



『嵌入式系統』—課程作業(專題實作)

File Name: E2_160-H-AY109B-Project.doc

一、注意事項：

- (1) 本作業主題為期末專題製作說明文件。繳交方式：以上傳電子檔方式繳交，並應配合下列之說明。
 - (A) 請使用本作業電子檔之作答頁依說明完成本作業，於首頁填入個人識別資料後，以[學號]-project.doc之格式命名存檔，完成作業後繳交。
 - (B) 將完成本作業相關檔案(含本作業檔案)收集於一資料夾內(資料夾之命名方式：[學號]-**Project**)，並附加資料夾內容說明檔；將該資料夾壓縮後，上傳至成功大數位學習平台(Moodle)之指定位置。
 - (C) 壓縮檔名之格式為：[學號]-Project.(rar|zip)，壓縮格式可為 rar 或 zip。
- (2) 切勿更改本作業原始版面設定，並依規定保存所有相關資料與程式等檔案。
- (3) 獨立完成作業，如有抄襲或剽竊情事，除酌情議處並扣減學期成績。
- (4) 評分基準：各項配分×80%，優增劣減。請勿遲交，以免影響學期成績。

二、基本資料：

繳交期限：**2021/07/08**

繳交日期：**2021/ 06 / 30**

班級：航太碩一

學號：P46091204

姓名：蔡承穎

三、作業內容：

This homework is intended to be the presentation of the documentation for the course mini-project. Please complete based on your design and implementation.

[Note]: Please complete the table in next page before sending in your homework.

- (1) Abstract – Please briefly describe your project including, but not limited to, the functions, the main algorithm, and the results (testing ok?). (10 points)
- (2) Please describe the architecture of your design using block diagram. (20 points)
- (3) Please describe your design using state diagram. (20 points)
- (4) Describe, using flow chart or similar presentation, the operation flow of the main program in your implementation. (20 points)
- (5) Please describe the main and important data structures and present the source code in your implementation. (10 points)
- (6) Please describe the challenging and/or difficult parts of your project and how you solve them. (20 points)





專題名稱		P01 - 摩斯電碼顯示器		
開始日期		2021/ 06 / 23	完成日期	2021/ 06 / 30
時間統計	總時數	約 4 小時 0 分鐘	百分比(%)	備 註
	設 計	約 1 小時 0 分鐘	25.00 %	
	實 作	約 0 小時 50 分鐘	20.83 %	
	測 試	約 0 小時 30 分鐘	12.50 %	
	除 錯	約 1 小時 40 分鐘	41.67 %	

【注意】

1. 上表已格式化，填寫時請使用拉下式選單選擇，或將游標移至 “0” 之位置後輸入。
2. 填寫上表各欄位時，請勿使用或按 [↵ Enter] 鍵。可使用 F1 按鍵顯示說明。
3. 欄位填寫後，可使用 [Tab] 按鍵離開或移至下一欄位。

請使用本頁以下起空間完成本作業。(注意事項：請標明題號並依題序完成作業。)

(1) Abstract:

(請填寫於框格內)

(1)		
Project : Morse Code 顯示器 簡介		
主要功能	由 em_up 端顯示可以輸入欲轉換成摩斯密碼的文字的提示，接受使用者所輸入的文字資料，當使用者輸入 Enter，則將文字資料傳送至 em_host 端，em_host 加密其文字資料成摩斯密碼，並以 1 與 0 方式顯示於 em_host 端。其中 1 表示 LED1 顯示一個點樣式持續 1 秒，111 表示 LED1 顯示一條線樣式持續 3 秒，0 則是沒有顯示，1 個 0 代表持續 1 秒。	
主要函式	void toMorseCode(char* line, int length)	
	輸入	line：由 getline(&line, &(size_t){0}, file)得到的使用者文字內容
		length：由 getline(&line, &(size_t){0}, file)得到的使用者文字內容長度
	內容	透過使用者輸入的文字內容，經由摩斯密碼轉換，得到對應的 dot 與 dash pattern，並顯示在 upBoard 上的 LED1
演	回傳	無回傳





算法

```
for (int i = 0; i < length-1; i++){
    c = line[i];

    if (c >= 32 && c <= 122){ // Step1. 判斷 c 是否能轉成摩斯密碼
        if (c==33) { temp = "-.-.--"; }
        ...
        if (temp == NULL) break;

        if (c != 32){ // Step2-1. 如果 c 不是空格，那我們就輸出對應的圖案
            for (size_t i = 0; i < strlen(temp); i++){
                if (temp[i] == '.') { showDot();}
                if (temp[i] == '-') { showBar();}
                if (i != strlen(temp)-1){
                    // Step2-1-1. dot 與 dash 之間需停頓 1 秒
                    printf("0");
                    delay_ms(1000);
                }
            }
            temp = NULL;
        } else if (c == 32) { // Step2-2. 如果 c 是空格，就停頓 7 秒
            delay_ms(7000);
            continue;
        }
        // Step3. 如果下一個字元不是空格，代表屬於字元間隔，停頓 3 秒
        if (!(line[i+1] == 32 || i+1 == length-1)){
            delay_ms(3000);
        }
    } else {
        break;
    }
}
```





程式結果

1. 可以轉換成摩斯密碼的字元

```
Server will show what you input with corresponding Morse Codes. Let's test !
Enter quit to quit

Input> a b

Bye !

embedded@upboard:~$ sudo ./P01
[sudo] password for embedded:
101110000000111010101
Client says: a b (length=3)
```

由上圖，我們的測試資料為 a[空格]b，並可以在 em_host 端接受到回傳的摩斯密碼。

2. 不可轉換成摩斯密碼的字元

```
Server will show what you input with corresponding Morse Codes. Let's test !
Enter quit to quit

Input> \

Bye !

embedded@upboard:~$ sudo ./P01
Something wrong..., there are wrong characters which can't be converted to Morse Codes.

Client says: \ (length=1)
```

由上圖，我們的測試資料為 \，並可以在 em_host 端接受不能轉換成摩斯密碼的提示。

3. 中斷程式執行

```
Server will show what you input with corresponding Morse Codes. Let's test !
Enter quit to quit

Input> a b

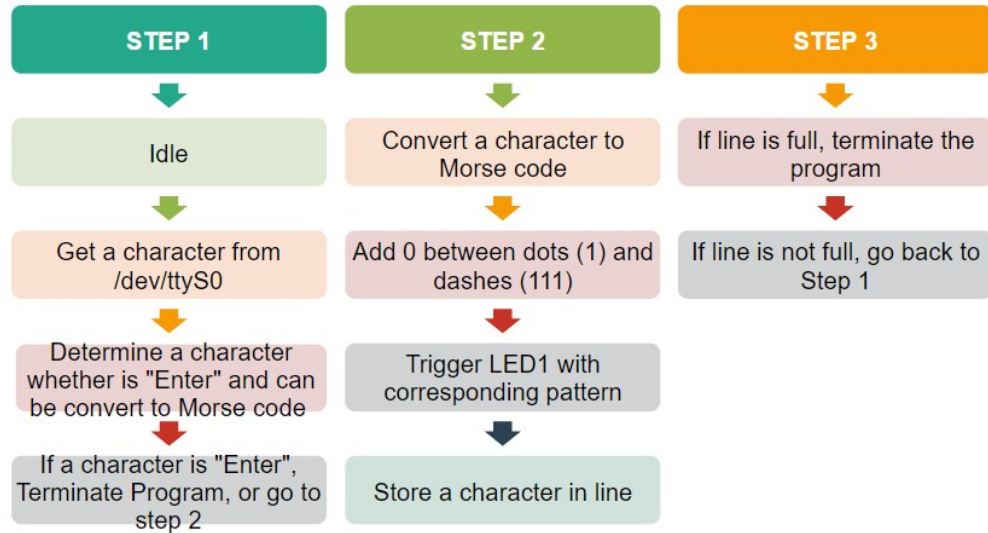
embedded@upboard:~$ sudo ./P01
^C10Received signal 2
```

由上圖，當我們在程式執行時，輸入 ctrl+c 可以中斷程式

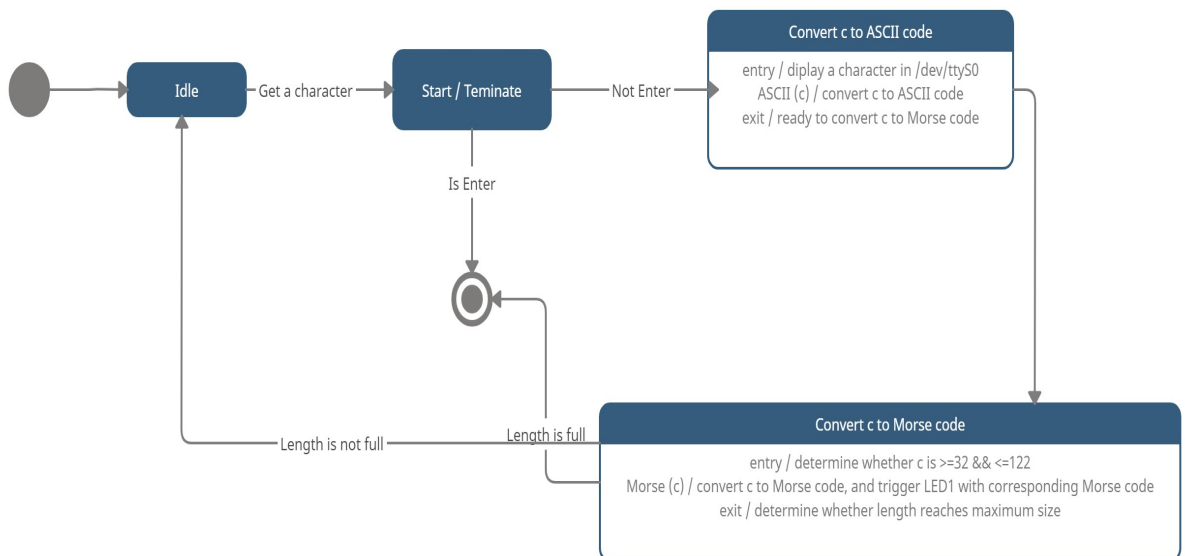




(2)

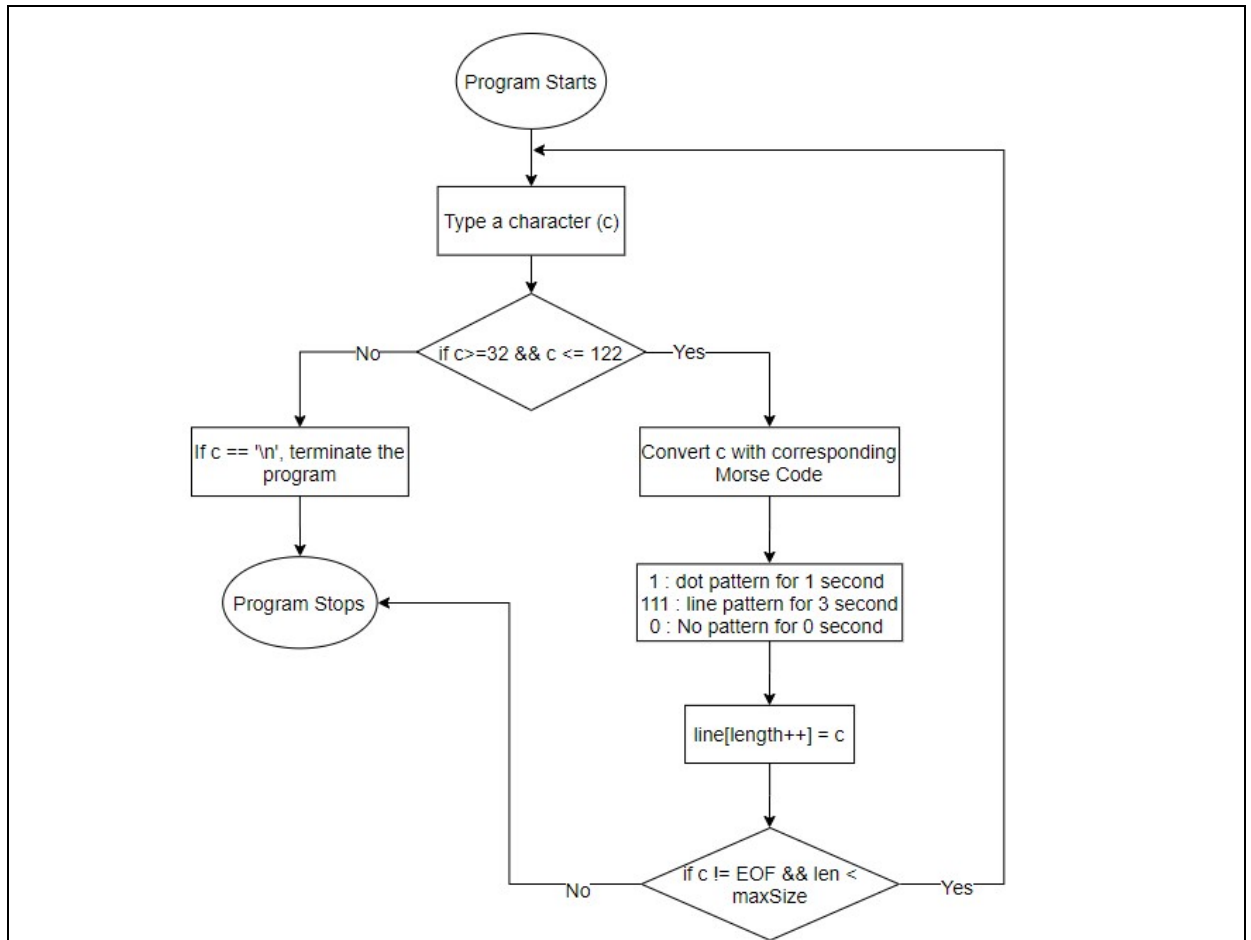


(3)



(4)





(5)

在 void setup_terminal(int fd, struct termios *old_term) 中	
當使用者輸入 Enter 時才回傳資料，需要將 ICANON 打開	<code>new_term.c_lflag = ECHO ECHOE ICANON ICRNL;</code>
在 main() 函式中	
將 /dev/ttyS0 以檔案指針的方式打開，未來就能讀取使用者輸入的資料	<code>FILE *file = fdopen(fd, "r+");</code>
/用來將 ToMorse 回傳的長度與文字資料做存取	<code>int length; char* line;</code>
在 void toMorseCode(char* line, int length) 函式中	
用來從 /dev/ttyS0 得到使用者輸入的文字資料	<code>char c;</code>
用來將輸入文字資料轉變成對應的摩斯密碼	<code>char *temp = NULL;</code>
在 void showDot() 函式中	





在 8x8 的 LED1 矩陣的第四列與第五列顯示出一個小方塊圖示當作摩斯密碼的 dot

```
const uint8_t dot_pattern = 0x18;  
write_reg(4, dot_pattern);  
write_reg(5, dot_pattern);
```

在 void showBar() 函式中

在 8x8 的 LED1 矩陣的第四列與第五列顯示出兩條線合併成的粗線當作摩斯密碼的 dash

```
const uint8_t bar_pattern = 0xFF;  
write_reg(4, bar_pattern);  
write_reg(5, bar_pattern);
```

(6)

1. 挑戰：

在這個 project 為了解決每個字母對應的摩斯密碼，就手打了其對應的 1 與 0，dot 與 dash，列出了一堆的 If，同時給清楚知道如何判斷是字母間隔或是單字間隔的顯示情況。

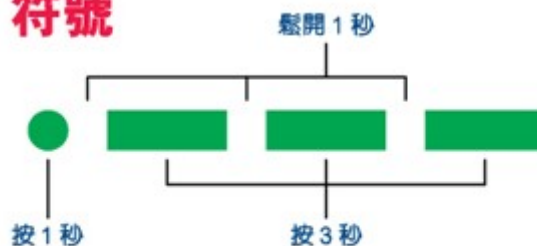
2. 困難：

a. 根據摩斯密碼的規範（下圖所示），我們 dot 與 dash 之間需要補 0 之外，字母之間也需要補 000，表示停頓三秒再顯示下個字母的摩斯密碼，然而在單字之間需要補 0000000 表示停頓 7 秒。

英文字母

J

符號



因此我在這裡的解決方案是，預先判斷下一個字是空格還是字元，若是字元則輸出 000，若是空格就輸出 0000000，避免輸出 10 個 0 的錯誤。

b. debug 時，沒有發現輸出 ctrl+c 無法中斷程式，故將 void int_handler(int sig) 裡多加了 exit(sig)而可以中斷程式。





c. 在 ASCII code 裡，32 到 122 之間還是有些字是沒有對應的摩斯密碼，需要再多寫個判斷式，判斷是否無法轉成摩斯密碼，並 print 以下這句話，Something wrong..., there are wrong characters which can't be converted to Morse Codes. 提示使用者發生了些錯誤。

