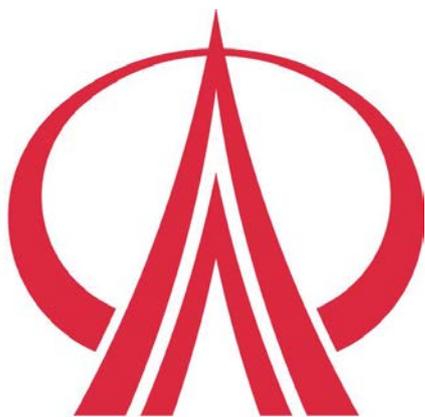


修平科技大學 電機工程系

DEPARTMENT OF ELECTRICAL ENGINEERING
HSIUPING UNIVERSITY OF SCIENCE AND TECHNOLOGY

實務專題報告書

微電腦控制風扇



指導老師：呂奇明

專題製作學生：四季電四乙 紀筌淵 BD99058

四技電四乙 張曜任 BD99100

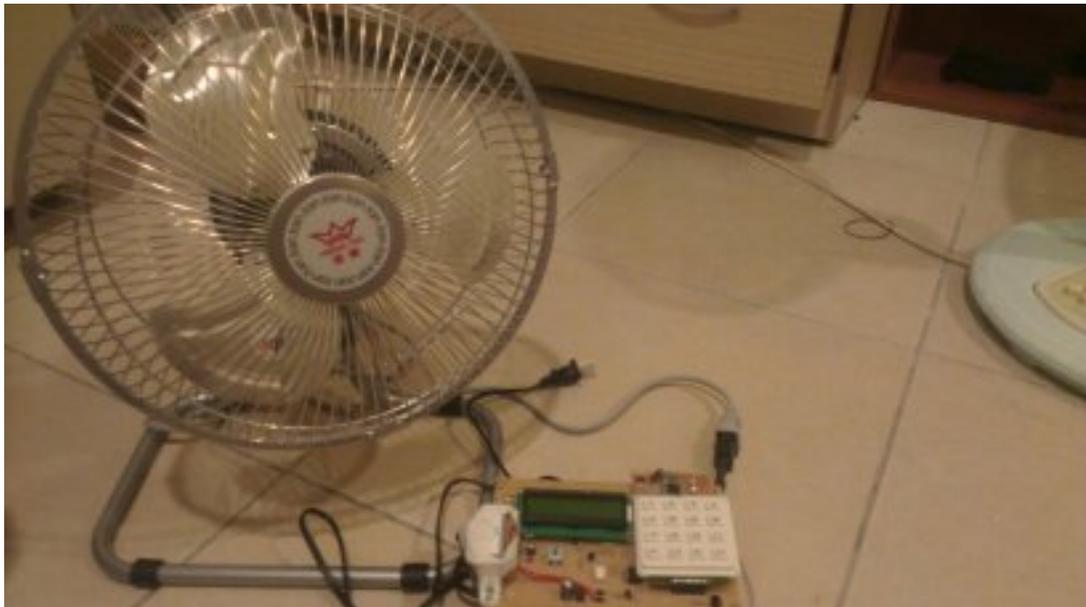
中華民國 一百零二年 十二月三十日

目錄

摘要.....	2
第一章導論.....	3
■ 1-1 序論	3
■ 1-2 制作動機	3
■ 1-3 系統架構圖	4
第二章專題主要元件介紹.....	5
■ 2-1 單晶片 8051.....	5
■ 2-2 類比/數位轉換.....	8
■ 2-3 熱敏電阻.....	9
■ 2-4 掃描鍵盤.....	9
■ 2-5 紅外線.....	10
■ 2-6 Eclipse 開發軟體介紹.....	11
第三章軟硬體實作.....	12
■ 3-1 程式流程圖.....	12
■ 3-2 硬體電路.....	13
■ 3-3 測試狀況.....	14
第四章結論.....	15
參考文獻.....	16

摘要

本專題只要使用單晶片 8051 與 Android 開發平台使用 JAVA 撰寫應用程式，透過藍芽使智慧型手機與電路板上的 Arduino 做連結，可對風扇進行開關作用。



第一章導論

1-1 序論：

市上有各種功能的風扇，但我目前還沒看過使用遠端來控制風扇的功能出現。

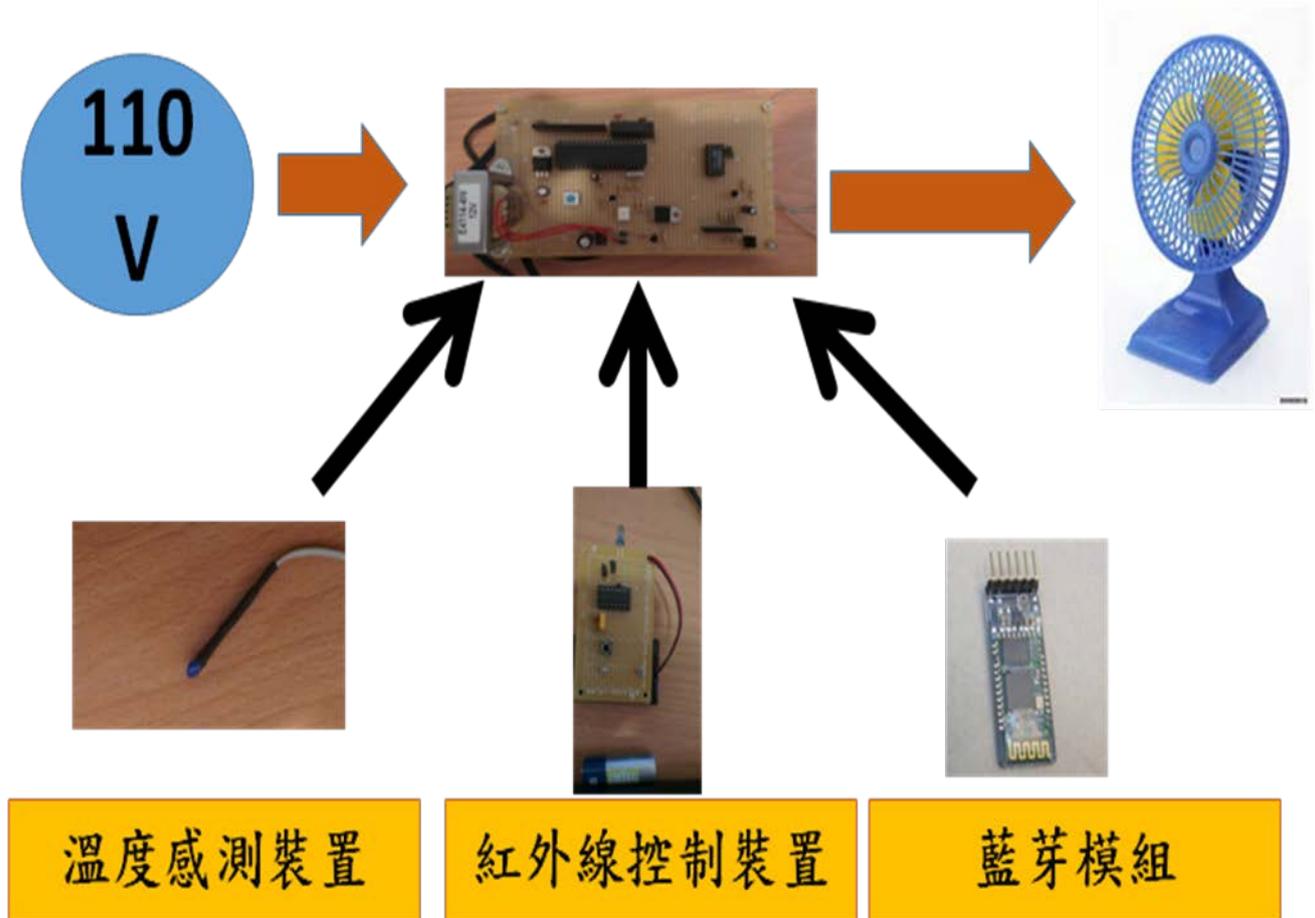
所以可利用藍芽與紅外線來控制風扇開關簡單的方式，在加上能夠自動感測溫度讓風扇能夠自己微調風扇轉速與定時功能，讓風扇使用起來能夠更加便利。

1-2 制作動機：

每年到夏天電風扇就是生活中的必需品，不時又看新聞在報導電費又要調漲，所以想說是否能在風扇上面做些改變讓風扇能夠更便利與省電，但跟冷氣的耗電量來相比的話差很多，可是省電有時是必須從小地方開始。

有時看著家裡的長輩在家裡休息，還要起身起來按電風扇開關再回去繼續休息，每次看到就覺得我專題可以做風扇在風扇上面做些改變，讓風扇看起來更加便利，於是我想說是否可以在專題上加些遠端方面的控制或是溫度感測與定時方面讓風扇使用起來可以更加便利。

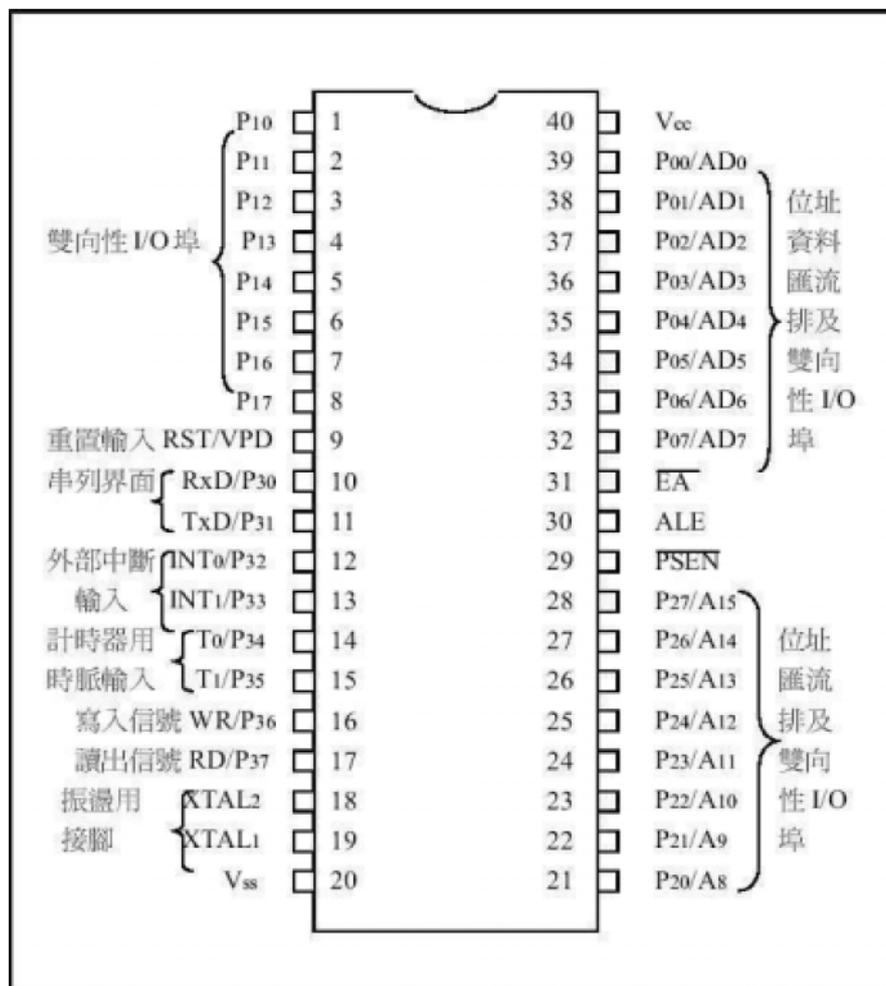
1-3 專題架構圖：



第二章 專題主要元件介紹

2-1 單晶片 8051

所謂單晶片微電腦 (Single Chip Microcomputer)，簡單的說就是：本身除了具備中央處理單元的功能外，將記憶體單元、輸入/輸出單元組合在同一顆晶片內，只須一些支援電路就能獨立作業。



如圖所示

V_{SS}(20)：接地(GND)電源信號。

Vcc(40)：+5V 電源輸入端。

XTAL1(19)，XTAL2(18)：工作頻率輸入信號。

RST/VST(9)：重置(Reset)輸入信號。

/EA(31)：外部存取致能(External Access Enable)輸入信號。決定程式記憶體最前面 bytes(0000H-0FFFH)是來自晶片內部的 ROM/EPROM(/EA=H)，或來自外部的 ROM/EPROM(/EA=L)。注意：由於 8031 晶片無內部的 ROM/EPROM，故 8031 的 /EA 必須接地。

T2(P1.0)：計時器/計數器 2 外部信號輸入端。

T2EX(P1.1)：計時器/計數器 2 在捕捉(Capture)模式時的觸發/重新載入信號輸入端。

RXD(P3.0)：串列埠信號輸入端。

TXD(P3.1)：串列埠信號輸出端。

/INT0(P3.2)：外部中斷 0 信號輸入端。

/INT1(P3.3)：外部中斷 1 信號輸入端。

T0(P3.4)：計時器/計數器 0 外部信號輸入端。

T1(P3.5)：計時器/計數器 1 外部信號輸入端。

/WR(P3.6)：外部資料記憶體寫入閃控(Strobe)信號輸出。

/RD(P3.7)：外部資料記憶體讀取閃控(Strobe)信號輸出。

/PSEN(29)：

程式儲存致能(Program Store ENable)閃控輸出信號。在外部程式記憶體指令碼擷取週期時，/PSEN 將送出一個閃控信號，以表示 CPU 正自外部的程式記憶體中讀取指令碼。

ALE(30)：

位址閃鎖致能(Address Latch Enable)輸出信號。當 CPU 自外界記憶體中擷取指令碼或存取資料時，ALE 將會在一個匯流排週期(Bus Cycle)開始時送出 H 的信號，表示 P0.0-P0.7(AD0-AD7)正送出低階位址 A0-A7 信號，以供外界電路鎖定這些低階位址信號。由於 ALE 信號頻率固定為振盪器工作頻率的 1/6，故 ALE 信號也可當作脈波信號源。



2-2 類比/數位轉換

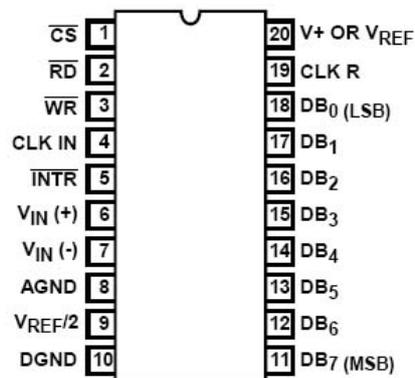
類比/數位轉換器(analog to digital converter)簡稱為 A/D 轉換器(A/D converter)。A/D 轉換器的功能是将輸入之類比信號轉換成數位信號輸出。

(1) 功能：8 位元 A/D 轉換器

(2) 接腳圖：如右圖 (3)

特點：※8 位元 CMOS 連續漸進型 ADC。 ※

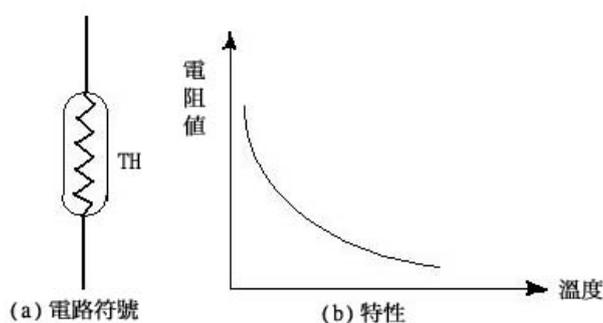
三態門鎖輸出。※最大誤差 $\pm 1\text{LSB}$ 。※內部含有時脈產生電路，頻外加 R、C



2-3 熱敏電阻

量測溫度的高低，熱敏電阻是最常見且便宜的元件，英文名稱為 thermistor。熱敏電阻是一種電阻值對溫度極為敏感的半導體元

件，分為兩種：正溫度係數及負溫度係數，而正溫度係數熱敏電阻 (Positive Temperature Coefficient Thermistors；簡稱PTC熱敏電阻) 為隨著溫度升高，電阻值增加的熱敏電阻，反之，負溫度係數熱敏電阻 (Negative Temperature Coefficient Thermistors；簡稱NTC熱敏電阻) 為隨著溫度升高，電阻值降低的熱敏電阻。



NTC 特性圖

2-4 掃描鍵盤

鍵盤的輸入採用掃描法，由 8051 的埠 1(P1)控制，利用 8051 提供的單一位元定址 I/O，將 P1.0 接到鍵盤 1, P1.1 接到鍵盤 2, 發出掃描信號，而 P1.4 接到鍵盤 A, P1.5 接到鍵盤 B,.....接收掃描信號。掃描時，由埠 P1.0 發出一個低電位，8051 依序檢查信號接收端，若接收到低電位，表示按鍵被按下，否則收到高電位。繼續下一個。讀取

鍵盤時，遇到的主要問題為彈跳，用軟體去彈跳，也就是若偵測到按鍵被按下，先 delay 18ms. 再進行下一個動作。

2-5 紅外線

在按下驅動按鍵後，經由PT2248發射訊號至振盪器，振盪器所振盪出的電波來指示LED發光發射紅外線波，在RPM6938紅外線模組接收到訊號後，將指示的光波指令交由PT2249A將指令使A1015電晶體導通，C極為低態，8051之P3.1腳依訊號來控制風扇開/關。要達到紅外線遙控，必須包含發射模組、接收模組，分別詳述如下：

發射模組

運用按鍵式按鈕，藉由按鍵的方式將此訊號傳給PT2248，好讓PT2248將此訊號轉成脈波，經由震盪再藉由IR LED紅外線發射器發射此訊號。

PT2248(3)是一個CMOS紅外線遙控發射器，它有18種功能及75個指令可供使用，單擊或是連續按鍵皆可，也有組合鍵的功能。可應用於電視、錄放影機的遙控。

PT2248之特性：

1. 低功率消耗。
2. 寬電源供應範圍， $V_{cc}=2.2\sim 5.0\text{ V}$ 。
3. 最多6鍵組合。
4. 需較少外接元件。
5. RC共振當作振盪步率來源。

2-6 Eclipse 開發軟體介紹



Eclipse 是著名的跨平台開源整合式開發環境 (IDE)，最初主要用來 Java 語言開發，目前亦有人透過外掛模組使其作為 C++、Python 等其他語言的開發工具，Eclipse 的本身只是一個框架平台，但是眾多外掛模組的支援，使得 Eclipse 擁有較佳的靈活性。

版；2009 年 6 月發行代號為 Galileo 的 3.5 版；2010 年 6 月發行代號為 Helios 的 3.6 版。

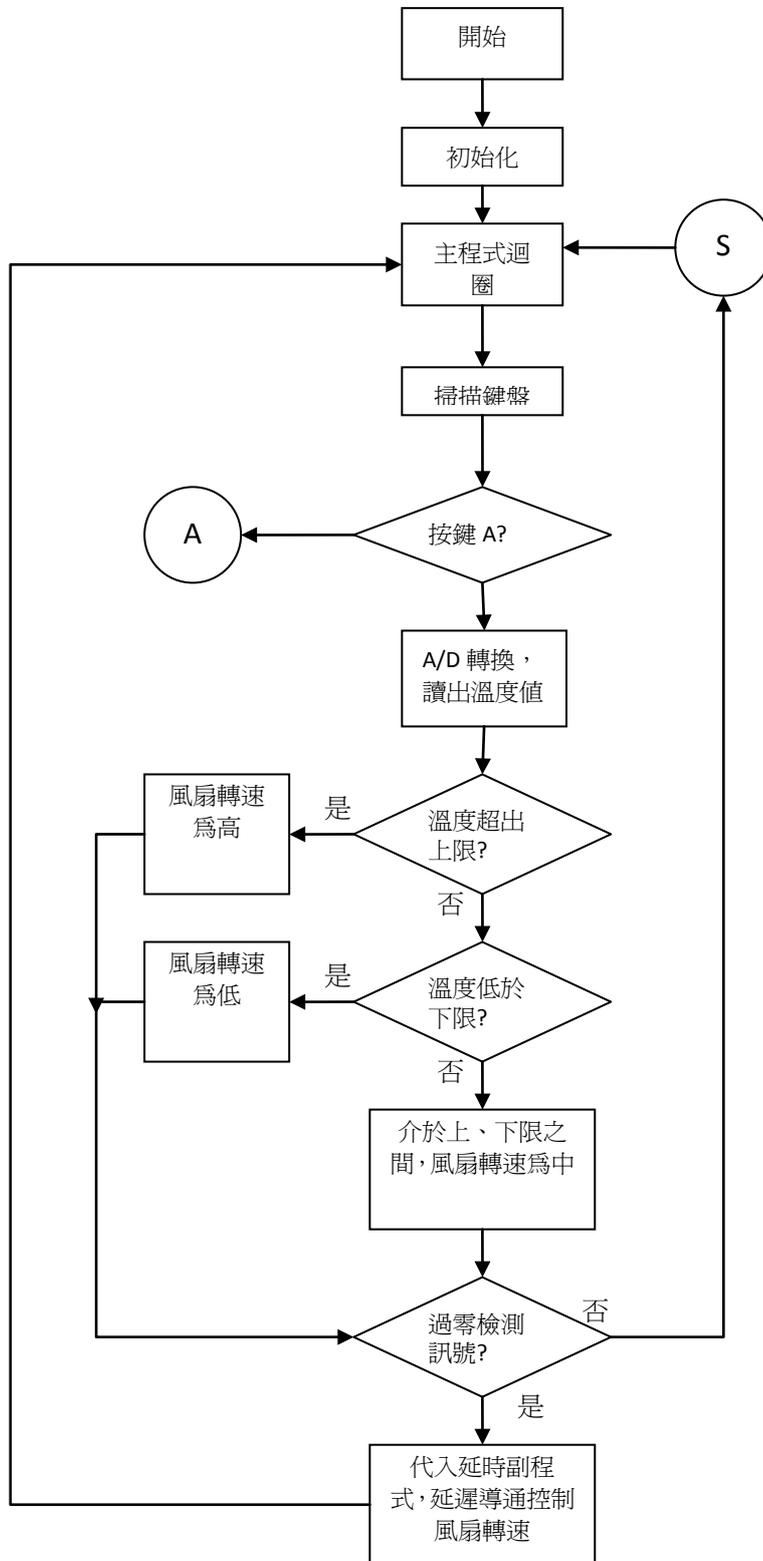
一、架構：

Eclipse 的基礎是富客戶機平台（即 RCP）。RCP 包括下列元件

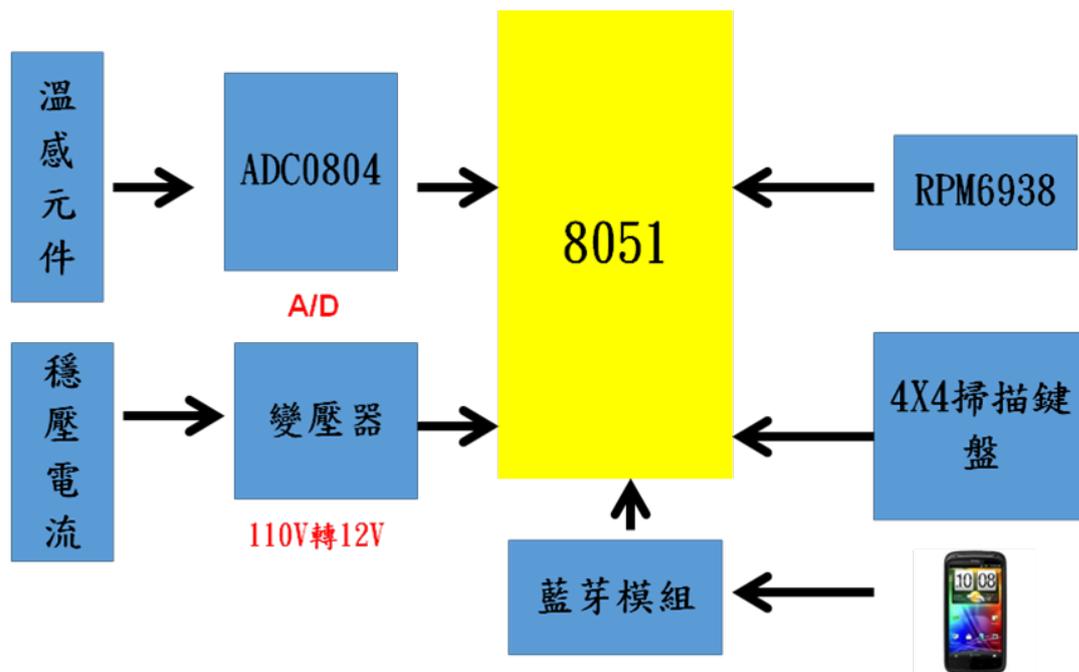
- 核心平台（啟動 Eclipse，執行外掛模組）。
- OSGi（標準框架）。
- SWT（可移植構件工具包）。
- JFace（檔案緩衝，文字處理，文字編輯器）。
- Eclipse 工作台（Workbench），包含檢視（views）、編輯器（editors）、視角（perspectives）、和精靈（wizards）。

第三章 軟硬體實作

3-1 程式流程圖：



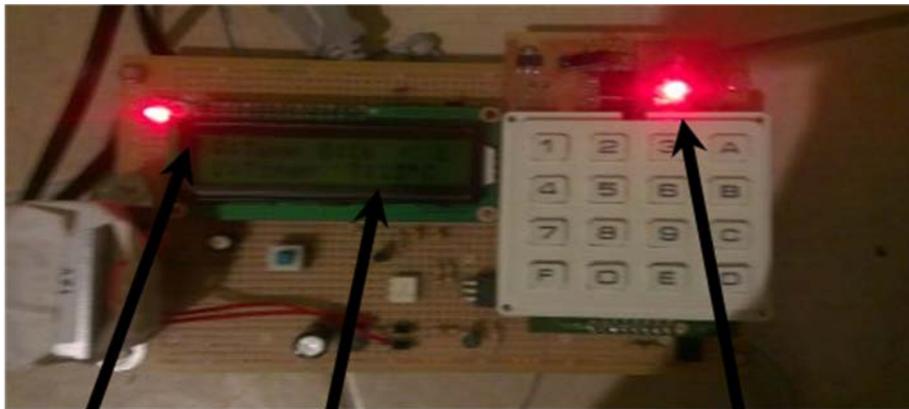
3-2 硬體電路：



3-3 測試狀況：



關機時



開機時

指示燈

顯示數字

藍芽指示燈會閃爍

第五章總結

結論：

這次專題從剛開始不知做什麼專題到回家看著家中長輩每次要切換電扇還要起身無法讓他們安靜的休息，於是開始慢慢在專題上立下目標，可以不需起身就能夠輕鬆的操作風扇。

專題期間雖然口頭上想加入功能，但時常遇到程式方面的困難，還好有朋友願意幫我們，願意教導我們如何寫，讓我們能夠在寫出這些功能，中途雖然時常程式寫出時發現功能沒出來，卻容易以為是程式方面又出錯，卻到頭來是電路接錯導致功能沒出來，但中間程式經過好幾次的微調，才有今日的效果，做專題過程中時常想著為什麼要做那麼辛苦，腦中想著為何不再找個更簡單的專題目標做呢？但卻又想著現在都以這個目標在努力了，又換個目標一切都要從頭開始讓我們保持著這種心態繼續做與不想延畢被當的心態繼續做下去。

這是專題加上溫度感測、計時、紅外線與藍芽功能，經過上述等功能，經過小小的測試，確實可以達到便利與省電效果，但是藍芽與紅外線卻會因為障礙物導致無法正常接收，讓我們必須再去好好檢討如何在有障礙物收訊還可以正常操作與距離拉長。

參考文獻

- 吳朗，感測與轉換原理元件與應用，全欣資訊圖書
- 蔡朝洋，單晶片微電腦8051原理與應用，全華
- 陳明榮，單晶片8051實作入門，文魁
- ADC0804的DATA SHEET
- 89S51的DATA SHEET
- 文淵閣工作室，Android 初學特訓班(第三版)，基峯資訊股份有限公司。
- 陳會安，Android SDK程式設計範例教本，旗標出版股份有限公司。

專題組員介紹

組長：

班級：四技電機四年乙班

學號：BD99058

姓名：紀荃淵

工作項目：硬體架構、撰寫報告

專題製作心得：

我們這組作的是微電腦控制風扇，主要是因為夏天的時候看到家裡的長輩們在休息時，還要起身走動啟動開關就覺得很麻煩，無法讓他們好好休息，於是就想說能在適當的溫度中吹著想要的溫度，能夠好好的休息，在加上遠端控制讓長輩們可以不需起身走動，剛開始作時，遇到蠻多問題，但這些問題能夠讓我從中學習到許多我不了解的事，這次專題讓我從中學習到很多，也感謝老師不停丟問題給我們讓我們可以從中學習。

組員：

班級：四技電機四年乙班

學號：BD990100

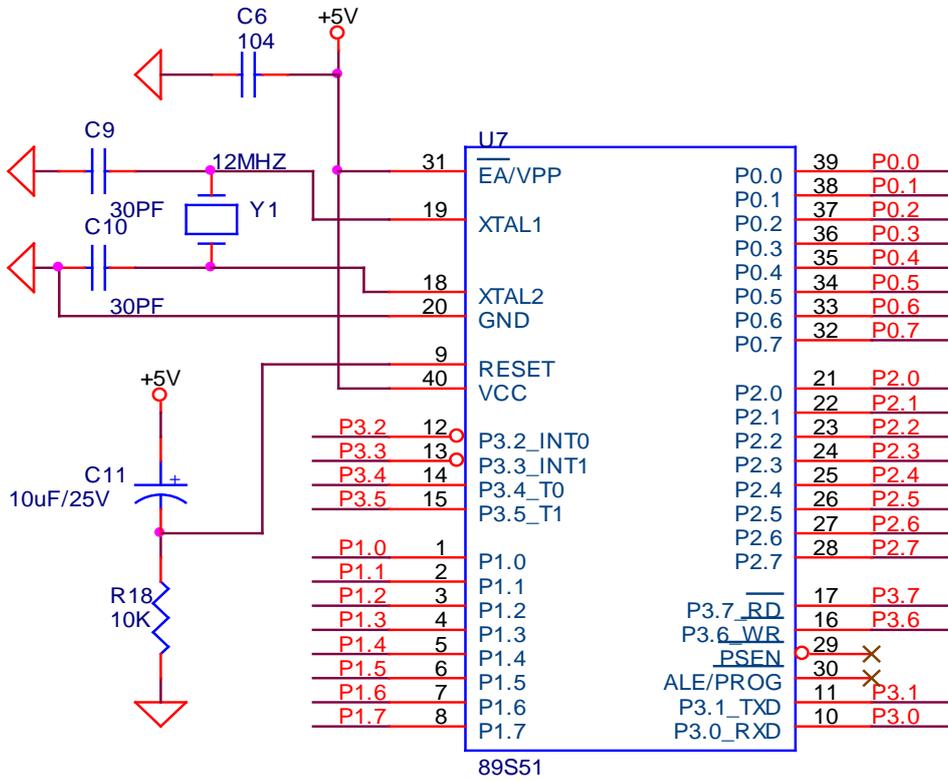
姓名：張曜任

工作項目：撰寫 C++程式、撰寫 Android 程式

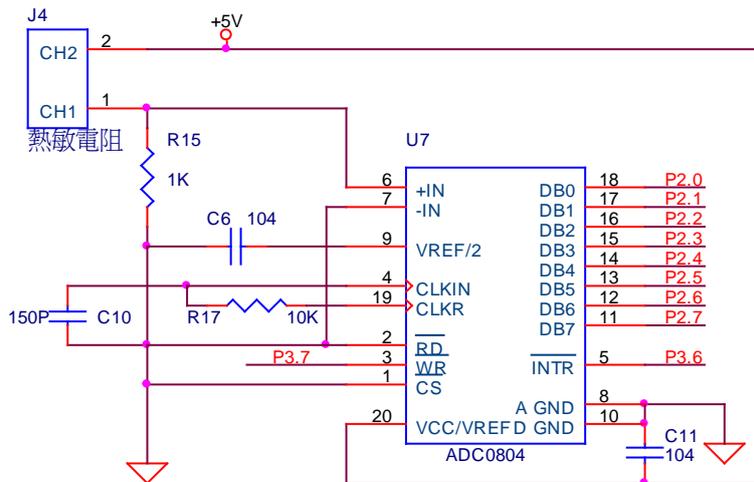
專題製作心得：

這個專題真的對我來說是一大挑戰，主要是想做些有趣的題目，工具最好是手邊就有的，但沒想到難度還蠻大的，程式的部分就讓我困擾了一段時間，解決了一個問題又有另一個問題出來，可是跟專題指導老師以及組長一起討論過後問題也漸漸解決了，這次專題實作讓我學到很多，也感謝老師的指導還有一路上幫助我們度過難關的人讓我們能夠有這樣的成果。

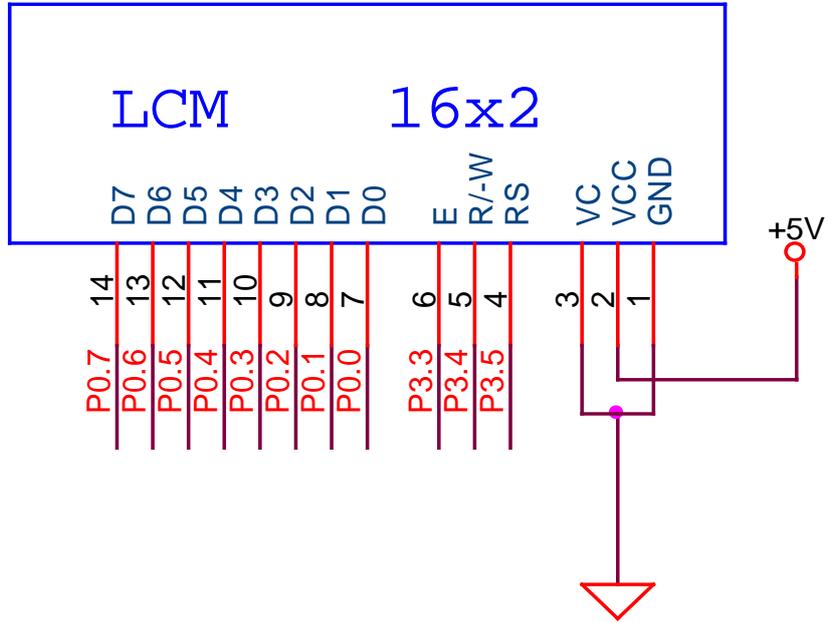
附錄一



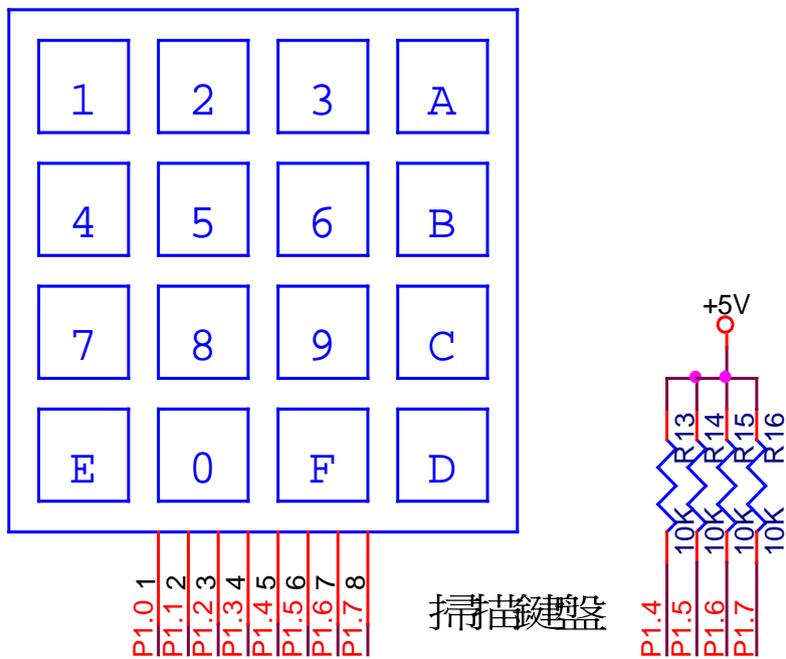
8051 電路



ADC0804 電路

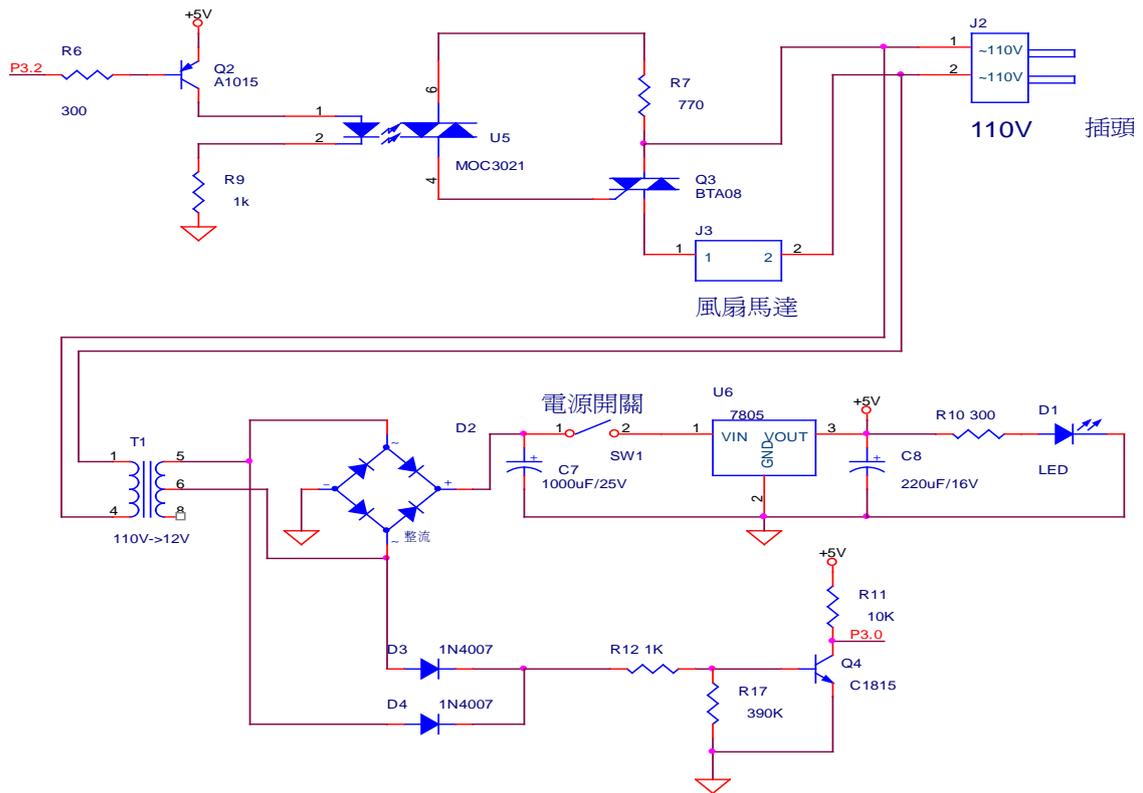


LCD 電路

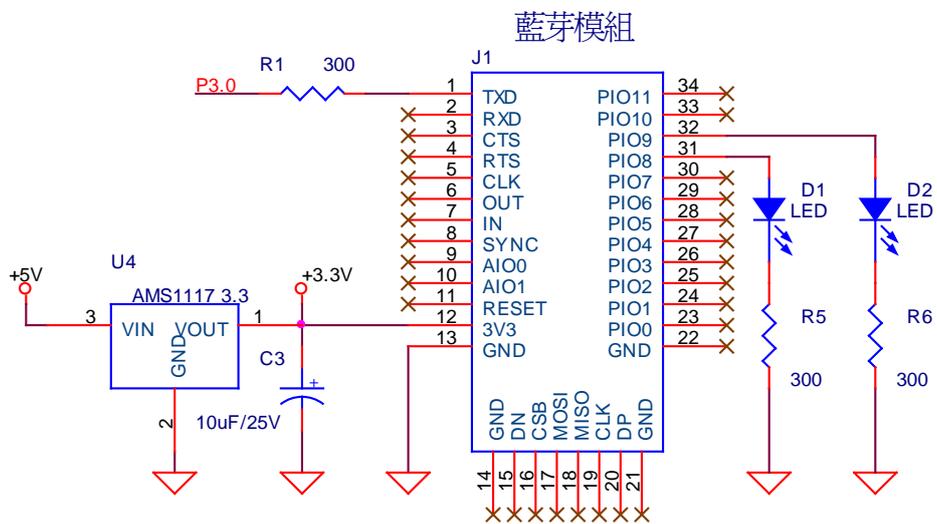


掃描鍵盤

鍵盤電路



紅外線電路



藍芽電路

附錄二

零件表

項目	數量	編號	規格	
1	1	BT1	3V	
2	1	C1	102	陶瓷電容
3	1	C2	100UF	電解電容
4	2	C3,C12	10uF/25V	電解電容
5	3	C4,C6,C11	104	陶瓷電容
6	2	C9,C5	30PF	陶瓷電容
7	1	C7	1000uF/2V	電解電容
8	1	C8	220uF/16V	電解電容
9	1	C10	150P	陶瓷電容
10	2	C13,C14	120PF	陶瓷電容
11	3	D1,D2,D3	LED	
12	1	D4	整流	
13	2	D5,D6	1N4007	二極體
14	1	D7	1N4148	二極體
15	1	D8	IR LED	紅外線發射

LED

16	1	J1	藍芽模組	
17	1	J2	110V 插頭	
18	1	J3	風扇馬達	
19	1	J4	熱敏電阻	
20	3	Q1,Q2,Q5	A1015	電晶體 PNP
21	1	Q3	BTA08	雙向矽控整流器
22	2	Q6,Q4	C1815	電晶體 NPN
23	5	R1,R5,R6,R8,R16	300	電阻
24	2	R7,R2	39K	電阻
25	8	R3,R10,R11,R12,R13,R17, R18,R19	10K	電阻
26	1	R4	47	電阻
27	1	R9	770	電阻
28	3	R14,R15,R20	1K	電阻
29	1	R21	390K	電阻
30	1	SW1	電源開關	
31	1	SW2	開/關	
32	1	T1	110V->12V	變壓器線圈

33	1	U1	IR RECEIVER	紅外線接收模 組
34	1	U2	LCD	16X2 文字型 LCD
35	1	U3	PT2249	紅外線解碼 IC
36	1	U4	AMS1117 3.3	3.3V 穩壓 IC
37	1	U5	KEYING	4X4 鍵盤
38	1	U6	MOC3021	SCR 光耦合 IC
39	1	U7	ADC0804	類比轉數位 IC
40	1	U8	89S51	單晶片
41	1	U9	7805	5V 穩壓 IC
42	1	U10	PT2248	紅外線編碼 IC
43	1	Y1	12MHZ	振盪器
44	1	Y2	455K	振盪器

附錄三

```

EN      REG      P3.3      ;
RS      REG      P3.5      ;
ADDR    EQU      37H
CNT     EQU      38H
SEC     EQU      39H
MIN     EQU      3AH

        ORG      00H      ;
        AJMP     START    ;
        ORG      0BH      ;
        AJMP     TIMER0   ;

        ORG      0023H    ;
        CLRRI
RECEV:  MOV      A,SBUF    ;
        CJNE    A,#'G',OKBACK ;
        CPL 20H.0        ;
NON:    CLR TI
OKBACK:
        RETI
;-----
; MAIN
;-----
START:  MOV      SP,
        MOV      30H,#100;
        MOV      32H,#2    ;
        MOV      31H,#50   ;
        CALL     INIT      ;
        CALL     ON        ;
        MOV      48H,#3    ;
        MOV      49H,#0    ;
        MOV      4AH,#2    ;
        MOV      4BH,#5    ;
        MOV      4CH,#0    ;
        MOV      4DH,#0    ;
        MOV      4EH,#1    ;

        CALL     CONVERT_T ;
        CALL     CONVERT   ;
        ACALL    SET0      ;
        CLR20H.0          ;
        CLR20H.1          ;
LOOP:   ACALL    SCAN      ;
        MOV      A,R5      ;
        CJNE    A,#CH,L3   ;
        SETB    P3.2        ;
        AJMP    F2          ;
L3:    CJNE    A,#EH,L4    ;
        SETB    P3.2        ;
        SETB    20H.1        ;

```

```

        AJMP    F4          ;
L4:     CJNE   A,#BH,L5    ;
        CPL 20H.0          ;
L5:     JNB P3.1,L6        ;
        JB  P3.1,$         ;
        CPL 20H.0          ;
L6:
        JNB 20H.0,LW3     ;
        JNB P3.4,LOOP     ;
LW:
        JB  P3.4,LW        ;
        MOV  A,31H         ;
        CJNE A,#0,LW2     ;
        JMPLW3
LW2:
        CALL DLX          ;
        CLR P3.2          ;
LW3:
        CALL DELAYB       ;
        SETB P3.2         ;
        DJNZ 30H,LOOP     ;
        MOV  30H,#100     ;
        DJNZ 32H,LOOP     ;
        MOV  32H,#2       ;
        JNB 20H.0,LW4     ;
        CLR P3.2
LW4:   CALL  TEMPUA       ;
        SETB P3.2
        JMP LOOP
;=====
;===延時副程式=====
;=====
DLX:
        MOV  R4,31H
DLX1:  CALL  DELAYY
        DJNZ R4,DLX1
        RET
DELAYY: MOV  R2,#18H      ;
DELAY1:
        DJNZ R2,DELAY1
        RET
;=====
;===延時副程式=====
;=====
DELAYB: MOV  R2,#FFH     ;
DELAY1B:
        NOP
        NOP
        DJNZ R2,DELAY1B
        RET
;=====
;=A/D 轉換讀取溫度副程式=====
;=====
TEMPUA:
        ACALL ATOD
        MOV  R3,#00H
LOOP1: CLR C

```

```

ACALL READTEMP ;
INC R3 ;
MOV A,B ;
SUBB A,R1 ;
JNC LOOP1 ;
;
;
DEC R3 ;
DEC R3
MOV A,#CBH ;
ACALL W_INS ;
MOV A,R3
MOV B,#10
DIV AB
ADD A,#30H ;
ACALL W_DATA ;
MOV A,B
ADD A,#30H ;
ACALL W_DATA ;
CALL TEMPCONTROL ;
RET
;----
; F2 ;
;----
F2:
ACALL CLRLCD
MOV R0,#48H ;
MOV A,#80H ;
ACALL W_INS ;
MOV CNT,#16
MOV DPTR,#MSG3 ;
ACALL W_STR
MOV A,#C0H ;
ACALL W_INS ;
MOV CNT,#16
MOV DPTR,#MSG4 ;
ACALL W_STR
MOV A,#C5H ;
ACALL W_INS ;
MOV A,48H
ADD A,#30H ;
CALL W_DATA ;
MOV A,49H
ADD A,#30H ;
CALL W_DATA ;

MOV A,#C5H ;
ACALL W_INS ;
F21: ACALL SCAN ;
MOV A,R5 ;
XRL A,#16 ;
JZ F21
MOV A,R5

F22: CJNE A,#BH,F23
JMP F21
F23: CJNE A,#CH,F24 ;

```

```

        JMP F21
F24:    CJNE    A,#DH,F25    ;
        AJMP   SETF2
F25:    CJNE    A,#EH,F26    ;
        JMP F21
F26:    CJNE    A,#FH,F27    ;
        JMP F21
F27:
        CJNE    A,#AH,F28    ;
        JMP F21
F28:

        MOV     A,R5
        ORL     A,#30H        ;
        MOV     @R0,A         ;
        ACALL   W_DATA        ;
        INC     R0
        CJNE    R0,#4AH,F29
        MOV     R0,#48H
        MOV     A,#C5H        ;
        ACALL   W_INS         ;
F29:    SJMP    F21
;=====
; F2-SET
;=====
SETF2:
;----
; F3          ;
;----
F3:
        MOV     A,#CCH        ;
        ACALL   W_INS         ;
        MOV     A,4AH         ;
        ADD     A,#30H        ;
        CALL    W_DATA        ;
        MOV     A,4BH         ;
        ADD     A,#30H        ;
        CALL    W_DATA        ;
        MOV     R0,#4AH       ;
        MOV     A,#CCH        ;
        ACALL   W_INS         ;
F31:    ACALL   SCAN          ;
        MOV     A,R5          ;
        XRL    A,#16         ;
        JZ     F31
        MOV     A,R5

F32:    CJNE    A,#BH,F33    ;
        JMP F31
F33:    CJNE    A,#CH,F34    ;
        JMP F31
F34:    CJNE    A,#DH,F35    ;
        AJMP   SETF3
F35:    CJNE    A,#EH,F36    ;
        JMP F31
F36:    CJNE    A,#FH,F37    ;
        JMP F31

```

```

F37:
    CJNE    A,#AH,F38        ;
    JMP F31
F38:
    MOV     A,R5
    ORL    A,#30H           ;
    MOV     @R0,A           ;
    ACALL   W_DATA          ;
    INC     R0
    CJNE   R0,#4CH,F39
    MOV     R0,#4AH
    MOV     A,#CCH          ;
    ACALL   W_INS           ;
F39:    SJMP    F31
;=====
; F3-SET
;=====
SETF3:

    CALL    CONVERT_T ;
    AJMP    LOOP

;----
; F4
;----
F4:
    ACALL   CLRLCD
    MOV     R0,#4CH         ;
    MOV     A,#80H         ;
    ACALL   W_INS          ;
    MOV     CNT,#16
    MOV     DPTR,#MSG5     ;
    ACALL   W_STR
    MOV     A,#C0H         ;
    ACALL   W_INS          ;
    MOV     CNT,#16
    MOV     DPTR,#MSG6     ;
    ACALL   W_STR

    MOV     A,#C4H         ;
    ACALL   W_INS          ;

    MOV     A,4CH          ;
    ADD     A,#30H          ;
    CALL    W_DATA         ;
    MOV     A,4DH          ;
    ADD     A,#30H          ;
    CALL    W_DATA         ;
    MOV     A,4EH          ;
    ADD     A,#30H          ;
    CALL    W_DATA         ;
    MOV     A,#C4H         ;
    ACALL   W_INS          ;

```

```

F41:  ACALL  SCAN      ;
      MOV   A,R5      ;
      XRL  A,#16      ;
      JZ   F41
      MOV  A,R5

F42:  CJNE  A,#BH,F43 ;
      JMP  F41
F43:  CJNE  A,#CH,F44 ;
      JMP  F41
F44:  CJNE  A,#DH,F45 ;
      AJMP F41
F45:  CJNE  A,#EH,F46 ;
      JMP  F41
F46:  CJNE  A,#FH,F47 ;
      JMP  SETF4
F47:  CJNE  A,#AH,F48 ;
      JMP  F41
F48:
      MOV  A,R5
      ORL  A,#30H    ;
      MOV  @R0,A     ;
      ACALL W_DATA   ;
      INC  R0
      CJNE R0,#4FH,F49
      MOV  R0,#4CH
      MOV  A,#C4H    ;
      ACALL W_INS    ;
F49:  SJMP  F41
;=====
; F4-SET
;=====
SETF4:
      CALL CONVERT   ;
      MOV  SEC,#100  ;
      MOV  MIN,#60
      AJMP LOOP
; SET TIMER0
;-----
SET0:  MOV  SEC,#100 ;
      MOV  MIN,#60
      MOV  TMOD,#21H ;
      MOV  TH0,
      MOV  TL0,
      SETB EA      ;
      SETB ET0
      SETB TR0
      MOV  TH1,#253 ;
      MOV  TL1,#253
      SETB TR1      ;
      MOV  SCON,#01110000B ;
      CLRRI         ;
      CLRTI         ;
      SETB ES      ;

```

```

                RET
;-----
; TIMER0
;-----
TIMER0: PUSH    A                ;
        MOV     TH0,#>(65536-10000)
        MOV     TL0,#<(65536-10000)
        DJNZ   SEC,EXIT3        ;
        MOV     SEC,#100
        DJNZ   MIN,EXIT3        ;
        MOV     MIN,#60
        DJNZ   5AH,EXIT3        ;
        MOV     5AH,5BH
        JNB    20H.1,EXIT3       ;
        SETB   P3.2             ;
        CLR    20H.0
        CLR    20H.1
EXIT3:  POP     A
        RETI

```

```

;=====
;====按鍵值轉成關機時間值====
;=====
CONVERT:

```

```

        MOV     A,4CH           ;
        ANL     A,#0FH
        MOV     B,#100         ;
        MUL     AB
        MOV     4FH,A

        MOV     A,4DH           ;
        ANL     A,#0FH
        MOV     B,#10          ;
        MUL     AB
        MOV     B,A
        MOV     A,4EH           ;
        ANL     A,#0FH
        CLRC
        ADD     A,B             ;
        MOV     B,4FH
        ADD     A,B
        MOV     5BH,A          ;
        MOV     5AH,A
        CALL    ON
        RET

```

```

;=====
;====按鍵值轉成溫度比較值====
;=====
CONVERT_T:

```

```

        MOV     A,48H           ;
        ANL     A,#0FH
        MOV     B,#10          ;
        MUL     AB
        MOV     B,A
        MOV     A,49H           ;
        ANL     A,#0FH

```

```

CLRC
ADD    A,B    ;
MOV    5EH,A  ;

MOV    A,4AH  ;
ANL    A,#0FH
MOV    B,#10  ;
MUL    AB
MOV    B,A
MOV    A,4BH  ;
ANL    A,#0FH
CLRC
ADD    A,B    ;

MOV    5CH,A  ;

CALL   ON
RET

```

```

;=====
; SCAN KEY
;=====
SCAN:  MOV    R4,#00H    ;
        MOV    R7,#4    ;
        MOV    R6,#1111110B ;
COL:   MOV    A,R6
        MOV    P1,A
        JB     P1.4,S1  ;
        JNB    P1.4,$   ;

        ACALL  DELAY
        ACALL  DELAY
        ACALL  DELAY
        SJMP   S5    ;
S1:    INC    R4
        JB     P1.5,S2
        JNB    P1.5,$   ;
        ACALL  DELAY
        ACALL  DELAY
        ACALL  DELAY
        SJMP   S5    ;
S2:    INC    R4
        JB     P1.6,S3
        JNB    P1.6,$   ;
        ACALL  DELAY
        ACALL  DELAY
        ACALL  DELAY
        SJMP   S5    ;

```

```

S3:   INC     R4
      JB     P1.7,S4
      JNB    P1.7,$
      ACALL  DELAY
      ACALL  DELAY
      ACALL  DELAY
      SJMP   S5   ;
S4:   INC     R4
      MOV    A,R6
      RL    A     ;
      MOV    R6,A
      DJNZ  R7,COL

S5:   CJNE   R4,#00H,S6
      MOV    R5,#01H
      RET
S6:   CJNE   R4,#01H,S7
      MOV    R5,#04H
      RET
S7:   CJNE   R4,#02H,S8
      MOV    R5,#07H
      RET
S8:   CJNE   R4,#03H,S9
      MOV    R5,#0AH
      RET
S9:   CJNE   R4,#04H,SA
      MOV    R5,#02H
      RET
SA:   CJNE   R4,#05H,SB
      MOV    R5,#05H
      RET
SB:   CJNE   R4,#06H,SC
      MOV    R5,#08H
      RET
SC:   CJNE   R4,#07H,SD
      MOV    R5,#00H
      RET
SD:   CJNE   R4,#08H,SE
      MOV    R5,#03H
      RET
SE:   CJNE   R4,#09H,SF
      MOV    R5,#06H
      RET
SF:   CJNE   R4,#0AH,SG
      MOV    R5,#09H
      RET
SG:   CJNE   R4,#0BH,SOK
      MOV    R5,#0BH
      RET

SOK:  MOV    A,R4
      MOV    R5,A
      RET
;-----
; POWER ON
;-----
ON:   PUSH   A

```

```

MOV      A,#80H      ;
ACALL   W_INS       ;
MOV      CNT,#16
MOV      DPTR,#MSG1 ;
ACALL   W_STR
MOV      A,#C0H      ;
ACALL   W_INS       ;
MOV      CNT,#16
MOV      DPTR,#MSG2 ;
ACALL   W_STR

MOV      A,#CDH      ;
ACALL   W_INS       ;
MOV      A,#11011111B;
ACALL   W_DATA
POP      A
RET

;-----
; INIT LCD
;-----
INIT:
MOV      A,#38H
ACALL   W_INS       ;
MOV      A,#08H
ACALL   W_INS       ;
MOV      A,#01H      ;
ACALL   W_INS       ;
MOV      A,#06H      ;
ACALL   W_INS       ;
MOV      A,#0CH      ;
ACALL   W_INS       ;
RET

;-----
; CLEAR LCD
;-----
CLRLCD: MOV      A,#01H
ACALL   W_INS       ;
RET

;=====
;====溫度比對副程式====
;=====
TEMPCONTROL:
MOV      A,5EH      ;
CLRC    ;
SUBB    A,R3        ;
JNC    NACT        ;

```

```

        MOV     A,#8FH      ;
        ACALL  W_INS       ;
MOV     A,#'H'           ;
        ACALL  W_DATA      ;
MOV     31H,#1          ;
        JMP   CONTRBACK
NACT:
        MOV     A,5CH      ;
        CLRC                    ;
        SUBB   A,R3        ;
        JC    TOK          ;
        MOV     A,#8FH      ;
        ACALL  W_INS       ;
MOV     A,#'L'           ;
        ACALL  W_DATA      ;
MOV     31H,#70         ;
        JMP   CONTRBACK

TOK:
        MOV     A,#8FH      ;
        ACALL  W_INS       ;
MOV     A,#'M'           ;
        ACALL  W_DATA      ;

        MOV     31H,#50    ;
        JMP   CONTRBACK

```

```

CONTRBACK:
    RET

```

```

;-----
; WRITE STRING TO LCD
;-----
W_STR:  PUSH   A
        MOV   R5,#0
STR:    MOV   A,R5
        MOVC  A,@A+DPTR
        ACALL W_DATA      ;
        INC  R5
        DJNZ CNT,STR
        POP  A
        RET

```

```

; =====
;  =把指令送入 LCD 模組=
; =====
W_INS:  CLRRS      ;

```

```

        CLREN      ;
        NOP                    ;
        SETB   EN        ;
        MOV   P0,A        ;
        NOP                    ;
        CLREN      ;E=0
        ACALL DLY1      ;

```

```

    RET
;
; =====
; ==把資料送入 LCD 模組==
; =====
W_DATA:  SETB  RS      ;

        CLREN      ;
        NOP         ;
        SETB  EN    ;
        MOV   P0,A   ;
        NOP         ;
        CLREN      ;
        ACALL DLY2   ;
        RET

DLY1:  MOV   R6,
DL3:   MOV   R7,
        DJNZ R7,
        DJNZ R6,
        RET
DLY2:  MOV   R6,
        DJNZ R6,
        RET

;-----
; DELAY
;-----
DELAY:  MOV   R6,
D1:     MOV   R7,
        DJNZ R7,
        DJNZ R6,
        RET

;
; =====
; == A/D 轉換副程式 ==
; =====
; 起動 ADC0804
ATOD:
        CLR   P3.7   ;
        SETB  P3.7   ;
;
WAIT:  JB   P3.6,WAIT ;
;
        MOV   A,P2    ;
        MOV   B,A     ;
        RET

;
; =====
; ==讀溫度資料副程式==
; =====
READTEMP:MOV   DPTR,
        MOV   A,R3
        MOVC  A,@A+DPTR
        MOV   R1,A
        RET

;
; =====
; == 溫度資料表 ==

```

; =====
TABLE:

- DB 64
- DB 66
- DB 68
- DB 70
- DB 72
- DB 74
- DB 76
- DB 78
- DB 80
- DB 83
- DB 85
- DB 87
- DB 90
- DB 92
- DB 95
- DB 98
- DB 101
- DB 103
- DB 105
- DB 108
- DB 110
- DB 112
- DB 115
- DB 118
- DB 120
- DB 123
- DB 125
- DB 128
- DB 130
- DB 133
- DB 135
- DB 138
- DB 140
- DB 142
- DB 145
- DB 148
- DB 150
- DB 152
- DB 155
- DB 157
- DB 159
- DB 161
- DB 163
- DB 165
- DB 167
- DB 169
- DB 171
- DB 173
- DB 175
- DB 177
- DB 179
- DB 181
- DB 183
- DB 185

DB 187
DB 189
DB 191
DB 193
DB 195
DB 197
DB 199
DB 201
DB 203
DB 205
DB 207
DB 209
DB 211
DB 213
DB 215
DB 217
DB 219

```
=====
;-----初値設定-----
=====
```

```
;-----
; MESSAGE
;-----
MSG1:  DB      'A:Temp B:Ok      '
MSG2:  DB      'C:Timer T:  C '
MSG3:  DB      '  Set Temp:    '
MSG4:  DB      'High:  Low:    '

MSG5:  DB      '  Set Timer:    '
MSG6:  DB      '    000 Min    '
```

END