

中華大學資訊工程學系

智慧型教室教學輔助系統之研製(二)

專題成果報告

The Development of Intelligent Classroom Teaching System

專題編號:

執行期間:100 年 02 月 至 101 年 06 月

指導老師:曾秋蓉 教授

小組成員:

資訊工程四年甲班 B09702004 王儷蓉

資訊工程四年甲班 B09702064 林淑萍

資訊工程四年甲班 B09702133 彭彥彰

目 錄

第一章 簡介

第二章 研究背景

第三章 研究方法

第四章 遭遇問題與解決方案

第五章 研究成果

第六章 結論與未來研究方向

第一章 簡介

本專題的目的在於打造一個舒適的教學環境，讓老師能在一個舒適的環境下授課，學生則是在舒適的環境下接受更好的教育。因此，我們延續去年學長的研究，將他們所設計的智慧型教學輔助系統功能做進一步的擴充，包含以下六個部分：

[1] 教學環境管理系統：利用感測器偵測教室的環境因素，當某些環境因素數值超出理想值許多，則後端系統會給予應有的回饋，使教室環境可以保持一個最佳的狀態。延續學長之前的研究，本專題擴充紅外線分離式冷氣和排風機。

[2] 教學模式管理系統：為了節省上課時，啟動一些器材的時間，我們假設一些教學環境，來撰寫程式，由電腦控制，希望可以利用處控式按鈕來控制某教學環境所需要的所有器材。延續學長之前的研究，本專題擴充投影布幕的控制。

[3] 學習狀態管理系統：使用影像辨識模組便是學生是否於課堂上打瞌睡，或者學生在上課時不專心，東張西望等，若偵測到有學生上課不專心的情況，則有課桌椅上的 LED 打出強光，提醒學生專心上課。本專題將原本的 8 位學生座位擴充為 24 位學生座位。

[4] 遠端監控系統：讓家長或老師可以知道學生上課情形如何。

[5] 系統整合：將所有系統功能集資料庫與人臉辨是自動點名系統整合。

[6] 資料庫完整建置：供系統整合一即實驗收集數據分析用。

第二章 研究背景

2.1 專題開發之目的

隨著科技發展日新月異，無線網路與感測器等裝置的運用已漸漸在日常生活普及化，因此，本專題針對之前的相關研究，把智慧型家庭成熟的 ZigBee 無線感測網路拿來運用在偵測教室環境的參數和學生學習行為上，系統更進一步計算分析這些環境參數與行為參數，並給予老師與學生適當的回饋。

2.2 專題內容

延續去年學長姐的研究，我們在教學環境管理系統、教學模式管理系統、學習狀態管理系統中增加幾個功能，並多增加遠端監控系統、系統整合、資料庫完整建置等三樣功能。

[1] 教學環境管理系統：利用感測器偵測教室的環境因素，當某些環境因素數值超出理想值許多，則後端系統會給予應有的回饋，使教室環境可以保持一個最佳的狀態，擴充紅外線遙控分離式冷氣和排風扇的控制。

在環境方面，系統對於各個感應器所做的回饋為：

感測器	條件	回饋裝置
照度感測器	500~1000 亮度	電燈、窗簾
溫度感測器	溫度 > TS+3 < TS > TS+5 < TS+4 > TS+7 < TS+6 (TS預設25度)	空調開 空調關 分離式冷氣開(溫度降三度) 分離式冷氣關 電扇開 電扇關
濕度感測器	濕度 > 80 RH%	分離式冷氣開
CO2感測器	大於850 PPM	14吋抽風機啟動

[2] 教學模式管理系統：為了節省上課時，啟動一些器材的時間，我們想假設一些情境，再撰寫程式後由電腦控制，希望可以利用處控式按鈕來控制某一情境所需要的所有器材。擴充投影布幕的控制。

在教學控制系統中，分別將情境分為以下四種：

1. 黑板教學模式
2. 影音教學模式
3. 投影片教學模式
4. 下課模式



在教學方面，系統對於各個感應器所做的回饋為：

教學模式	需求	條件	回饋方式
黑板教學模式	照度高		布幕上 窗簾前、後開
		照度太低 < 500	電燈1、2、3、4排開
		照度太高 > 500 & 照度 < 1000	電燈1、2排開，第3、4排關
		照度 > 1000	電燈1、2、3、4排關
投影片教學模式	照度前低後高		布幕下 窗簾前關、後開 電燈1、2排關3、4排開
影音教學模式	照度低		布幕下 窗簾前、後關 電燈1、2、3、4排關
下課模式	照度低		布幕上 窗簾前開、後開 電燈1、2、3、4排關

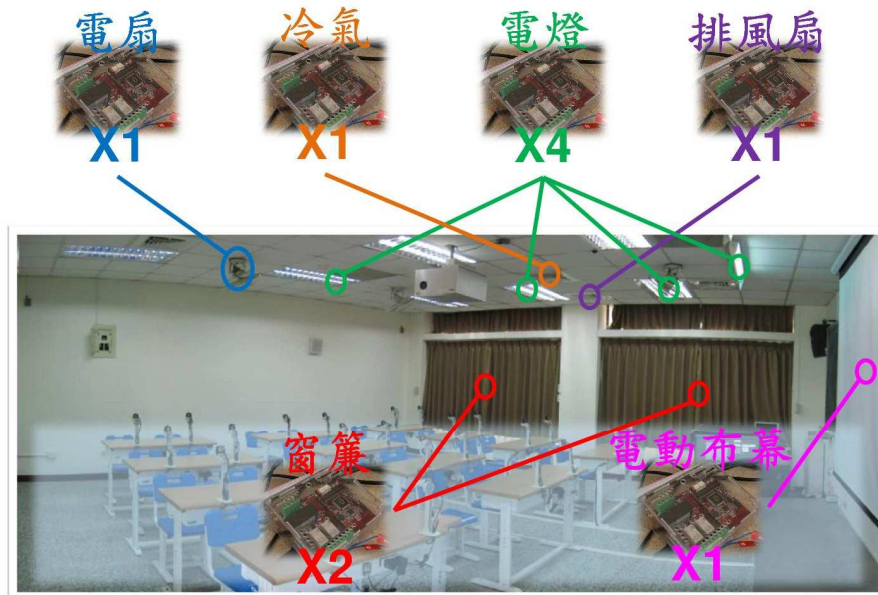
[3] 學習狀態管理系統：使用影像辨識模組辨識學生是否於課堂上打瞌睡，或者學生在上課時不專心，東張西望等，若偵測到有學生上課不專心的情況，則有課桌椅上的 LED 打出強光，提醒學生專心上課。本專題將原本的 8 位學生座位擴充為 24 位學生座位。

[4] 遠端監控系統：利用網頁可隨時知道教室環境狀態、查詢學生上課最終狀態、老師可利用網頁補點名。

[5] 系統整合：將教學環境管理系統、教學模式管理系統、學生學習狀態系統、人臉自動辨識點名整合，讓使用者使用起來更為方便。

[6] 資料庫完整建置：將資料紀錄到資料庫，供以後實驗紀錄用。

2.3 智慧型教室配置圖



第三章 研究方法

3.1 開發原理

我們延續學長的研究，擴充了布幕、排風機、CO2 感測、紅外線冷氣，將此運用到程式上，讓教學品質變得更好，最後再將教室環境/模式管理系統、學習狀態系統和人臉自動點名系統整合起來，並把資料記錄到資料庫。

布幕和排風機延續學長的寫法續寫，紅外線冷氣和 CO2 偵測則是對 ZIGBEE 裝置進行接收與發送訊號。

寫法如下：

3.1.1 CO2 接收端

```
private void serialPort2_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    byte[] tmp1 = new byte[41];
    try
    {
        //回傳封包格式
        // [HEX] (41) <- 7E 45 00 FF FF FF FF 1C 00 93 00 00 02 00 00 01 00 41 0C E0 0C A9 0C A6 0C AD 0C E0 0C A6 0C A6 0C A3 0C E0 0C A9 DC F6 7E
        // 頭          GroupID          CO2感測值          end
        CO2 = "";
        for (int i = 0; i < tmp1.Length; i++)
        {
            tmp1[i] = Convert.ToByte(serialPort2.ReadByte());
            if (i == 19 || i == 20)
            {
                CO2 = CO2.Insert(0, Convert.ToString(tmp1[i].ToString("X2")));
            }
        }
        if (tmp1[20] == 01 || tmp1[20] == 02 || tmp1[20] == 03)
        {
            this.BeginInvoke(new EventHandler(delegate
            {
                SetTheText2(tmp1);
            }));
            Application.DoEvents();
        }
    }
    catch (System.Exception)
    {
        if (serialPort2.IsOpen == true)
        {
            serialPort2.DiscardInBuffer();
            tmp1 = null;
        }
    }
}
```


3.1.2 CO2 接收端處理

```
//CO2接收端處理
private void SetTheText2(byte[] tmp1)//string strText)
{
    try
    {
        CO2_P = Convert.ToInt32(CO2, 16).ToString(); //16進制轉10進制

        if (tmp1[20] == 01)
        {
            label73.Text = CO2_P;
        }
        else
        {
            label73.Text = CO2_P;
        }
        //CO2濃度 > 850 ppm -> 排風扇開
        if (Int32.Parse(CO2_P) > 850)
        {
            cmd = "7E 42 FF FF 17 27 0D 49 0A 01 02 CC CC CC CC CC CC CC CC CC 00 00 7E"; //開
            this.BeginInvoke(new EventHandler(delegate
            {
                SendControl(cmd);
            }));
            Application.DoEvents();
            label29.Text = "ON";
            label29.ForeColor = Color.LawnGreen;
        }
        //CO2濃度 < 850 ppm -> 排風扇閉
        else
        {
            cmd = "7E 42 FF FF 17 27 0D 49 0A 01 01 CC CC CC CC CC CC CC CC CC 00 00 7E"; //閉
            this.BeginInvoke(new EventHandler(delegate
            {
                SendControl(cmd);
            }));

            Application.DoEvents();
            label29.Text = "OFF";
            label29.ForeColor = Color.Red;
        }
    }
    catch (System.Exception)
    {
    }
}
}
```

3.1.3 冷氣發送封包

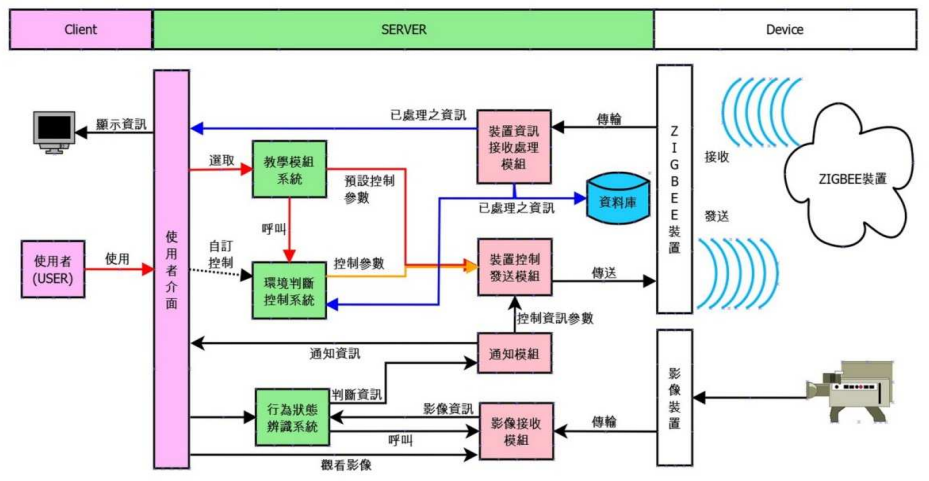
```
//冷氣封包格式
private void SendCmd(byte node, byte cmd1, byte cmd2)
{
    byte[] buffer = new byte[3];
    buffer[0] = node;
    buffer[1] = cmd1;
    buffer[2] = cmd2;
    serialPort3.Write(buffer, 0, 3);
}
```

3.1.4 冷氣發送封包 做何種功能

```
//冷氣溫度(上)
private void button41_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x02);
}
//冷氣溫度(下)
private void button42_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x03);
}
//冷氣機能
private void button43_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x04);
}
//冷氣風速
private void button44_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x06);
}
//冷氣PUZZY
private void button45_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x08);
}
```

3.2 教學環境管理系統和教學模式管理系統

3.2.1 系統架構圖



Client :

1. 使用者啟動程式。
2. 將Server 端傳回的資訊顯示於介面。

Server :

1. 使用者選取教學模組系統後，會將預設控制參數傳送到裝置控制發送系統，接著發送到ZIGBEE 裝置。
2. 使用者選取教學模組系統後，會跟著呼叫環境判斷控制系統，再將控制參數傳送到裝置控制發送系統，接著發送到ZIGBEE 裝置。
3. 當使用者啟動行為狀態辨識系統後，會呼叫影像接收模組，之後將接收到的影像資訊做判斷，判斷好的資訊傳送到通知模組，再將控制參數傳送到裝置控制發送系統，接著發送到ZIGBEE 裝置；且通知模組會將資訊傳回使用者介面。

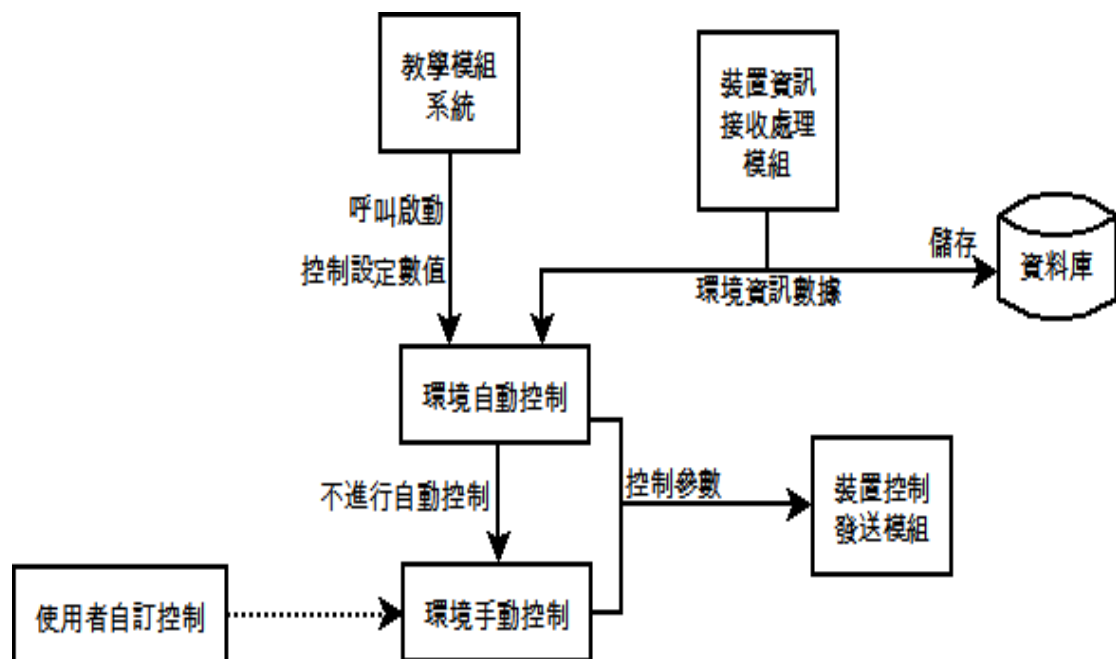
4. 把資訊接收到ZIGBEE 裝置，再將資訊傳輸到裝置資訊接收處理模組，以處理的資訊會顯示於使用者介面；並且以處理的資訊也會分別傳送到資料庫儲存以及傳送到環境判斷控制系統。

Device :

1. 會將Server 接送以及發送的資訊傳送到ZIGBEE 裝置。
2. 攝影機所接收的影像傳送到 Server 端。

3.2.2 教學環境管理系統

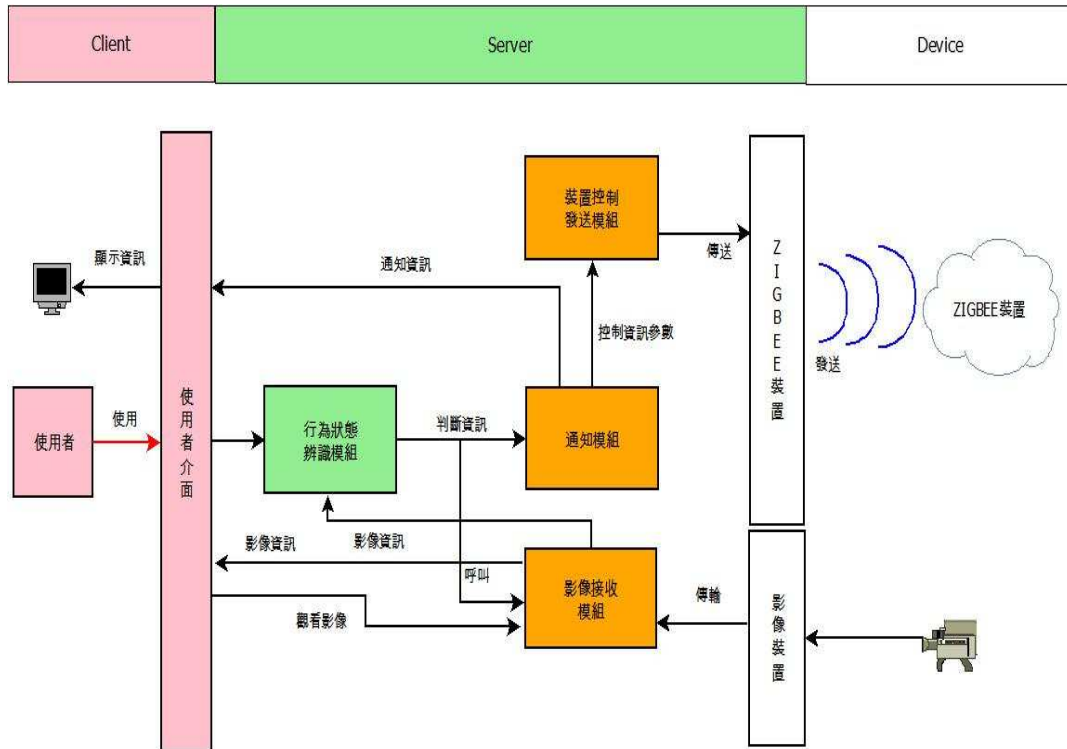
教學環境管理系統之模組圖



當系統啟動後，裝置接收處理模組會自動處理接收到的環境資訊做分析，並將處理好的資訊存入資料庫且傳送至環境自動控制模組跟教學模組所傳送的預定數值做分析。

3.3 學習狀態管理系統

3.3.1 系統架構圖



Client :

1. 使用者啟動程式。
2. 將 Server 端傳回的資訊顯示於介面。

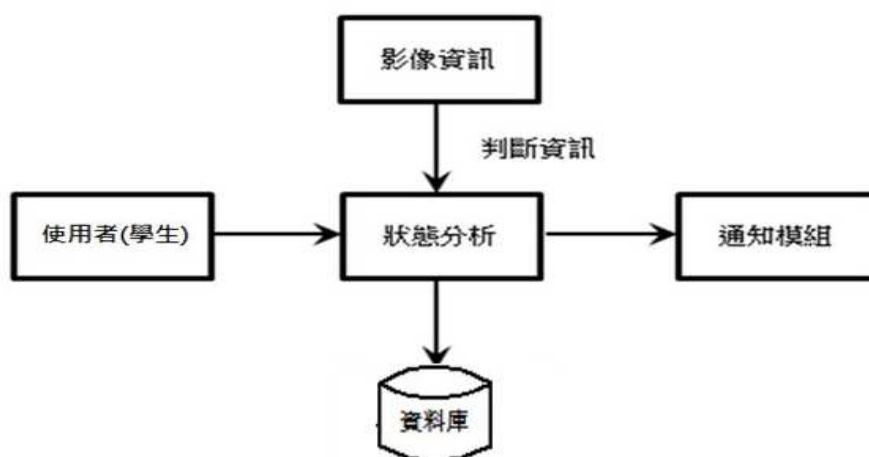
Server :

1. 當使用者啟動行為狀態辨識系統後，會呼叫影像接收模組，之後將接收到的影像資訊做判斷，判斷好的資訊傳送到通知模組，再將控制參數傳送到裝置控制發送系統，接著發送到 ZIGBEE 裝置；且通知模組會將資訊傳回使用者介面。

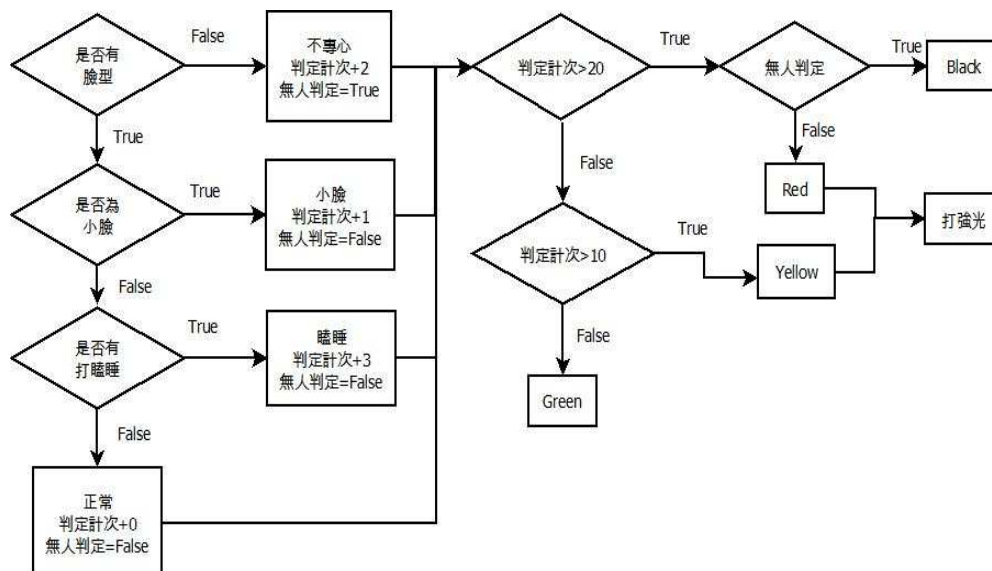
Device :

1. 會將 Server 接送以及發送的資訊傳送到 ZIGBEE 裝置。
2. 攝影機所接收的影像傳送到 Server 端。

3.3.2 學習狀態管理系統之模組圖



3.3.3 學習狀態管理系統流程圖



鏡頭一次拍 20 張照片，以下列表格將 20 張照片的分數累加，及最後判定分數

狀態	加權分數
正常	+0
小臉	+1
臉不在畫面中	+2
瞌睡	+3

最後判定分數對照表範圍，即為最終判定

最終判定狀態	分數範圍
綠色(專心)	0~8
黃色(不專心)	9~25
紅色(極度不專心)	26~40
黑色(離席)	連續兩次紅色

3.3.4 學習狀態管理系統回饋

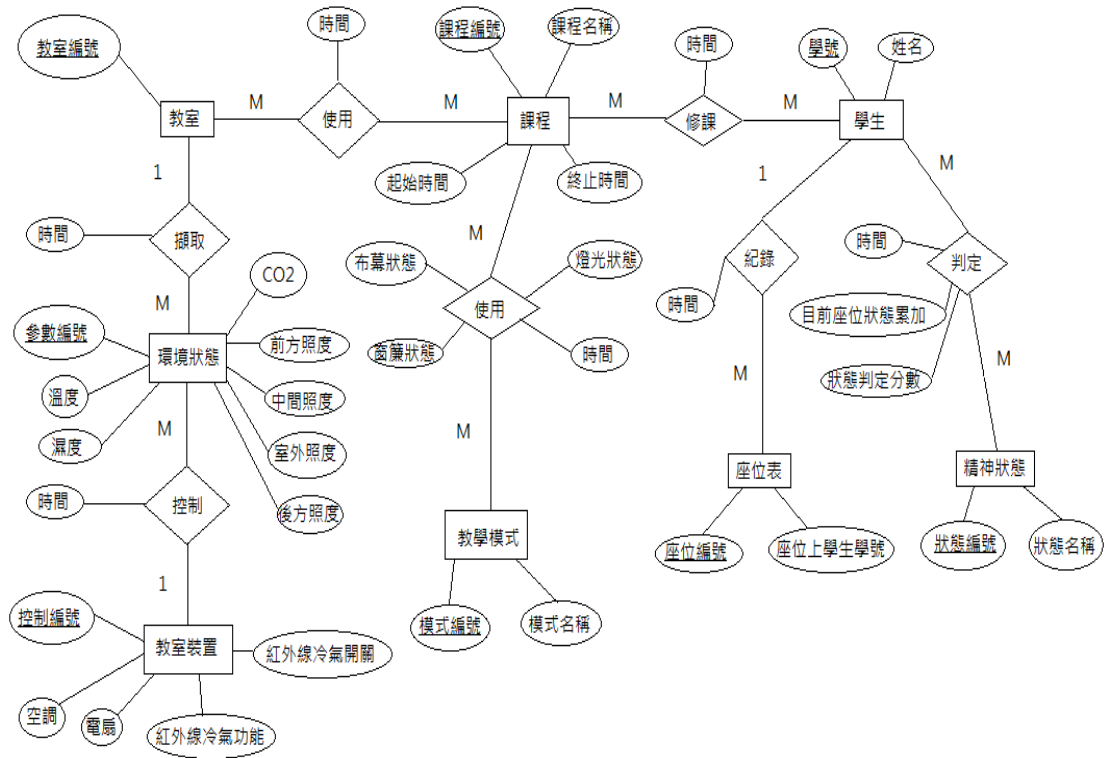
回饋對象	回饋方式
學生	偵測該學生為極度不專心狀態或不專心狀態時，會給予LED警示，警告同學現在上課狀態是不好的
老師	當系統偵測到某位同學為極度不專心或不專心狀態時，介面上會在該座位顯示紅色或黃色，當上課不專心+極度不專心超過一定比例時，會跳出視窗提醒授課老師是否改變教學策略

3.4 遠端監控系統

利用 Visual 2010 平台撰寫網頁

1. 從資料庫中，抓取最新一筆教室環境資料 (溫度、濕度、CO2、教室前方亮度，教室中間亮度、教室後面亮度以及室外亮度)。
2. 利用網頁查詢學生上課狀態，輸入學號，就會從資料庫抓出該學生所有課堂上課情形。
3. 利用網頁補點名，老師只須開啟網頁，輸入帳號密碼，即可登入，當選好該學生位置，並點選該學生姓名，按下開始補點，就會在資料庫新增該學生資料。

3.5 本系統所需之資料庫 ER-model 和 Table Schema



3.6 分工規劃

在研究撰寫程式的過程中，我們將程式分成三個主要類別以及資料庫，進行程式撰寫，分別由各個同學進行實作：

A：教學模式管理系統

建構方法：制定相關裝置設定，進行發送訊號，進行測試

負責人：林淑萍

額外工作：CO2 封包發送與偵測程式撰寫

B：紅外線冷氣發送控制

建構方法：用 C#建置函式，將封包發送碼放入

負責人：彭彥彰

額外工作：紅外線冷氣功能撰寫

C：整合 (教室環境/模式 管理系統 & 學系狀態系統 & 人臉自動點名系統)

建構方法：使用 Microsoft SQL 2008 進行資料庫架設

負責人：王儷蓉、彭彥彰、林淑萍

額外工作：建立資料庫、撰寫資料庫程式

第四章 系統開發

4.1 前言

在了解此程式之前，須預先準備幾項步驟：

- 1.了解Microsoft Visual Studio C# 程式語言寫法及使用
- 2.初步了解 TinyOS (ZIGBEE 內部的嵌入式語法)

4.2 硬體/系統需求

CPU：intel Pentium 4 2.0GHz 以上 或是 AMD Athlon 2400+ 以上

記憶體：1G 以上

顯示卡：相容於 DirectX 9b 以上之晶片

Framework：.NET 4.0 以上

作業程式：Microsoft Windows XP、VISTA 32/64bit、W7 32/64bit

無線裝置：採用瑞帝電通代理之ZIGBEE 模組(ATMEL ATmega 128L 微處理器、TI Chipcon 2420 傳輸器)

4.3 開發環境

Microsoft Visual Studio C# 2010

Microsoft SQL Server 2008

4.4 程式碼簡介

1. 布幕控制

```
//電動布幕上
private void button1_Click(object sender, EventArgs e)
{
    cmd = "7E 42 FF FF 17 27 0D 24 0A 01 07 CC CC CC CC CC CC CC CC 00 00 7E"; //布幕上
    this.BeginInvoke(new EventHandler(delegate
    {
        SendControl(cmd);
    }));
    Application.DoEvents();
    label31.Text = "UP";
    label31.ForeColor = Color.LawnGreen;
}

//電動布幕下
private void button21_Click(object sender, EventArgs e)
{
    cmd = "7E 42 FF FF 17 27 0D 24 0A 01 08 CC CC CC CC CC CC CC CC 00 00 7E"; //布幕下
    this.BeginInvoke(new EventHandler(delegate
    {
        SendControl(cmd);
    }));
    Application.DoEvents();
    label31.Text = "DOWN";
    label31.ForeColor = Color.Red;
}
```

```
cmd = "7E 42 FF FF 17 27 0D 24 0A 01 07/08 CC CC CC CC CC CC CC
CC CC 00 00 7E"
```

裝置控制的代碼

```
this.BeginInvoke(new EventHandler(delegate
```

```
{
```

```
SendControl(cmd);
```

```
});
```

呼叫副程式 SendControl 並將 cmd 傳送至副程式

```
Application.DoEvents();
```

可以讓您的應用程式在執程式碼的同時，也能夠處理可能所引發的其他事

2. 接收觸發 CO2

```
//接收觸發co2
private void serialPort2_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    byte[] tmp1 = new byte[41];
    try
    {
        //回傳封包格式
        //[RECV (41)] <- 7E 45 00 FF FF FF FF 1C 00 93 00 00 02 00 00 01 00 41 0C E0 0C A9 0C A6 0C A0 0C E0 0C A6 0C A6 0C A3 0C E0 0C A9 DC F6 7E
        //頭          GroupID          CO2感測值          end
        CO2 = "";
        for (int i = 0; i < tmp1.Length; i++)
        {
            tmp1[i] = Convert.ToByte(serialPort2.ReadByte());
            if (i == 19 || i == 20)
            {
                CO2 = CO2.Insert(0, Convert.ToString(tmp1[i].ToString("X2")));
            }
        }
        if (tmp1[20] == 01 || tmp1[20] == 02 || tmp1[20] == 03)
        {
            this.BeginInvoke(new EventHandler(delegate
            {
                SetTheText2(tmp1);
            }));
            Application.DoEvents();
        }
    }
    catch (System.Exception)
    {
        if (serialPort2.IsOpen == true)
        {
            serialPort2.DiscardInBuffer();
            tmp1 = null;
        }
    }
}
```

接收方式採用 serialPort 的觸發事件，先宣告陣列來接收傳回的資料封包FOR
迴圈內個別抓取CO2數據，並轉換成10 進位，IF 來判斷回傳的封包，是否為正
確裝置所回傳的

```
this.BeginInvoke(new EventHandler(delegate
```

```
{
```

```
SetTheText(tmp);
```

```
});
```

```
Application.DoEvents();
```

呼叫處理端副程式進行運算分析

3. CO2 接收端處理

```
//CO2接收端處理
private void SetTheText2(byte[] tmp1)//string strText)
{
    try
    {
        CO2_P = Convert.ToInt32(CO2, 16).ToString(); //16進制轉10進制

        if (tmp1[20] == 01)
        {
            label173.Text = CO2_P;
        }
        else
        {
            label173.Text = CO2_P;
        }
        //CO2濃度 > 850 ppm -> 排風扇開

        if (Int32.Parse(CO2_P) > 850)
        {
            cmd = "7E 42 FF FF 17 27 0D 49 0A 01 02 CC CC CC CC CC CC CC CC 00 00 7E"; //開
            this.BeginInvoke(new EventHandler(delegate
            {
                SendControl(cmd);
            }));
            Application.DoEvents();
            label129.Text = "ON";
            label129.ForeColor = Color.LawnGreen;
            CO2_counter1 = 1;
        }

        //CO2濃度 < 850 ppm -> 排風扇關
        else
        {
            if ((CO2_counter1 == 1) && (CO2_counter2 != 20))
            {
                CO2_counter2++;
            }
            else
            {
                cmd = "7E 42 FF FF 17 27 0D 49 0A 01 01 CC CC CC CC CC CC CC CC 00 00 7E"; //關
                this.BeginInvoke(new EventHandler(delegate
                {
                    SendControl(cmd);
                }));

                Application.DoEvents();
                label129.Text = "OFF";
                label129.ForeColor = Color.Red;
            }
        }
    }
}

catch (System.Exception)
{
    cmd = "7E 42 FF FF 17 27 0D 49 0A 01 01 CC CC CC CC CC CC CC CC 00 00 7E"; //關
    this.BeginInvoke(new EventHandler(delegate
    {
        SendControl(cmd);
    }));
    Application.DoEvents();
    label129.Text = "OFF";
    label129.ForeColor = Color.Red;
}
}
```

將接收到的資訊 作運算處理.....及 各項數據的運算方式並輸出至介面，並自動設計，當 CO2 值>850 時，排風機開，<850 時，排風機關。

4. 發送端 CO2

```
private void SendControl2(string cmd)
{
    string tmp1 = cmd.Replace(" ", string.Empty);
    byte[] buff1 = new byte[tmp1.Length / 2];
    for (int i = 0; i < buff1.Length; i++)
    {
        buff1[i] = Convert.ToByte(tmp1.Substring(2 * i, 2), 16);
    }
    try
    {
        // Send the character buffer.
        serialPort2.Write(buff1, 0, buff1.Length);
    }
    catch (System.Exception)
    {
        MessageBox.Show("無法與裝置聯繫(CO2)", "E408");
    }
}
```

```
private void SendControl(string cmd)
```

副程式名稱 所傳送至的資料形態及命名

```
string tmp = cmd.Replace(" ", string.Empty);
```

tmp 為cmd 中的資料拿掉空白格

```
cmd = "7E 42 FF FF 17 27 0D 48 0A 01 01 CC CC CC CC CC CC CC CC  
00 00 7E"
```

```
tmp = "7E42FFFF17270D480A0101CCCCCCCCCCCCCCCC00007E"
```

```
byte[] buff = new byte[tmp.Length / 2];
```

宣告 byte型態之buff陣列 的大小

```
for (int i = 0; i < buff.Length; i++)
```

```
{
```

```
buff[i] = Convert.ToByte(tmp.Substring(2 * i, 2), 16);
```



```
}
```

將tmp 的資料每2 個字元 轉換成16 進位 存入byte 型態陣列

```
try
```

```
{
```

```
serialPort1.Write(buff, 0, buff.Length);
```

```
}
```

```
catch (System.Exception)
```

```
{
```

```
MessageBox.Show("無法與裝置聯繫", "E408");
```

```
}
```

try catch 容錯用法

try 內的程式碼為可能出錯的程式

catch 內則定義try 內出錯時 該執行哪些動作，後方System.Exception 定義為

出現何種錯誤時，try 內的serialPort1.Write(buff, 0, buff.Length)； 為將buff 陣

列中的資料 1 個個 寫入serialPort；MessageBox.Show("無法與裝置聯繫",

"E408")； "出現通知視窗" 及 "視窗內容文字" 和 "視窗標頭" 。

5. 紅外線冷氣

```
//冷氣封包格式
private void SendCmd(byte node, byte cmd1, byte cmd2)
{
    byte[] buffer = new byte[3];
    buffer[0] = node;
    buffer[1] = cmd1;
    buffer[2] = cmd2;
    serialPort3.Write(buffer, 0, 3);
}
//冷氣電源
private void button40_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x01);
    air_count++;

    if (air_count % 2 == 1)
    {
        label28.Text = "ON";
        label28.ForeColor = Color.LawnGreen;
    }
    else if (air_count % 2 == 0)
    {
        label28.Text = "OFF";
        label28.ForeColor = Color.Red;
    }
}
//冷氣溫度(上)
private void button41_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x02);
}
//冷氣溫度(下)
private void button42_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x03);
}
//冷氣機能
private void button43_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x04);
}
//冷氣風速
private void button44_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x06);
}
//冷氣FUZZY
private void button45_Click(object sender, EventArgs e)
{
    SendCmd(0x14, 0x07, 0x08);
}
```

第五章 研究成果

5.1 硬體/系統需求

CPU : intel Pentium 4.0 GHz 以上 或是 AMD Athlon 2400+ 以上

記憶體 : 1G 以上

顯示卡 : 相容於 DirectX 9b 以上之晶片

Framework : .NET`4.0 以上

作業程式 : Microsoft Windows XP、VISTA 32/64bit、W7 32/64bit

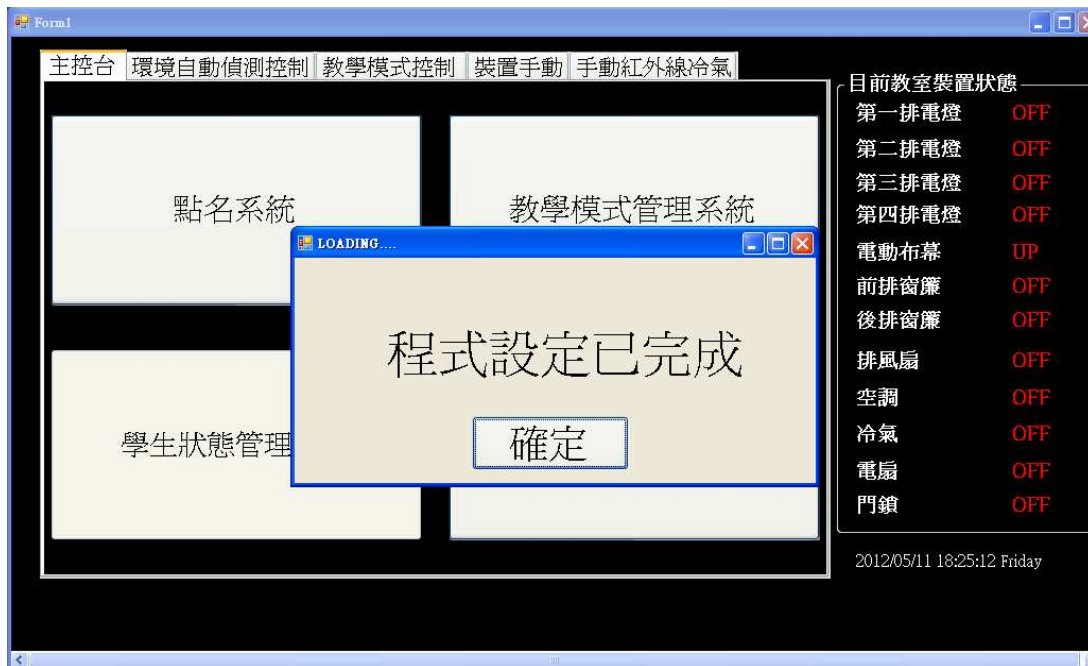
無線裝置 : 採用瑞帝電通代理之ZIGBEE 模組(ATMEL ATmega 128L 微處理器、TI Chipcon 2420 傳輸器)

5.2 程式操作簡介

5.2.1 點選E408.exe檔 啟動程式畫面Lading中



5.2.2 Loading好會顯示程式設定以完成，此時按下確定即可開始操作

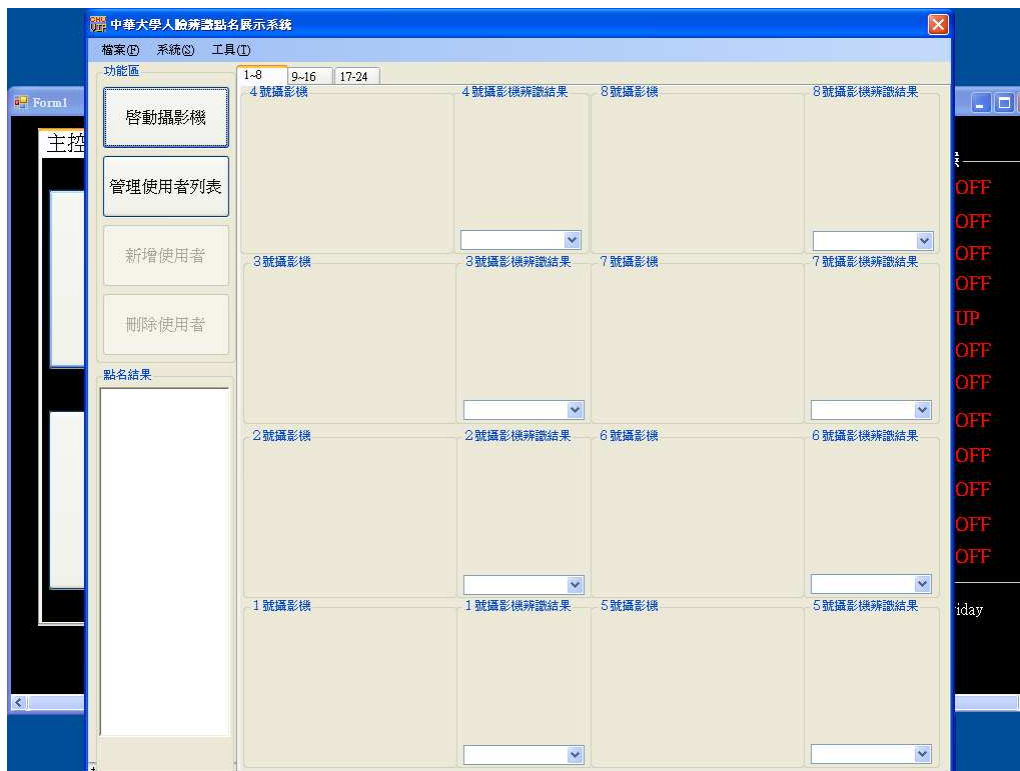


開始介面只有點名系統、教學模式管理系統、教學環境管理系統可以點選，在還沒點選時，其它頁面是無法看到的。當點選教學模式管理系統按鈕時，程式頁面會跳到那頁，此時當個頁面內容才可使用，點選教學環境管理系統時，環境方面才可以看到。如須點選學生狀態管理系統時，須先點完名才可使用，所以一開始按鈕是不可以點選的。

5.2.3 點選點名系統時，會呼叫出另一支程式



↑ 程式讀取中



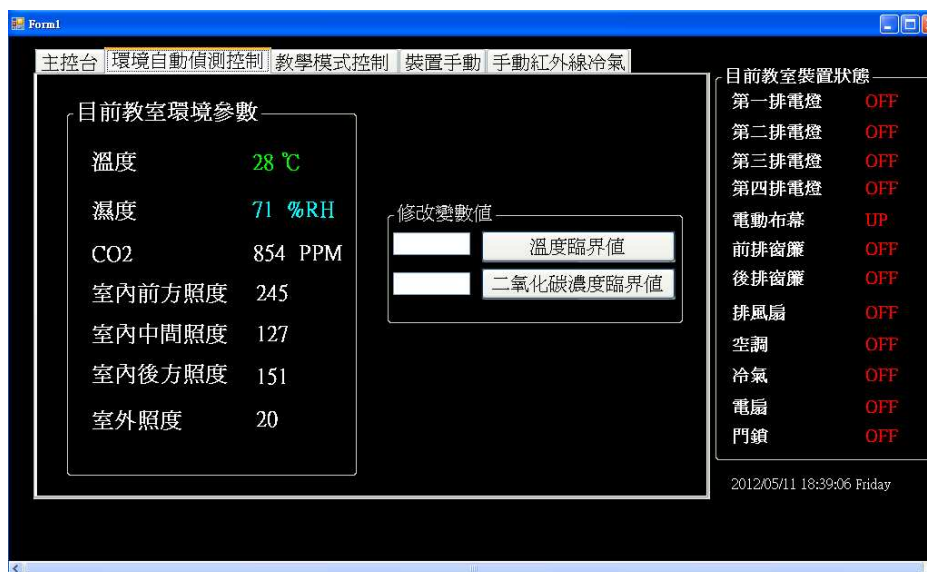
↑ 即可開始點名，點完名就可以把此視窗關掉

5.2.4 點完名才可以點選學生狀態管理系統，此時會再呼叫出另一支程式



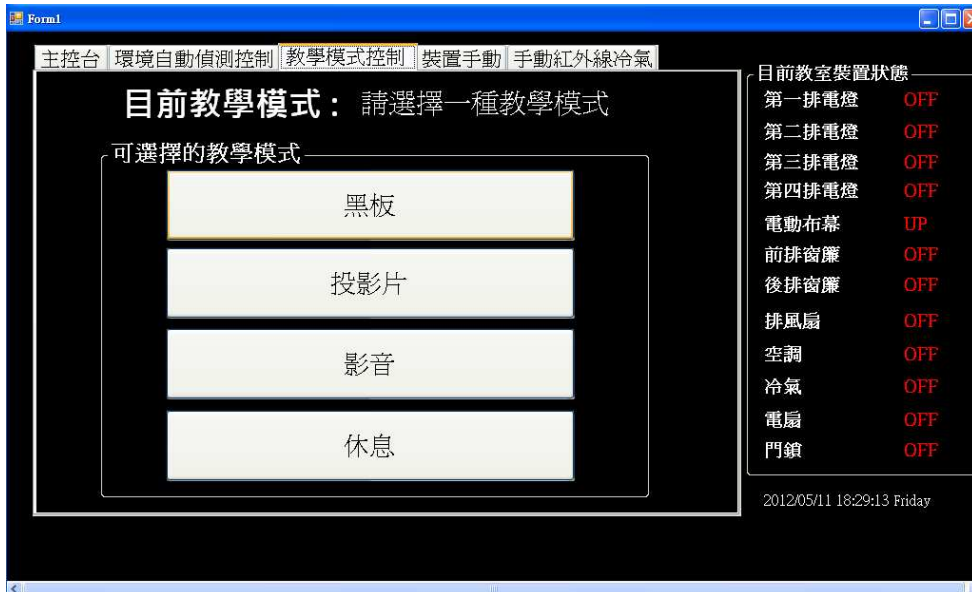
介面顯示剛點名結果完，有出席學生的照片、未出席名單、不專心名單、專心、不專心、極度不專心、缺席數據；當不專心加上極度不專心超過一定比例時，會彈出視窗來提醒老師是否應該請學生起來問問題，提高學生精神狀態；當要結束程式時，請按結束程式按鈕。

5.2.5 教學環境管理系統介面



可以知道目前教室內的所有感測值，並可以手動設定數據

5.2.6 教學模式控制



在左方教學模組介面裡，有4種模式能選擇，如下(預設)

黑板：當室外照度 <500 時，1&2&3&4排燈開、窗簾前、後開，布幕上

，當室外照度 >500 時，1&2排燈開，3&4排關，窗簾前關後開，布幕上

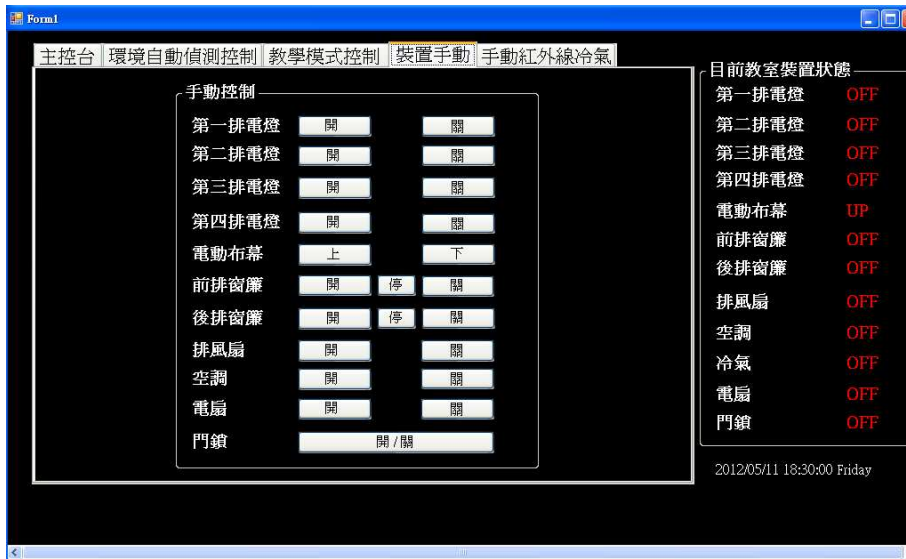
投影片：1&2排燈關、3&4排燈開、窗簾前關後開、布幕下

影音：1&2&3&4排燈關、窗簾前、後關、布幕下

休息：1&2&3排燈關、第3排燈開、窗簾前、後開、布幕上

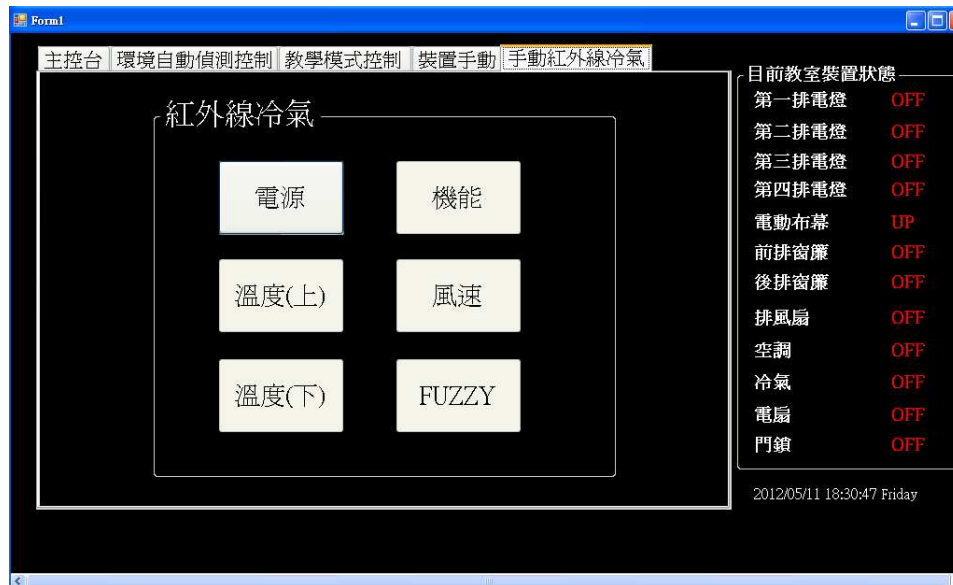
(附註：當選擇模組後，須稍待片刻，等待系統將訊號發送出去。)

5.2.7 裝置手動控制 (下圖)



點選按鈕會執行按鈕上的功能

5.2.8 手動紅外線冷氣



點選按鈕會執行按鈕上的功能

5.2.9 教學遠端監測系統



請在網址列輸入140.126.11.163/E408.aspx，此時會出現首頁

5.2.10 點選即時環境感測值按鈕會出現下列網頁



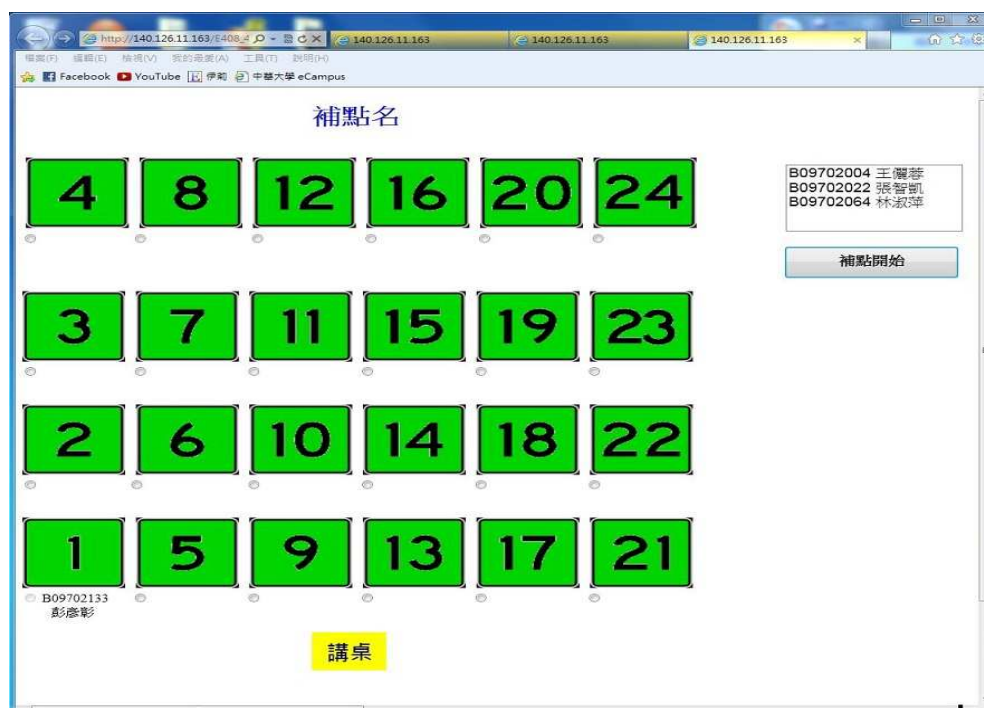
點選即時感測環境(查詢)，即可看到最新的環境狀態

5.2.11 點選學生狀態(查詢)



在欄位裡輸入學號，按下學生狀態(查詢)按鈕即會出現學生姓名、課堂最終狀態

5.2.12 點選補點名按鈕(須帳號密碼)



會顯示當節課位出席名單，先點選位置，在點選右邊名字，按下補點開始按鈕，

即會開始點名，補點成功後，會顯示在座位名單中。

第六章 結論與未來研究方向

結論

由於近年來大學生活越來越自由，少了高中時期以前的約束感，大學生們變得時差顛倒，早上起不來、晚上不睡覺；並且現在的孩子們都在父母親的寵愛下成長，養成了一種「想做什麼，就做什麼」的慣性，導致上課人數的減少，翹課率大增。而現今的氣候卻因聖嬰現象導致「冬天寒冷，夏天酷熱」的天氣，使得學生們上課會因環境因素而影響了上課的心情。

因此，我們期望此系統的研發「教室環境管理系統」之功能可以使學生們不會因為環境因素等問題，而影響了上課心情，而此系統可以給學生們一個最舒適的學習環境，以提升學生的學習效率。

另外，此系統研發了一個「教學模式管理系統」功能，希望對於老師們在上課時，切換教學系統可以更加快速以及方便。由於老師在上課時，有時會使用投影片、有時會使用黑板、或者是放影片等，那在切換這些教學模式時，會耗費不少時間，而導致學生的學習時間減少，所以我們在「教學模式管理系統」中預設了四種模式：「黑板教學模式」、「投影片教學模式」、「影片教學模式」以及「休息模式」。希望因此系統而增加學生的學習時間，也可以使老師們不會因切換教學模式中斷上課節奏。

並將學生學系狀態管理系統整合進來，讓老師使用更為方便，只要開一個執行檔就可以同時偵測學生上課狀態。

未來研究方向

1. 實際測試及獲得實際測試數據資料：

由於，目前只把系統研發出來，但尚未實際測試，因此，希望可以在上課時，實際操作此系統，並且獲得測試後的數據資料，以此數據知道測試後的結果是否如預期的理想。

2. 提高判斷準確度：

目前的系統在準確度方面尚有一些判斷錯誤，並沒有百分之百的準確率，因此，希望未來可以將準確度提升到百分之百。