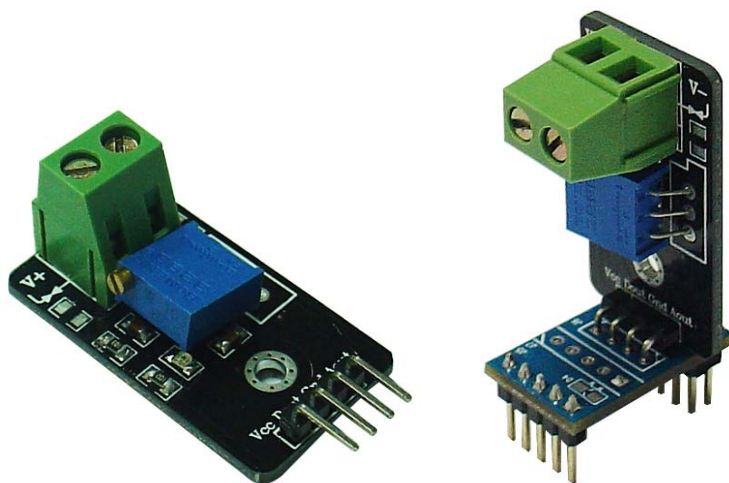


# 電壓過壓檢測模組

## 1-1 電壓過壓檢測模組 (E380)



與轉接板連接圖

## 1、相關知識：

全球的感測器市場在不斷變化的創新之中呈現出快速增長的趨勢。有關專家指出，感測器領域的主要技術將在現有基礎上予以延伸和提高，各國將競相加速新一代感測器的開發和產業化，競爭也將日益激烈。新技術的發展將重新定義未來的感測器市場，比如無線感測器、光纖傳感器、智慧感測器和金屬氧化感測器等新型感測器的出現與市場份額的擴大。

而過壓感測器則可算是智慧感測器中的一種，主要利用弱電控制強電的原理，這樣做即方便於電路控制，更有利的是對人身更安全，可靠。

### 技術參數：

當大於當前設定值時，本產品透過信號處理後，在Dout接腳會有TTL電平輸出，其模組相關資訊如下：

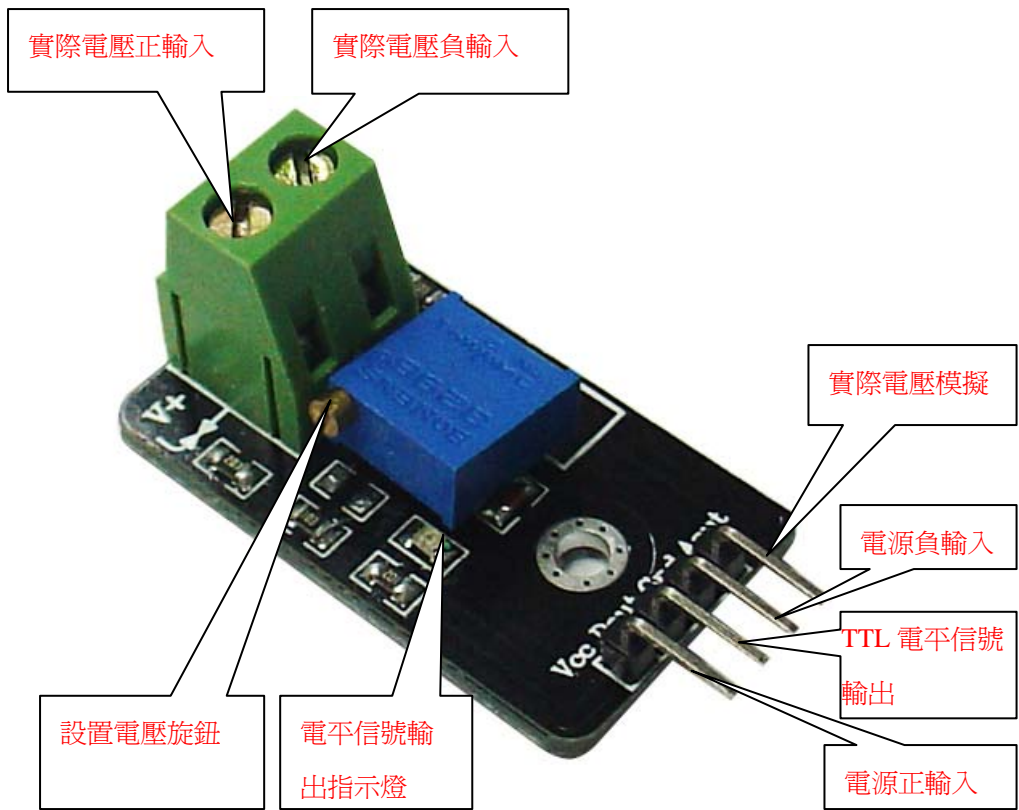
- 一、 尺寸：長 28mm X 寬 16mm X 高 15mm
- 二、 主要晶片：LM393
- 三、 工作電壓：直流 DC 5V
- 四、 特點：
  - 1、具有信號輸出指示；
  - 2、電壓檢測範圍寬 DC 0V-25V 輸入；
  - 3、電壓解析度範圍寬 DC 0V-25V 輸入；
  - 4、輸出過欠壓信號指示；
  - 5、過欠壓臨界點可調設置，解析度 0.2V；
  - 6、帶安裝孔，方便固件安裝；
  - 7、帶採樣電壓類比信號輸出，可直接輸入 AD，TTL 電平信號輸出，可直接接單片機 IO 口控制。

### 應用範圍：

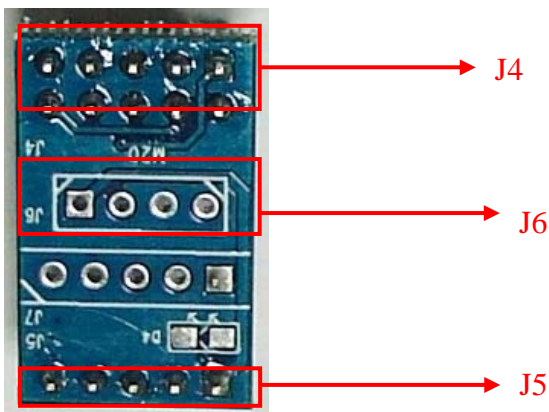
感測器學習、電壓過欠壓檢測，智慧車鋰電池電壓檢測，電子競賽、產品開發、畢業設計等。

## 2、接腳定義：

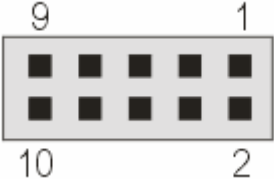
管腳說明：




轉接板接腳定義：



**J4 接腳定義：**

	接腳	功能
	1	P1.7
	2	P2.0
	3	SDI
	4	SCL
	5	SDO
	6	SSS
	7	I2C_SCL
	8	INT
	9	I2C_SDA
	10	ADC0

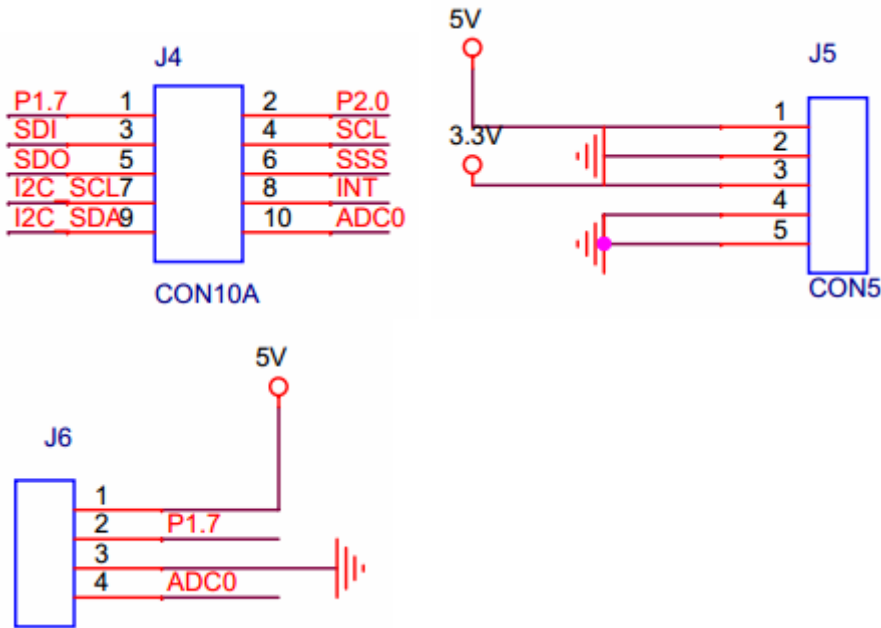
**J5 接腳定義：**

	接腳	功能
	1	5V
	2	GND
	3	3.3V
	4	GND
5	GND	

**J6 接腳定義：**

	接腳	功能
	1	5V
	2	P1.7
	4	ADC0

### 3、模組相關電路介紹：

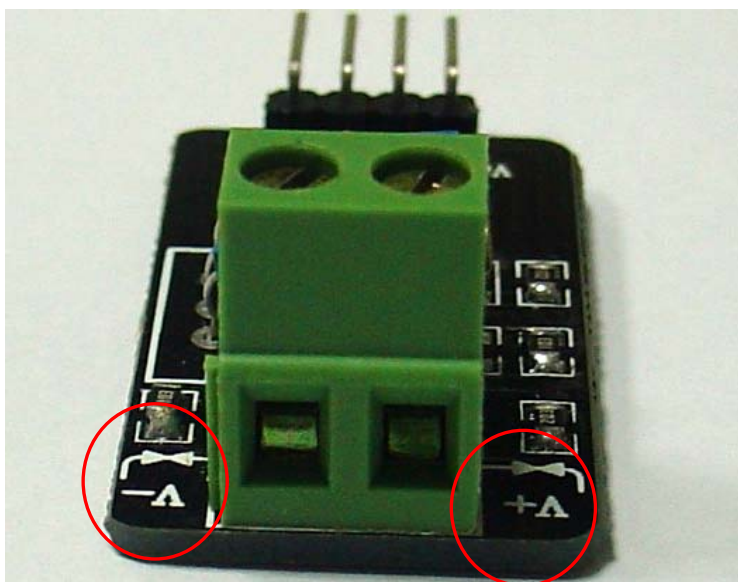


### 4、模組使用方法

- 1、在模組的 VCC 介面，接入+5V 電源。
- 2、在模組的 GND 介面，接入地。
- 3、模組的 Dout 是一個數字高低電平輸出腳位，可以連接到單片機的一個 IO 口上。
- 4、模組的 Aout 是一個類比電壓輸出腳位，可以連接到單片機的一個 ADC 採集口上，從而讀取到電壓值。
- 5、在模組的 V+和 V-口連接電源線，電壓源正極接到 V+端，負極接到 V-端。

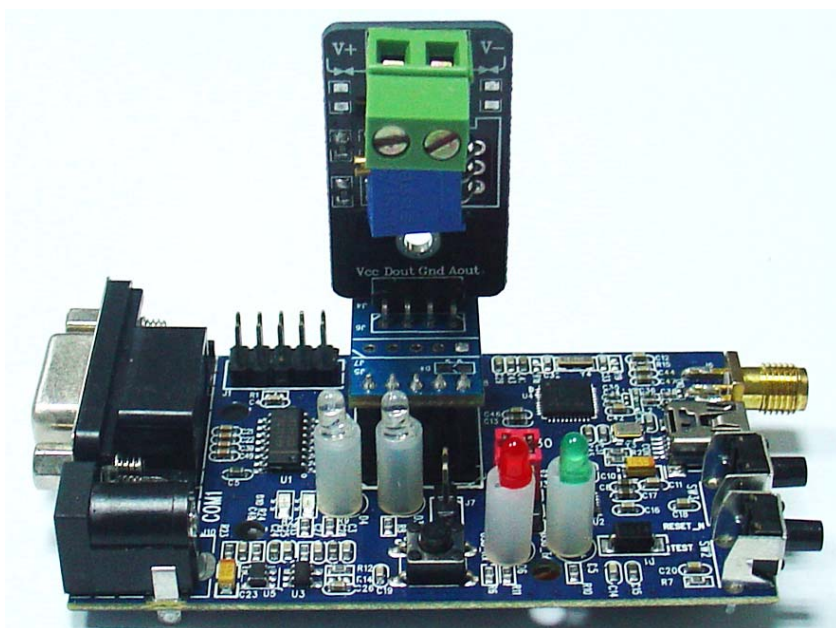
#### 注意：

- 1、V+和 V-兩端的電源線不要接反，否則會導致測量資料不準確
- 2、被測電源的地應該要和模組的 GND 介面連在一起，否則會導致測量資料不準確。



### 模組與 ZB2530-01 實驗板的硬體連接：

先將模組插接到轉接板的 J6 介面，然後將轉接板插接到 ZB2530-01 實驗板上，具體的插接方式如下圖所示：



## 模組輸出腳位說明：

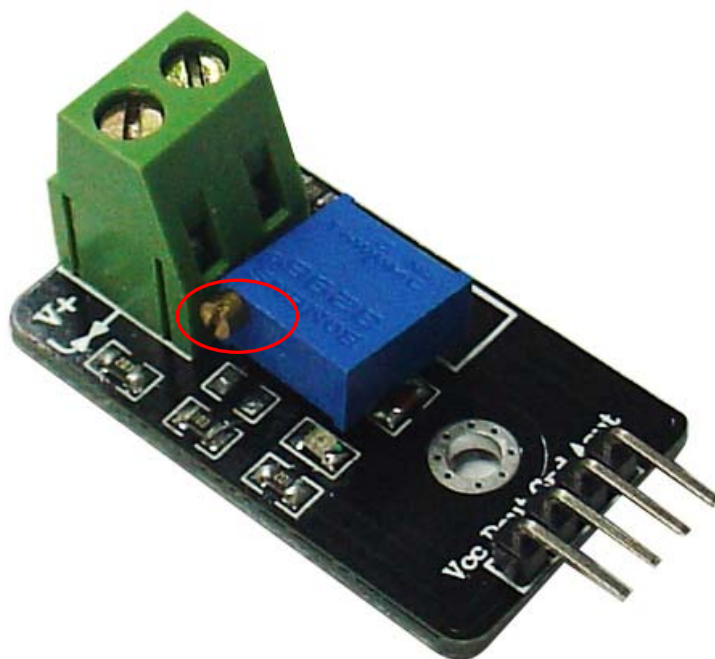
### Dout 輸出腳位

Dout 輸出腳位，是一個數位的高低電平輸出腳位，該腳位的輸出狀態與當前的被測電路的“實際電壓”，和設置的“監測電壓”有關係。

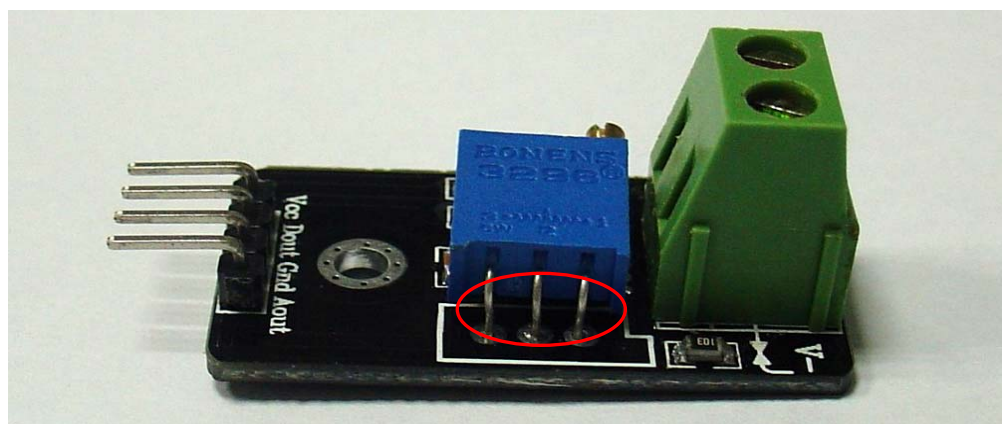
當被測電路“實際電壓”小於設置的“監測電壓”時，該腳位輸出低；

當被測電路“實際電壓”大於設置的“監測電壓”是，該腳位輸出高；

“監測電壓”的大小可以透過模組上的旋鈕來設置。



透過調整旋鈕，可以調整該控制器的中間腳位的“設置電壓”值，該“設置電壓”與“監測電壓”的換算關係如下所示：



$V$  監測電壓 =  $V$  設置電壓 \* ((  $R4+R5$  )/ $R4$ )

註： $R4$ ， $R5$  的電阻值可查看原理圖獲得： $R4 = 5K$ ， $R5 = 20K$ 。

### Aout 輸出腳位

Aout 輸出腳位是一個模擬輸出腳位，它的電壓值用於及時的反應，在  $V+$ 和  $V-$  介面所接的“實際電壓”。

$VAout = V$  實際電壓 \* ( $R4/(R4+R5)$ )

例如：當  $VCC$  以  $5V$  輸入時。

當 Aout 的值為  $3V$ ，那麼說明在  $V+$ 和  $V-$ 介面上所接電壓源電壓為  $15V$ ；

當 Aout 的值為  $2V$ ，那麼說明在  $V+$ 和  $V-$ 介面上所接電壓源電壓為  $10V$ 。

## 5、ZB2530-01 實驗板資料發送程式說明

明白了腳位的輸出含義之後，在程式開發上就可以明確目標。在有上位機的情況下，能夠即時的讀取到被測電路的採樣電壓值，應該比較有意義，所以，該程式，以獲取被測電路的採樣電壓值為重點開發。

程式分為兩大部分功能：

- 1、讀取 Dout 腳位的狀態，當該腳位為高時，點亮 LED 燈，用於採樣電壓超過設置電壓的警告標示，即此時處於過壓狀態；該腳位為低時，關閉 LED 燈，表示採樣電壓未超過設置電壓，即此時處於欠壓狀態。



2、讀取 Aout 腳位的電壓值，然後把它轉換為實際的電壓值，把此值從串列埠輸出。

下面是關鍵部分程式碼說明：

```
#include "hal_defs.h"

#include "hal_cc8051.h"

#include "hal_int.h"

#include "hal_mcu.h"

#include "hal_board.h"

#include "hal_uart.h"

#include "hal_led.h"

#include "hal_adc.h"

#include "hal_rf.h"

#include "basic_rf.h"

//-----

// CONSTANTS

//-----

// Application parameters

#define RF_CHANNEL          18      // 2.4 GHz RF channel

// BasicRF address definitions

#define PAN_ID              0x1111

#define SEND_ADDR          0x2222

#define RECV_ADDR          0x3333

#define APP_PAYLOAD_LENGTH  32

// Application states

#define IDLE                0
```

```
#define SEND_CMD                1

#define MAX_SEND_BUF_LEN    128
static uint8 pTxData[MAX_SEND_BUF_LEN]; //定義資料存儲緩衝區的大小
static basicRfCfg_t basicRfConfig;      //zigbee的配置資訊

void ADC_INIT(void)
{
    MCU_IO_PERIPHERAL(HAL_BOARD_IO_ADC_PORT, HAL_BOARD_IO_ADC_CH);
}

unsigned int ADC_GetValue(void)
{
    uint16 adcValue;//
    adcValue = adcSampleSingle(ADC_REF_AVDD, ADC_12_BIT, HAL_BOARD_IO_ADC_CH);
    return (adcValue & 0x3FFF);
}

void main(void)
{
    int32 val = 0;
    float tmpvalue = 0; /*using for valtage*/
    uint8 checksum = 0;    //check code
    // Config basicRF
    basicRfConfig.panId = PAN_ID;
    basicRfConfig.channel = RF_CHANNEL;
    basicRfConfig.ackRequest = TRUE;

    ADC_INIT();
    halBoardInit();
}
```

```
halUartInit(115200);//初始化串列埠0的串列傳輸速率為38400

basicRfConfig.myAddr = RECV_ADDR;

if (basicRfInit(&basicRfConfig) == FAILED);

basicRfReceiveOn();

#if BOARD_TYPE

P1SEL &= ~(1<<7);

P1DIR &= ~(1<<7);

#else

P1SEL &= ~(1<<6);

P1DIR &= ~(1<<6);

#endif

while(1)

{

pTxData[7]=(HAL_OUT_VAL())%10;

if(!pTxData[7])

{

halLedSet(1);//燈亮

}

else{

halLedClear(1);//燈滅

}

halMcuWaitMs(25);

val = ADC_GetValue();

if(val > 0)

{

tmpvalue = (float)val * 3.3/4096/2;//voltage

tmpvalue = tmpvalue * 5;

pTxData[4] = (char)tmpvalue /10;
```

```
pTxData[5] = (char)tmpvalue % 10;
pTxData[6] = (char)(tmpvalue * 10) % 10;

uint8 i = 0;
pTxData[0] = 0xFE;    //head
pTxData[1] = 0xFF;
pTxData[2] = 0x0C;    //data_length
pTxData[3] = 0x11;    //待定data_type    according to 《Table of Communication Data
```

Format》

```
for( i = 8; i < 14; i++)
    pTxData[i] = 0;

for(i = 2; i < 14; i ++ )
    checksum += pTxData[i];
checksum = (~checksum + 1) & 0xFF;
pTxData[14] = checksum;
pTxData[15] = 0xFC;
pTxData[16] = 0xFD;

basicRfSendPacket(RECV_ADDR, pTxData,17);
halUartWrite(pTxData,17);
checksum = 0;
halLedToggle(1);
halMcuWaitMs(100);
}
}
}
```

## 6、Android 平台核心程式說明

```
package com.example;

import jni.Linuxc;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.os.Message;
import android.util.Log;
import android.widget.EditText;

public class MainActivity extends Activity {

    static String myDate=null,datamygad,date_last="";
    char datetou[],date[];long i=0;
    int sum;
    EditText etguoya;
    MyHandler myHandler;

    @Override//3,9600 //5 38400, //7,115200
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        etguoya = (EditText) this.findViewById(R.id.et_cs_send);

        ShallDate.fd = Linuxc.openUart(3, 0);
        if(Linuxc.setUart(ShallDate.fd, 7, 0,0)==0)
```

```
        {  
            Log.i("TAG", "Uart Set Fault!!!");  
        }  
        myHandler = new MyHandler();  
        MyThread m = new MyThread();  
new Thread(m).start();  
    }  
    @SuppressWarnings("HandlerLeak")  
    class MyHandler extends Handler {  
public MyHandler() {  
    }  
  
    @SuppressWarnings("HandlerLeak")  
        public MyHandler(Looper L) {  
            super(L);  
        }  
  
        // 子類必須重寫此方法,接受資料  
        @Override  
        public void handleMessage(Message msg) {  
            // TODO Auto-generated method stub  
            Log.d("MyHandler", "handleMessage.....");  
            super.handleMessage(msg);  
            // 此處可以更新UI  
            Bundle b = msg.getData();  
            String thedate = b.getString("color");  
            MainActivity.this.etguoya.setText(thedate);  
        }  
    }  
}
```

```

class MyThread implements Runnable {
public void run() {
    while(true){
        datamygad = Linuxc.receiveMsgUartHex(ShallDate.fd);
        if(datamygad==null){
            continue;
        }
        datetou = datamygad.toCharArray();
        if((datetou[0]==0xFE)&&(datetou[1]==0xFF)){
            myDate += datamygad;
            if(myDate.length()<17){
                while(true){
                    datamygad =
Linuxc.receiveMsgUartHex(ShallDate.fd);
                    if(datamygad==null){
                        continue;
                    }
                    myDate += datamygad;
                    if(myDate.length()>16){
                        break;
                    }
                }
            }
        }else{
            Log.d("dddd.....", "起始幀錯誤");
            myDate = "";
        }

        if(myDate.length()==17){

```

```
Long.toString(myDate.length()));
```

```
date_last += "檢測值：";
```

```
Log.d("thread1.....", myDate);
```

```
Log.d("aaaaaaaaaaaaaaaa",
```

```
Log.d("aaaaaaaaaaaaaaaa", Long.toString(i++));
```

```
date = myDate.toCharArray();
```

```
if((date[15]==0xFC)&&(date[16]==0xFD)){
```

```
    for(char m = 2;m < 14;m++){
```

```
        sum += date[m];
```

```
    }
```

```
    sum = (~sum + 1) & 0xff;
```

```
if((date[14]==sum)){//&&(date[3]==0x0B)
```

```
    for(char m = 4;m < 8;m++){
```

```
        if(m==4)    date_last = "\n當前電壓：";
```

```
            if(m==6)    date_last += ".";
```

```
        if(m==7)    date_last += "\n當前狀態：";
```

```
            date_last += (date[m]+0);
```

```
            if(m==6)    date_last += "V";
```

```
    }
```

```
Message msg = new Message();
```

```
Bundle b = new Bundle();// 存放資料
```

```
Log.d("bbbb.....", date_last);
```

```
b.putString("color",date_last);
```

```
msg.setData(b);
```

```
myDate = ""; sum=0; date_last = "";
```

```
MainActivity.this.myHandler.sendMessage(msg); // 向Handler發送消息,更新UI
```



```
        }else{
            Log.d("bbb.....", "校驗錯誤");
            myDate = "";
        }
    }else{
        Log.d("cccc.....", "結束幀錯誤");
        myDate = "";
    }
}
}
}
}
}
```

## 7、Android 平台程式執行截圖

