Lab 1 : hands-on SSD 10/4

江冠德 P76124493@gs.ncku.edu.tw
方呈祐 NE6121107@gs.ncku.edu.tw
CSSLAB 65603

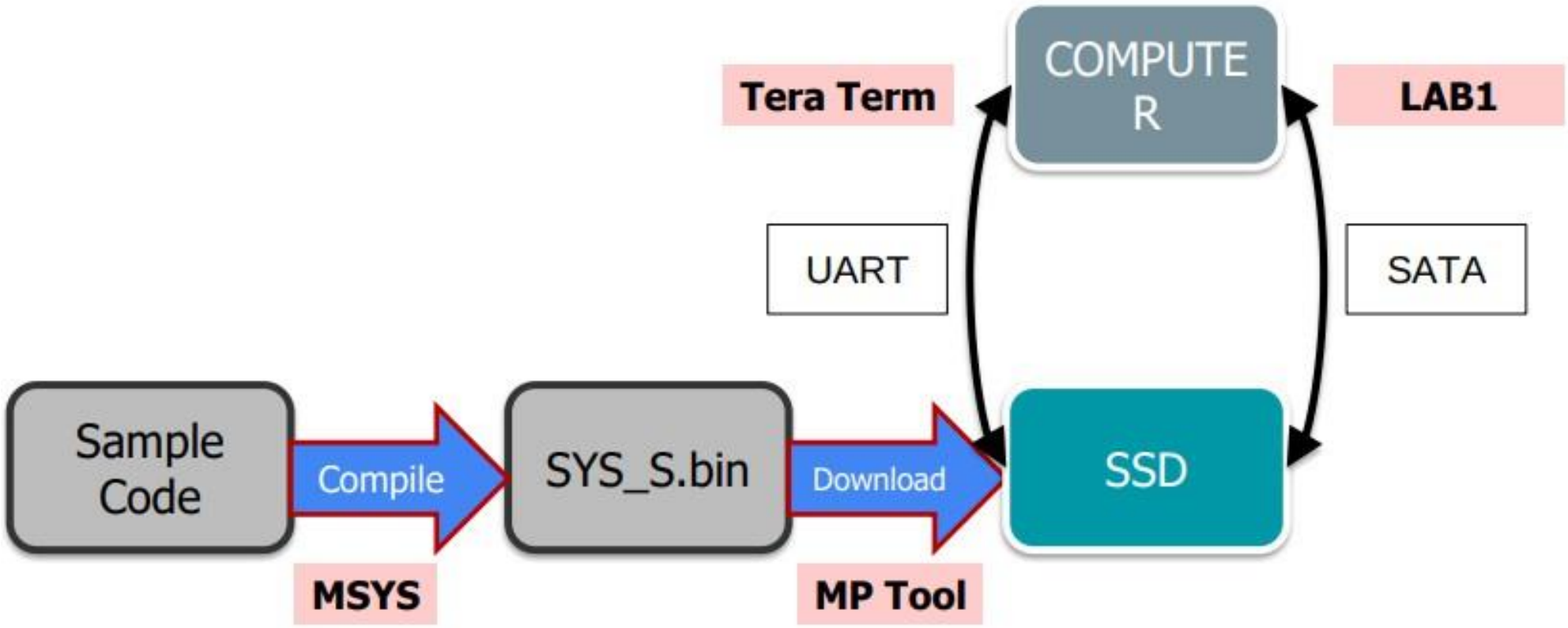
Goals

1. Setup the environment for the firmware development

2. Understand the steps of downloading firmware & initailizing SSD

3. Message printing through UART

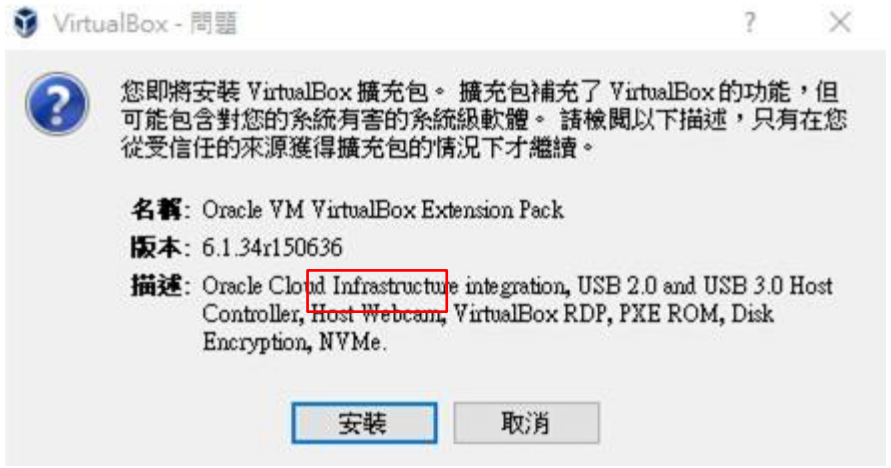
Overview

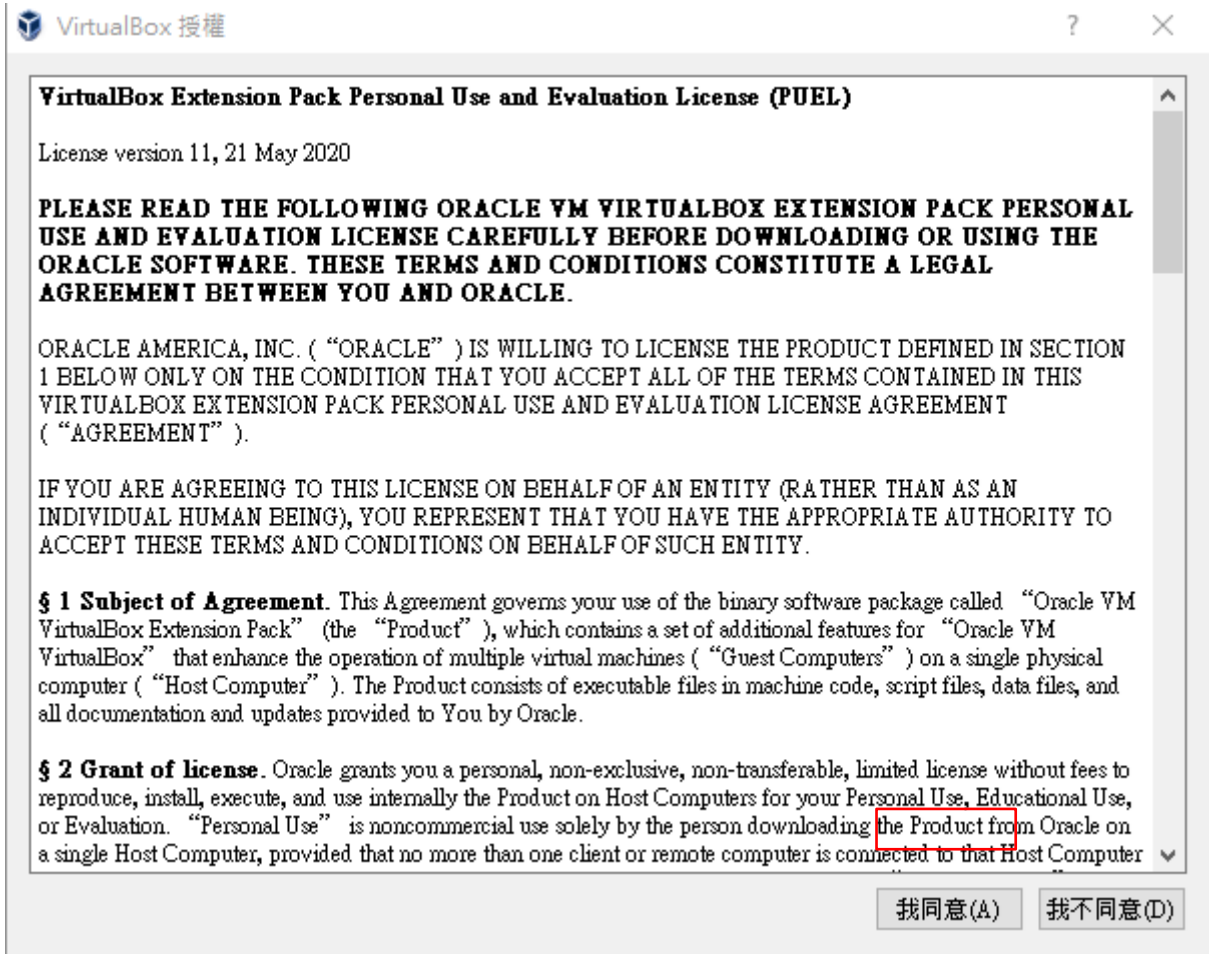

Tera Term
COMPUTE R
LAB1
UART
SATA
Sample Code
Compile
SYS_S.bin
Download
SSD
MSYS
MP Tool

安裝VirtualBox-6.1.34a-150636-Win

- 下一步直到結束

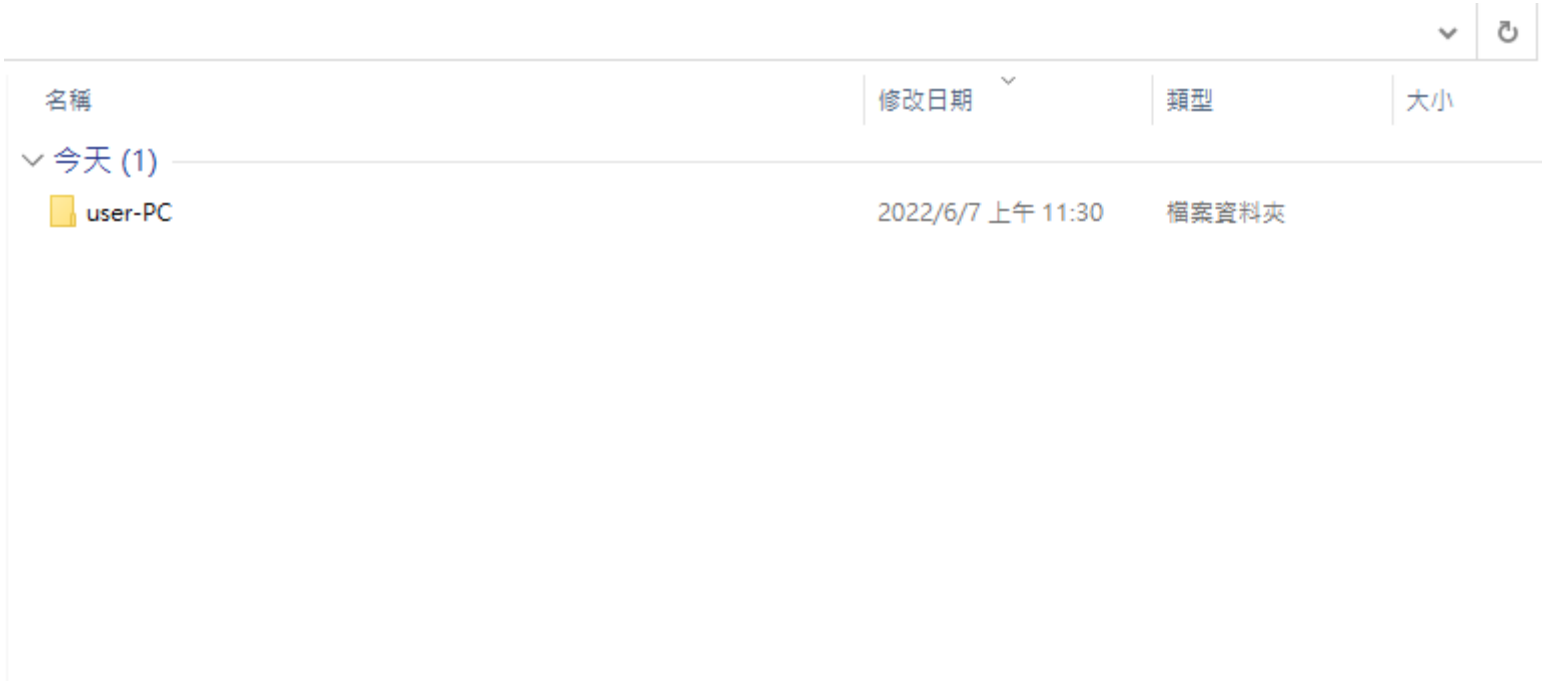
安裝Oracle_VM_VirtualBox_Extension_Pack-6.1.34





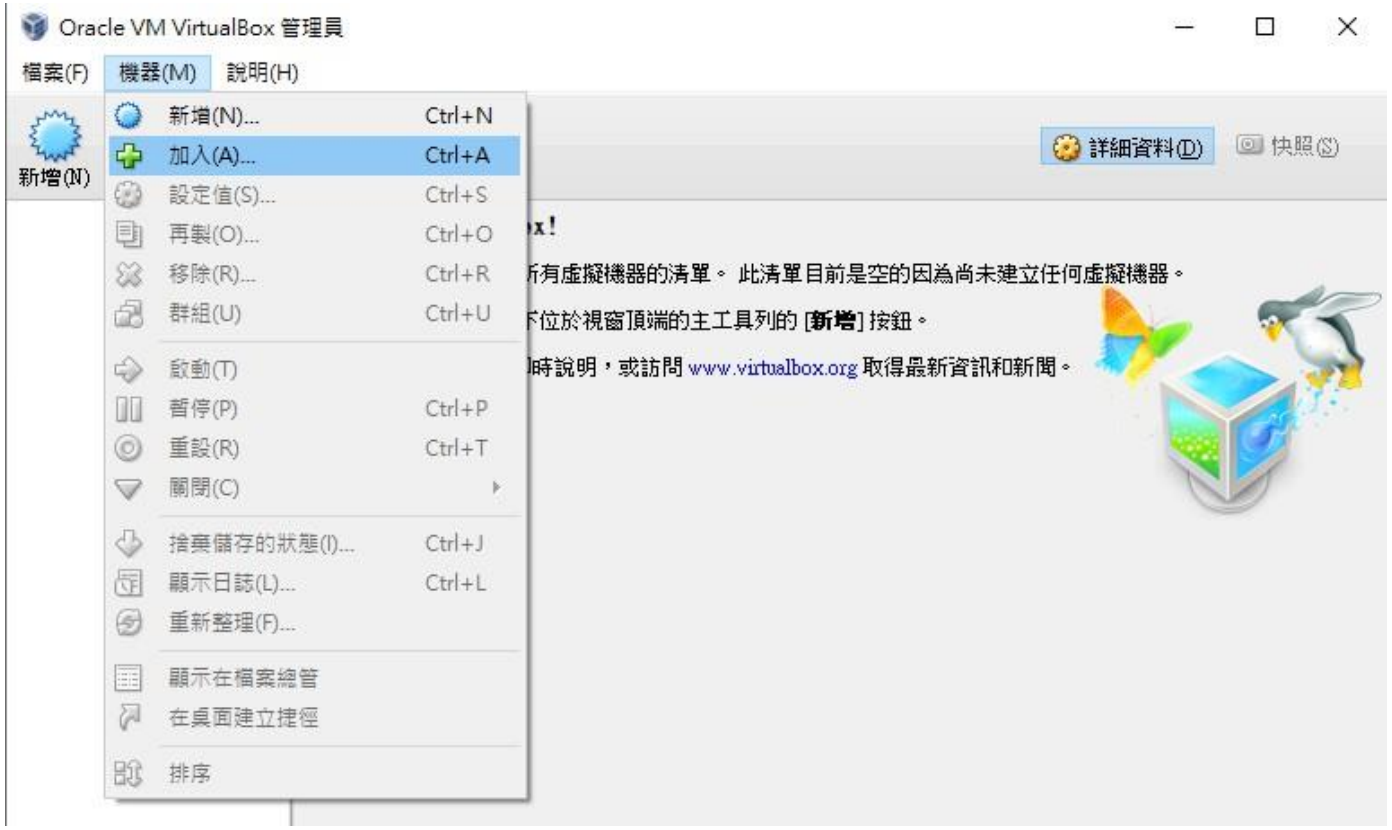
安裝步驟

- 請確認絕對路徑中無中文



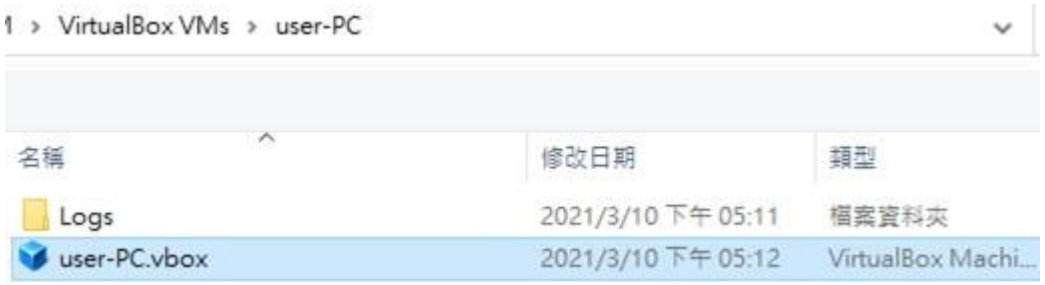
安裝步驟

- 開啟安裝好的Oracle VM VirtualBox並選擇　機器->加入



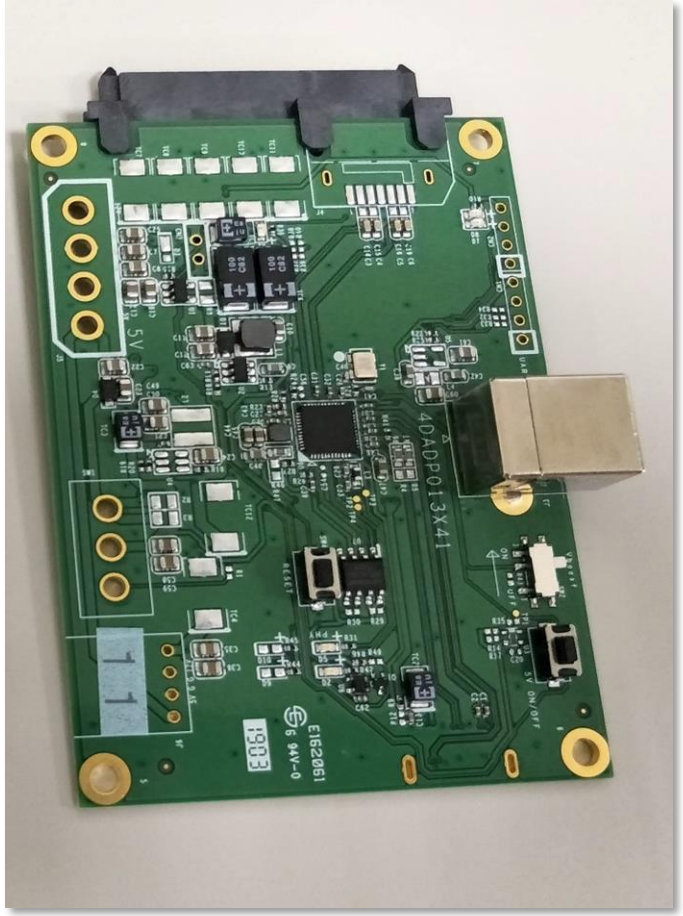
安裝步驟

- 選擇user-PC內的user-PC.vbox，就會產生一個新的虛擬機



硬體設備介紹

Flash

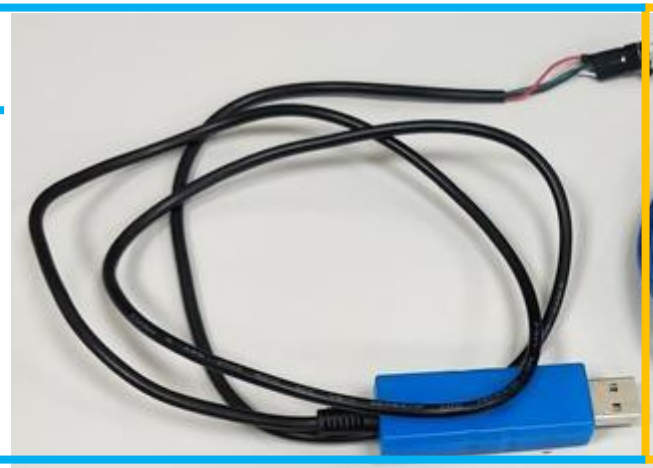


轉接板

硬體設備介紹

UART線

連接flash，顯示log。



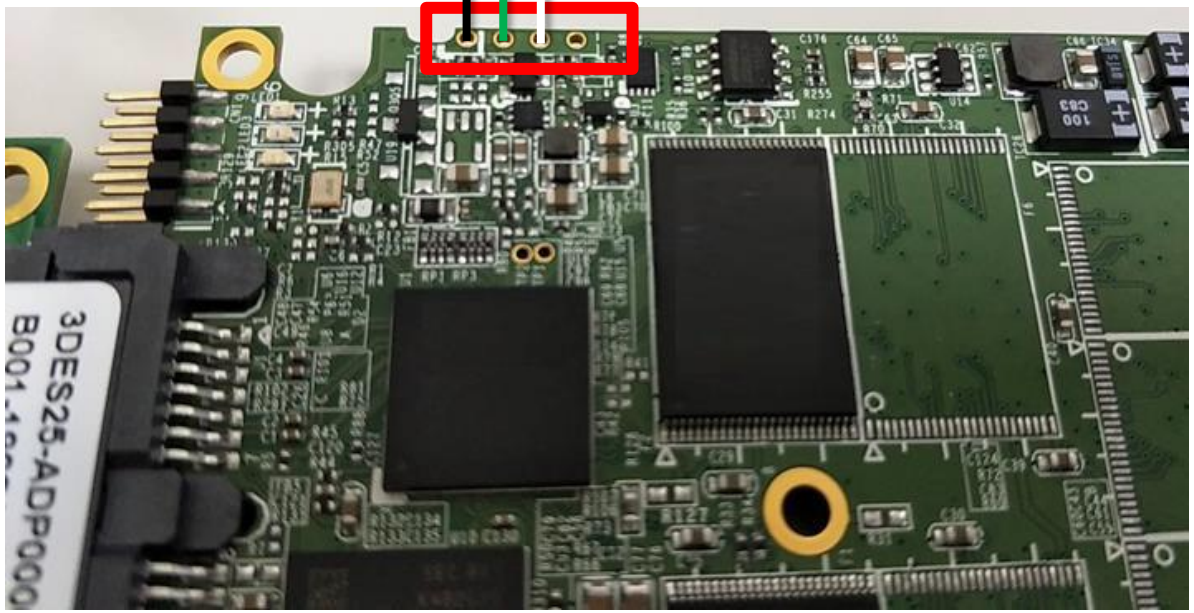
轉接線

USB轉SATA，供電及傳輸

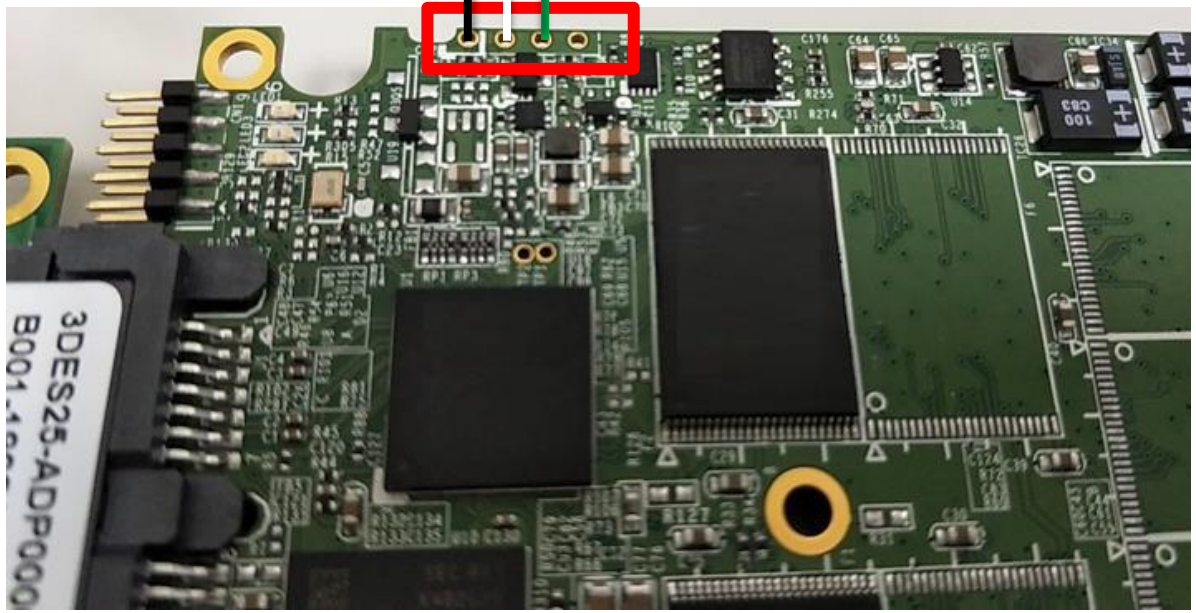
UART使用

- 請於斷電狀態下連接UART。
- 依照顏色插到對應的孔，接UART時要注意顏色不可以接反。
- 紅色的線為電源線不可接到Flash上。
- 若開卡時黑綠白接線UART沒跳東西，請嘗試使用黑白綠接線

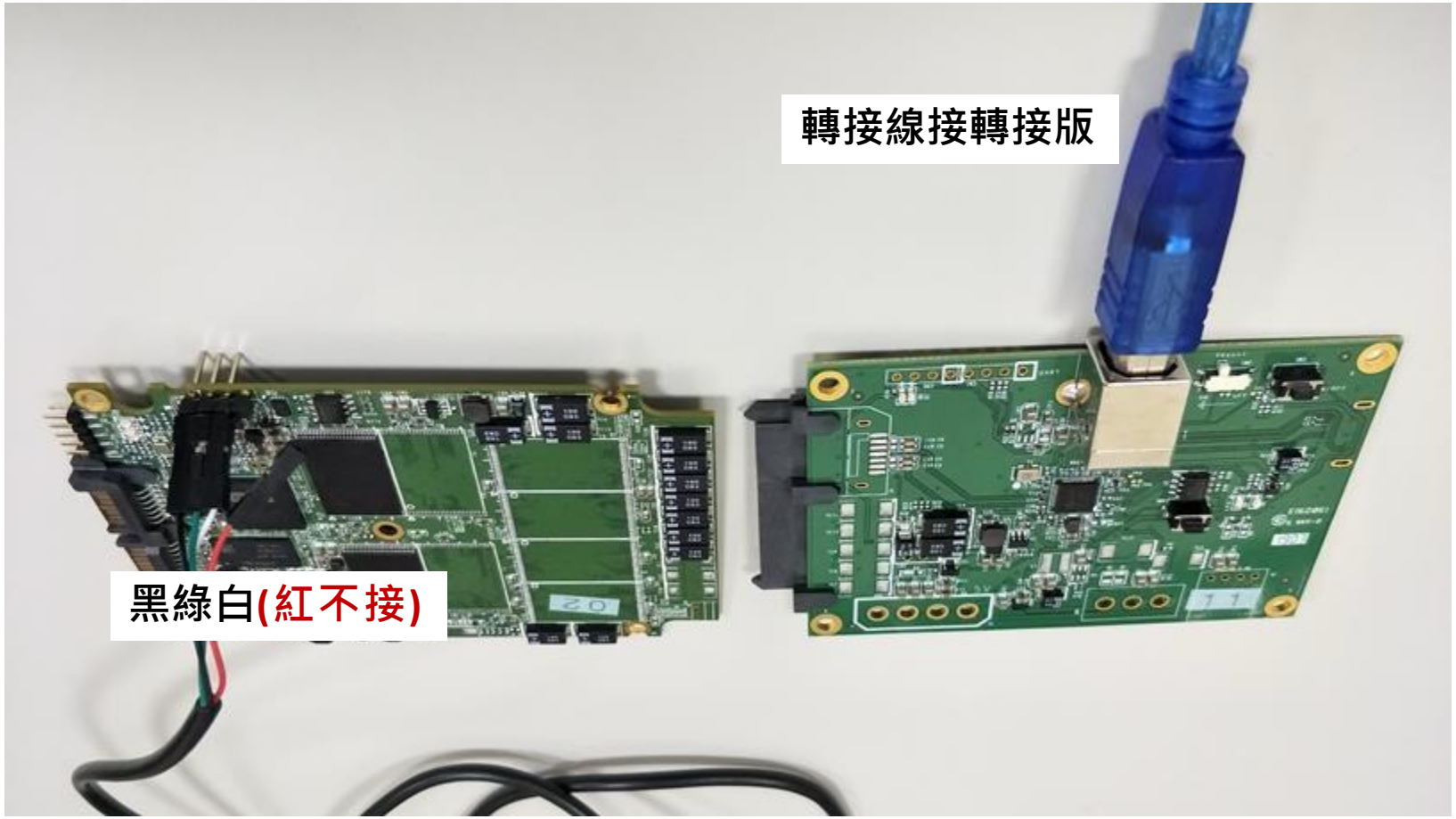
黑綠白



黑白綠



接線



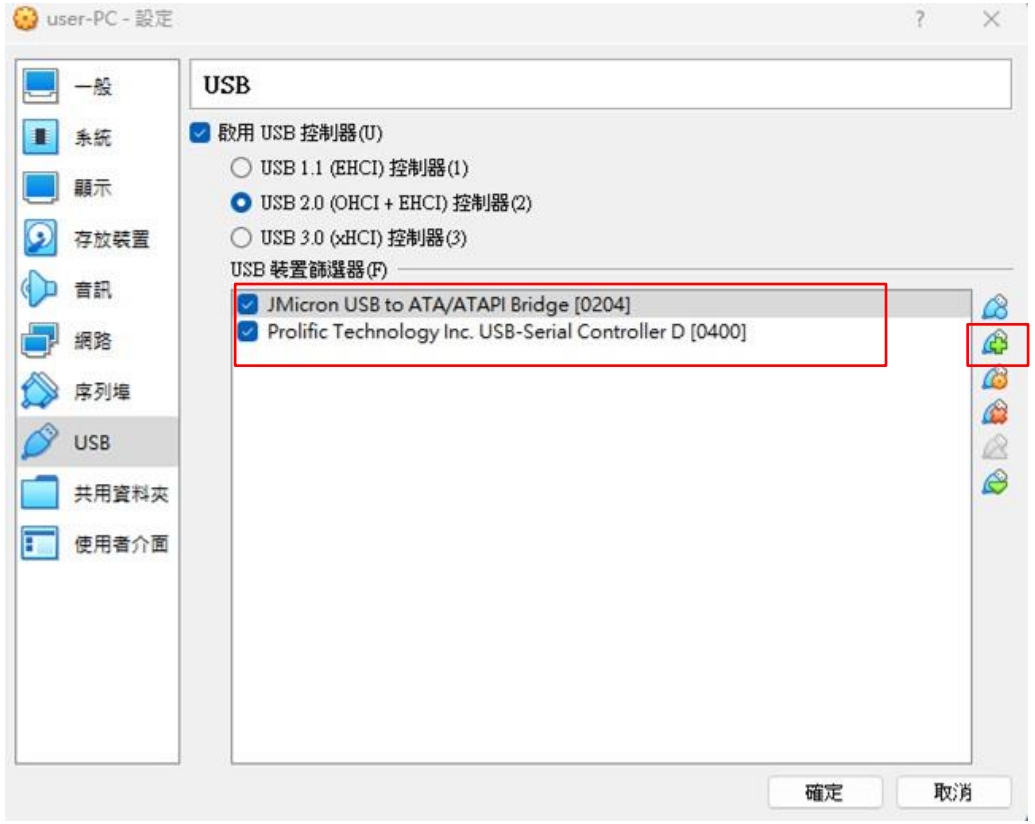
上電

1. 將轉接線及UART都接上板子
2. 兩個版子都連接電腦的USB
3. 轉接版和Flash做連接

安裝步驟

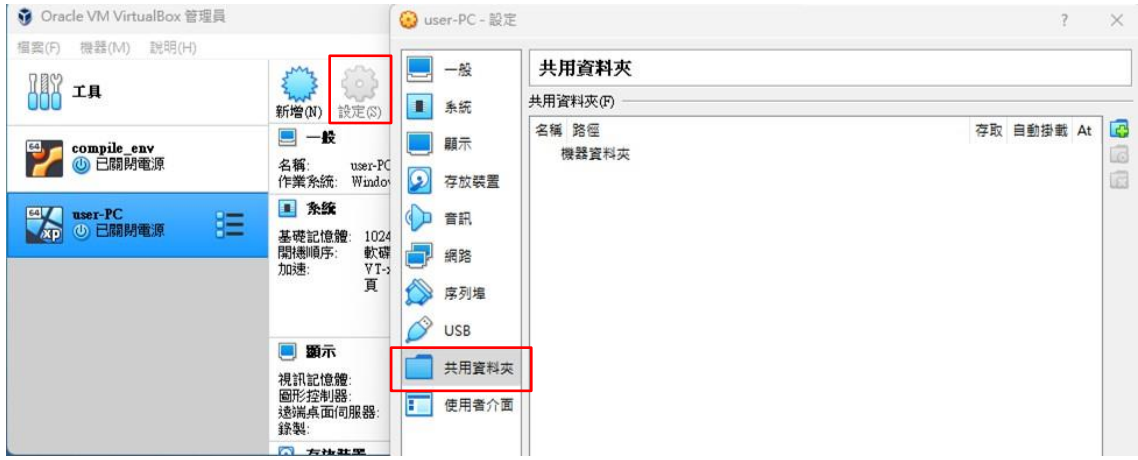
- 接上開發板USB口 以及 UART線
  - ✓紅色的不要接，顏色一定要對

- 進入設定值後在USB選項點選右方分別加入圖示的USB裝置



共享資料夾設置

- 按下設定，選取共用資料夾

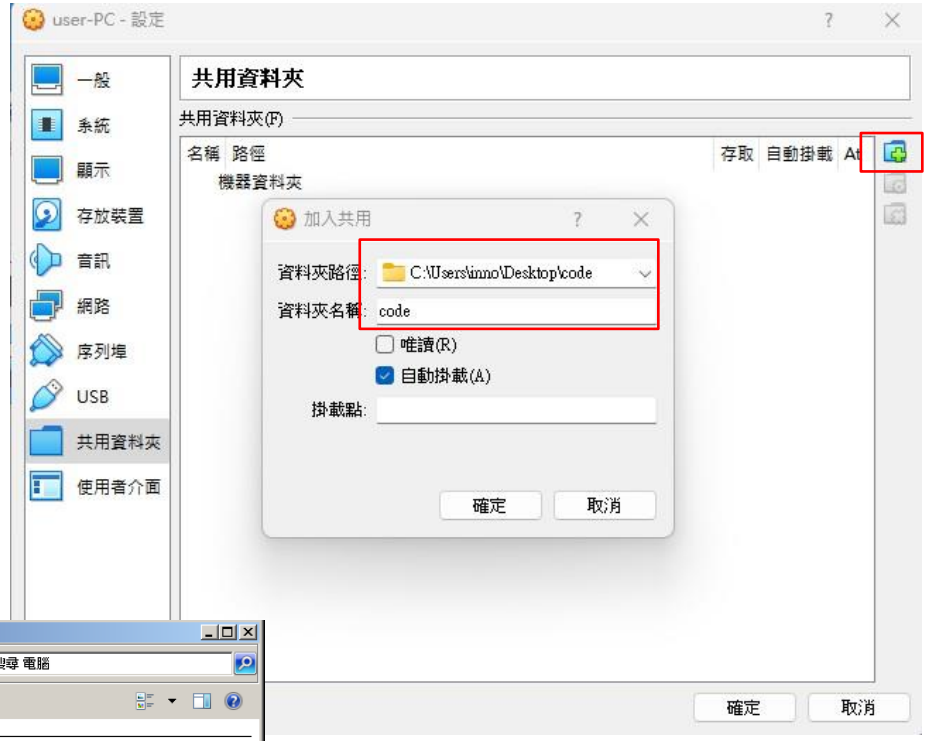


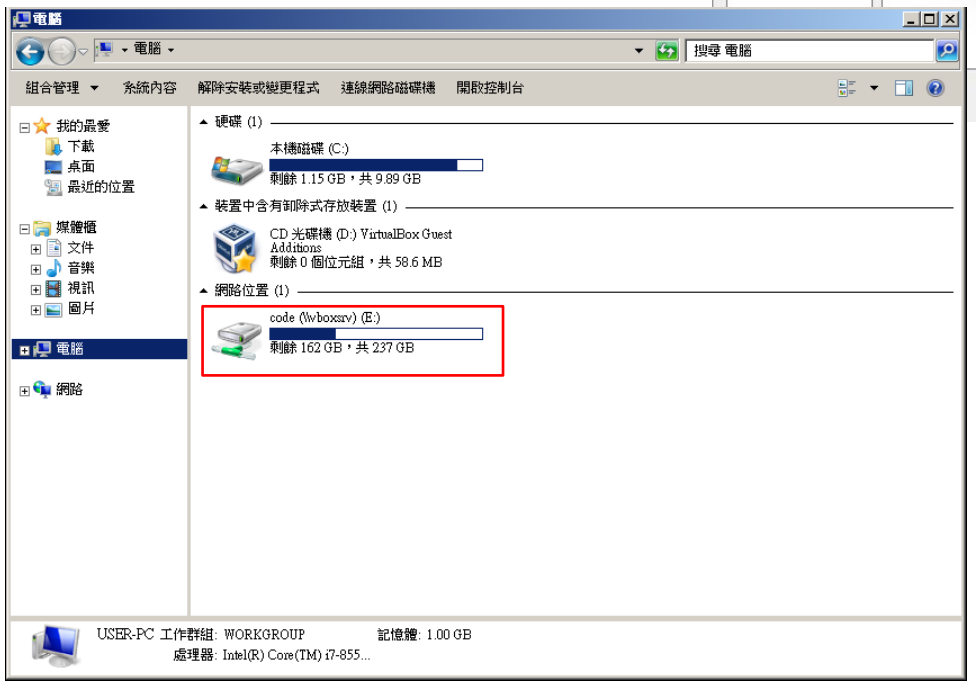
- 桌面先建立一個資料夾，裡面放code的檔案

共享資料夾設置

- 選取剛剛建立的資料夾

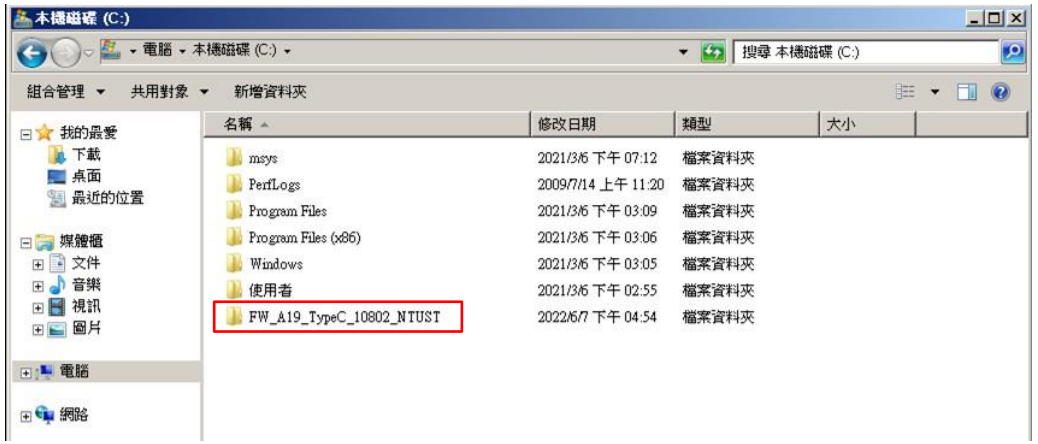
- 開啟虛擬機，即可看到資料夾
(如果有資料共享，只需把資料放在這個資料夾即可)



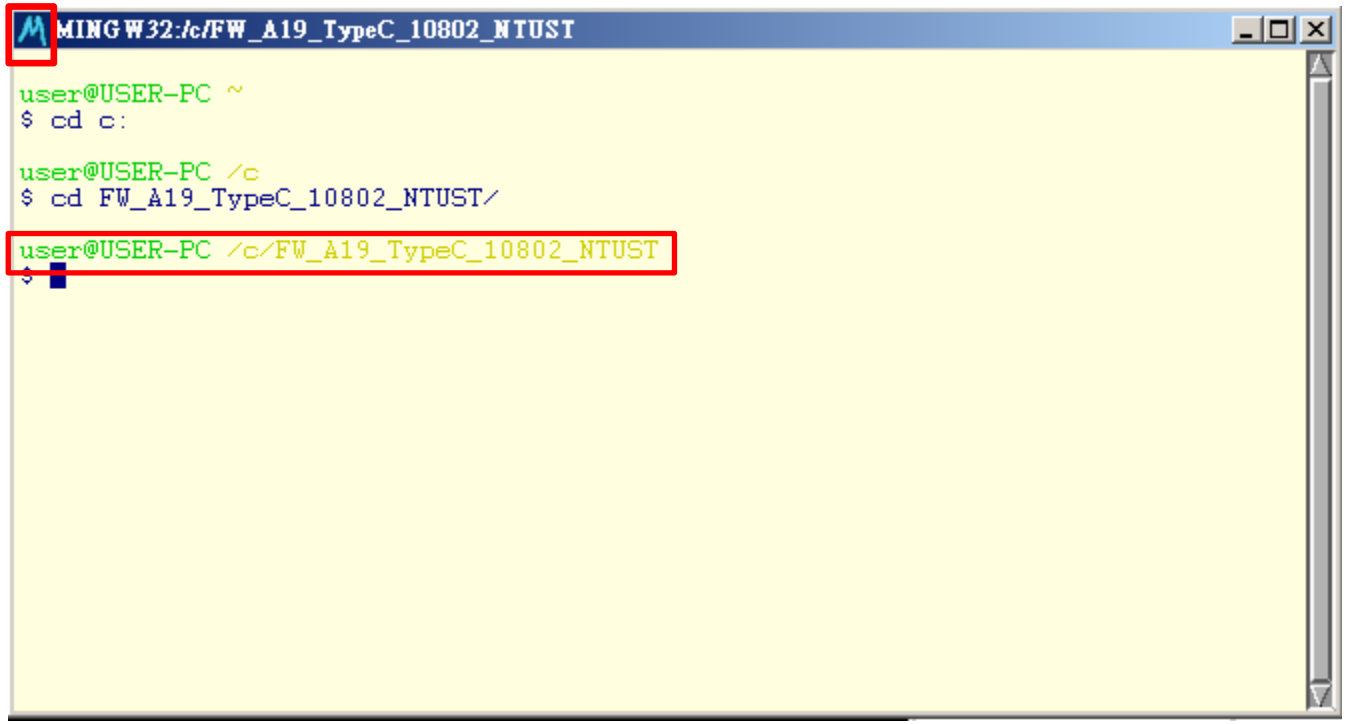


程式編譯

- 把程式從共享資料夾複製到C槽



- 打開桌面上的MSYS，把路徑改到程式所在位置



程式編譯

- 下 make cleandep 的指令

```
user@USER-PC /c/FW_A19_TypeC_10802_NTUST
$ make cleandep
> include/autocfg.h
cd drive; make cleandep
make[1]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/drive'
rm -f
> ./depend.mk
make[1]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/drive'
cd bin; make cleandep
make[1]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin'
for x in rvct/Makefile; do cd `dirname $x`; make cleandep -f `basename $x`; cd /
c/FW_A19_TypeC_10802_NTUST/bin; done
make[2]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin/rvct'
rm -f rvctISR.d rvctRetarget.d
> ./depend.mk
make[2]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin/rvct'
for x in  sys/Makefile; do cd `dirname $x`; make cleandep -f `basename $x`; cd /
c/FW_A19_TypeC_10802_NTUST/bin; done
make[2]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin/sys'
rm -f main.d bget.d rvctSubMain.d uart_cmd_channel.d ftl1.d ftl_wbm.d dpd.d dram
_init.d SYS_S.d
#- > SYS_S.d
make[2]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin/sys'
make[1]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin'
```

- 接著下make指令

```
$ make
> include/autocfg.h
cd drive; make dep
make[1]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/drive'
make[1]: Nothing to be done for `dep'.
make[1]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/drive'
cd bin; make dep
make[1]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin'
for x in rvct/Makefile; do cd `dirname $x`; make dep -f `basename $x`; cd /c/FW_
A19_TypeC_10802_NTUST/bin; done
make[2]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin/rvct'
make[2]: *** No rule to make target `rvctISR.d', needed by `dep'.  Stop.
make[2]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin/rvct'
for x in  sys/Makefile; do cd `dirname $x`; make dep -f `basename $x`; cd /c/FW_
A19_TypeC_10802_NTUST/bin; done
make[2]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin/sys'
/c/FW_A19_TypeC_10802_NTUST/make/rules_image.mk:125: SYS_S.d: No such file or di
rectory
> SYS_S.d
make[2]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin/sys'
make[2]: Entering directory `/c/FW_A19_TypeC_10802_NTUST/bin/sys'
armcc   -I./ -I/c/FW_A19_TypeC_10802_NTUST/include -DPROG_ROOT_PATH=\"/c/FW_A19_
TypeC_10802_NTUST\" -DUSER_NAME=\"user\" main.c -M -o main.o --depend=main.d --d
epend_format=unix_escaped
#-sed /main\.d/d ./SYS_S.d > ./depend.tmp
grep --invert-match "main.d" ./SYS_S.d > ./depend.tmp
make[2]: [main.d] Error 1 (ignored)
cat ./depend.tmp > ./SYS_S.d
rm -f ./depend.tmp
make[2]: *** No rule to make target `bget.d', needed by `dep'.  Stop.
make[2]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin/sys'
make[1]: Leaving directory `/c/FW_A19_TypeC_10802_NTUST/bin'
```

程式編譯

- 如果有build成功 FW_A19_TypeC_10802_NTUST\bin\sys路徑下的SYS_S.bin已被更新

- 如果沒有代表code有錯誤->debug
  - ✓可以注意上面的error information

UART顯示Tool

- 接上硬體後，開啟桌面的Tera Term

- 選擇Serial及對應的COM
  - 若無法選取請見下頁

UART顯示Tool

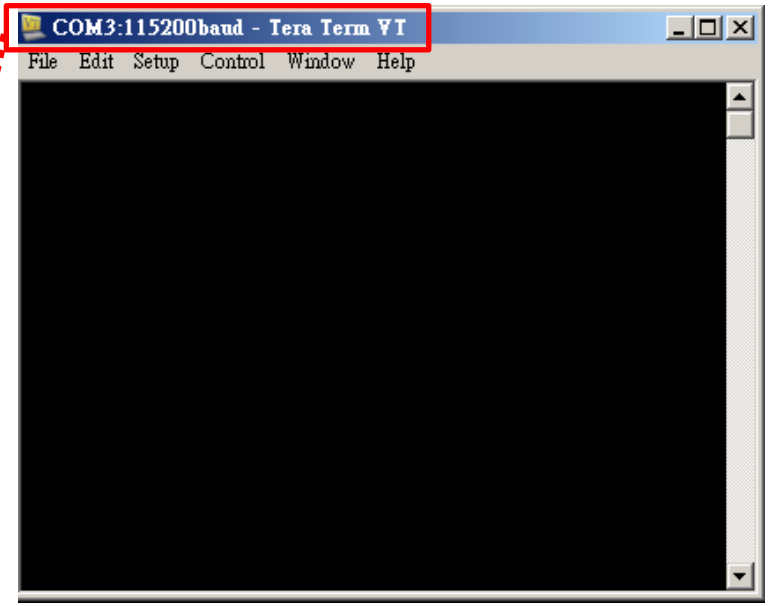
- 有些同學Tera Term打開以後會無法選取COM3

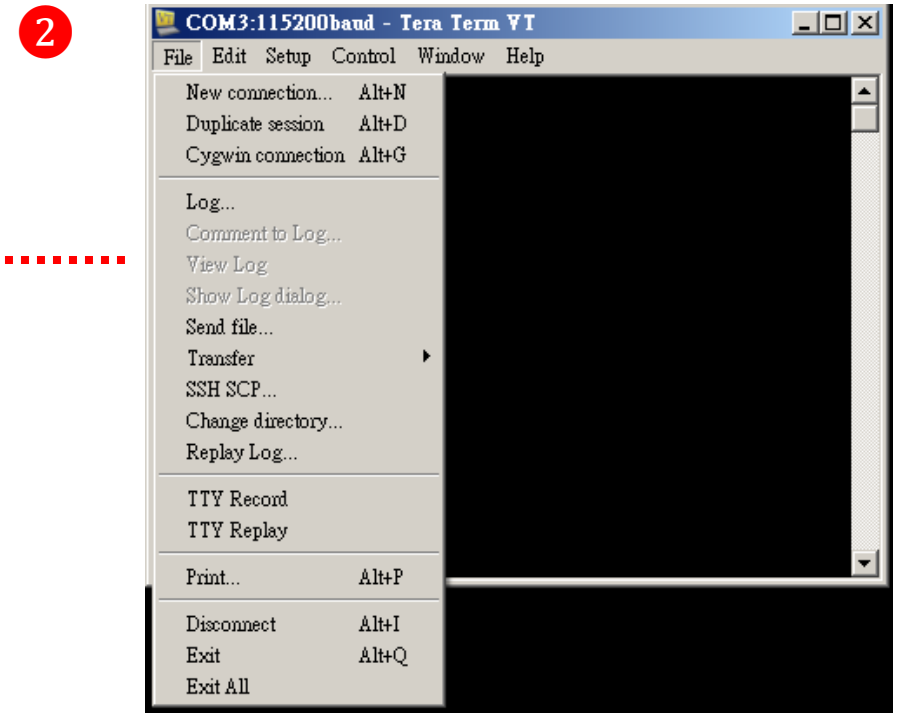
① 可以先看一下Tera Term視窗的標題是否有有出現COM3

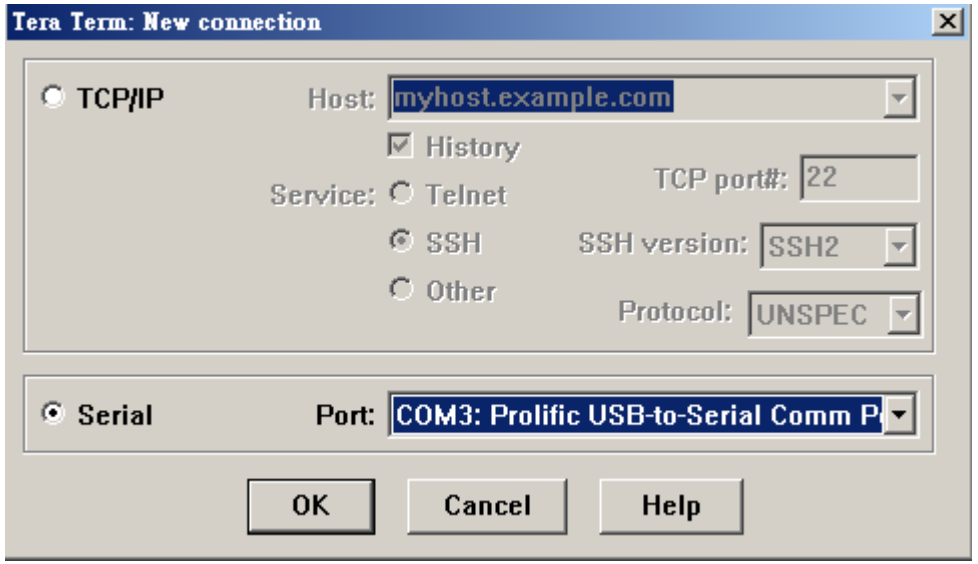
② 若有出現我們需要先disconnect

- File → Disconnect

③ 之後再到New Connection就可以選擇COM3了





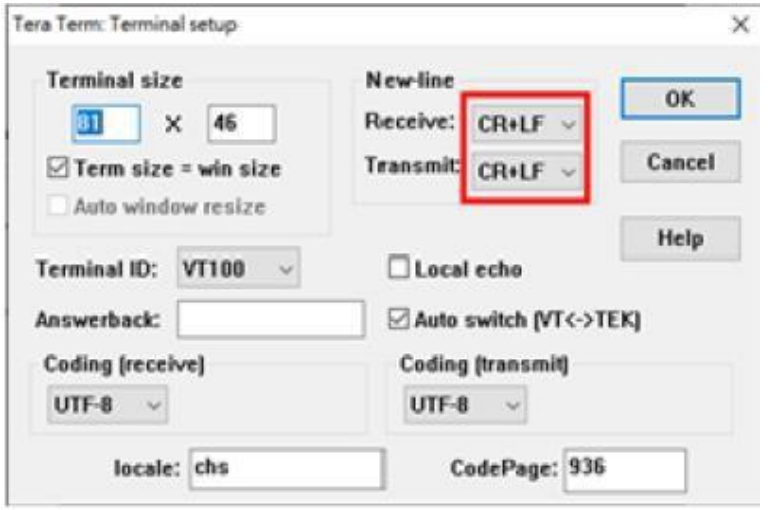


UART顯示Tool

- 點選Setup → Serial port
  - ✓Baud rate選擇115200(若顯示亂碼則改為19200)
  - ✓parity : none

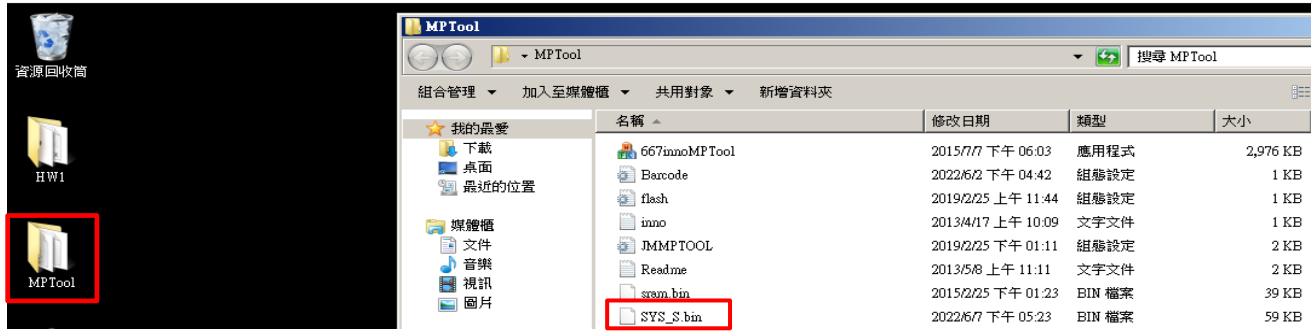
- 點選Setup → Terminal
  - ✓Receive 和 Transmit要設為[CR+LF]

- 可點選Setup → Save setup (下次就不用重做3~4步)



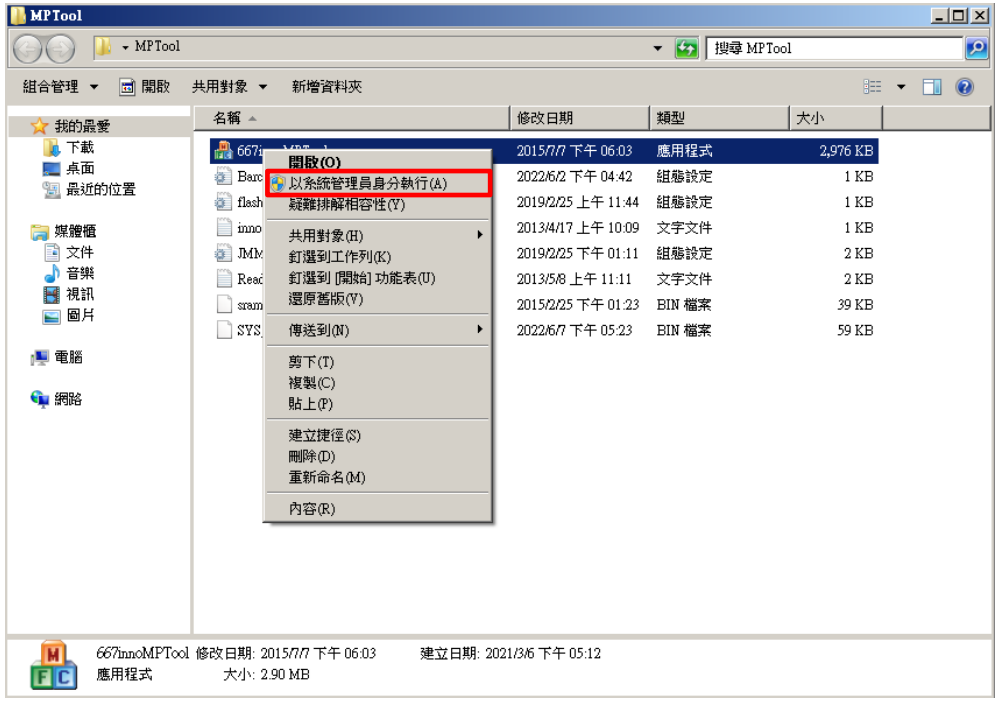
開卡流程

- 打開桌面的MPTool，SYS_S.bin複製到資料夾下



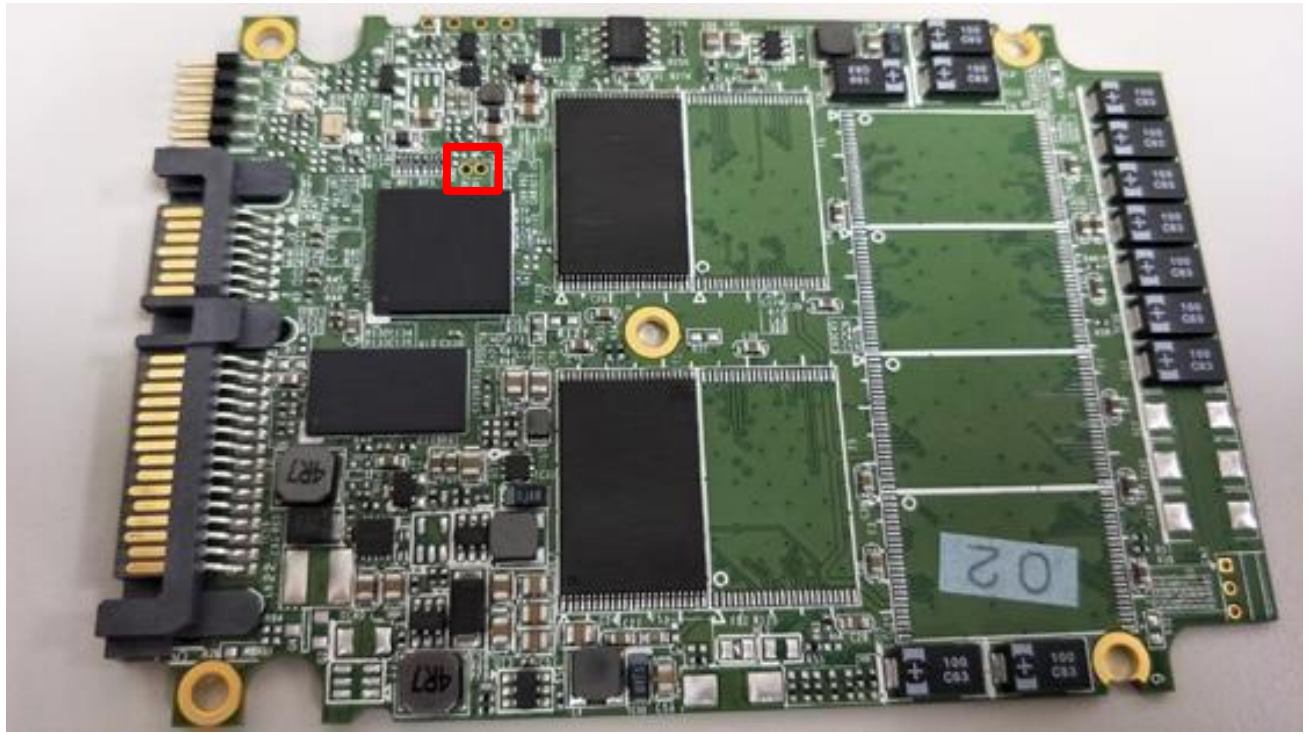
- 先斷電再短路，再上電

- 以系統管理員身分執行MPTooL



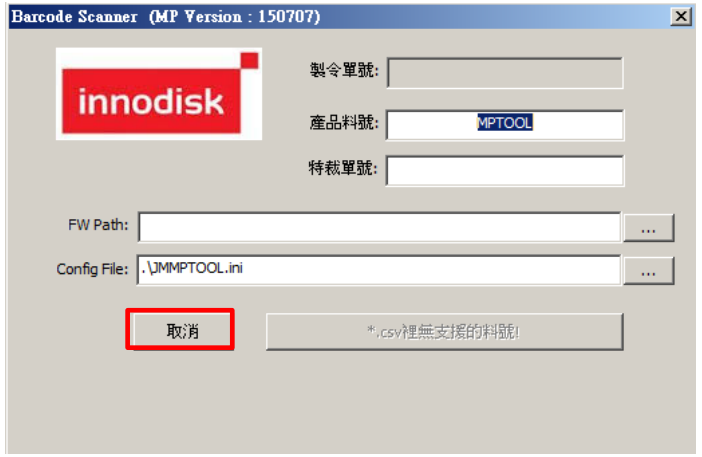
短路步驟

1. 斷電(不能接任何電源)
2. 拿短路工具接觸兩邊短路孔
3. 上電
4. 移開短路工具

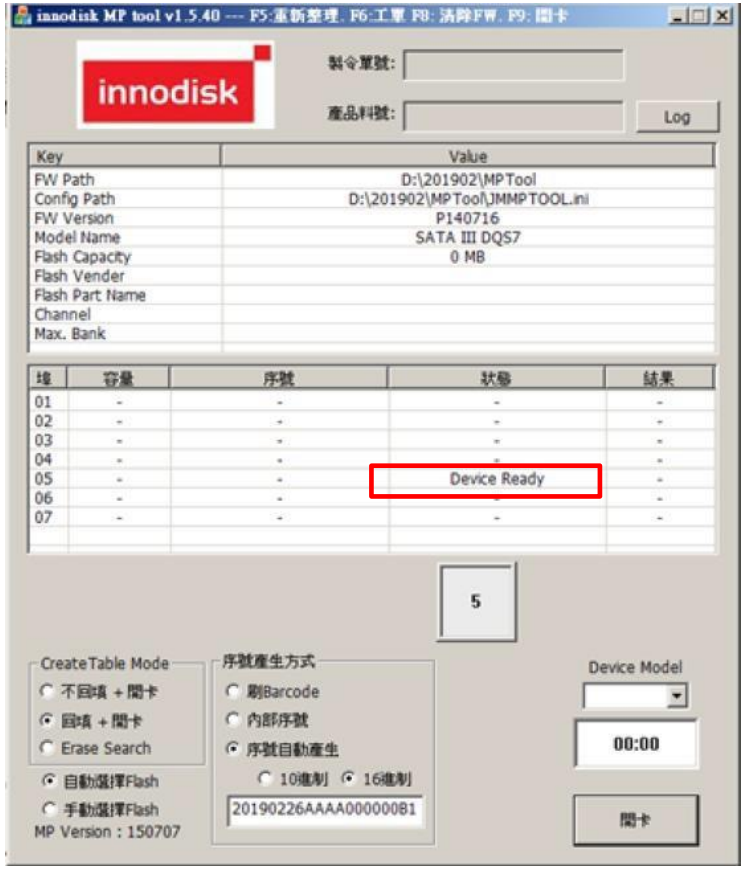


開卡流程

- 點選取消



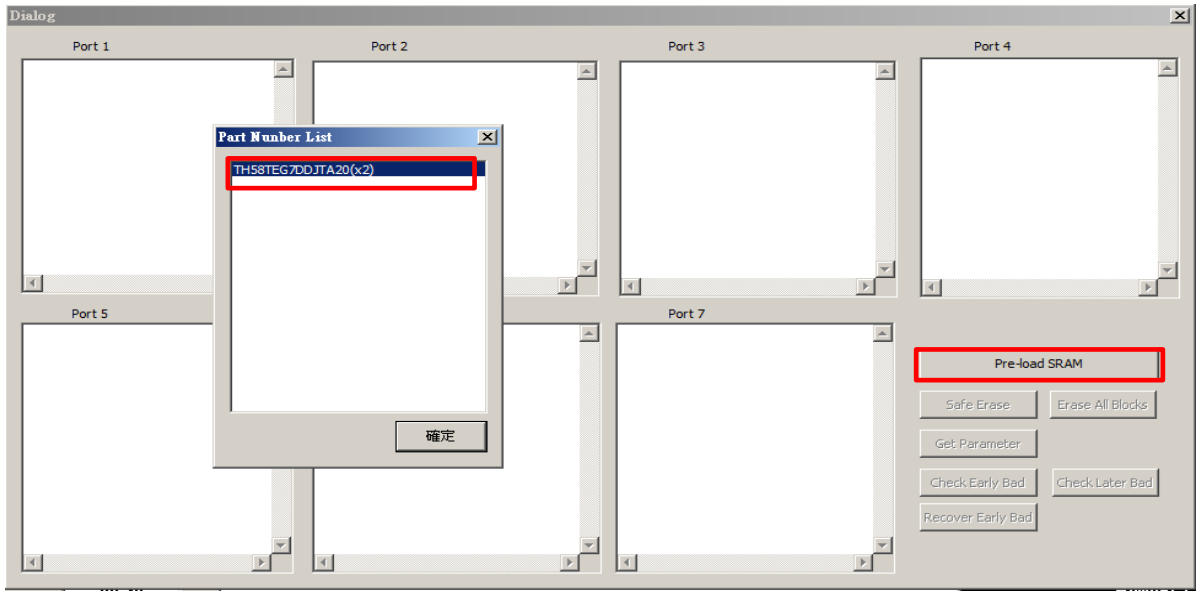
- 之後會跳出頁面如下右
  - ✓如未偵測到硬體，請換個轉板或是重新斷上電
  - ✓若還是沒有，請重新短路重新上電



開卡流程

- 按下F8→點選Preload SRAM→確定→選擇TH58TEG7DOJTA20(x2) 後關掉視窗
  - ✓開卡過程中不可以短路，要不然code會燒不進去

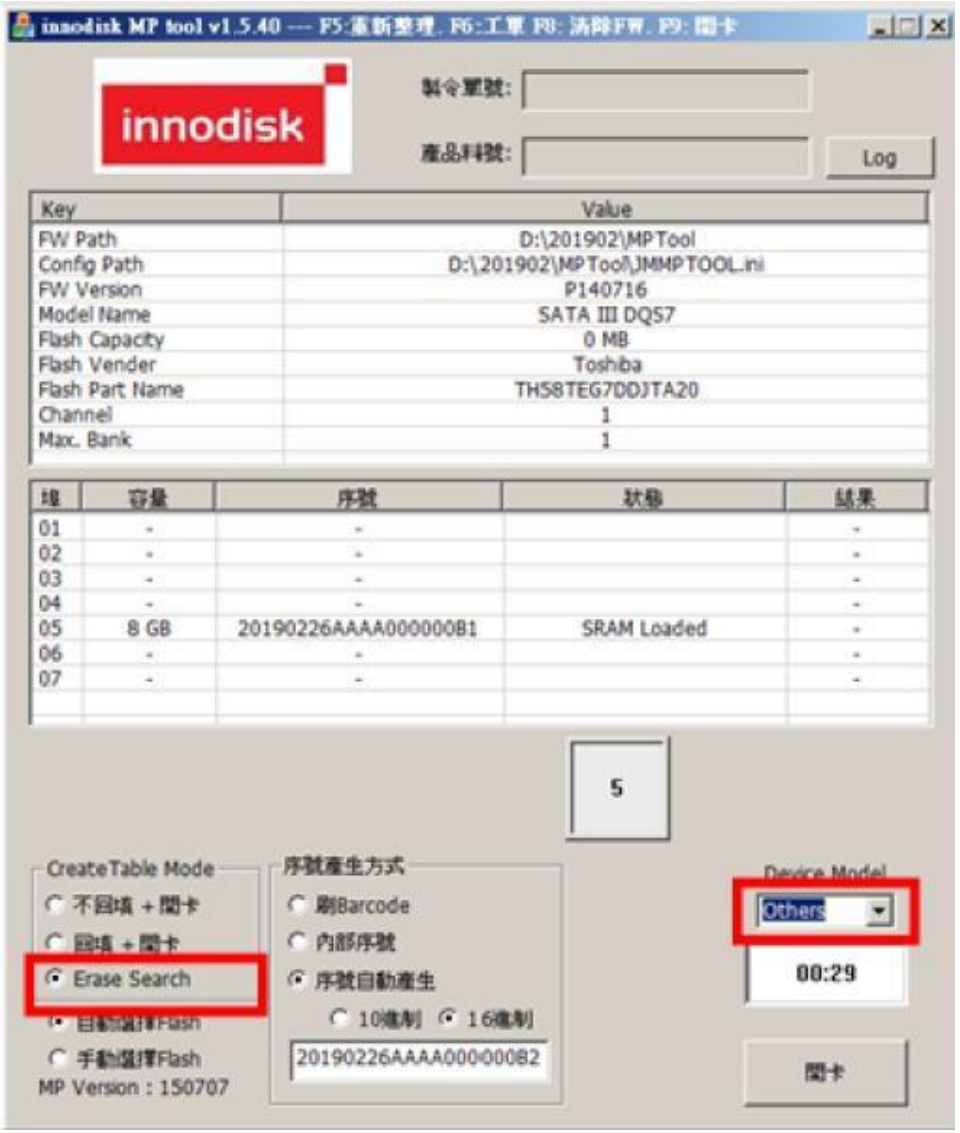
- 注意:此過程Tera Term都會跳uart(沒跳就是uart線接錯或是壞掉)



開卡流程

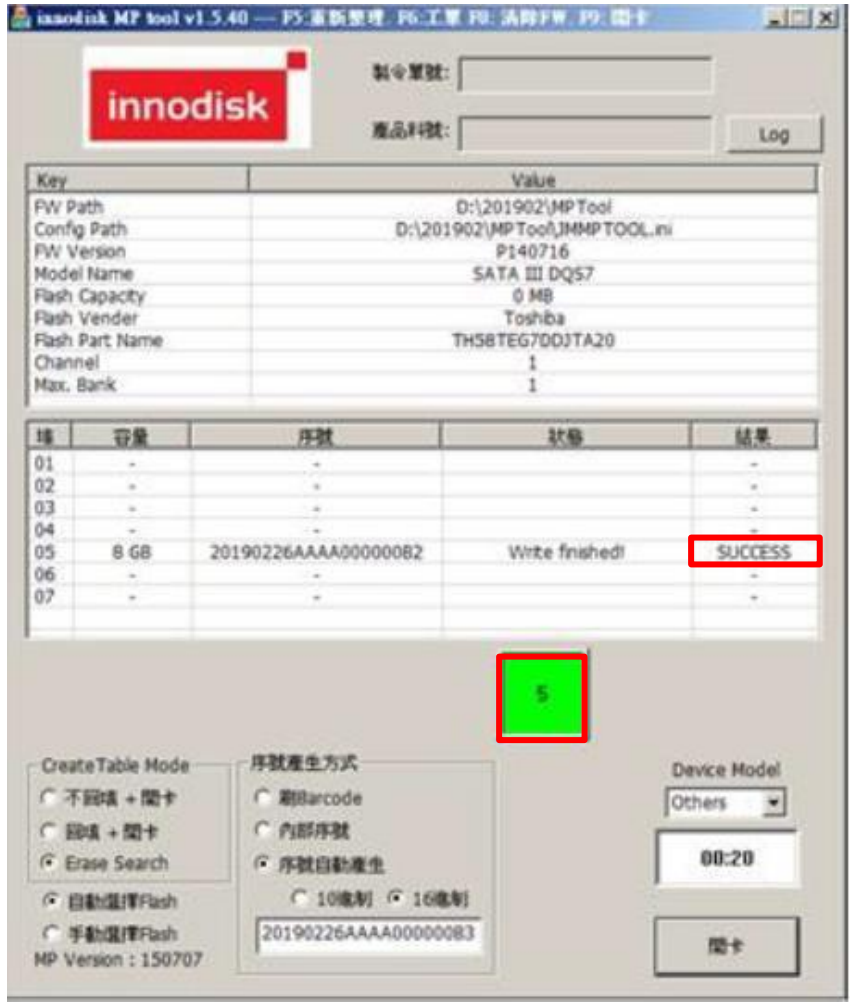
- Create Table Mode 選**Erase Search**，且Device Model選**Others(**一定要)，之後按下開卡

- 如果正常開卡在狀態列會顯示 SRAM Loader -> Build Map -> Save Bad -> Write Finished



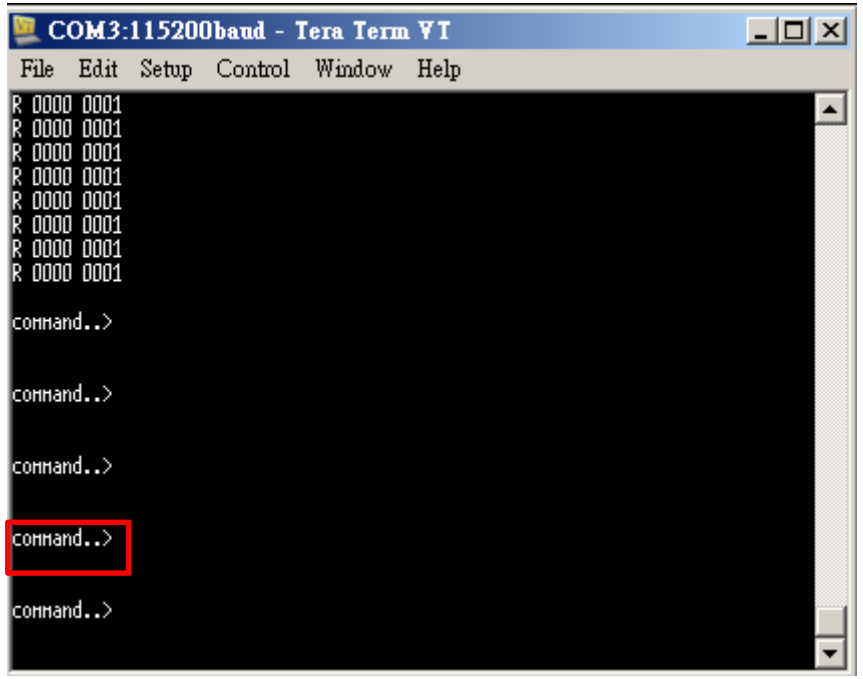
開卡流程

- 出現綠色框框就是成功了，若出現黃色或是一直不出現Success。

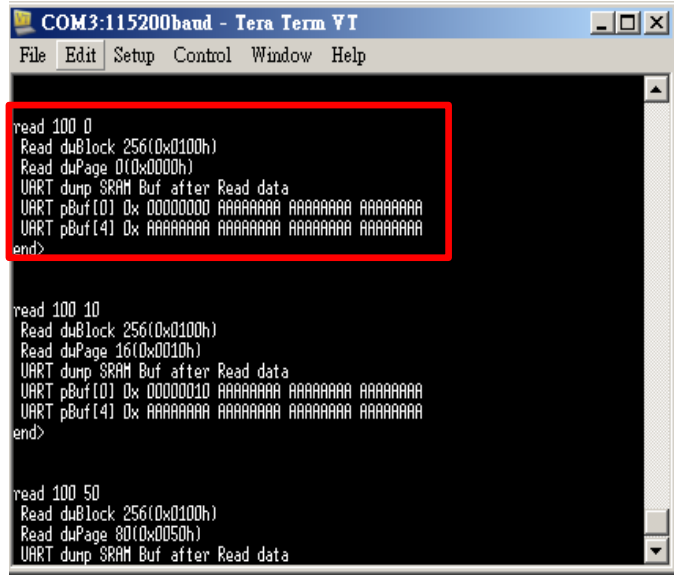


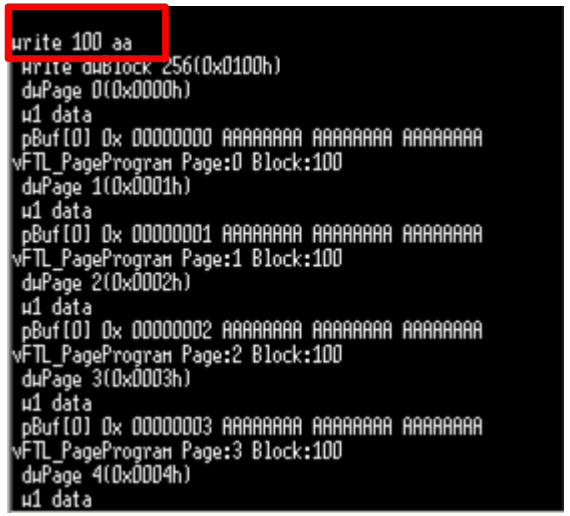
程式執行

- 可以關閉MPtool並將硬體重新插拔再試一次。
  - ✓ 等到UART跳完，按enter
  - ✓ 出現command..即為開卡成功



- 下command 做測試
  - ✓ [erase][blockid]
  - ✓ [write][blockid][pattern]
  - ✓ [read][blockid][pagenum]
  - ✓ 讀出來的資料會是寫入的資料





Flash Translation Layer

- Recieve command from host
- Perform LBA to PBA mapping
- Allocate data from buffer into flash blocks

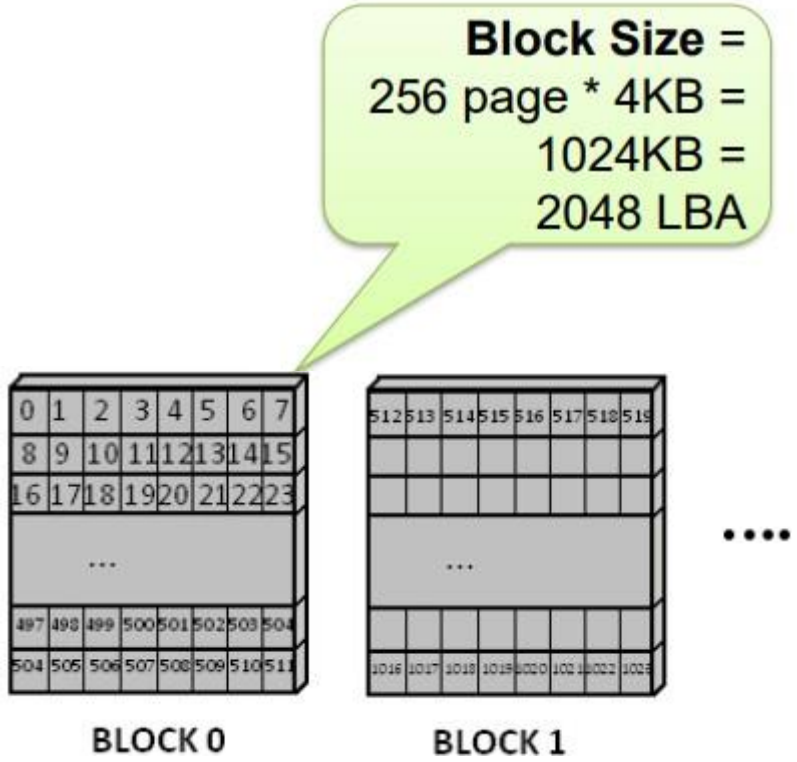
Command

- UART
  - Mainly for the development stage debugging

- SATA & SCSI
  - Host read/write
  - To be continued...

Flash LBA Address

- Block → 256 page
- Page → 4K Byte → 8 LBA
- LBA = 512 Byte



Requirement 1

- Follow the setup guide
- implement “info” command to print your information through UART
- After typing "info", SSD should print:
  - “HELLO LAB 1”
  - your name in english
  - your student ID
  - finish date

Requirement 2

- implement the "read“ & “write” command
- There are only a fraction of pages been printed through UART, please output the whole pages in one block by a single command
- Hint: There are 512 Byte in one LBA sector, and 256 pages in one block

Hint

- 為達成 Lab1 的目標，請修改/bin/sys/uart_command_channel.c 之實作。
  原先的 read command 只會印出兩行，共32 byte 的 data ，請修改至能夠印出整個page ，
  即 “read a0 1” 會印出 block a0 中 page 1 共 8 LBA = 4KB 之所有資料 。
  原先的write command 也是只會修改部分byte的data，請修改至能夠寫入整個 page 。

- Ps: 每次改完code 需要重新編譯的話，都需要重新複製到MPTool，再次下載韌體至SSD中。

Grading

- Please come to CSSLAB(CSIE 65603) for the offline demo:
  - Download the firmware _____ 30%
  - Requirement 1. (briefly explain your implementation)_____ 30%
  - Requirement 2. (briefly explain your implementation)_____ 30%
  - Complete all steps before the due time_____ 10%

- Please mail me days before the appointment
  - I'll mail you back for the exact time
  - Available time : Friday 16:00 ~ 18:00
  - due time : 10/25(五)
  - You can join the discord group for discussion: https://discord.gg/GxNUyzxkZr