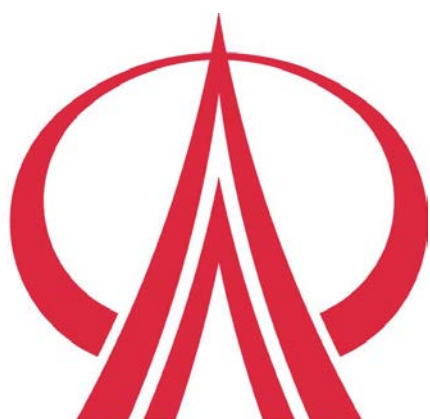


# 修平科技大學 電機工程系

DEPARTMENT OF ELECTRICAL ENGINEERING  
HSIUPING UNIVERSITY OF SCIENCE AND TECHNOLOGY

## 實務專題報告書

### 冷氣自動化之節能控制



指導老師：陳正宗

專題製作學生：

四技電四甲 王建順 BD99507

四技電四甲 江仕堯 BD99035

中華民國 一零二 年 十二 月 三十 日

# 摘要

1. 前言(緒論)	2
1-1 動機	3
1-2 目的	4
2. 相關原理	
2-1 溫溼度感測器	5
2-2 紅外線發射原理	6
2-3 主程式板子傳輸原理	9
2-4 CC2530 介紹	11
3. 系統設計圖	
3-1 溫溼度感測器圖	13
3-2 紅外線發射原理圖	14
3-3 主程式板子傳輸原理圖	15
4. 結果與結論	

## 前言：

冷氣機在近年來，已經是夏天不可或缺的家庭用品。雖然冷氣是一項讓我們可以舒舒服服度過夏天的一項利器，但是冷氣機帶來的影響並不全然是正面的；就拿台北為例，台北夏天酷暑難耐，除了地形關西外，台北人幾乎 24 小時都把冷氣打開，因為大量的使用冷氣機，造成熱空氣不斷的被製造出來，使得戶外的溫度快速升高，同時也造成了許多不必要的浪費。

尤其是最近大家為了核四鬧得沸沸揚揚的，也因為核一核二廠在近十年內就要關閉，使得供電量的減少，而需求量增加的情況下，漲電費已經是不可避免的事情了，因此大家都在計較該如何節約用電，才能讓電費少一些，但是說的永遠比做的輕鬆，若要解決此問題最簡單的口號就是開源與節流。只不過台灣是海島型國家，本身並不出產礦類等所需的燃料，而這些燃料我們必須仰賴海運進口，所以電來的十分不容易，既然無法在近期內開源的情況下，那節流在相比之下要來實行就顯得容易許多

## 動機

因為全球的暖化，又加上學生一進教室就是把冷氣調到最低溫度，因此本專題想出如何優化使用冷氣機舒適與最省電的平衡點，也因核四鬧得沸沸揚揚的，必盡核一核二廠在近十年內就要關閉，使得供電量的減少，而需求量增加的情況下，漲電費已經是不可避免的事情了，因此大家都在計較該如何節約用電，才能讓電費少一些。

## 目的

冷氣機在近年來，已經是夏天不可或缺的家庭用品，但也因為冷氣機使用過度，造成熱空氣不斷的被冷氣機大量排放出來，而本專題將使冷氣機達到舒適、省電之平衡，因此本專題將冷氣機使用成自動控制化，達到調賤不到時不與啟動冷氣機，而是以送風或除濕來代替冷氣這項功能。

# 相關原理

## 2-1 溫溼度感測器

**1. 溫度感測器：**一般常見的溫度感測器有熱電偶、電阻式與熱敏電阻，而感測元件可以為金屬或是半導體，而感測元件都以金屬包覆以保護元件，而感測的準確性將影響其價格。

**2. 濕度感測器：**濕度感測器具有兩種不同的型式，其一為電容變化式高分子溼度感測器，其動作為膜上下各蒸鍍一電極膜片，上方之電極為多孔性用以吸收水份，使水分子上升或是下降，來判定環境的濕度數值為其多少。其二為電阻式高分子濕度感測器，其動作為利用材料在吸附水分子之後，能藉由電荷遷移或解離的機構，造成阻抗的降低機制所用，而阻抗降低與相對濕度增加的線性關係即為阻抗式感測器的原理。

# 相關原理

## 2-2 紅外線發射器

前最常見的一種無線通訊，普遍使用在家電以及玩具產品，如電視、音響、錄放影機、冷氣機、DVD、MP3 Player、遙控車等。紅外線遙控之所以被大量採使用，主要是因為紅外線裝置體積小、成本低、耗電少及硬體設計容易。下圖是紅外線發射器(Transmitter 或稱 IR LED)和接收器(Receiver)常見外觀，一般來說，紅外線遙控系統由發射器和接收器這兩部份組成。

可見光，其實生活中充滿了紅外線，只是我們看不到。紅外線主要來自太陽，不過很多物體也會發射紅外線，例如燈泡、蠟燭、中央空調等，甚至人體也會散發紅外線。人體所發出的紅外線的量是可以偵測的，耳溫槍就是利用這個道理測量人的體溫。有這麼多紅外線光源，當然會對遙控造成干擾，所以得做一些預防措施確保通訊正確。

的解藥是 Modulation。我們講話速度若適當，不徐不急，聽得舒服，聽者自然不漏接。相同的道理，利用 Modulation 讓 IR LED 以特定的頻率閃爍，Receiver 端也調整到同樣的頻率，便可以忽略干

擾。在上圖中，可以看到調變訊號 (Modulated signal) 在驅動 IR LED 發射訊號，而偵測到的訊號則從右手邊的 receiver 跑出來。

## 相關原理

在 Serial 通訊中常會提到 mark 和 space 狀態。space 是紅外線的預設訊號，Transmitter 處於 off 狀態，這時 IR LED 不會發射光亮；而在 mark 狀態 IR LED 會以特定的頻率送出 on/off 脈衝 (Pulse)。消費電子一般使用 30kHz 到 60kHz 的頻率。

### 紅外線發射器：一般由三大部分組成：

#### 1. 按鍵式掃描矩陣：

由集成電路的掃描輸出、輸入電路引腳組成橫豎交叉矩陣。無鍵按下時，輸入輸出互不相連。輸入口（即 KI）為低電平，當某一鍵按下時，相應的輸入口即有信號送達，使專用集成電路得知哪一個按鍵被按下。每一隻按鍵對應一組編碼。如 NEC6121 集成電路共有 32 組不同的編碼，NEC6122 集成電路則有 64 組不同的編碼。在實際使用中，當兩鍵同時按下時，不輸出信號。當然，也有一些電路特設“雙鍵”，當指定的雙鍵按下時，它會發出一種指定的信號



**2. 專用集成電路：**俗稱遙控器專用集成電路遙控器專用集成電路（俗稱發射塊）是遙控器的核心部分。一般情況下，一

## 相關原理

種型號的電路隻對應一種格式。所謂格式，就是數據碼 $_1$ ，和 $.a$ ，的高低電平的脈寬及組成方式。一種 CPU 隻接收規定的一種格式。現在也有將多種不同格式編碼集成在一塊電路中，通過外部引腳的接線來挑選編碼格式，那麼它可以適用多種 CPU。至於萬能遙控器的編碼格式不是通過外部接線來選擇，而是通過按鍵的輸入信號來設置。現將集成電路內部

**3. 紅外線發射部分：**該部分由晶體三極管提供功率放大，以足夠的功率驅動紅外線發光二極管，發射出紅外線脈衝信號。編碼信號之所以要調製在 38kHz 的載波信號上，因為驅動紅外發射管工作的脈衝的最佳頻率在 38kHz 附近，調製後的編碼脈衝占空比降低了，這就使發射器工作的平均電流也變小了，從而降低了对電池的消耗。不按鍵時，振蕩電路不起振，此時靜態電流在微安級，所以遙控器不用設置電源開關。

# 相關原理

## 2-3 主程式板子傳輸原理

### IEEE802.15.4/ZigBee 協定概述

---

IEEE802.15.4 協定是 IEEE802.15.4 工作群組為低速率無線個人區域網(WPAN:Wireless Personal Area Network)制定的標準，該工作群組成立於 2002 年 12 月，致力於定義一種廉價的，固定、便攜或行動裝置使用的，低複雜度、低成本、低功耗、低速率的無線連線技術，並於 2003 年 12 月透過了第一個 802.15.4 標準。隨著無線感測器網路技術的發展，無線感測器網路的標準也得到了快速的發展。802.15.4 標準定義了在個人區域網中透過射頻方式在裝置間進行互連的方式與協定，該標準使用避免衝突的載波監聽多址接入方式作為媒體存取機制，同時支援星型與對等型拓撲結構。

在 802.15.4 標準中指定了兩個物理頻段和的直接擴頻序列實體層頻段:868/915MHz 和 2.4GHz 的直接序列擴頻(DSSS)實體層頻段。2.4GHz 的實體層支援空氣中 250kb/s 的速率，而 868/915MHz 的實體層支援空氣中 20kb/s 和 40kb/s 的傳輸速率。由於封包開銷和處理延遲，實際的資料吞吐量會小於規定的位元率。作為支援低速率、低功耗、短距離無線通訊的協定標準，802.15.4 在無線電頻率和資料率、資料

傳輸模型、裝置型別、網路工作方式、安全等方面都做出了說明。並且將協定模型劃分為實體層和媒體接入控制層兩個子層進行實作。

ZigBee 協定是由 ZigBee 聯盟制定的無線通訊標準，該聯盟成立於 2001 年 8 月。2002 年下半年，英國 Invensys 公司、日本三菱電力公司、美國摩托羅拉公司以及荷蘭飛利浦半導體公司共同宣布加入 ZigBee 聯盟，研發名為「ZigBee」的下一代無線通訊標準，這一事件成為該技術發展過程中的里程碑。ZigBee 聯盟現有的理事公司包括 BM Group，Ember 公司，飛思卡爾半導體，Honeywell，三菱電機，摩托羅拉，飛利浦，三星電子，西門子，及德州儀器。ZigBee 聯盟的目的是為了在全球統一標準上實作簡單可靠、價格低廉、功耗低、無線連線的監測和控制產品進行合作，並於 2004 年 12 月發行了第一個正式標準。

## 2-4 CC2530 單晶片



CC2530 是用於 2.4-GHz IEEE 802.15.4、ZigBee 和 RF4CE 應用的一個真正的片上系統（SoC）解決方案。它能夠以非常低的總的材料成本建立強大的網絡節點。CC2530 結合了領先的 RF 收發器的優良性能，業界標準的增強型 8051 CPU，系統內可編程閃存，8-KB RAM 和許多其它強大的功能。CC2530 有四種不同的閃存版本：CC2530F32/64/128/256，分別具有 32/64/128/256KB 的閃存。CC2530 具有不同的運行模式，使得它尤其適應超低功耗要求的系統。運行模式之間的轉換時間短進一步確保了低能源消耗。

CC2530F256 結合了德州儀器的業界領先的黃金單元 ZigBee 協議棧（Z-Stack™），提供了一個強大和完整的 ZigBee 解決方案。

CC2530F64 結合了德州儀器的黃金單元 RemoTI，更好地提供了一

個強大和完整的 ZigBee RF4CE 遠程控制解決方案。

## 特性-低功耗

- 主動模式 RX (CPU 空閒):

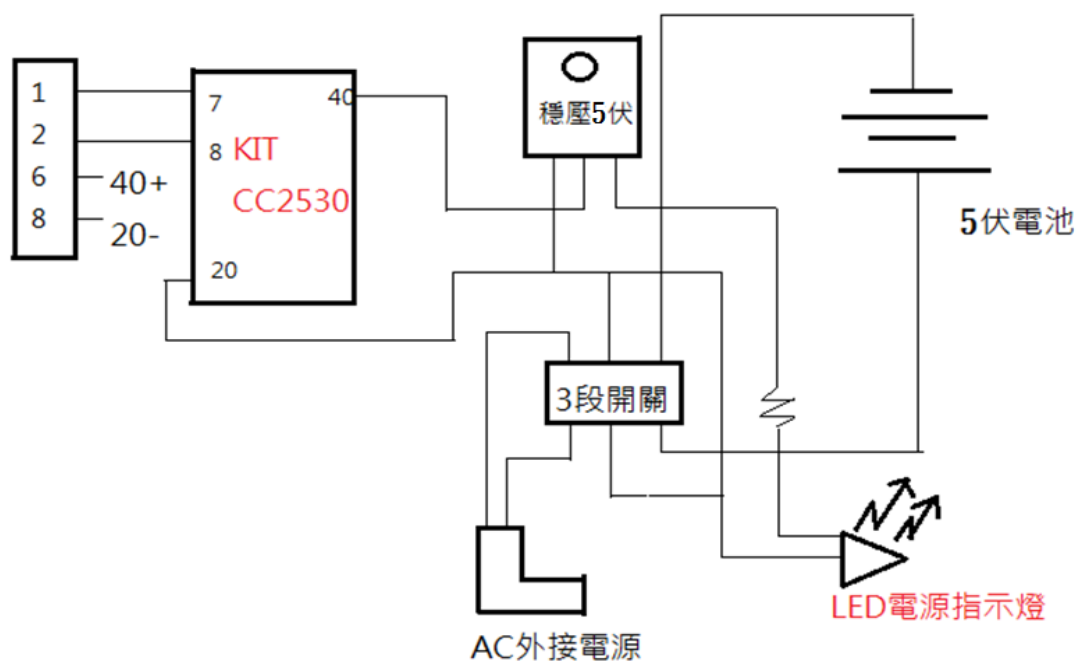
24 Ma - 主動模式 TX 在 1dBm(CPU 空閒): 29mA - 供電模式 1(4  $\mu$  s 喚醒): 0.2 mA - 供電模式 2 (睡眠定時器運行): 1  $\mu$  A - 供電模式 3 (外部中斷): 0.4  $\mu$  A - 寬電源電壓範圍 (2 V - 3.6 V)

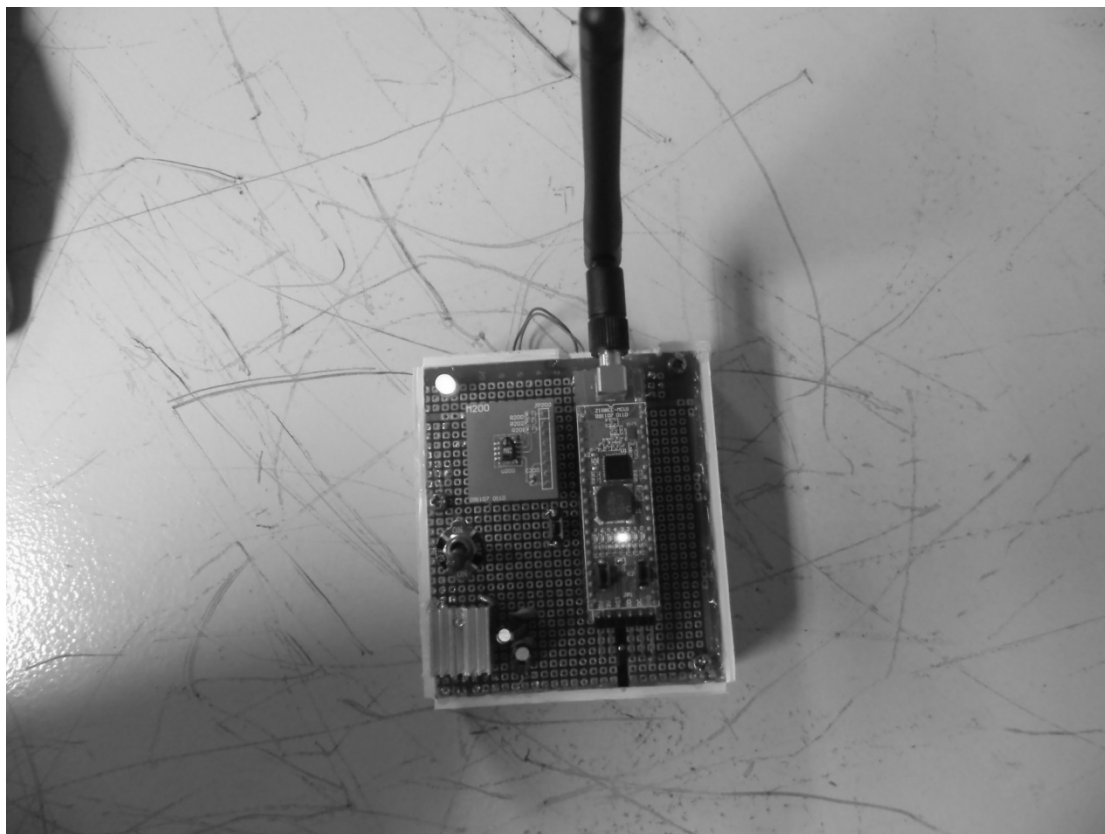
## 應用

- 2.4-GHz IEEE 802.15.4 系統
- RF4CE 遠程控制系統 (需要大於 64-KB 閃存)
- ZigBee 系統 (256-KB 閃存)
- 家庭/樓宇自動化
- 照明系統
- 工業控制和監控
- 低功耗無線傳感網絡
- 消費型電子
- 醫療保健

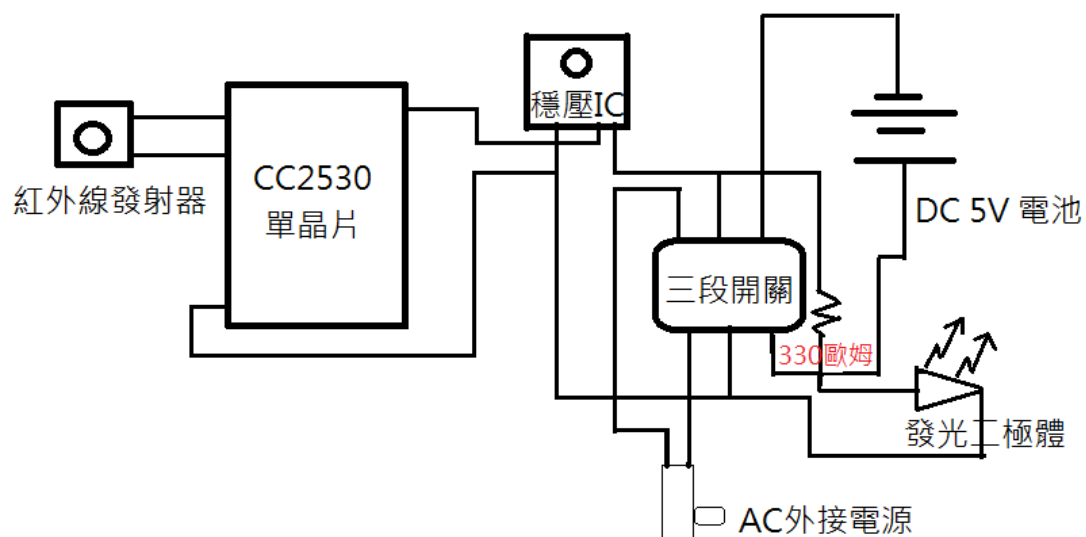
## 系統設計圖

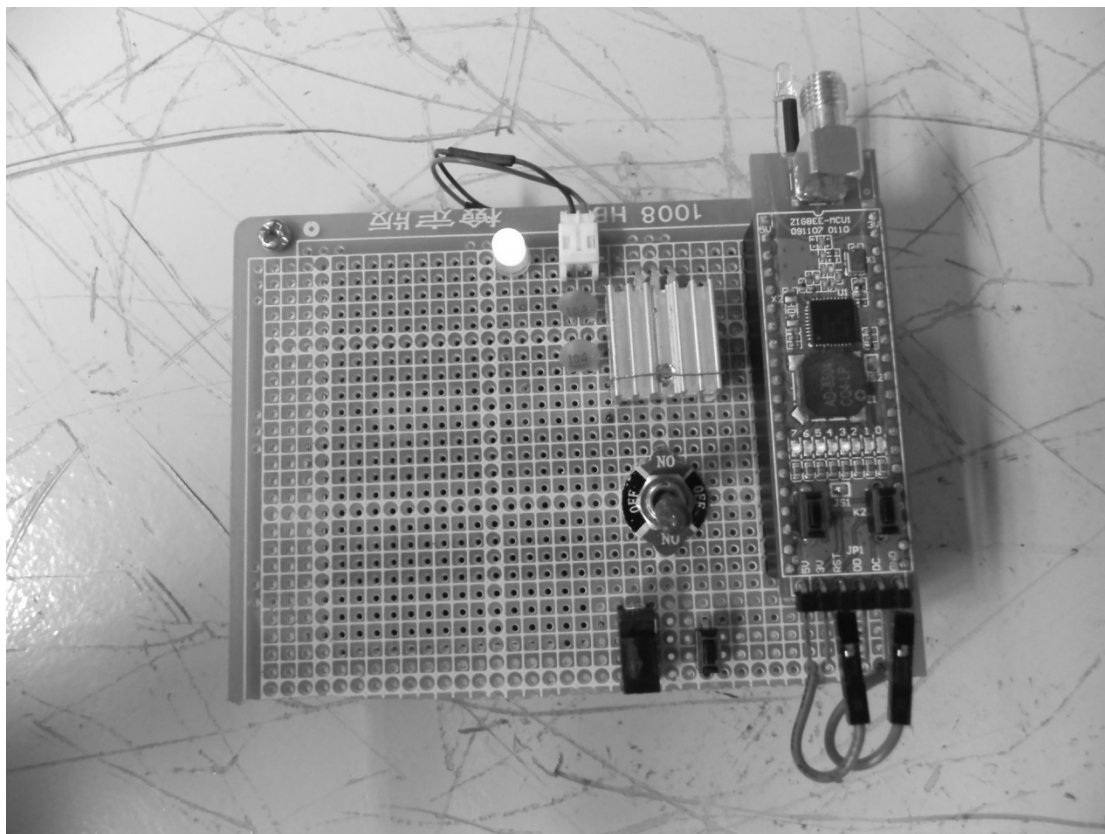
### 溫濕度感測模型電路圖



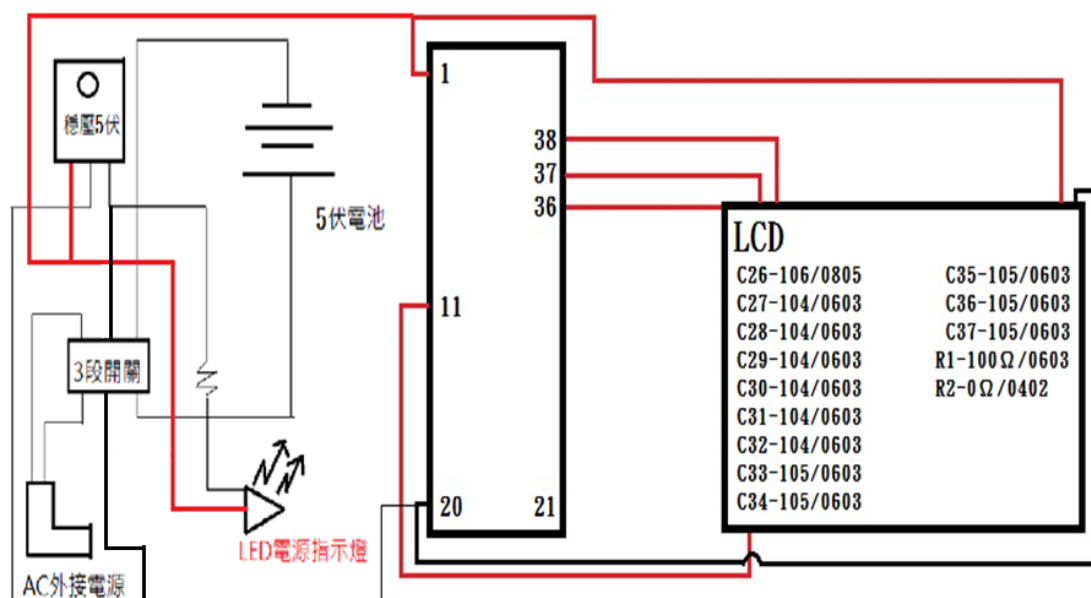


紅外線發射器模型電路圖

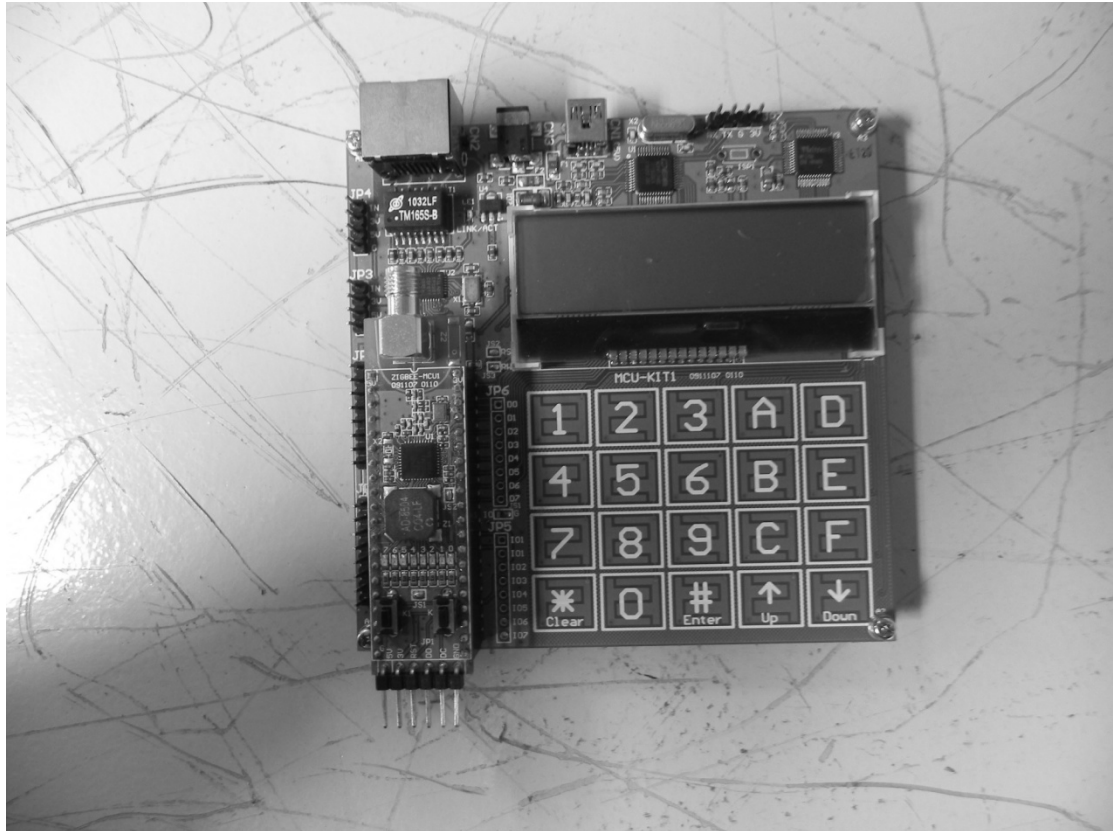




主程式模型電路圖



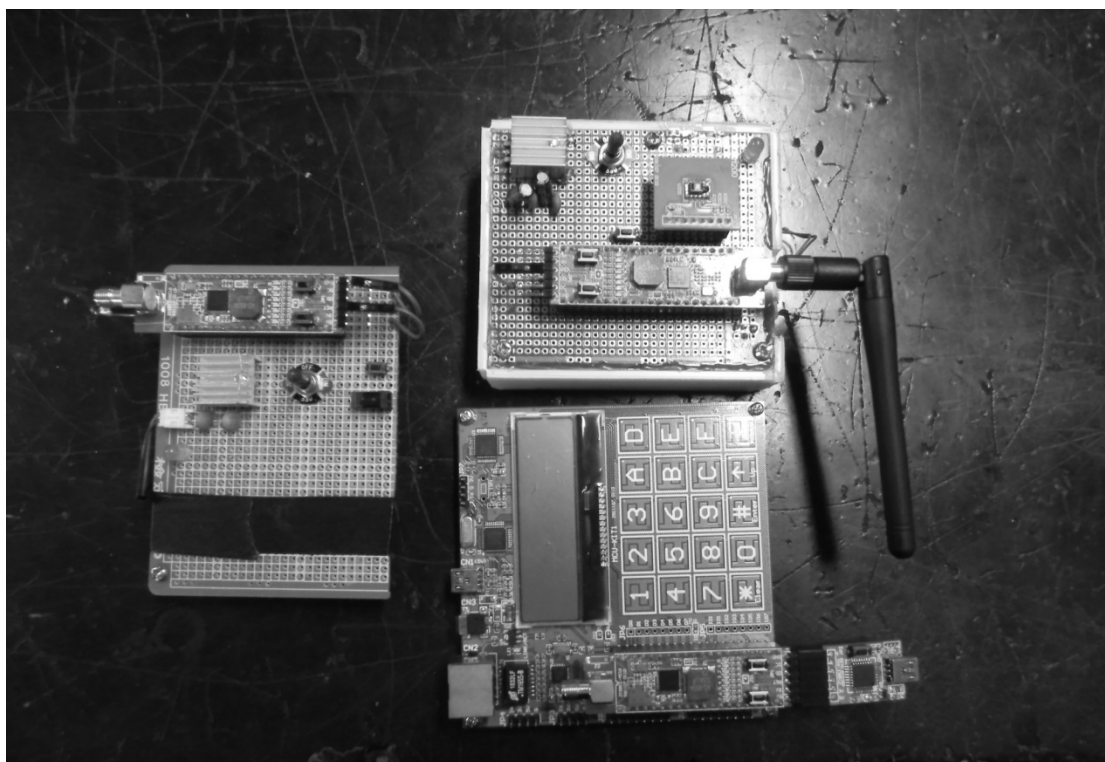




## 結果與討論

全球因為溫度極具差異性，使得冷氣機頻繁使用下造成大量的能源耗損。我們在這次專題研究的冷氣優良化自動控制，了解學生手動控制與電腦自動控制的不同，使每年冷氣機的耗能可以大幅減少，因此本專題從修評科技大學 B320 叫是隻冷氣機開始做起，使其其它教室也能夠參考其資料，使學校知道學生喜歡將較是冷氣機開到最低溫度與電腦自動控制之差異性，一方面不僅能夠節省電費，也能夠減少全球暖化現象時間上的到來。

而本專題在往後將會使紅外線發射距離拉大，並且以一對多控制，來達到最低成本，最高自動控制效率。



### 王建順學生心得：

在這次專題當中，因為對電路不太熟悉，焊接的也不是太好，所以在製作過程中有失敗過幾次，然而請教學長後解決電路問題後，感覺自己有所收穫，而這次的專題中，我想也能夠讓學校了解到，如果使用自動化冷氣機，比起學生手動開冷氣機，都調到最低溫，而以自動化，已寫好最舒適的程式，以最省電最舒適的條件下使用冷氣機，不僅可以達到省電，也可以達到舒適度。

### 江仕堯學生心得：

本次專題當中，第一次做大量的程式書寫，而且對這次使用的程式並不太熟悉，非常感謝郭學長的大力教導下，在學長的教導下，一步步的熟悉這些程式到底是哪一些的指令，有什麼樣的功能，也感謝 B217 教室的學生幫忙下，能夠完成部分不懂的紅外線波型程式，而在這次的專題中，讓我了解到了，在家靠父母，在外靠朋友的道理，也感謝陳正宗老師教導我們書面如何修整、排版等問題。

## 參考資料

紅外線感測器 (9~11 頁)

<http://zh.wikipedia.org/wiki/%E7%BA%A2%E5%A4%96%E7%BA%BF>

溫度感測器 (5 頁)

<http://zh.wikipedia.org/wiki/%E6%B8%A9%E5%BA%A6%E4%BC%A0%E6%84%9F%E5%99%A8>

濕度感測器 (9~11 頁)

<http://big5.china.com/gate/big5/sinometer.blog.china.com/201305/11328386.html>

紅外線發射器 (6~8 頁)

<http://www.dzsc.com/data/html/2009-2-11/75877.html>

溫度感測器圖片 (13 頁)

[https://www.google.com.tw/search?espv=210&es\\_sm=122&tbm=isch&sa=1&q=%E6%BA%AB%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8&oq=%E6%BA%AB%E5%BA%A6&gs\\_l=img.1.2.0l10.29234.32822.0.36301.3.3.0.0.0.156.250.2j1.3.0...0...1c.1j4.32.img..0.3.250.zEbTXFsXslg](https://www.google.com.tw/search?espv=210&es_sm=122&tbm=isch&sa=1&q=%E6%BA%AB%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8&oq=%E6%BA%AB%E5%BA%A6&gs_l=img.1.2.0l10.29234.32822.0.36301.3.3.0.0.0.156.250.2j1.3.0...0...1c.1j4.32.img..0.3.250.zEbTXFsXslg)

### 溼度感測器圖片 (13 頁)

[https://www.google.com.tw/search?espv=210&es\\_sm=122&tbm=isch&sa=1&q=%E6%BF%95%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8&oq=%E6%BF%95%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8&gs\\_l=img.1.0.0l2j0i24l4j0i5i2l2.32740.36796.0.38855.8.6.2.0.0.0.47.188.6.6.0....0...1c.1j4.32.img..7.1.47.Mu1bwHGwsBM](https://www.google.com.tw/search?espv=210&es_sm=122&tbm=isch&sa=1&q=%E6%BF%95%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8&oq=%E6%BF%95%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8&gs_l=img.1.0.0l2j0i24l4j0i5i2l2.32740.36796.0.38855.8.6.2.0.0.0.47.188.6.6.0....0...1c.1j4.32.img..7.1.47.Mu1bwHGwsBM)

### 紅外線發射器圖片 (14 頁)

[https://www.google.com.tw/search?q=%E7%B4%85%E5%A4%96%E7%B7%9A%E7%99%BC%E5%B0%84%E5%99%A8%E5%8E%9F%E7%90%86&espv=210&es\\_sm=122&tbm=isch&tbo=u&source=univ&sa=X&ei=9VbOUtWFEczRkwWx64HoAQ&ved=0CF4QsAQ](https://www.google.com.tw/search?q=%E7%B4%85%E5%A4%96%E7%B7%9A%E7%99%BC%E5%B0%84%E5%99%A8%E5%8E%9F%E7%90%86&espv=210&es_sm=122&tbm=isch&tbo=u&source=univ&sa=X&ei=9VbOUtWFEczRkwWx64HoAQ&ved=0CF4QsAQ)

## 紅外線發射程式碼

```
void Cooling(void);
void conditioning(void);
void abc(void);
void Ventilation(void);
void Desiccant(void);

static uint8 pRxData[127];
uint8 buf[127];
uint8 len;

//-----
-----
void main(void)
{
    uint8 v;
    halBoardInit();
    Setbasic(25, 0x2007, 0xCC01);
    halMcuWaitMs(300);
    while (TRUE)
    {
        M330_Init();

        while (TRUE)
        {

            v = halButtonPushed();

            if (v == HAL_BUTTON_1)//冷氣
            {
                M330_TX(130);
                M330_NTX(127);

                M330_SendCodeAircon(0xB2);
                M330_SendCodeAircon(0xAE);
                M330_SendCodeAircon(0x4D);
```

```
M330_SendCodeAircon(0x51);  
M330_SendCodeAircon(0xF0);  
M330_SendCodeAircon(0x64);  
M330_SendCodeAircon(0xE4);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x10);  
M330_SendCodeAircon(0x98);
```

```
M330_TX(16);  
/* M330_TX(130);  
M330_NTX(127);
```

```
M330_SendCodeAircon(0xB2);  
M330_SendCodeAircon(0xAE);  
M330_SendCodeAircon(0x4D);  
M330_SendCodeAircon(0x51);  
M330_SendCodeAircon(0xF0);  
M330_SendCodeAircon(0x68);  
M330_SendCodeAircon(0xD4);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x00);  
M330_SendCodeAircon(0x28);
```

```
M330_TX(16);*/
```

```
/*M330_TX(130);//送風  
M330_NTX(127);
```

```
M330_SendCodeAircon(0xB2);  
M330_SendCodeAircon(0xAE);  
M330_SendCodeAircon(0x4D);  
M330_SendCodeAircon(0x51);
```



```

        M330_SendCodeAircon(0xF0);
        M330_SendCodeAircon(0x42);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x60);

        M330_TX(16);*/
    }
    if (v == HAL_BUTTON_2)//除濕
    {
        M330_TX(130);
        M330_NTX(127);

        M330_SendCodeAircon(0xB2);
        M330_SendCodeAircon(0xAE);
        M330_SendCodeAircon(0x4D);
        M330_SendCodeAircon(0x51);
        M330_SendCodeAircon(0xF0);
        M330_SendCodeAircon(0x28);
        M330_SendCodeAircon(0x84);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x10);

        M330_TX(16);
    }

    //

    if(basicRfPacketIsReady())
    {

```

```

len = basicRfReceive(pRxData , 127 , NULL);
if(pRxData[0] == 1 )
{
    halMcuWaitMs(300);
    if(pRxData[1] == 1)
    {
        Cooling();
    }
    if(pRxData[1] == 2)
    {
        Ventilation();
    }
    if(pRxData[1] == 3)
    {
        Desiccant();
    }

}

if(pRxData[0] == 0 )
{
    abc();
    halMcuWaitMs(300);
}

}

halMcuWaitMs(10);
}

}

}

//-----on

void conditioning(void)
{
    M330_TX(130);

```

```

M330_NTX(127);

M330_SendCodeAircon(0xB2);
M330_SendCodeAircon(0xAE);
M330_SendCodeAircon(0x4D);
M330_SendCodeAircon(0x51);
M330_SendCodeAircon(0xF0);
M330_SendCodeAircon(0x64);
M330_SendCodeAircon(0xE4);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x10);
M330_SendCodeAircon(0x98);

M330_TX(16);
}
//-----off

void abc(void)
{
    M330_TX(130);
    M330_NTX(127);

    M330_SendCodeAircon(0xB2);
    M330_SendCodeAircon(0xAE);
    M330_SendCodeAircon(0x4D);
    M330_SendCodeAircon(0x51);
    M330_SendCodeAircon(0xF0);
    M330_SendCodeAircon(0x64);
    M330_SendCodeAircon(0xE4);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x10);
    M330_SendCodeAircon(0x98);

```

```

        M330_TX(16);
    }
    //----->>|

void Cooling(void)
{
    M330_TX(130);
    M330_NTX(127);

    M330_SendCodeAircon(0xB2);
    M330_SendCodeAircon(0xAE);
        M330_SendCodeAircon(0x4D);
        M330_SendCodeAircon(0x51);
        M330_SendCodeAircon(0xF0);
        M330_SendCodeAircon(0x68);
        M330_SendCodeAircon(0xD4);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x28);

        M330_TX(16);
    }
    //-----|<<

void Ventilation(void)
{
    /* M330_TX(130); //送風
        M330_NTX(127);

        M330_SendCodeAircon(0xB2);
        M330_SendCodeAircon(0xAE);
        M330_SendCodeAircon(0x4D);
        M330_SendCodeAircon(0x51);
        M330_SendCodeAircon(0xF0);

```

```

        M330_SendCodeAircon(0x42);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x60);

        M330_TX(16);*/
    M330_TX(130);
    M330_NTX(127);

    M330_SendCodeAircon(0xB2);
    M330_SendCodeAircon(0xAE);
        M330_SendCodeAircon(0x4D);
        M330_SendCodeAircon(0x51);
        M330_SendCodeAircon(0xF0);
        M330_SendCodeAircon(0x68);
        M330_SendCodeAircon(0xD4);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x00);
        M330_SendCodeAircon(0x28);

        M330_TX(16);
    }
void Desiccant(void)
{
    /*M330_TX(130);
    M330_NTX(127);

    M330_SendCodeAircon(0xB2);
    M330_SendCodeAircon(0xAE);
    M330_SendCodeAircon(0x4D);
    M330_SendCodeAircon(0x51);

```

```

M330_SendCodeAircon(0xF0);
M330_SendCodeAircon(0x28);
M330_SendCodeAircon(0x84);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x00);
M330_SendCodeAircon(0x10);

M330_TX(16);*/
    M330_TX(130);
    M330_NTX(127);

M330_SendCodeAircon(0xB2);
M330_SendCodeAircon(0xAE);
    M330_SendCodeAircon(0x4D);
    M330_SendCodeAircon(0x51);
    M330_SendCodeAircon(0xF0);
    M330_SendCodeAircon(0x68);
    M330_SendCodeAircon(0xD4);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x00);
    M330_SendCodeAircon(0x28);

    M330_TX(16);
}
.

```

## 主控程式碼

```
//-----  
-----  
// Filename: ex.c  
// Description: 主控板  
//-----  
-----  
//-----  
-----  
// INCLUDES  
//-----  
-----  
#include "hal_defs.h"  
#include "hal_cc8051.h"  
#include "hal_int.h"  
#include "hal_mcu.h"  
#include "hal_board.h"  
#include "hal_lcd.h"  
#include "hal_keypad.h"  
#include "hal_uart.h"  
#include "hal_buzzer.h"  
#include "hal_button.h"  
#include "hal_led.h"  
#include "hal_rf.h"  
#include "util.h"  
#include "util_lcd.h"  
#include "basic_rf.h"  
#include "M200.h"  
  
//-----  
-----  
// CONSTANTS  
//-----  
-----  
// Application parameters  
#define RF_CHANNEL 25 // 2.4 GHz RF  
channel
```

```

// BasicRF address definitions
#define PAN_ID                0x2007
#define SWITCH_ADDR           0x2520
#define LIGHT_ADDR            0xBEEF
#define APP_PAYLOAD_LENGTH    127
#define LIGHT1_TOGGLE_CMD     '1'
#define LIGHT2_TOGGLE_CMD     '2'

// Application states
#define IDLE                   0
#define SEND_CMD               1

//-----
// LOCAL VARIABLES
//-----

static uint8 pTxData[APP_PAYLOAD_LENGTH];
static uint8 pRxData[APP_PAYLOAD_LENGTH];
static basicRfCfg_t basicRfConfig;
uint8 a;
uint16 temp;
uint16 humi;
uint16 temp_ch;
uint16 humi_ch;
uint16 work_ch;
uint16 work_a;
uint16 work_b;
uint8 OPinit;
void main(void)
{
    // Initialise board peripherals
    halBoardInit();
    //旗標
    temp_ch = 0; humi_ch = 0; OPinit = 0;
    //ZigBee Ini
    basicRfConfig.panId = PAN_ID;

```



```

basicRfConfig.channel = RF_CHANNEL;
basicRfConfig.ackRequest = TRUE;
basicRfConfig.myAddr = 0XC001;
if (basicRfInit(&basicRfConfig) == FAILED){}
basicRfReceiveOn();

// Indicate that device is powered
halLedSet(4);
// Print Logo and splash screen on LCD
utilPrintLogo("** M200 Test  **");
//halMcuWaitMs(300);
//halBuzzer(300);
//while (halKeypadPushed() == 0)
//    ;
utilMenuSelect(NULL);
basicRfSendPacket(0xCC01, "AA", 2);
halMcuWaitMs(500);
basicRfSendPacket(0xCC01, "AA", 2);
halMcuWaitMs(500);
while (TRUE)
{
    halLcdClear();
    //halBuzzer(300);
    halLcdWriteLine(HAL_LCD_LINE_1, "T=      C    H=
%" );
    halLcdWriteChar(HAL_LCD_LINE_1, 6, 0XDF);
    halLcdWriteChar(HAL_LCD_LINE_1, 9, 0X2F);
    halLcdWriteLine(HAL_LCD_LINE_2, "Conditioned OFF");
    M200_Init();
    work_a = work_b = 0 ;
    work_ch = 1;
    temp = 0;
    while (TRUE)
    {
        if(basicRfPacketIsReady())
        {
            basicRfReceive(pRxData , 127 , NULL);
            halLcdWriteChar(HAL_LCD_LINE_1, 2,

```

```

pRxData[0]);
    halLcdWriteChar(HAL_LCD_LINE_1, 3,
pRxData[1]);
    halLcdWriteChar(HAL_LCD_LINE_1, 4,
pRxData[2]);
    halLcdWriteChar(HAL_LCD_LINE_1, 5,
pRxData[3]);

    halLcdWriteChar(HAL_LCD_LINE_1, 13,
pRxData[4]);
    halLcdWriteChar(HAL_LCD_LINE_1, 14,
pRxData[5]);

    temp = ((pRxData[0]- '0') *100) +
    ((pRxData[1]- '0') * 10) + (pRxData[3]- '0');
    humi = ((pRxData[4]- '0') *100) +
    ((pRxData[5]- '0') * 10);
    if(OPinit == 0)
    {
        OPinit+=1;

        pTxData[0]=0;
        pTxData[1]=5;
        halLcdWriteLine(HAL_LCD_LINE_2, " Air
Cooling  ON");

        basicRfSendPacket(0xCC01, pTxData, 8);
        halMcuWaitMs(1000);
    }
}

//M200_GetValue();
//humi = (uint16)((((float)M200_Humi) *0.0405) -
(((float)M200_Humi)*((float)M200_Humi) *0.0000028) - 4.0)
*10.0);
//temp = (uint16)((((float)M200_Temp) *0.01) -
40.0) *10.0);
//halLcdWriteChar(HAL_LCD_LINE_1, 2, (temp / 100)
+ '0');

```

```

if(OPinit == 1)
if (temp > 100)
{
    if (temp > 270)
    {
        work_a = 'X' ;
    }
    else if (temp <= 270 && temp > 230)
    {
        work_a = 'Y' ;
    }
    else if (temp <= 230)
    {
        work_a = 'Z' ;
    }

    if ( work_a != work_b)
    {

        if( work_b == 'Z' )
        {
            if(temp_ch == 2 && temp >=250)
            {
                pTxData[0]=0;
                pTxData[1]=5;
                halLcdWriteLine(HAL_LCD_LINE_2,
"Air Cooling  ON");

basicRfSendPacket(0xCC01,pTxData, 8);
                halMcuWaitMs(1000);
                work_ch = 0;
            }
        }
        else if(work_b != 'Z' )
            work_ch = 0;
    }
}

```

```

if (humi > 650 && humi_ch == 0 && work_ch == 0)
{
    if(temp >270 && temp_ch == 0)
    {
        pTxData[0]=1;
        pTxData[1]=1;
        halLcdWriteLine(HAL_LCD_LINE_2, " Air
Cooling  ON");

        basicRfSendPacket(0xCC01, pTxData, 8);

        temp_ch = 1;
        work_b = work_ch = 'X' ;
    }
    if (temp <= 270 && temp >230 && temp_ch ==0)
    {
        pTxData[0]=1;
        pTxData[1]=3;
        halLcdWriteLine(HAL_LCD_LINE_2,
"Dehumidify ON");

        basicRfSendPacket(0xCC01, pTxData, 2);
        work_b = work_ch = 'Y' ;
    }
    if (temp <= 230 && temp_ch ==0 && temp > 160)
    {
        pTxData[0]=0;
        pTxData[1]=3;
        halLcdWriteLine(HAL_LCD_LINE_2,
"Close");

        basicRfSendPacket(0xCC01, pTxData, 2);
        temp_ch = 2;
        work_b =work_ch = 'Z' ;
    }
    if (temp_ch == 1 && temp <250)
    {
        pTxData[0]=1;
        pTxData[1]=3;
        halLcdWriteLine(HAL_LCD_LINE_2,
"Dehumidify ON");

```

```

        basicRfSendPacket(0xCC01, pTxData, 2);
        temp_ch = 0;
        work_b = work_ch = 'Y' ;
    }
    if (temp_ch == 2 && temp >=250)
    {
        pTxData[0]=1;
        pTxData[1]=3;
        halLcdWriteLine(HAL_LCD_LINE_2,
"Dehumidify ON");
        basicRfSendPacket(0xCC01, pTxData, 2);
        temp_ch = 0;
        work_b = work_ch= 'Y' ;
    }
}
if (humi <= 650 && humi_ch == 0 && work_ch == 0)
{
    if(temp >270 && temp_ch ==0 )
    {
        pTxData[0]=1;
        pTxData[1]=1;
        halLcdWriteLine(HAL_LCD_LINE_2, "Air
Cooling  ON");
        basicRfSendPacket(0xCC01, pTxData, 2);
        temp_ch = 1;
        work_b = work_ch = 'X' ;
    }

    if(temp <=270 && temp >230 && temp_ch ==0)
    {
        pTxData[0]=1;
        pTxData[1]=2;
        halLcdWriteLine(HAL_LCD_LINE_2,
"Ventilation ON");
        basicRfSendPacket(0xCC01, pTxData, 2);
        work_b = work_ch = 'Y' ;
    }
}

```

```

160 )
        if (temp <= 230  && temp_ch ==0 && temp >
        {
            pTxData[0]=0;
            pTxData[1]=3;
            halLcdWriteLine(HAL_LCD_LINE_2,
"Close");

            basicRfSendPacket(0xCC01, pTxData, 2);
            temp_ch = 2;
            work_b = work_ch = 'Z' ;
        }
        if (temp_ch == 1 && temp <250)
        {
            pTxData[0]=1;
            pTxData[1]=2;
            halLcdWriteLine(HAL_LCD_LINE_2,
"Ventilation ON");

            basicRfSendPacket(0xCC01, pTxData, 2);
            temp_ch = 0;
            work_b = work_ch = 'Y' ;
        }
        if (temp_ch == 2 && temp >=250)
        {
            pTxData[0]=1;
            pTxData[1]=2;
            halLcdWriteLine(HAL_LCD_LINE_2,
"Ventilation ON");

            basicRfSendPacket(0xCC01, pTxData, 2);
            temp_ch = 0;
            work_b =work_ch = 'Y' ;
        }
    }
}
    }
}
}

```

## 溫濕度感測程式碼

```
//-----  
-----  
// Filename: ex.c  
// Description: 溫濕度偵測  
//-----  
-----  
//-----  
-----  
// INCLUDES  
//-----  
-----  
#include "hal_defs.h"  
#include "hal_cc8051.h"  
#include "hal_int.h"  
#include "hal_mcu.h"  
#include "hal_board.h"  
#include "hal_lcd.h"  
#include "hal_keypad.h"  
#include "hal_uart.h"  
#include "hal_buzzer.h"  
#include "hal_led.h"  
#include "hal_rf.h"  
#include "util.h"  
#include "util_lcd.h"  
#include "basic_rf.h"  
#include "M200.h"  
  
//-----  
-----  
// CONSTANTS  
//-----  
-----  
// Application parameters  
#define RF_CHANNEL                25        // 2.4 GHz RF  
channel
```

```

// BasicRF address definitions
#define PAN_ID                0x2007
#define SWITCH_ADDR           0x2520
#define LIGHT_ADDR            0xBEEF
#define APP_PAYLOAD_LENGTH    127
#define LIGHT1_TOGGLE_CMD     '1'
#define LIGHT2_TOGGLE_CMD     '2'

// Application states
#define IDLE                   0
#define SEND_CMD               1

//-----
//-----
// LOCAL VARIABLES
//-----
//-----

static uint8 pTxData[APP_PAYLOAD_LENGTH];
static uint8 pRxData[APP_PAYLOAD_LENGTH];
uint8 len;
static basicRfCfg_t basicRfConfig;
uint8 a;
uint16 temp;
uint16 humi;
void main(void)
{
    // Initialise board peripherals
    halBoardInit();

    //ZigBee Ini
    basicRfConfig.panId = PAN_ID;
    basicRfConfig.channel = RF_CHANNEL;
    basicRfConfig.ackRequest = TRUE;
    basicRfConfig.myAddr = 0xB001;
    if (basicRfInit(&basicRfConfig) == FAILED){}
    basicRfReceiveOn();

    // Indicate that device is powered

```



```

halLedSet(4);
// Print Logo and splash screen on LCD
utilPrintLogo("** M200 Test  **");
halMcuWaitMs(300);
//halBuzzer(300);
//while (halKeypadPushed() == 0)
//    ;
utilMenuSelect(NULL);
while (TRUE)
{
    halLcdClear();
    //halBuzzer(300);
    halLcdWriteLine(HAL_LCD_LINE_1, "T=      C    H=
%");
    halLcdWriteChar(HAL_LCD_LINE_1, 6, 0XDF);
    halLcdWriteChar(HAL_LCD_LINE_1, 9, 0X2F);

    M200_Init();
    temp = 0;
    while (TRUE)
    {

        M200_GetValue();
        humi = (uint16)((((float)M200_Humi) *0.0405) -
(((float)M200_Humi)*((float)M200_Humi) *0.0000028) - 4.0)
*10.0);
        temp = (uint16)((((float)M200_Temp) *0.01) -
40.0) *10.0);
        halLcdWriteChar(HAL_LCD_LINE_1, 2, (temp / 100) +
'0');
        halLcdWriteChar(HAL_LCD_LINE_1, 3, ((temp / 10) %
10) + '0');
        halLcdWriteChar(HAL_LCD_LINE_1, 4, '.' );
        halLcdWriteChar(HAL_LCD_LINE_1, 5, (temp % 10) +
'0');

        halLcdWriteChar(HAL_LCD_LINE_1, 13, (humi / 100) +

```

```

'0' );
        halLcdWriteChar(HAL_LCD_LINE_1, 14, ((humi / 10) %
10) + '0' );
        pTxData[0] = (temp / 100) + '0';
        pTxData[1] = (temp / 10) % 10 + '0';
        pTxData[2] = '.';
        pTxData[3] = (temp % 10) + '0';
        pTxData[4] = (humi / 100) + '0';
        pTxData[5] = ((humi / 10) % 10) + '0';
        basicRfSendPacket( 0xc001, pTxData, 6 );
    }
}
}

```

# 作者簡介

指導老師：

陳正宗 老師

專題學生：

班級：四技電機四甲      學號：BD99507      王建順 學生

班級：四技電機四甲      學號：BD99035      江仕堯 學生