

# 行動嵌入式系統設計

## 第二小組

### 作業一

### Arduino與電腦連線

指導老師：陳永隆

小組成員：鄭晉丞-電腦端控制C#設計

邢弘宇-實驗架構設計

駱俊吉-Arduino設計

黃拓景-PPT報告準備

劉金昌-實驗模組採購

張平和-電路模組配線

洪祐民-測試實驗結果解決Bug

許宏恩-模組個別測試

# OUTLINE

- 實驗目的
- 實驗材料與架構
- 實作方法
  - 電腦端控制-使用C#
  - Arduino實習板-使用C

# 實驗目的

- 利用電腦的C#應用程式透過USB連接Arduino Duemilanove實習板，控制實習板所連接的5V繼電器模組，藉此達到控制110V電源開關。



# 實驗材料與架構

## 實驗材料

1. Arduino Duemilanove x1
2. 繼電器模組(5V) x1
3. 電源線捆 x1
4. AC電源座(110V) x1
5. 電腦 x1



Arduino Duemilanove



繼電器模組



電源線捆



AC電源座(110V)

# 實驗架構

C#程式-Control

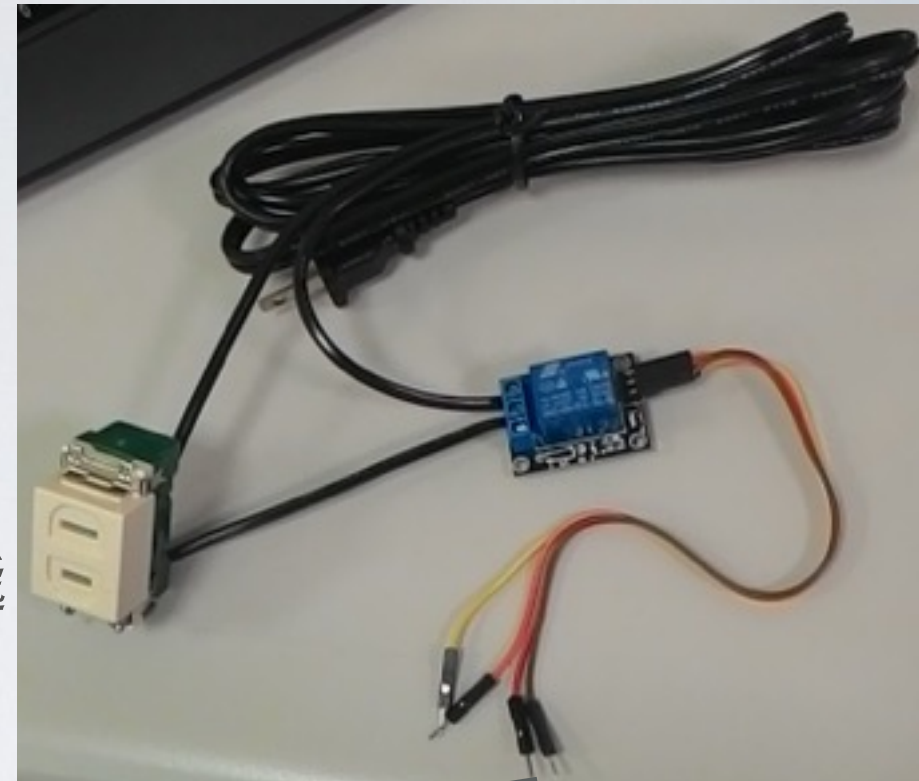


USB  
→  
控制/燒錄

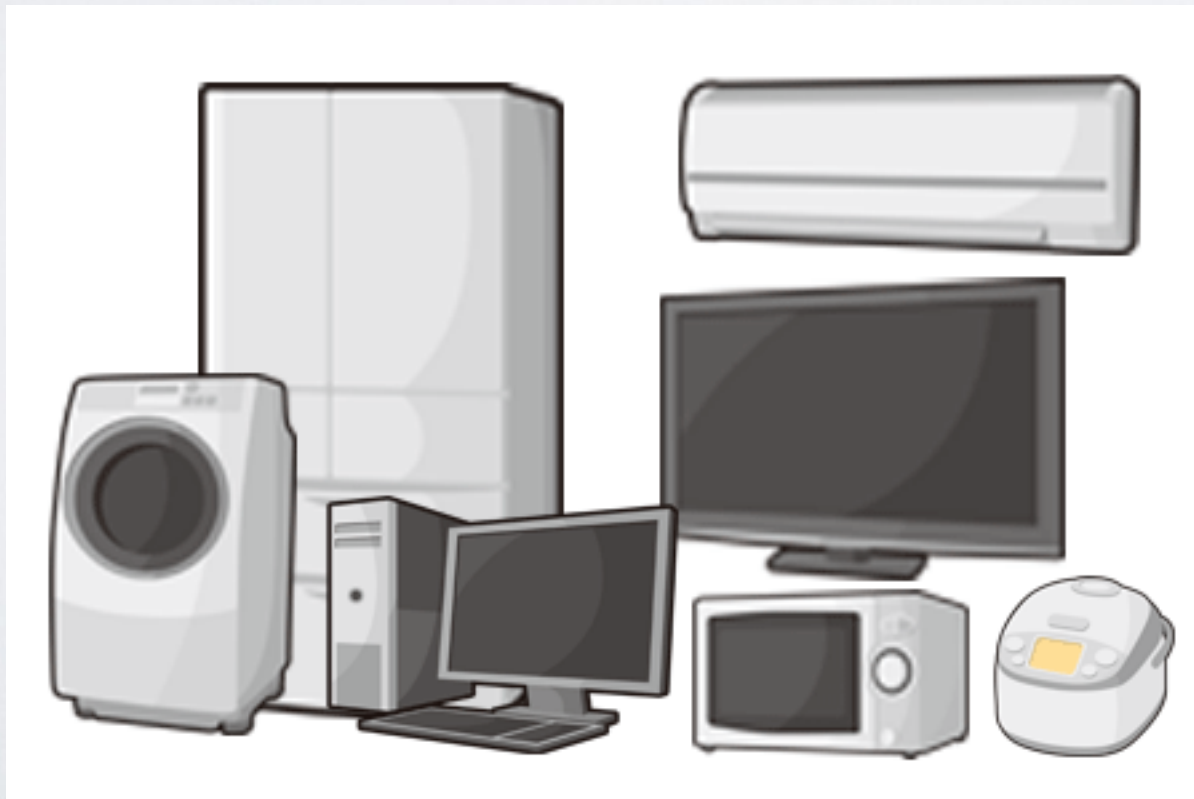
接收/發送訊號



TTL  
→  
訊號傳遞



各種家電



# ARDUINO實習板-使用C

```
//-----宣告訊號輸出、各個狀態、字元|-----//
int M_P[7] = {13,12,11,10,9,8,7};
boolean loadopen[5] = {false,false,false,false,false};
char cmd;
void setup()
{
    // define the PIN as OUTPUT
    for(int i = 0;i<7;i++){
        pinMode(M_P[i],OUTPUT);
    }

    // set baud rate 9600
    Serial.begin(9600);
}
```



```
void loop()
{
  while (Serial.available()) //connected
  {
    //讀取字元
    cmd = Serial.read();
    //選擇目標執行
    switch(cmd){
      case '0' :
        //全關
        for(int i = 0; i<5; i++){
          digitalWrite(M_P[i],0);
          loadopen[i] = false;
        }
        Serial.print("2");
        break;
```

//如果狀態是關閉的，則將狀態改為開，並傳出1的訊號。反之則改為關，傳出0的訊號。  
// digitalWrite()為板子上的訊號輸出函數 Serial.print則是傳送給SerialPort的訊息

```
case '1' :
  if(loadopen[cmd-49] == false){
    loadopen[cmd-49] = true;
    digitalWrite(M_P[cmd-49],1);
    Serial.print("1");
  }else if(loadopen[cmd-49] == true){
    loadopen[cmd-49] = false;
    digitalWrite(M_P[cmd-49],0);
    Serial.print("0");
  }
  break;
```

```

case '2' :
    if(loadopen[cmd-49] == false){
        loadopen[cmd-49] = true;
        digitalWrite(M_P[cmd-49],1);
        Serial.print("1");
    }else if(loadopen[cmd-49] == true){
        loadopen[cmd-49] = false;
        digitalWrite(M_P[cmd-49],0);
        Serial.print("0");
    }
    break;
case '3' :
    if(loadopen[cmd-49] == false){
        loadopen[cmd-49] = true;
        digitalWrite(M_P[cmd-49],1);
        Serial.print("1");
    }else if(loadopen[cmd-49] == true){
        loadopen[cmd-49] = false;
        digitalWrite(M_P[cmd-49],0);
        Serial.print("0");
    }
    break;

```

```

case '4' :
    if(loadopen[cmd-49] == false){
        loadopen[cmd-49] = true;
        digitalWrite(M_P[cmd-49],1);
        Serial.print("1");
    }else if(loadopen[cmd-49] == true){
        loadopen[cmd-49] = false;
        digitalWrite(M_P[cmd-49],0);
        Serial.print("0");
    }
    break;
case '5' :
    if(loadopen[cmd-49] == false){
        loadopen[cmd-49] = true;
        digitalWrite(M_P[cmd-49],1);
        Serial.print("1");
    }else if(loadopen[cmd-49] == true){
        loadopen[cmd-49] = false;
        digitalWrite(M_P[cmd-49],0);
        Serial.print("0");
    }
    break;
}

```



# 電腦端應用程式-使用C#

```
namespace Comport_8051
{
    public partial class Form1 : Form
    {
        //宣告變數
        SerialPort serialport = new SerialPort();
        Boolean serialportopen = false,mysqlopen = false;
        private Color[] MsgTypeColor = { Color.Blue, Color.Green, Color.Black, Color.Orange, Color.Red };
        public enum MsgType { System, User, Normal, Warning, Error };
        int id;
        string msbuff;
        OvalShape[] array;
        private string dbhost, dbname, dbpass, dbuser, dbchar, connstr;
        MySqlConnection conn;

        public Form1()
        {
            InitializeComponent();
            //宣告button.click
            button1.Click += new EventHandler(button_Click);
            button2.Click += new EventHandler(button_Click);
            button3.Click += new EventHandler(button_Click);
            button4.Click += new EventHandler(button_Click);
            button5.Click += new EventHandler(button_Click);
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            comboBox1.Items.Clear();
            array = new OvalShape[5] { ovalShape1, ovalShape2, ovalShape3, ovalShape4, ovalShape5 };
            //找尋可用的串列埠
            foreach (string com in SerialPort.GetPortNames())
            {
                comboBox1.Items.Add(com);
            }
        }
    }
}
```

```

//LOAD1~5
private void button_Click(object sender, EventArgs e)
{
    Button btn = (Button)sender;
    id = int.Parse(btn.Text.Substring(4));
    try
    {
        serialport.Write(btn.Text.Substring(4)); //透過串列埠傳送字串1~5
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}

//ALL OFF
private void btn_off_Click(object sender, EventArgs e)
{
    serialport.Write("0");
    if (mysqlopen == true)
    {
        try
        {
            for (int i = 1; i < 6; i++)
            {
                update(i.ToString(), 0, 99);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
}

//結束
private void btn_end_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("確定要關閉程式?", "提醒", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
    if (result == DialogResult.OK)
    {
        this.Close();
    }
}

```

//串列埠連線、中斷

```
private void btn_connect_Click(object sender, EventArgs e)
{
    //串列埠連線
    if (serialportopen == false && !serialport.IsOpen)
    {
        try
        {
            //設定連接埠9600、n、8、1、n
            serialport.PortName = comboBox1.Text;
            serialport.BaudRate = 9600;
            serialport.DataBits = 8;
            serialport.Parity = Parity.None;
            serialport.StopBits = StopBits.One;
            serialport.Encoding = Encoding.Default;//傳輸編碼方式
            serialport.Open();//串列埠開始連線
            serialportopen = true;
            btn_true();

            //開始背景執行緒
            if (this.backgroundWorker1.WorkerReportsProgress != true)
            {
                this.backgroundWorker1.WorkerReportsProgress = true;
                this.backgroundWorker1.RunWorkerAsync();
            }
            btn_connect.Text = "中斷";

            serialport.Write("0");
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
}
```

//串列埠中斷

```
else if (serialportopen == true && serialport.IsOpen)
{
    try
    {
        serialport.Close();
        serialportopen = false;
        btn_false();
        this.backgroundWorker1.WorkerReportsProgress = false;
        this.backgroundWorker1.CancelAsync();
        this.backgroundWorker1.Dispose();
        btn_connect.Text = "連線";
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
```



```
private void backgroundWorker1_ProgressChanged(object sender, ProgressChangedEventArgs e)
{
    try
    {
        if (serialport.BytesToRead != 0)
        {
            //接收訊號
            msbuff = serialport.ReadExisting();

            //判斷燈號
            if (msbuff == "1")
            {
                array[id - 1].FillColor = Color.FromArgb(0, 255, 0);
                update(Convert.ToString(id), 0, id);
            }
            else if (msbuff == "0")
            {
                array[id - 1].FillColor = Color.FromArgb(255, 0, 0);
                update(Convert.ToString(id), 1, id);
            }
            else if (msbuff == "2")
            {
                for (int i = 0; i < 5; i++)
                {
                    array[i].FillColor = Color.FromArgb(0, 255, 0);
                }
            }
            else
            {
                AddText(MsgType.Error, "連線失敗" + "\r\n");
            }

            serialport.DiscardInBuffer();
        }
    }
    catch (Exception)
    {
    }
}
```

//資料庫初始化設定

```
private void Initialize()
```

```
{
```

```
    dbhost = host.Text;//host位址
```

```
    dbuser = user.Text;//帳號
```

```
    dbpass = pwd.Text;//密碼
```

```
    dbname = "8051_test";//資料庫
```

```
    dbchar = "utf8";//編碼格式
```

```
    connstr = "server=" + dbhost + ";uid=" + dbuser + ";pwd=" + dbpass + ";database=" + dbname + ";CharSet=" + dbchar;
```

```
    conn = new MySqlConnection(connstr);
```

```
}
```

//資料庫連線

```
private void mysql_connect_Click(object sender, EventArgs e)
```

```
{
```

```
    timer1.Enabled = true;
```

```
    Initialize();
```

```
    try
```

```
    {
```

```
        AddText(MsgType.System, "IP : " + dbhost + "\r\n");
```

```
        conn.Open();
```

```
        mysql_connect.Enabled = false;
```

```
        mysql_disconnect.Enabled = true;
```

```
        mysqlopen = true;
```

```
    }
```

```
    catch (MySqlException ex)
```

```
    {
```

```
        switch (ex.Number)
```

```
        {
```

```
            case 0:
```

```
                AddText(MsgType.Warning, "無法連線到資料庫\r\n");
```

```
                break;
```

```
            case 1045:
```

```
                AddText(MsgType.Warning, "使用者帳號或密碼錯誤\r\n");
```

```
                break;
```

```
            case 1042:
```

```
                AddText(MsgType.Warning, "無效的主機名稱\r\n");
```

```
                break;
```

```
        }
```

```
    }
```

```
}
```

//資料庫中斷

```
private void mysql_disconnect_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    try
    {
        conn.Close();
        mysqlopen = false;
        mysql_connect.Enabled = true;
        mysql_disconnect.Enabled = false;
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.ToString());
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
```

//查詢

```
private void select()
{
    string sql = "SELECT * FROM test_8051";
    MySqlCommand cmd = new MySqlCommand(sql, conn);
    MySqlDataReader mydata = cmd.ExecuteReader();

    if (!mydata.HasRows)
    {
        AddText(MsgType.User, "no data");
    }
    else
    {
        while (mydata.Read())
        {
            for (int i = 1; i < 6; i++)
            {
                AddText(MsgType.User, mydata.GetString(i));
            }
            AddText(MsgType.User, "\r\n");

            for (int i = 1; i < 6; i++)
            {
                if (mydata.GetString(i) == "0")
                {
                    if (array[i - 1].FillColor == Color.FromArgb(255, 0, 0))
                    {
                        serialport.Write(Convert.ToString(i));
                        array[i - 1].FillColor = Color.FromArgb(0, 255, 0);
                    }
                }
                else if (mydata.GetString(i) == "1")
                {
                    if (array[i - 1].FillColor == Color.FromArgb(0, 255, 0))
                    {
                        serialport.Write(Convert.ToString(i));
                        array[i - 1].FillColor = Color.FromArgb(255, 0, 0);
                    }
                }
                else
                {
                    AddText(MsgType.Error, "資料錯誤" + "\r\n");
                }
            }
        }
        mydata.Close();
    }
}
```



```

//更新
private void update(string a, int o, int s)
{
    try
    {
        string sql = "UPDATE test_8051 SET A10" + a + " = '" + o + "', state = '" + s + "'";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.ToString());
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (mysqlopen == true)
    {
        select();
    }
}

private void btn_false()
{
    button1.Enabled = false;
    button2.Enabled = false;
    button3.Enabled = false;
    button4.Enabled = false;
    button5.Enabled = false;
    btn_off.Enabled = false;
}

private void btn_true()
{
    button1.Enabled = true;
    button2.Enabled = true;
    button3.Enabled = true;
    button4.Enabled = true;
    button5.Enabled = true;
    btn_off.Enabled = true;
}

```

THE END