



G-4510 系列

使用者手冊

可程式自動化控制器(具備太陽能充電功能)

Version 1.0 August 2019

本手冊的技術服務與操作資訊適用於

G-4510 系列



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

2018 © 泓格科技股份有限公司版權所有。

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。

E-mail 信箱：service@icpdas.com

目錄

目錄.....	III
1. 簡介	1
2. 硬體規格	2
2.1 G-4510 系列	2
2.2 G-4510 系列規格	3
3. 應用架構	5
3.1 水文/風場監測應用	5
3.2 車輛監測/追蹤系統	5
3.3 備援通訊系統	6
4. 硬體	7
4.1 腳位配置	7
4.2 尺寸	8
4.3 作業模式切換	9
4.4 LED 指示燈	10
4.5 I/O 接線圖	11
4.6 從睡眠模式中喚醒	13
4.7 啟動 G-4510 系列	14
5. 省電模式與充電保護	15
5.1 省電模式	15
5.2 如何選擇電池	17
5.3 如何選擇太陽能板	20
6. API 與範例程式參考	23
6.1 用於 I/O 的 API	23
6.1.1 X305IO_Init	24
6.1.2 X305IO_GetLibVersion	25
6.1.3 X305IO_Read_AD_CalibrationGain	26
6.1.4 X305IO_Read_AD_CalibrationOffset	27
6.1.5 X305IO_AnalogIn	28
6.1.6 X305IO_Read_All_DI	29

6.1.7 X305IO_Read_One_DI	30
6.1.8 X305IO_Write_All_DO	31
6.1.9 X305IO_Write_One_DO.....	32
6.1.10 X305IO_Read_All_DO	33
6.1.11 X305IO_Read_One_DO.....	34
6.1.12 X305IO_AnalogIn_SetChannel	35
6.1.13 X305IO_AnalogIn_Hex	36
6.1.14 X305IO_AnalogIn_HexToFloat.....	37
6.2 用於 MMC/SD 的 API.....	38
6.3 用於 LCD 的 API	43
6.3.1 LCD_Init	44
6.3.2 LCD_BackLight_On	45
6.3.3 LCD_BackLight_Off	46
6.3.4 LCD_ShowText	47
6.3.5 LCD_ClrScrn	48
6.3.6 LCD_StandByMode.....	49
6.3.7 LCD_NormalMode	50
6.3.8 LCD_GotoPosition	51
6.3.9 LCD_CursorDisplay	52
6.3.10 LCD_LineReverse.....	53
6.3.11 LCD_LineRestore.....	54
6.3.12 LCD_GetLibDate.....	55
6.3.13 LCD_GetLibVersion	56
7. 程式編譯與下載步驟	57
8. 版本記錄	63

1. 簡介

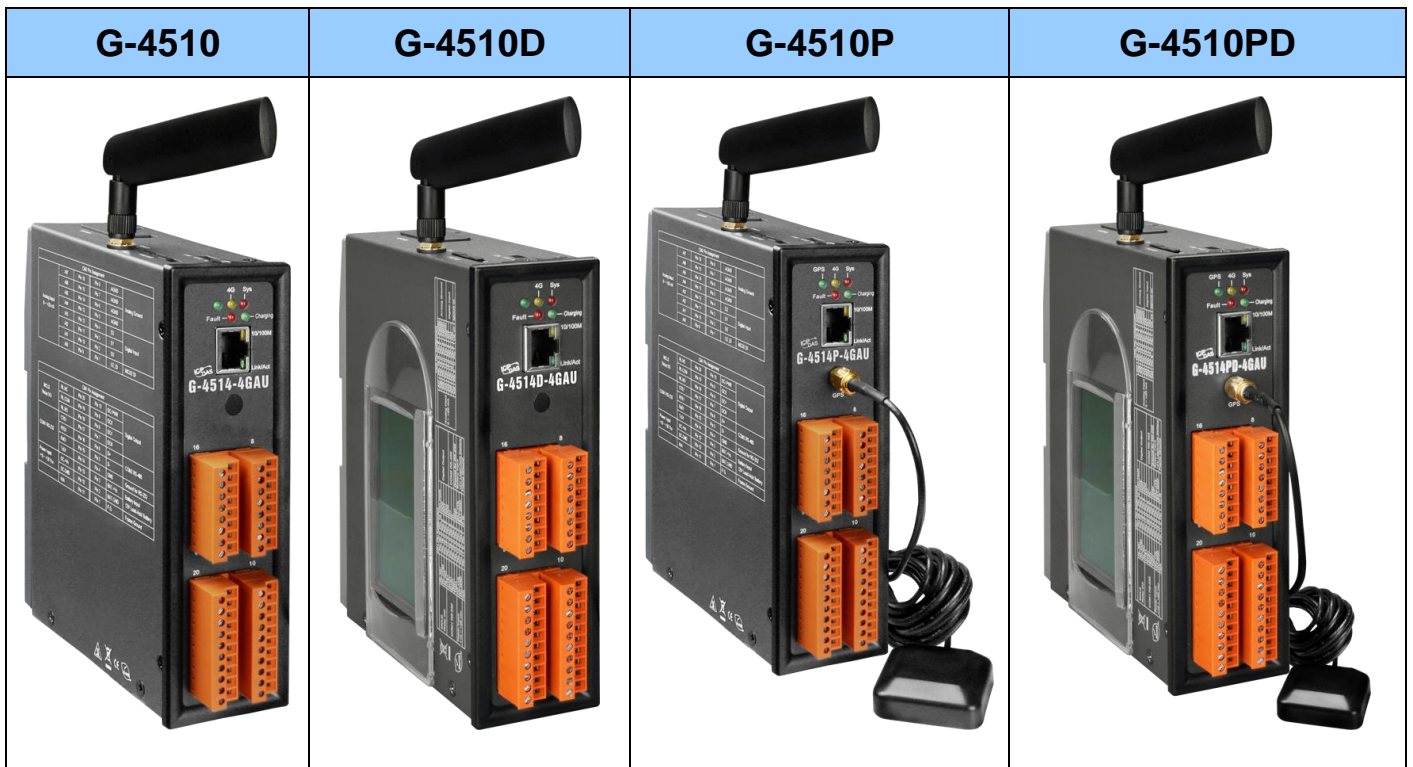
G-4510 系列為 M2M 省電型 PAC 產品，其內建太陽能/鉛酸電池之充電器。G-4510 極適合用於水文監控、土石流監控領域，或是不便取得電源，需要使用太陽能的應用。若配合其 GPS 功能(選購)，更可以應用於車載，河川船泊監控之領域。

G-4510 系列擁有太陽能充電器、GPS 模組(選購)、乙太網路、3 DI、3 DO、8 AI、1 Relay、1 RS-232 和 1 RS-485。可用於各種需要 4G、3G、GPRS、SMS、乙太網路或串列埠的領域。其內建的 MiniOS7 提供與 I-7188/I-7186 系列相同的開發環境，對於 I-7188/I-7186 的用戶來說，G-4510 系列屬於熟悉的開發環境，較容易上手。



2. 硬體規格

2.1 G-4510 系列



2.2 G-4510 系列規格

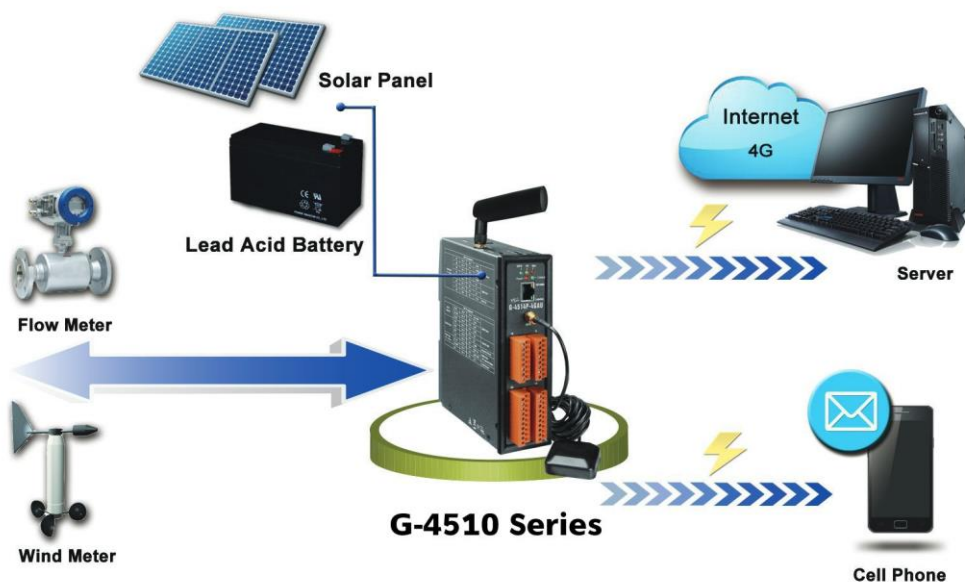
產品規格

項目	G-4510 Series	G-4510D Series	G-4510P Series	G-4510PD Series	
CPU	內建 80 MHz 微處理器				
SRAM/Flash	512/512 KB, 及時時鐘(RTC), 看門狗(Watchdog timer)				
NVRAM	31 bytes, 電池備援, 資料有效期達 10 年				
EEPROM	16 KB, 資料保存> 40 年. 1,000,000 次清除/寫入				
通訊介面					
COM ports	COM1:5-wire RS-232; COM2: RS-485				
乙太網路(Ethernet)	10/100 Base-TX 乙太網路控制器				
Digital Input					
通道數量	3				
輸入型態	Source(Dry Type), Common Ground				
截止電位	+1 V max.				
道通電位	+3.5 ~ +30 V				
隔離保護電壓	Non-isolated				
Digital Output					
通道數量	3				
輸出型態	3 Open Collector (Sink/NPN)				
負載電壓	+30 VDC max.				
負載電流	100 mA max.				
隔離保護電壓	Non-isolated				
Analog Input					
通道數量	8				
解析度	12 - bit				
輸入範圍/型態	0 ~ 20 mA				
取樣頻率	1 KHz max. (只讀取一個通道時)				
精度	+/- 2 LSB (+/- 0.0097 mA)				
隔離保護電壓	2500Vrms 3000Dc to DC				
Relay					
通道數量	1				
輸出型態	Form C				
輸入範圍	2A@30 Vdc ; 0.25 A @250 Vac				
機械操作壽命	typ. 10 ⁸ 次操作				
GPS 介面					
支援通道數量	-		32		
一般資訊	-		Tracking = up to -159 dBm (with external LNA) Cold start = up to -146 dBm (with external LNA)		
抓取時間	-		Hot start (Open Sky) = 2 s(typical) Cold start (Open Sky) = 36 s(typical)		
支援協定	-		NMEA 0183 version 3.01		
LCD 介面					
一般資訊	顯示尺寸	-	80.61 mm x 14.37 mm (W x H)	-	80.61 mm x 14.37 mm (W x H)
	模組尺寸	-	93 mm x 70 mm x 1.6 mm (W x H x T)	-	93 mm x 70 mm x 1.6 mm (W x H x T)

壽命	-	正常操作下，可超過 100,000 小時	-	正常操作下，可超過 100,000 小時
電源 (太陽能輸入端)				
保護	Power reverse polarity protection			
外殼接地保護	ESD, Surge, EFT, Hi-Pot			
電源輸入範圍	+10 V _{DC} ~ +30 V _{DC} , (Max. Voltage of Solar Panel must less +30V)			
耗電量	Deep Sleep: < 10 mA@12V _{DC} ; Deep Sleep(With LCD): < 11 mA@12V _{DC} ; Sleep: < 15 mA@12V _{DC} ; Idle: 90 mA @ 24 V _{DC} ; Data Link: 150 ~ 400 mA (peak) @ 24 V _{DC}			
電池需求規格				
電池	12V 鉛酸電池			
充電電壓條件	電源端的電壓需大於+16V			
低電壓保護	低電壓斷電電壓 = 11.1V / 低電壓恢復供電電壓 = 12.6V			
LED 指示燈				
系統指示燈	紅			
4G	黃			
GPS	-		綠	
充電指示 / 充電失敗指示	綠 / 紅			
機構				
外殼	金屬			
尺寸	47 mm x 142 mm x 168 mm (W x L x H)			
安裝	DIN-Rail and Wall mount			
環境				
操作溫度	-20 ~ +70 °C	-15 ~ +55 °C	-20 ~ +70 °C	-15 ~ +55 °C
存放溫度	-40 ~ +80 °C	-20 ~ +70 °C	-40 ~ +80 °C	-20 ~ +70 °C
濕度	5~90% RH, non-condensing			

3. 應用架構

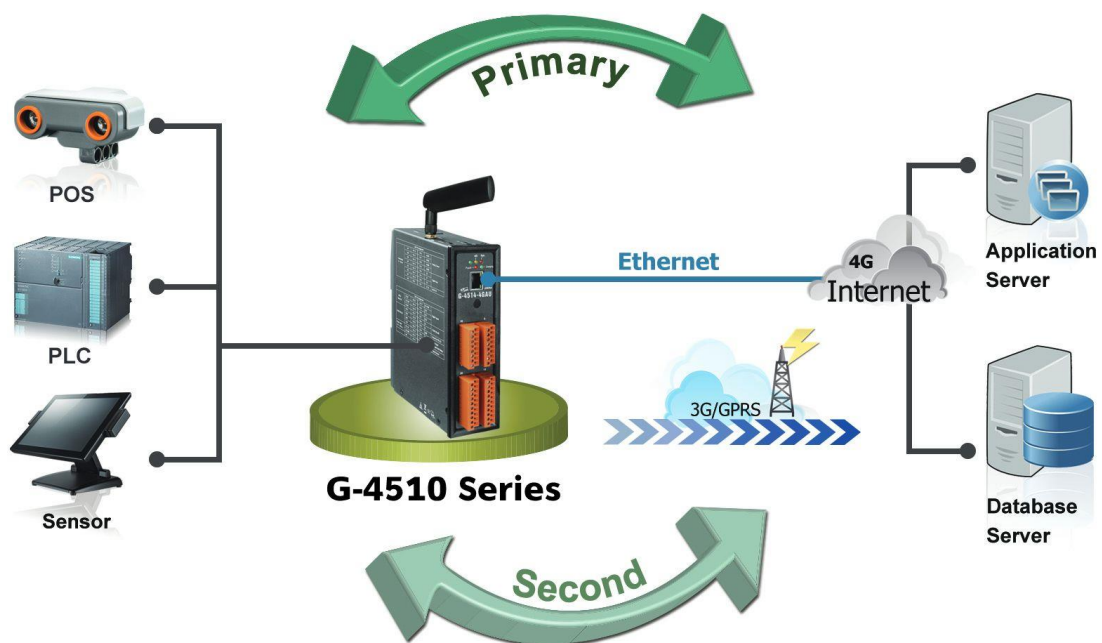
3.1 水文/風場監測應用



3.2 車輛監測/追蹤系統

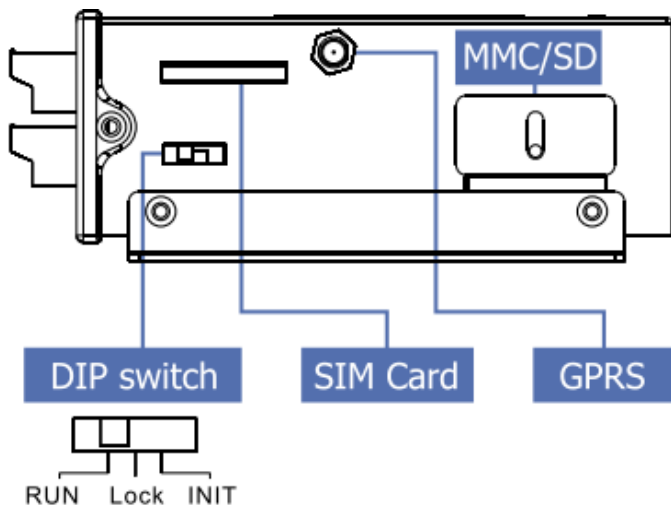
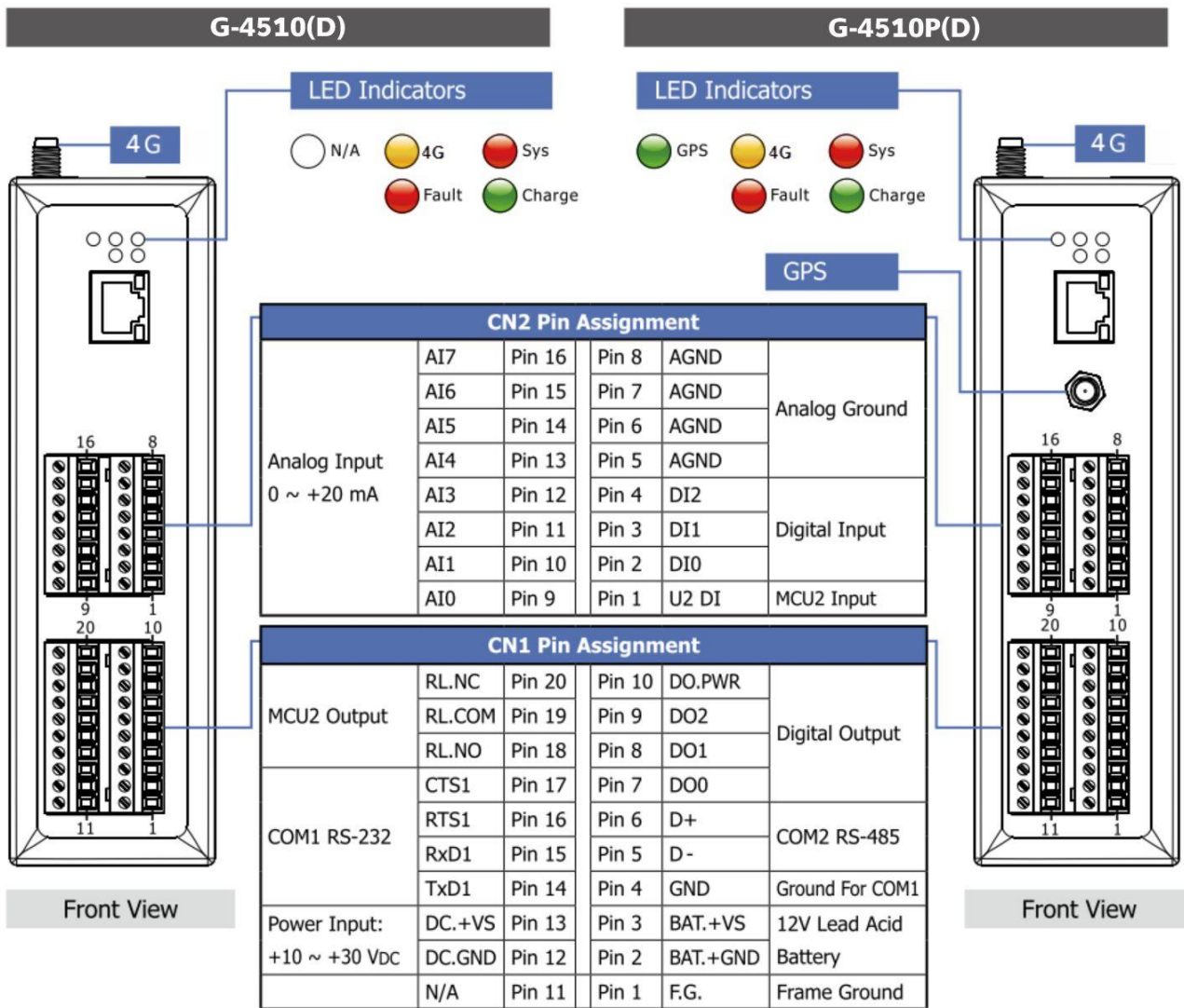


3.3 備援通訊系統

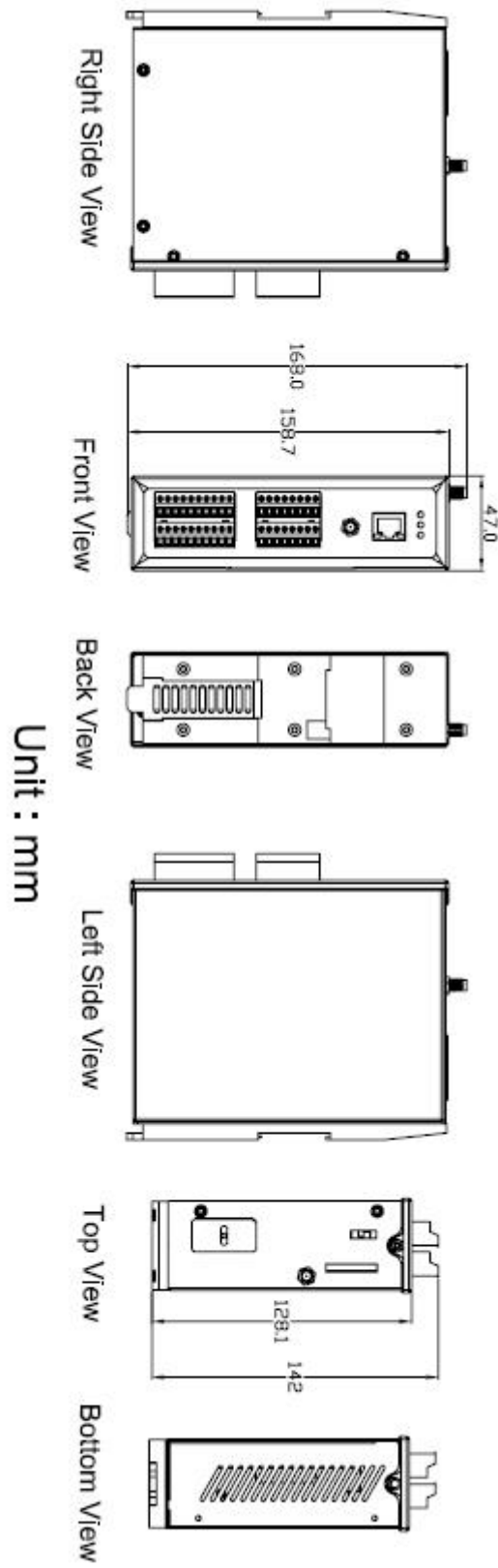


4. 硬體

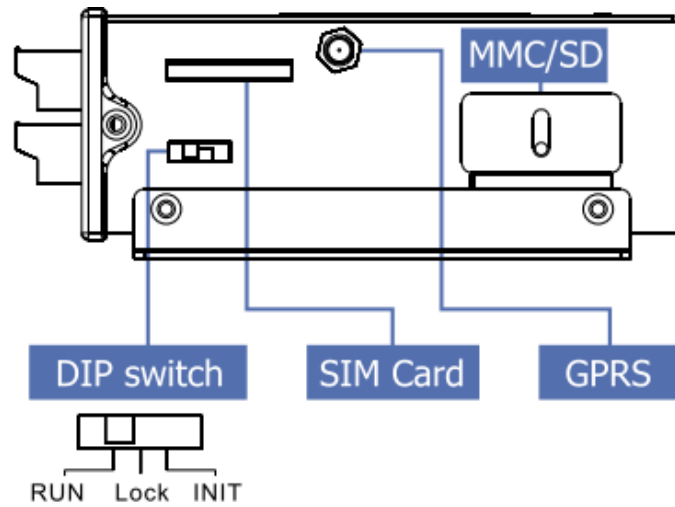
4.1 腳位配置



4.2 尺寸

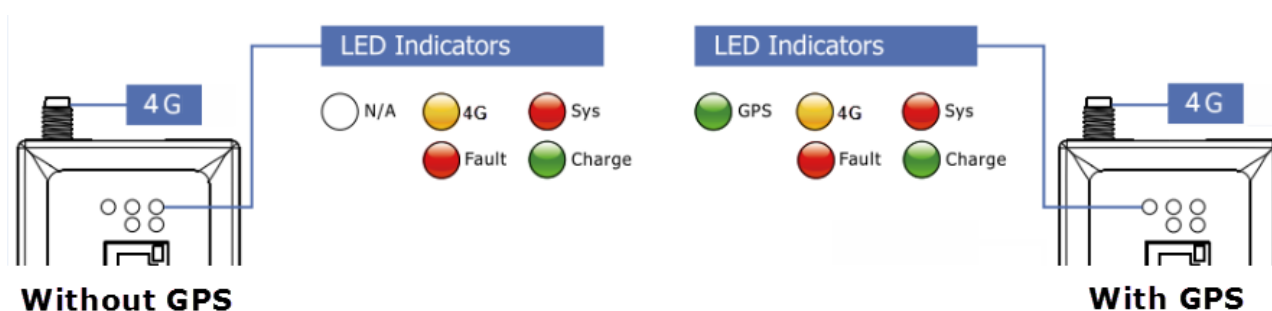


4.3 作業模式切換



作業模式切換	
RUN	OS 可執行 autoexec.bat
	Flash 可被讀/寫
LRun	OS 可執行 autoexec.bat
	Flash 只能被讀取(Lock)
INIT	OS 不可執行 autoexec.bat
	Flash 可被讀/寫

4.4 LED 指示燈



G-4510 有 5 顆 LED 指示燈，幫助使用者判斷系統目前的狀態，其說明如下：

- A. Sys (紅色)：System LED 是可編程的
- B. 4G (黃色)：可判斷 4G 模組是否正常(模組註冊後才會開始閃爍)

4G 模組正常	4G 模組異常	資料傳輸中
亮 2 秒暗 1 秒	不亮或亮 1 秒暗 2 秒	每 0.2 秒閃爍 1 次

- C. GPS (綠色)(選購)：可判斷 GPS 模組是否正常

GPS 異常	GPS 定位中	GPS 正常
不亮	恆亮	1 秒閃爍 1 次

- D. Charging (綠色)：顯示充電狀態

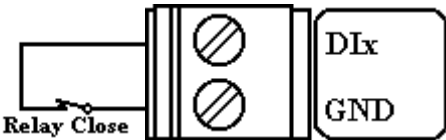
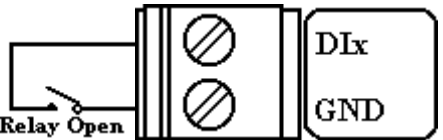
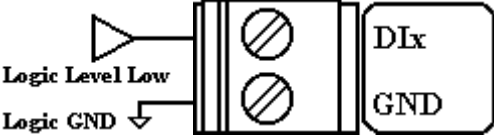
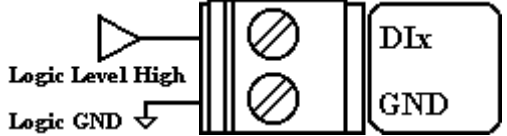
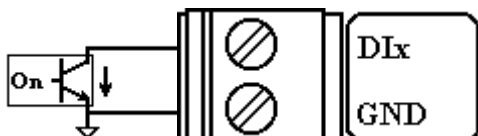
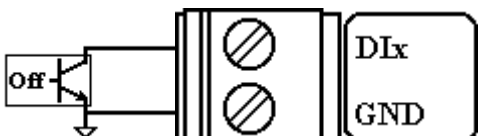
充電中	充電停止
恆亮	不亮

- E. Fault (紅色)：充電錯誤指示燈

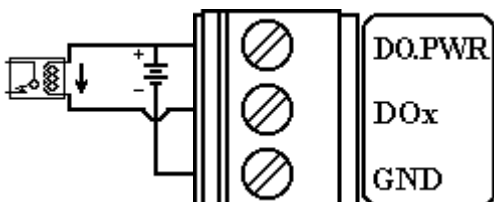
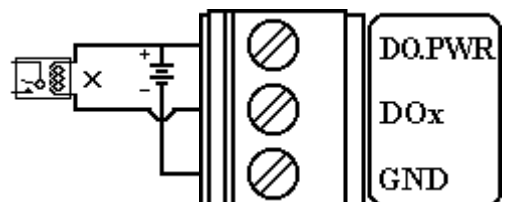
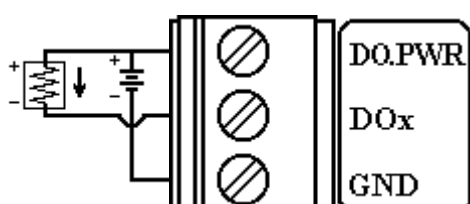
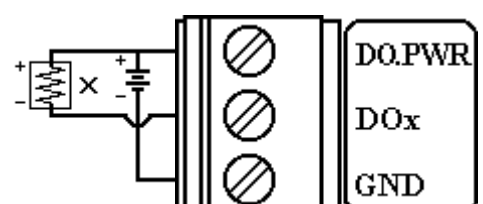
正常	錯誤
不亮	恆亮

4.5 I/O 接線圖

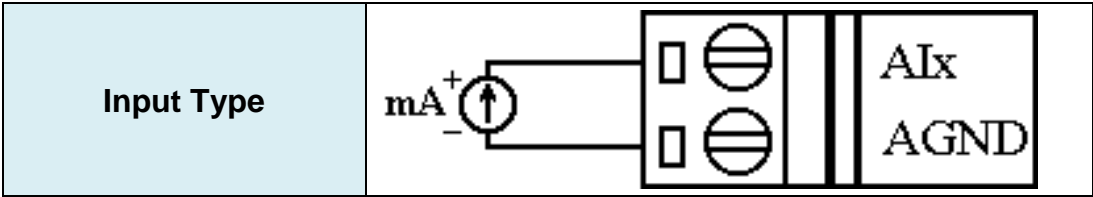
➤ DI 接線說明

Input Type	ON State DI value as 0	OFF State DI value as 1
Relay Contact		
TTL/CMOS Logic		
Open Collector		

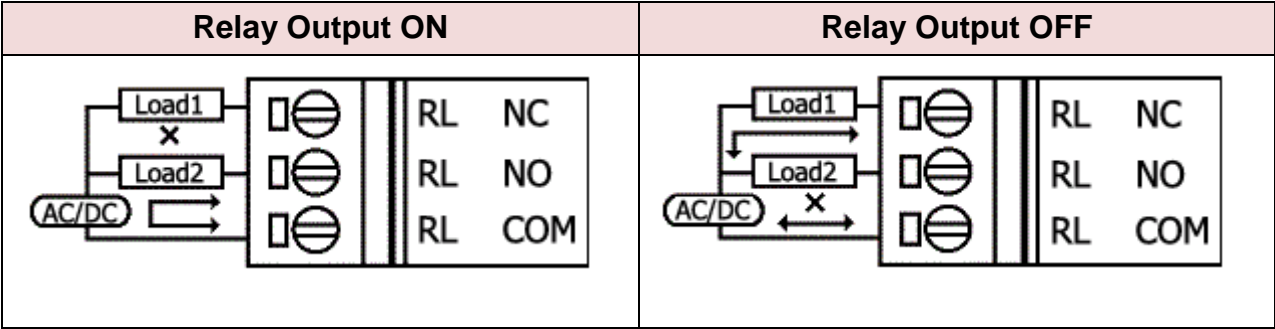
➤ DO 接線說明

Input Type	ON State DO value as 1	OFF State DO value as 0
Drive Relay		
Resistance Load		

➤ AI 接線說明

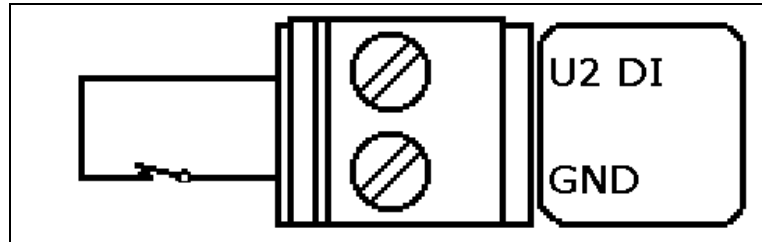


➤ Relay 接線說明



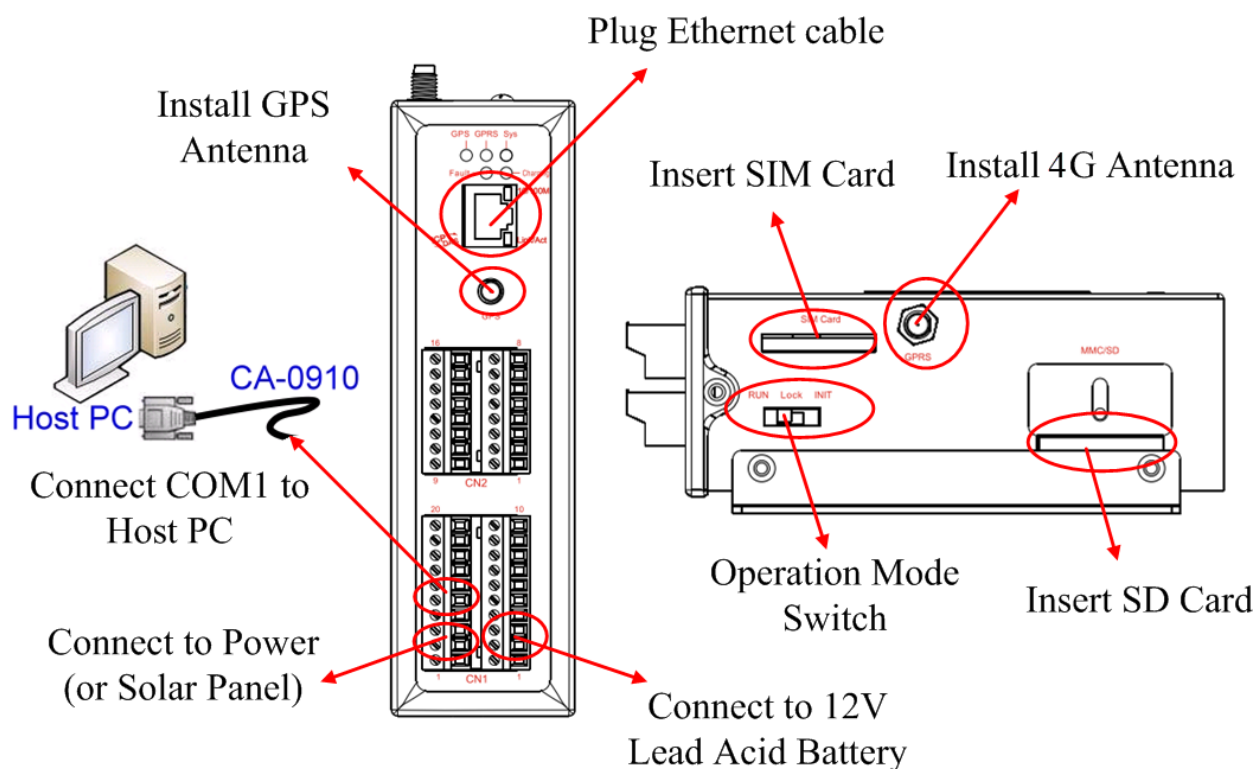
4.6 從睡眠模式中喚醒

1. 當 G-4510 處於睡眠模式時，將“U2 DI” (MCU2 DI)接至 GND，藉此將 G-4510 從睡眠中喚醒
2. 當“U2 DI” 被連接至 GND 時，讀取“U2 DI”會是 0



4.7 啟動 G-4510 系列

1. 安裝 4G 天線
2. 插入已確認正常的 SIM 卡(可先用手機測試)
3. 連接 DC.+VS 和 DC.GND 到電源供應器或太陽能板
4. 連接 BAT.+VS 和 BAT.GND 到 12V 鉛酸電池



警告! 表面發燙，請勿碰觸



產品外殼可能處於高溫狀態，在外殼冷卻以前，請勿觸摸，否則可能會被燙傷。

安全指令說明



此裝置的電源輸入腳位(DC.+VS/DC.GND)與直流電源(SELV，有限電源)連接時，應該要符合 EN60950-1 的要求。請務必確認接線是否正確。

5. 省電模式與充電保護

5.1 省電模式

▶ 睡眠模式

此模式會關閉 7186 CPU，所有 I/O(3DI, 3DO, 8AI, 除了 MCU2 I/O)和 GPS，但 4G 模組會持續運作

- 功耗：14~15 mA@12V
- 如何喚醒 G-4510：
 - (1) 睡眠時間到
 - (2) 觸發 U2_DI (連接 U2_DI 至 GND.; 讀取 U2_DI 的值為 0)
 - (3) 撥電話給 G-4510

▶ 深層睡眠模式

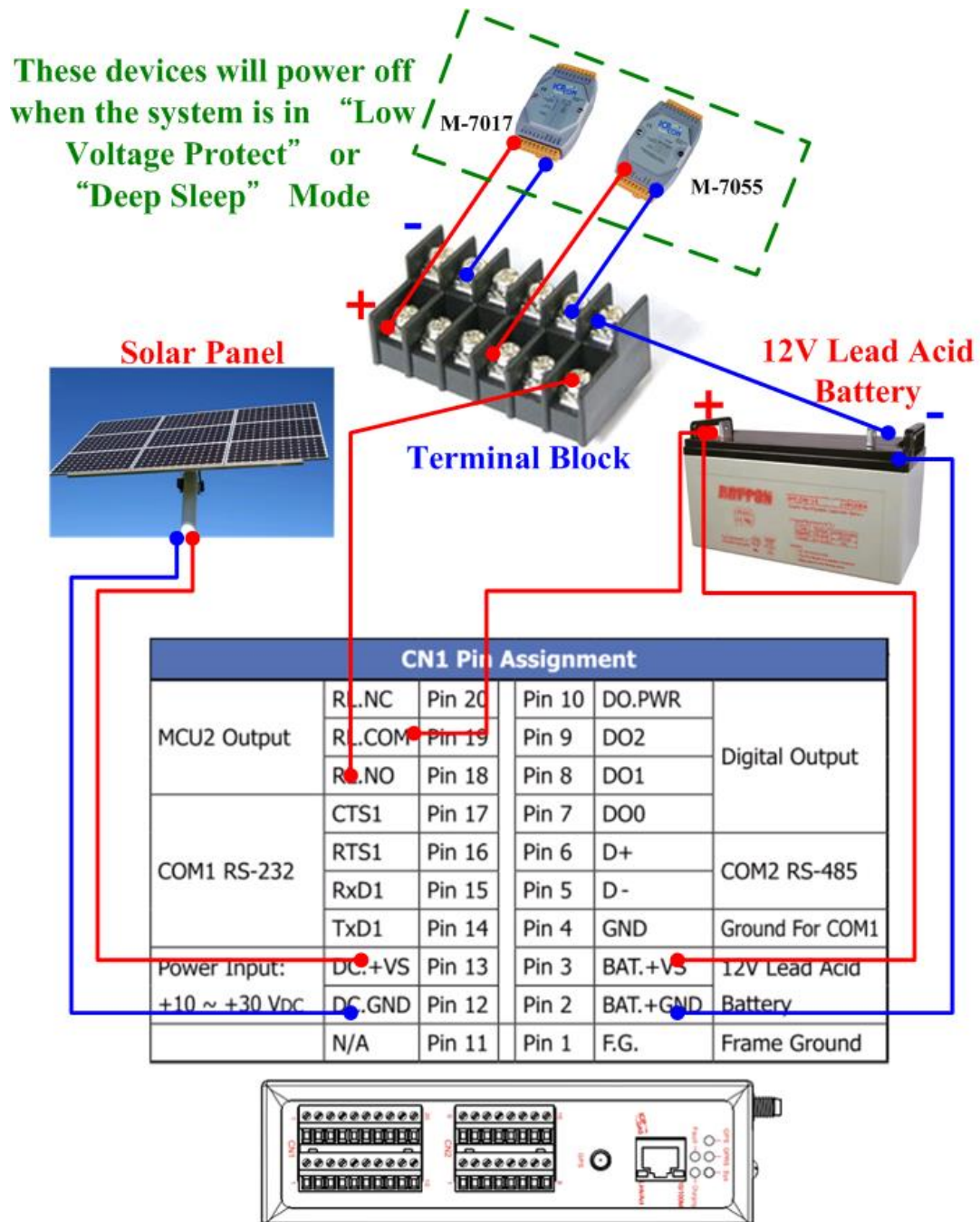
此模式會關閉所有介面，除了 MCU2 的 I/O

- 功耗：9~10 mA@12V
- 如何喚醒 G-4510：
 - (1) 睡眠時間到
 - (2) 觸發 U2_DI (連接 U2_DI 至 GND.; 讀取 U2_DI 的值為 0)

▶ 低電壓保護

預設是關閉的，可在程式內透過函式開啟低電壓保護的功能。此功能可避免電池過放電，當電池電壓低於 11.1V 時，G-4510 會進入低電壓保護模式並關閉系統電源。直到電池電壓到達 12.6V，G-4510 才會醒來開始工作

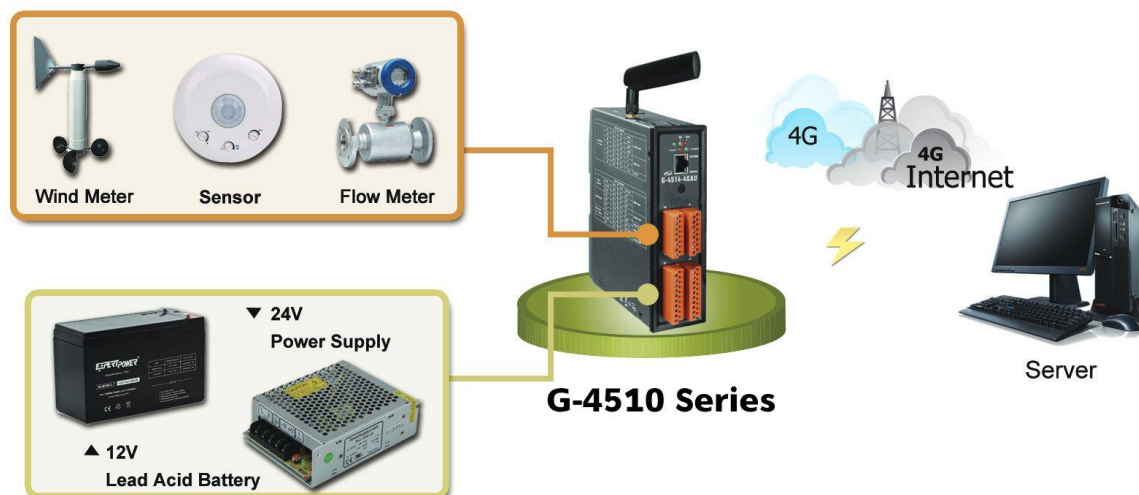
- 低電壓斷開電壓 = 11.1 V
- 低電壓重新連接電壓 = 12.6 V
- 如何使用：請參考下圖接線方式及 MCU2 的 demo 程式



5.2 如何選擇電池

此章節討論如何選擇適合您系統的電池，由於系統的存活時間取決於系統功耗及電池容量，以下範例將針對不同情況做說明

➤ 範例 1：



條件：

- 系統接 24V 外部電源
- 當外部電源關閉，必須要能持續工作兩個星期
- 系統每 10 分鐘傳輸一次資料到伺服器 (工作 1 分鐘，睡眠 9 分鐘)
- 深層睡眠的功耗：7.2 mA@12V
- 全功能運作下的平均功耗：245 mA@12V

計算：

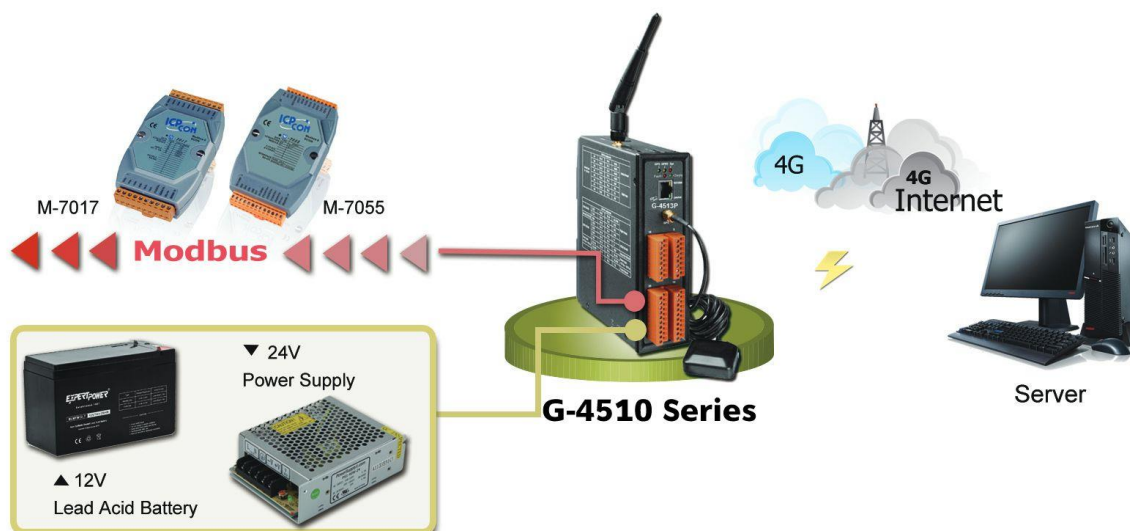
- 平均功耗 = $245 \times (1/10) + 7.2 \times (9/10) = 31 \text{ (mA)}$
- $31 \text{ (mA)} \times 24 \text{ (hours)} \times 14 \text{ (days)} = 10416 \text{ mAh}$
- 我們可能選擇 “12V, 14Ah Lead Acid Battery”
- 由於 10%的電池容量是屬於低電壓的狀況，G-4510 不在該狀態下工作，故使用 90%電池容量來做計算
- 再次確認電池容量是否符合：
 $14\text{Ah} \times 90\% \times 1000 = 12600 \text{ mAh} > 10416 \text{ mAh}$
 → 我們將選擇 “12V, 14Ah Lead Acid Battery”

不同的回報頻率該選擇的電池請參考表 5.2.1：

表 5.2.1

回報資料頻率	平均功耗	14 天的功耗 (mAh @12V)
每分鐘 (不睡眠)	245	82320
每 10 分鐘	31	10416
每小時	11.2	3763.2
每天	7.4	2486.4
每月	7.2	2419.2

➤ 範例 2：



條件：

- 系統接 24V 外部電源
- 當外部電源關閉，必須要能持續工作兩個星期
- 系統每 10 分鐘傳輸一次 3 台 Modbus 裝置的資料到伺服器 (工作 1 分鐘，睡眠 9 分鐘)
- 當 G-4510 進入深層睡眠模式時，系統會透過 “MCU2 Relay Output” 關閉所有 Modbus 裝置的電源
- 深層睡眠的功耗：7.2 mA@12V
- 全功能運作下的平均功耗：424 mA@12V

計算：

- 平均功耗 = $424 \times (1/10) + 7.2 \times (9/10) = 49 \text{ (mA)}$
- $424 \text{ (mA)} \times 24 \text{ (hours)} \times 14 \text{ (days)} = 16430.4 \text{ mAh}$
- 我們可能選擇 “12V, 22Ah Lead Acid Battery”
- 由於 10%的電池容量是屬於低電壓的狀況，G-4510 不在該狀態下工作，故使用 90%電池容量來做計算
- 再次確認電池容量是否符合：
 $22\text{Ah} \times 90\% \times 1000 = 19800 \text{ mAh} > 16430.4 \text{ mAh}$
 → 我們將選擇 “12V, 22Ah Lead Acid Battery”

不同的回報頻率該選擇的電池請參考表 5.2.2：

表 5.2.2

回報資料頻率	平均功耗	14 天的功耗 (mAh @12V)
每分鐘 (不睡眠)	424	142464
每 10 分鐘	48.9	16430.4
每小時	14.1	4737.6
每天	7.5	2520
每月	7.2	2419.2

5.3 如何選擇太陽能板

此章節討論如何選擇適合您系統的太陽能板，太陽能板的功率必須大於系統的功耗，以下範例將以模擬條件做說明

在計算前您必須知道的事情

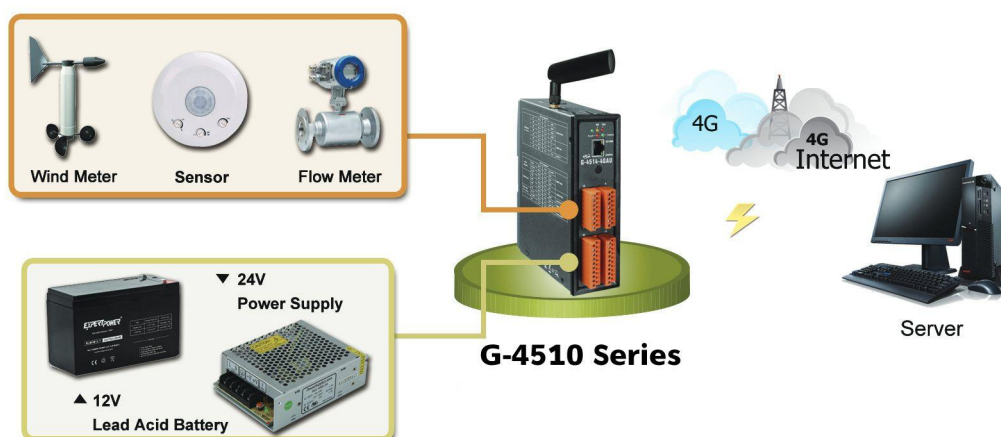


充電電壓：必須大於+16V

最大充電電流：2A

日照時數：取平均值。如果該區域的平均太陽輻射是 3 kW/m²，那麼該區域的日照時數為 3 小時

➤ 範例：



條件：

- 深層睡眠的功耗：7.2 mA@12V
- 全功能運作下的平均功耗：245 mA@12V
- 系統每 10 分鐘傳輸一次資料到伺服器 (工作 1 分鐘，睡眠 9 分鐘)
- 日照時數為 4 小時/天
- 使用 10W 的太陽能板

計算：

- 太陽能板最大電流 = $10 \text{ (w)} / 12 \text{ (V)} = 0.833 \text{ (A)} = 833 \text{ (mA)}$
- 充電電流會被很多因素所影響，例如：擺放角度、建築或其他環境因素...等
- 我們使用最大電流的一半來做計算
- $1/2$ 最大電流 = $833/2 = 416 \text{ mA}$
- 平均電流 = $416 \times 4 / 24 = 69 \text{ (mA/hr)}$
- 請參閱表 5.3.2，我們可以得知“平均功耗”是 31 mA (請參考表 5.3.2 或 5.2 節)
→ **10W** 太陽能板可用於系統，因為 $69 > 31$

表 5.3.1

太陽能板功率 (W)	最大電流 (mA)	1/2 電流 (mA)	日照時數	一天的平均電流 (mA/hr)
10	833	416	4	69
20	1666	833	4	138
30	2500	1250	4	208
40	3333	1666	4	277
50	4166	2083	4	347

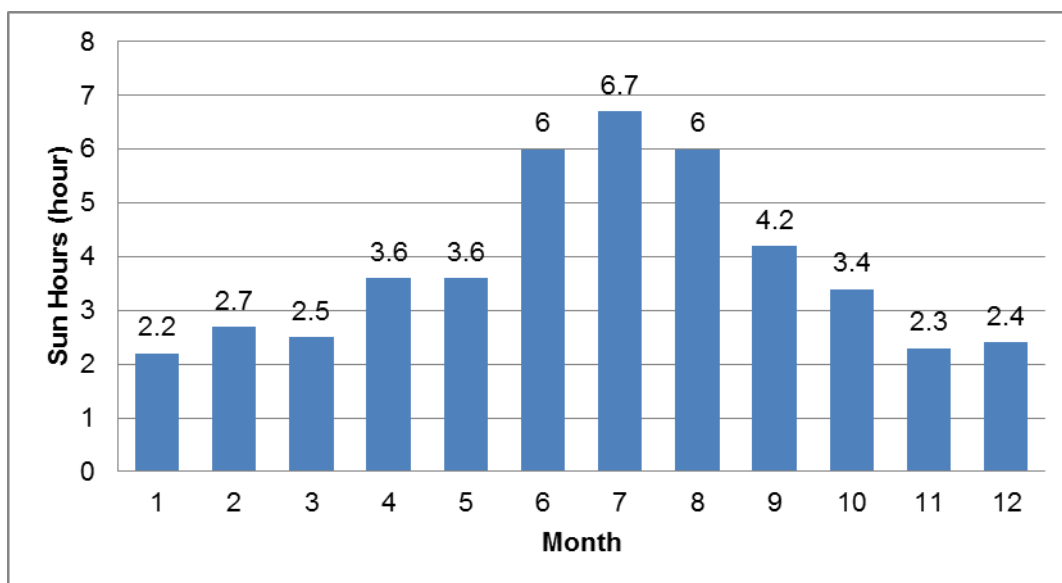
表 5.3.2

回報資料頻率	平均功耗	14 天的功耗 (mAh @12V)
每分鐘 (不睡眠)	245	82320
每 10 分鐘	31	10416
每小時	11.2	3763.2
每天	7.4	2486.4
每月	7.2	2419.2

關於“日照時數”您必須知道的事



“日照時數”是隨著季節變動的，必須使用最小的日照時數值來計算與選擇太陽能板及電池。例如：每月的日照時數如下表所示，需以 2.2 小時來計算，而不是選擇 6.7 小時來做計算。



6. API 與範例程式參考

6.1 用於 I/O 的 API

函式定義	功能說明
X305IO_Init	Initial I/O
X305IO_GetLibVersion	Get X305IO_LIB Version
X305IO_Read_AD_CalibrationGain	Read AD Calibration Gain
X305IO_Read_AD_CalibrationOffset	Read AD Calibration Offset
X305IO_AnalogIn	Read value from assign AI channel
X305IO_Read_All_DI	Read All DI
X305IO_Read_One_DI	Read the value form assign DI channel
X305IO_Write_All_DO	Write All DO
X305IO_Write_One_DO	Write the value to the assign DO channel
X305IO_Read_All_DO	Read All DO state
X305IO_Read_One_DO	Read the DO state form the assign DO channel.
X305IO_AnalogIn_SetChannel	Set the AI channel that users want to read.
X305IO_AnalogIn_Hex	Read the value from the specific A/D channel (12 bits)
X305IO_AnalogIn_HexToFloat	Transfer the AI value from 12 bits to float

6.1.1 X305IO_Init

Initial X305IO.

Syntax

```
int X305IO_Init(void);
```

Parameters

None

Return values

0 : success

<>0 : error

6.1.2 X305IO_GetLibVersion

Get X305IO_Lib Version.

Syntax

```
unsigned X305IO_GetLibVersion(void);
```

Parameters

None

Return values

Version Number

6.1.3 X305IO_Read_AD_CalibrationGain

Read the A/D Calibration Gain.

Syntax

```
float X305IO_Read_AD_CalibrationGain(void);
```

Parameters

None

Return values

Calibration Gain of the AD channels

6.1.4 X305IO_Read_AD_CalibrationOffset

Read the A/D Calibration Offset.

Syntax

```
float X305IO_Read_AD_CalibrationOffset(void);
```

Parameters

None

Return values

Calibration Offset of the AD channels

6.1.5 X305IO_AnalogIn

Read the value from the assign AI channel.

Syntax

```
float X305IO_AnalogIn(  
    int iChannel  
);
```

Parameters

iChannel

- 0 : channel 0
- 1 : channel 1
- 2 : channel 2
- 3 : channel 3
- 4 : channel 4
- 5 : channel 5
- 6 : channel 6
- 7 : channel 7

Return values

0.0mA ~ 20.0mA

6.1.6 X305IO_Read_All_DI

Read all DI values of the G-4510 series.

Syntax

```
int X305IO_Read_All_DI(void);
```

Parameters

None

Return values

0x00~0x07

Example

When DI0 Ground
DI1 Open
DI2 Open

```
value = X305IO_Read_All_DI( );
```

```
value = 0x6
```

6.1.7 X305IO_Read_One_DI

Read the value from the assign DI channel.

Syntax

```
int X305IO_Read_One_DI(  
    int iChannel  
);
```

Parameters

iChannel

0 : channel 0

1 : channel 1

2 : channel 2

Return values

1 : open

Logic high level (+3.5V ~ +30V)

0 : close to GND

Logic low level (0V ~ +1V)

6.1.8 X305IO_Write_All_DO

Write to all DO values of the G-4510 series.

Syntax

```
void X305IO_Write_All_DO(  
    int iOutValue  
);
```

Parameters

iOutValue
0x0~0x7

Return values

None

Example

```
X305IO_Write_All_DO(6);
```

After function execute :

DO0 OFF

DO1 ON

DO2 ON

6.1.9 X305IO_Write_One_DO

Write the specific value to the assign DO channel.

Syntax

```
void X305IO_Write_One_DO(  
    int iChannel, int iStatus  
);
```

Parameters

iChannel

- 0 : channel 0
- 1 : channel 1
- 2 : channel 2

iStatus

- 0 : Status is OFF
- 1 : Status is ON

Return values

None

6.1.10 X305IO_Read_All_DO

Read all DO values of the G-4510 series.

Syntax

```
int X305IO_Read_All_DO(void);
```

Parameters

None

Return values

0x0~0x7

Example

When DO0 OFF
DO1 ON
DO2 ON

Value = X305IO_Read_All_DO();

Value = 0x6

6.1.11 X305IO_Read_One_DO

Read the state from the assign DO channel.

Syntax

```
int X305IO_Read_One_DO(  
    int iChannel  
);
```

Parameters

iChannel

0 : channel 0

1 : channel 1

2 : channel 2

Return values

0 : OFF

1 : ON

6.1.12 X305IO_AnalogIn_SetChannel

Set the specific AI channel that users want to read.

Syntax

```
int X305IO_AnalogIn_SetChannel(  
    unsigned iChannel  
);
```

Parameters

iChannel

- 0 : channel 0
- 1 : channel 1
- 2 : channel 2
- 3 : channel 3
- 4 : channel 4
- 5 : channel 5
- 6 : channel 6
- 7 : channel 7

Return values

- 0 : Set up success
- 1 : Set iChannel number error

6.1.13 X305IO_AnalogIn_Hex

Read the value of the assign AI channel assigned by X305IO_AnalogIn_SetChannel function.

Syntax

```
int X305IO_AnalogIn_Hex(void);
```

Parameters

None

Return values

After Read assign AI channel value.

Example

```
X305IO_AnalogIn_SetChannel(0); // Set channel 0  
X305IO_AnalogIn_Hex( );
```


6.1.14 X305IO_AnalogIn_HexToFloat

Set the AI value from 12 bits to float format.

Syntax

```
float X305IO_AnalogIn_HexToFloat(  
    int iValue  
);
```

Parameters

iValue

A value want to 12 bits transform float.

Return values

The transferred AI value by float format.

Example

Set the channel 0 to read, and then transform the value to float.

```
float AdValue;
```

```
X305IO_AnalogIn_SetChannel(0);
```

```
AdValue=X305IO_AnalogIn_HexToFloat(X305IO_AnalogIn_Hex( ));
```

6.2 用於 MMC/SD 的 API



所需函式庫與標頭檔：
SD_Vnnn.LIB and microSD.h

函式定義	功能說明
pc_init	Initializes the SD socket library
pc_open	1. Open an existing file and return a file handle 2. Creates a new file
pc_close	Closes a file and release a file handle.
pc_read	Reads the specified file
pc_write	Writes the specified file
pc_seek	Moves the file pointer to relative offset from the current offset
pc_tell	Gets current offset of the file pointer
pc_eof	Checks whether the end-of-file is reached
pc_format	Formats the SD card as FAT (FAT32)
pc_mkdir	Creates a directory or subdirectory
pc_rmdir	Removes an existing directory
pc_move	Renames an existing file or a directory, including the subdirectory
pc_del	Deletes the specified file
pc_deltree	Deletes the specified directory or subdirectory
pc_isdir	Checks whether the file is a directory
pc_ismvol	Checks if is a volume
pc_size	Gets the size of the specified file

pc_set_cwd	Sets the current working directory
pc_get_cwd	Gets the pathname of the current working directory
pc_gfirst	Moves the pointer to the first element
pc_gnext	Moves the pointer to the next element
pc_gdone	Moves the pointer to the last element
pc_get_freeSize_KB	Gets the free space of the SD memory card
pc_get_usedSize_KB	Gets the used space of the SD memory card
pc_get_totalSize_KB	Gets the total size of the SD memory card
pc_get_attributes	Gets the file attributes
pc_set_attributes	Sets the file attributes
pc_get_errno	Gets the error number

▶ 開始使用 SD 之 API

1. pc_init()
使用 SD 功能前，需呼叫 pc_init() 來初始化 SD

▶ 啟用/關閉 SD 之 API

2. pc_open()
在寫入／讀取資料至 SD 卡之前，需呼叫 pc_open() 來開啟檔案
3. pc_close()
完成寫入／讀取資料至 SD 卡後，需呼叫 pc_close() 依檔案控制代碼 (File Handle) 來關閉檔案

▶ 寫入資料至 SD 之 API

4. pc_write()
此函式可添加一個指定數量之同等大小的資料項目於 SD 裡的檔案中

範例一 寫入資料至 microSD 卡：

```
#include <string.h>
#include <stdio.h>
#include "upac5000.h"
#include "microSD.h"
void main(void)
{
    int fd, iRet;
    InitLib();
    if(pc_init())
        Print("Init microSD ok\r\n");
    else
    {
        Print("Init microSD failed\r\n");
        iRet=pc_get_errno();
        switch(iRet)
        {
            case PCERR_BAD_FORMAT: //1
                Print("Error 01: format is not FAT\r\n");
                break;
            case PCERR_NO_CARD: //2
                Print("Error 02: no microSD card\r\n");
                break;
            default:
                Print("Error %02d: unknow error\r\n", iRet);
                break;
        }
    }
    fd=pc_open("test.txt", (word) (PO_WRONLY|PO_CREAT|PO_APPEND),
    (word) (PS_IWRITE|PS_IREAD));
    if(fd>=0)
    {
        pc_write(fd, "1234567890", 10); /* write 10 bytes */
        pc_close(fd);
    }
}
```

▶ 從 SD 卡中讀取資料之 API

5. pc_read()

使用 `pc_open()` 開啟檔案後，需呼叫 `pc_read()` 來讀取 SD 中的資料

範例－讀取 microSD 中的資料：

```
#include <string.h>
#include <stdio.h>
#include "upac5000.h"
#include "microSD.h"
void main(void)
{
    int fd, iRet;
    unsigned char Buffer[80];
    InitLib();
    if(pc_init())
        Print("Init microSD ok\r\n");
    else
    {
        Print("Init microSD failed\r\n");
        iRet=pc_get_errno();
        switch(iRet)
        {
            case PCERR_BAD_FORMAT: //1
                Print("Error 01: format is not FAT\r\n");
                break;
            case PCERR_NO_CARD: //2
                Print("Error 02: no microSD card\r\n");
                break;
            default:
                Print("Error %02d: unknow error\r\n", iRet);
                break;
        }
    }
    fd=pc_open("test.txt", (word) (PO_RDONLY), (word) (PS_IWRITE|PS_IREAD));
    if(fd>=0)
    {
```

```
iRet=pc_read(fd, Buffer, 10); /* reads 10 bytes */
Buffer[10]=0; /* adds zero end to the end of the string */
pc_close(fd);
Print("%s", Buffer);
}
}
```

請參閱下列位置，取得關於 microSD 的範例程式：

CD:\napdos\g-4510\software\demo\basic\microSD\

<http://ftp.icpdas.com/pub/cd/usbcd/napdos/g-4510/software/demo/basic/microsd/>

6.3 用於 LCD 的 API

函式定義	功能說明
LCD_Init	Initialize the library
LCD_BackLight_On	Turn on the LCD backlight
LCD_BackLight_Off	Turn off the LCD backlight
LCD_ShowText	Display one character on the LCD panel
LCD_ClrScrn	Clear the LCD panel
LCD_StandByMode	Enter the stand by mode
LCD_NormalMode	Restore the LCD to normal mode
LCD_GotoPosition	Move the cursor to the specified position
LCD_CursorDisplay	Set the Cursor display status
LCD_LineReverse	Select one of four line and reverse the display
LCD_LineRestore	Select one of four line and restore the display
LCD_GetLibDate	Gets the create date of funciton library
LCD_GetLibVersion	Gets the version number of function library

6.3.1 LCD_Init

Initialize parameters about LCD functions in the library.

Syntax

```
void LCD_Init(void);
```

Parameters

None

Return values

None

6.3.2 LCD_BackLight_On

Turn on the LCD backlight.

Syntax

```
void LCD_BackLight_On(void);
```

Parameters

None

Return

None

6.3.3 LCD_BackLight_Off

Turn off the LCD backlight.

Syntax

```
void LCD_BackLight_Off(void);
```

Parameters

None

Return values

None

6.3.4 LCD_ShowText

Display one character on the LCD panel, and the cursor will right-shifted by one character position automatically.

Syntax

```
void LCD_ShowText(  
    uchar Text  
);
```

Parameters

Text

Display character

Return values

None

6.3.5 LCD_ClrScrn

Clear the LCD panel.

Syntax

```
void LCD_ClrScrn(void);
```

Parameters

None

Return values

None

6.3.6 LCD_StandByMode

Enter the stand by mode, and it can be terminated by either LCD_NormalMode() or other function.

Syntax

```
void LCD_StandByMode(void);
```

Parameters

None

Return values

None

6.3.7 LCD_NormalMode

Restore the LCD to normal mode when it is in the stand by mode.

Syntax

```
void LCD_NormalMode(void);
```

Parameters

None

Return values

None

6.3.8 LCD_GotoPosition

Move the cursor to the specified position.

Syntax

```
void LCD_GotoPosition(  
    int Line,  
    int Offset  
);
```

Parameters

Line

One of four line numbers (1 to 4)

Offset

Cursor position (1 to 8)

Return values

None

6.3.9 LCD_CursorDisplay

Set the Cursor display status.

Syntax

```
void LCD_CursorDisplay(  
    int Display,  
    int Blink  
);
```

Parameters

Display

Cursor display on/off

1: Display on

0: Display off

Blink

Character blink on/off

1: Display on

0: Display off

Return values

None

6.3.10 LCD_LineReverse

Select one of four line and reverse the display.

Syntax

```
void LCD_LineReverse(int Line);
```

Parameters

Line

One of four line numbers (0 to 4)

Return values

None

6.3.11 LCD_LineRestore

Select one of four line and restore the display.

Syntax

```
void LCD_LineRestore(  
    int Line  
);
```

Parameters

Line

One of four line numbers (0 to 4)

Return values

None

6.3.12 LCD_GetLibDate

Gets the create date of funciton library.

Syntax

```
void LCD_GetLibDate(  
    unsigned char *LibDate  
);
```

Parameters

LibDate

Gets the create date of funciton library

Return values

None

6.3.13 LCD_GetLibVersion

Get the version number of function library.

Syntax

```
unsigned LCD_GetLibVersion(void);
```

Parameters

None

Return values

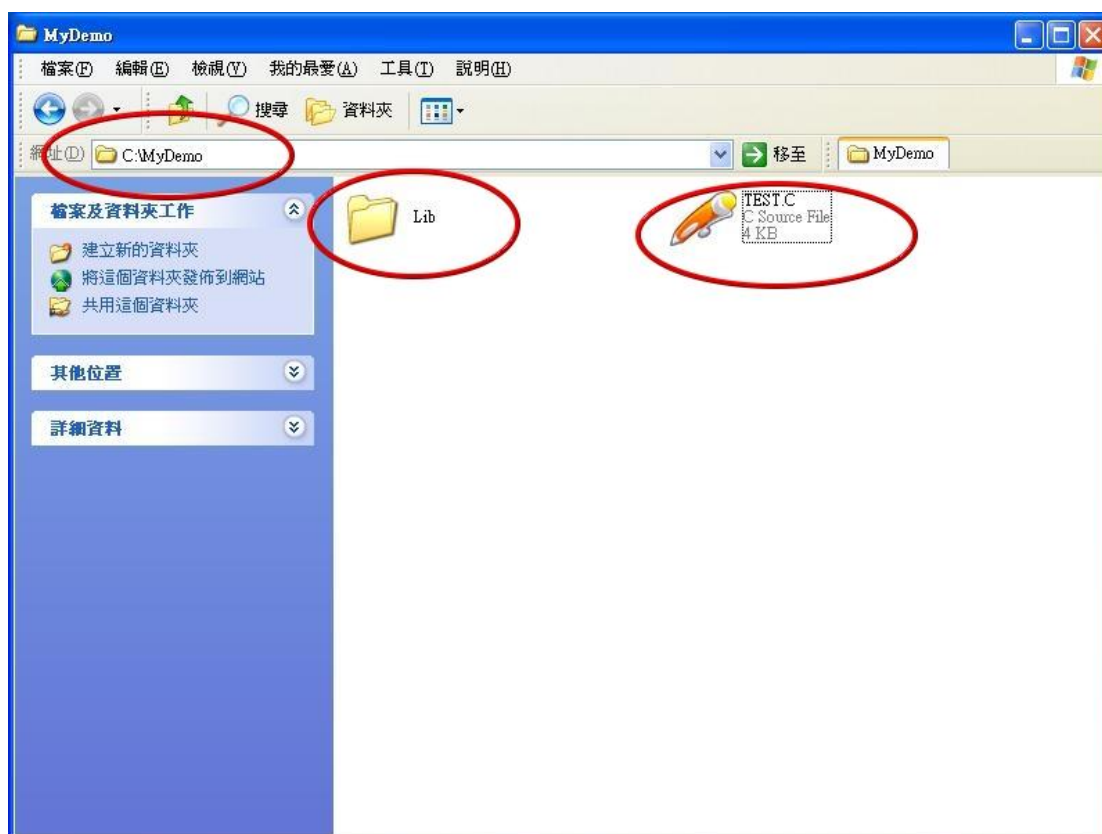
Return the current version number.

7. 程式編譯與下載步驟

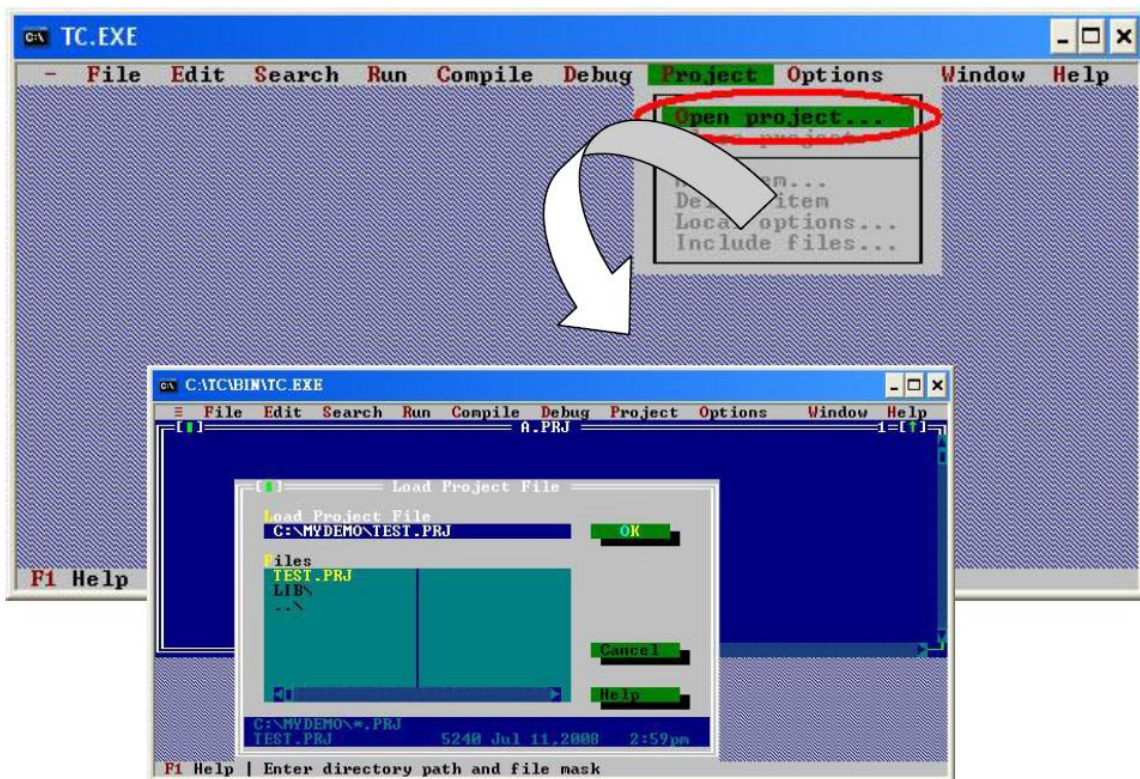
此章節將介紹如何編譯與執行 G-4510 的程式

Library	Description
G4500.LIB	G-4510 and DI/O、AI functions
GSM.LIB	GPRS functions
SD_Vnnn.LIB	MMC/SD functions
TCP_DM32.LIB	Ethernet functions
LCD.LIB	LCD functions

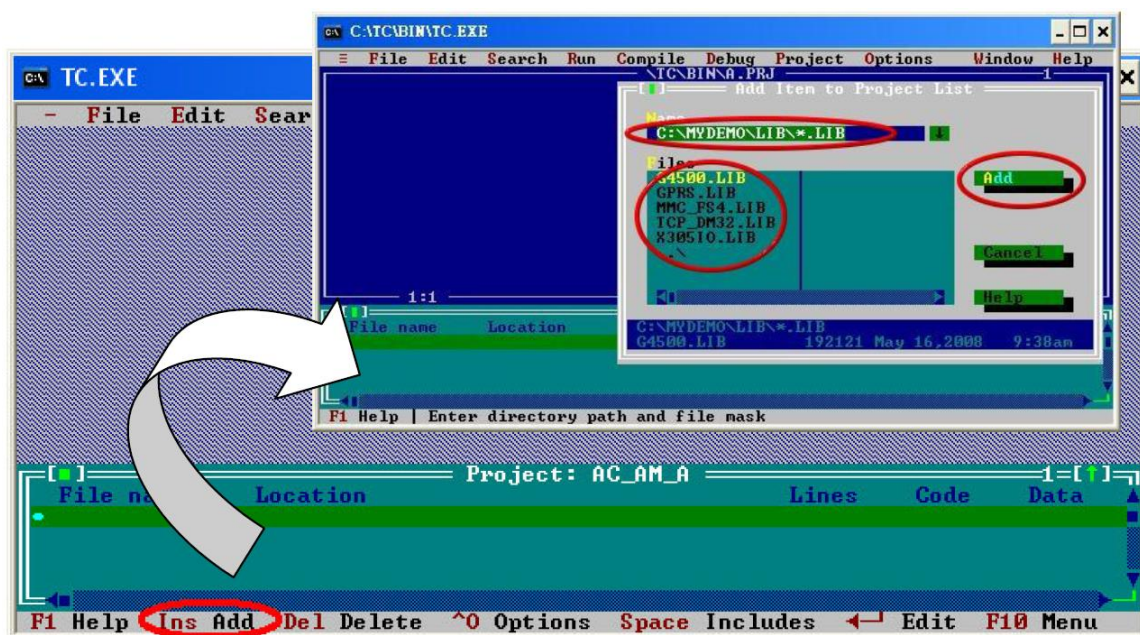
步驟 1：在 C 槽建立資料夾並取名為 “MyDemo”，複製 lib 及程式到 MyDemo 資料夾內



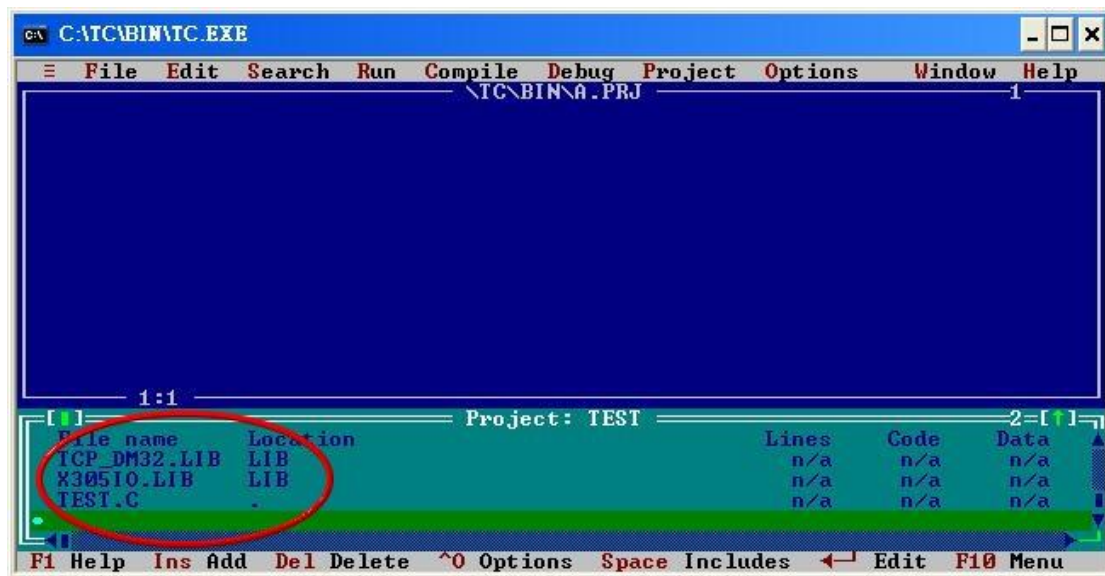
步驟 2：執行 TC++ 1.0，按下“Project\Open project...”建立新的專案檔並取名為“TEST.PRJ”



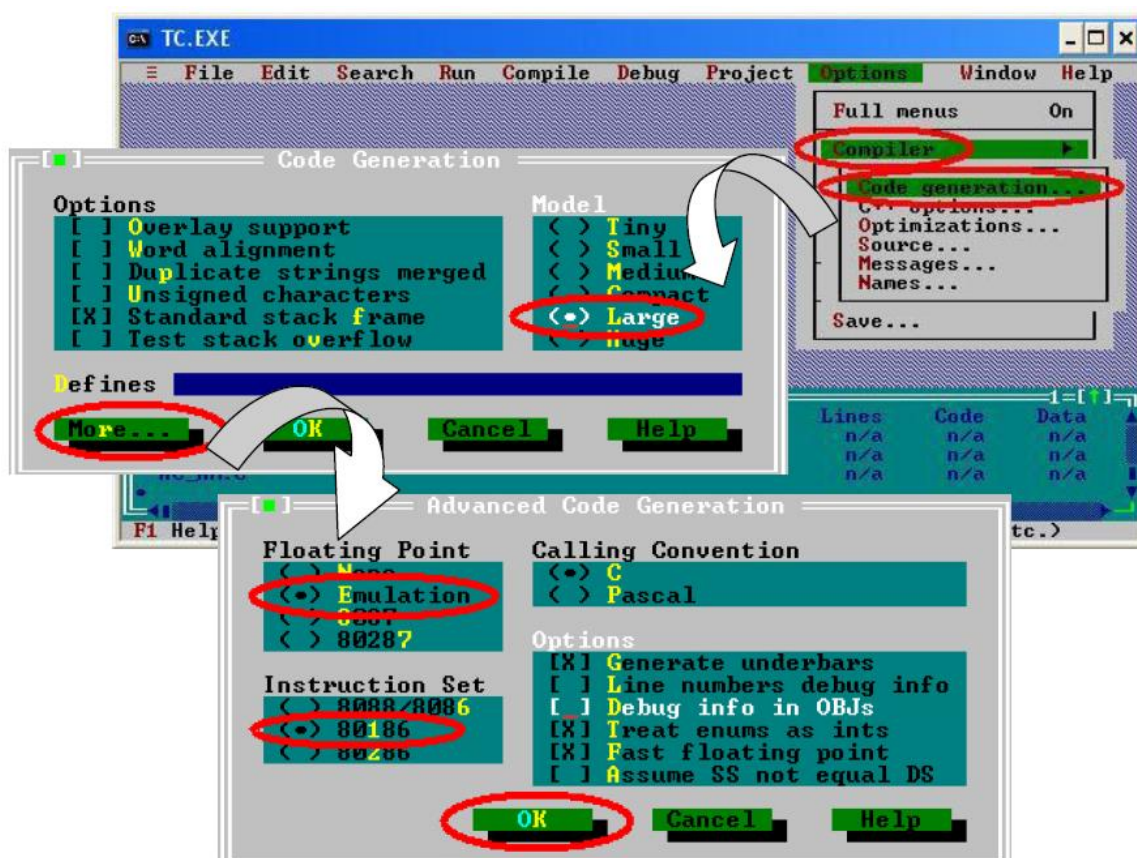
步驟 3：使用“Add”功能加入編譯需用到的函式庫檔案到 MyDemo 資料夾



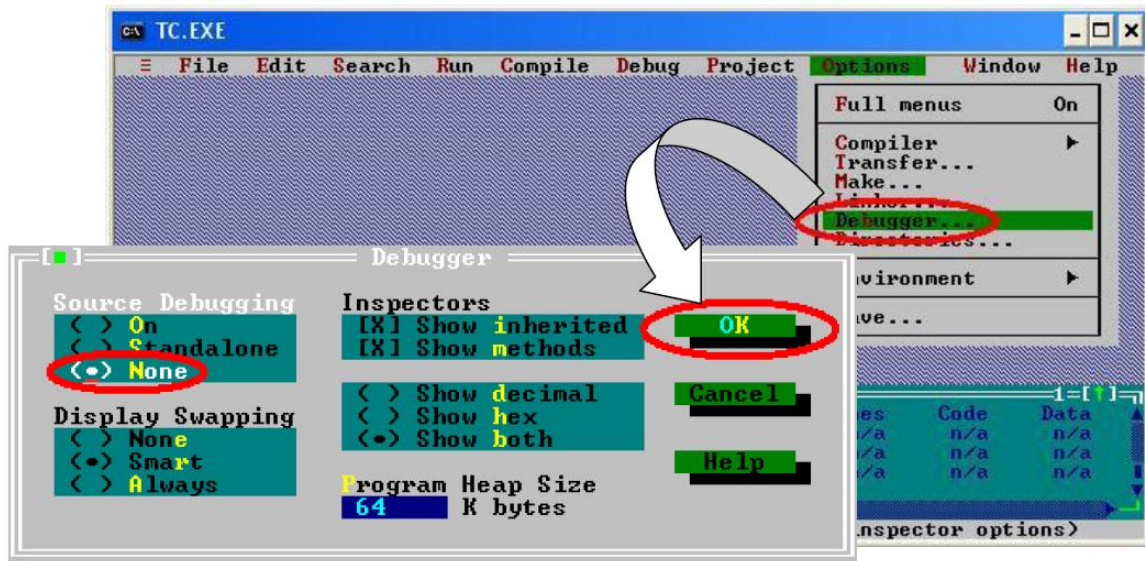
步驟 4：參照步驟 3，加入其它函式庫檔案及 TEST.c 到 MyDemo 資料夾



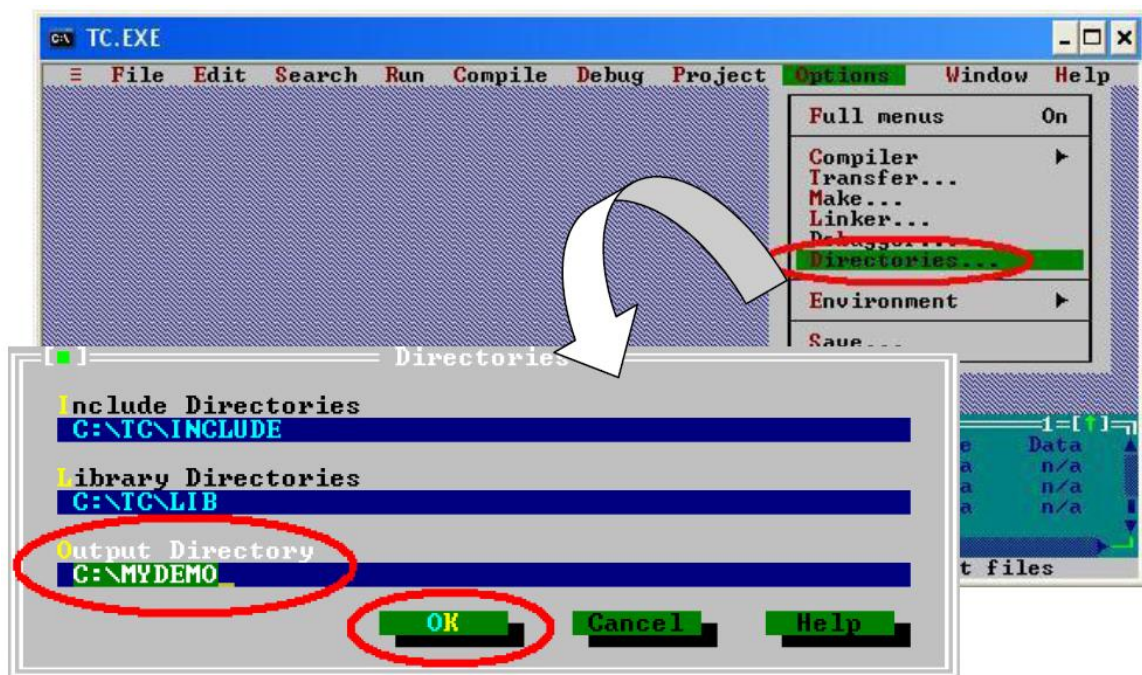
步驟 5：點選 “Options/Compiler/Code generation...” 將編譯模式設為 Large，接著點選 “More...” 分別設定 “Floating point” 和 “Instruction Set” 參數為 Emulation 和 80186。設定完後按 OK 儲存設定。



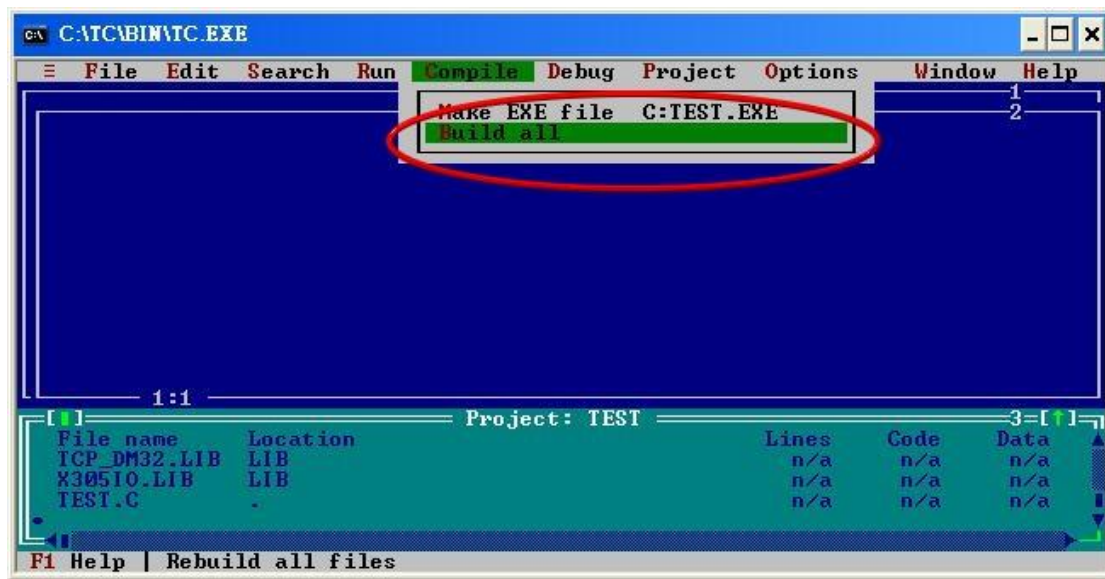
步驟 6：點選“Option/Debugger...”將“Source Debugging”參數設定為“None”



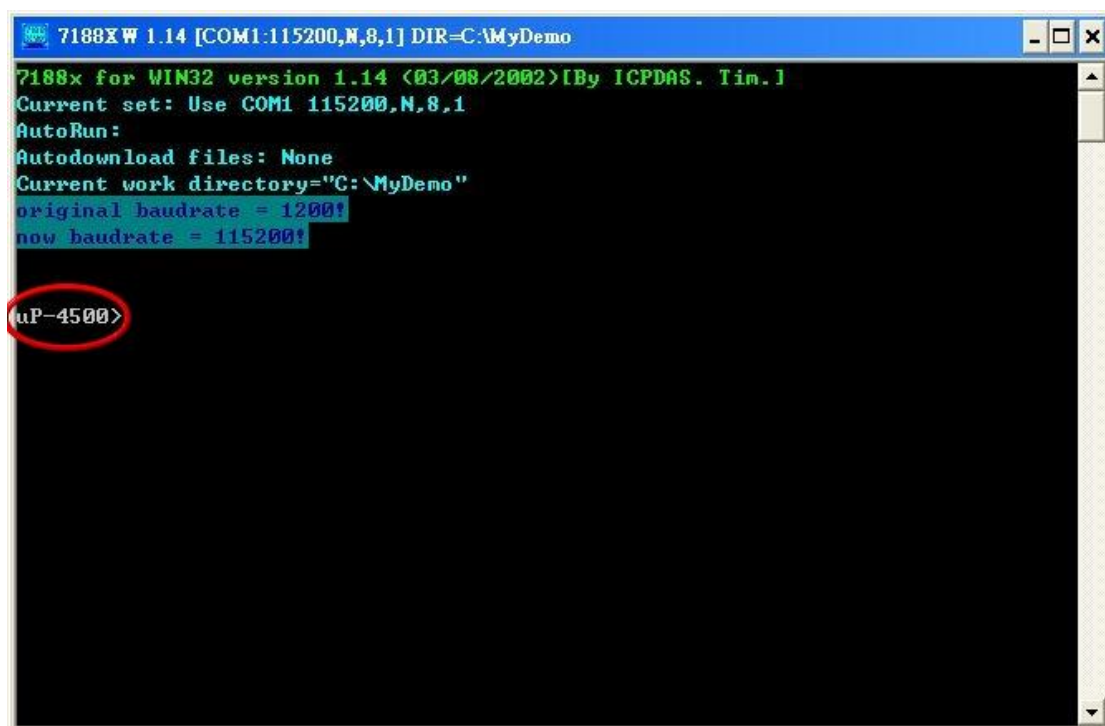
步驟 7：點選“Option/Directories...”設定“Output Directory”輸出目錄為“C:\MyDemo”



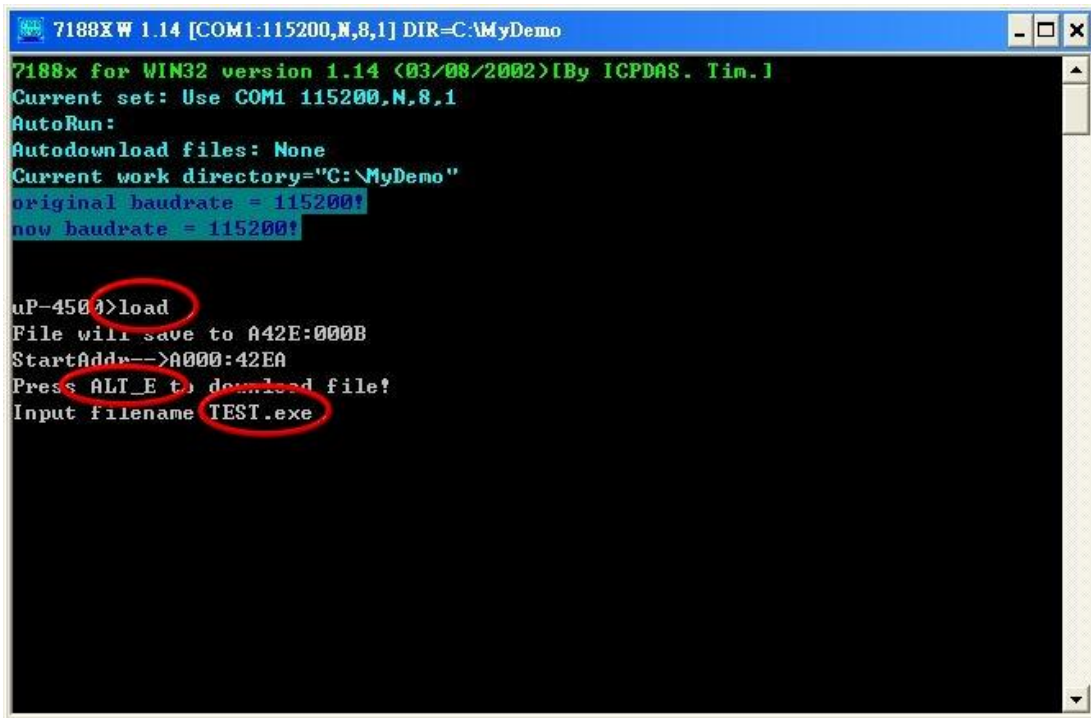
步驟 8：設定完所有參數後，按下“Compile/build all”產生執行檔“TEST.exe”



步驟 9：複製 7188XW.exe 到 MyDemo 目錄，然後雙擊左鍵執行 7188XW.exe，並將 G-4510 系列的 COM1 連接至 PC 的 RS-232



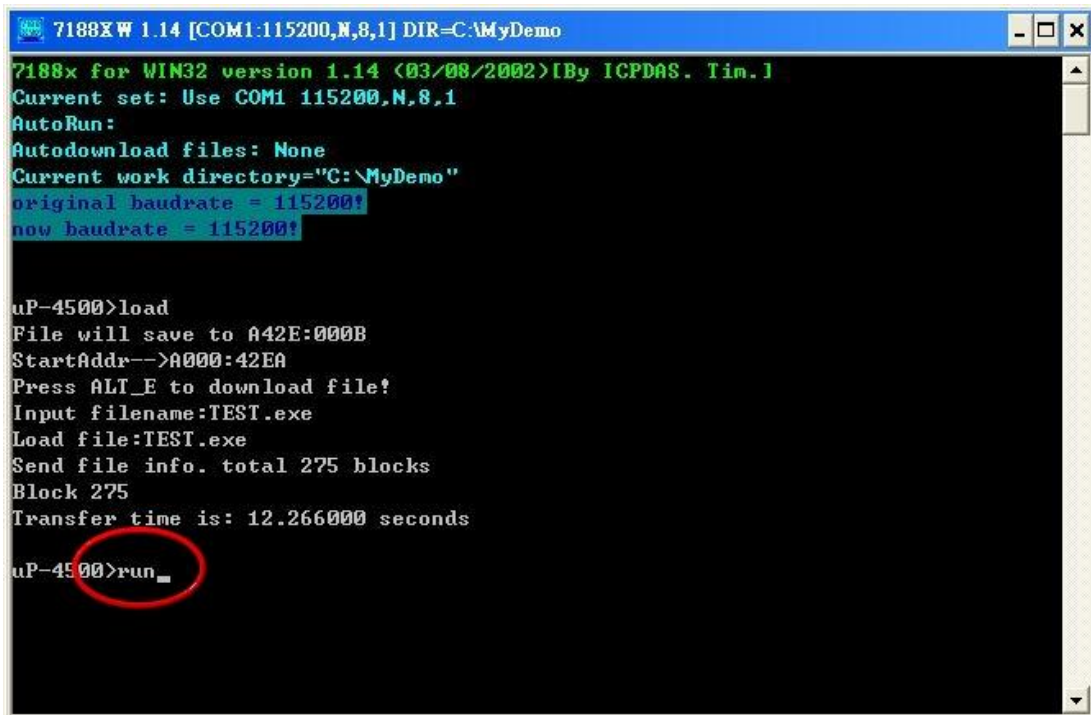
步驟 10：在 7188xw.exe 輸入指令 “load”，按下 “Alt+E” 並輸入 “TEST.exe” 後，開始下載程序



```
7188XW 1.14 [COM1:115200,N,8,1] DIR=C:\MyDemo
7188x for WIN32 version 1.14 (03/08/2002)[By ICPDAS. Tim.]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="C:\MyDemo"
original baudrate = 115200!
now baudrate = 115200!

uP-4500>load
File will save to A42E:000B
StartAddr-->A000:42EA
Press ALT_E to download file!
Input filename TEST.exe
```

步驟 11：下載完成後，輸入 “run” 開始執行 “TEST.exe”



```
7188XW 1.14 [COM1:115200,N,8,1] DIR=C:\MyDemo
7188x for WIN32 version 1.14 (03/08/2002)[By ICPDAS. Tim.]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="C:\MyDemo"
original baudrate = 115200!
now baudrate = 115200!

uP-4500>load
File will save to A42E:000B
StartAddr-->A000:42EA
Press ALT_E to download file!
Input filename:TEST.exe
Load file:TEST.exe
Send file info. total 275 blocks
Block 275
Transfer time is: 12.266000 seconds

uP-4500>run
```

8. 版本記錄

版本	日期	作者	說明
1.0.0	2019/08/22	Amon	第一版