G-4513 系列

3G Power Saving PAC with Solar Charger

操作手冊 **V1.0.1**

High Quality, Industrial Data Acquisition, and Control Products

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產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，但保留修定權利，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

版權

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

聯繫我們

若是您有任何問題，請隨時與我們聯繫

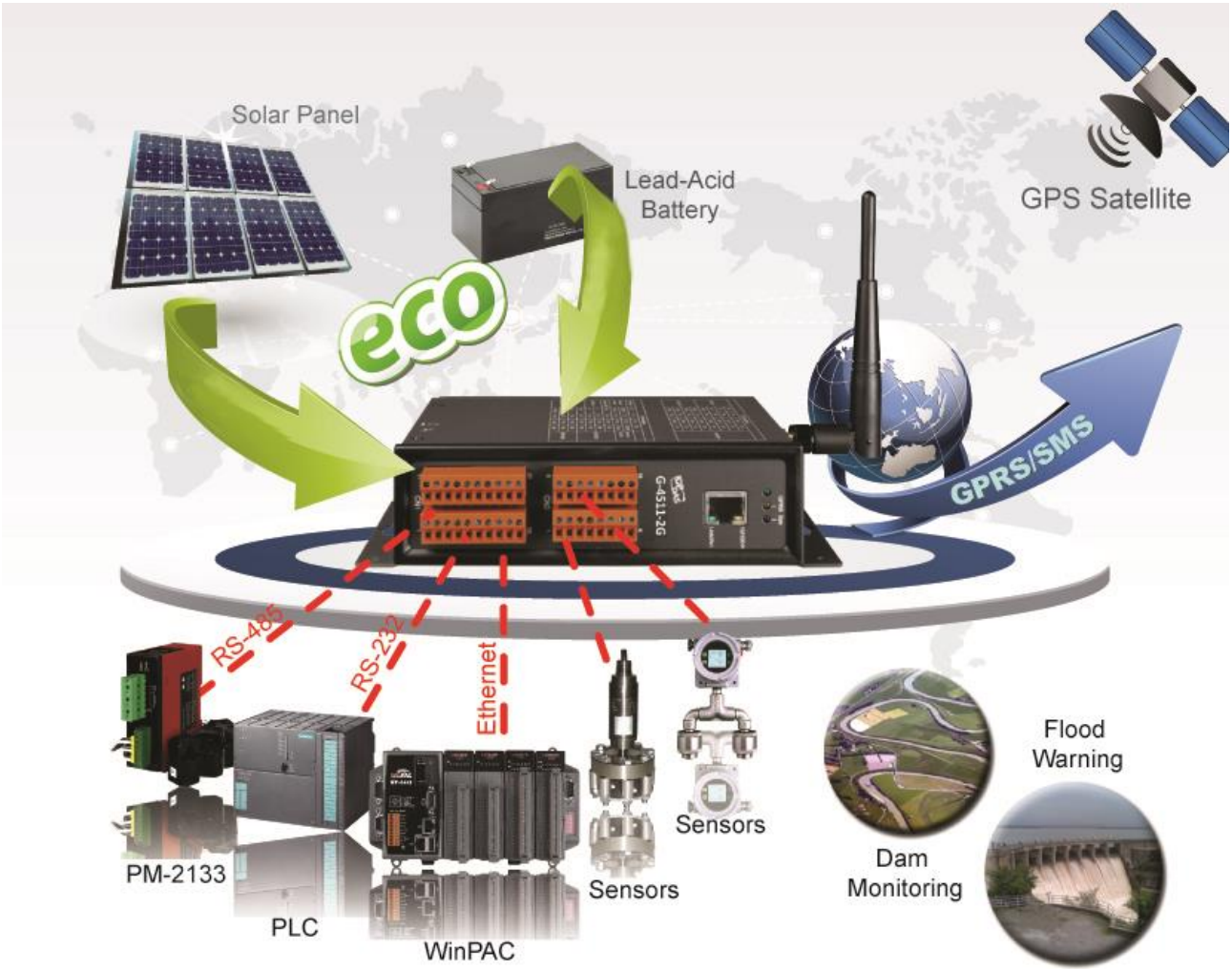
Email：service@icpdas.com

Table of Contents

1. 簡介

G-4513 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-4513 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-4513 系列擁有太陽能充電器、3G 模組、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 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-4513 系列屬於熟悉的開發環境，較容易上手。



2. 硬體規格

2.1 G-4513-3GWA 系列

G-4513-3GWA	G-4513D-3GWA	G-4513P-3GWA	G-4513PD-3GWA
			

2.2 G-4513-3GWA 系列規格

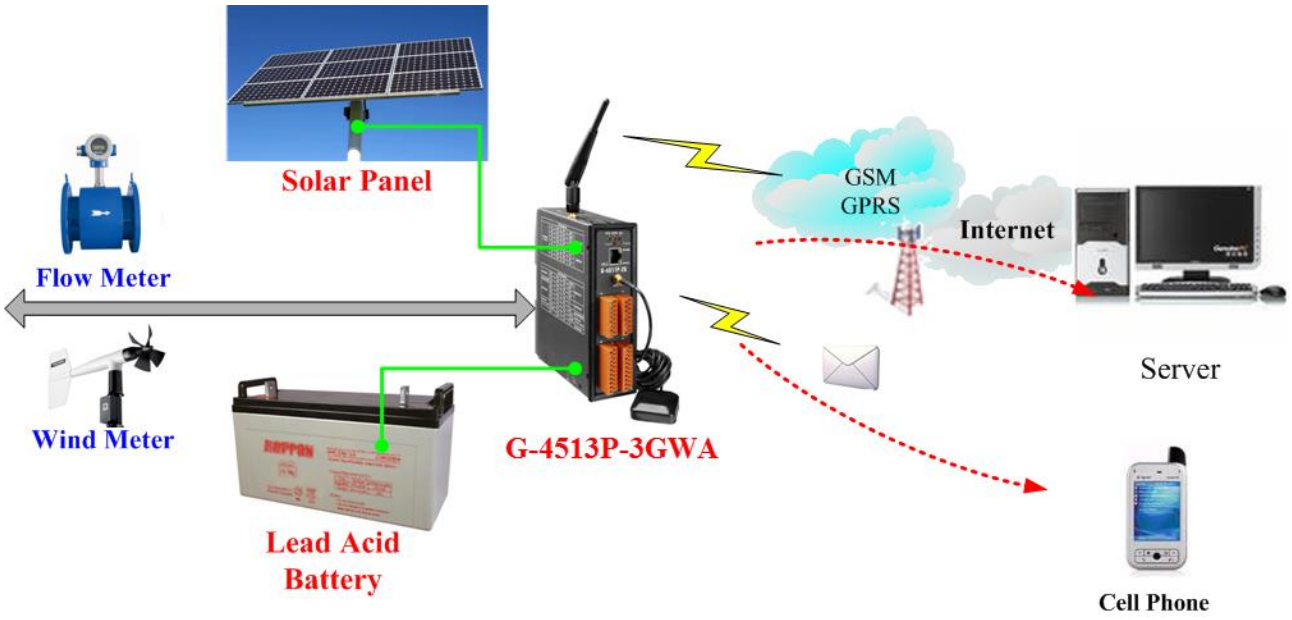
Item	G-4513-3GWA	G-4513D-3GWA	G-4513P-3GWA	G-4513PD-3GWA
CPU	80 MHz internal microprocessor			
SRAM/Flash	512K/512K , real time clock, watchdog timer			
NVRAM	31 bytes, battery backup, data valid up to 10 years			
EEPROM	16 KB, retention > 40 years. 1,000,000 erase/write cycles			
Comm. Interface				
COM ports	COM1:5-wire RS-232; COM2: RS-485			
Ethernet	10/100 Base-TX Ethernet controller			
GSM Interface				
Frequency Band	GSM 850/900/1800/1900 MHz			
GPRS connectivity	GPRS class 10/8; GPRS station class B			
SMS	MT, MO, CB, Text and PDU mode			
3G Interface				
Frequency Band	WCDMA 850/900/1900/2100 MHz			
Power Class	Class 3 (250mW @ WCDMA/HSPA)			
Digital Input				
Input Channel	3			
Input Type	Source(Dry Type), Common Ground			
Off Voltage Level	+1 V max.			
On Voltage Level	+3.5 ~ +30 V			
Isolated Voltage	Non-isolated			
Digital Output				
Output Channel	3			
Output Type	3 Open Collector (Sink/NPN)			
Load Voltage	+30 VDC max.			
Load Current	100 mA max.			
Isolated Voltage	Non-isolated			
Analog Input				
Input Channel	8			
Resolution	12 - bit			
Input Range/Type	0 ~ 20 mA			
Sample Rate	1 KHz max. (Read one channel)			
Accuracy	+/- 2 LSB (+/- 0.0097 mA)			
Isolated Voltage	2500Vrms 3000Dc to DC			
Relay				
Output Channel	1			
Type	Form C			

Input Range		2A@30 Vdc ; 0.25 A @250 Vac			
Mechanical endurance		typ. 10^8 operations			
GPS Interface					
Support Channels		-		32	
Sensitivity		-		Tracking = up to -159 dBm (with external LNA)	
				Cold start = up to -146 dBm (with external LNA)	
Acquisition Time		-		Hot start (Open Sky) = 2 s(typical)	
				Cold start (Open Sky) = 36 s(typical)	
Protocol Support		-		NMEA 0183 version 3.01	
LCD Interface					
General	Effective display area	-	80.61 mm x 14.37 mm (W x H)	-	80.61 mm x 14.37 mm (W x H)
	Module Dimension	-	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)
Life Time		-	Expected life is more than 100,000 hours under normal operation	-	Expected life is more than 100,000 hours under normal operation
Power (Solar Input)					
Protection		Power reverse polarity protection			
Frame Ground Protection		ESD, Surge, EFT, Hi-Pot			
Power Requirement		+10 V_{DC} ~ +30 V_{DC} , (Max. Voltage of Solar Panel must less +30V)			
Power Consumption		Deep Sleep: < 10 mA@12V_{DC}; Deep Sleep(With LCD): < 11 mA@12V_{DC}; Sleep: < 15 mA@12V_{DC}; Idle: 77 mA @ 24 V_{DC}; Data Link: 150 ~ 400 mA (peak) @ 24 V_{DC}			
Lead Acid Battery Requirement					
Battery		12V Lead-Acid Battery			
Charging Voltage		Voltage of Power Input must be over +16V			
Low Voltage Protect		Low Voltage disconnect = 11.1V / Low Voltage reconnect = 12.6V			
LED Indicators					
System		Red			

3G	Yellow			
GPS	-		Green	
Charging / Fault	Green / Red			
Mechanical				
Casing	Metal			
Dimensions	47 mm x 142 mm x 168 mm (W x L x H)			
Installation	DIN-Rail and Wall mount			
Environment				
Operating Temperature	-20 ~ +70 °C	-15 ~ +55 °C	-20 ~ +70 °C	-15 ~ +55 °C
Storage Temperature	-40 ~ +80 °C	-20 ~ +70 °C	-40 ~ +80 °C	-20 ~ +70 °C
Humidity	5~90% RH, non-condensing			

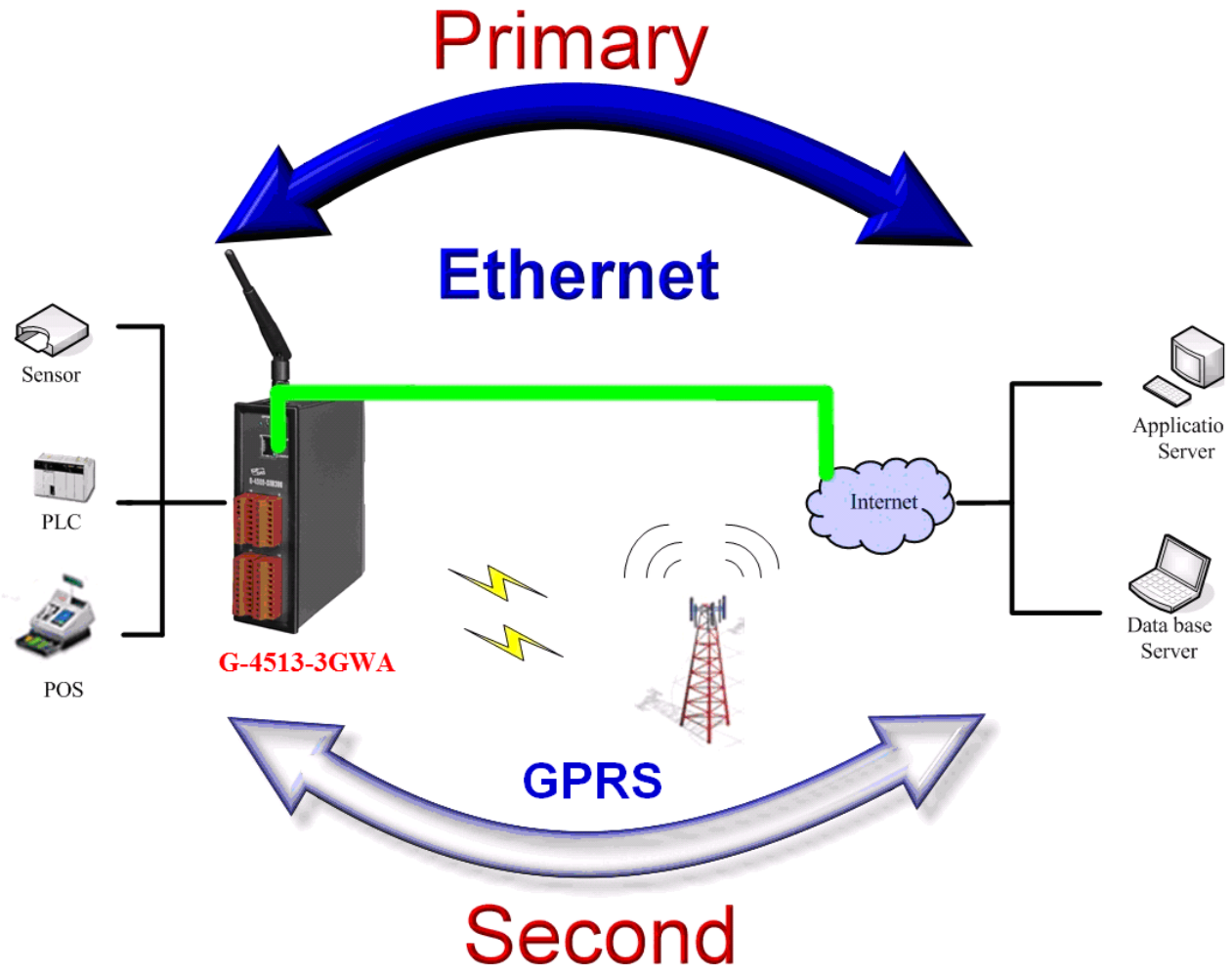
3. 應用架構

3.1 水文/風場監測應用



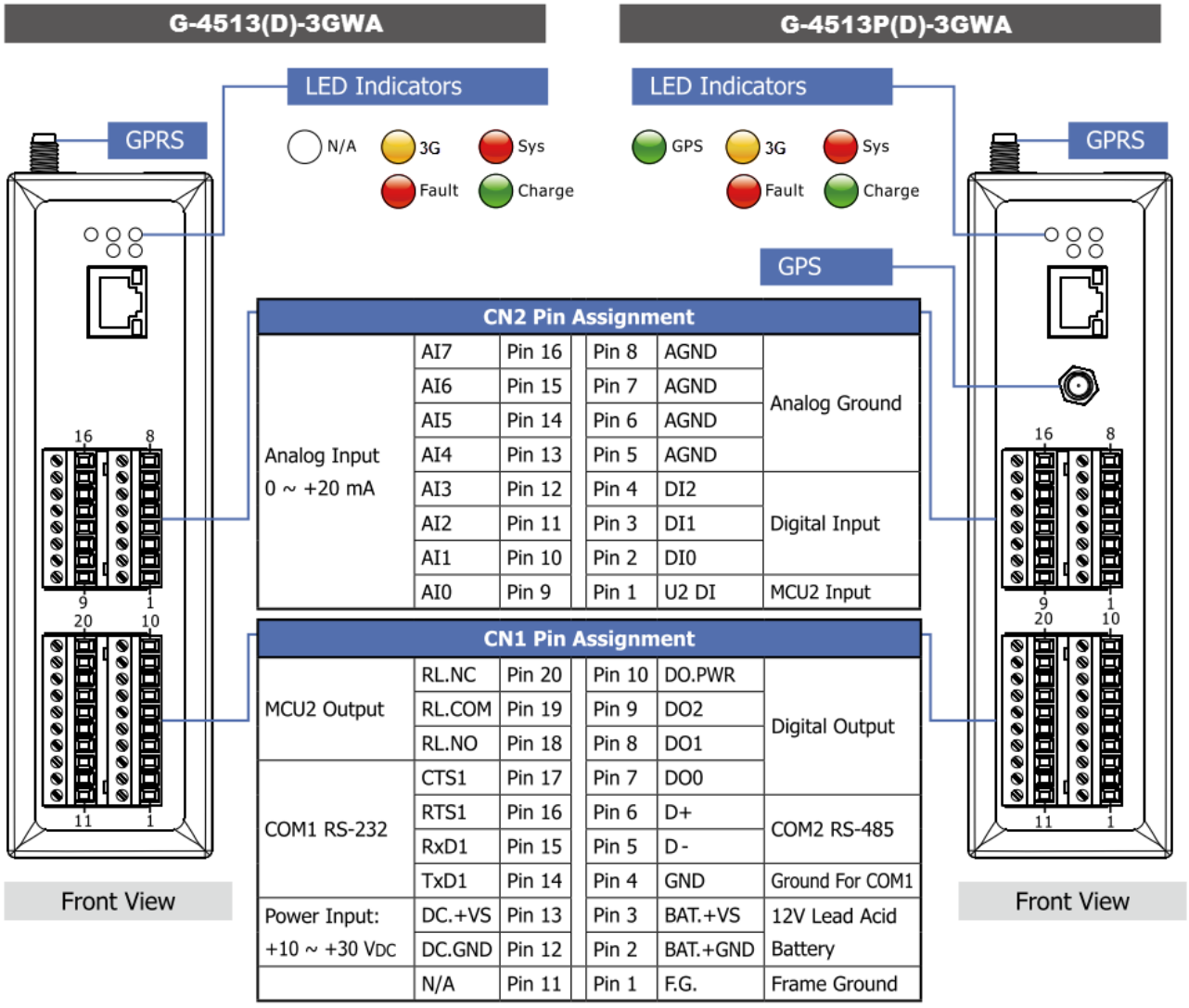
3.2 車輛監測/追蹤系統

3.3 備援通訊系統


Primary
Ethernet
Sensor
PLC
POS
G-4513-3GWA
Internet
Applicatio
Server
Data base
Server
GPRS
Second

4. 硬體

4.1 腳位配置

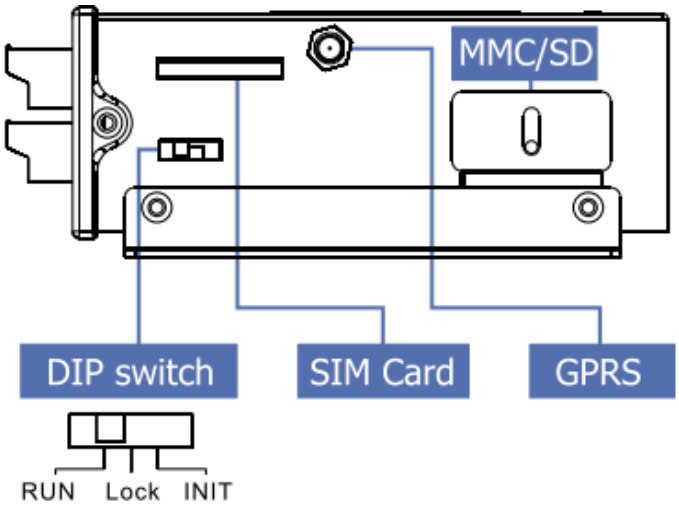


CN2 Pin Assignment

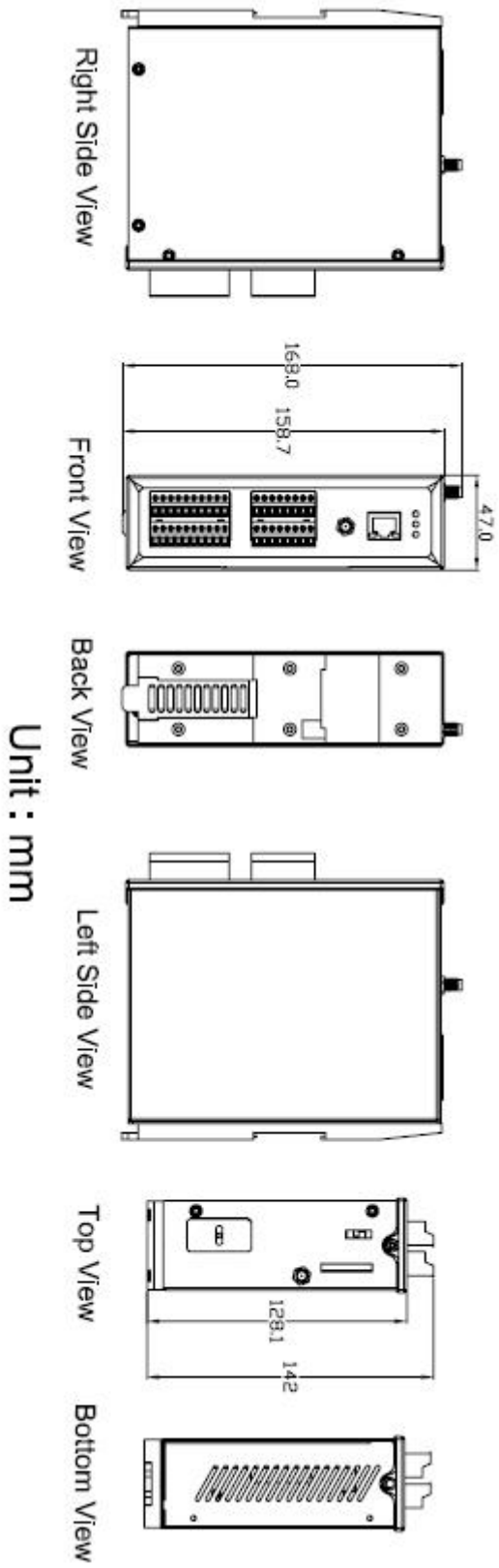
Analog Input 0 ~ +20 mA	AI7	Pin 16	Pin 8	AGND	Analog Ground
	AI6	Pin 15	Pin 7	AGND	
	AI5	Pin 14	Pin 6	AGND	
	AI4	Pin 13	Pin 5	AGND	
	AI3	Pin 12	Pin 4	DI2	Digital Input
	AI2	Pin 11	Pin 3	DI1	
	AI1	Pin 10	Pin 2	DI0	
	AI0	Pin 9	Pin 1	U2 DI	MCU2 Input

CN1 Pin Assignment

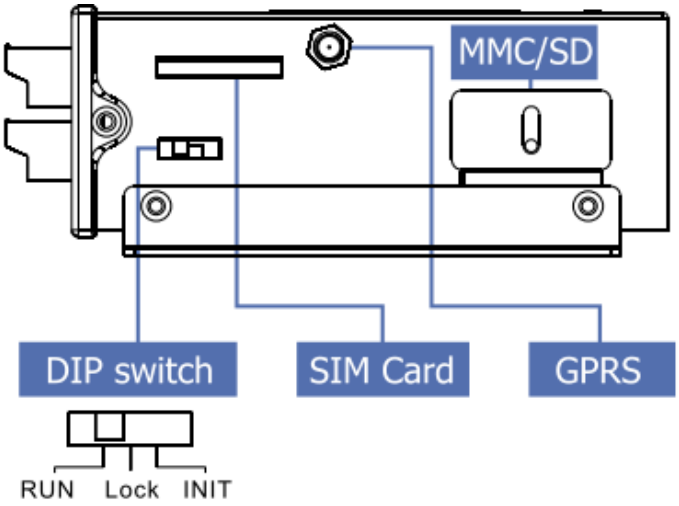
MCU2 Output	RL.NC	Pin 20	Pin 10	DO.PWR	Digital Output
	RL.COM	Pin 19	Pin 9	DO2	
	RL.NO	Pin 18	Pin 8	DO1	
COM1 RS-232	CTS1	Pin 17	Pin 7	DO0	
	RTS1	Pin 16	Pin 6	D+	COM2 RS-485
	RxD1	Pin 15	Pin 5	D-	
	TxD1	Pin 14	Pin 4	GND	Ground For COM1
Power Input: +10 ~ +30 VDC	DC.+VS	Pin 13	Pin 3	BAT.+VS	12V Lead Acid Battery
	DC.GND	Pin 12	Pin 2	BAT.+GND	
	N/A	Pin 11	Pin 1	F.G.	Frame Ground



4.2 尺寸

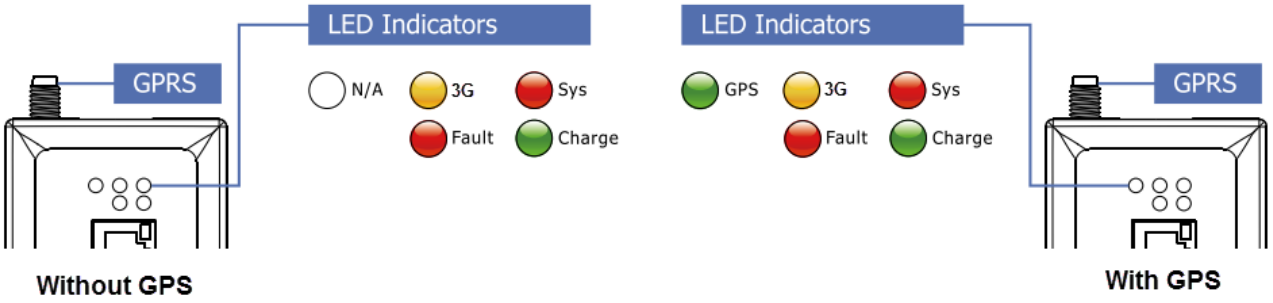

Right Side View
Front View
163.0
158.7
47.0
Back View
Left Side View
Top View
128.1
142
Bottom View
Unit : mm

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-4513 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. 3G (黃色)：可判斷 3G 模組是否正常(模組註冊後才會開始閃爍)

	3G 模組正常	3G 模組異常
2G 模式	約 2 秒閃一次	不亮或閃爍頻率不對
3G 模式	約 2 秒閃兩次	

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 秒閃爍一次

D. Charging (綠色)：顯示充電狀態

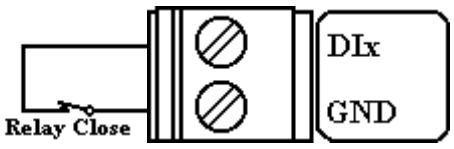
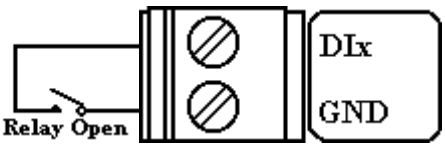
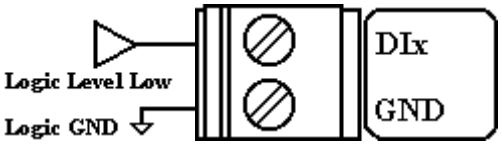
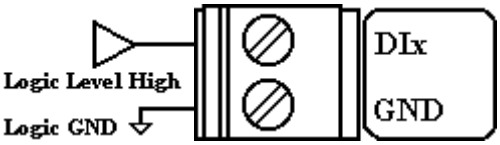
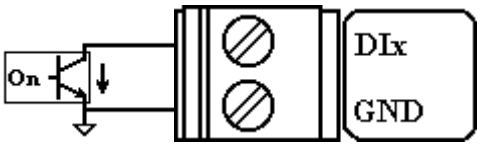
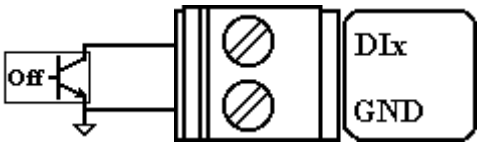
充電中	充電停止
恆亮	不亮

E. Fault (紅色)：充電錯誤指示燈

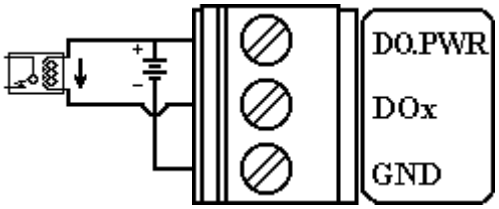
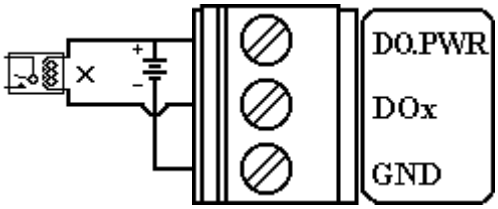
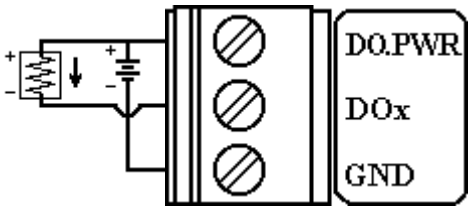
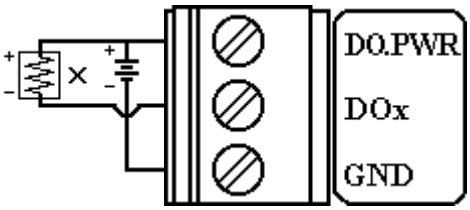
正常	錯誤
不亮	恆亮

4.5 I/O 接線圖

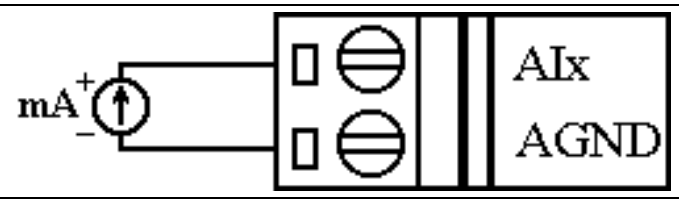
➢ DI 接線說明

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

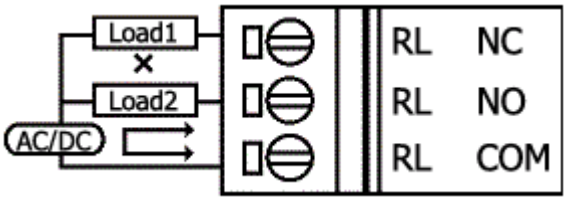
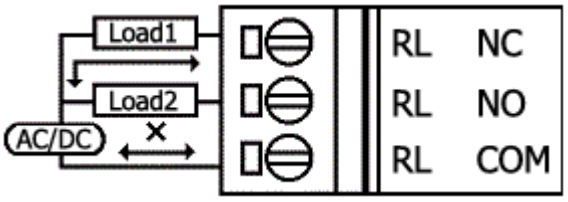
➢ DO 接線說明

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

- AI 接線說明

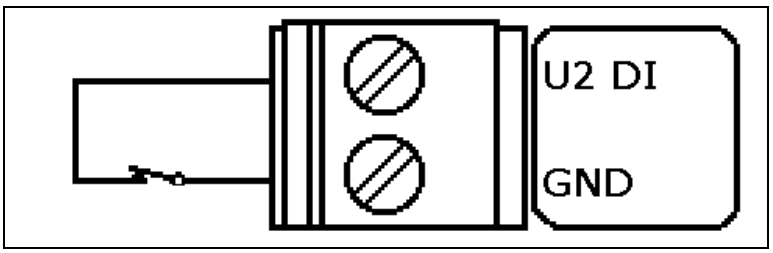
Input Type	

- Relay 接線說明

Relay Output ON	Relay Output OFF
	

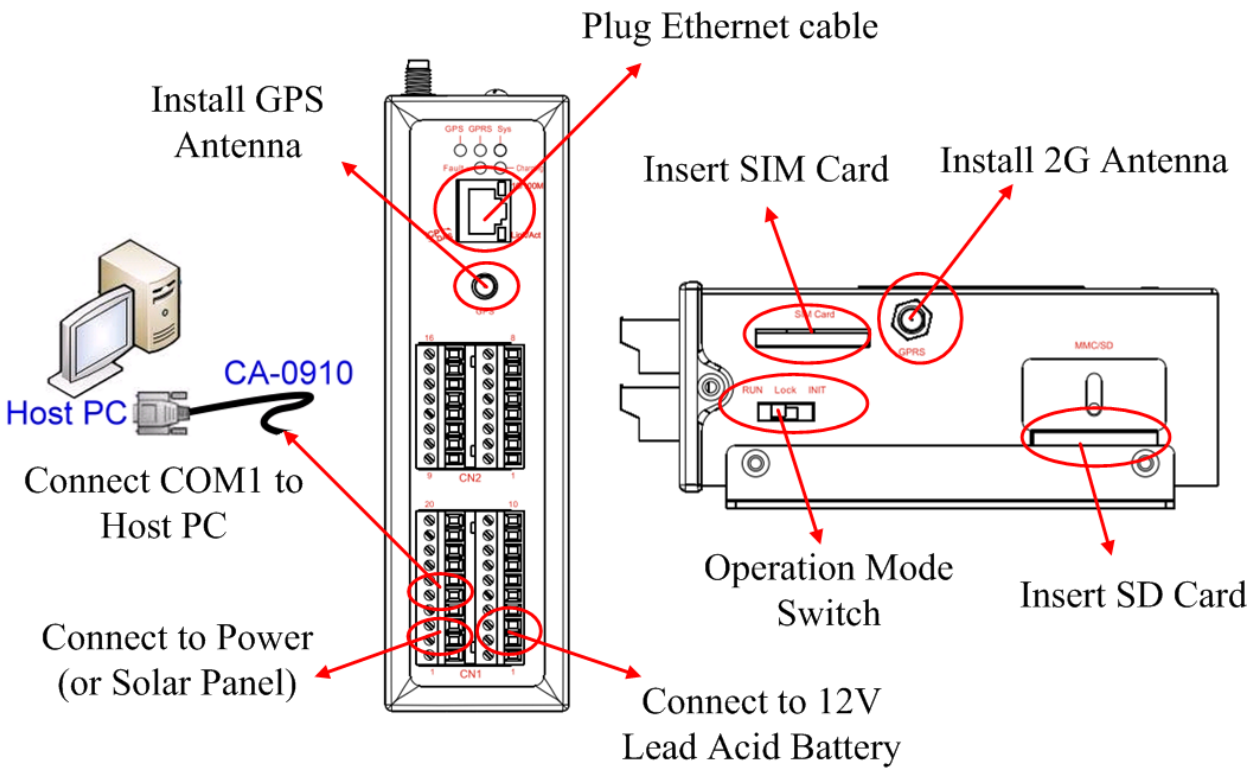
4.6 從睡眠模式中喚醒

1. 當 G-4513 處於睡眠模式時，將“U2 DI” (MCU2 DI)接至 GND，藉此將 G-4513 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-4513 系列

1. 安裝 2G/3G 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 3G 模組會持續運作

- 功耗：14~15 mA@12V
- 如何喚醒 G-4513：
  (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-4513

深層睡眠模式

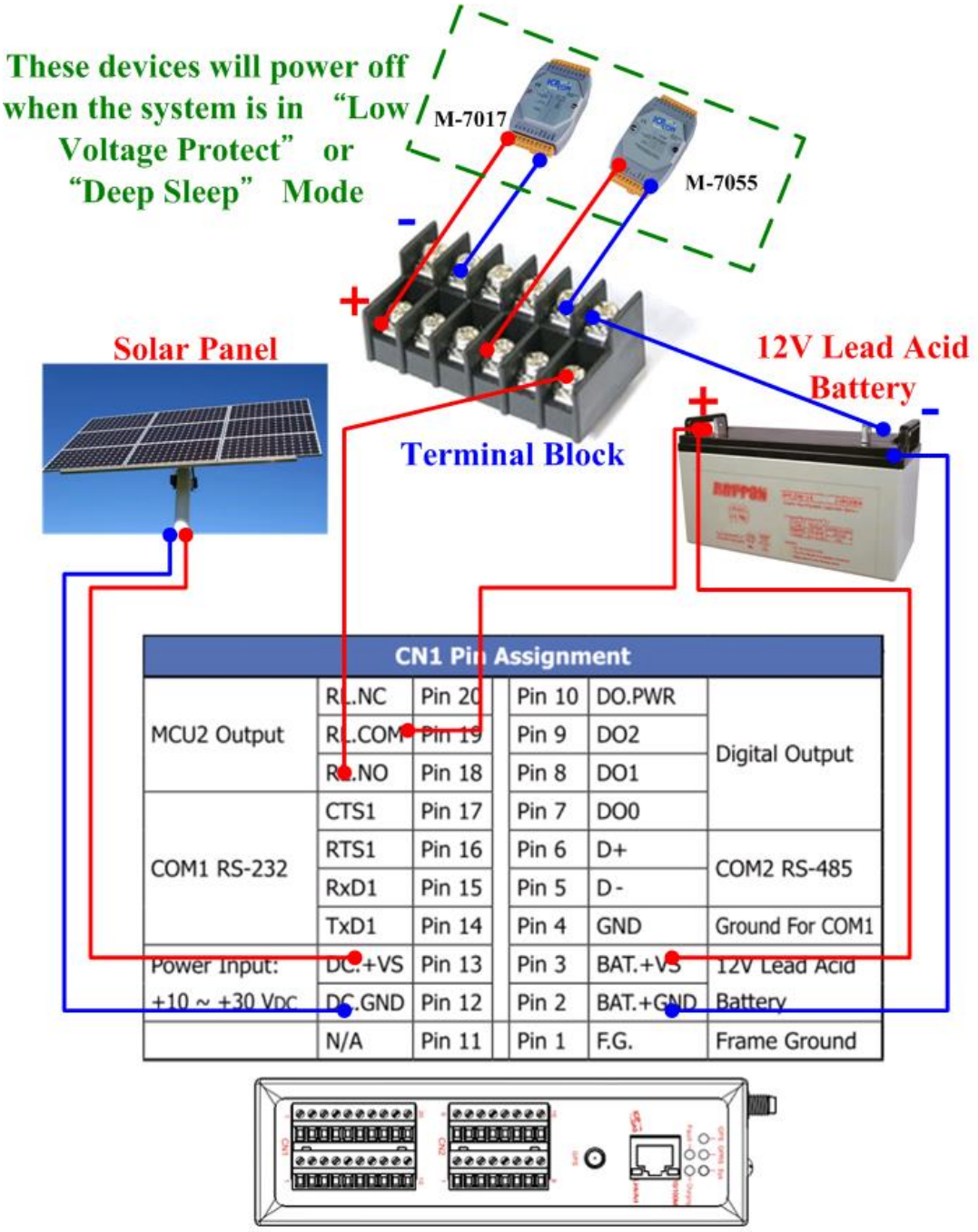
此模式會關閉所有介面，除了 MCU2 的 I/O

- 功耗：9~10 mA@12V
- 如何喚醒 G-4513：
  (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-4513 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-4513 才會醒來開始工作

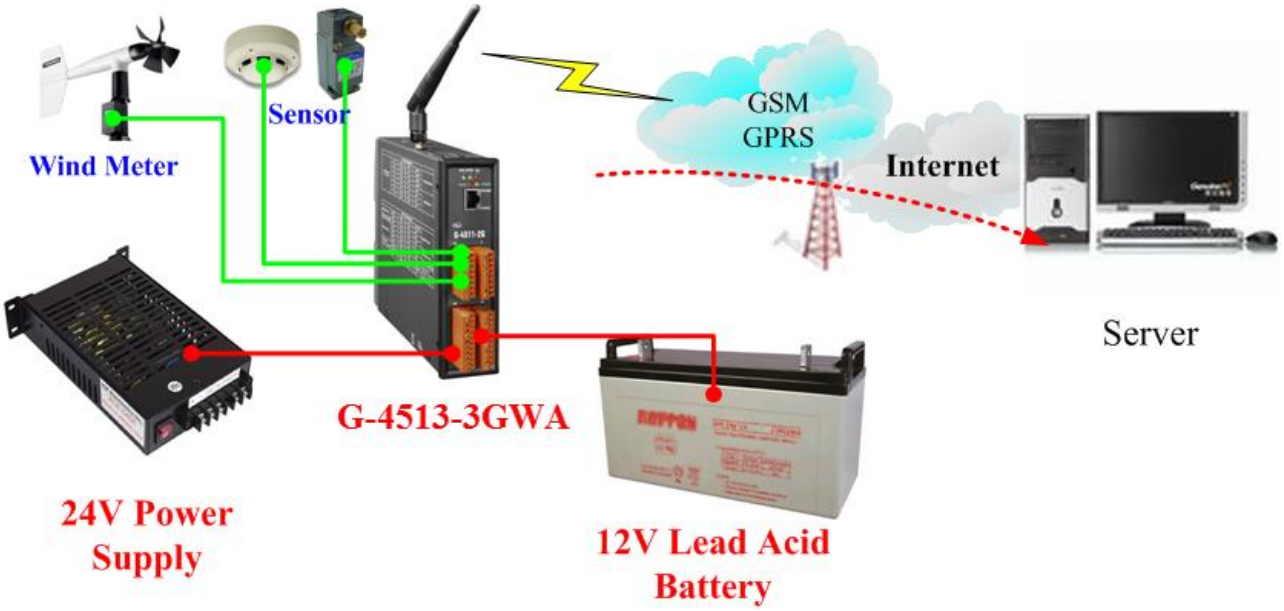
- 低電壓斷開電壓 = 11.1 V
- 低電壓重新連接電壓 = 12.6 V
- 如何使用：請參考下圖接線方式及 MCU2 的 demo 程式



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 x (1/10) + 7.2 x (9/10) = 31 (mA)
- 31 (mA) x 24 (hours) x 14 (days) = 10416 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-4513 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：
  14Ah x 90% x 1000 = 12600 mAh > 10416 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-4513 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 x (1/10) + 7.2 x (9/10) = 49 (mA)
- 424 (mA) x 24 (hours) x 14 (days) = 16430.4 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-4513 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：
  22Ah x 90% x 1000 = 19800 mAh > 16430.4 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

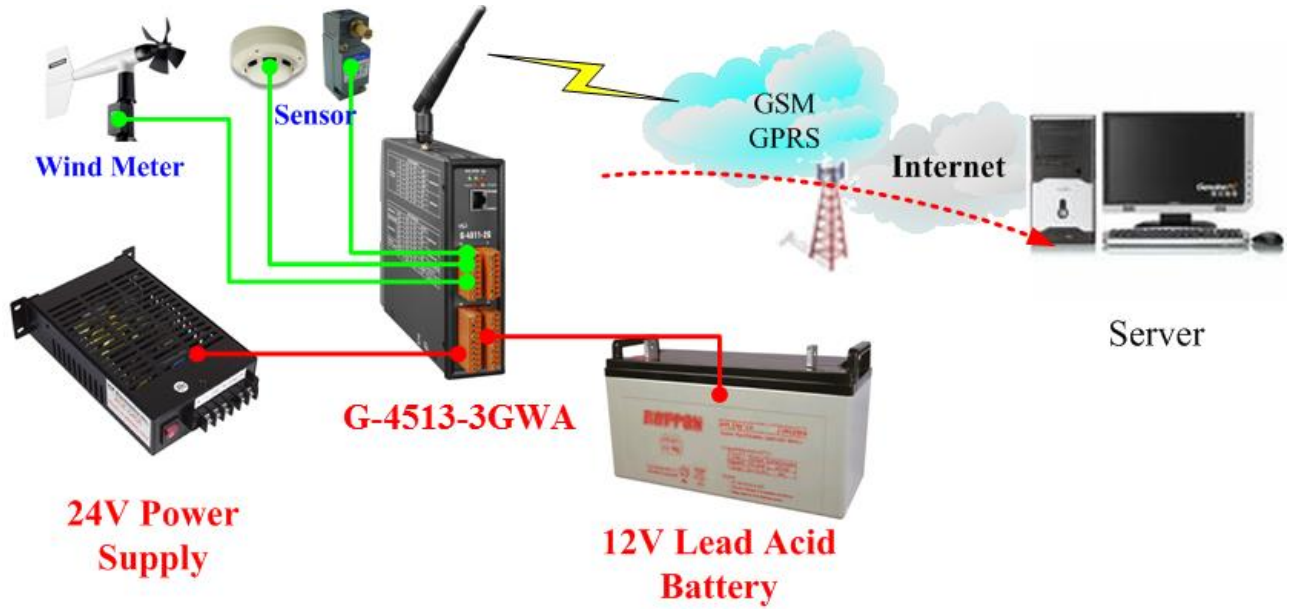
5.3 如何選擇太陽能板

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 (w) / 12 (V) = 0.833 (A) = 833 (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 mA
- 平均電流 = 416 x 4 / 24 = 69 (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

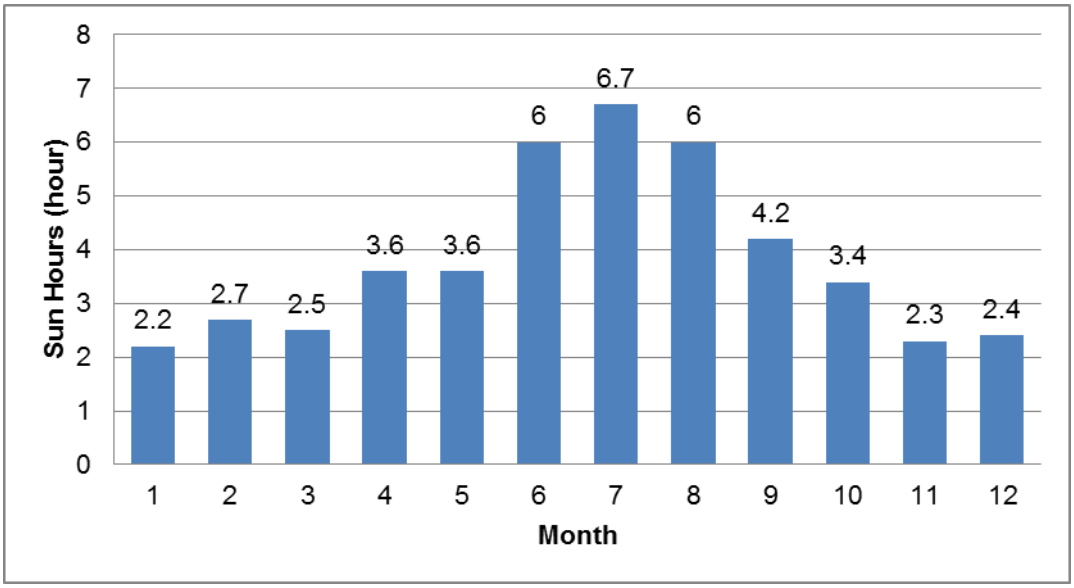
太陽能板功率 (W)	**最大電流 (mA)**	**1/2 電流 (mA)**	**日照時數**	**一天的平均電流 (mA/hr)**
10	833	416	4	69
20	1666	833	4	138
30	2500	1250	4	208
40	3333	1666	4	277
50	4166	2083	4	347

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-4513 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-4513 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-4513 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_isvol	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－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-4513-3gwa\software\demo\basic\microSD\

http://ftp.icpdas.com/pub/cd/usbcd/napdos/g-4513-3gwa/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.

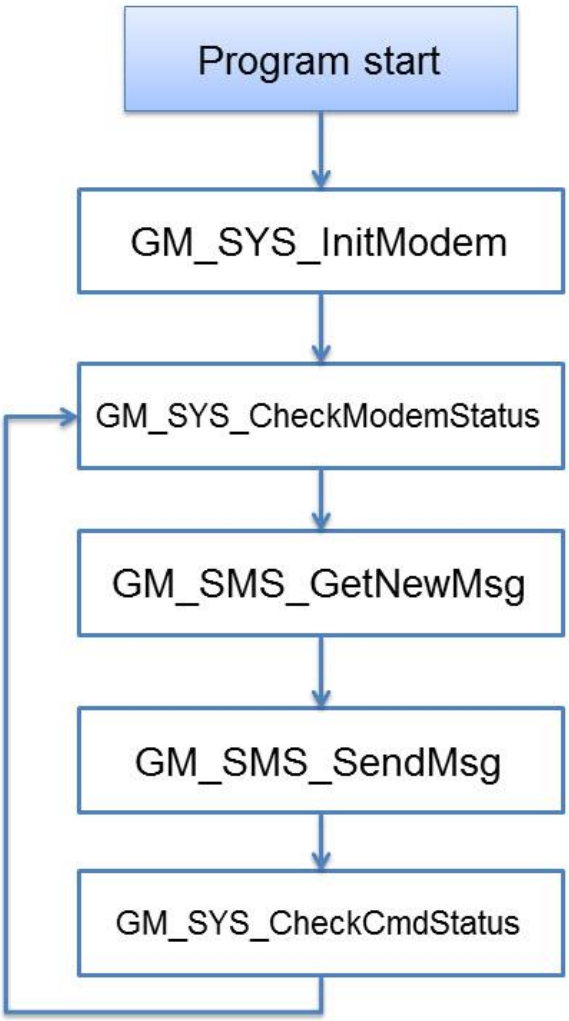
6.4 用於 GSM 的 API

更多 GSM API 相關資訊，請至官網下載、參考 GSM_U2_Library_User's Manual

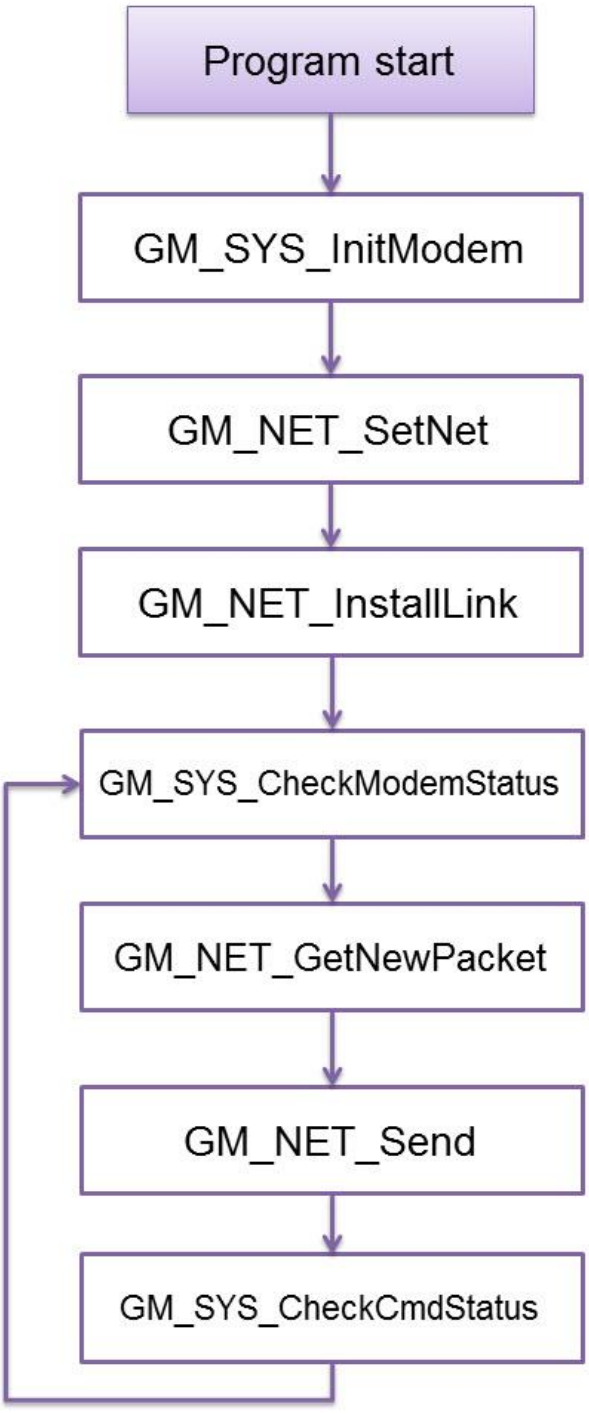
Function definition	Description
GM_SYS_InitModem	Initialize Modem
GM_SYS_CloseModem	Close the modem
GM_SYS_CheckModemStatus	Check modem status, and suggest you check it in your loop every time
GM_SYS_CheckSignal	Check signal quality
GM_SYS_CheckReg	Check register
GM_SMS_SendMsg	Send a message
GM_SMS_GetNewMsg	Get a new sms message
GM_NET_SetNet	Set Net profile data
GM_NET_InstallLink	Built TCP/UDP link
GM_NET_CloseNet	Close Network
GM_NET_CloseLink	Close client link[n]
GM_NET_Send	Send a packet
GM_NET_GetNewPacket	Get the new packet

6.4.1 GSM Design Flowchart

SMS Design Flowchart



GPRS Design Flowchart



6.4.2 GM_SYS_InitModem

Initialize Modem.

**must use GM_SYS_CheckModemStatus() to check modem status later

Syntax

```
int GM_SYS_InitModem(
    SYSProfile sysProfile
);
```

Parameters

sysProfile
set system profile

Return values

GM_NOERROR：success
GM_COMERROR：comport error
GM_INITERROR：init fail error

6.4.3 GM_SYS_CloseModem

Close the modem.

**Please call GM_SYS_InitModem() to wake up modem after using GM_SYS_CloseModem(1) to shut down the modem.

Syntax

```
int GM_SYS_CloseModem(
    int mode
);
```

Parameters

mode

0：close modem, but maintain it power on

1：close modem and set it power off

Return values

GM_NOERROR：no error

GM_CMDERROR：command error

6.4.4 GM_SYS_CheckModemStatus

Check modem status, and suggest you check it in your loop every time.

Syntax

```
int GM_SYS_CheckModemStatus(void);
```

Parameters

None

Return values

GM_NOERROR：modem register success, can service
GM_NOREG：modem not registered, can't service

6.4.5 GM_SYS_CheckSignal

Check signal quality.

Syntax

```
int GM_SYS_CheckSignal(void);
```

Parameters

None

Return values

signal quality

0	-113 dBm or less
1	-111 dBm
2...30	-109... -53 dBm
31	-51 dBm or greater

6.4.6 GM_SYS_CheckReg

Check register.

Syntax

```
int GM_SYS_CheckReg(void);
```

Parameters

None

Return values

Register flag

0：not registered
1：registered, home network
2：not registered, and searching...
3：registration denied
4：unknown
5：registered, roaming

6.4.7 GM_SMS_SendMsg

Send a message.

**must use "GM_SYS_CheckCmdStatus()" to check status later

Syntax

```
int GM_SMS_SendMsg(
    strEncode_Msg* strMsg
);
```

Parameters

strMsg
the message that will be sent.

Return values

GM_NOERROR：no error
GM_NOREG：not registered, or can't service
GM_BUSY：modem busy

6.4.8 GM_SMS_GetNewMsg

Get a new sms message.

Syntax

```
int GM_SMS_GetNewMsg(
    strEncode_Msg* msg
);
```

Parameters

msg
new sms message

Return values

0：no new message
1：new message coming

6.4.9 GM_NET_SetNet

Set Net profile data.

Syntax

```
int GM_NET_SetNet(
    NetProfile netProfile
);
```

Parameters

netProfile
Net profile data

Return values

GM_NOERROR：no error
GM_CMDERROR：command error

6.4.10 GM_NET_InstallLink

Built TCP/UDP link.

Syntax

```
int GM_NET_InstallLink(
    int n,
    int tcp,
    char* serverIP,
    unsigned int serverPort
);
```

Parameters

n

link number (0~6)
3G (G-4513 series)：0~6
2G (G-4511 series)：0

tcp

client type, tcp=1 for TCP client ; tcp=0 for UDP client

serverIP

IP or Domain name of the server, ex: "61.111.222.333", "test.com.tw"

serverPort

TCP/UDP Port of the server (1~65535), ex: 1234

Return

GM_NOERROR：correct parameter to install TCP/UDP link
GM_CMDERROR：command error

6.4.11 GM_NET_CloseNet

Close Network.

Syntax

```
int GM_NET_CloseNet(void);
```

Parameters

None

Return values

GM_NOERROR：no error
GM_CMDERROR：command error
GM_BUSY：modem busy

6.4.12 GM_NET_CloseLink

Close client link[n].

Syntax

```
int GM_NET_CloseLink(
    int n
);
```

Parameters

n

3G (G-4513 series)：0~6
2G (G-4511 series)：0

Return values

GM_NOERROR：no error
GM_CMDERROR：command error
GM_BUSY：modem busy

6.4.13 GM_NET_Send

Send a packet.

**must use "GM_SYS_CheckCmdStatus()" to check status later

Syntax

```
int GM_NET_Send(
    char link,
    char* data,
    int dataLen
);
```

Parameters

link

link number
3G (G-4513 series)：0~6
2G (G-4511 series)：0

data

data that will be sent

dataLen

data length, Max.=1000

Return values

GM_NOERROR：no error
GM_CMDERROR：command error
GM_BUSY：modem busy

6.4.14 GM_NET_GetNewPacket

Get the new packet.

Syntax

```
int GM_NET_GetNewPacket(
    GPRSData* gprsData
);
```

Parameters

gprsData
new data packet

Return values

0：no new packet
1：new packet coming

6.4.15 GSM Demo References

For example, send and receive sms message

```
#include <conio.h>
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include "../lib/G4500.h"
#include "../lib/GSM_U2.h"
#include "../lib/OS7_COM.h"
#include "../lib/MCU2LIB.h"

int main(void)
{
	int iAction=1, quit=1;
	int i, j, tmp;
	int Result=0;
	strEncode_Msg RecMsg, SendMsg;
	char sendNumber[20];
	int send_n;
	int sendStatus = 0;
	SYSProfile sysProfile;

	//-- init
	InitLib();

	/*---- init modem----*/
	strcpy(sysProfile.PINCode, "0000"); /*The pin code of SIM card, ex: "0000"*/
	sysProfile.modemPort = 4;   /*modem port number. G-4500 = 4, uP-5000 = 11*/
	sysProfile.hardware = 0;       /*hardware type. 1: G-4500, 2: uPAC-5000, 0: Other*/
	GM_SYS_SetPowerFunction(powerFunction); /*set power-control function*/


	if((Result = GM_SYS_InitModem(sysProfile)) == GM_NOERROR)
		Print("init_modem success!!\r\n");
	else
	{
		Print("init_modem fail!! return value is %d\r\n", Result);
```

```
        return 1;
}

/*--check Could the modem service? --*/
while(GM_SYS_CheckModemStatus() != GM_NOERROR)
{
        Print("wait modem register...\r\n");
        DelayMs(1000);
}
Print("modem registered!!\r\n");

while(iAction!=0)
{
        iAction=0;
        quit=0;
        Print("1) Send ASCII messages\r\n");
        Print("2) check signal quality\r\n");
        Print("3) check registered?\r\n");
        Print("0) Quits demo program\r\n");
        Print("Choose an option and press [Enter]: ");
        Scanf("%d", &iAction);

        switch(iAction)
        {
        case 1:       /*Send ASCII messages*/
                Print("Send ASCII messages\r\n");

                Print("Please Input Phone Number =");
                Scanf("%s", sendNumber);

                Print("How many message do you want to send?\r\n");
                Scanf("%d", &send_n);
                i = 0;
                Print("== start to send sms, please press <ESC> to exit ==\r\n");
                while(1)
                {
                        /* press "ESC" to exit */
                        if(Kbhit())
                        {
                                tmp = Getch();
```

```
            if( tmp == 27 || tmp == 'q')
                break;
        }

/*--(1) check that could the modem service, if it can't, skip operating the modem below --*/
        if(GM_SYS_CheckModemStatus() != GM_NOERROR)
            continue;

        /*--(2) send messages, when the modem can service--*/
        if(i<send_n)
        {
            switch((sendStatus=GM_SYS_CheckCmdStatus()))
            {
            case GM_READY:
                Print("sending message(%d)....\r\n", i);

                strcpy(SendMsg.phoneNumber, sendNumber);
                SendMsg.mode = GSM_7BIT;
                sprintf(SendMsg.msg, "GSM_Test(%2d)", i);
                SendMsg.dataLen = strlen(SendMsg.msg);
                GM_SMS_SendMsg(SendMsg);
                break;
            case GM_NOERROR:
                Print("send success!!\r\n");
                i++;
                break;
            case GM_BUSY:  //sending, and waiting reply
                break;
            default:
                Print("send error, and skip this one, error code=%d\r\n",
                sendStatus);
                i++;
                break;
            }
        }
        /*--(3) if any sms message come in, print it --*/
        if( GM_SMS_GetNewMsg(&RecMsg) != 0)
        {
            printMsg(RecMsg);
        }
```

```
                }
                break;

        case 2://check signal quality
                Result = GM_SYS_CheckSignal();
                Print("signal value = %d\r\n", Result);
                break;

        case 3://check register value
                Result = GM_SYS_CheckReg();
                Print("register value = %d    (0:no register, 1:registered, 2:registering)\r\n",
                Result);
                break;

        case 0:
        default:
                quit=1;
                break;

        }//end switch()

        if(!quit)
        {
                Print("Press any key to continue...\r\n");
                Getch();
        }
    }   /*end while(1)*/

    /*must close before program ending to release you resource*/
    /*-- Close the modem, 0:not turn off modem, 1:turn off modem*/
    GM_SYS_CloseModem(0);

    Print("Please press ENTER to exit...\r\n");
    Getch();
    return 0;
}
```

For example, TCP client Demo

```
#include <conio.h>
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include "../lib/G4500.h"
#include "../lib/GSM_U2.h"
#include "../lib/OS7_COM.h"
#include "../lib/MCU2LIB.h"

int main(void)
{
   int iAction=1, quit=1;
   int i, j,tmp;
   int Result=0;
   int send_n;
   NetProfile netProfile;
   SYSProfile sysProfile;
   GPRSData gprsData;
   char serverIP[16];
   int serverPort;
   long socket_n;
   int netSendStatus = 0;
   char myIP[16];

   InitLib();

   /*---- init modem ----*/
   strcpy(sysProfile.PINCode, "0000");     /*The pin code of SIM card, ex: "0000"*/
   sysProfile.modemPort = 4;     /*modem port number. G-4500 = 4, uP-5000 = 11*/
   sysProfile.hardware = 0;        /*hardware type. 1: G-4500, 2: uPAC-5000, 0: Other*/
   GM_SYS_SetPowerFunction(powerFunction); /*set power-control function*/

   if( (Result = GM_SYS_InitModem(sysProfile)) == GM_NOERROR)
         Print("init_modem success!!\r\n");
   else{
         Print("init_modem fail!! return value is %d\r\n", Result);
         return 1;
```

```
}

/*-- check Could the modem service? --*/
while(GM_SYS_CheckModemStatus() != GM_NOERROR)
{
    Print("wait modem register...\r\n");
    DelayMs(1000);
}
Print("modem registered!!\r\n");

while(iAction!=0)
{
    iAction=0;
    quit=0;
    Print("1) TCP client demo\r\n");
    Print("0) Quits demo program\r\n");
    Print("Choose an option and press [Enter]: ");
    Scanf("%d", &iAction);
    Print("\r\n");

    switch(iAction)
    {
      case 1:    //TCP client demo
      Print("TCP client demo start\r\n");

        /* set Network profile */
        /* APN for network provided by your cellular provider*/
        strcpy(netProfile.APN, "INTERNET");
        /*username for network provided by your cellular provider */
        strcpy(netProfile.pw, "guest");
        /*password for network provided by your cellular provider */
        strcpy(netProfile.user, "guest");
        /* The most basic task of DNS is to translate hostnames such as
        www.icpdas.com to IP address such as 96.9.41.131 */.
        strcpy(netProfile.DnsServerIP, "");       /*empty string = system default value*/
        GM_NET_SetNet(netProfile);

        /*set ip, port of server */
        Print("please input server IP:(ex: 74.125.227.48)\r\n");
        Scanf("%s", serverIP);
```

```
            Print("please input server Port:(ex: 80)\r\n");
            Scanf("%d", &serverPort);

            /*--(1) install link[0], GM_NET_InstallLink(0, 0, serverIP, serverPort) for UDP--*/
            GM_NET_InstallLink(0, 1, serverIP, serverPort);
            Print("linking...\r\n");
            socket_n = 0;    //count for the packets

            while (1)
            {
                /* press "ESC" to exit */
                if(Kbhit())
                {
                    tmp = Getch();
                    if( tmp == 27 || tmp == 'q')
                        break;
                }

/*-- (2) check that could the modem service, if it can't, skip operating the modem below --*/
                if(GM_SYS_CheckModemStatus() != GM_NOERROR)
                    continue;

                if(GM_NET_GetLinkStatus(0)!=1)
                    continue;
                else
                    GM_NET_GetIP(myIP);

                /*--(3) send the data to server, and when LinkStatus[1]=1 --*/
                switch((netSendStatus=GM_SYS_CheckCmdStatus()))
                {
                  case GM_READY:
                    Print("sending package[%8ld]..., myIP = %s\r\n", socket_n, myIP);
                    gprsData.link = 0;
                    sprintf(gprsData.data, "-<%8ld>-TCP send test!!", socket_n);

                    gprsData.dataLen = strlen(gprsData.data);
                    if(GM_NET_Send(gprsData.link,gprsData.data,gprsData.dataLen)!=G
                    M_NOERROR)
                        Print("can't send package[%8ld]\r\n");
                    break;
```

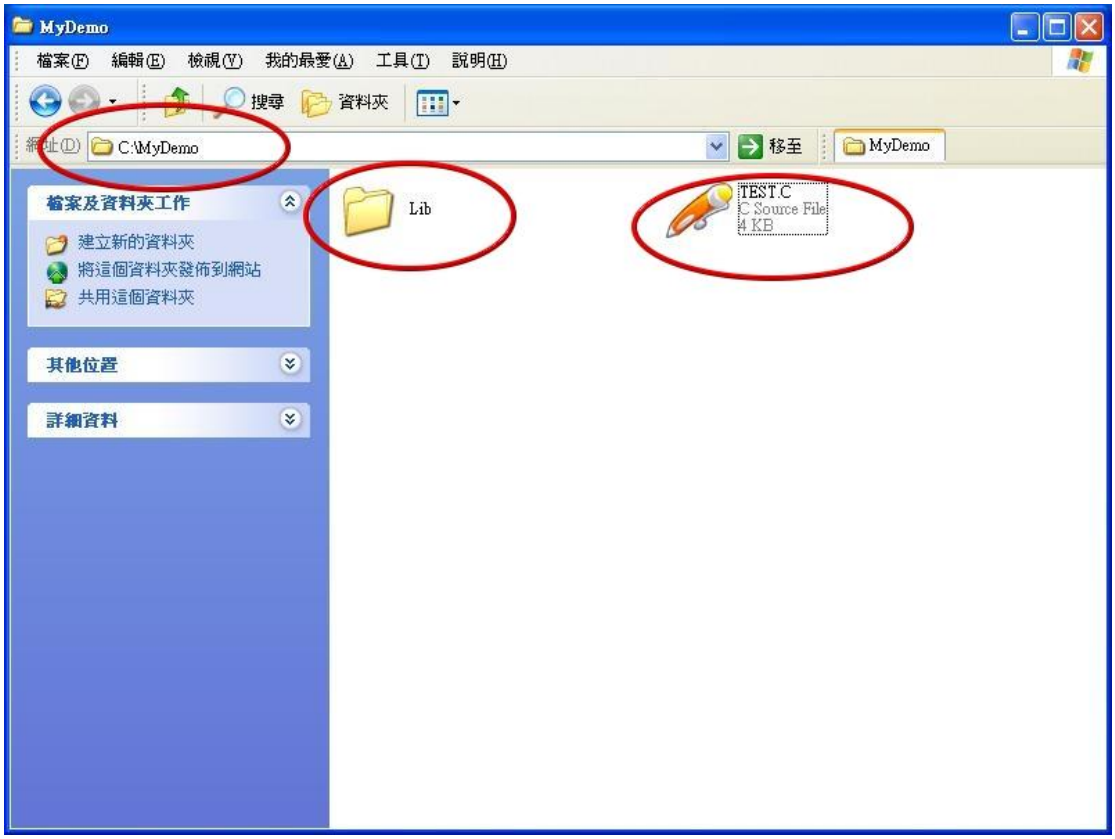
```
                    case GM_NOERROR:
                        Print("send success!!\r\n");
                        socket_n++;
                        break;
                    case GM_BUSY:  /*sending, and waiting reply*/
                        break;
                    default:
                        Print("send error, and re-send again, error code=%d\r\n",
                        netSendStatus);
                        break;
            }

            /*-- (4) if any new data packet come in, print it --*/
                if(GM_NET_GetNewPacket(&gprsData) != NULL)
                {
                    Print("\n== new data packet come in\r\n");
                    printPacket(gprsData);
                }
        }
        GM_NET_CloseLink(0); /*--Close client link[n], 3G:0~6, 2G:0 --*/
        DelayMs(1000);
        GM_NET_CloseNet();
        DelayMs(1000);

        break;
    }

}
```

7. 程式編譯與下載步驟

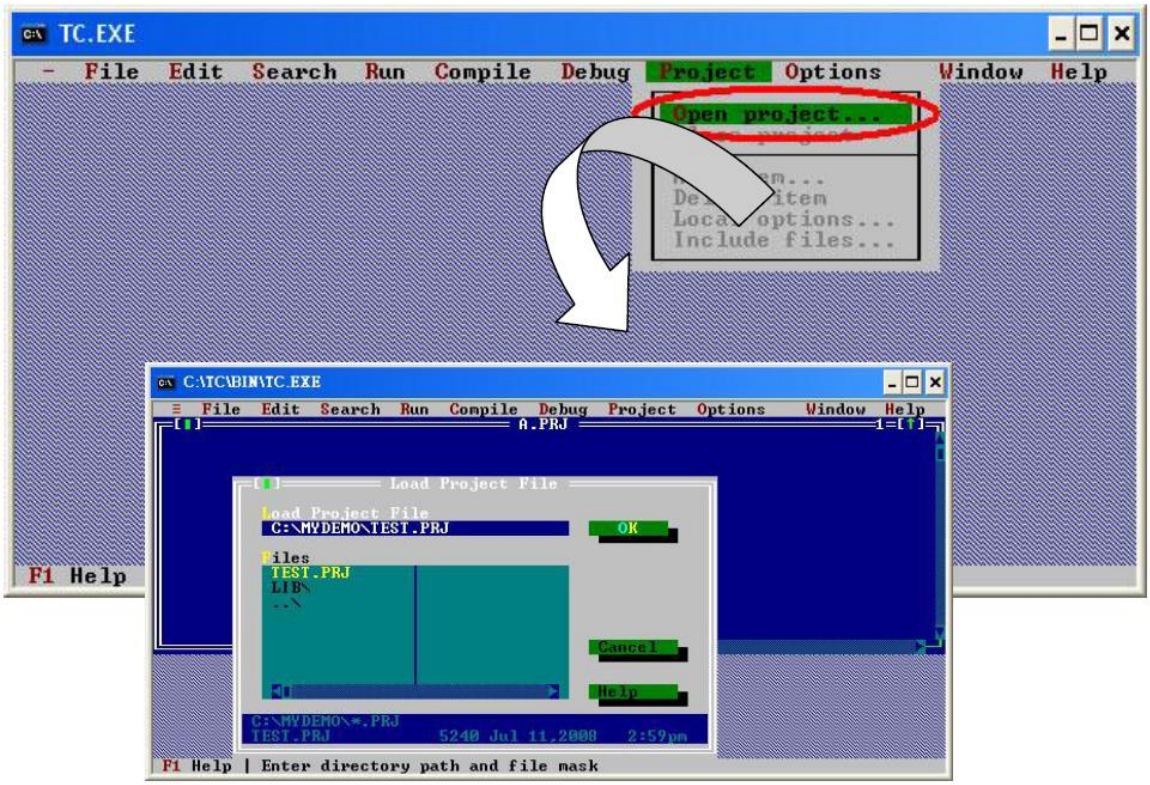
此章節將介紹如何編譯與執行 G-4513 的程式

Library	Description
G4500.LIB	G-4513 and DI/O、AI functions
GSM_U2.LIB	GPRS functions
SD_Vnnn.LIB	MMC/SD functions
TCP_DM32.LIB	Ethernet functions
LCD.LIB	LCD functions

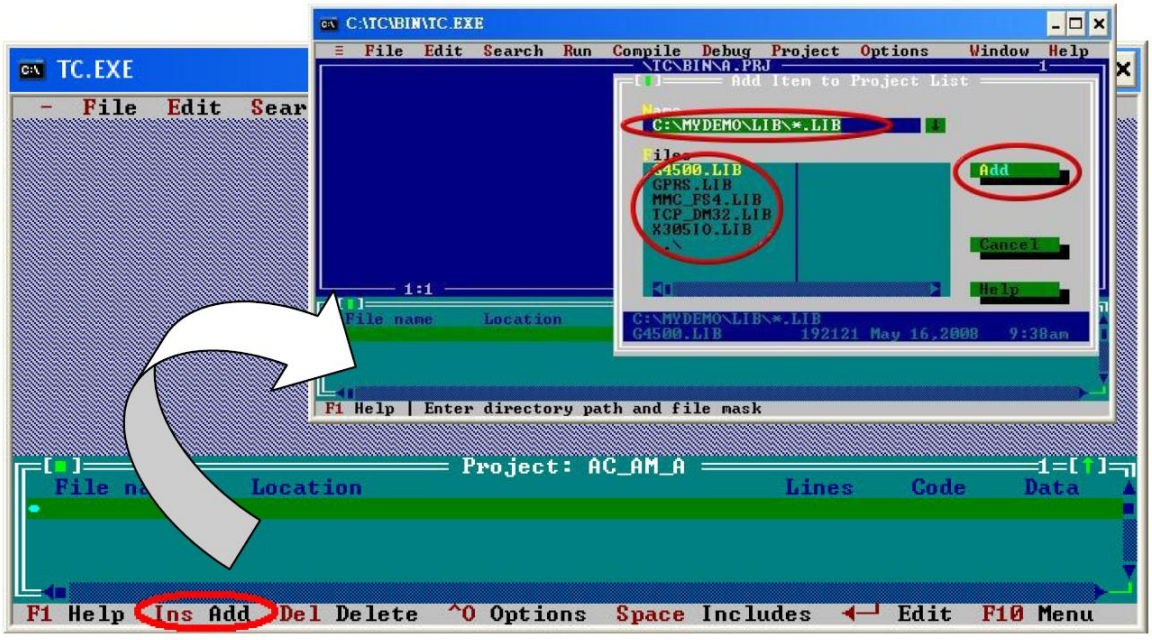
步驟 1：在 C 槽建立資料夾並取名為“MyDemo”，複製 lib 及程式到 MyDemo 資料夾內



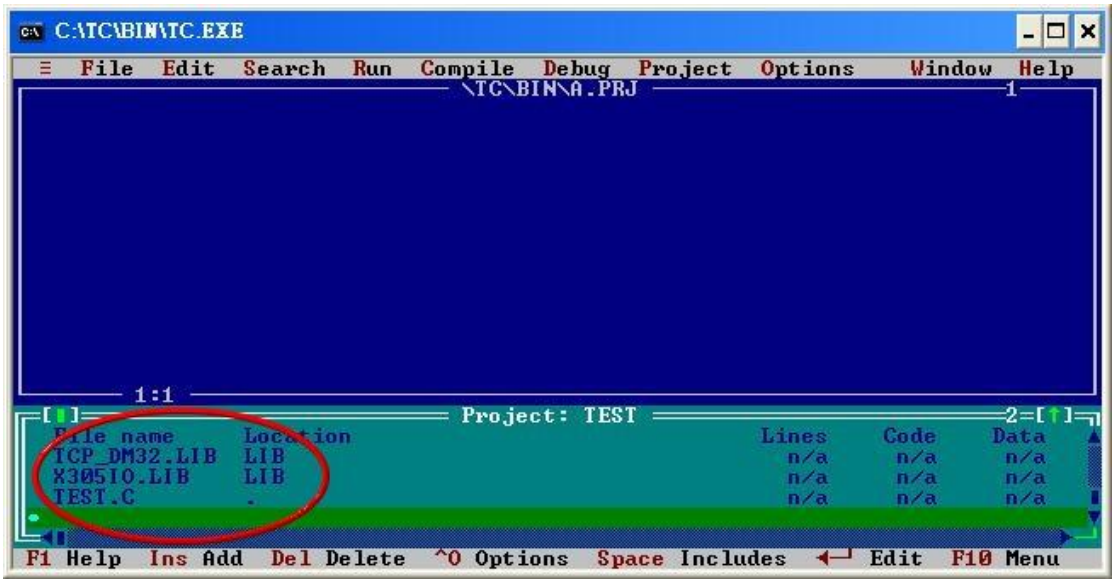
步驟 2：執行 TC++ 1.0，按下 “Project\Open project…” 建立新的專案檔並取名為 “TEST.PRJ”



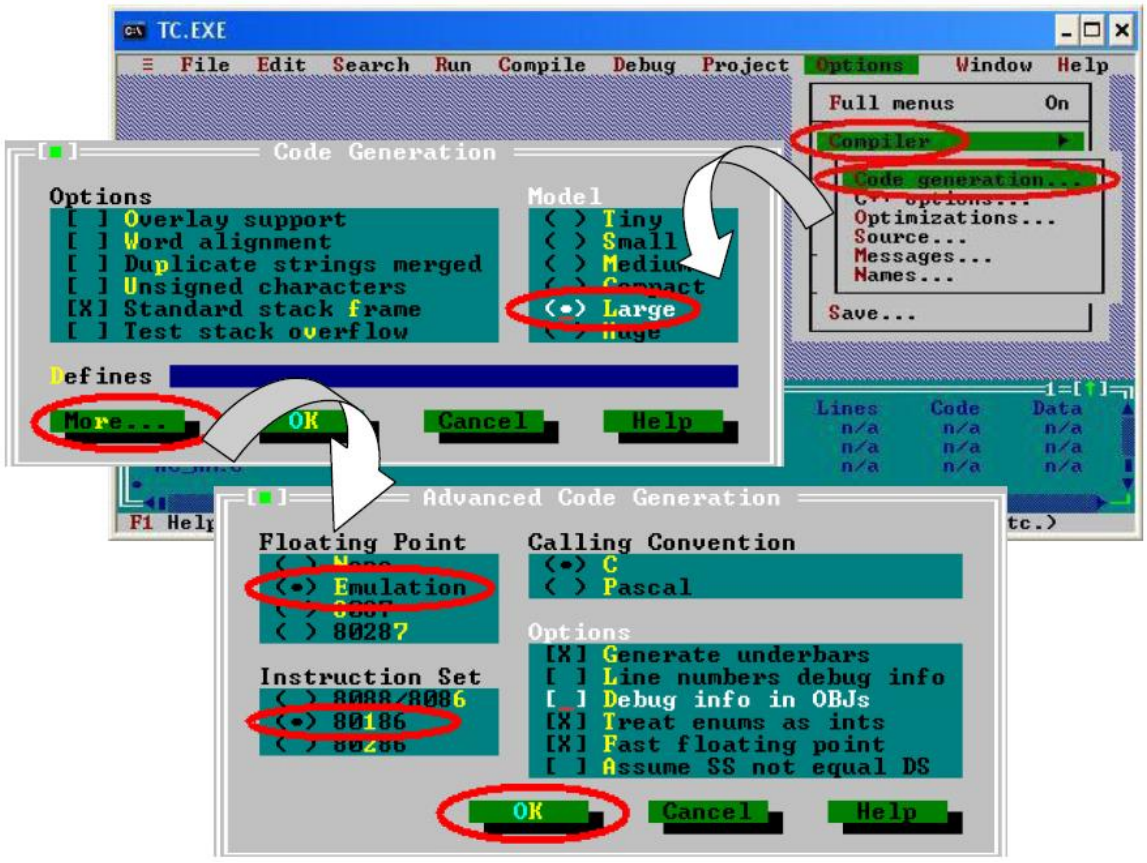
步驟 3：使用 “Add” 功能加入編譯需用到的函式庫檔案到 MyDemo 資料夾



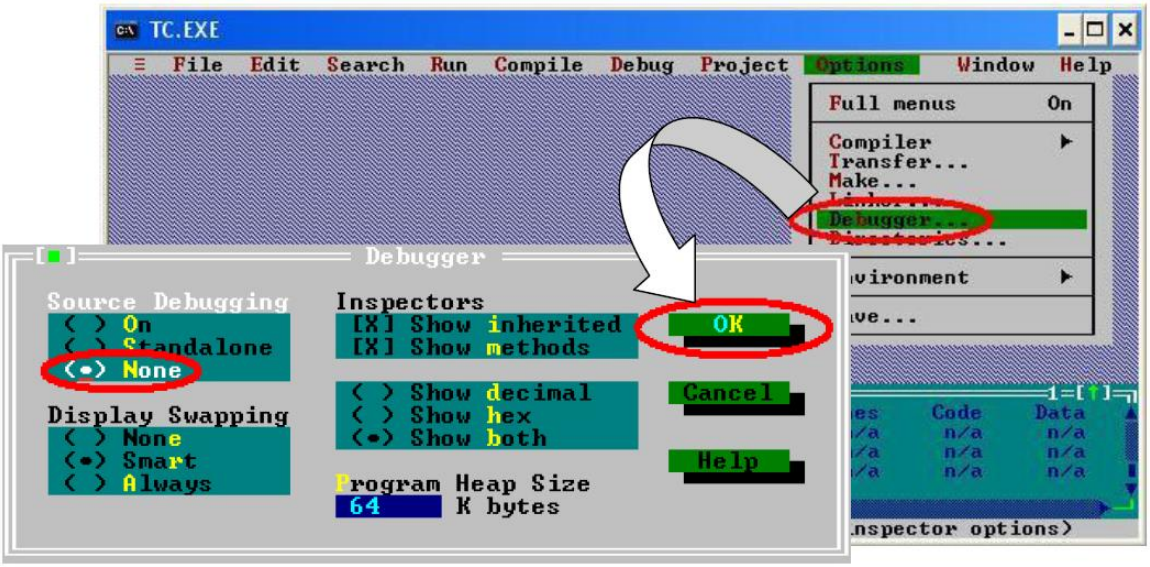
步驟 4：參照步驟 3，加入其它函式庫檔案及 TEST.c 到 MyDemo 資料夾



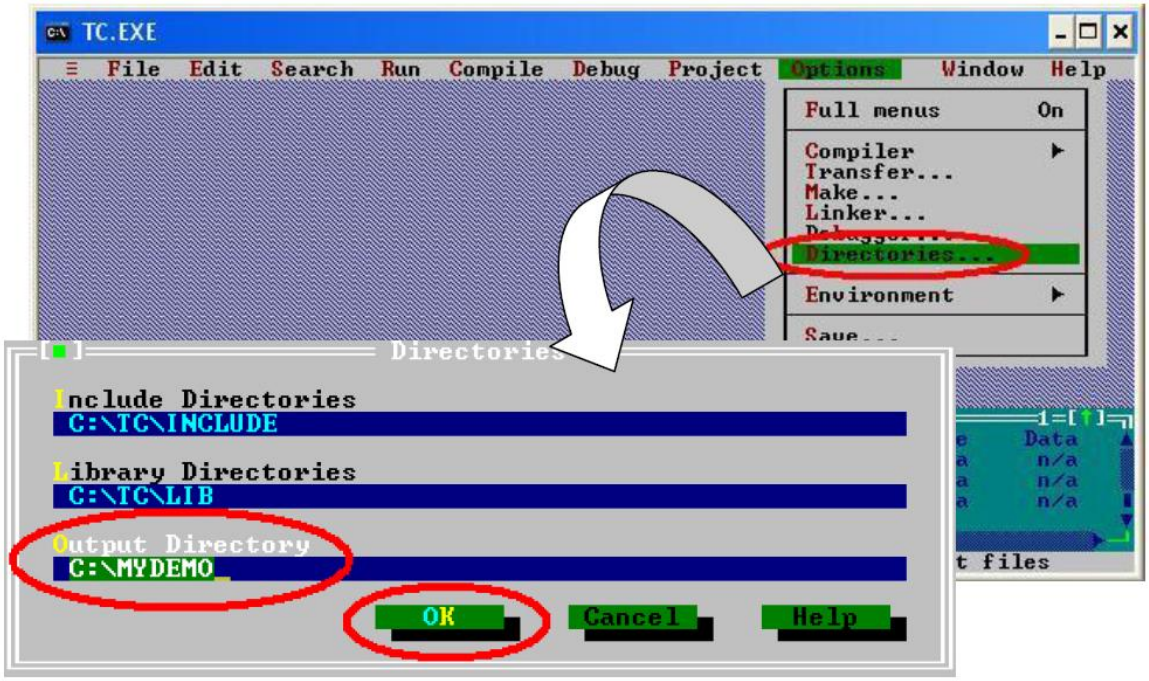
步驟 5：點選 “Options/Compiler/Code generation…” 將編譯模式設為 Large，接著點選 “More…” 分別設定 “Floating point” 和 “Instruction Set” 參數為 Emulation 和 80186。設定完後按 OK 儲存設定。



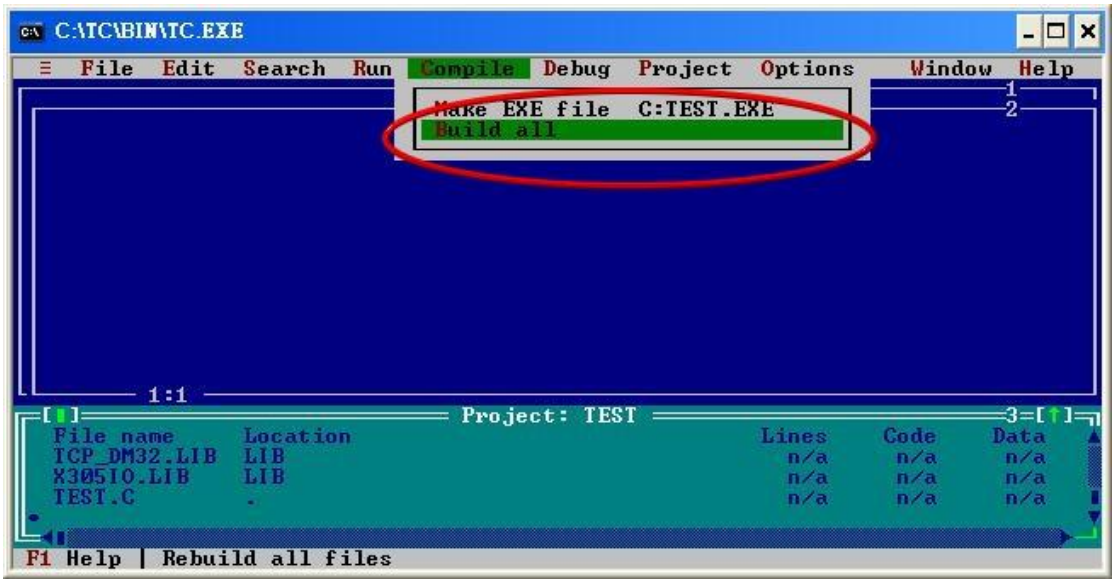
步驟 6：點選“Option/Debugger... ”將“Source Debugging”參數設定為“None”



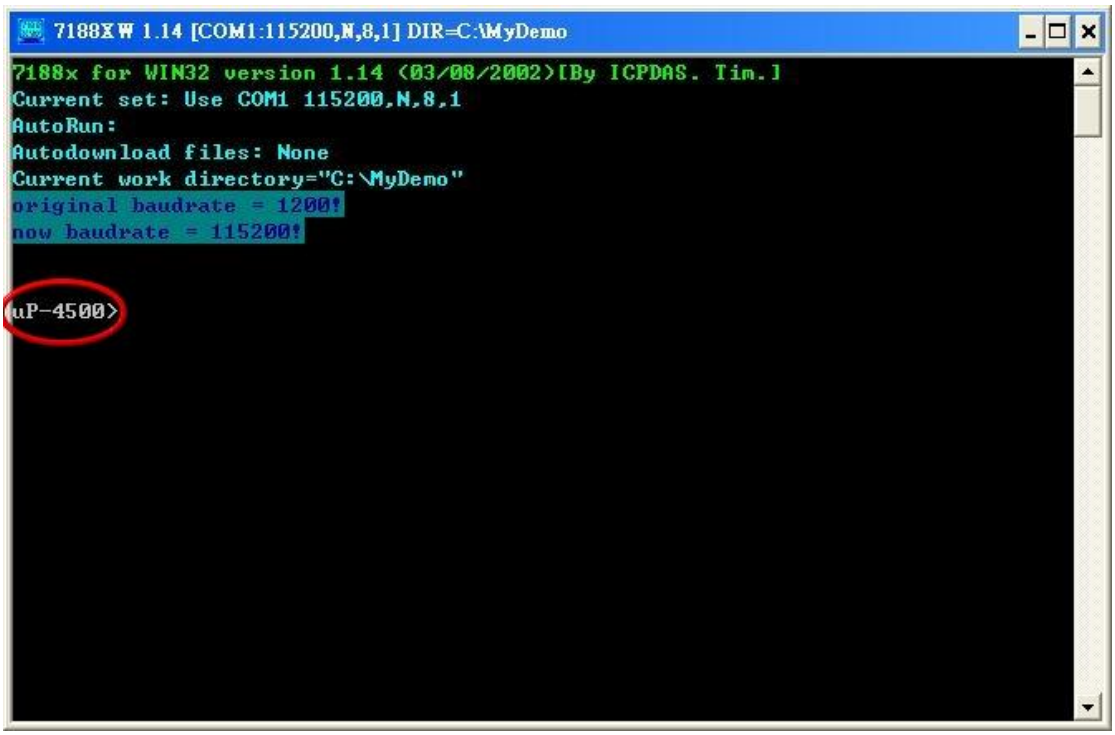
步驟 7：點選“Option/Directories... ”設定“Output Directory”輸出目錄為“C:\MyDemo”



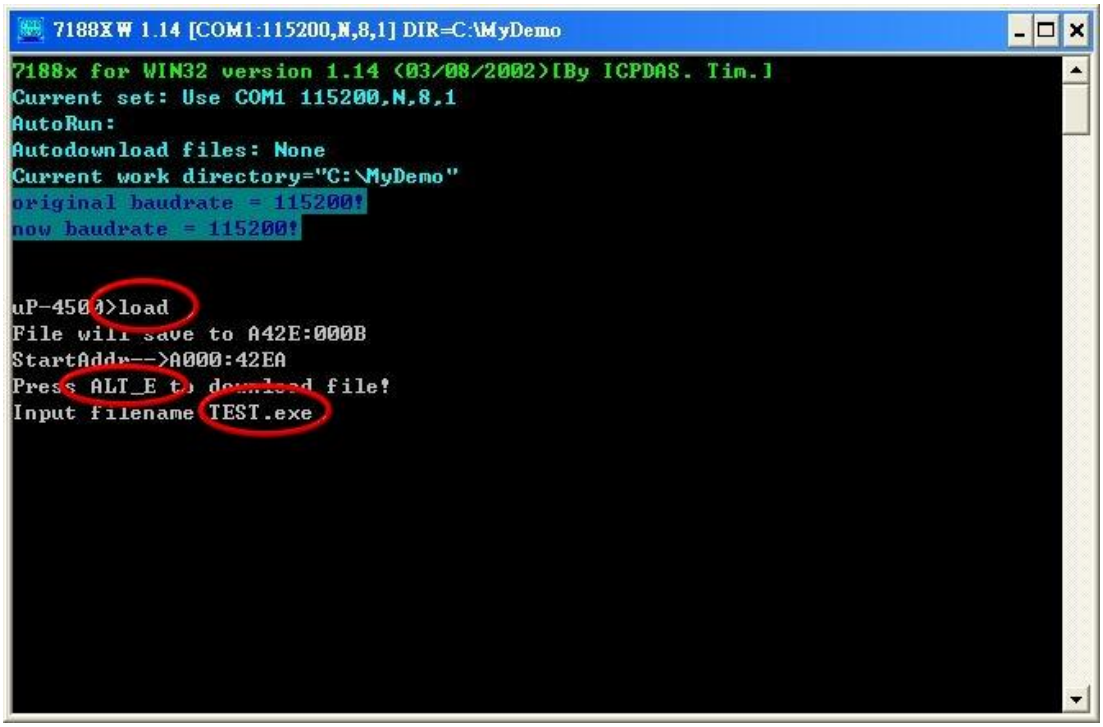
步驟 8：設定完所有參數後，按下“Compile/build all”產生執行檔“TEST.exe”



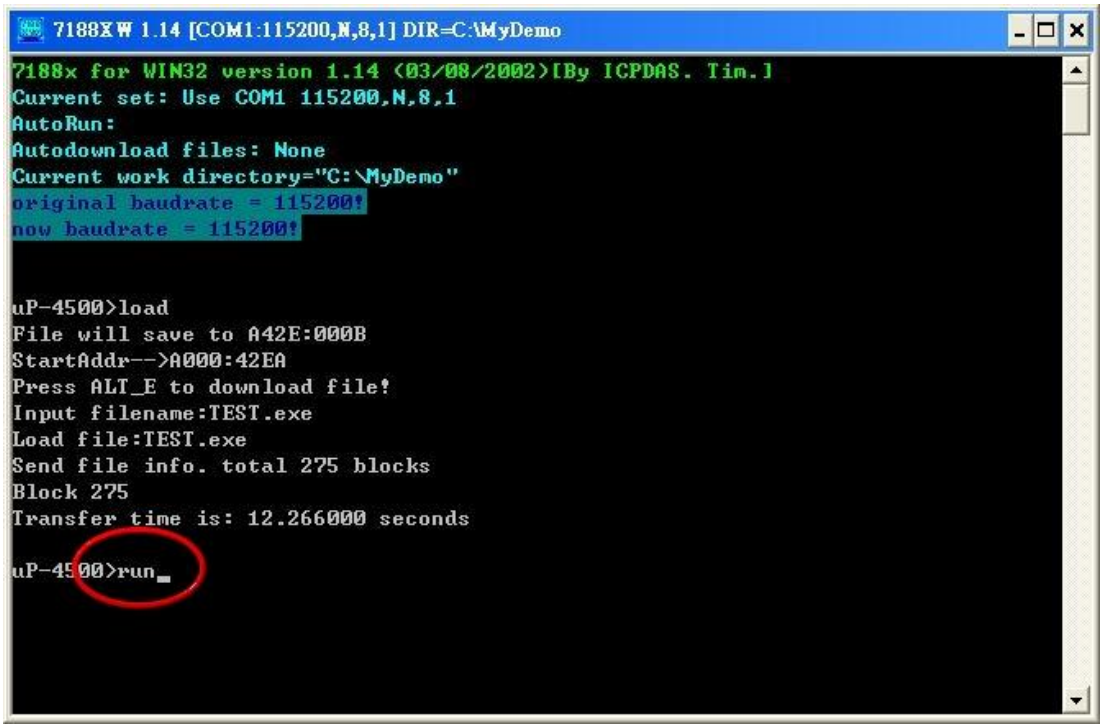
步驟 9：複製 7188XW.exe 到 MyDemo 目錄，然後雙擊左鍵執行 7188XW.exe，並將 G-4513 系列的 COM1 連接至 PC 的 RS-232



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



步驟 11：下載完成後，輸入“run”開始執行“TEST.exe”



8. 版本記錄

版本	日期	作者	說明
1.0.0	2014/09/01	William	第一版
1.0.1	2015/06/15	William	1. 修改第六章 API 與範例程式參考內容 2. 修改規格表中的功耗資訊