

數位系統導論實驗 實驗4

ZedBoard

負責助教：

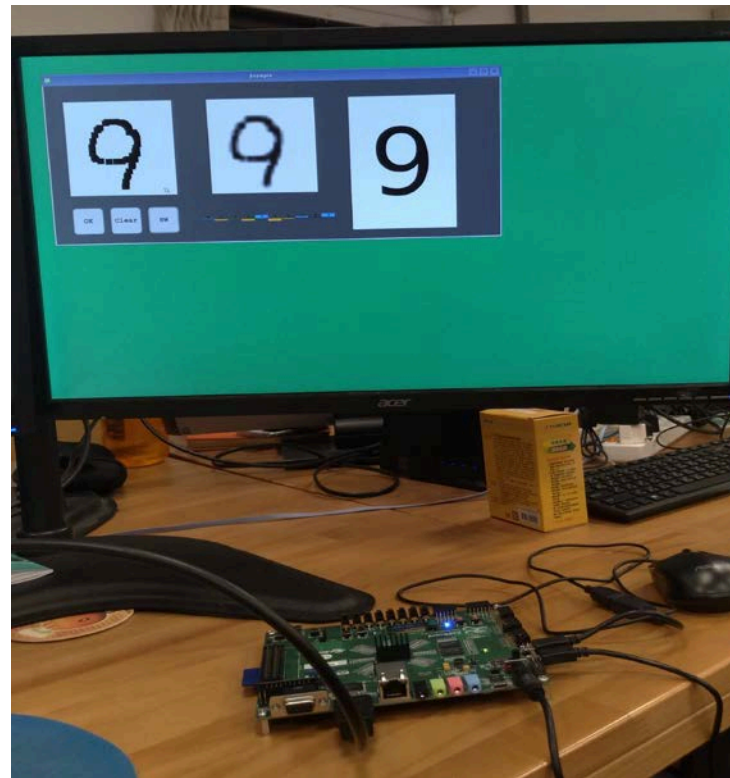
王璽喆 aaa0419347@gmail.com

Outline

- ▶ 課程目標
- ▶ Zedboard介紹、檢查與開機
- ▶ 移植手寫辨識程式至Zedboard
 - ▶ 安裝交叉編譯器
 - ▶ 編譯手寫辨識程式
- ▶ 在Zedboard上執行手寫辨識程式
- ▶ 課程評分

課程目標

- ▶ 同學們在Lab3已經運用Qt完成軟體版手寫辨識。而本次實驗會帶大家初步認識Zedboard開發板，與將手寫辨識軟體移植至Zedboard開發板上



Zedboard介紹、檢查與使用

Zedboard介紹 (1/3)

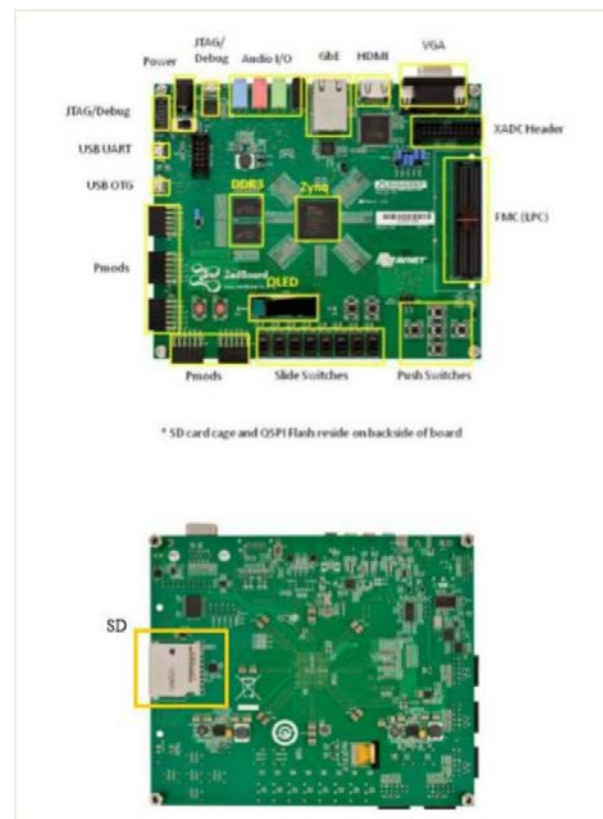
- ▶ ZedBoard是基於Xilinx Zynq-7000擴展式處理平臺(EPP)的嵌入式系統開發板



Zedboard介紹、檢查與使用

Zedboard介紹 (2/3)

- ▶ Zedboard上面有
 - ▶ 雙核心ARM Cortex-A9 667MHz
 - ▶ 512MB DDR SDRAM
 - ▶ 256MB Flash與SD卡插槽
 - ▶ D-Sub與HDMI輸出
 - ▶ USB On-The-Go與其他輸入輸出介面
- ▶ 可以執行Linux、Android、或其他OS/RTOS
- ▶ 就是一台小小電腦
只是CPU跟一般的電腦不一樣
- ▶ 詳細的介紹如[連結](#) (英文)



Zedboard介紹、檢查與使用

Zedboard介紹 (3/3)

- ▶ 貴！所以請小心使用
- ▶ 當然不會要每個人買
所以後面會說明借用流程
- ▶ 歸還時會檢查板子運作狀況，若有故障需要補一張全新品
或者實驗課那組整組學期成績0分

Home / By Product Status / Action / ZedBoard Zynq-7000 ARM/FPGA SoC Development Board



Digilent
ZedBoard Zynq-7000 ARM/FPGA SoC Development Board

\$449.00

SKU:
410-248

Add-ons:
☒ None

☐ Add \$050C Voucher

Quantity:

1

ADD TO CART

WISHLIST

Share

★★★★ 1 product review

Zedboard介紹、檢查與使用

Zedboard檢查

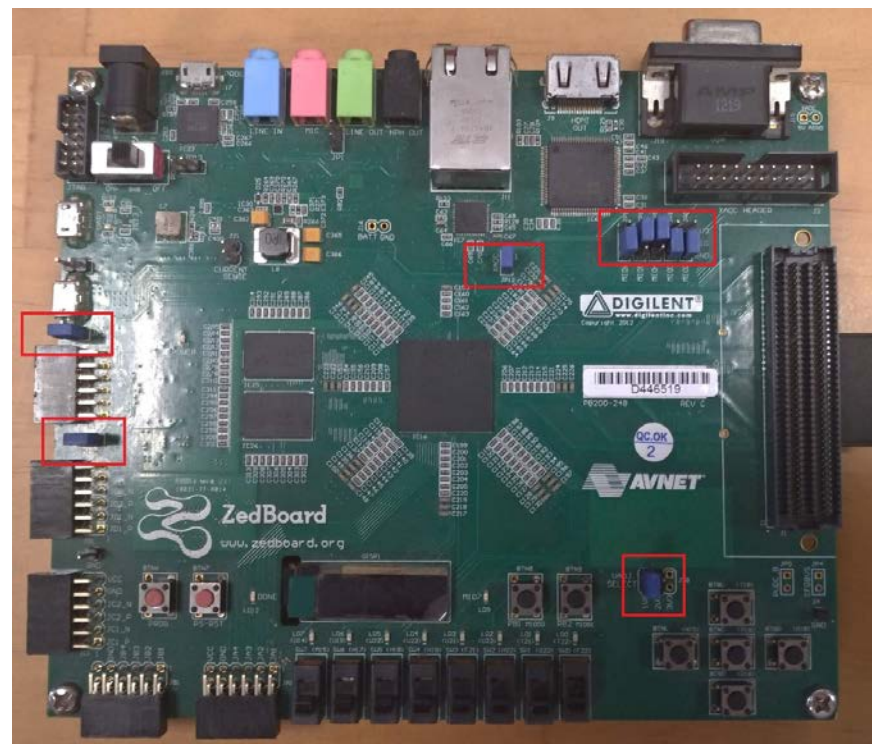
- 每盒Zedboard內應該都會有
 - Zedboard開發板
 - 12V變壓器
 - 4GB SD卡
 - Micro-USB延長線
 - USB Micro-B母口轉USB A型公口轉接線



Zedboard介紹、檢查與使用

Zedboard使用 (1/4)

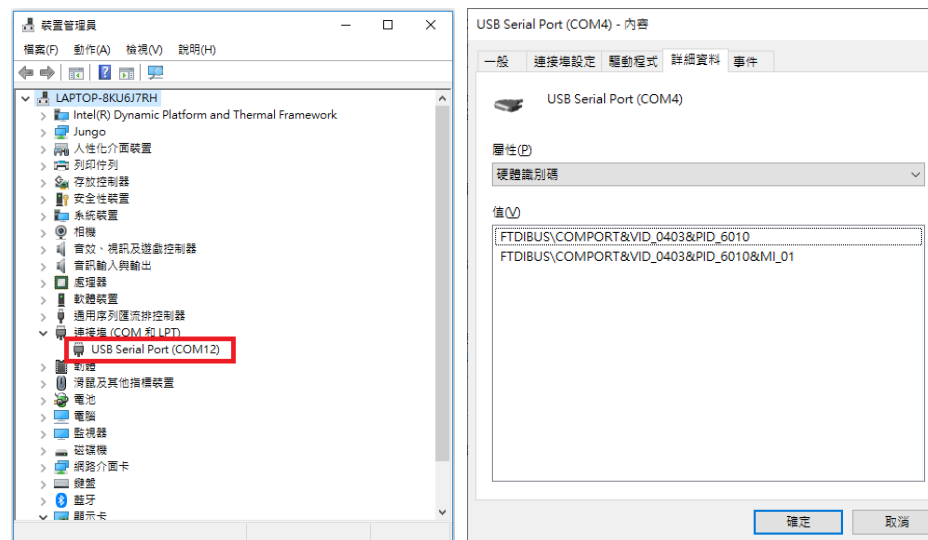
- ▶ 確認手上開發板的Jumper有沒有接對
 - ▶ 這些Jumper會影響到板子的行為，設定錯誤可能會導致開發板開不了機甚至燒毀



Zedboard介紹、檢查與使用

Zedboard使用 (2/4)

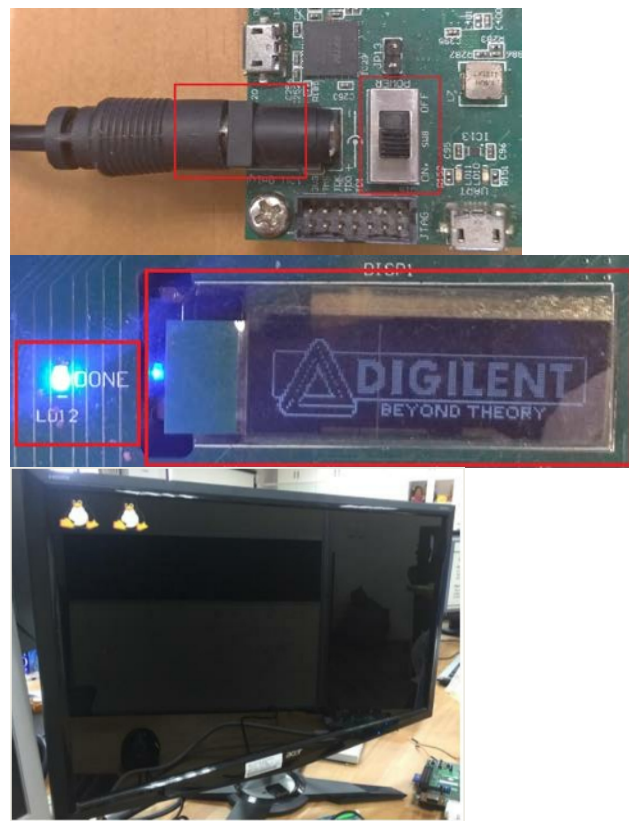
- ▶ 將ZedBoard接上電源、USB-UART連接至電腦、HDMI接上螢幕、SD卡插上，接著開啟ZedBoard電源
- ▶ 在裝置管理員內上會多出一個USB Serial Port，這即是連接至ZedBoard的Port，如範例為COM12
 - ▶ 若沒有出現USB Serial Port但是有出現未知的裝置再請同學利用裝置內容詳細資料內之硬體識別碼去尋找驅動程式並安裝



Zedboard介紹、檢查與使用

Zedboard使用 (3/4)

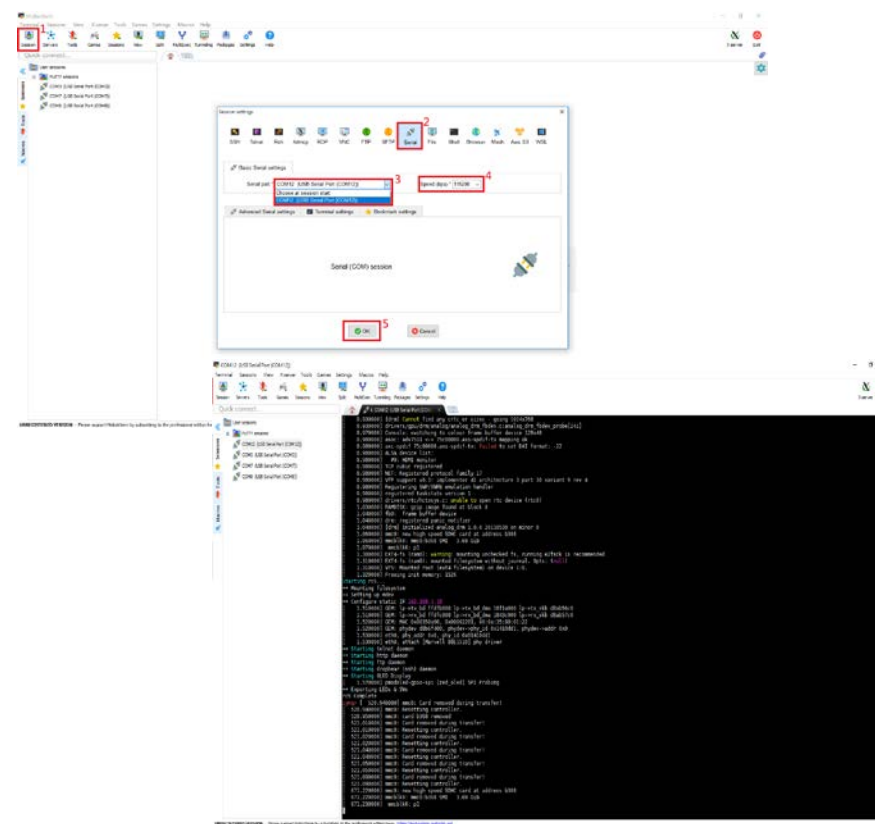
- ▶ 開關開啟並稍待片刻，若藍色LED變亮說明Zynq晶片已設定完成
- ▶ 啟動完成後
 - ▶ 板上OLED小顯示器會顯示一個DIGILENT的Demo圖像
 - ▶ HDMI螢幕會輸出Linux的企鵝圖案



Zedboard介紹、檢查與使用

Zedboard使用 (4/4)

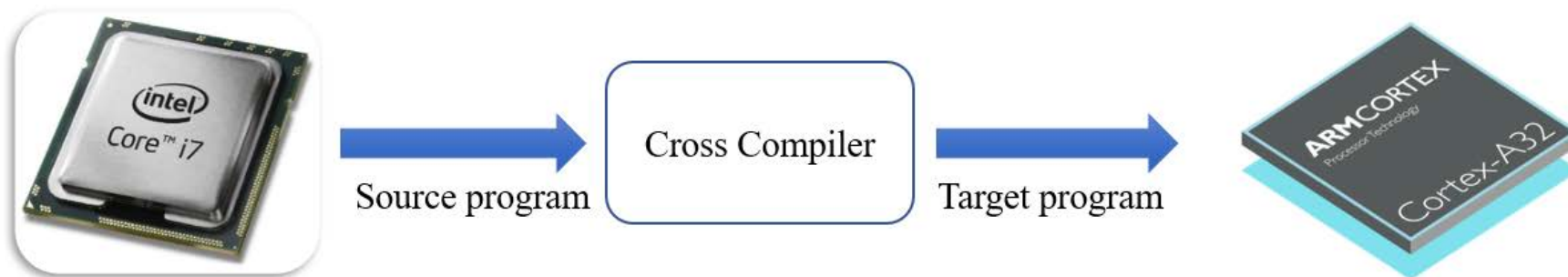
- ▶ 開啟MobaXterm選擇左上Session > 跳出視窗選擇Serial > Serial port選擇ZedBoard那個Port >Speed選擇115200 > 按 OK
- ▶ 若有連線成功則會出現Zedboard的終端機畫面
 - ▶ 若什麼都沒看到但又沒有錯誤訊息的話可以按一下Enter
 - ▶ 如果有看到**zynq>**就沒問題了



移植手寫辨識程式至Zedboard

安裝交叉編譯器 (1/2)

- ▶ 因為開發板用的處理器跟一般的電腦不一樣
(一般的電腦是x86 CPU，而開發板用的是手機比較常用的ARM CPU)
- ▶ 所以要在一般的電腦上安裝交叉編譯器，讓一般的電腦有辦法編譯出給ARM CPU用的程式



移植手寫辨識程式至Zedboard

安裝交叉編譯器 (2/2)

- ▶ 在PC上開啟Linux作業系統
- ▶ 從教學平台下載cross_compiler.tar.gz及Makefile.zip到Downloads資料夾
- ▶ 開啟終端機進行以下步驟與指令: (綠字請替換成同學自己的路徑)
 - ▶ `sudo -i` 取得最高權限
 - ▶ `apt-get install lsb-core` 安裝 lsb-core
 - ▶ `mv 壓縮檔存放路徑 /` 將壓縮檔移至系統根目錄
 - ▶ `tar zxvf /cross_compiler.tar.gz` 在系統根目錄下解壓縮資料夾

移植手寫辨識程式至Zedboard

編譯手寫辨識程式

- ▶ 在Downloads資料夾新增檔名為ex資料夾，並放入Qt的C code及存放weight與bias之文件檔
- ▶ 將Makefile.zip解壓縮到 Downloads資料夾
- ▶ 將終端機目前所在路徑移動到ex資料夾
 - ▶ `export PATH=/opt/cross_compiler/arm_linux_eabi/bin:$PATH` 設定環境變數
 - ▶ `/opt/cross_compiler/pandaboard/qt-pandaboard-lib/bin/qmake -project` 建立 Qt 專案
 - ▶ `/opt/cross_compiler/pandaboard/qt-pandaboard-lib/bin/qmake -spec /opt/cross_compiler/pandaboard/qt-pandaboard-lib/mkspecs/qws/linux-omap4430-g++`
生成 Makefile 檔
 - ▶ 將終端機所在路徑移至Downloads
 - ▶ `cp Makefile資料路徑 ex資料夾路徑` 將Makefile.zip解壓縮出的Makefile移至ex資料夾覆蓋
 - ▶ 將終端機路徑從Downloads移至ex資料夾
 - ▶ `make` 在ex資料夾下輸入產生ex執行檔
- ▶ 此執行檔即可在Zedboard上使用
- ▶ 詳細安裝過程可參考連結影片

在Zedboard上執行手寫辨識程式 (1/2)

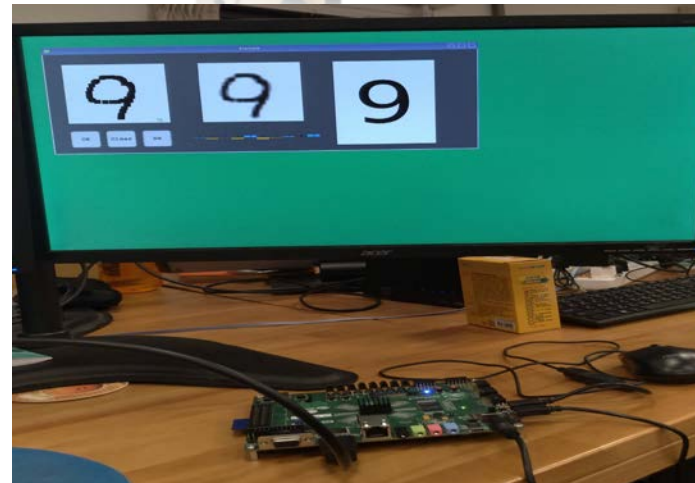
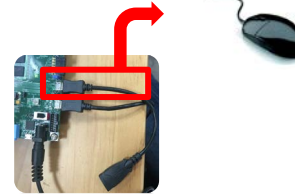
- ▶ 請同學將編譯好的執行檔、參數檔 及 SDCard壓縮檔內容放入SD卡
- ▶ 在Zedboard 上USB OTG口接上公對母轉接頭並連上滑鼠
- ▶ 首先需確認 Qt 函式庫是否安裝正確，在Zedboard的終端機中輸入：
 - ▶ `mount /dev/mmcblk0p1 /mnt` 掛載filesystem
 - ▶ `cd /mnt` 跳至mnt資料夾，即SD卡資料位置
 - ▶ `source init` 設定環境變數及掛載Qt_lib.img 映像檔
 - ▶ `cd Qtdemo` 切換終端機環境至demo資料夾
 - ▶ `./demo -qws` 執行Qt測試檔案
- ▶ 最後螢幕會呈現示範程式(如圖)
- ▶ 若呈現如圖畫面，即可先將此demo關閉



在Zedboard上執行手寫辨識程式 (2/2)

- ▶ 回到MobaXterm
- ▶ 使用cd指令跳至mnt資料夾底下 (cd mnt)，輸入ls即會看到SD卡內的檔案
- ▶ 接著輸入: (綠字替換成自己的檔名)
 - ▶ source init
 - ▶ ./執行檔檔名 -qws
- ▶ 螢幕會呈現如圖畫面，並且可以用滑鼠寫上數字

```
zynq> cd /mnt
zynq> ls
01.jpg      goodd150
02.jpg      grayImg.ppm
03.jpg      init
04.jpg      lab5_sw
05.jpg      lib
BOOT.BIN   main.c
```



課程評分

- ▶ 評分方式：
 - ▶ 每位同學都要有自己的作品，Demo時執行檔請命名為自己的學號
 - ▶ 當天Demo時請同學攜帶lab3程式，現場編譯執行檔。並自行把執行檔移至Zedboard。以Zedboard呈現自己的手寫辨識軟體，完成得100分
- ▶ Demo時間：
 - ▶ 2019/4/08或2019/4/10
 - ▶ Demo時間共分四梯次，分別為19:30、19:50、20:10與20:30
- ▶ Demo 地點：資工館206教室
- ▶ 注意事項：請各組的第一位同學將Zedboard帶至Demo場地，由該組最後一名成員收回保管

END

一些已知可能會遇到的問題與解決辦法

- ▶ Q:
我的電腦跟我說arm-none-linux-gnueabi-g++: Command not found 耶
- ▶ A:
你忘了裝lsb-core
執行[sudo apt-get install lsb-core](#)補裝

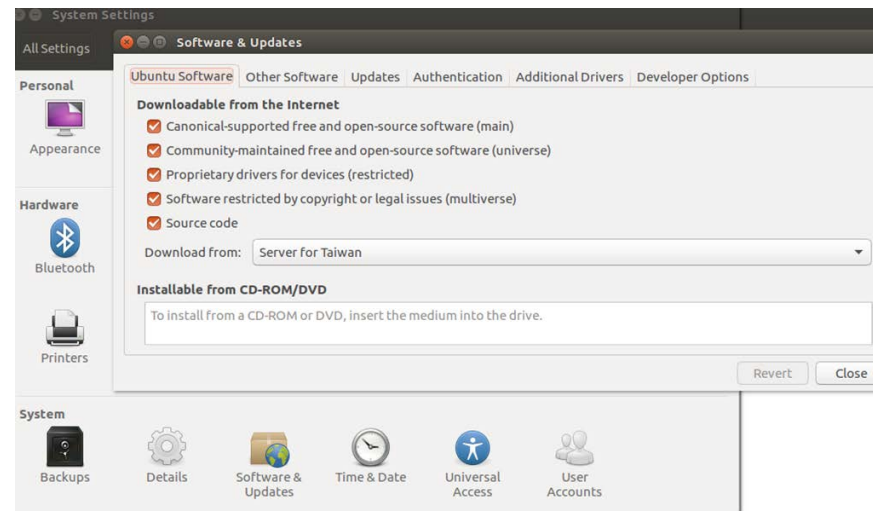
```
root@dante-VirtualBox:/home/dante/Downloads/ex# make
/opt/cross_compiler/arm_linux_eabi/bin/arm-none-linux-gnueabi-g++ -c -pipe -O3 -march=armv7-a -mtune=cortex-a9 -mfloat-abi=softfp -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_NETWORK_LIB -DQT_CORE_LIB -I/opt/cross_compiler/pandaboard/qt-pandaboard-lib/mkspecs/qws/linux-omap4430-g++ -I. -I/opt/qt-pandaboard-lib//include/QtCore -I/opt/qt-pandaboard-lib//include/QtNetwork -I/opt/qt-pandaboard-lib//include/QtGui -I/opt/qt-pandaboard-lib//include -I. -I. -o main.o main.cpp
make: /opt/cross_compiler/arm_linux_eabi/bin/arm-none-linux-gnueabi-g++: Command not found
Makefile:197: recipe for target 'main.o' failed
make: *** [main.o] Error 127
```

一些已知可能會遇到的問題與解決辦法

- ▶ Q:
我的電腦又跟我說lsb-core沒有安裝
需要的證書耶
- ▶ A:
去Ubuntu的控制台裡面的軟體
把來源那邊全開 如圖

```
ubuntu@ubuntu: ~/Downloads/example
ubuntu@ubuntu:~/Downloads/example$ sudo apt-get install lsb-core
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package lsb-core is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'lsb-core' has no installation candidate
ubuntu@ubuntu:~/Downloads/example$ ^C
ubuntu@ubuntu:~/Downloads/example$
```



一些已知可能會遇到的問題與解決辦法

- ▶ Q:
我的電腦跟我說他找不到
QtGui/Qapplication之類的東西
- ▶ A:
檔案位置可能亂掉了
檢查需要的檔案的位置或是去修改
Makefile

或參閱投影片第14頁步驟將課程提供的 Makefile 覆蓋上去

```
main.cpp:1:30: fatal error: QtGui/QApplication: No such file or directory
compilation terminated.
Makefile:197: recipe for target 'main.o' failed
make: *** [main.o] Error 1
root@dante-VirtualBox: /home/dante/Downloads/ex#
```

一些已知可能會遇到的問題與解決辦法

- ▶ Q:
我的板子不鳥我！
- ▶ A:
~~好喔 你們死定了~~
先檢查電源、板子上面Jumper的狀態與記憶卡有沒有問題
還是有問題就趕快跟助教們說

如果是玩到一半沒反應了
那就先把電關掉重開機看看吧，畢竟電腦都會當機嘛
然後電直接關掉就好不用像電腦要打關機指令什麼的