

iTE SDK

開發工具操作指南

V0.91

ITE TECH. INC.

修訂記錄

修訂日期	修訂說明	頁次
2013/10/16	初建版本 V0.9	
2014/12/1	V0.91 修改小轉板 SPI/Nor 切換方式	12,14,17,22
2015/2/24	修改 USB to SPI 轉接器之使用切換方式	12,14,22
2015/3/2	修改 5. 使用燒錄軟件更新方式	17-23
2015/4/16	修改 5. 使用燒錄軟件連接方式	17-21
2015/08/20	修改編譯軟件名稱	20,22,25
2016/5/4	修改章節 3/4/5	13-22
2016/5/13	新增 SPI-NAND 燒錄方式	20-23
2016/9/2	新增 dbg Debug Tool	27-31
9/9	刪除"7.2 確認 USB-to-SPI 小轉板 Chip IC"	29
12/7	新增 dbg 範例	30-31

目錄

1. 前言	1
1.1 編寫目的	1
1.2 適用範圍	1
1.3 適用人員	1
2. 開發工具設定環境	2
2.1 安裝 USB TO SPI 轉接板驅動程式	2
2.2 安裝除錯訊息工具軟件	5
3. 如何印出除錯訊息	12
3.1 USB TO SPI 連接板訊息線連接方式	12
3.2 開啟偵錯訊息軟件	12
3.3 基板連接電源	13
3.4 列印偵錯訊息	13
4. 使用 USB TO SPI 轉接板讀取暫存器方式	14
4.1 USB TO SPI 轉接板準備狀態	14
4.2 USB TO SPI 連接板連接方式	14
4.3 執行 USB TO SPI 軟件	15
4.4 進入 USB TO SPI 視窗畫面	15
4.5 確認基板連接狀態	16
5. 使用燒錄軟件更新方式	17
5.1 USB TO SPI 連接板連接基板	17
5.2 開啟燒錄軟件	17
5.3 進入 USB TO SPI 視窗畫面	18
5.4 切換 SPI WRITER 視窗	19
5.5 讀取 NOR FLASH INFORMATION	19
5.6 選定欲燒錄更新軟件(FOR NOR BOOT)	20
5.7 軟件檔案燒錄中	20
5.8 燒錄完成	21
5.9 SPI-NAND 燒錄方式	21
5.10 讀取 NAND FLASH INFORMATION	22
5.11 選定欲燒錄更新軟件(FOR SPI-NAND BOOT)	22
5.12 軟件檔案燒錄中	23
5.13 燒錄完成	23
5.14 改變基板啟動模式	24
5.15 新增 SUPPORT NOR FLASH 方式	24
6. 使用隨身碟更新軟件	25
更新前請確認 NOR 裡面已經有 FIRMWARE. 若無請用 SPI 方式.	25
6.1 隨身碟更新軟件	25
6.2 隨身碟使用方式	25
6.3 連接電源開機	25
6.4 更新完成	26
7. IT9850_SERIES 使用 GDB DEBUG TOOL(以 INDOOR2 為例)	27
7.1 BUILD GDB DEBUG 軟件	27
7.2 SPI 切換連接方式	29

7.3	執行 JP2.CMD.....	29
7.4	執行 GDB.CMD.....	30
7.5	開機過程.....	30
範例 -	CALL STACK 被破壞	30

1. 前言

1.1 編寫目的

1. 介紹 iTE SDK 相關的開發工具之用途與操作，說明 iTE SDK 在 EVB 上 uart debug 接線及燒錄 Image 之使用方式。

1.2 適用範圍

使用於 iTE SDK 下相關設備。

1.3 適用人員

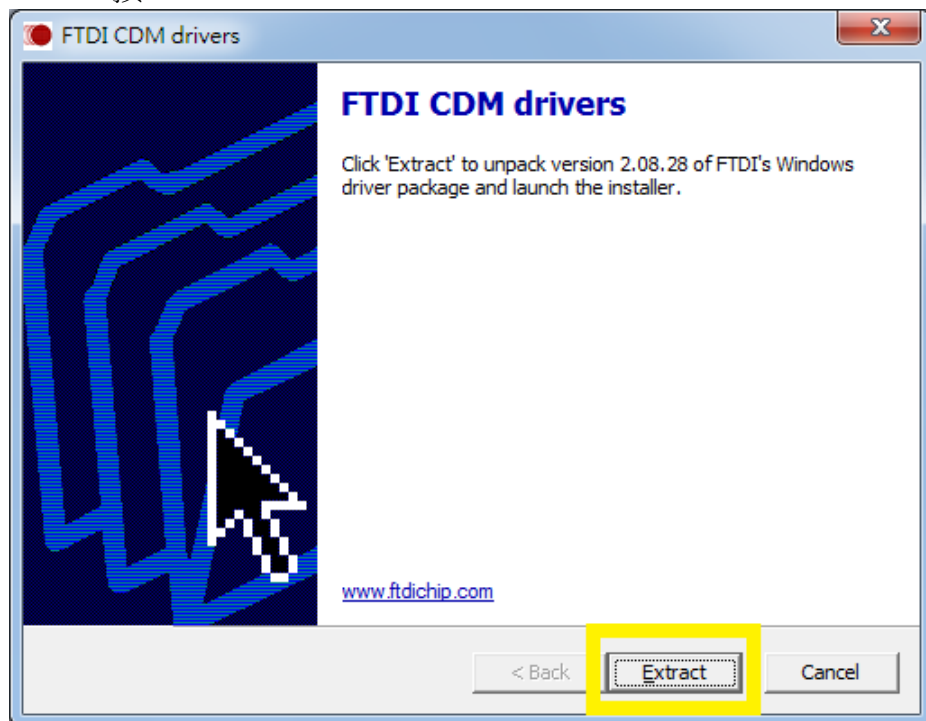
適用於開發 iTE SDK 之嵌入式開發相關人員。

2. 開發工具設定環境

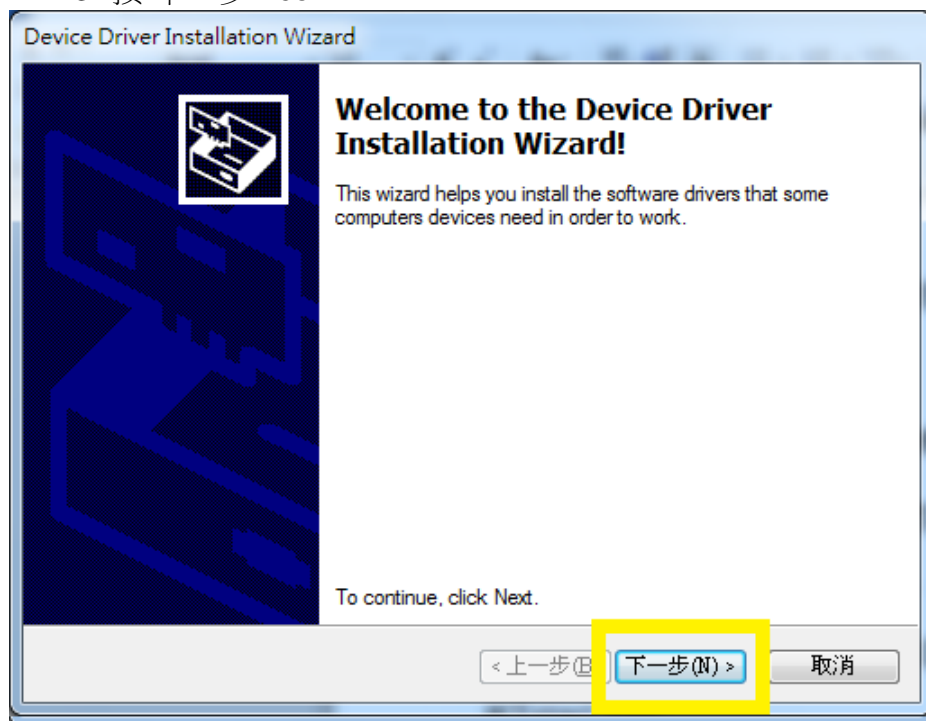
2.1 安裝 USB to SPI 轉接板驅動程式

2.1.1 在 PC 端執行安裝 CDM20828_Setup.exe。

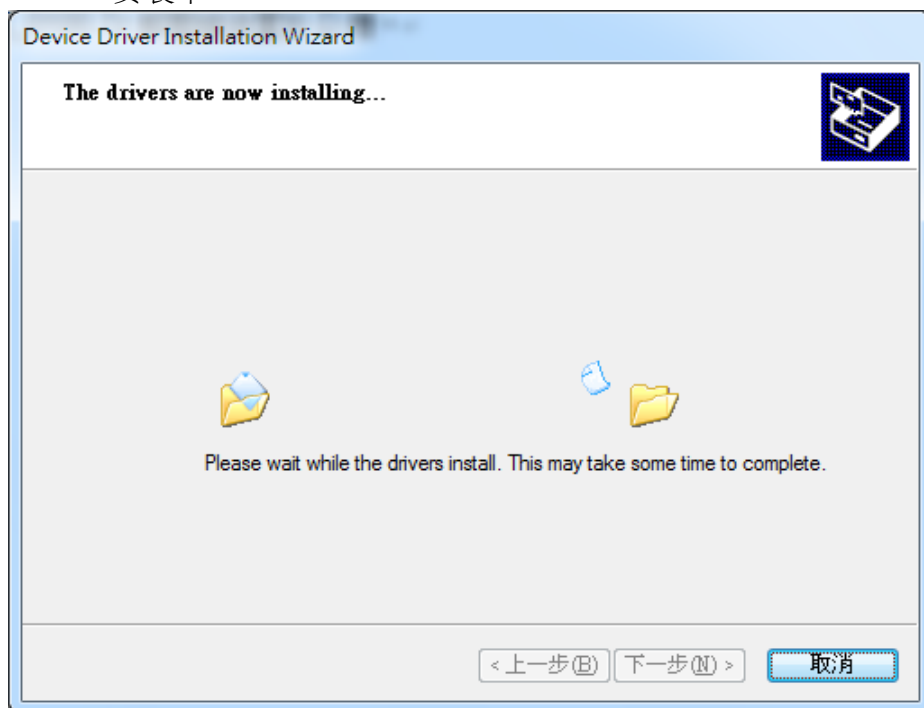
2.1.2 按"Extract" icon。



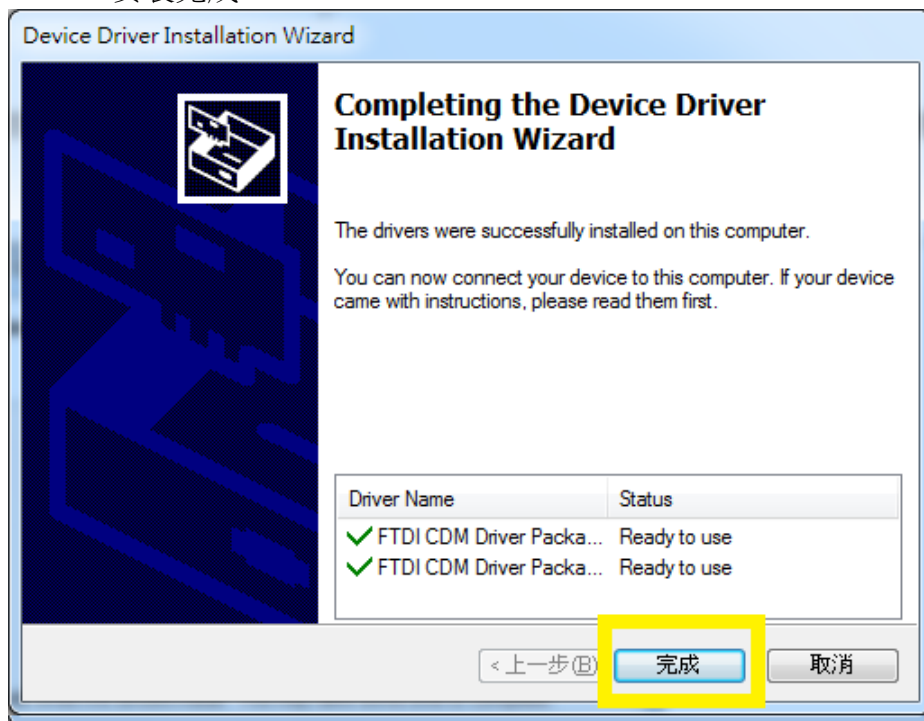
2.1.3 按"下一步" icon。



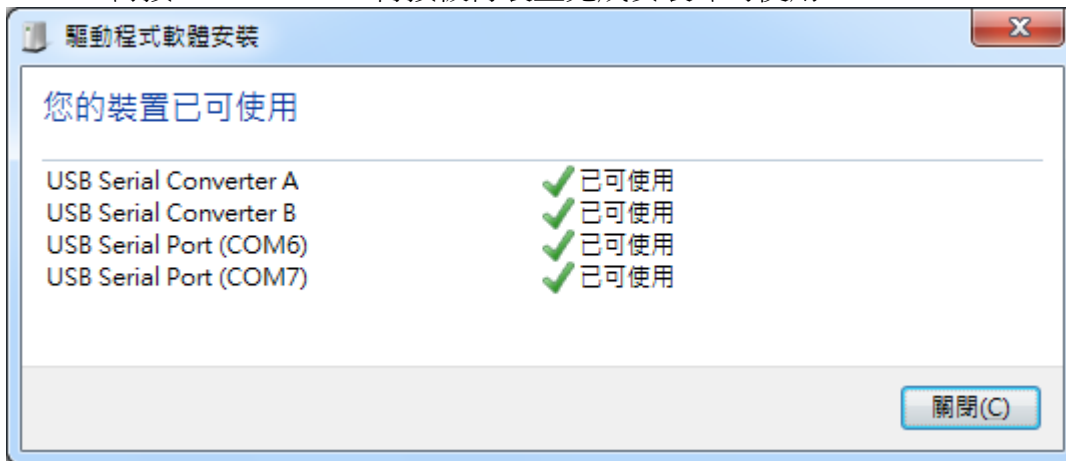
2.1.4 安裝中。



2.1.5 安裝完成。



2.1.6 再接上 Usb to SPI 轉接板待裝置完成安裝即可使用。

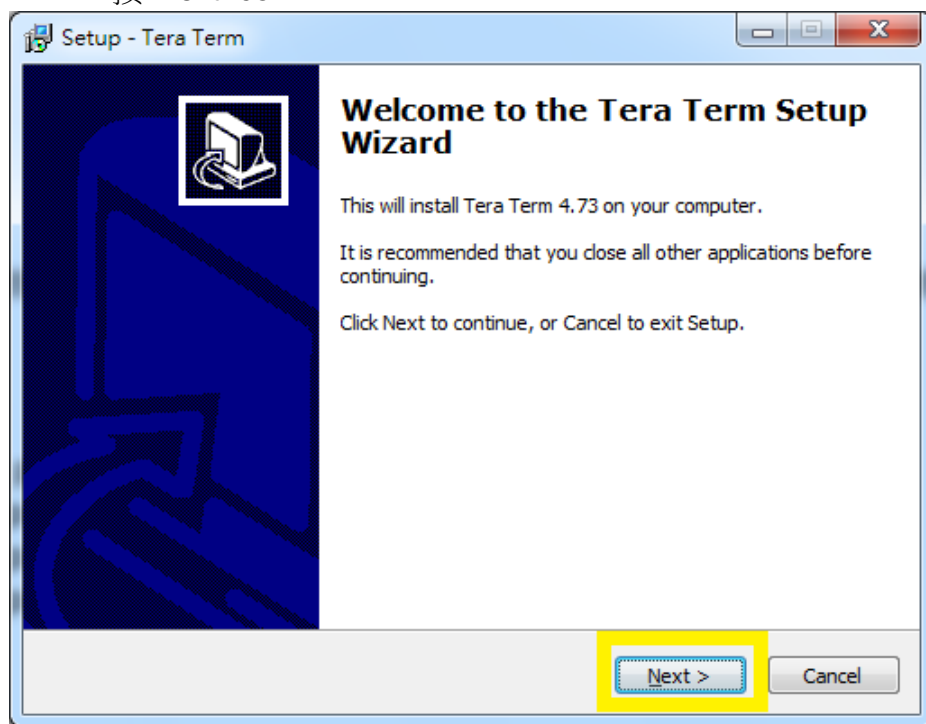


2.2 安裝除錯訊息工具軟件

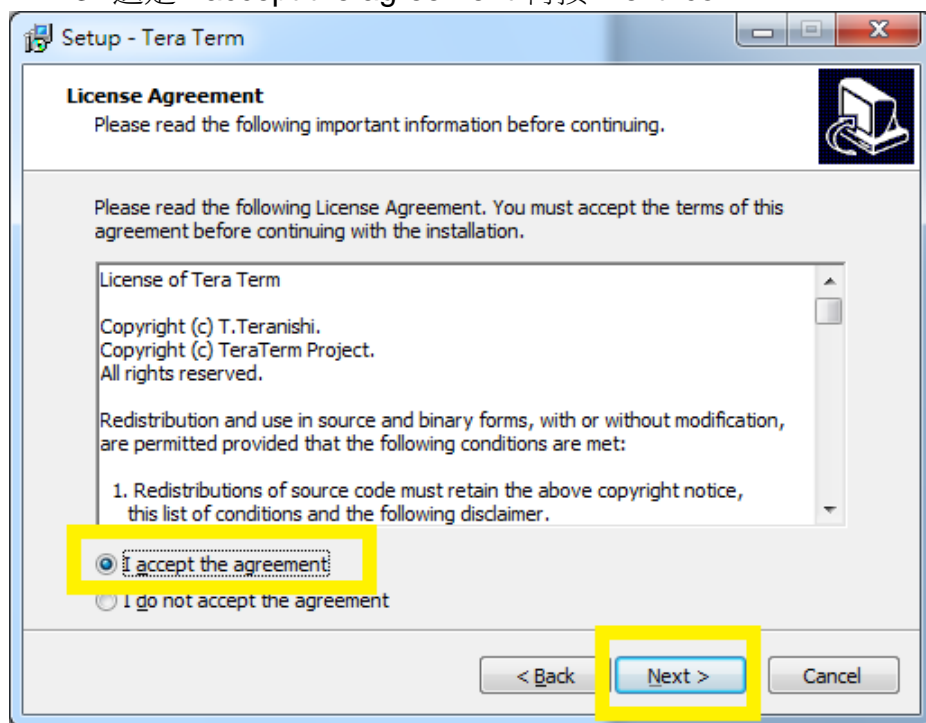
2.2.1 在 PC 端執行安裝 teraterm473.exe 。

Official website: <http://en.sourceforge.jp/projects/ttssh2/releases/>

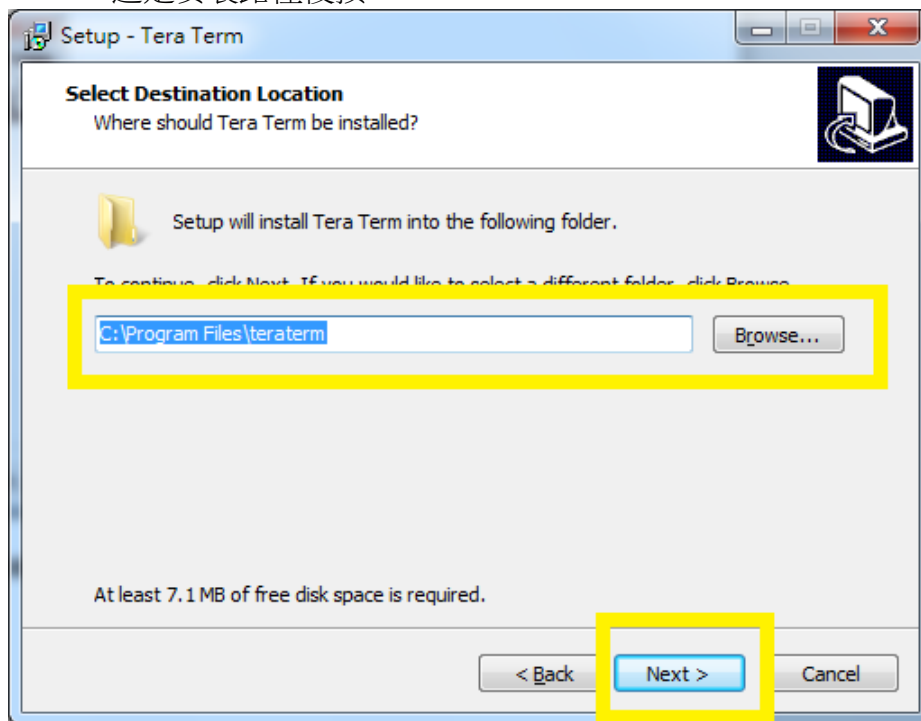
2.2.2 按”Next”icon 。



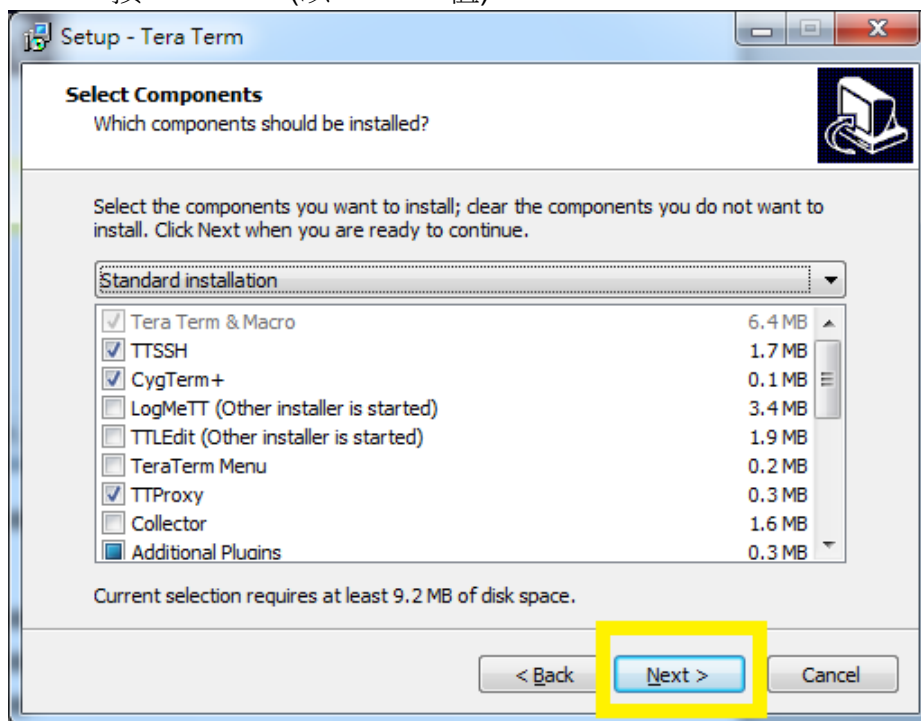
2.2.3 選定”I accept the agreement”再按”Next”icon 。



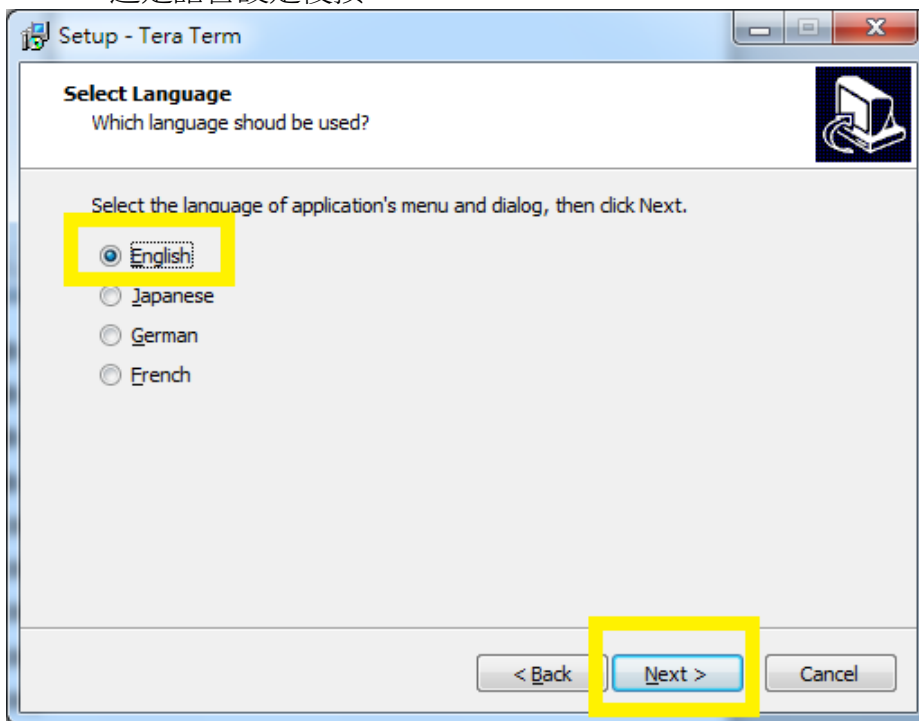
2.2.4 選定安裝路徑後按”Next”icon。



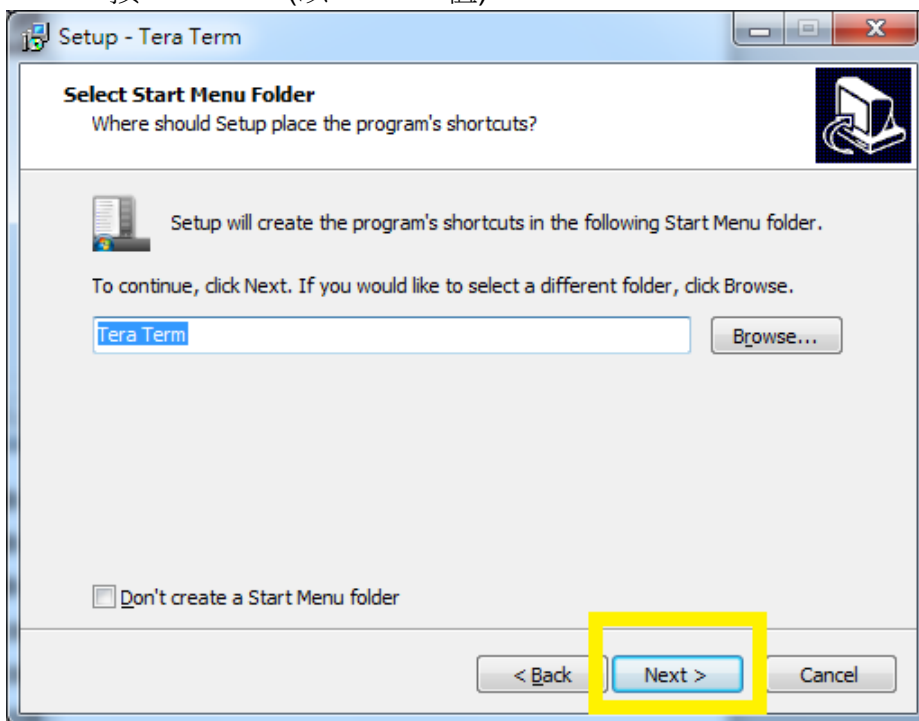
2.2.5 按”Next”icon(以 Default 值)。



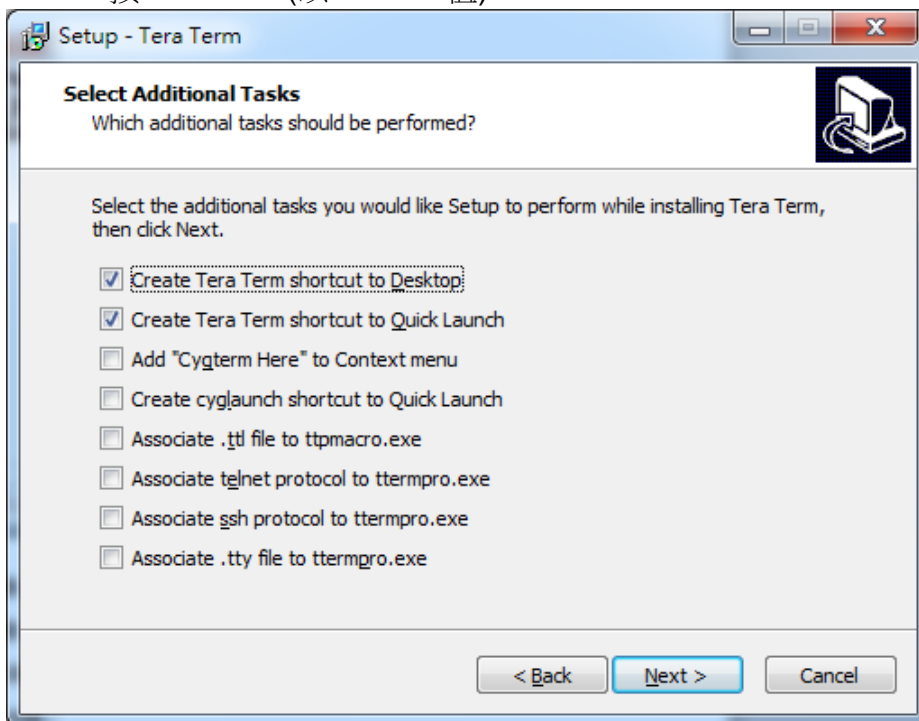
2.2.6 選定語言設定後按”Next”icon。



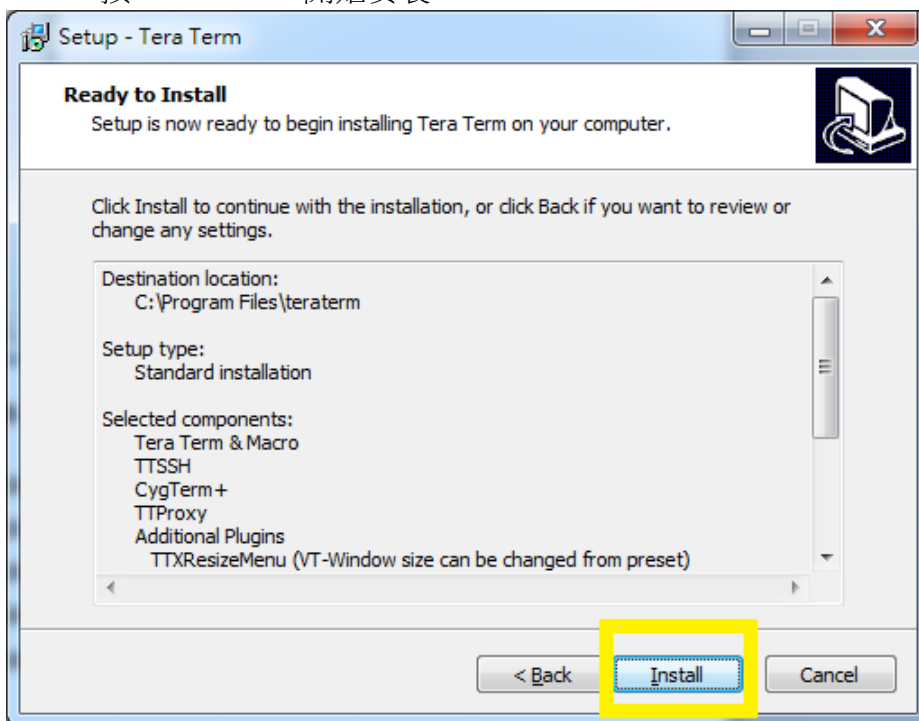
2.2.7 按”Next”icon(以 Default 值)。



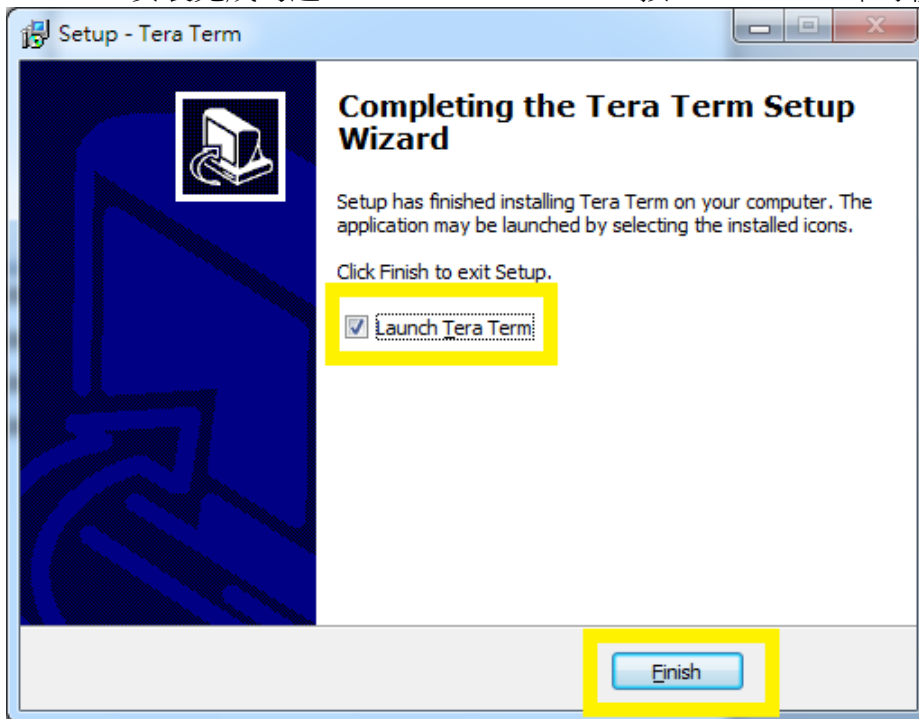
2.2.8 按”Next”icon(以 Default 值)。



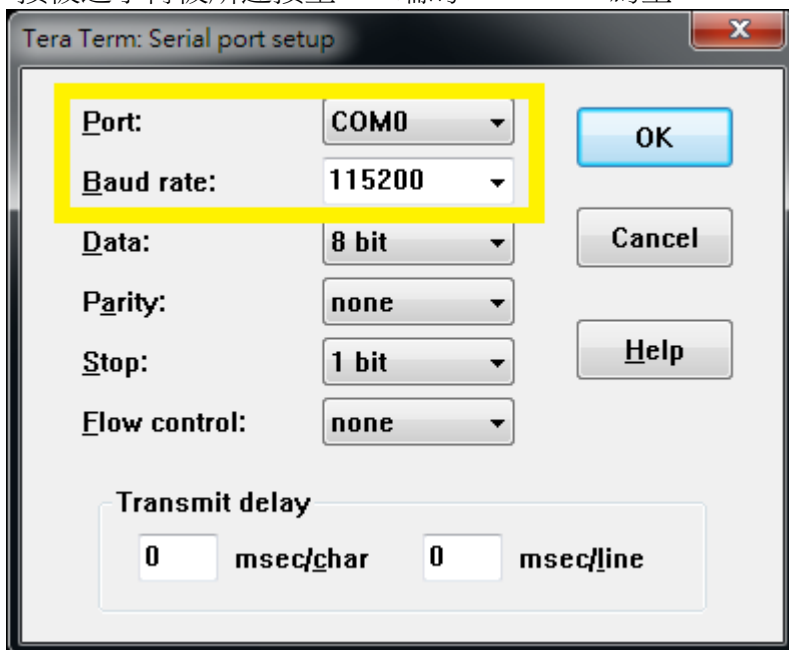
2.2.9 按”Install”icon 開始安裝。



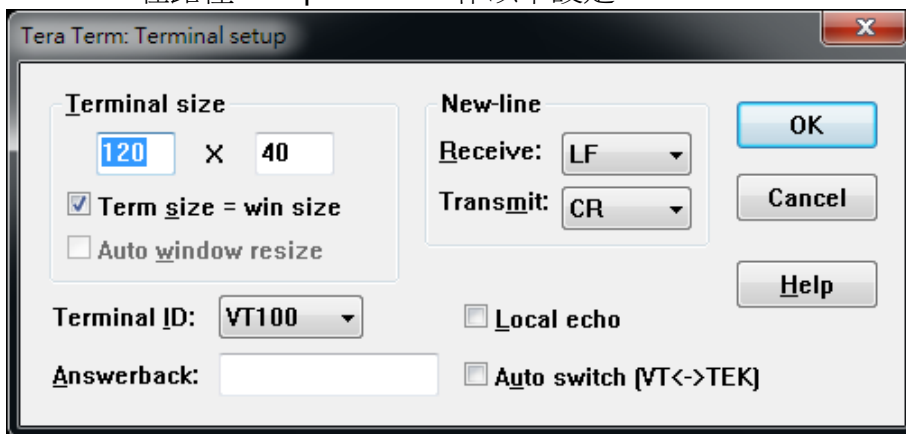
2.2.10 安裝完成勾選“Launch Tera Term”，按“Finish”icon 即可執行 Tera Term。



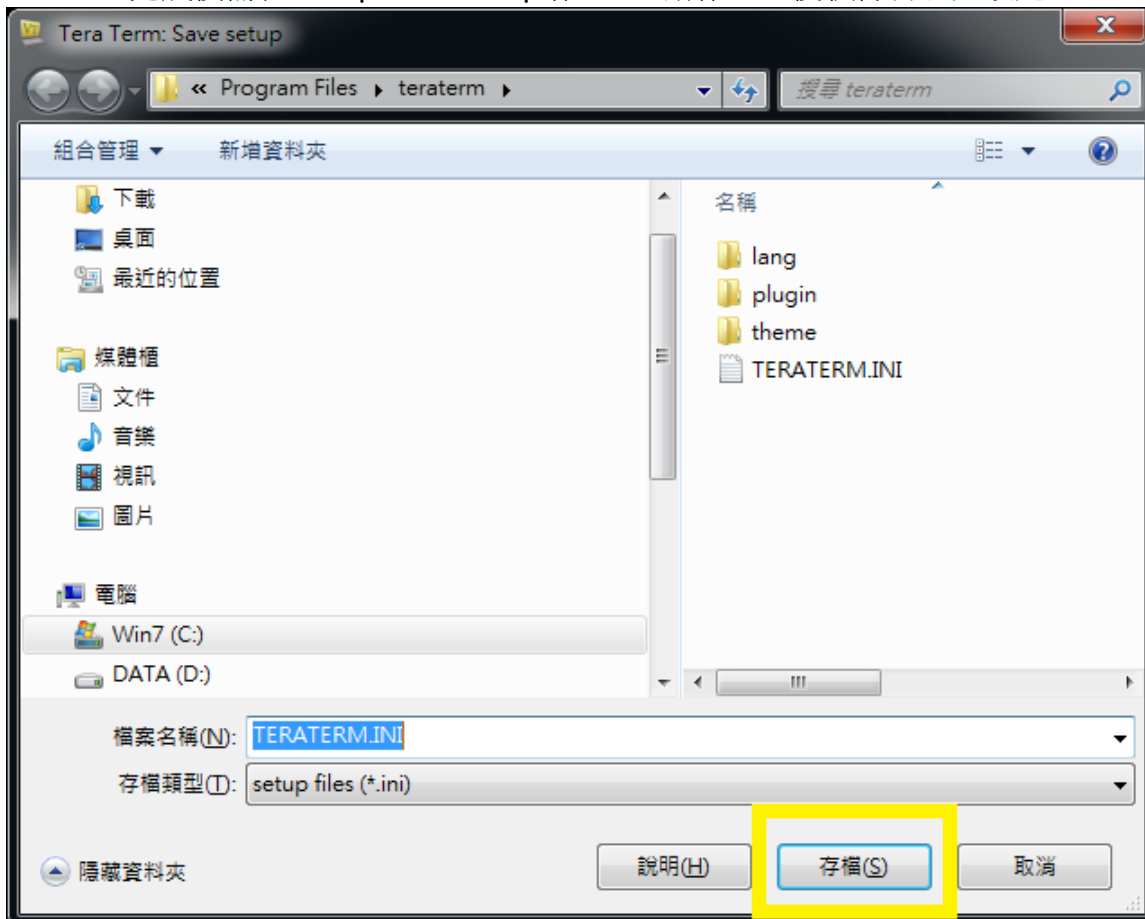
2.2.11 在路徑\setup\serial port 作 Port 及 Baud rate 設定，其中 Port 是依照 USB to SPI 連接板之小轉板所連接至 PC 端的 USB Port 為主，Baud rate 設定值固定為 115200。



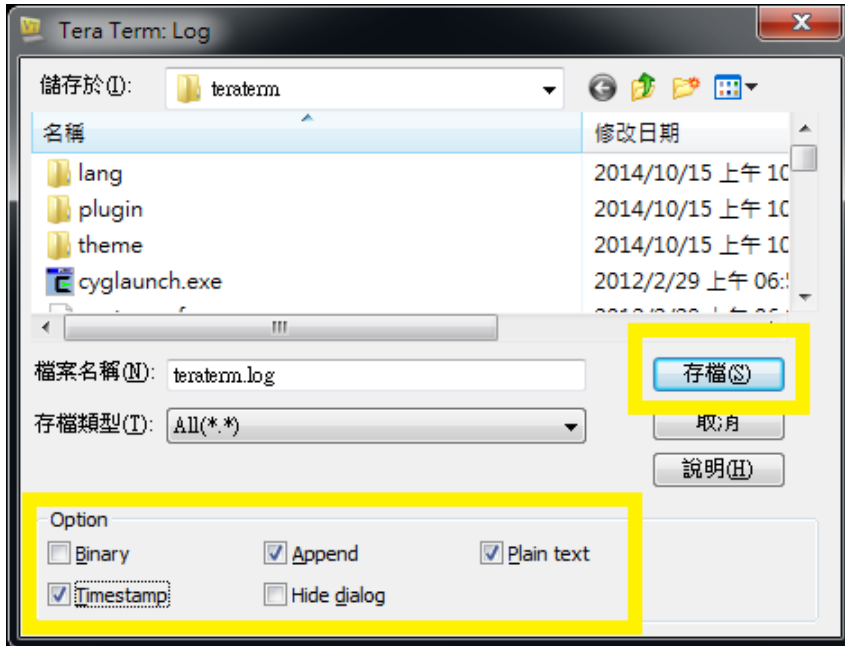
2.2.12 在路徑\setup\Termnal 作以下設定。



2.2.13 完成後需在\setup\save setup 作 save 動作，之後執行皆以此設定。



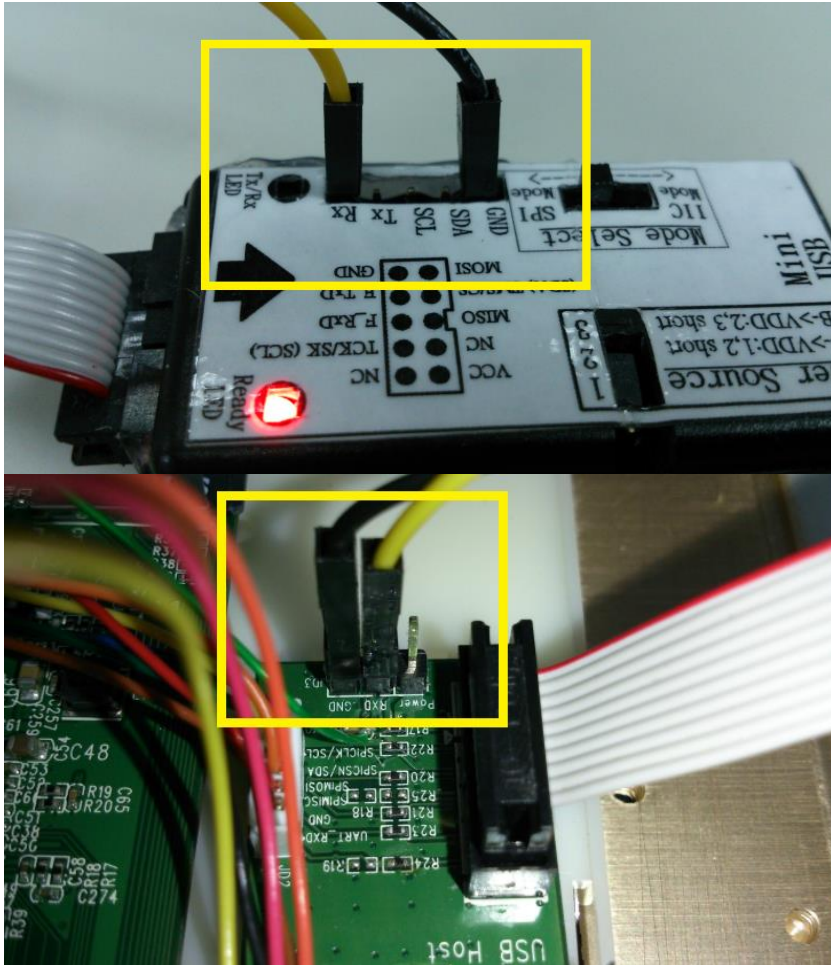
2.2.14 執行 teraterm473.exe 需作 log 記錄可在\File\log。
其中需要記錄 log 時間，則在 Option 多勾選”Timestamp”再存檔即可。



3. 如何印出除錯訊息

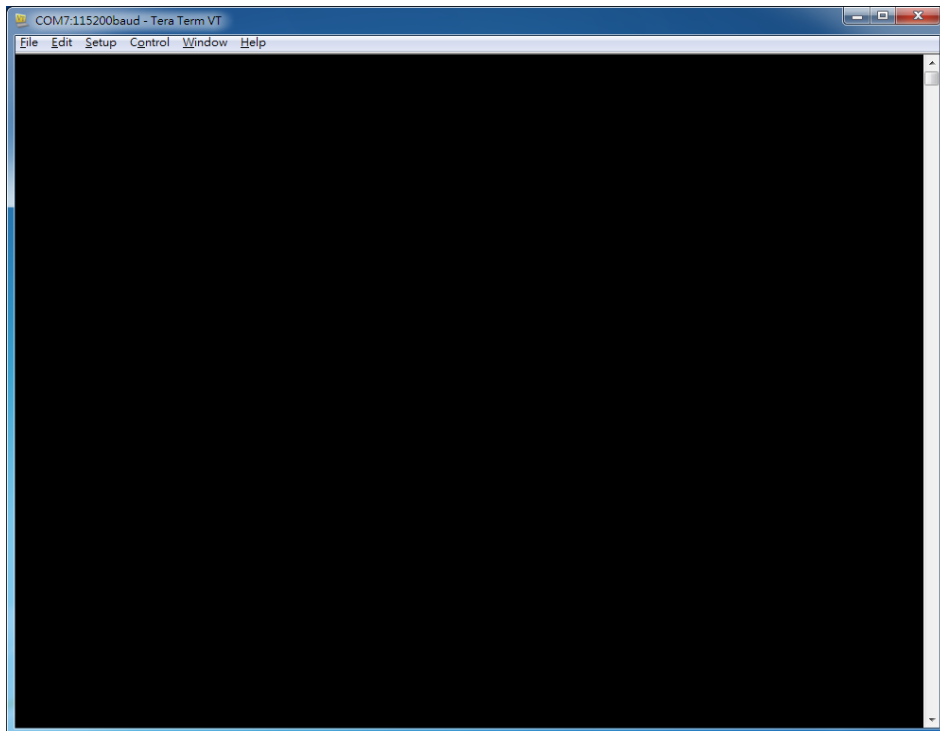
3.1 USB to SPI 連接板訊息線連接方式

將 USB to SPI 連接板之 RX/GND 訊息線橋接至基板之小轉板之 RXD/GND。



3.2 開啟偵錯訊息軟件

開啟 PC 之印出偵錯訊息軟件(ex: teraterm)。



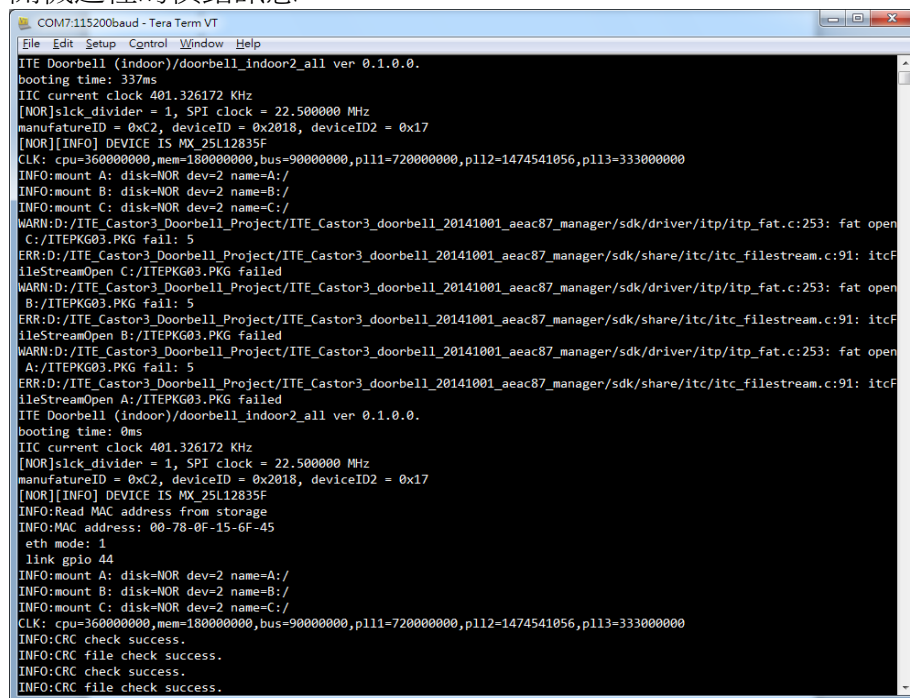
3.3 基板連接電源

連接 DC power(12V)後 Nor boot 開機。

(在 NOR 啟動模式前需確認 USB to SPI 轉接器之 10PIN(JD1)插槽排線已移除)

3.4 列印偵錯訊息

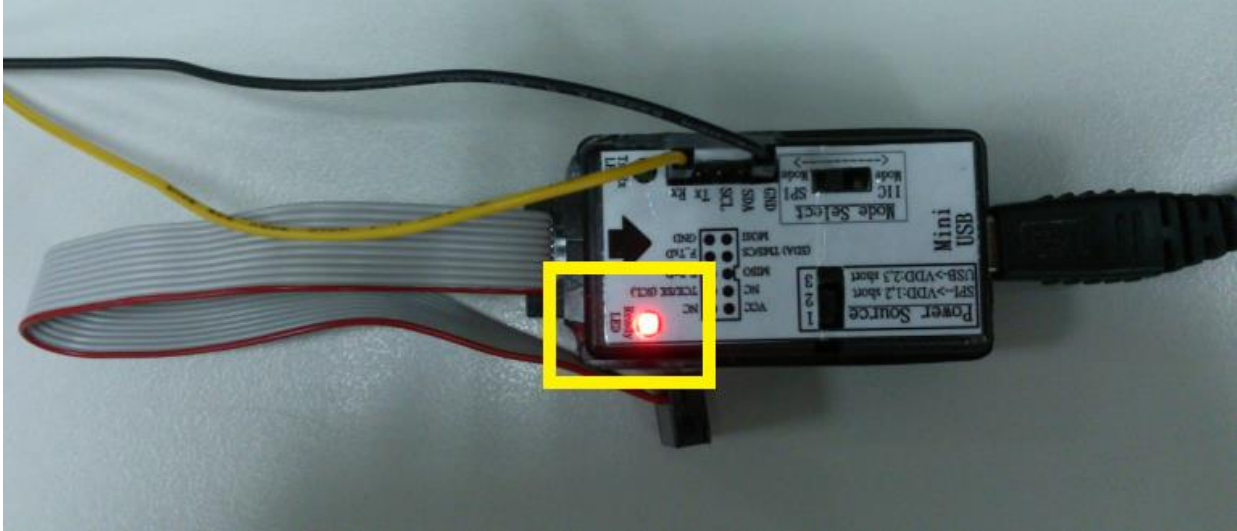
開機過程的偵錯訊息。



4. 使用USB to SPI轉接板讀取暫存器方式

4.1 USB to SPI 轉接板準備狀態

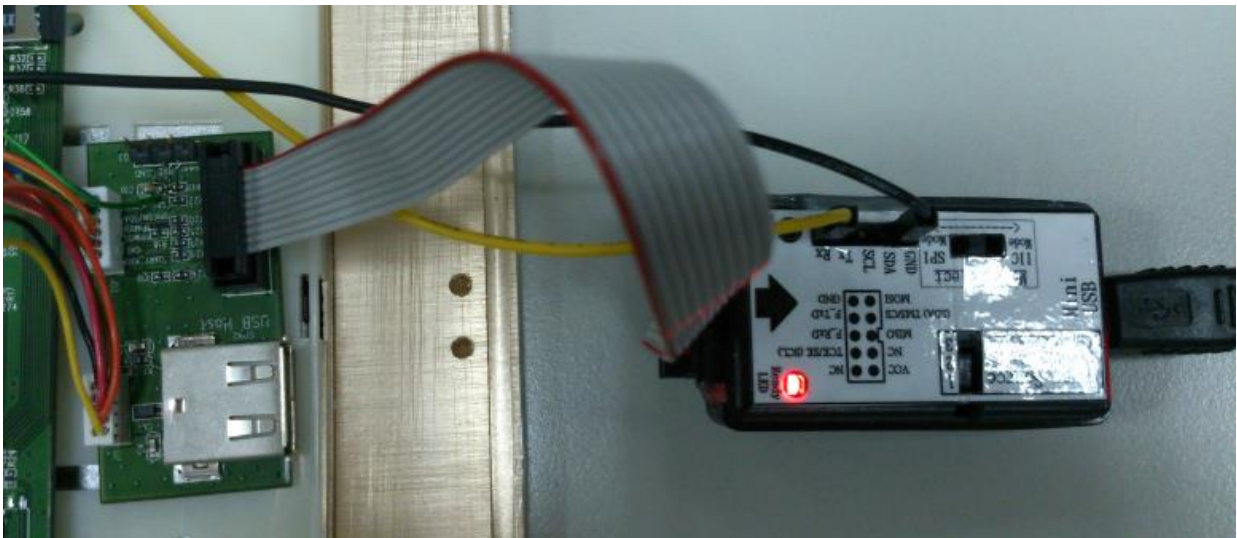
將 USB to SPI 轉接板透過 MiniUSB 與 PC 連接，正確連接會有 Ready LED 亮燈。



4.2 USB to SPI 連接板連接方式

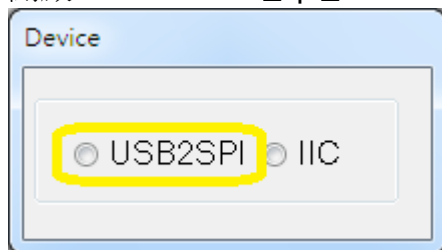
先將 USB to SPI 連接板另一端 10PIN(JD1)插槽排線與基板之小轉版連接，再連接基板電源。

#注意: 因電路設計 **SPI** 與 **IIC** 共用 **GPIO**，所以在 **SPI mode** 時 **IIC**(如:touch function)功能會不 **work**。



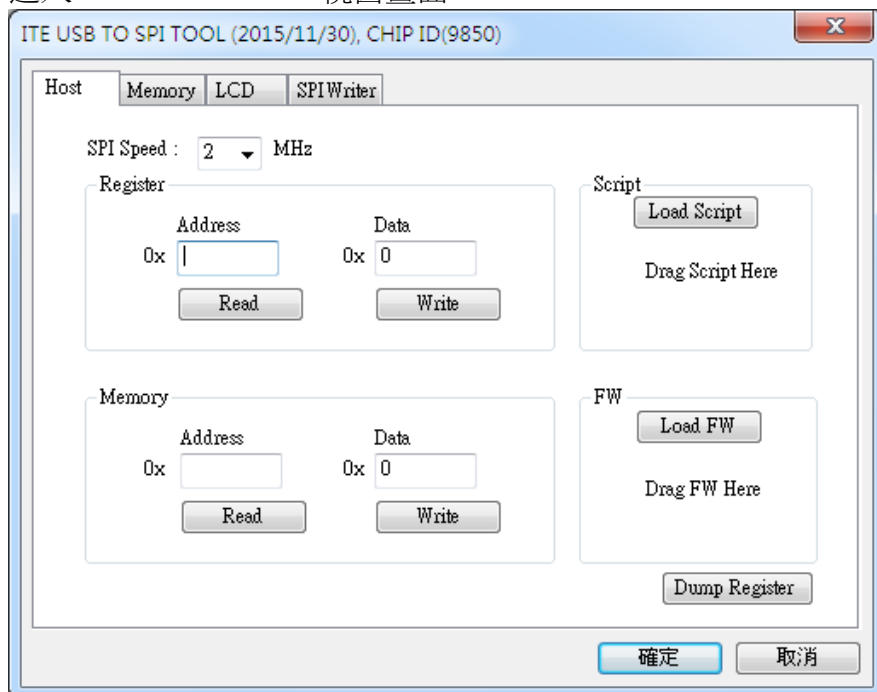
4.3 執行USB to SPI 軟件

開啟 \tool\bin\usb_spi_tool.exe，並勾選“USB2SPI”。



4.4 進入 USB to SPI 視窗畫面

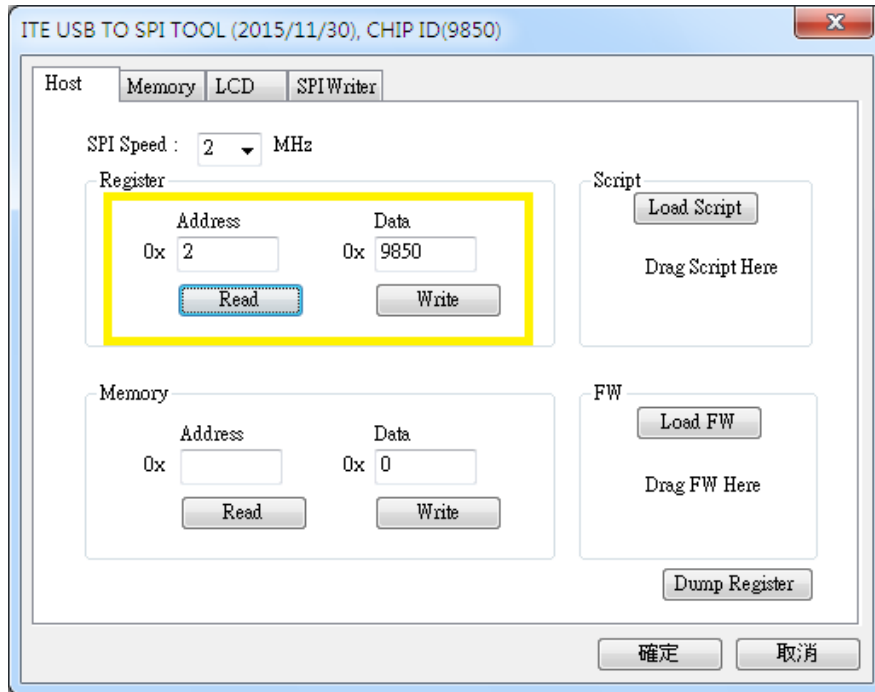
進入 USB to SPI Tool 視窗畫面。



4.5 確認基板連接狀態

在 Register/Address key-in “2”後按”Read”icon，在\Register\Data 會顯示 Chip ID，即是準備好狀態，若無則表示硬體尚有問題，需作以下動作：

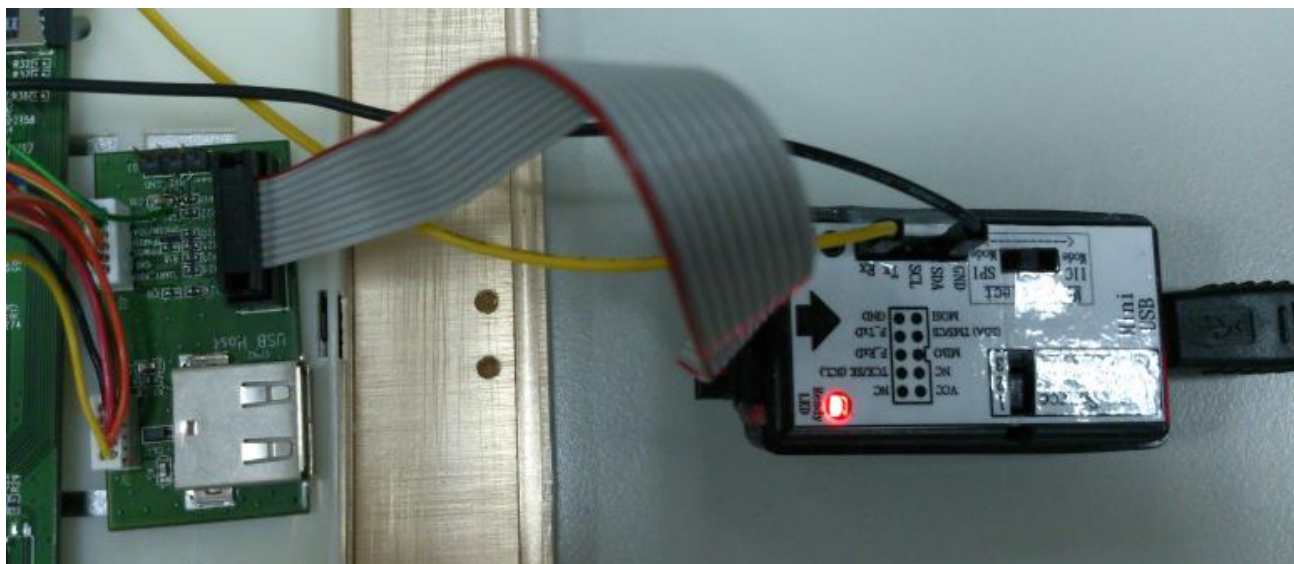
- 1.請先重新上電後再試。
- 2.確認主板工作正常。



5. 使用燒錄軟件更新方式

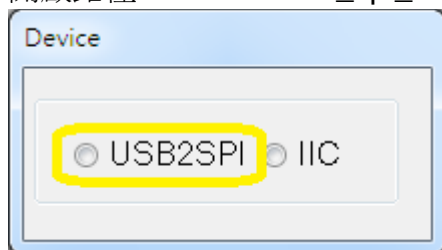
5.1 USB to SPI 連接板連接基板

將小轉板透過 USB to SPI 連接板連接基板之小轉板，再接 DC power(12V)。



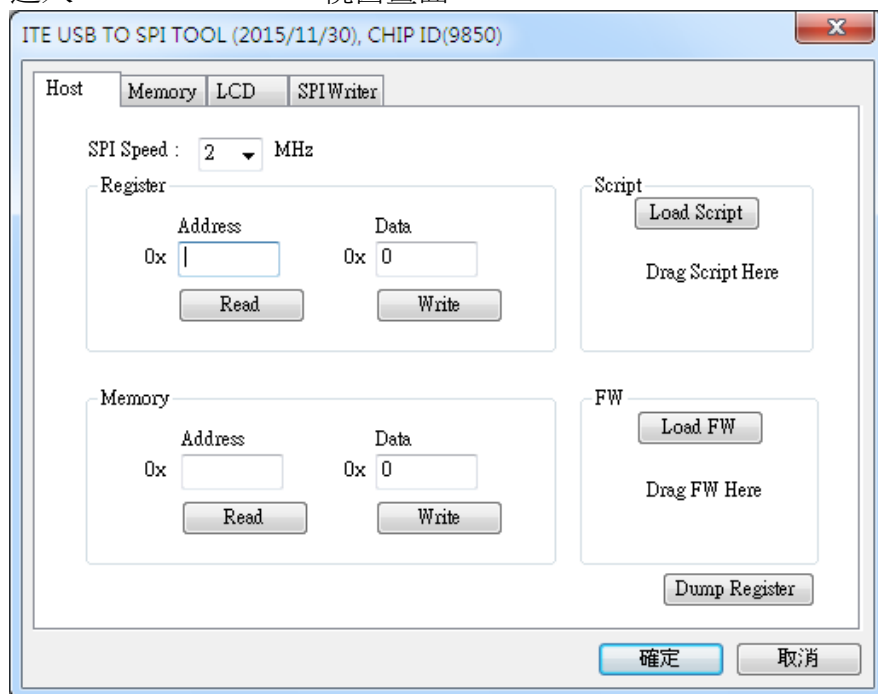
5.2 開啟燒錄軟件

開啟路徑: \tool\bin\usb_spi_tool.exe，並勾選"USB2SPI"。

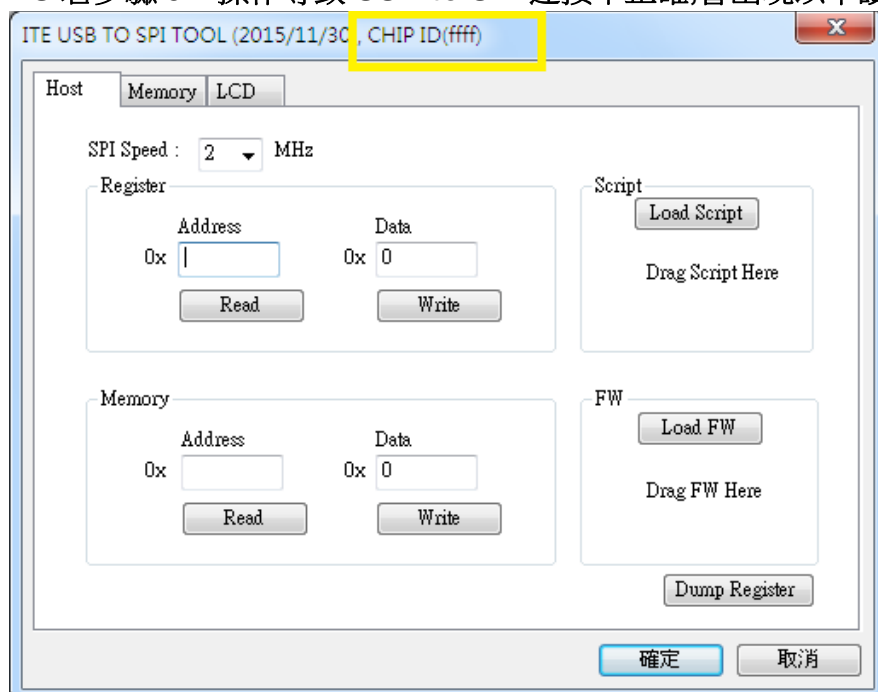


5.3 進入 USB to SPI 視窗畫面

進入 USB to SPI Tool 視窗畫面。

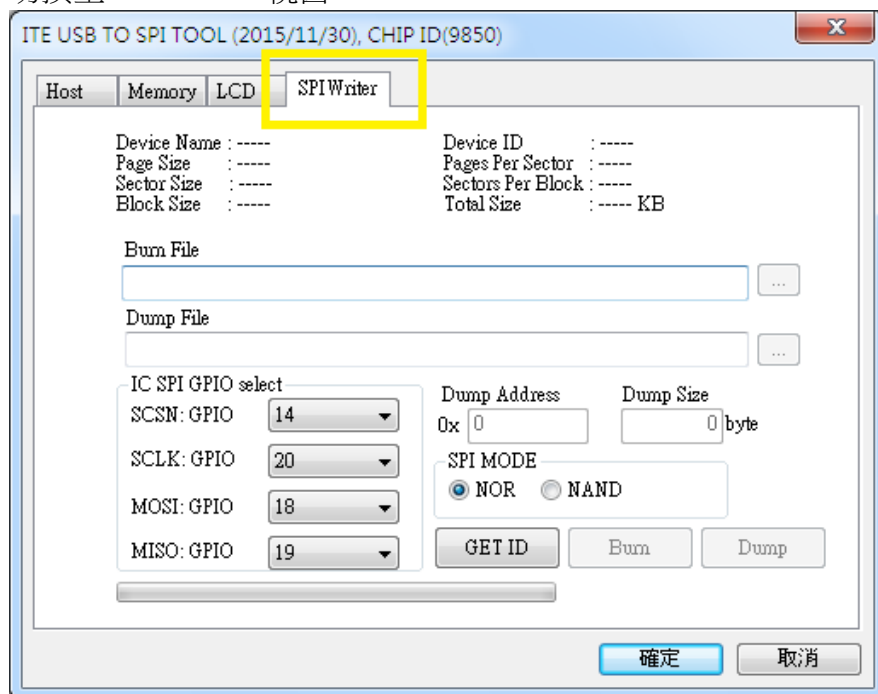


PS.若步驟 5.1 操作導致 USB to SPI 連接不正確,會出現以下讀取不到"CHIP ID"畫面:



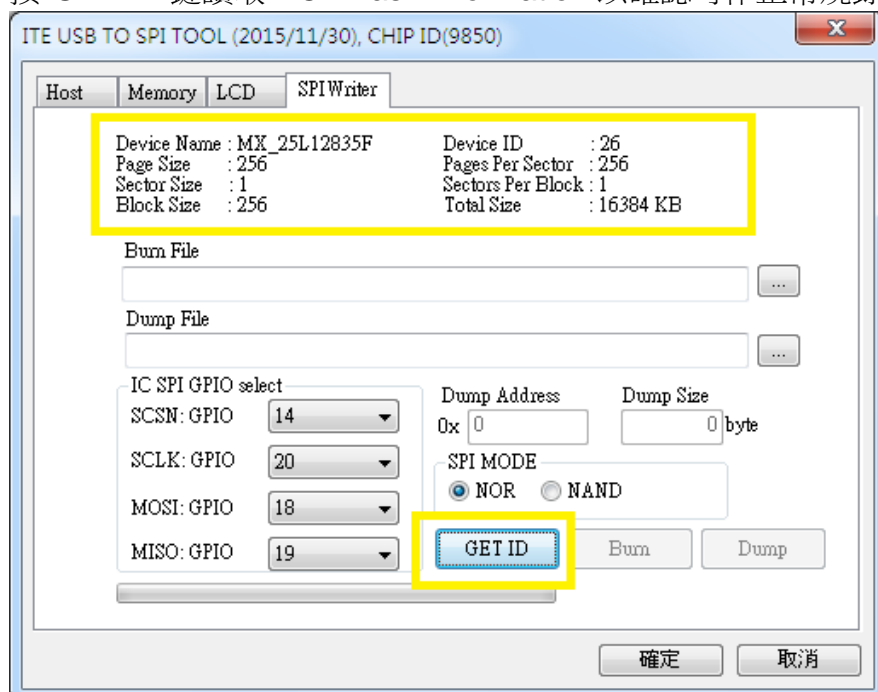
5.4 切換 SPI Writer 視窗

切換至 SPI Writer 視窗。



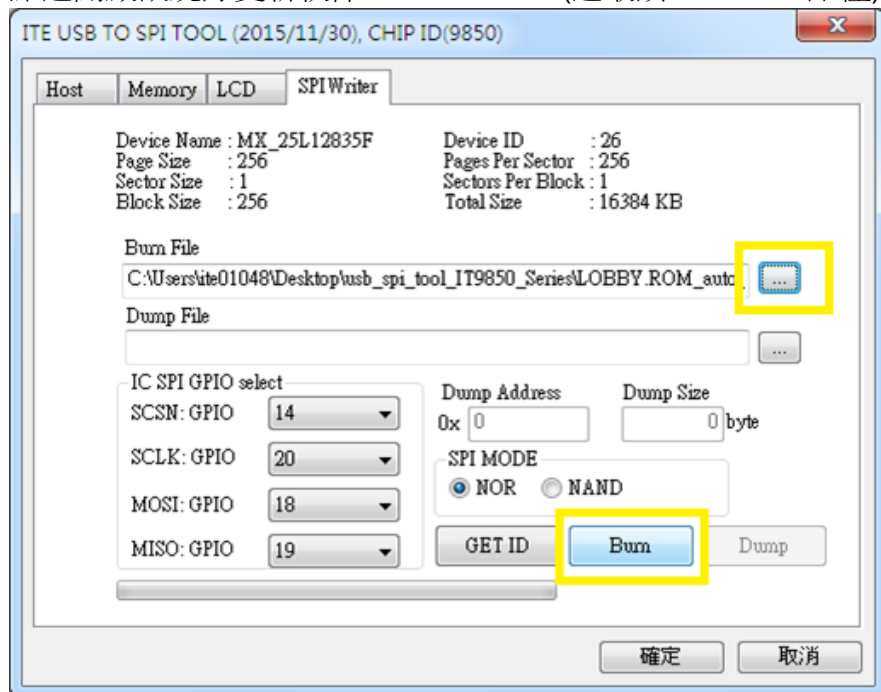
5.5 讀取 NOR flash information

按“GET ID”鍵讀取 NOR flash information 以確認可作正常燒錄。

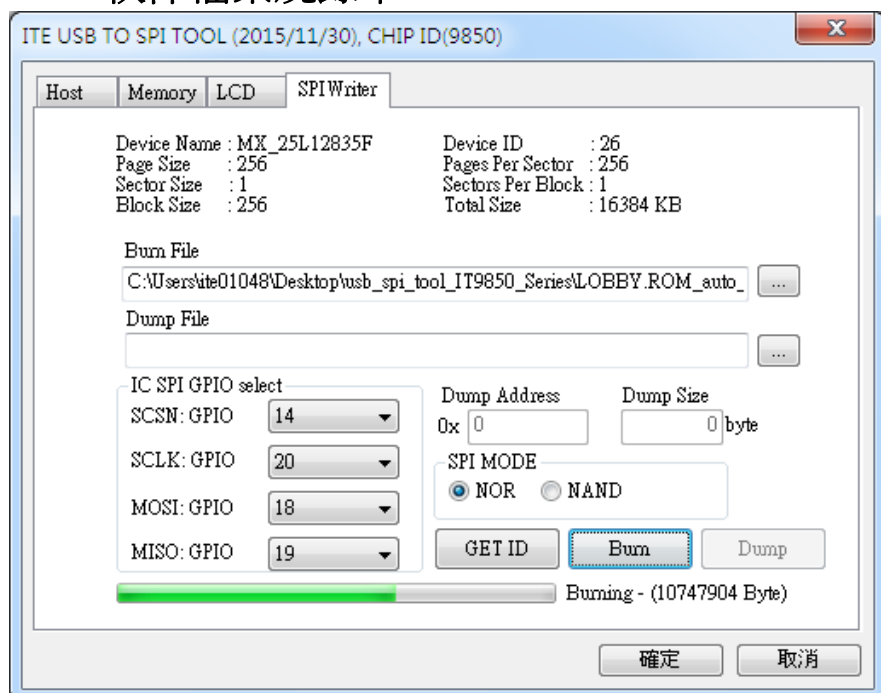


5.6 選定欲燒錄更新軟件(for Nor boot)

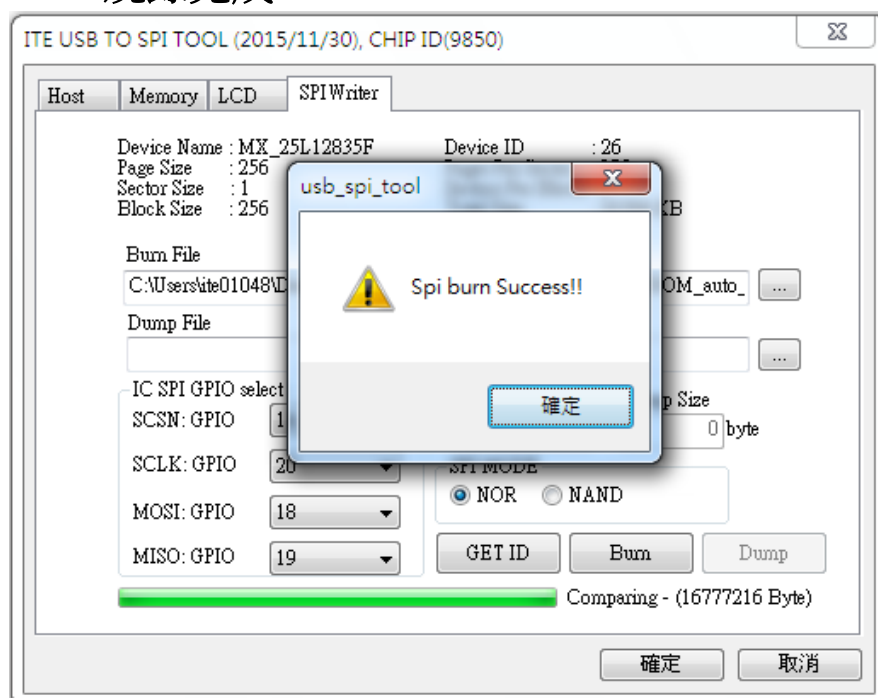
點選開啟欲燒錄更新軟件- XXX.ROM。(選取該 ROM file 路徑)



5.7 軟件檔案燒錄中



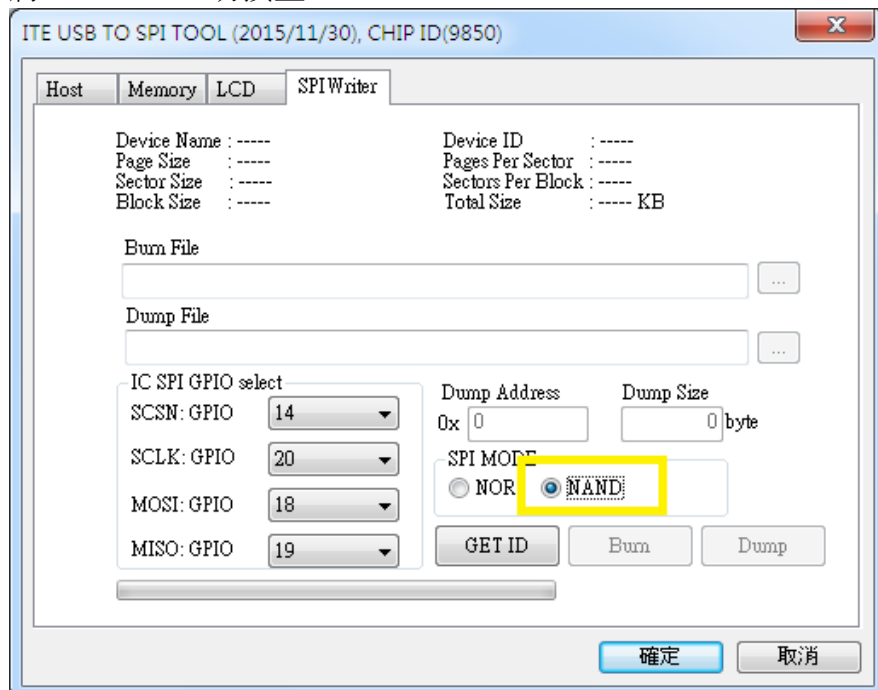
5.8 燒錄完成



燒錄完成後會出現“Spi burn Success!!”視窗。

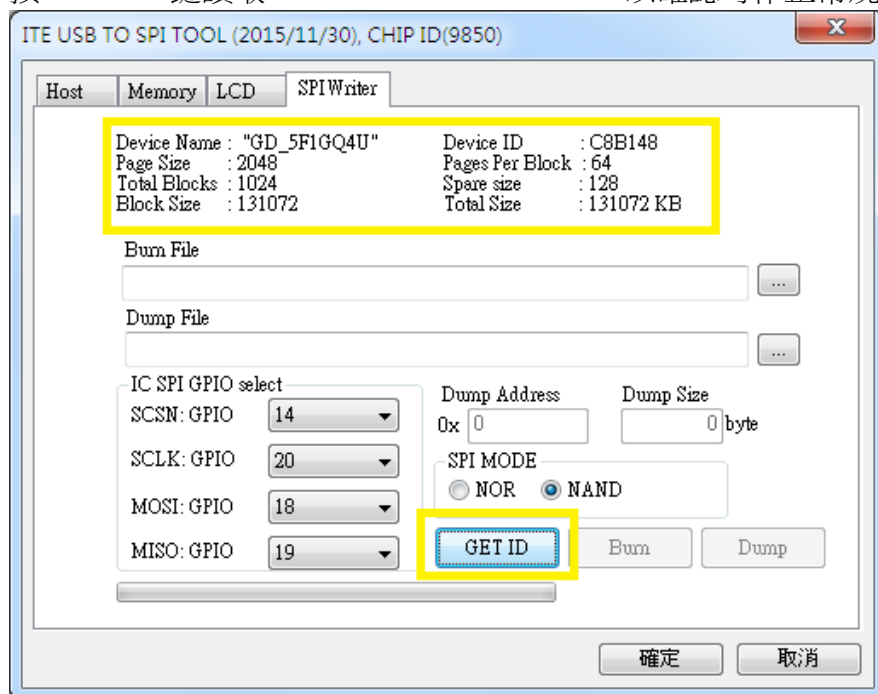
5.9 SPI-NAND 燒錄方式

將 SPI-NAND 切換至“NAND”。



5.10 讀取 NAND flash information

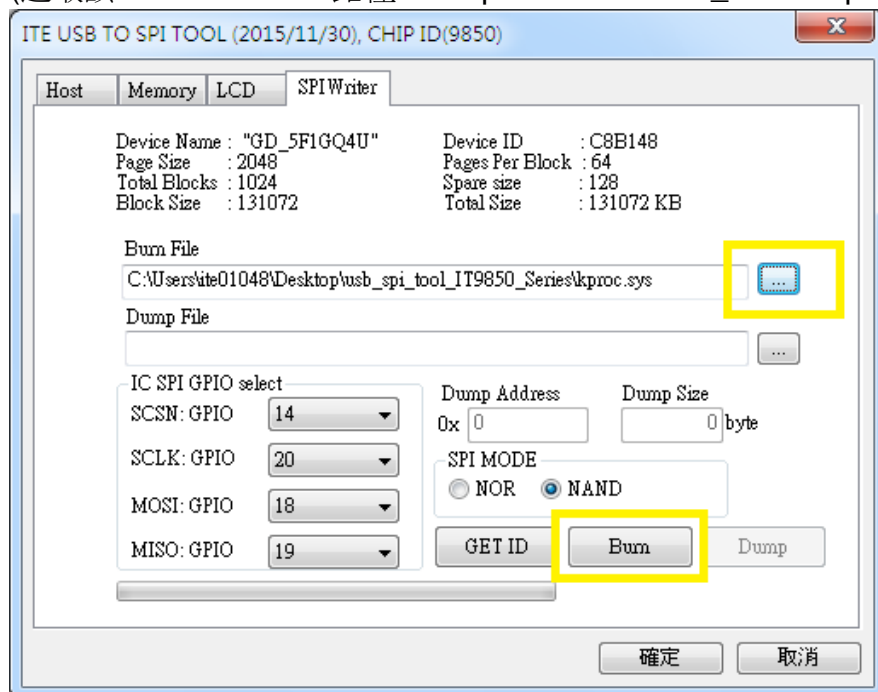
按“GET ID”鍵讀取 NAND flash information 以確認可作正常燒錄。



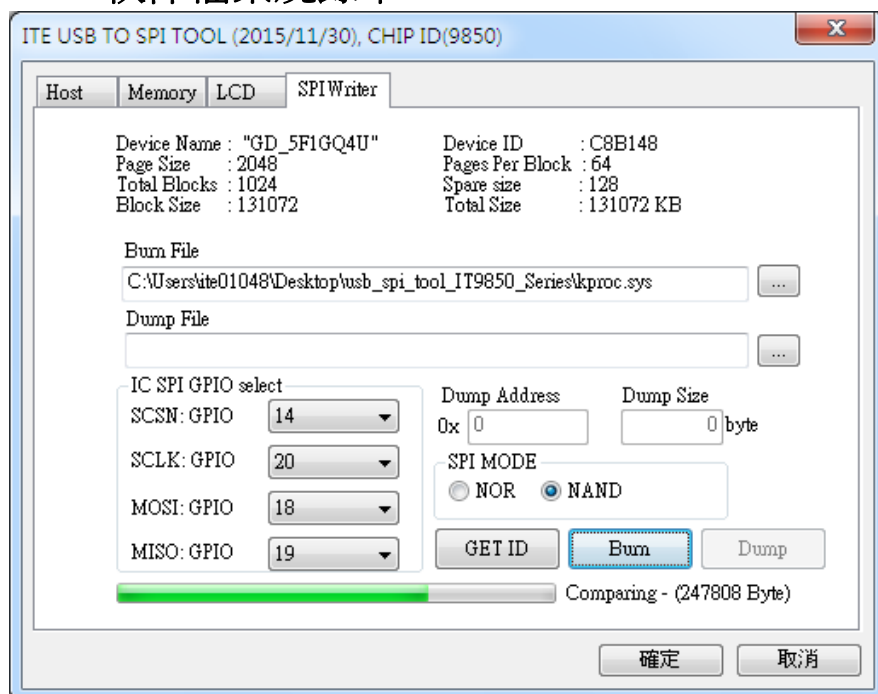
5.11 選定欲燒錄更新軟件(for SPI-NAND boot)

點選開啟欲燒錄 bootloader 軟件- kproc.sys。

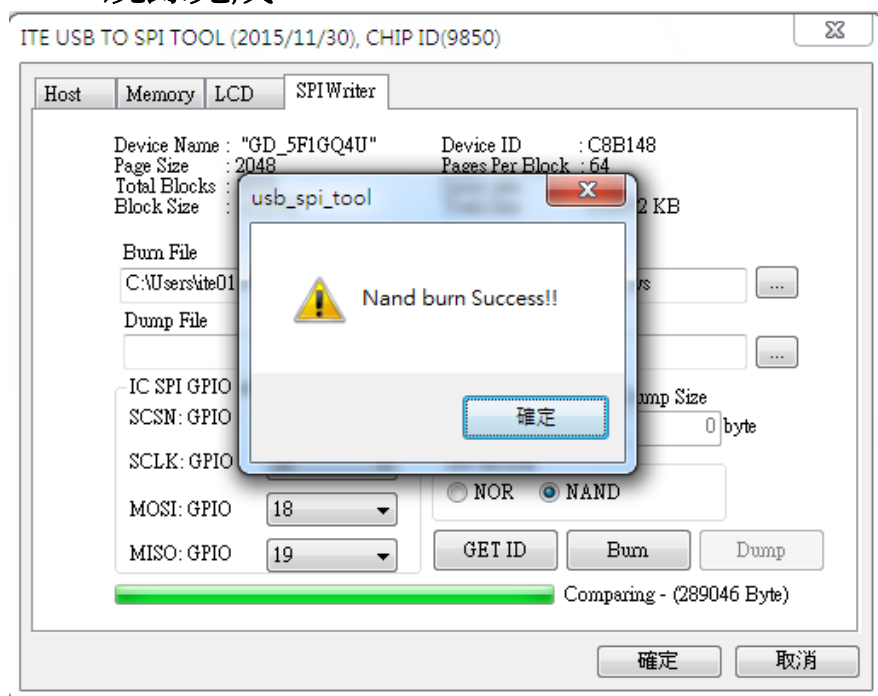
(選取該 bootloader file 路徑 ex: \openrtos\doorbell_indoor2\project\bootloader)



5.12 軟件檔案燒錄中



5.13 燒錄完成



燒錄完成後會出現"Nand burn Success!!"視窗。

注意事項：使用燒錄軟件作更新有分 NOR boot 及 SPI-NAND boot,

NOR boot :透過燒錄軟件更新 xxx.ROM 後開機即可 boot 完成。

SPI-NAND boot: 透過燒錄軟件更新 kproc.sys，仍需再透過 Pen-Drive 更新 xxx.PKG 後才能 boot 完成。

5.14 改變基板啟動模式

先將 USB to SPI 轉接器連及 DC power(12V)線移除，再重新接上 DC power(12V)電源即可。
(因電路設計 SPI 與 IIC 共用 GPIO，在 NOR 啟動模式前需確認 USB to SPI 轉接器之 10PIN(JD1)插槽排線已移除)

5.15 新增 Support NOR Flash 方式

開啟路徑: \tool\bin\nor_config.txt,配合 Nor flash Spec 將 Nor Config Format 依照 name,ManufactureID,DeviceID,DeviceID2,bytesPerPage,pagesPerSector,bytesPerSector, sectorsPerBlock, totalBlocks 作新增。

其每個項目定義如下:

name: string(Max 32bytes)
ManufactureID: Hexformat 0xXX (upper case)
DeviceID: Hexformat 0XXXXX (upper case)
DeviceID2: Hexformat 0xXX (upper case)
bytesPerPage: integer
pagesPerSector: integer
bytesPerSector: integer
sectorsPerBlock:integer
totalBlocks: integer。

完成後以 NOR Flash – SPAN_S25FL127S 為例:

{SPAN_S25FL127S, 0x01, 0x0017, 0xFF, 256, 256, 65536, 1, 256},

注意事項：

此手動新增 Nor flash support 方式僅限 norwriter 可燒錄所新增的 nor flash,但涉及 boot 仍需 在\sdk\driver\nor.c 新增該 nor flash 相關 Spec 才能正常 work。

6. 使用隨身碟更新軟件

更新前請確認 nor 裡面已經有 firmware. 若無請用 spi 方式.

6.1 隨身碟更新軟件

在基板可正常開機下，將 pkg 置於隨身碟可作為更新軟件之用途。

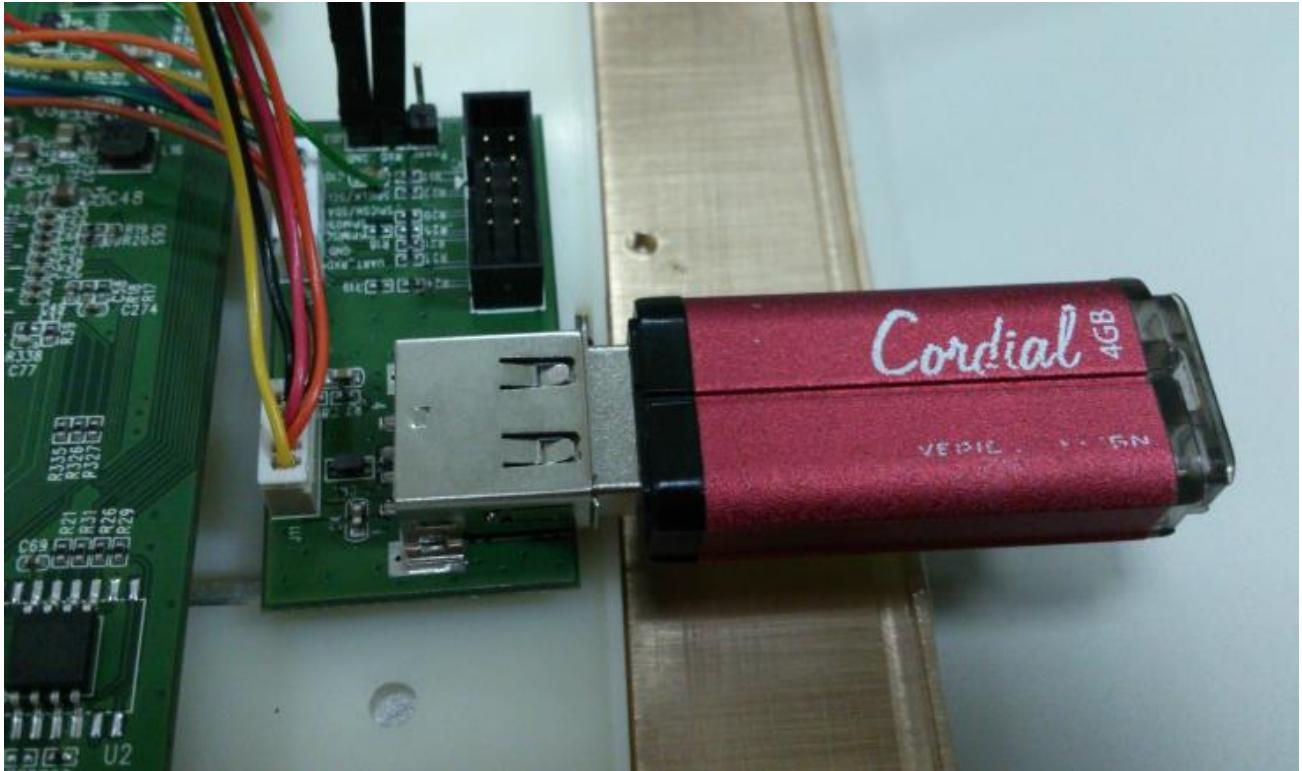
首先將路徑:build\openrtos\project_name\project project_name \xxxx.PKG

確定 PKG 名稱正確搭配到原先 firmware 的否則無法升級.

PS. 隨身碟請務必 Format 成 FAT32 格式後再放置更新軟件。

6.2 隨身碟使用方式

將隨身碟置入基板之小轉板的 USB Port。



6.3 連接電源開機

上電後開機若系統有正確偵測到相對應的 PKG,即進入讀取隨身碟的更新軟件作更新動作升級畫面如下。

```
INFO:Verifying...
INFO:Write data to position 0x00000, size 3663452 bytes
INFO:Writing.....
INFO:Verifying.....
INFO:Create directory A:/backup
INFO:Create directory A:/backup/B
INFO:Writing A:/backup/B/addressbook.xml, 5063 bytes.
INFO:Verifying.
INFO:Writing A:/backup/B/doorbell_indoor.ini, 1280 bytes.
INFO:Verifying.
INFO:Writing A:/backup/B/ite_crc.dat, 50 bytes.
INFO:Verifying.
INFO:Create directory A:/backup/C
INFO:Writing A:/backup/C/ite_crc.dat, 20 bytes.
INFO:Verifying.
INFO:Writing A:/backup/C/lifphonerc, 199 bytes.
INFO:Verifying.
INFO:Create directory A:/codec
INFO:Writing A:/codec/mp3.codecs, 55156 bytes.
INFO:Verifying.
INFO:Writing A:/codec/auac.codecs, 21394 bytes.
INFO:Verifying.
INFO:Writing A:/doorbell_indoor.itu, 925194 bytes.....
INFO:Verifying.....
INFO:Writing A:/doorbell_indoor_chs.itu, 925194 bytes.....
INFO:Verifying.....
INFO:Writing A:/doorbell_indoor_cht.itu, 925194 bytes.....
INFO:Verifying.....
INFO:Create directory A:/font
INFO:Writing A:/font/AmQuanyINMicroBellHomo.ttf, 4454704 bytes.....
```

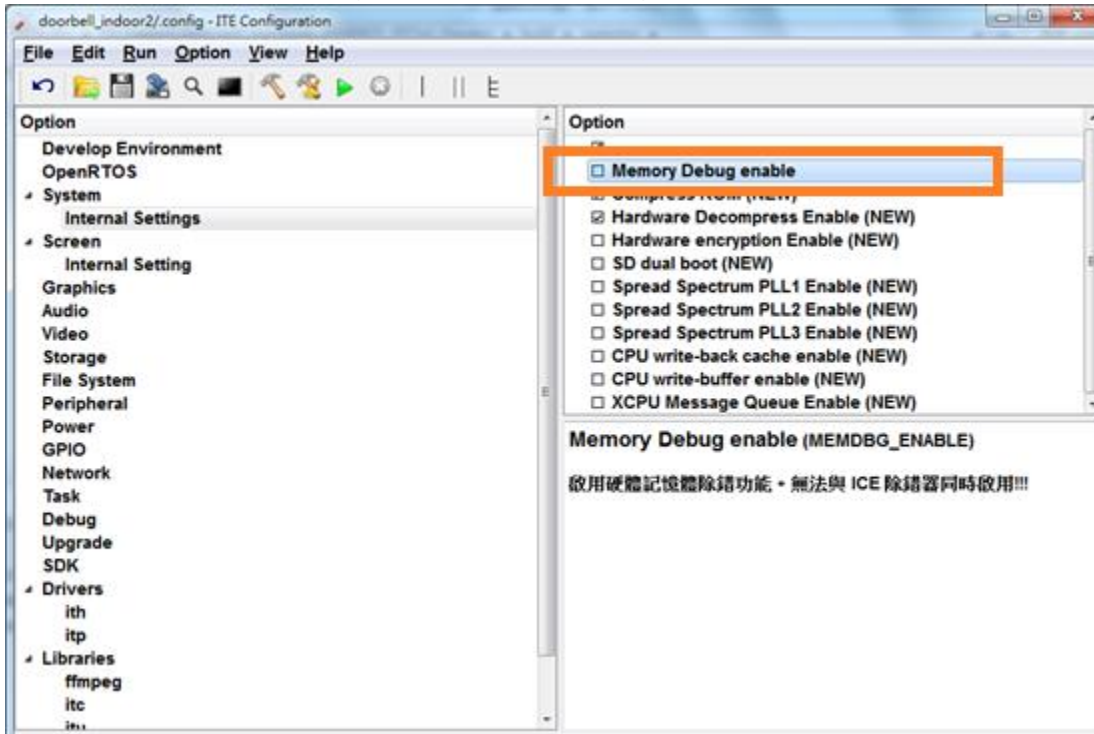
6.4 更新完成

更新完成會自動重新開機。

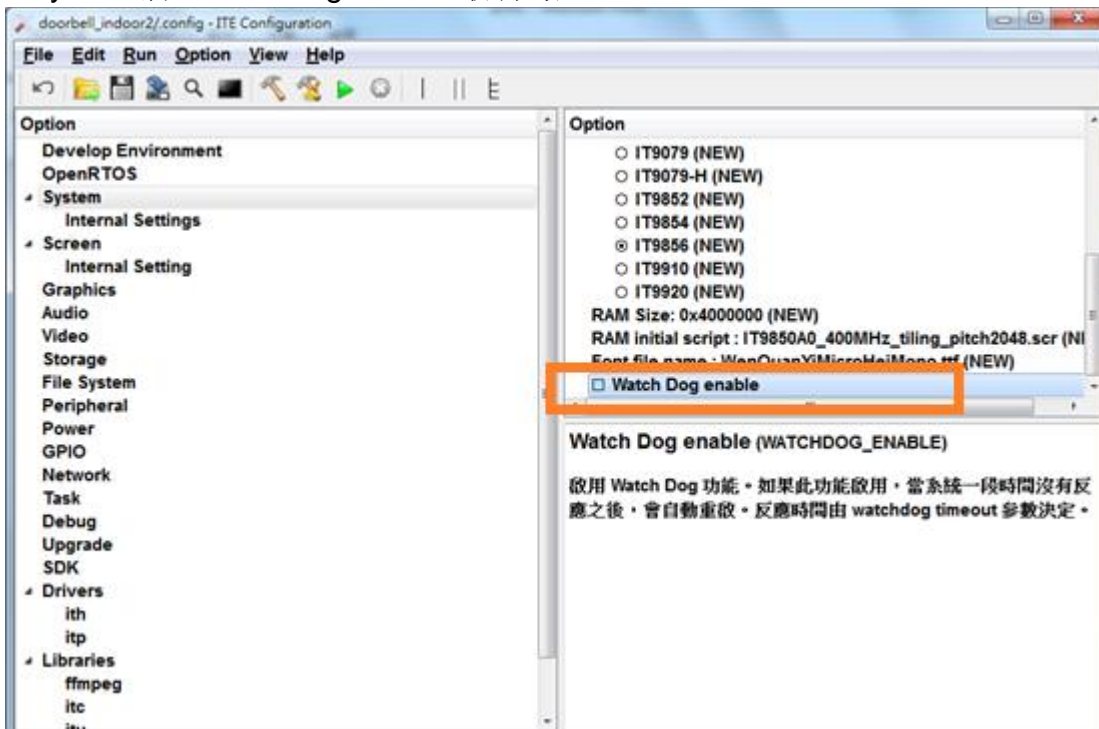
7.IT9850_Series使用gdb Debug Tool(以Indoor2為例)

7.1 Build gdb Debug 軟件

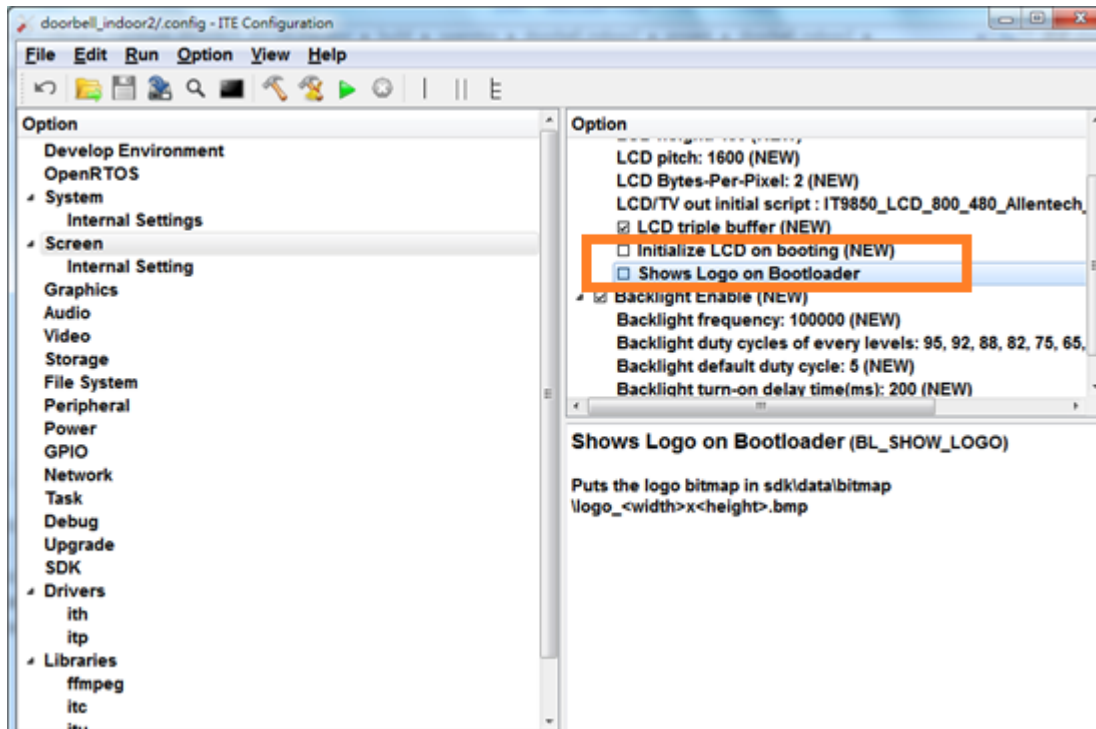
1.在 System/Internal Settings 將”Memory Debug enable”取消勾選。



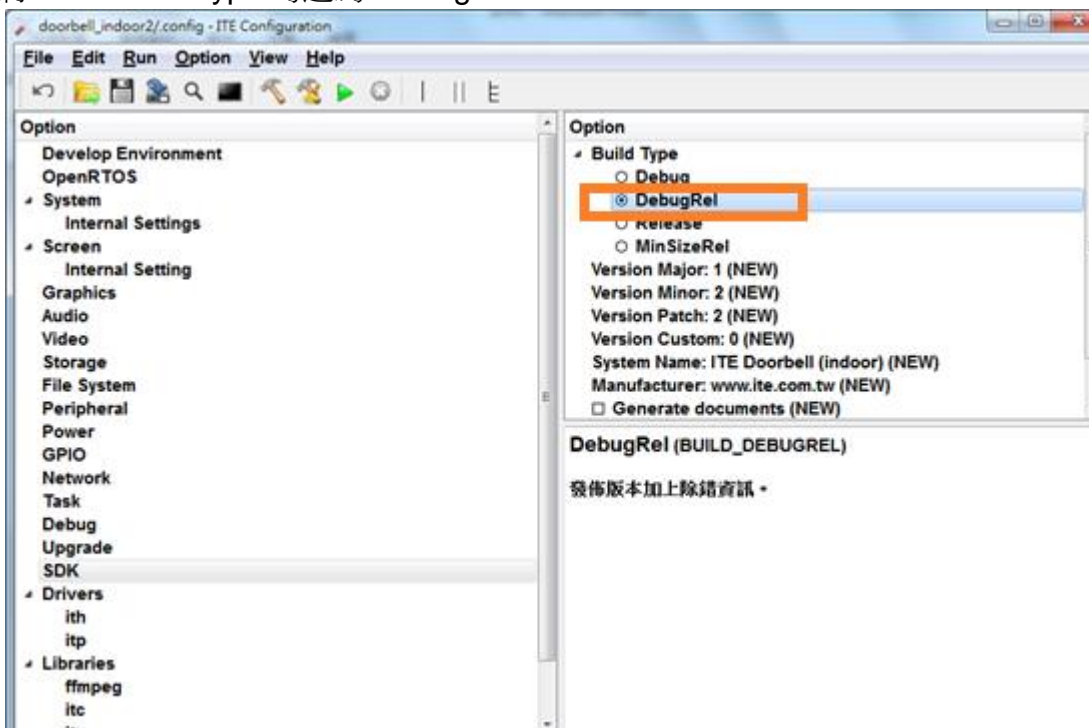
2.在 System 將”Watch Dog enable”取消勾選



3. 在 Screen/LCD enabel 將”Initialize LCD on booting”/”Shows Logo on Bootloader”都取消勾選。



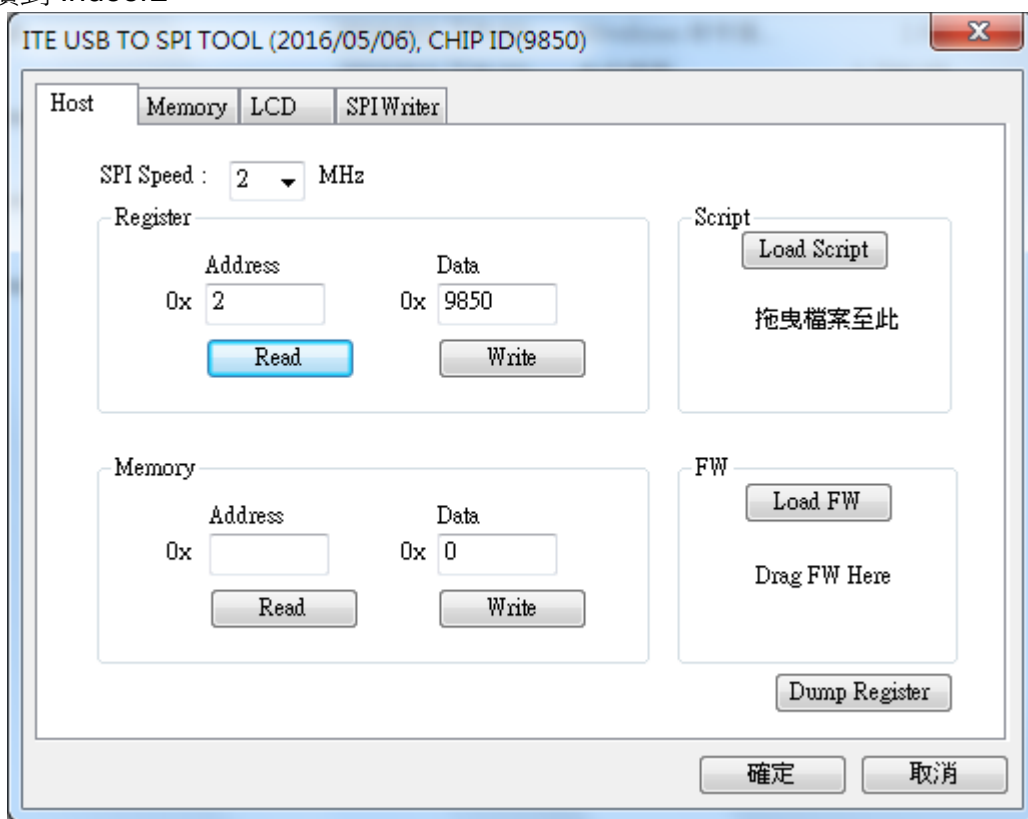
4. 將 SDK/Build Type 勾選為”DebugRel”。



5. build 出來的 image 需先升級至 Indoor2 target board。

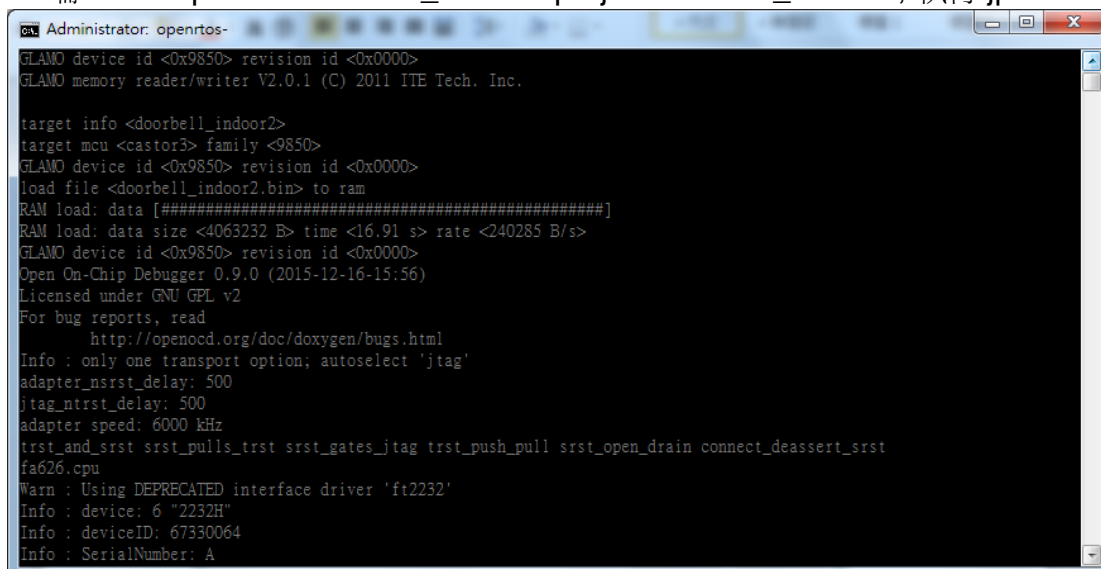
7.2 SPI 切換連接方式

將 Indoor2 切為 SPI mode,並接上 SPI 轉板後開機,並使用 USB-to-SPI Tool 確認 SPI-Mode 可讀到 Indoor2。



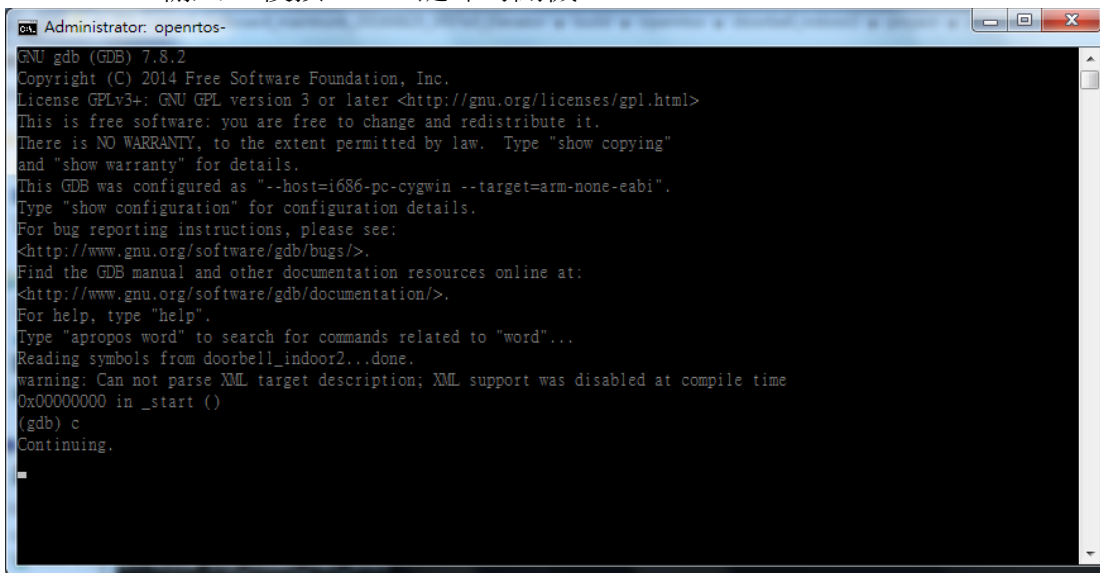
7.3 執行 jp2.cmd

在 PC 端\\build\\openrtos\\doorbell_indoor2\\project\\doorbell_indoor2, 執行 jp2.cmd。



7.4 執行 gdb.cmd

在 PC 端 `\\build\openrtos\doorbell_indoor2\project\doorbell_indoor2`, 執行 `gdb.cmd`, 並在 (gdb) command line 輸入 "c" 後按 Enter 鍵即可開機。



```
Administrator: openrtos-
GNU gdb (GDB) 7.8.2
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-cygwin --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from doorbell_indoor2...done.
warning: Can not parse XML target description; XML support was disabled at compile time
0x00000000 in _start ()
(gdb) c
Continuing.
```

Note.已知可作 Debug 功能:

- 1.進入(gdb) command
- 2.在(gdb) command line 下指令: c

攔中斷點方式:1.進入(gdb) command

- 2.在(gdb) command line 下指令: b -NOR_cache_writeSectors
- 3.在(gdb) command line 下指令: where

7.5 開機過程

在(gdb) command line 下指令: c 按 Enter 鍵後,開機後可到主畫面即開機完成。

Note.開機過程沒有 ITE logo。

範例 - call stack 被破壞

- 1.先將 break point 設在 data abort 的 exception address。

(gdb) b *0x10

- 2.開始執行

(gdb) c

- 3.將 PC 設到 data abort 的發生點 LR (link register)

(gdb) set \$pc = \$lr

- 4.看 call stack

(gdb) where

```
GNU openrtos-
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-cygwin --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from test_lcd...done.
0x00000000 in start (<)
(gdb) b *0x10
Breakpoint 1 at 0x10
(gdb) c
Continuing.

Breakpoint 1. 0x00000010 in start (<)
(gdb) set $pc = $lr
(gdb) where
#0  0x0000273c in TestFunc (arg=<optimized out>)
    at D:/USS/ITE_Castor3/project/test_lcd/test_lcd.c:35
#1  0x65b23774 in ?? (<)
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb)
```