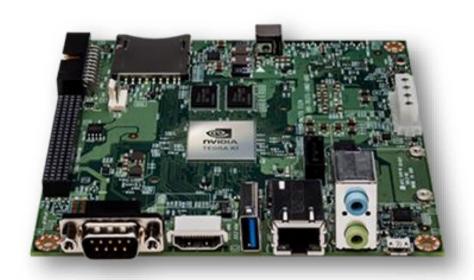
微算機系統

實驗二



組別:第九組

組員名稱:

資工二 106590034 吳陽生

資工二 106590036 梁博瑜

日期: 2019-03-17

實驗過程

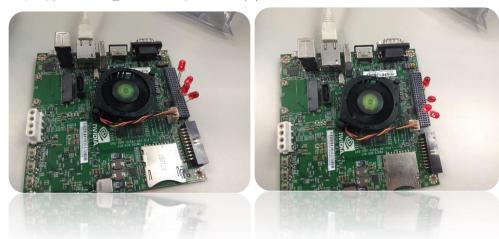
撰寫程式碼: 將報告上的程式碼輸入編輯中並修改 main 以完成實驗要求

```
ioExport(unsigned int gpio)
                                                    nt main(int argc, char* argv[])
int fd, len;
                                                       int num = 0, gpios[] = {160, 161, 162, 163};
char buf[64];
                                                       string arg1, arg2;
fd = open("/sys/class/gpio/export", O_WRONLY);
                                                       cout << "初始化中..." << endl;
                                                       for (int i = 0; i < 4; i++) {
   perror("gpio/export");
                                                           ioExport(gpios[i]);
                                                           ioSetDir(gpios[i], "out");
ioSetValue(gpios[i], 1);
len = snprintf(buf, sizeof(buf), "%d", gpio);
                                                       sleep(1);
                                                       for (int i = 0; i < 4; i++) {
close(fd);
return 0;
                                                           ioSetValue(gpios[i], 0);
```

Makefile: 編寫 makefile, 輸出本機執行檔及跨平台執行檔

```
編譯器
                                           all: $(TARGET1) $(TARGET2)
CC1 = g++
                                            $(TARGET1): $(DEPENS)
CC2 = arm-linux-gnueabihf-g++ # 跨平台編譯
                                               $(CC1) $(STANDARD) -o $@ $^
STANDARD = -std=c++11 #使用標準
                                            $(TARGET2): $(DEPENS)
TARGET1 = main
                                               $(CC2) $(STANDARD) -o $@ $^
TARGET2 = main_tk1
                                           move:
DEPENS = main.cpp sort.cpp
                                               scp $(USER)@$(IP):DIR
USER = ubuntu # 使用者名稱
                                               ./$(TARGET)
ID = 192.168.0.1 # tk1對應ida
DIR = /home/ubunto/... # 本機位置
                                            clean:
TK1_DIR = /home/ubunto/ # 目標位置
                                             rm $(TARGET)
```

結果: 輸入"Modo_Shine 5" 使 LED 燈閃爍五次



組員貢獻比例

心得

吳陽生

本次實驗其實不難, 照著講義上給的程式碼就可以完成基本題了, 而進階題需要使用到函數 sleep, 加入命令列參數, 其中 sleep 在 windows 系統下需引入標頭檔<windows.h>, 而 linux 則需引入<unistd.h>; 命令列參數則要在 main 的參數列輸入"int argc, char* argv[]", 變數 argc 存放輸入參數個數, argv 則存放參數的字元陣列. 我後來又去查了一下資料, 實際上, 使程式停下來, 在 c++11 中還可以使

用 "std::this_thread::sleep_for(std::chrono::milliseconds(500));" 需要引入的標頭檔則改為 <thread>

這次實驗的程式部分我們並沒有花太多時間, 比較氣人的是學校的電腦實在太爛, 虛擬機每次開都要等很久, 寫到一半電腦還直接死當, 只能打掉重練, 重新開始. 最後好不容易寫好的時候, 卻在傳送執行檔過程中, 板子一直無法存取, 到助教來檢查的時候, 才知道板子壞掉了...... 總之, 禮拜一的實驗時間都浪費在這些鳥事中, 直到禮拜二來學校的時候才用新板子完成這次實驗.

值得一提的是,本來上禮拜打算用 makefile 使傳送執行檔到板子中的過程自動化,當然是失敗了,目前遇到的問題是,makefile 中變數在帶入的時候都會在前後加入空白,使得我的指令格式跑掉,造成命令列無法判讀,當時因為時間緊張就沒有繼續試下去,下禮拜實驗如果有機會可以繼續研究下去.

梁博瑜

這一次的實驗中途一職遇到一些奇怪的問題,一開始是電腦直接當機,重開之後換虛擬機裡面ubuntu打不開,再重開又當機,光是這樣開開關關大概就花了我們半個小時,好不容易搞定之後,換成沒有辦法把傳到板子上的檔案砍掉,只好把板子重開,結果就完全連不上板子了,時間也差不多來到要下課的時候了,最後助教直接換一塊板子給我們,做了一下簡單的測試,能夠正常上傳、刪除,看來原本的板子是報銷了,我們的實驗進度也跟著板子一起報銷了,只能夠在星期二的三四節完成實驗,中途還是有遇到一些小bug而且莫名其妙的就好了,完全搞不懂到底是發生了什麼事,像是檔案莫名的傳不到板子上面去,一直找不到指令哪裡打錯,然後再打一次就過了,其實真的在弄實驗的時間並不多倒是在奇怪的地方花了一堆時間,希望下次不再遇到這些問題了。