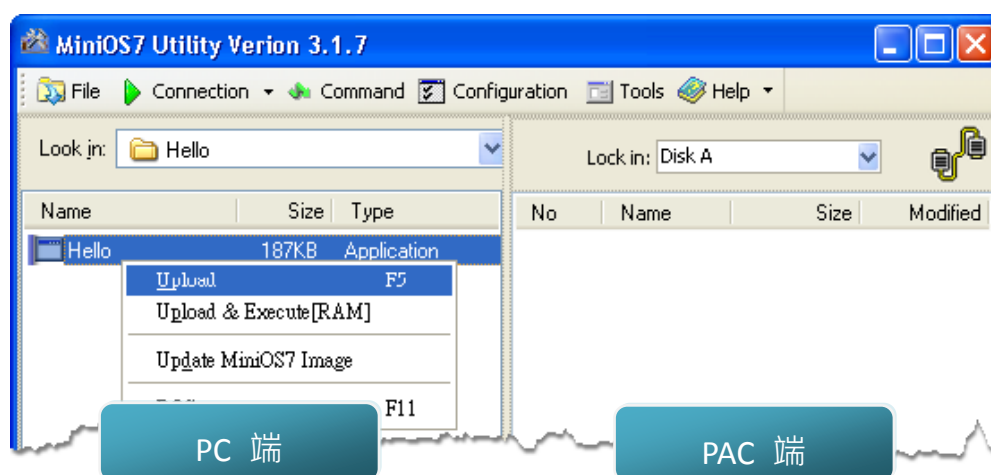
步驟 10: 已建立連線。



2.4.2. 上傳並執行 ViewPAC 程式

上傳並執行 ViewPAC 程式之前，您必須先建立 PC 與 ViewPAC 之間的連線。請參閱章節“2.4.1. 建立連線”以取得詳細內容。

步驟 1: PC 端，於欲上傳的檔案上按滑鼠右鍵並點選“Upload”。



步驟 2: PAC 端，於欲執行的檔案上按滑鼠右鍵並點選“Run”。



2.4.3. 設定自動執行程式

上傳程式至 ViewPAC 後，若您希望於 ViewPAC 啟動時能自動地執行程式，有個很簡單的方式，您可建立一個名稱為 `autoexec.bat` 的批次檔並將其上傳至 ViewPAC，於下次啟動時程式將會自動執行。

例如，設定在啟動時即執行“hello”程式。

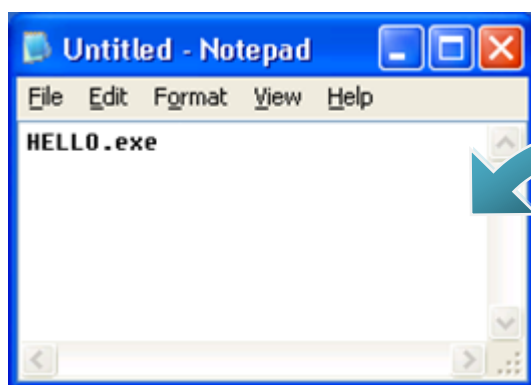
步驟 1: 建立一個 `autoexec.bat` 檔案。

i. 開啟“Notepad”

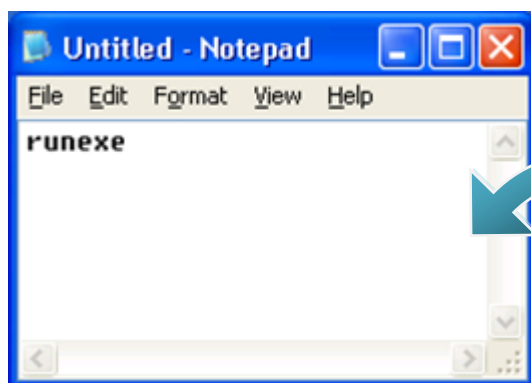
ii. 輸入命令

此命令可以是檔案名稱“`hello.exe`”(執行特定的檔案)或是“`runexe`”(執行上次的執行檔)

iii. 儲存檔案為 `autoexec.bat`。



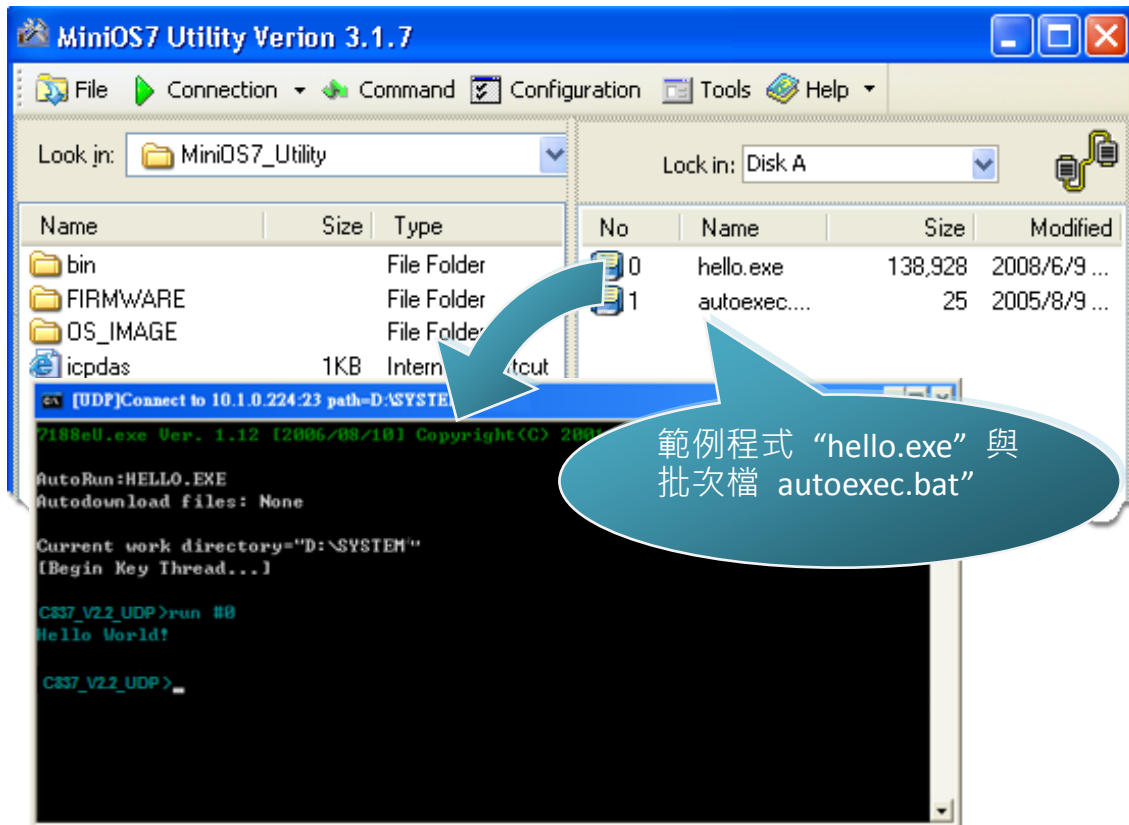
檔案名稱:
執行特定檔案。



Runexe:
執行上次的執行檔。

步驟 2: 使用 MiniOS7 Utility 將程式上傳至 ViewPAC。

請參閱章節 “2.4.1. 建立連線” 以取得詳細資訊。



小技巧 與 安全警告



在重新啟動讓設定生效前，您必須將 Init 開關切換至 “OFF”。



2.5. 更新 ViewPAC 的 OS image

泓格科技 (ICP DAS) 將持續地新增功能於 ViewPAC 中，建議您定期地參閱我們的網站以取得 ViewPAC 的最新資訊。

步驟 1: 取得 ViewPAC 最新版本的 OS image。



可至下列位置，取得 ViewPAC 最新版本的 OS image：

VP-2111:

CD:\NAPDOS\vp-2000\os_image\vp-2111\

http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/os_image/vp-2111/

VH-2110:

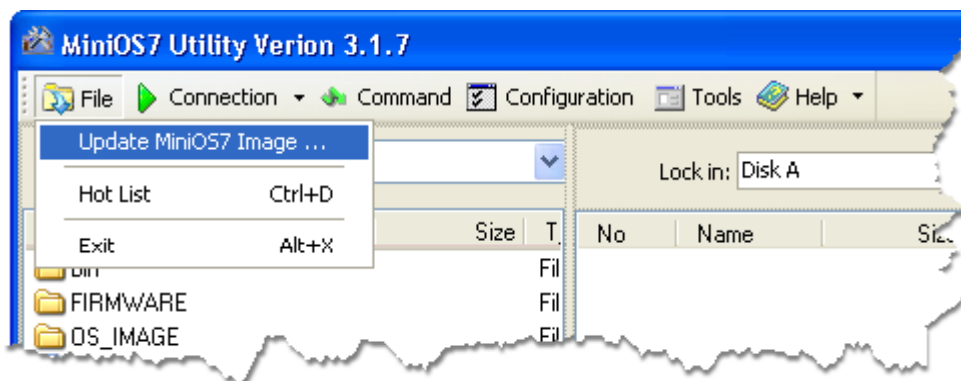
CD:\NAPDOS\vp-2000\os_image\vh-2110\

http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/os_image/vh-2110/

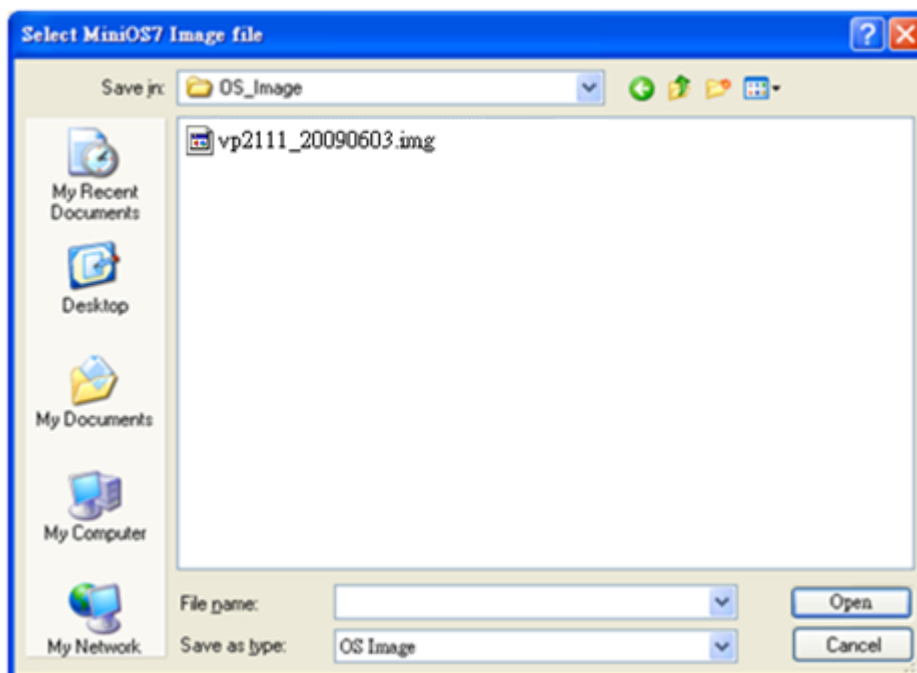
步驟 2: 建立連線。

請參閱章節 “2.4.1. 建立連線” 以取得詳細內容。

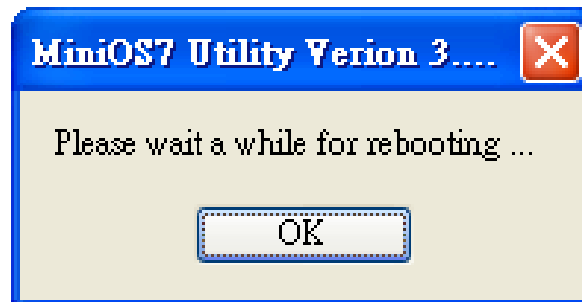
步驟 3: 點選功能表 “File” > “Update MiniOS7 Image ...”。



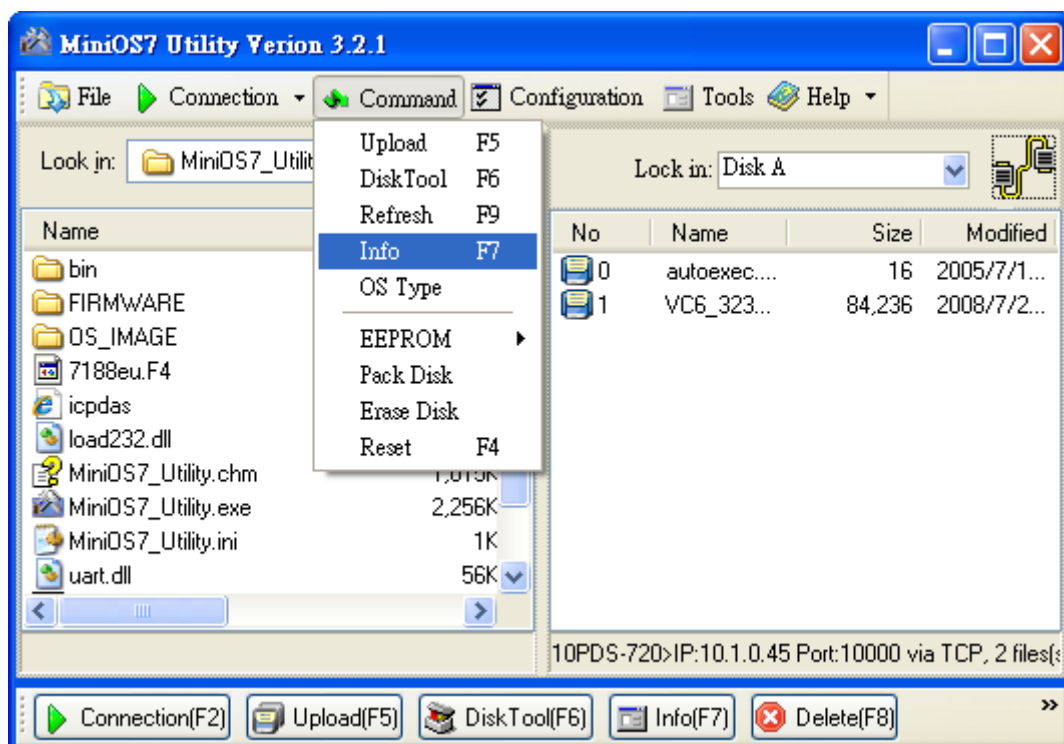
步驟 4: 選取最新版本之 MiniOS7 OS image。



步驟 5: 點選 “OK”。



步驟 6: 點選功能表 “Command” > “Info” 確認 OS image 版本。



3. 第一個範例程式- “Hello World”

您可發現在學習每一種電腦編程語言時，首要的範例程式即為 "Hello World"，它簡略地介紹了程式語言的語法與輸出方式。

以下將一步步地引導您，如何編寫第一個 ViewPAC 程式 - “Hello World”。

3.1. C 編譯器安裝

C 語言以它的效率聞名，並且是多數人編寫應用時，所使用的編程語言。

在編寫第一個 μ PAC-5000 程式前，請確認您的系統中已安裝了必要的 C/C++ 編譯器與相關的函式庫。

下列為應用程式開發服務中常用的 C 編譯器：

- Turbo C++ 版本 1.01
- Turbo C 版本 2.01
- Borland C++ 版本 3.1 - 5.2.x
- MSC
- MSVC ++

建議您使用 Borland C++ 編譯器，如同隨貨光碟中已建立之函式庫的編譯器。

小技巧 與 安全警告



在編譯應用程式之前，請先注意以下事項：

- 建立標準 DOS 可執行程式。
- 設定 CPU 選項為 80188/80186。
- 如需使用浮點數計算，設定浮點數選項為 EMULATION。
(請勿選取 8087)
- 取消除錯資訊功能，以減少程式大小。(MiniOS7 支援此功能)

此處我們將採用 Turbo C++ 1.01 編寫您的第一個範例程式。

3.1.1. 安裝編譯器

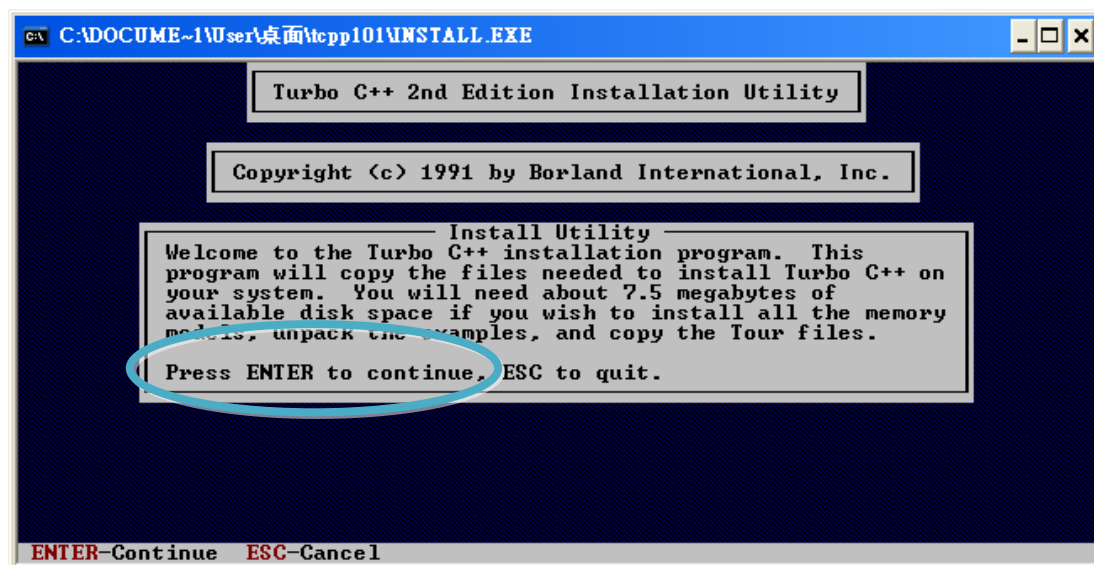
若您的系統尚未安裝任何的編譯器，首要步驟即為安裝編譯器。

以下將一步步地引導您，於系統中安裝 Turbo C++ 版本 1.01。

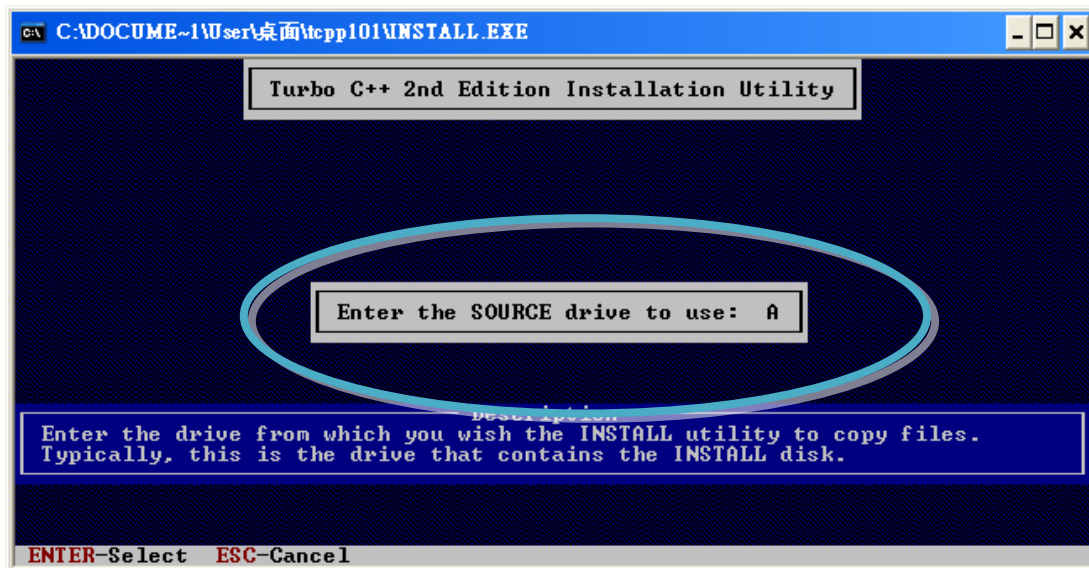
步驟 1: 滑鼠雙擊 Turbo C++ 執行檔來啟用安裝精靈。



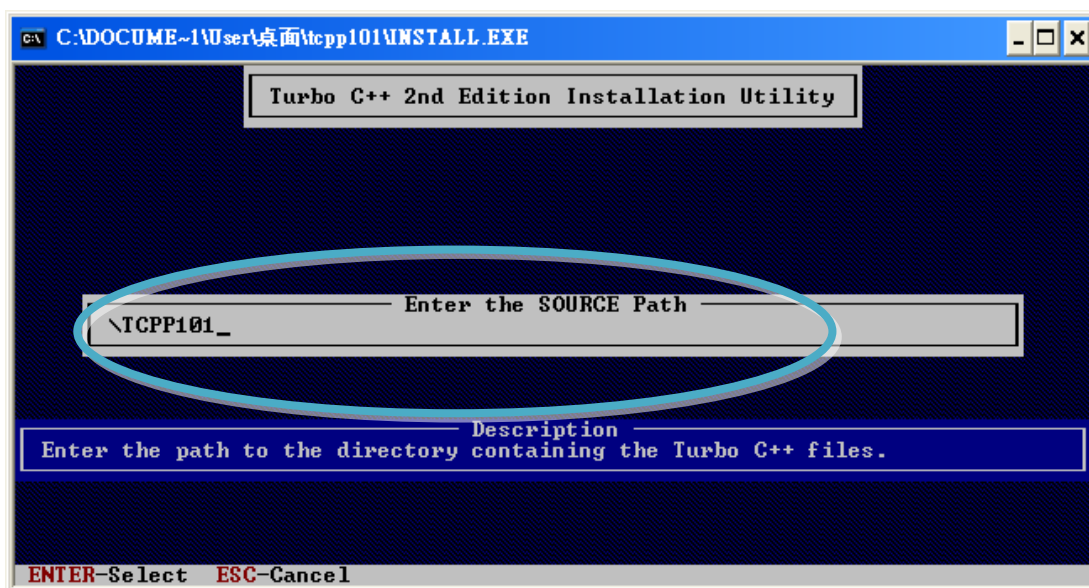
步驟 2: 按 “Enter” 鍵繼續。



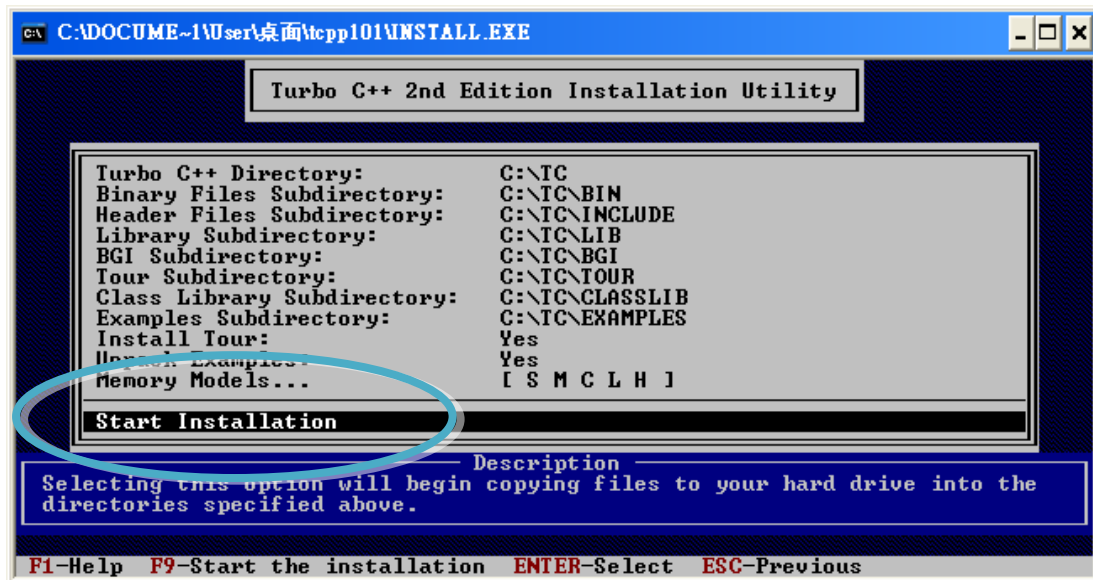
步驟 3: 輸入您欲安裝此軟體的硬碟名稱。(EX. C)



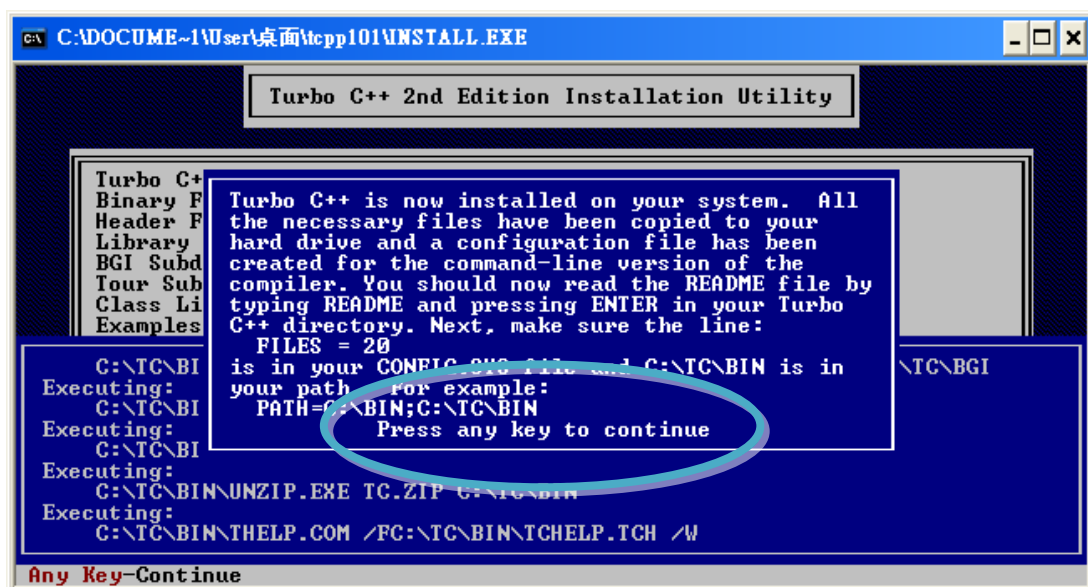
步驟 4: 輸入您欲安裝相關檔案的目錄路徑。(EX. \TC)



步驟 5: 選取 “Start Installation” 開始安裝程序。



步驟 6: 按任意鍵繼續。



步驟 7: 按任意鍵繼續。

```
C:\DOCUME~1\User\Desktop\tcpp101\INSTALL.EXE
Turbo C++ 2nd Edition Installation Utility

Turbo C++ Directory:      C:\TC
Binary Files Subdirectory: C:\TC\BIN
Header Files Subdirectory: C:\TC\INCLUDE
Library Subdirectory:     C:\TC\LIB
BGI Subdirectory:         C:\TC\BGI
Tour Subdirectory:        C:\TC\TOUR
Class Library Subdirectory: C:\TC\CLIB
Examples Subdirectory:     C:\TC\EXAMPLES

For a tutorial on the Turbo C++ integrated
environment, change to the Turbo C++ Tour
directory and run the Turbo C++ Tour.
For example:
Change to: C:\TC\TOUR
And type: TC\TOUR

Press any key to continue

C:\TC\BIN
Executing:
C:\TC\BIN
Executing:
C:\TC\BIN\THELP.COM /M /FC:\TC\BIN\TCHELP.TCH
Executing:
C:\TC\BIN\UNZIP.EXE TC.ZIP C:\TC\BIN
Executing:
C:\TC\BIN\THELP.COM /FC:\TC\BIN\TCHELP.TCH /W

Any Key-Continue
```

步驟 8: 您已完成安裝。

3.1.2. 設定環境變數

完成編譯器安裝後，在 Windows 命令列中有多種編譯器可用。您可設定環境變數的路徑，以便輸入簡單的名稱即可執行命令列上的編譯器，而不用使用完整的路徑名稱。

步驟 1: 滑鼠右鍵點選桌面圖示“我的電腦”並選取功能表選項“內容”。



步驟 2: (“系統內容” 視窗)

點選 “進階” 頁籤中的 “環境變數” 按鈕。

步驟 3: (“環境變數” 視窗)

點選 “系統變數” 項目中的 “Path” 再點選 “編輯” 按鈕。



步驟 4: 在“變數值”欄位後面加上目標路徑。

分號 (;) 用來表示變數值之間的分隔符號。例如, “;c:\TC\BIN;c:\TC\INCLUDE\”



步驟 5: 重新啟動電腦，讓您的變更生效。

3.2. ViewPAC 之應用程式介面 (API)

以下之 API 可提供使用者自訂一些標準功能並可與其它應用程式、設備和服務相整合。

請參閱下列位置，取得 ViewPAC API 之詳細資訊：

CD:\NAPDOS\vp-2000\Readme.txt

<http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/Readme.txt>

建立應用程式之前，請確認您已完成軟體安裝。若您尚未安裝，請參閱章節“2.2. 軟體安裝”。

3.3. ViewPAC 中的第一個程式

此章節，我們假設您已在 C 硬碟區的根目錄下，安裝了 Turbo C++ 1.01 (如章節 “3.1. C 編譯器安裝”) 與 μ PAC-5000 的 API (如章節 “2.2. 軟體安裝”)。

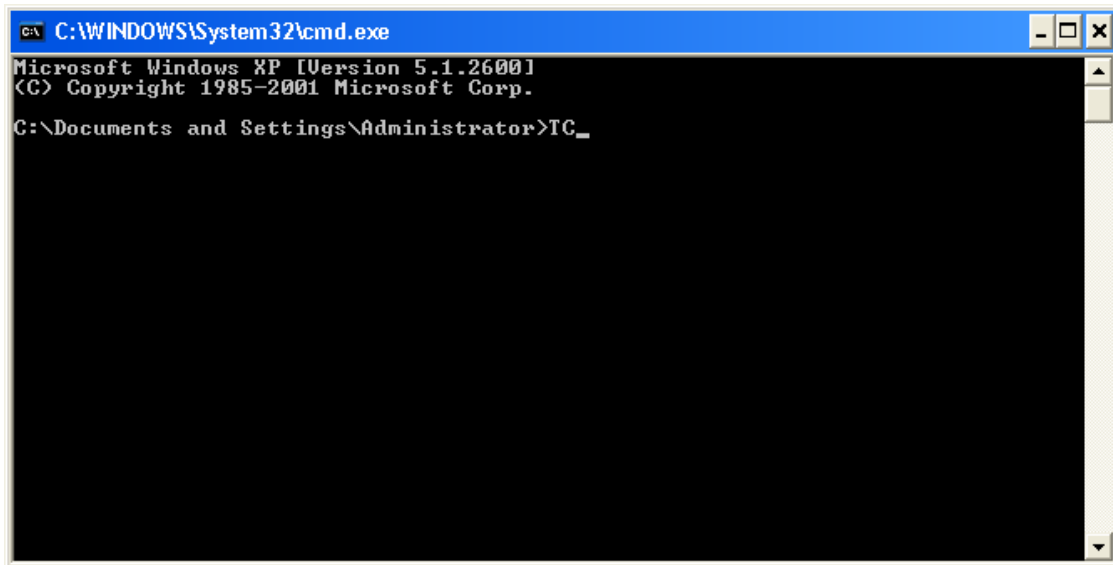
以下將一步步地引導您，編寫第一個程式。

步驟 1: 開啟 MS-DOS 的命令提示字元。

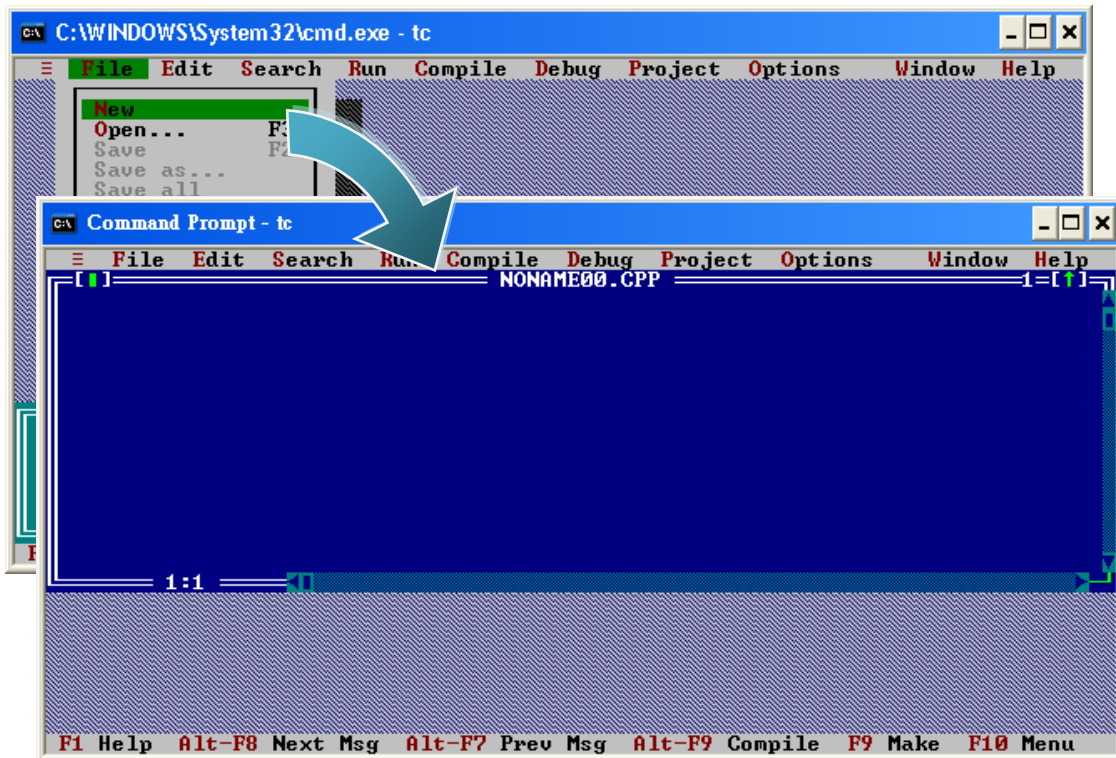
- i. “開始” 選單中，點選 “執行”。
- ii. “執行” 視窗中，輸入 “cmd”。
- iii. 點選 “確定” 按鈕。



步驟 2: (命令提示字元視窗) 輸入“TC” 並按 “Enter” 鍵。



步驟 3: 點選功能表 “File” > “New” , 新增一個原始檔。



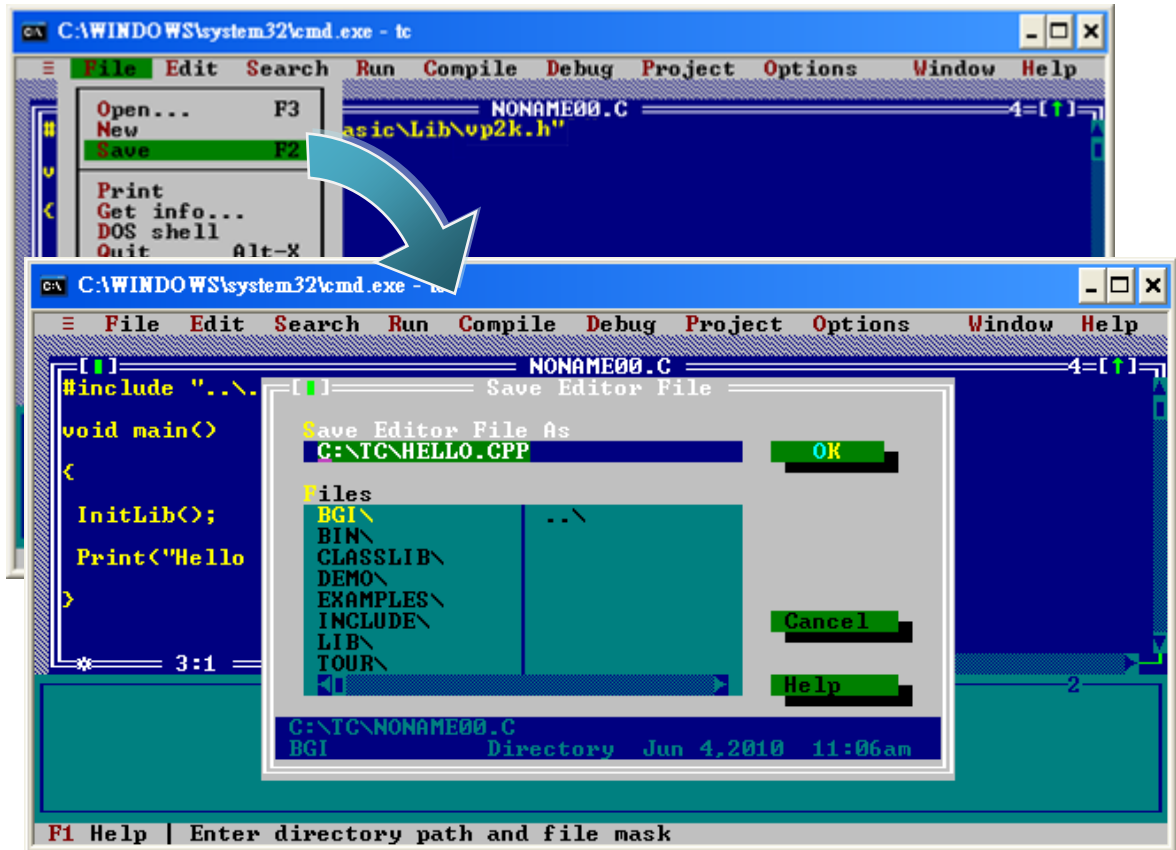
步驟 4: 輸入以下程式碼。(注意: 程式碼有區分大小寫)

```
#include "..\..\Demo\basic\Lib\vp2k.h"
/* 引入標題檔以允許使用 uPAC5000.lib 函式 */

void main(void)
{
    InitLib(); /* 初始化 ViewPAC 函式庫 */
    Print("Hello world!\r\n"); /* 螢幕上印出訊息 */
}
```

步驟 5: 儲存原始檔。

- i. 點選功能表 “File” > “Save”。
- ii. 輸入檔名 “HELLO”。
- iii. 點選 “OK”。



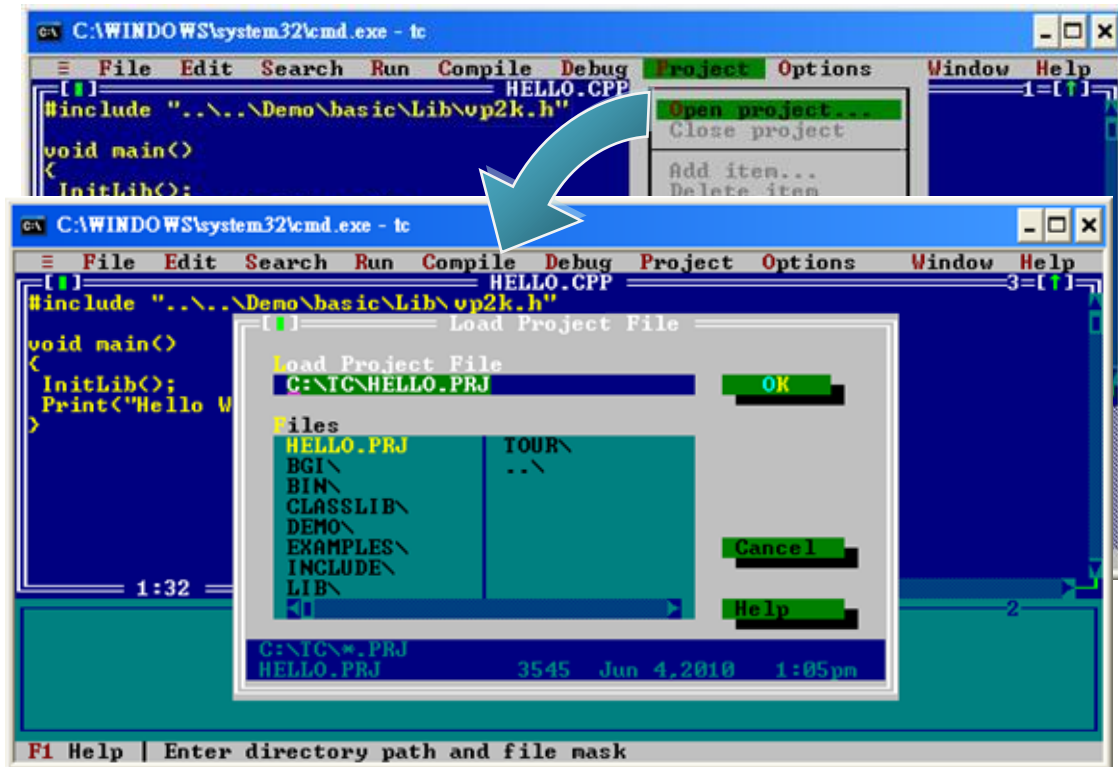
小技巧 與 安全警告



您可使用熟悉的文字編輯器或其他工具來編寫以下程式碼，
但您必須將原始檔儲存為 .c 的副檔名。

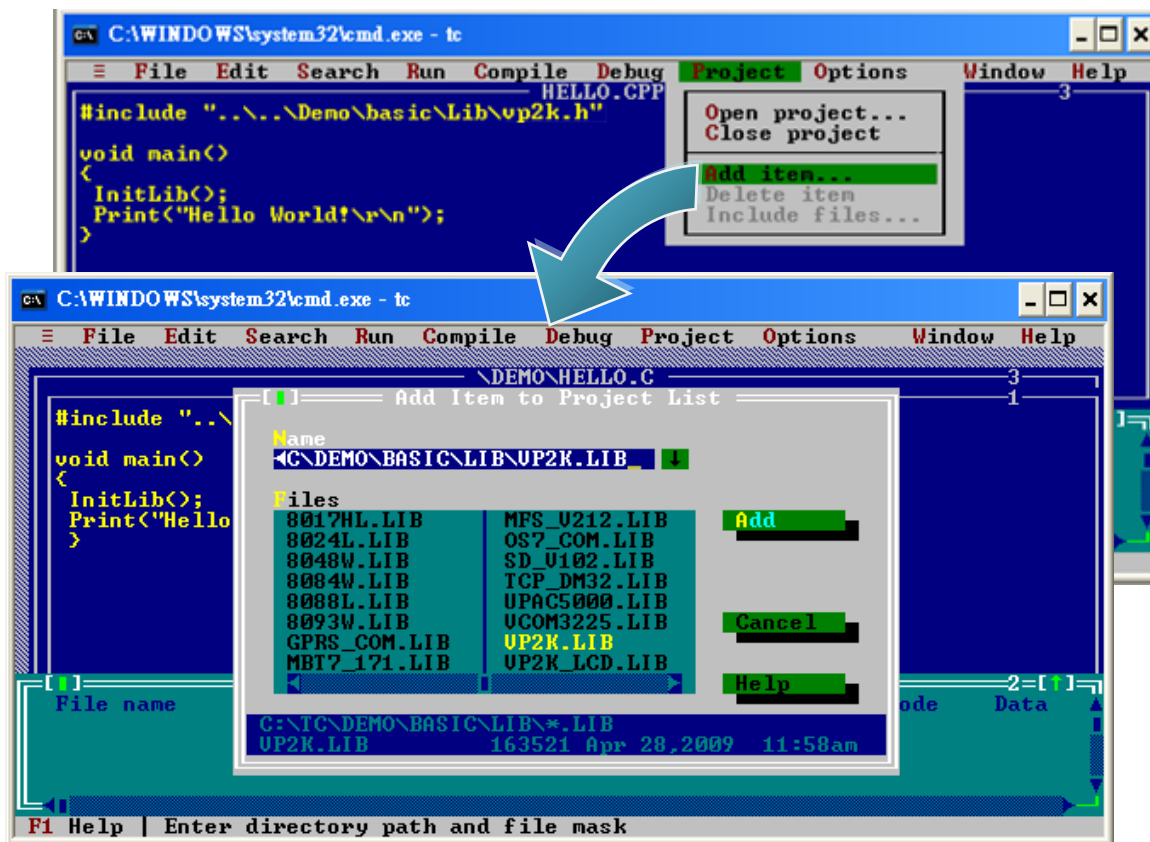
步驟 6: 新建專案 (*.prj)

- i. 點選功能表 “Project” > “Open project...”。
- ii. 輸入專案名稱 “HELLO”。
- iii. 點選 “OK”。



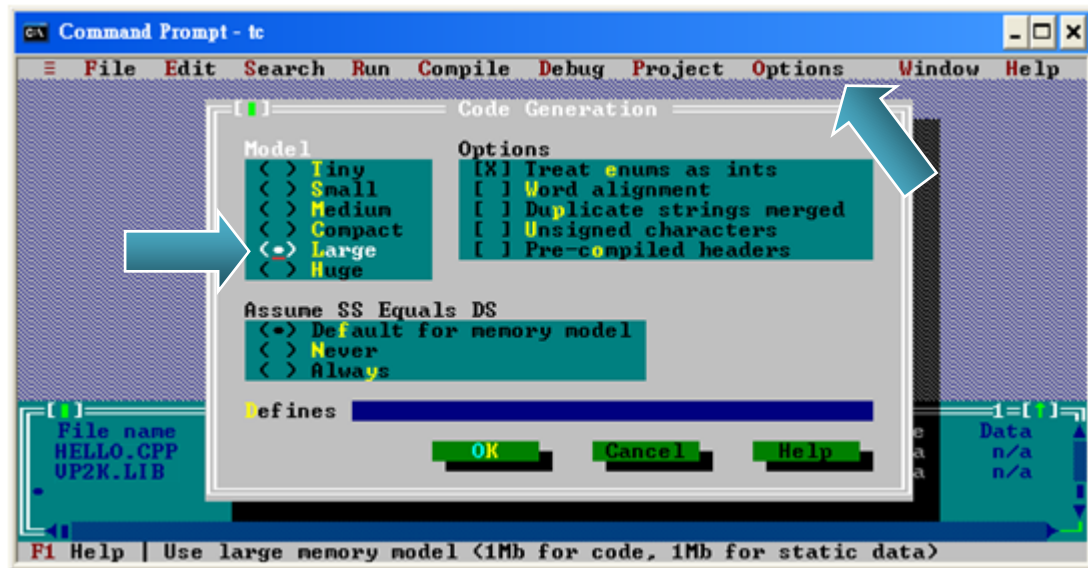
步驟 7: 加入需要的函式庫 (*.lib) 於專案中。

- i. 點選功能表 “Project” > “Add item...”。
- ii. 輸入 “*.LIB” 以顯示所有可用的函式庫。
- iii. 選取您所需要的函式庫。
- iv. 點選 “Add” 加入。
- v. 點選 “Done” 完成設定。



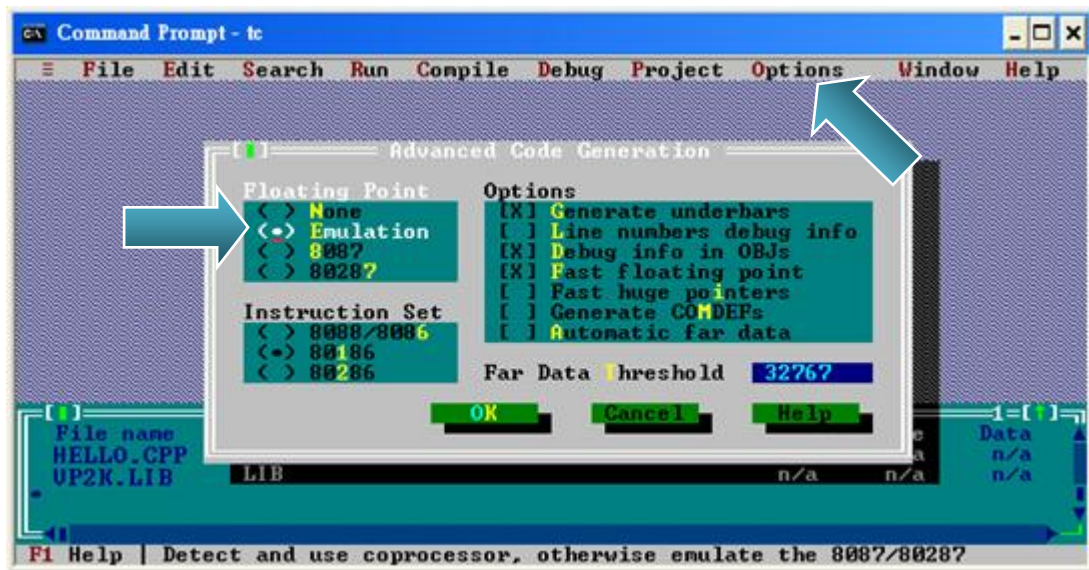
步驟 8: 設定記憶體模式為 “Large”。

- i. 點選功能表 “Options” > “Compiler” > “Code generation...”。
- ii. 於 “Model” 項目，選取 “Large”。
- iii. 點選 “OK”。



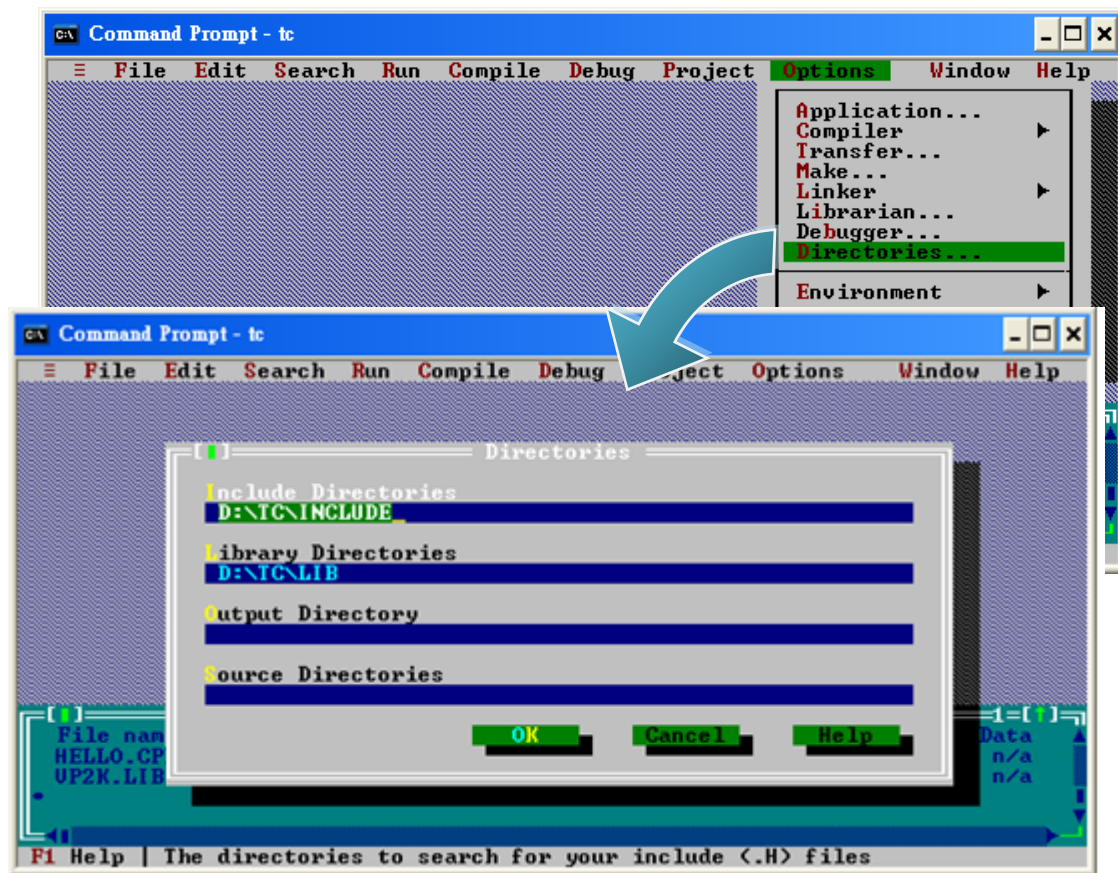
步驟 9: 設定浮點數為 “Emulation”，指令集為 “80186”。

- i. 點選功能表 “Options” > “Compiler” > “Advanced code generation...”。
- ii. 於 “Floating Point” 項目，選取 “Emulation”。
- iii. 於 “Instruction Set” 項目，選取 “80186”。
- iv. 點選 “OK”。

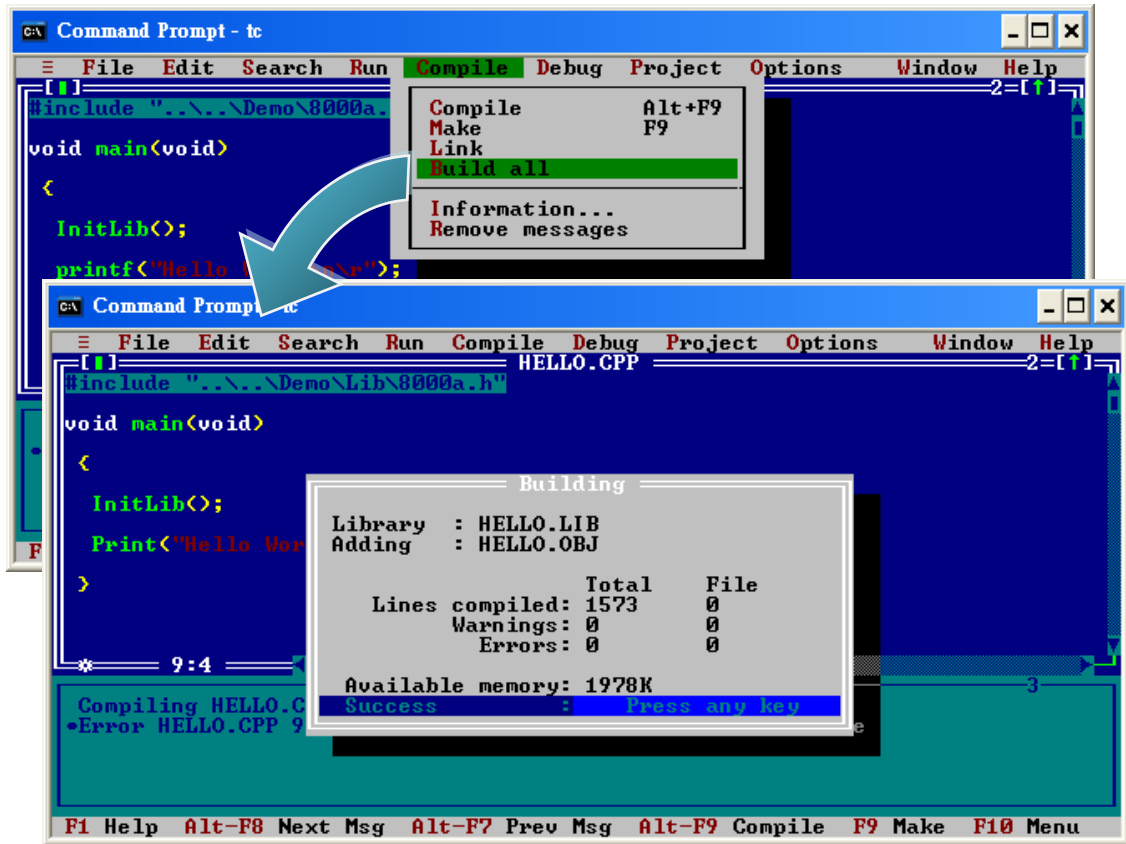


步驟 10: 設定 TC 編譯器的引入檔 (Include) 與函式庫之目錄。

- i. 點選功能表 “Options” > “Directories...”。
- ii. 於 “Include Directories” 項目，設定標頭檔之存放目錄。
- iii. 於 “Library Directories” 項目，設定函式庫之存放目錄。
- iv. 點選 “OK”。



步驟 11: 點選功能表 “Compile” > “Build all” 以建置專案。

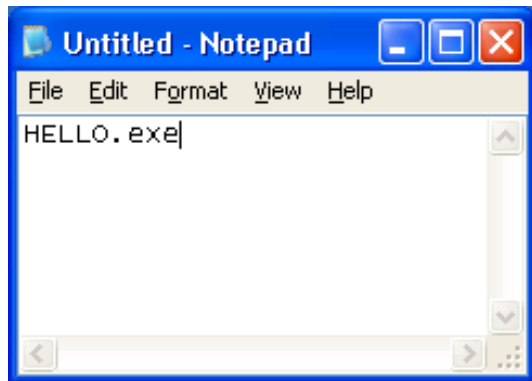


步驟 12: 設定工作模式。

請確認 Unlock 開關已經切換至 “ON”，且 Init 已切換至 “ON”。



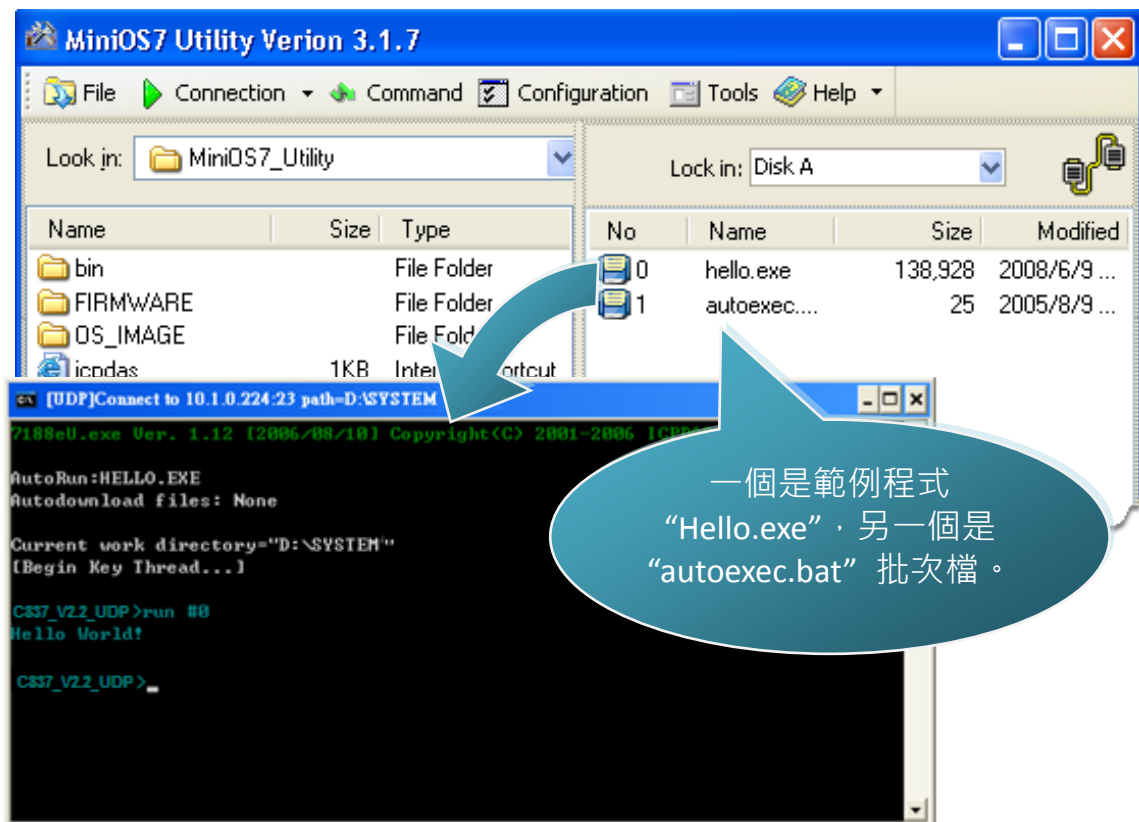
步驟 13: 建立自動執行檔 (autoexec.bat)。



- i. 開啟 “Notepad”。
- ii. 輸入 “HELLO.exe”。
- iii. 將檔案儲存為 “autoexec.bat”。

步驟 14: 使用 MiniOS7 Utility 將程式上傳至 ViewPAC。

請參閱章節 “2.4.1. 建立連線”，取得此設定之詳細資訊。



4. API 與參考範例

下列是專為 ViewPAC 設計的 API 與範例程式。您可檢視這些 API 與範例程式的原始碼，裡面含有許多的函式與註解，可讓您熟悉 MiniOS7 的 API 並可藉由修改範例程式快速地開發自己的應用程式。

API 功能分類表：

API 說明	標頭檔		函式庫	
	VP-2111	VH-2110	VP-2111	VH-2110
CPU	vp2k.h	VH2K.h	vp2k.lib	VH2K.lib
LCD	vp2k_lcd.H		vp2k_lcd.lib	
Ethernet	Tcpi32.h		tcp_dm32.lib	
64MB Flash Disk	MFS.h	None	MFS_V211.lib	None
Framework	MFW.H		FW_09314.lib	

請參閱下列位置，以取得更多關於 ViewPAC API 的詳細資訊：

CD:\NAPDOS\vp-2000\demo\Readme.txt

<http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/readme.txt>

以下所介紹的核心 API 是已整合至 ViewPAC API Set 之 MiniOS7 API。

函式庫 (Functions Library) — VH-2110: vh2k.lib · VP-2111: vp2k.lib

此檔案包含了 MiniOS7 API (應用程式介面) 與數百個和 ViewPAC 相關的預定義函式。

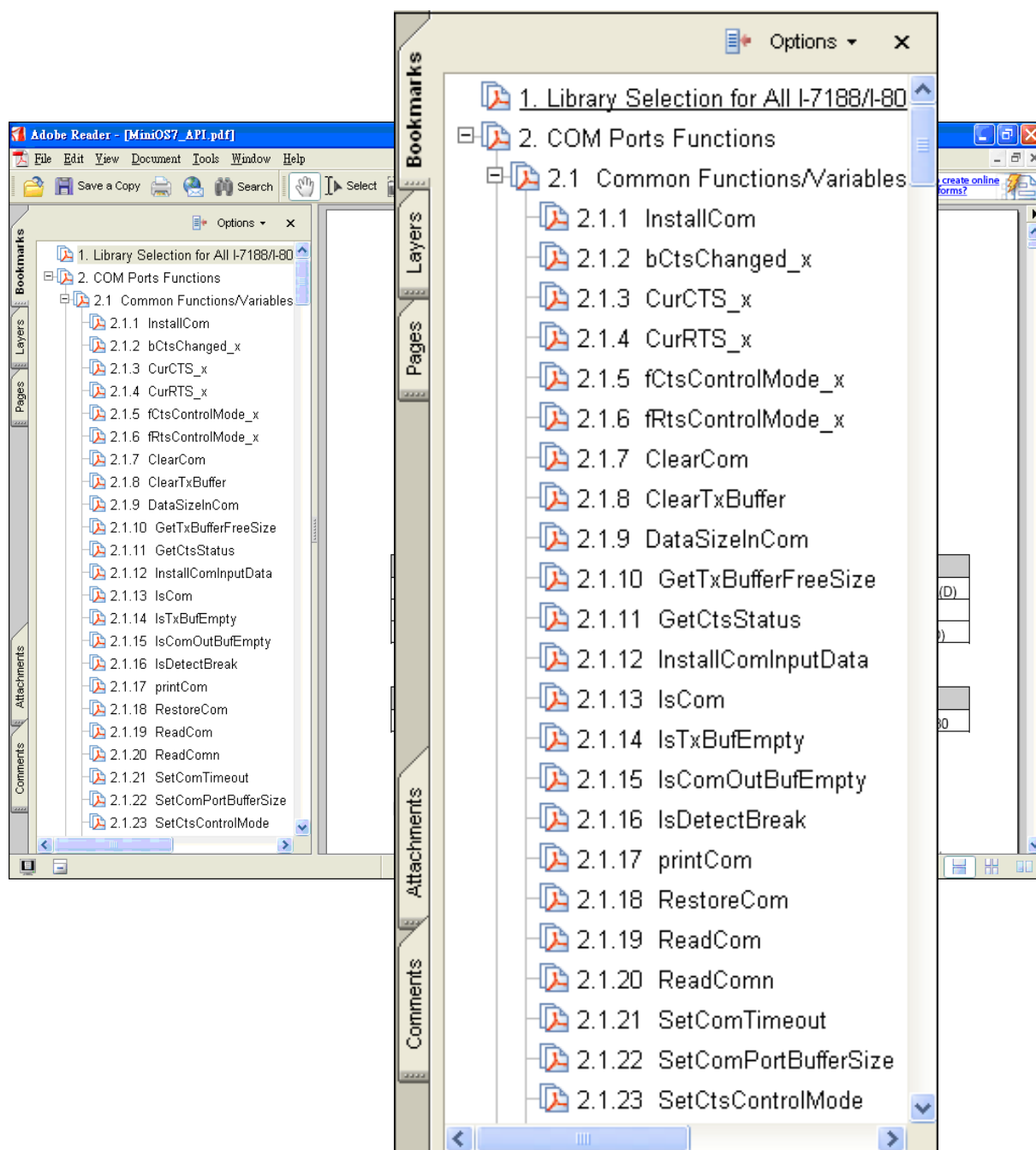
標頭檔 (Header File) — VH-2110: vh2k.h · VP-2111: vp2k.h

此檔案包含了用於 MiniOS7 API 之副程式、變數與其它識別符號的前置宣告。



請參閱下列位置的“MiniOS7 的 API 函式使用手冊”，以取得關於函式的說明、原型 (prototype) 與引數的完整使用資訊：

CD:\Napdos\MiniOS7\Document\
<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/minios7/document/>



► 用於 LCD 顯示器與聲音控制之範例程式



資料夾	範例	說明
lcd_key	Showchar	顯示字元
	showbmp	顯示 BMP 圖片
	showTC	顯示繁體中文字型
	lcd_backlight	調整 LCD 背光
	cursor	顯示游標
	keystatus	顯示 key down/up 的狀態
	presskey	顯示 key 值
	testkey	顯示 key 值
	ViewPAC	所有的功能測試
	showSC	顯示簡體中文字型
sound	change_key_sound	按鍵來改變聲音
	sound1	按鍵來改變聲音

► 基本應用之範例程式

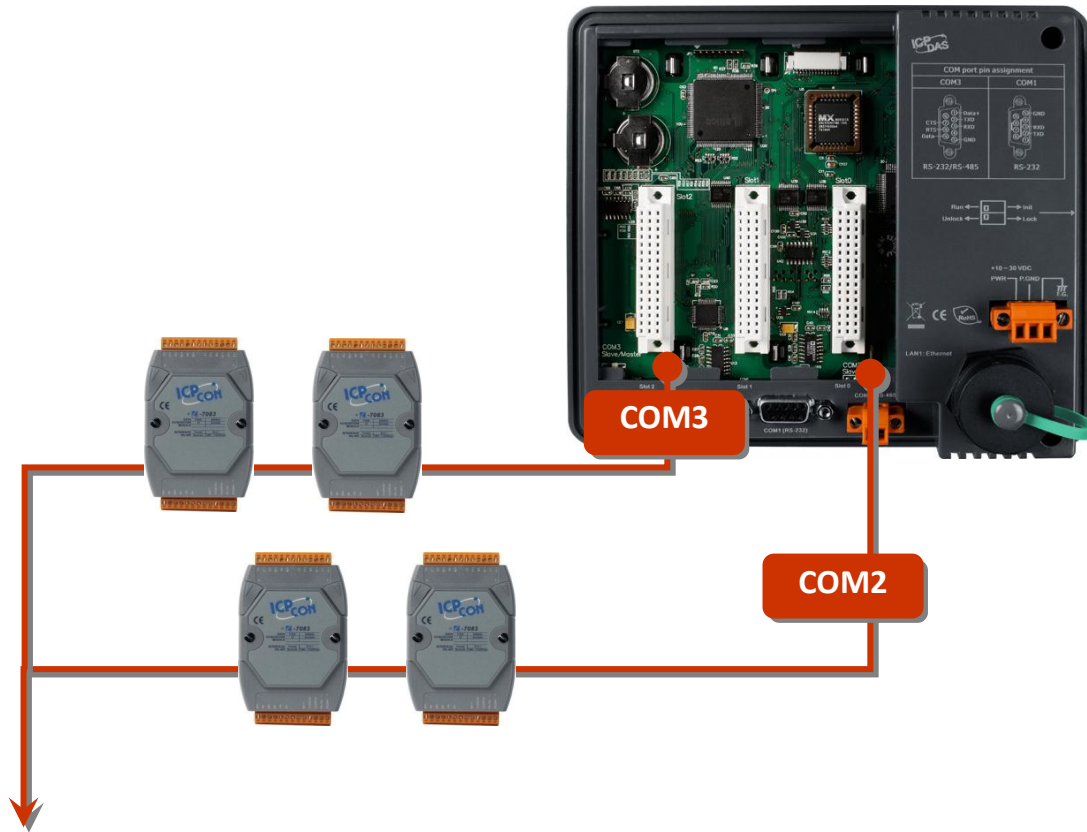
資料夾	範例	說明
File	Config_1_Basic	讀取文字檔中的資訊 (基本)
	Config_2_Advanced	讀取設定檔 (文字檔) (進階)
Hello	Hello_C	讀取函式庫版本與快閃記憶體容量
	Hello_C++	讀取函式庫版本與快閃記憶體容量
Misc	Reset	軟體的重置
	Runprog	選取項目 (item) 並執行
	SerialNumber	取得 64-bit 硬體唯一序號
	Watchdog	啟用/關閉/刷新 WDT 函式
Misc\Memory	512k_flash	可從 Flash 讀取/寫入整數、浮點數與字串資料
	512k_SRAM	可從 SRAM 讀取/寫入資料
	eeprom	可從 EEPROM 讀取/寫入整數、浮點數資料
DateTime	DateTime	示範如何從 RTC 讀寫日期與時間
Timer	Timer	請參閱下列位置: ftp://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/8000/common/minios7/demo/
Com_Ports	C_Style_IO	(1) 示範如何寫資料到輸入暫存區 (2) 示範如何接收一個字串(string) (3) 示範如何使用 C function: scanf 或只使用 Scanf()
	Receive	從 COM Port 接收資料 Slv_COM.c 是 non-blocked 模式 Receive.c 是 blocked 模式
	Slv_COM	slave COM Port 範例，使用於 請求/應答 (request/reply) 或 命令/回應 (command/response) 的應用。
	ToCom_In_Out	說明如何透過 COM Port 讀/寫 byte 資料

► 用於 64 MB 快閃記憶體之範例程式 (僅適用於型號 VP-2111)



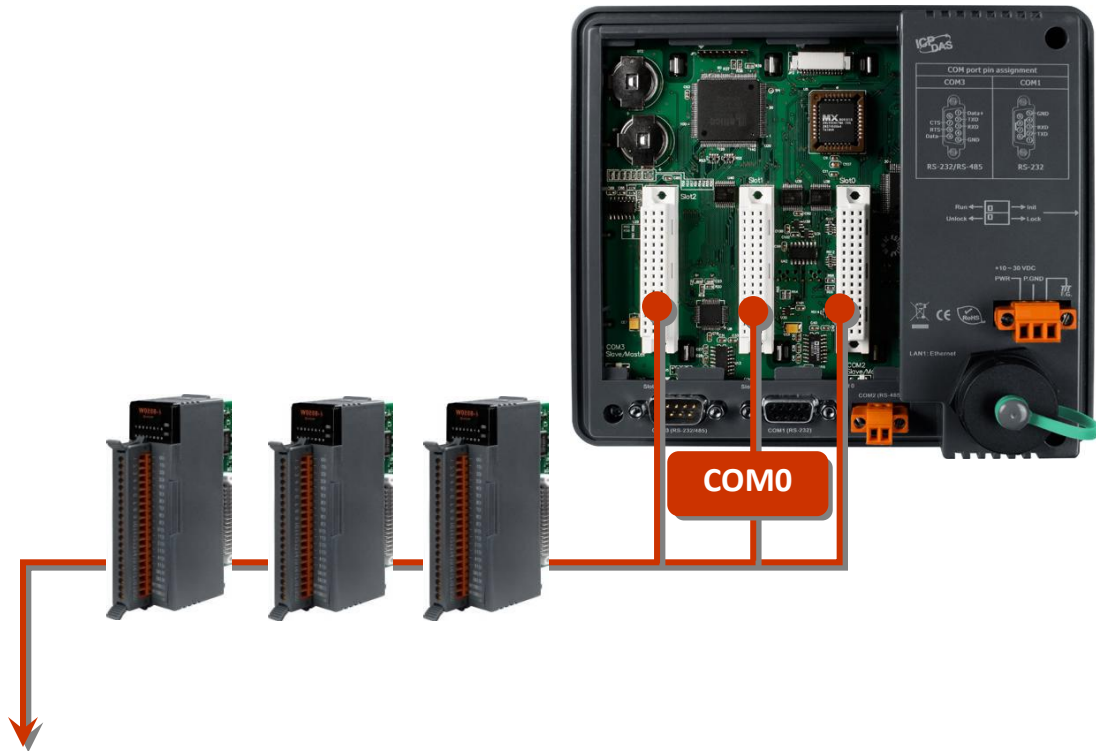
資料夾	範例	說明
64MB_Flash	Gets	如何取得 64 MB 快閃記憶體中檔案的字串。
	mFS_QA	MiniOS7 檔案系統的品質保證計畫，包括功能測試、讀取 / 寫入效能測試。
	Puts	示範如何寫入字串至 64 MB 快閃記憶體中的檔案。
	Utility	MiniOS7 檔案系統中的軟體工具，包括 Dir、Read、Write... 等操作。

► 用於擴展 I/O 之範例程式



資料夾	範例	說明
7K87K_for_COM	7K87K_demo_for_com	使用者可使用 "Com port" 來連接並控制 I-7K 或 I-87K 模組。 COM Port: *8410/8810/8411/8811 可使用 COM 2, COM 3; *8430/8830/8431/8831(CPU 40 M / 80 M) 可使用 COM 3, COM 4
	7K87K_DI_for_Com	
	7K87K_DO_for_Com	
	7K87K_DIO_for_Com	
	7K87K_AI_for_Com	
	AO_22_26_for_Com	
	AO_024_for_Com	

► 用於 I/O 插槽之範例程式 (僅適用於型號 VP-2111)



資料夾	範例	說明
IO_in_Slot	8K_DI	此範例程式用於 I-8K DI 模組。 例如: I-8040W · I-8051W。
	8K_DO	此範例程式用於 I-8K DO 模組。 例如: I-8041W · I-8056W。
	8K_DIO	此範例程式用於 I-8K DIO 模組。 例如: I-8042W · I-8054W。
	8050w	I-8050W 是 16 通道通用型數位 I/O 模組。 I/O 類型: 透過編程來選取 I/O。 使用者可參考範例來編寫 I-8050W 的程式碼。
	8017hw	I-8017HW 是 14 位元, 取樣率為 100K 的 8 通道類比輸入模組。使用者可參考範例來編寫 I-8017HW 的程式碼。

資料夾	範例	說明
IO_in_Slot	8024w	I-8024W 是 4 通道隔離型類比輸出模組。使用者可參考範例來編寫 I-8024W 的程式碼。
IO_in_Slot	87K_demo	此範例程式用於 COM0 上 I-87K 模組的一般功能。
	87K_DI	此範例程式用於 COM0 上 I-87K 數位輸入模組。 例如: I-87040W · I-87051W。
	87K_DO	此範例程式用於 COM0 上 I-87K 數位輸出模組。 例如: I-87041W · I-87057W。
	87K_DIO	此範例程式用於 COM0 上 I-87K 數位輸入/輸出模組。例如: I-87042W · I-87054W · I-87055W。
	87K_AI	此範例程式用於 COM0 上 I-87K 類比輸入模組。 例如: I-87013W · I-87015W · I-87017W · I-87018W。
	8088w	I-8088W 是擁有 8 個 PWM 輸出通道與 8 個數位輸入的模組。使用者可參考範例來編寫 I-8088W 的程式碼。
	87024w	此範例程式用於 I-87024W 類比輸出模組。
	Find_IO	此範例程式為基本的功能。您可知道如何在 I-8000 系列的微控制器 (MCU) 中取得 I/O 卡的資訊。
IO_in_Slot/ 8084w/ BC_Demo	ABPhase	此範例適用於 I-8084W (Pulse/Dir 模式)。
	Freq	此範例適用於 I-8084W (Frequency 模式)。
	PulseDir	此範例適用於 I-8084W (Pulse/Dir 模式)。
	Up	此範例適用於 I-8084W (Up counter 模式)。
	Updown	此範例適用於 I-8084W (Up/Down 模式)。

請參閱以下位置，取得更多關於 I/O 擴充模組之 API 的詳細資訊：

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\lib

<http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/lib/>

4.1. COM Port 之 API

ViewPAC 內建有多個 COM Port，如下圖所示。

► VP-2111



► VH-2110



► VH-2111P/VH-2211/VH-2311



4.1.1. COM Port 函式類型

以下為兩種適用於 COM Port 函式的類型：

1. MiniOS7 COM Port 函式
2. (C style) 標準 COM Port 函式

小技巧 與 安全警告



(C style) 標準 COM Port 函式只能在 COM1 使用。若需使用 COM1，您必須在 MiniOS7 COM Port 函式或 (C style) 標準 COM Port 函式中選擇其一。選擇了其中一項，另一項則無法使用。

MiniOS7 COM Port 函式與 (C style) 標準 COM Port 函式之比較結果：

函式類型	COM Port	Buffer	函式				
MiniOS7 COM Port	1、2,..等	1 KB	1 KB	IsCom()	ToCom()	ReadCom()	printCom()
(C style) 標準 COM Port	1	512 Bytes	256 Bytes	Kbhit()	Puts() Putch()	Getch()	Print()

4.1.2. MiniOS7 COM Port 之 API

COM Port 之 API

1. InstallCom()

使用 COM Port 之前，必須先呼叫 InstallCom() 函式，將 COM Port 初使化。

2. RestoreCom()

若程式呼叫了 InstallCom()，於使用完後，需再呼叫 RestoreCom() 函式來釋放 COM Port 的佔用。

檢查 COM Port 輸入暫存器內是否有資料之 API

3. IsCom()

從 COM Port 讀取資料之前，需呼叫 IsCom() 函式，檢查目前 COM Port 輸入暫存器內是否有資料。

從 COM Port 讀取資料之 API

4. ReadCom()

以 IsCom() 函式確認輸入暫存器內有資料後，需呼叫 ReadCom() 函式，讀取 COM Port 輸入暫存器內的資料。

傳送資料給 COM Port 之 API

5. ToCom()

傳送資料至 COM Port 之前，須先呼叫 ToCom() 函式。

範例 – 透過 COM1 來讀取與接收資料。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{

int quit=0 , data;

InitLib(); /* 初始化 ViewPAC 的程式庫 */

InstallCom(1 , 115200 , 8 , 0 , 1); /* 初始化 COM1 */

while(!quit)
{
if(IsCom(1)) /* 檢查 COM Port 輸入暫存器是否有資料 */
{
data=ReadCom(1); /* 從 COM1 Port 讀取資料 */
ToCom(1 , data); /* 透過 COM1 Port 傳送資料 */
if(data=='q') quit=1; /* 若接收到 'q' , 則離開程式 */
}
}
RestoreCom(1); /* 釋放 COM1 */
}
```

由 COM Port 顯示資料之 API

6. printCom()

在 C 函式庫中，像是 printfCom() 函式，可經由 COM Port 輸出資料。

範例 - 由 COM1 Port 顯示資料。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    int i;

    /* 初始化 ViewPAC 的程式庫 */
    InitLib();
    InstallCom(1, 115200, 8, 0, 1); /* 初始化 COM1 */
    for (i=0;i<10;i++)
    {
        printCom(1,"Test %d\n\r", i);
    }
    Delay(10); /* 等待所有資料傳送到 COM Port */
    RestoreCom(1);
}
```


4.1.3. 標準 COM Port 之 API

標準 COM Port 用來從 PC 上傳程式到 ViewPAC。

小技巧 與 安全警告



(C style) 標準 COM Port 函式只能使用 COM1，而 COM1 的通訊設定需固定如下：

Baudrate = 115200 bps · Data format = 8 bits

Parity check = none · Start bit = 1 · Stop bit = 1

檢查輸入暫存器內是否有資料之 API

1. Kbhit()

從標準 I/O Port 讀取資料前，需先呼叫 kbhit() 函式，檢查目前輸入暫存器中是否有資料。

從標準 I/O Port 讀取資料之 API

2. Getch()

以 kbhit() 函式確認輸入暫存器有資料後，需呼叫 Getch() 函式，讀取輸入暫存器內的資料。

傳送資料至標準 I/O Port 之 API

3. Puts() – 用來傳送字串

呼叫 Puts()，可傳送字串資料到標準 COM Port。

4. Putch() – 用來傳送一個字元

呼叫 Putch() 函式，可傳送字元資料到 COM Port。

由標準 I/O Port 顯示資料之 API

5. Print()

C 函式庫中，像是 Print() 函式，可經由 COM Port 輸出資料。

範例 – 透過 COM1 來讀取與接收資料。

```
#include<stdio.h>
#include "vp2k.h"

void main(void)
{
    int quit=0 , data;

    InitLib(); /* 初始化 ViewPAC 的程式庫 */
    while(!quit)
    {
        if(Kbhit()) /* 檢查輸入暫存器內是否有資料 */
        {
            data=Getch(); /* 從 COM1 Port 讀取資料 */
            Putch(data); /* 傳送資料至 COM1 */
            if(data=='q') quit=1; /* 若接收到 'q'，則離開程式 */
        }
    }
}
```

範例 – 透過 COM1 來顯示資料。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    int i;

    /*初始化 ViewPAC 的程式庫 */
    InitLib();
    for(i=0;i<10;i++)
    {
        Print("Test %d\n\r",i);
    }
}
```

4.1.4. COM Port 函式之對照

範例 – 學習如何顯示 ASCII 碼。

MiniOS7 COM Port 函式	標準 COM Port 函式
<pre>#include<stdio.h> #include "vp2k.h" void main(void) { unsigned char item; InitLib(); InstallCom(1 · 115200 · 8 · 0 · 1); printCom(1,"Hits any key.\n"); printCom(1,"Hit the ESC to exit!\n"); for(;;) { if(IsCom(1)) { item=ReadCom(1); if(item=='q') { return; } } else {</pre>	<pre>#include<stdio.h> #include "vp2k.h" void main(void) { unsigned char item; InitLib(); Print("Hits any key.\n"); Print("Hits the ESC to exit !\n"); for(;;) { if(kbhit()) { item=Getch(); if(item=='q') { return; } } else { Print("-----\n\r");</pre>

<pre> printCom(1,"-----\n\r"); printCom(1,"char:"); ToCom(1,item); printCom(1,"\n\rASCII(%c)\n\r",item); printCom(1,"Hex(%02X)\n\r",item); } } } Delay(10); RestoreCom(1); } </pre>	<pre> Print("char:"); Putch(item); Print("\n\rASCII(%c)\n\r",item); Print("Hex(%02X)\n\r",item); } } } } </pre>
--	---

4.1.5. COM Port 請求/回應 的通訊協定定義

請求/回應 (Request/Response) 通訊方式是一種很典型的通訊協議架構。若您想設計出通訊協定的指令集 (如下表)，請參考“slave_com”範例。

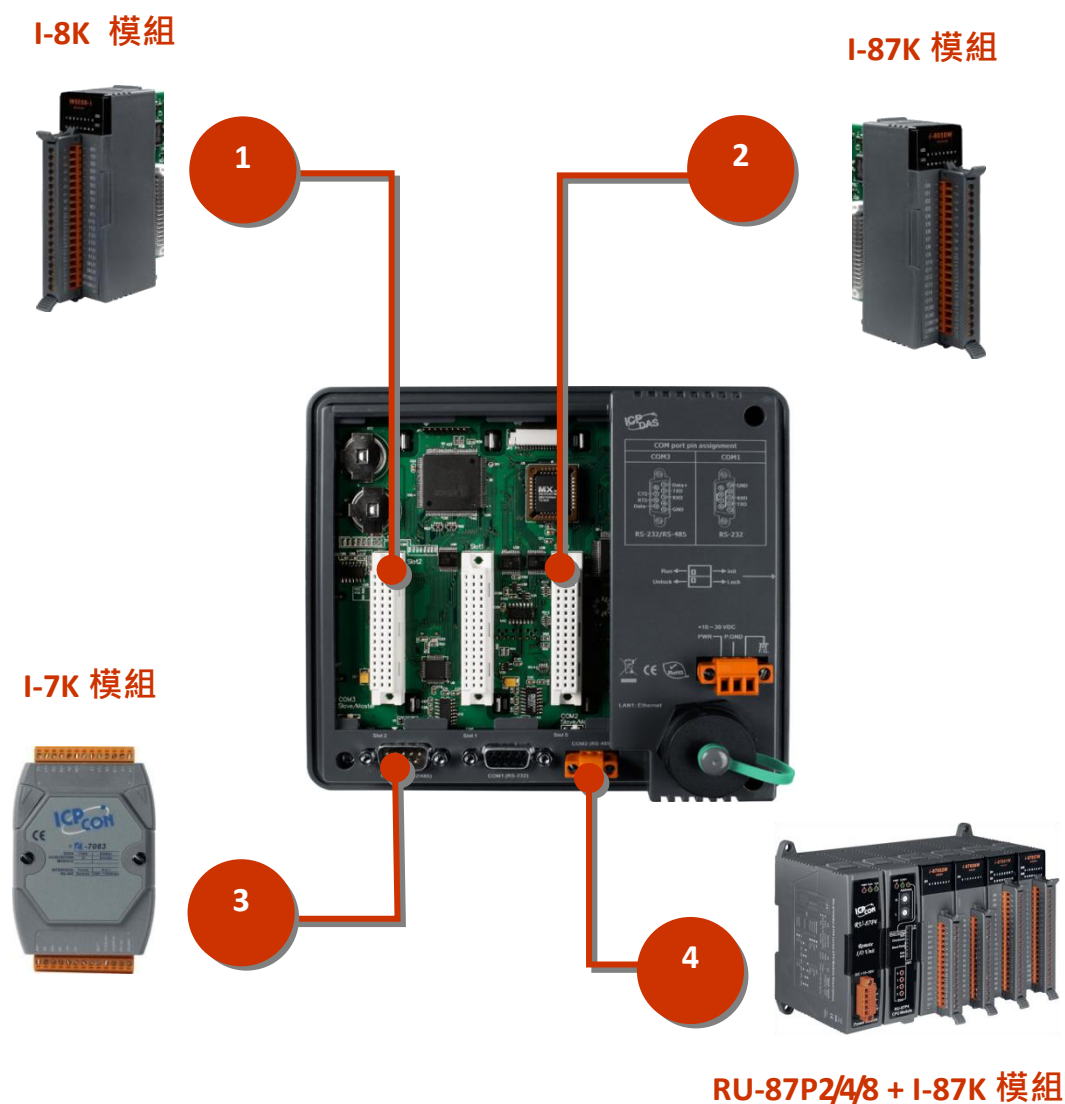
有關 請求/回應的應用，請參考
“slave_com” 範例。



請求 (Request)	回應 (Response)
c1	除錯資訊: 指令 1 (Command1) 指令 1 (Command1)
c2	除錯資訊: 指令 2 (Command2) 指令 2 (Command2)
Q	除錯資訊: 快速程式 (Quick program)
其它指令	除錯資訊: 未知指令 (Unknown command)

4.2. 用於 I/O 模組之 API

- ▶ ViewPAC 擁有 3 個 I/O 插槽，可用來存取 I-8k 與 I-87k 系列 I/O 模組 (高卡版本，如下圖第 1、2 點所示)。(僅適用於型號 VP-2111)
- ▶ ViewPAC 擁有多序列埠，適用於在廣大範圍的 RS-485 網路應用中存取 I-7K 系列 I/O 模組，如下圖第 3 點所示。(僅適用於型號 VP-2111)
- ▶ ViewPAC 可連接至 RU-87P2/4/8 並透過 RS-485 來存取 I-87k I/O 系列模組，如下圖第 4 點所示。



適用於 I-7K、I-8k 與 I-87k I/O 模組之範例程式，可區分為下列兩種：

1. 安裝在插槽上的 I-8k 與 I-87k I/O 模組，請參閱：

VH-2110:

CD:\NAPDOS\vp-2000\demo\vh-2110\Basic\IO_in_Slot\

http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vh-2110/basic/io_in_slot/

VP-2111:

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\IO_in_Slot\

http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/io_in_slot/

2. 由 COM Port 連接之 I-7K 與 I-87k I/O 模組，請參閱：

VH-2110:

CD:\NAPDOS\vp-2000\demo\vh-2110\Basic\7K87K_for_COM\

http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vh-2110/basic/7k87k_for_com/

VP-2111:

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\7K87K_for_COM\

http://ftp.icpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/7k87k_for_com/

4.2.1. 於插槽上使用 I-8K 系列 I/O 模組 (適用型號: VP-2111)

讀取 DI 模組之 API



DI_8()、DI_16()、DI_32()

需呼叫 DI_8()、DI_16()、DI_32() 函式來讀取 DI 模組的輸入值。

傳送指令至 I-8K 系列 I/O 模組。

範例：讀取插槽 3 之 DI 模組的輸入值。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    Int DI_data, iSlot=3;
    InitLib(); /*初始化 vp2k 函示庫*/
    For(;;)
    {
        DI_data=DI_8(iSlot); /*讀取插槽 3 的 DI 模組輸入值*/

        printf("DI Status==%x\n\r", DI_data);
    }
}
```

4.2.2. 於插槽上使用 I-87K 系列 I/O 模組 (適用型號: VP-2111)

請依下列步驟，使用插槽上的 I-87K 系列 I/O 模組：

1. 使用 Installcom()，將 COM Port 初始化。
2. 使用 ChangeToSlot()，指定 I-87k I/O 模組所使用的插槽位置。
3. 使用 SendCmdTo7000(0,...)，傳送指令。
4. 使用 ReceiveResponseFrom7000_ms()，取得回應。
5. 使用 RestoreCom()，釋放所佔用的 COM Port。



小技巧與安全警告



COM0 的設定值固定為：

Baud rate = 115200 bps · Data bit = 8 bits,

Parity check = None · Stop bit = 1

插槽上之 I-87k 系列 I/O 模組的設定值固定為：

Address = 0 · Check sum = Disable

此外，需呼叫 ChangeToSlot() 函式。

範例：傳送 '\$00M' 指令至插槽 7 上的 I-87k I/O 模組以取得模組名稱。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    unsigned char InBuf0[60];
    InitLib(); /*初始化 vp2k 函式庫*/
    InstallCom(0 · 115200 · 8 · 0 · 1); /*初始化 COM0 */
    InstallCom(1 · 115200 · 8 · 0 · 1); /*初始化 COM1 */

    ChangeToSlot(7);
    SendCmdTo7000(0 · "$00M" · 0); /*傳送指令給 COM0*/

    /*Timeout=50ms · check sum= disabled*/

    ReceiveResponseFrom7000_ms(0 · InBuf0 · 50 · 0);

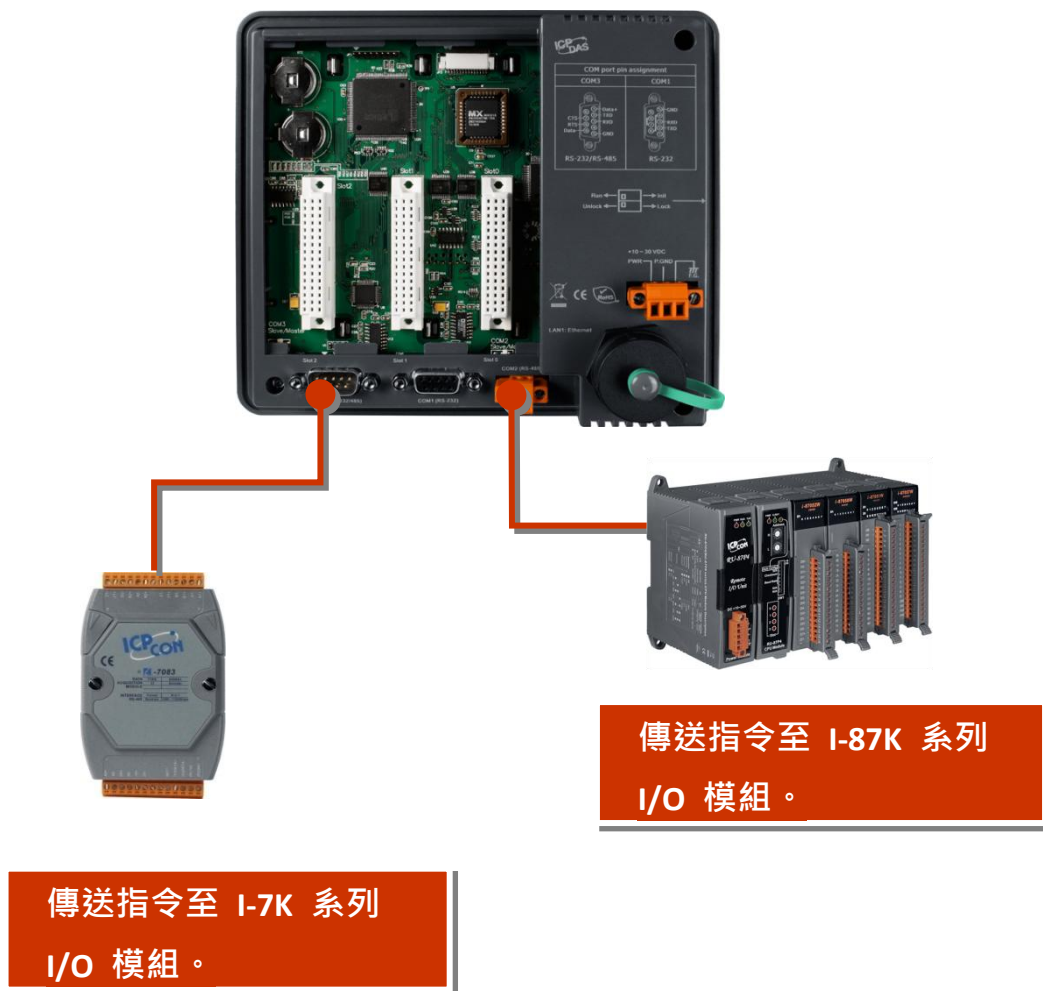
    printCom(1 · "Module Name=%s" · InBuf0);

    Delay(10); /*等待所有資料傳送至 COM Port*/
    RestoreCom(0); /*釋放 COM0 */
    RestoreCom(1); /*釋放 COM1 */
}
```

4.2.3. 使用連接於 COM Port 的 I-7K 與 I-87K 系列 I/O 模組

請依下列步驟，使用連接於 COM Port 的 I-7K 與 I-87K 系列 I/O 模組：

1. 使用 Installcom()，初始化 COM Port。
2. 使用 SendCmdTo7000(0,...)，傳送指令。
3. 使用 ReceiveResponseFrom7000_ms()，取得回應。
4. 使用 RestoreCom()，釋放 COM Port。



範例：傳送 '\$00M' 指令給連接於 COM2 的 I-87k I/O 模組以取得模組名稱。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    unsigned char InBuf0[60];
    InitLib(); /*初始化 vp2k 函式庫*/
    InstallCom(1, 115200, 8, 0, 1); /*初始化 COM1 */
    InstallCom(2, 115200, 8, 0, 1); /*初始化 COM2 */

    SendCmdTo7000(2, "$00M", 0); /*傳送指令給 COM2*/

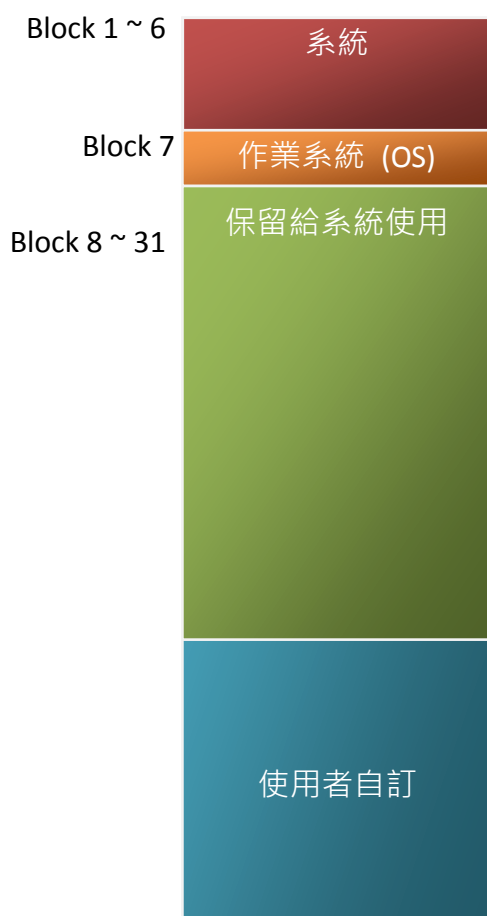
    /*Timeout=50ms · check sum=disabled*/

    ReceiveResponseFrom7000_ms(2, InBuf0, 50, 0);
    printCom(1, "Module Name=%s", InBuf0);
    Delay(10); /*等待所有資料傳送至 COM Port */
    RestoreCom(1); /*釋放 COM1 */
    RestoreCom(2); /*釋放 COM2 */
}
```

4.3. 用於 EEPROM 之 API

- EEPROM 分為 64 個區塊 (block 0 ~ 63) , 而每個區塊有 256 byte (位址 0 ~ 255) , 總容量為 16,384 byte (16 KB) 。
- EEPROM 預設為防寫模式 。
- EEPROM 的系統程式與作業系統儲存位置 , 如下圖所示 。

寫入資料至 EEPROM 之 API



1. `EE_WriteEnable()`
將資料寫入至 EEPROM 之前 , 需先呼叫 `EE_WriteEnable()` 來啟用 EEPROM 的寫入功能 。
2. `EE_WriteProtect()`
完成 EEPROM 的資料寫入後 , 需呼叫 `EE_WriteProtect()` 函式 , 啟用 EEPROM 的防寫功能 。
3. `EE_MultiWrite()`
使用 `EE_WriteEnable()` 啟用 EEPROM 的寫入功能後 , 需再呼叫 `EE_MultiWrite()` 來寫入資料 。

從 EEPROM 讀取資料之 API

4. `EE_MultiRead()`

此函式可從 EEPROM 中讀取多個 byte 的資料 。

範例：寫入資料至 EEPROM 的區塊 1、位址 10：

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    int data=0x55, data2;

    InitLib(); /* 初始化 ViewPAC 函式庫 */

    EE_WriteEnable();
    EE_MultiWrite(1,10,1,&data);
    EE_WriteProtect();

    EE_MultiRead(1,10,1,&data2); /* 現在 data2=data=0x55 */
}
```

請參閱以下位置，取得更多關於 EEPROM 的範例程式：

VP-2111:

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\Misc\

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/misc/>

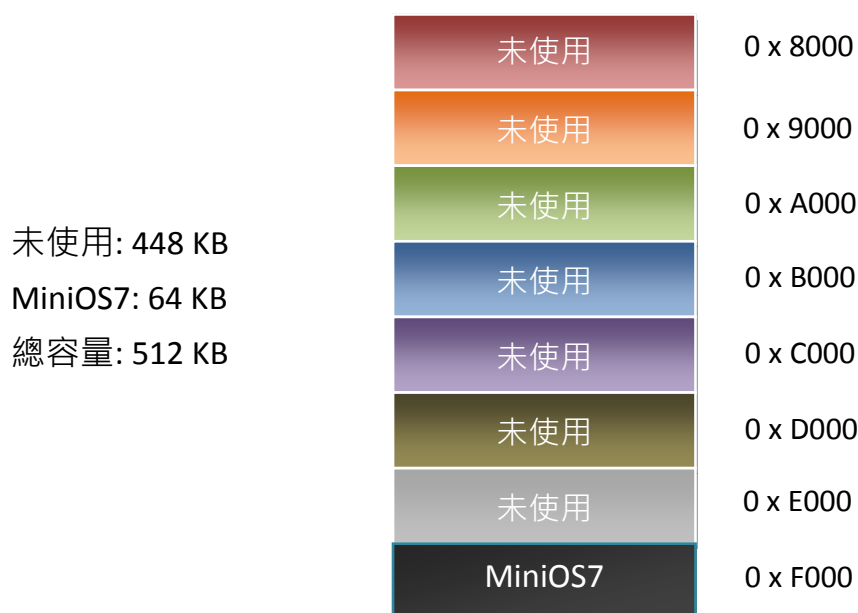
VH-2110:

CD:\NAPDOS\vp-2000\demo\vh-2110\Basic\Misc\

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vh-2110/basic/misc/>

4.4. 用於快閃記憶體之 API

- ViewPAC 控制器配備有一個 512 KB 的快閃記憶體 (Flash Memory)。
- MiniOS7 使用了最後 64 KB ; 記憶體其餘部分被用來儲存使用者程式或資料。
- 快閃記憶體只能在空的位址被寫入，因此在進行寫入動作之前需先抹除 (Erase) 記憶體，即表示將所有的資料位元設定為 1，此步驟完成，即可寫入新的資料。



抹除快閃記憶體之 API

1. EraseFlash()

寫入資料前，您需使用 EraseFlash() 函式來抹除快閃記憶體中的區塊。(即將所有資料位元皆設為 1)

將資料寫入快閃記憶體之 API

2. FlashWrite()

呼叫 FlashWrite() 函式，可將資料寫入至快閃記憶體。

從快閃記憶體讀取資料之 API

2. FlashWrite()

呼叫 FlashRead() 函式，可從快閃記憶體讀取資料。

範例：將整數值寫入至快閃記憶體的區段 0xD000、偏移位址 (offset) 為 0x1234。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    int data=0xAA55 · data2;
    char *dataptr;
    int *dataptr2;

    InitLib(); /* 初始化 ViewPAC 函式庫*/
    EraseFlash(0xd000); /* 清除快閃記憶體的一個區段 */
    dataptr=(char *)&data;
    FlashWrite(0xd000,0x1234 · *dataptr++);
    FlashWrite(0xd000,0x1235 · *dataptr);

    /* 讀取快閃記憶體的資料 (方法一) */
    dataptr=(char *)&data2;
    *dataptr=FlashRead(0xd000,0x1234);
    *(dataptr+1)=FlashRead(0xd000,0x1235);

    /* 讀取快閃記憶體的資料 (方法二) */
    dataptr2=(int far *)_MK_FP(0xd000,0x1234);
    data=*data;
}
```

4.5. 用於 NVRAM 之 API

- ViewPAC 配備有一個 RTC (Real Time Clock) 和一個用來儲存資料的 31 bytes NVRAM。
- NVRAM 其實是 SRAM，但它使用了電池來保持資料。因此，斷電時 NVRAM 中的資料不會遺失。
- NVRAM 沒有寫入次數的限制 (Flash 與 EEPROM 皆有寫入限制)。若未發生漏電流(leakage current) 的狀況，電池可使用 10 年。

將資料寫入 NVRAM 之 API

1. WriteNVRAM()

需呼叫 WriteNVRAM() 函式，以將資料寫入至 NVRAM。

從 NVRAM 讀取資料之 API

2. ReadNVRAM()

需呼叫 ReadNVRAM() 函式，以讀取 NVRAM 中的資料。

範例：使用以下程式碼來將資料寫入 NVRAM 的位址 0。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    int data=0x55 · data2;

    InitLib(); /* 初始化 ViewPAC 函式庫*/
    WriteNVRAM(0,data);
    data2=ReadNVRAM(0); /*現在 data2=data=0x55 */
}
```

範例：以下程式碼可用來將整數值 (2 byte) 寫入至 NVRAM。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    int data=0xAA55 · data2;
    char *dataptr=(char *)&data;

    InitLib(); /* 初始化 ViewPAC 函式庫 */
    WriteNVRAM(0 · *dataptr); /*寫入低位位元組*/
    WriteNVRAM(1 · *dataptr+1); /*寫入高位位元組*/
    dataptr=(char *) &data2;
    *dataptr=ReadNVRAM(0); /*讀取低位位元組*/
    (*dataptr+1)=ReadNVRAM(1); /*讀取高位位元組*/
}
```

4.6. 用於計時器之 API

ViewPAC 的 OS 提供一個系統的計時信號 (Time tick) ，其為 16 位元且精度為 1 ms ，可用來支援 8 個碼錶計時器 與 8 個倒數計時器。

開始使用計時器 (Timer) 之 API

1. TimerOpen()

使用計時器功能之前，需在程式的開頭呼叫 TimerOpen() 函式。

讀取計時器之 API

2. TimerResetValue()

讀取計時器之前，需呼叫 TimerResetValue() 函式，將 Time Tick 重置為 0。

3. TimerReadValue()

使用 TimerResetValue() 函式，將 Time Tick 重置為 0 後，需呼叫 TimerReadValue() 函式，來讀取 Time Tick。

停止計時器之 API

4. TimerClose()

程式的結尾，需呼叫 TimerClose() 函式，來停止計時器。

範例：以下程式碼可用來從 0 開始讀取 Time Tick。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    Unsigned long time iTime;

    InitLib(); /* 初始化 ViewPAC 函式庫*/
    TimerOpen();
    While(!quit)
    {
        If(Kbhit())
            TimerResetValue(); /*重置 Time Tick 為 0*/

        iTime=TimerReadValue(); /*由 0 開始讀取 Time Tick*/
    }
    TimerClose(); /*停止 ViewPAC 的計時器功能*/
}
```

請參閱以下位置，取得更多關於計時器的範例程式：

VP-2111:

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\Timer\

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/timer/>

VH-2110:

CD:\NAPDOS\vp-2000\demo\vh-2110\Basic\Timer\

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vh-2110/basic/timer/>

4.7. 用於看門狗計時器 (WDT) 之 API

- ViewPAC 配備有 MiniOS7 - 小型核心作業系統。MiniOS7 採用計時器 2 (CPU 內部計時器) 為系統計時器，它是 16-bits 的計時器且每 1 ms 產生一次中斷，因此系統精度為 1 ms。
- 看門狗計時器始終處於啟用狀態，並由系統計時器 ISR (中斷服務常式) 刷新它。
- 當系統發生停滯或錯誤時，可使用看門狗計時器將系統重置以恢復正常。其逾時週期為 0.8 秒。

開始使用看門狗功能 (WDT) 之 API

1. EnableWDT()

啟動 WatchDog Timer 功能，當系統發生遲緩或停擺時可立即將系統重置，確保系統正常工作。

2. RefreshWDT()

更新 WatchDog Timer 狀態，使用 WatchDog Timer 功能將系統重置後，需再呼叫 RefreshWDT() 回復系統。

3. DisableWDT()

關閉 WatchDog Timer 功能。

範例：刷新看門狗計時器。

```
#include <stdio.h>
#include "vp2k.h"

void main(void)
{
    Unsigned long time iTime;

    InitLib(); /* 初始化 ViewPAC 函式庫*/

    Enable WDT();
    While(!quit)
    {
        RefreshWDT();
        User_function();
    }
    DisableWDT();
}
```

請參閱以下位置，取得更多關於看門狗計時器的範例程式：

VP-2111:

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\Timer\

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/timer/>

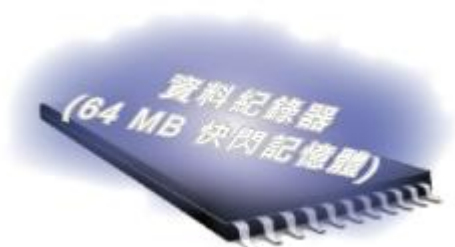
VH-2110:

CD:\NAPDOS\vp-2000\demo\vh-2110\Basic\Timer\

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vh-2110/basic/timer/>

4.8. MFS 之 API

(適用型號: VP-2111)



所需之函式庫與標頭檔：

MFS_V2111.lib 與 MFS.h

VP-2111 配備有額外的 64 MB 快閃記憶體，MFS 是設計用來由 64 MB 快閃記憶體讀取/寫入檔案。

請參閱章節“附錄 C. 什麼是 MiniOS7 檔案系統 (MFS)”·取得所有關於硬體支援、應用與規格之詳細使用資訊。

• MFS 函式功能說明:

函式	說明
mfs_Init	初始化檔案系統。
mfs_Stop	關閉時釋放已分配的緩衝區 (Buffer)。
mfs_ResetFlash	初始化檔案系統。(所有檔案將移失)
mfs_X600Fs_GetLibVersion	取得函式庫的版本編號。
mfs_GetLibDate	取得函式庫的建立日期。
mfs_GetFileNo	取得 NAND Flash 中所有的檔案數。
mfs_GetFreeSize	取得可添加檔案的空間大小。
mfs_GetBadSize	取得無法使用空間的大小。
mfs_GetUsedSize	取得已使用空間的大小。
mfs_GetFileSize	取得儲存在 NAND Flash 中的檔案大小。
mfs_GetFileInfoByName	使用指定的檔名來檢索檔案資訊。
mfs_GetFileInfoByNo	使用檔案編號索引來檢索檔案資訊。
mfs_DeleteAllFiles	刪除所有儲存在 NAND Flash 中的檔案。
mfs_DeleteFile	刪除一個已寫入 NAND Flash 中的指定檔案。
mfs_OpenFile	1. 依檔名開啟檔案。 2. 建立新檔。

函式	說明
mfs_CloseFile	依檔案控制代碼(File Handle)來關閉檔案，關閉前會先將緩衝區的資料寫至檔案中，以確保檔案完整性。
mfs_ReadFile	從檔案中讀取資料的特定 byte。
mfs_WriteFile	附加資料中的特定 byte 至檔案中。
mfs_Getc	從檔案中取得字元。
mfs_Putc	輸出一個字元至檔案中。
mfs_Gets	從檔案中取得一個字串。
mfs_Puts	輸出一個字串至檔案中。
mfs_EOF	檢查檔案指標(File Pointer) 是否已到檔案結尾(EOF)。
mfs_Seek	重定位檔案的指標。
mfs_Tell	回傳目前檔案的指標。
mfs_EnableWriteVerify	啟用資料驗證功能。 預設，資料驗證為啟用狀態。
mfs_DisableWriteVerify	關閉資料驗證功能。

開始使用 64 MB 快閃記憶體之 API

1. mfs_Init()

使用 MFS 功能之前，需呼叫 mfs_Init() 來初始化 64 MB 快閃記憶體。

2. mfs_Stop()

若程式已呼叫 mfs_Init() 來初始化 64 MB 快閃記憶體，需再呼叫 mfs_Stop() 於關閉時釋放已分配的緩衝區 (buffer)。

從 64 MB 快閃記憶體寫入/讀取檔案之 API

3. mfs_OpenFile()

從 64 MB 快閃記憶體寫入/讀取資料之前，需呼叫 OpenFile() 來開啟檔案。

4. mfs_CloseFile()

完成寫入/讀取資料至 64 MB 快閃記憶體之後，需呼叫 mfs_CloseFile() 以檔案控制代碼 (File Handle) 來關閉檔案。

寫入資料至 64 MB 快閃記憶體之 API

5. mfs_Puts()

此函式可將一個字串複製到指定的檔案中。它會在檔案中字串的結尾加上一個分行符號。

範例：寫入資料至 64 MB 快閃記憶體。

```
#include <stdio.h>
#include "vp2k.h"
#include "MFS.h"

#define _DISK_A 0
#define _DISK_B 1

int main(void)
{
    int iFileHandle, iRet;

    InitLib(); /* 初始化 ViewPAC 函式庫*/
    iRet=mfs_Init();
    if(iRet<=0) return;

    iFileHandle=mfs_OpenFile(_DISK_A,"Test.txt","w");
    if(iFileHandle>0)
    {
        Print("Write string to Test.txt...");
        mfs_Puts(iFileHandle,"test mfs on 64MB flash");
        mfs_CloseFile(iFileHandle);
        Print("done");
    }
    else
        Print("Open file error\n\r");
    mfs_Stop();
    return;
}
```

從 64 MB 快閃記憶體中讀取資料之 API

6. mfs_Gets()

使用 mfs_OpenFile() 開啟檔案後，需呼叫 mfs_Gets() 來從 64 MB 快閃記憶體中讀取資料。

範例 - 從 64 MB 快閃記憶體中讀取資料：

```
#include <stdio.h>
#include "vp2k.h"
#include "MFS.h"

#define _DISK_A 0
#define _DISK_B 1

int main(void)
{
    int iFileHandle, iRet;

    InitLib(); /* 初始化 ViewPAC 函式庫 */
    iRet=mfs_Init();
    if(iRet<=0) return;

    iFileHandle=mfs_OpenFile(_DISK_A,"Test.txt","r");
    if(iFileHandle>0)
    {
        Print("Read from Test.txt...\n\r");

        iRet=mfs_Gets(iFileHandle,Data, 80); /*最大資料長度為 80 byte */
        if(iRet>0) Print("Data=%s\n\r",Data);
    }
}
```

```
mfs_CloseFile(iFileHandle);  
Print("done");  
}  
else  
Print("Open file error\n\r");  
mfs_Stop();  
return;  
}
```

請參閱以下位置，取得更多關於快閃記憶體之詳細資訊：

CD:\NAPDOS\vp-2000\demo\vp-2111\Basic\64MB_Flash\

http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/64mb_flash/

附錄 A. 什麼是 MiniOS7?

MiniOS7 是泓格科技 (ICP DAS) 所設計之內嵌式 ROM-DOS 作業系統。功能上與其它 DOS 版本相同，並可運行符合標準 DOS 的可執行檔。

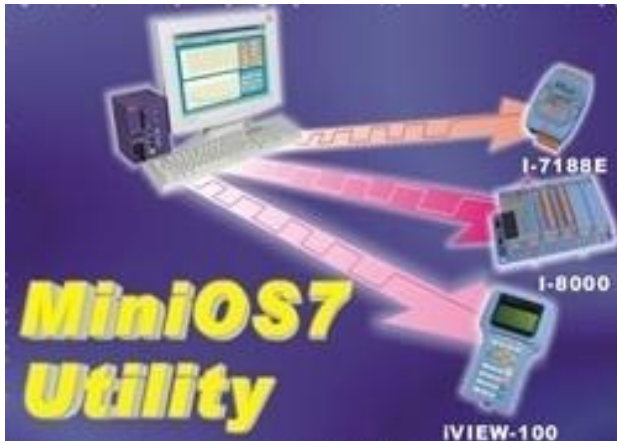


DOS (無論 PC-DOS、MS-DOS 或 ROMDOS) 是用來告訴電腦如何處理資訊之指令集或代碼。DOS 執行程式、管理檔案、控制信息處理、指定輸入與輸出，並執行許多其他相關功能。

MiniOS7 與 ROM-DOS 之特色比較表：

特色	MiniOS7	ROM-DOS
啟動時間	0.1 sec	4 ~ 5 sec
更輕巧的大小	< 64 K bytes	64 K bytes
支援 I/O 擴充匯流排	是	否
支援 ASIC 碼	是	否
Flash ROM 記憶體管理	是	否
OS 更新 (上傳)	是	否
內建硬體診斷功能	是	否
直接以指令控制 I-7000 系列模組	是	否
客製化 (ODM) 功能	是	否
免費	是	否

附錄 B. 什麼是 MiniOS7 Utility?



MiniOS7 Utility 是一種軟體工具，可輕鬆與迅速地進行設定，並將檔案上傳至所有內嵌有泓格科技 (ICP DAS) 之 MiniOS7 的產品。

注意：版本 3.1.1 起，MiniOS7 Utility 可允許使用者透過乙太網路來進行遠端存取控制器 (7188E、8000E..等)。

功能

- 支援的連線方式
 1. COM Port 連線 (RS-232)
 2. 乙太網路連線 (TCP & UDP)
(支援版本： 3.1.1 起)
- 設定
 1. 日期與時間
 2. IP 位址
 3. COM Port
 4. 磁碟大小 (Disk A · Disk B)
- 檔案維護
 1. 上傳檔案
 2. 刪除檔案
 3. 更新 MiniOS7 image
- 檢查產品資訊
 1. CPU 類型
 2. Flash 容量
 3. SRAM 容量
 4. COM Port 數.....等等。

包括經常使用的工具

- a. 7188XW
 - b. 7188EU
 - c. 7188E
-
- d. SendTCP
 - e. Send232
 - f. VxComm Utility

PC 系統需求

1. IBM 相容之 PC
2. Windows 95 /98/NT/2000/XP

支援的產品

1. 7188XA/XB/XC
2. 7188EX 系列
3. All i-8000 系列
4. iView100
5. uPAC-7186EX
6. ET-6000 系列
7. ET-7000 系列

下載位置：

http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/minios7/utility/minios7_utility/

附錄 C. 什麼是 MiniOS7 檔案系統 (MFS)? (適用型號: VP-2111)

MiniOS7 檔案系統 (MFS) · 為機械化儲存系統提供了穩固的替代選擇。專用於 NAND 快閃記憶體，MFS 實現了在 MiniOS7 上使用 C 語言 API，供內嵌式資料記錄器應用之可靠地檔案系統



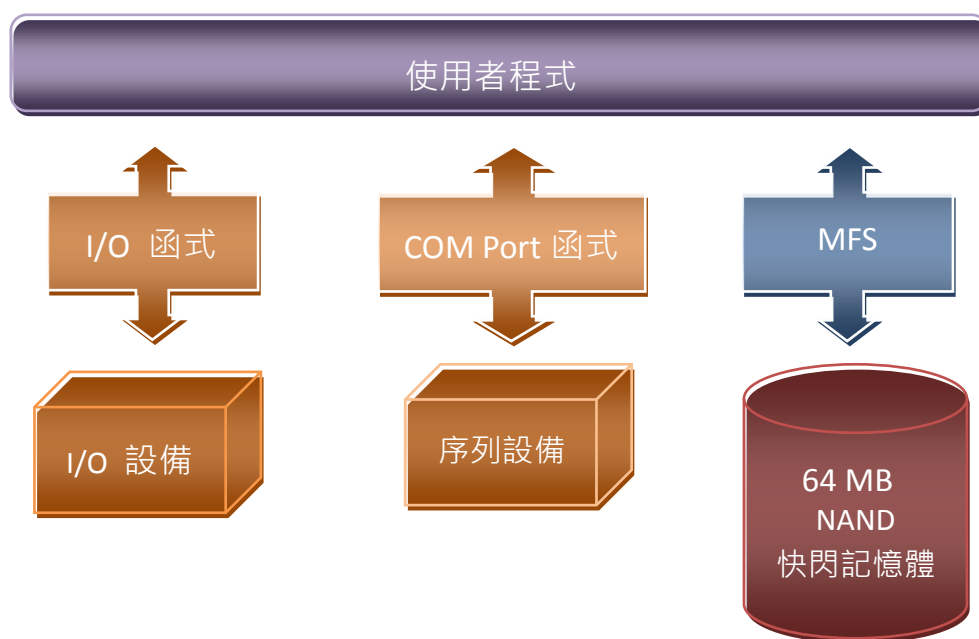
您可使用 MiniOS7 檔案系統 (MFS) 所提供的函式庫，動態地讀取/寫入快閃記憶體 (64 MB) 中的檔案，並可實現各式各樣的資料記錄應用。例如，使用電子時戳 (timestamp) 記錄類比訊號值、記錄 RS-232/485 通訊資料以供分析使用。

▶ 支援產品

- uPAC-7186ED-FD
- iP-8441-FD
- IP-8841-FD
- VP-2111

▶ 應用

- 使用電子時戳 (timestamp) 記錄資料。
- 記錄資料並透過網路 (Ethernet) 傳送資料。



► MFS 規格

項目	說明
磁碟大小	快閃記憶體之一半的容量。
檔案數	每磁碟最多 456 個檔案。
檔案大小	每個檔案最大可為磁碟容量大小。
檔案名稱	最大 12 bytes (case sensitive)
檔案操作模式	<ol style="list-style-type: none"> 1. 唯讀。 2. 唯寫: 建立新檔來寫入資料或覆寫檔案 (若該檔案已存在)。 3. 添加: 添加資料至檔案。
檔案控制代碼 (File Handle)	<p>每個磁碟最多 10 個。</p> <p>讀取模式: 於每個磁碟中執行讀取動作, 可使用 10 個檔案控制代碼 (File Handle)。讀取模式下, 總共可開啟 20 個檔案。</p> <p>寫入與添加模式: 於所有磁碟中執行寫入動作, 僅可使用 1 個 檔案控制代碼 (File Handle)。</p>
寫入驗證	<p>是。預設為啟用。</p> <p>可呼叫 <code>mfs_EnableWriteVerification</code> 與 <code>mfs_DisableWriteVerification</code> 函式來變更設定。</p>
自動檔案系統回復	<p>是。</p> <p>當非預期的系統重新啟動或是電源意外斷電發生時, 關閉中的檔案或是開啟在唯讀模式的檔案不會受到損壞, 只有最後一次執行寫入函式 (<code>mfs_WriteFile</code>) 之後的採集的資料會消失。當系統重新啟動時, MFS 會參考 NVRAM 中紀錄的檔案資訊回復系統。</p>
寫入速度	<p><code>mfs_WriteFile</code>:</p> <p>147.5 KB/Sec (啟用驗證) (預設)</p> <p>244.0 KB/Sec (關閉驗證)</p> <p><code>mfs_Puts</code>:</p> <p>142.1 KB/Sec (啟用驗證) (預設)</p> <p>229.5 KB/Sec (關閉驗證)</p>
讀取速度	<p><code>mfs_ReadFile</code>: 734.7 KB/Sec</p> <p><code>mfs_Gets</code>: 414.2 KB/Sec</p>
最大寫入資料長度	32767 byte。
最大讀取資料長度	32767 byte。

► 相關資源下載

- MFS SDK :

VP-2111:

<http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/basic/lib/>

- MFS 範例 :

VP-2111:

http://ftp.lcpdas.com/pub/cd/8000cd/napdos/vp-2000/demo/vp-2111/64mb_flash/

附錄 D. I-8K 與 I-87K 系列模組

(適用型號: VP-2111)

VP-2111 擁有 3 個可用來擴充本地 I/O 的插槽。I/O 模組可分為並列型 (Parallel) I-8k 系列 與 串列型 (Serial) I-87k 系列模組 (需使用高卡版本)。

此兩種系列的差異如下表：

項目	I-8K 系列	I-87K 系列
微處理器	無	有 (8051)
通訊介面	並列匯流排	串列匯流排
通訊速度	快	慢
DI 兼具 Latch 功能	無	有
DI 兼具計數器功能	無	有
開機值	無	有
安全值	無	有
可調整之 AO 輸出斜率控制	無	有

附錄 E. RS-485 網路之應用

於單一的雙絞線配置環境下，RS-485 最長可達 4000 英尺或 1.2 公里，若 RS-485 網路超過 4000 英尺或 1.2 公里，則需加裝 RS-485 中繼器 (Repeater) 來擴展 RS-485 網路。

E.1. 基本的 RS-485 網路

RS-485 網路的基本元件包含了一個主控制器 (或使用一台電腦為主機控制器) 與一些 RS-485 設備。

小技巧與安全警告



若您的電腦 (或筆電) 並沒有 COM Port，可使用 I-7561 (USB 至 RS-485 轉換器) 來連接 ViewPAC 與電腦 (或筆電)。

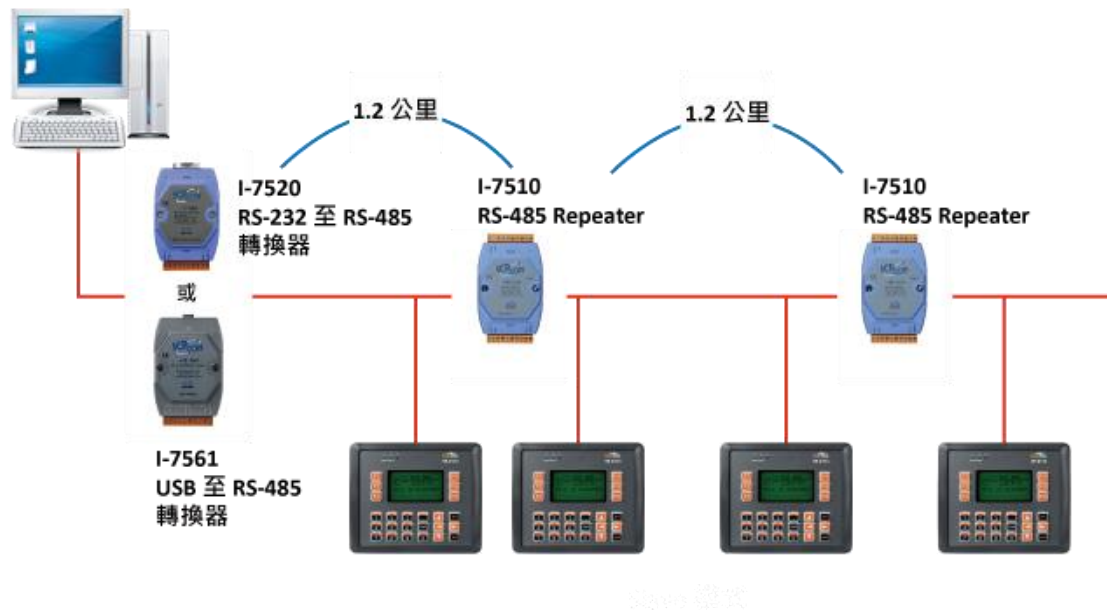
使用 I-7561 轉換器之前，必須先安裝 USB 驅動程式。您可至下列網址取得 USB 驅動程式：

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/7000/756x/>



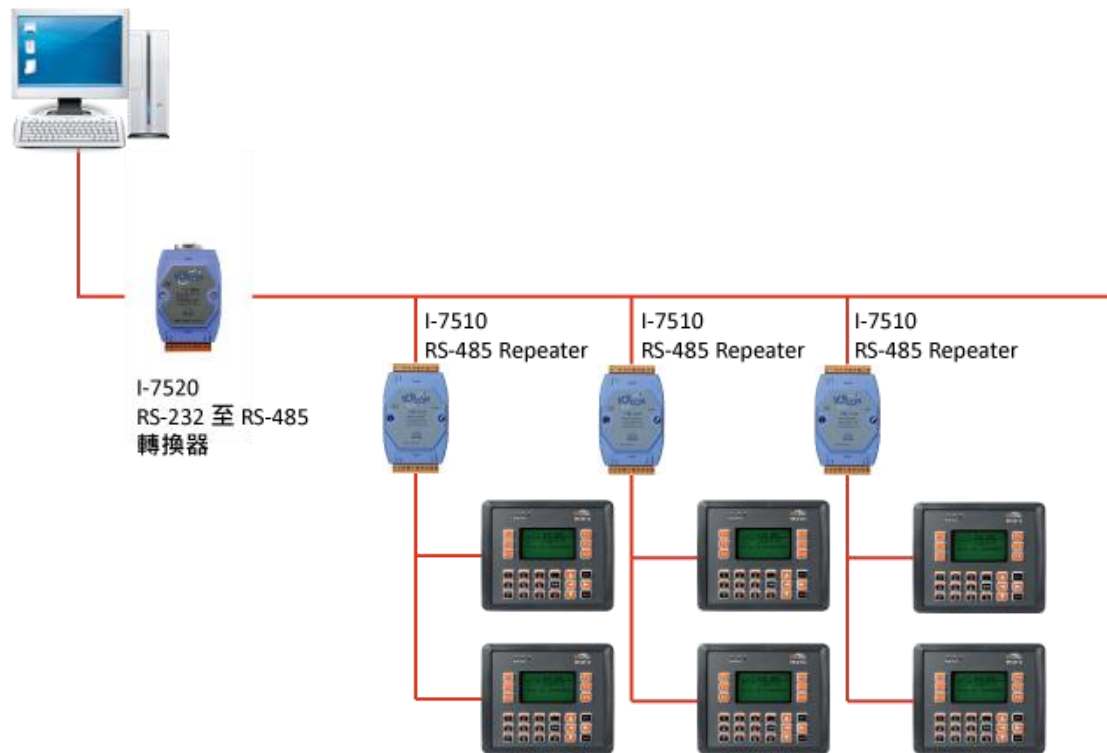
E.2. 鏈狀 (Daisy Chain) RS-485 網路

所有的 RS-485 設備皆直接連到主網路，若該網路連線長達 1.2 公里，則需要一個中繼器 (Repeater，I-7510 系列產品) 來擴展網路長度。

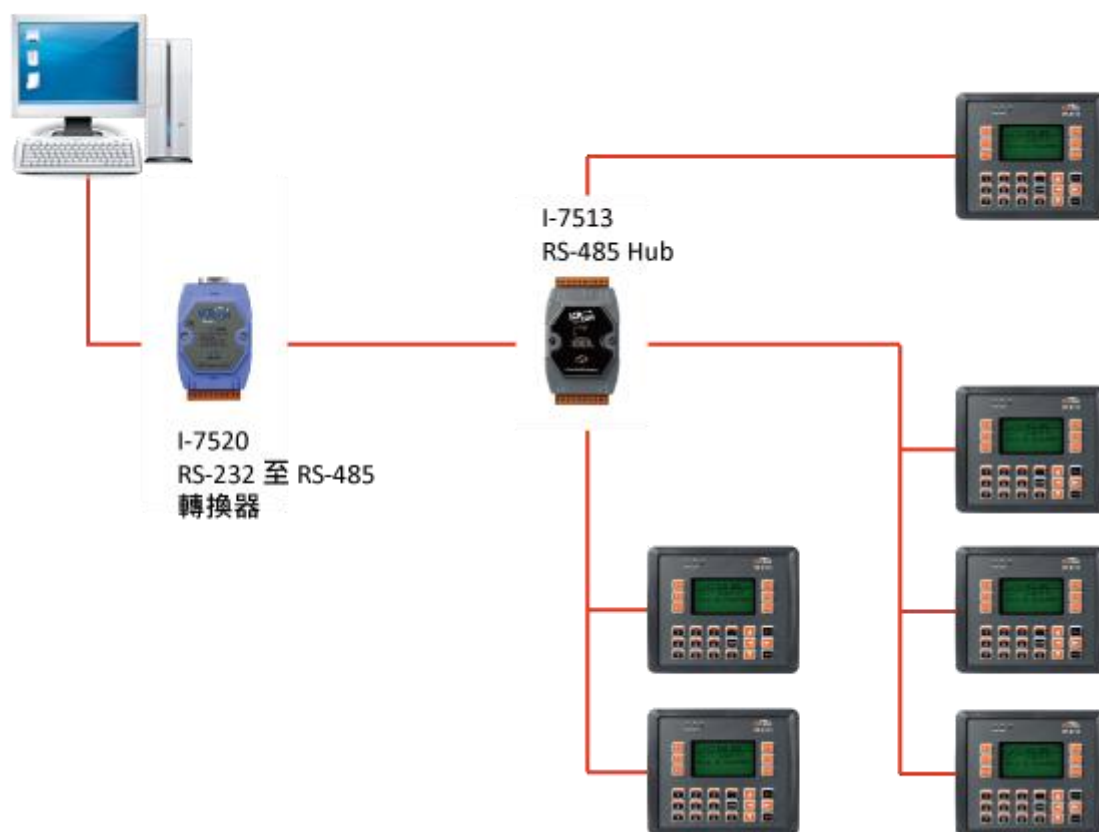


E.3. 星狀 RS-485 網路

沿著主網路有多個分支設備。於此例中，最好可加裝中繼器 (Repeater) 來隔離或過濾設備所產生的雜訊。

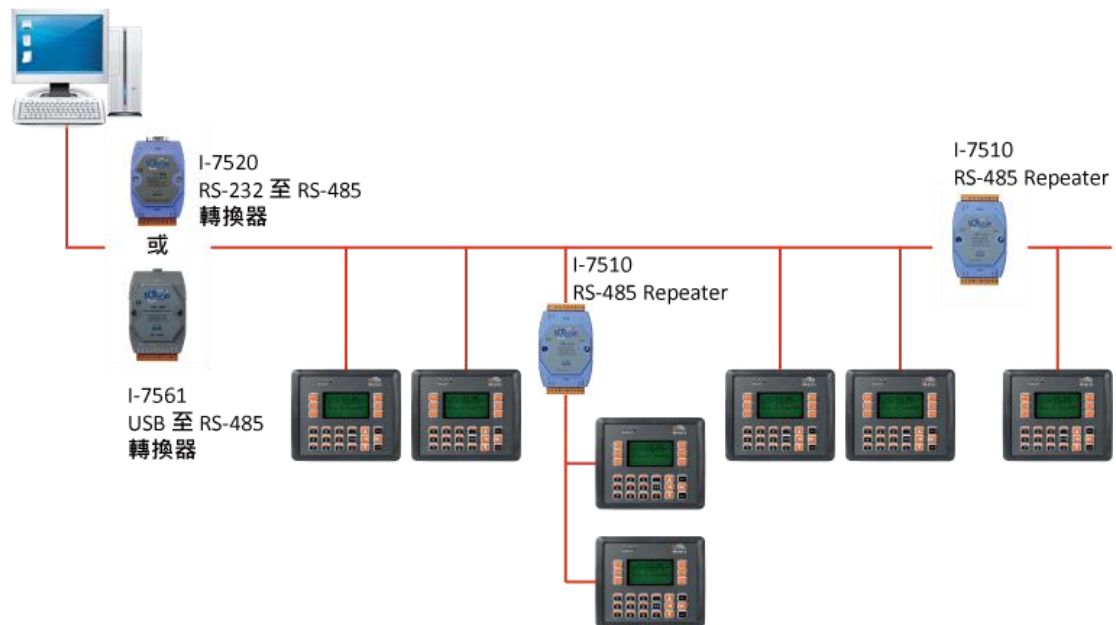


於星狀網路中，您可選用 I-7513 做為 RS-485 集線器 (Hub)。



E.4. 隨機 RS-485 網路

如下圖，沿著主要接線上有許多分支設備。於此種狀況，最好使用中繼器 (Repeater) 來隔離並過濾設備所產生的雜訊。



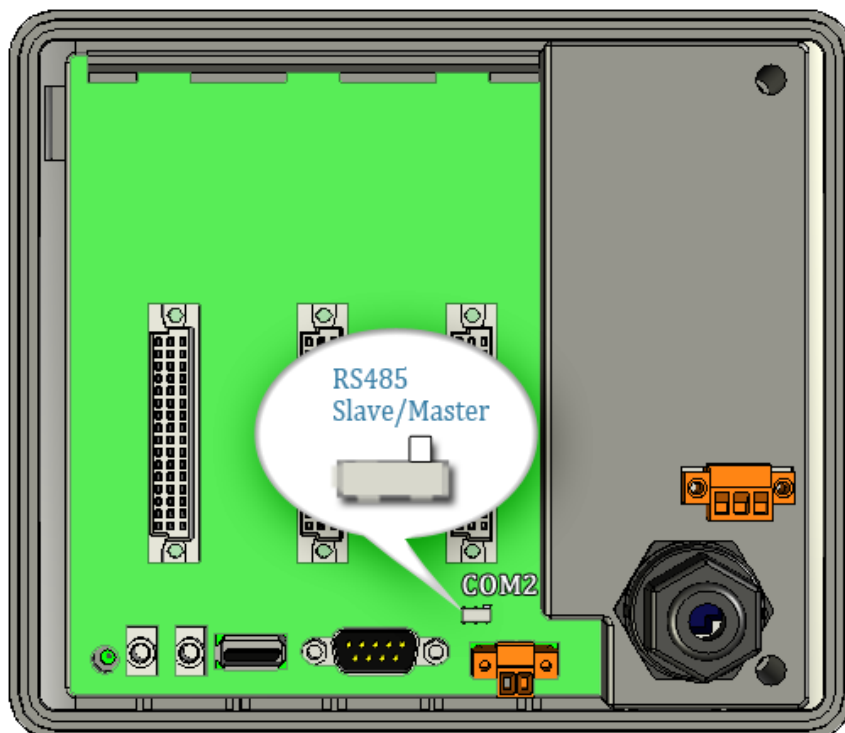
E.5. Master/Slaves 設定

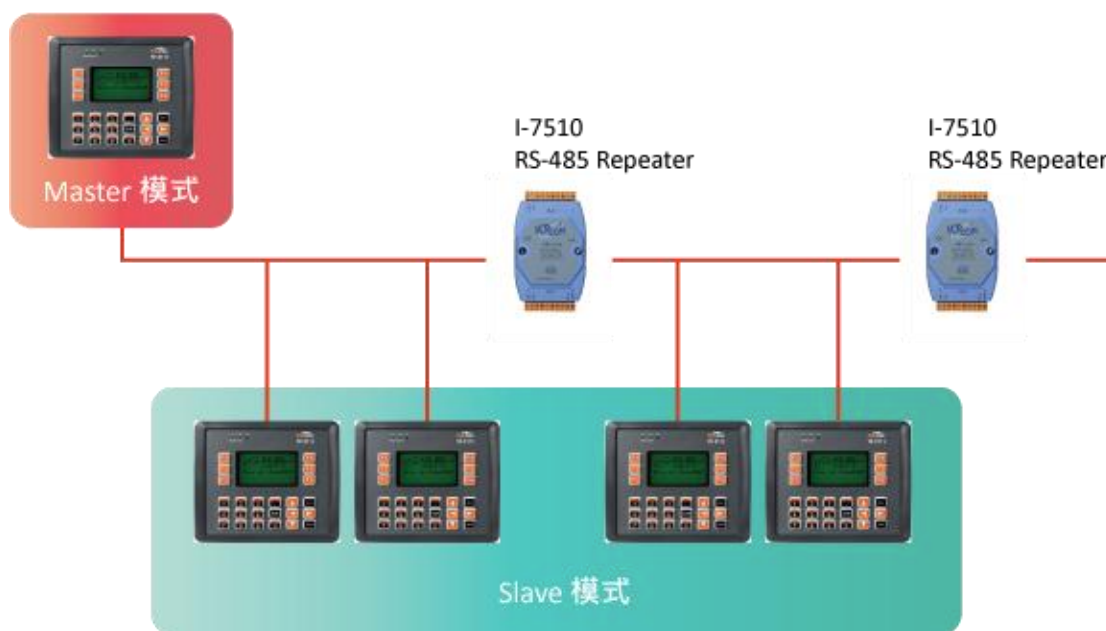
同一個網路中並須有一台為 Master 設備並配有一個 Jumper。在 Master/Slave 應用中，ViewPAC 的預設值即為 “Master”。

E.5.1. ViewPAC 作為 Master 設備 (預設)

同一個網路中，若有一台 ViewPAC 設定為 Master 模式，則其它設備必須設定為 Slave 模式。

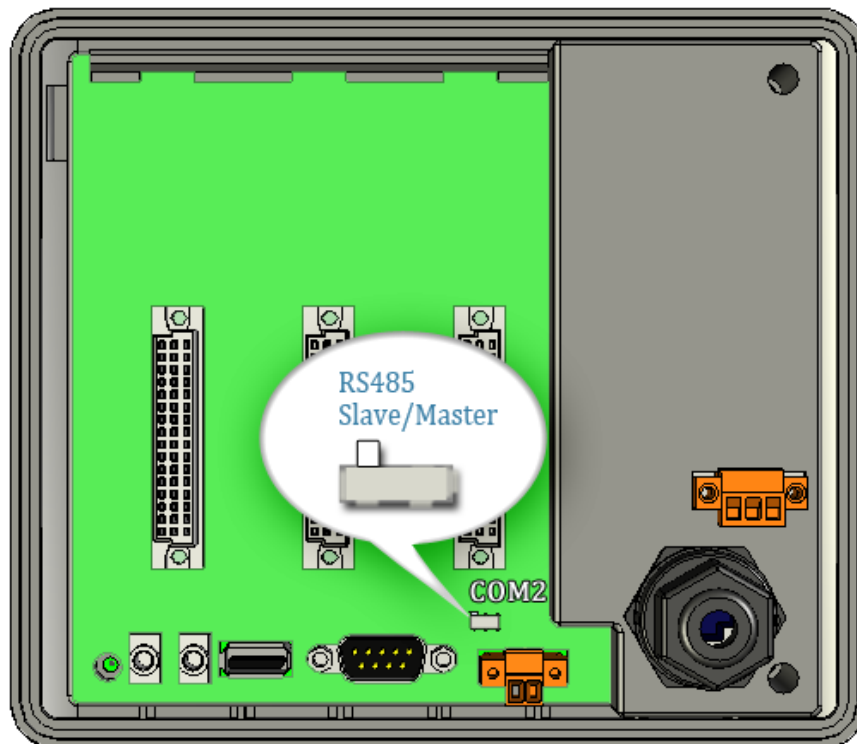
您只需調整電路板上的 Jumper，即可設定 ViewPAC 為 Master 模式。請參考下圖：





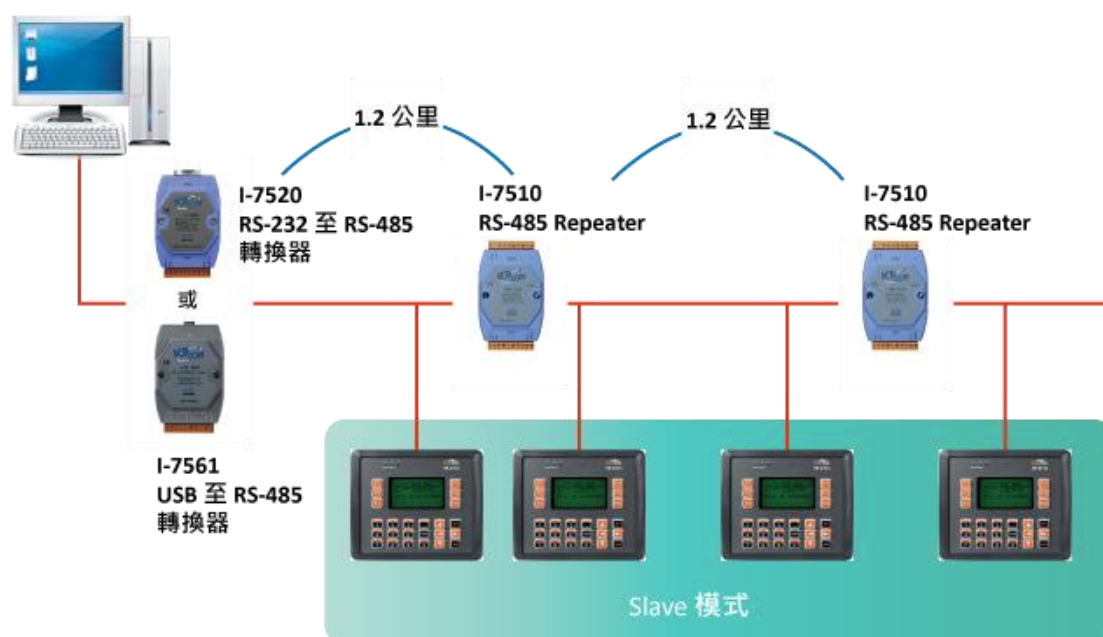
E.5.2. ViewPAC 作為 Slave 設備

您可藉由調整 ViewPAC 電路板上的 Jumper，將其設定為 Slave 模式。請參考下圖：





於 RS-485 網路中，不需加裝 Repeater 的最大距離為 1,200 米 (4,000 英尺)。如下圖所示，您可藉由每 1,200 米加裝一個 RS-485 Repeater 來延展通訊距離。



附錄 F. 版本修訂紀錄

下表說明了手冊版本的修訂紀錄：

版本	日期	說明
1.0.1	2009 年 6 月	創刊版。
1.0.2	2010 年 7 月	新增 VH-2110/VH-2111P/VH-2211/VH-2311 安裝與設定資訊。
1.0.3	2011 年 7 月	<ol style="list-style-type: none">1. 刪除 VH-2111P/VH-2211/VH-2311安裝與設定資訊。2. 修改章節 3.3，關於開發環境設定的資訊。3. 修改章節 4.4，關於快閃記憶體的函式。4. 修改附錄 E，關於 RS-485 網路的應用。