

ICPDAS Java I/O Driver

Implement Web base industry control with Java Technique

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assume no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 1997-1999 by ICPDAS Inc. All rights are reserved.

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software **on a single machine.**
The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

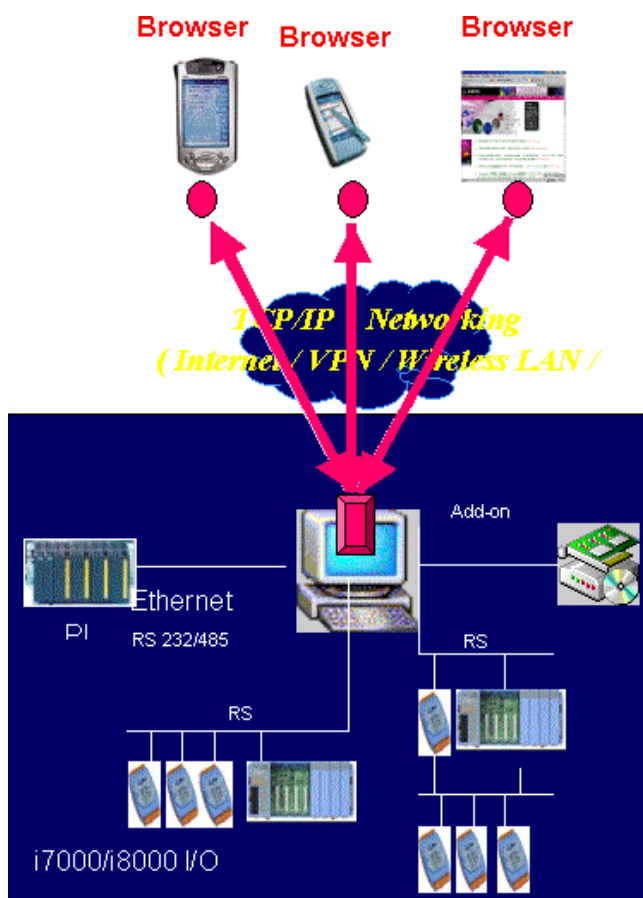


目錄

1.前言.....	3
2. 遠端控制架構.....	4
2.1 Client/Server 架構	4
2.2 Browser/Server 架構	4
3. 建構遠端控制環境.....	5
3.1 安裝 JIOD	5
3.2 建構 Servlet 執行環境	6
3.3 Servlet程式架構	7
3.4 撰寫第個一個 Servlet 程式	8
4. 遠端監控系統.....	9
4.1 遠端監控制模組介紹.....	9
4.2 Send Receive Command 網路版.....	10
4.3 WDT03 多功能監控卡介紹.....	12
4.4 乙太網遠端監控實例.....	12
5. 結論.....	20
參考資料：	20

1.前言

坐在咖啡廳裡享受著下午茶突然想起家中電燈好像沒有關或是門好像沒有上鎖，拿起隨身的 PDA 經由咖啡廳提供的無線上網服務，連上家中的電腦確認一下是否真的有什麼忘記關趕快把它關掉，聽起來好像科幻小說的情節可是感覺起來又那麼真實，拜科技進步所賜這一切都將真實陳現在您的面前。近年來網路技術快速發展就像洪水般衝擊著每個人的生活短短幾年間已經擴散到全世界，相信今天沒聽過網路這個名詞的人已經很少了吧，現在又將進步到無線網路行動運算的時代，由看不見的無線網路取代網路間實體的電纜連線，無線網路已經開始漸漸普目前已經在許多定點提供無線上網服務。未來的工業控制當然也不落人後的進入網路遠端控制時代，未來的控制人員不在侷限在控制中心中才能操作及監控儀表，取而代之的也許是一台能無線上網的 PDA 筆記型電腦，不受時間與空間限制到處都可以成為行動控制中心，本文以 ICPDAS 公司生產的 WDT03 多功能監控卡片示範如何一步一步打造屬於自己的行動控制中心。



2. 遠端控制架構

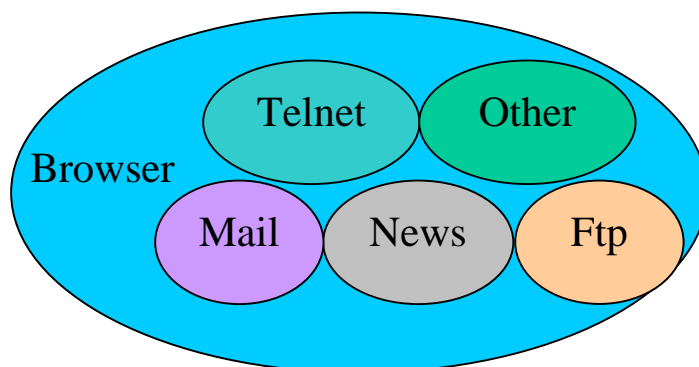
2.1 Client/Server 架構

由於人們對於資訊取得的慾望早在 Internet 普及前便已存在遠端存取的概念，人們可以坐在家中電腦前取得千里之外的訊息，傳統遠端存取均建立在所謂的主從式架構上(Client/Server)，服務端有不同的通訊協定需使用相對應的客戶端(Client)才能存取需要的資訊，例如讀信/寫信(Mail)需要使用信件軟體，閱讀新聞(News)需要安裝新聞軟體，上傳/下載檔案又需要使用檔案傳輸軟體(FTP)。雖然說現在要取得適合家中電腦使用的個戶端軟體並不困難，可是在網咖或受到限制的電腦要安裝及執行自己的軟體就會不太方便。無法任意安裝客戶端軟體的問題在行動上網的時代顯得更為嚴重，因為可能會找不到適合自己行動裝置的客戶端軟體，要是能夠使用單一通訊協定來存取各種不同的服務就太好了，可是前提是要找到一種夠普及到各種平台且功能強大的客戶端，Browser/Server 架構就在這種需求下誕生了。

2.2 Browser/Server 架構

瀏覽器已經普遍內建於各種不同的系統與平台上，瀏覽器使用的超文本(http)通訊協定功能強大足以建構不同的服務，WebMail 便是 Browser/Server 架構成功的應用實例，使用 WebMail 服務只需要在瀏覽器中便可以完成讀信及寫信，不需要再安裝讀信軟體也不需做複雜設定就可以在任一有瀏覽器的裝置中收發 Email。

Browser/Server 架構也非常適合應用在乙太網遠端工業控制的場合，系統操作人員可以使用熟悉的操作介面而不需要重新學習，也不會受限在裝有特定客戶端的電腦才能作監控，實現打造行動控制中心的夢想。

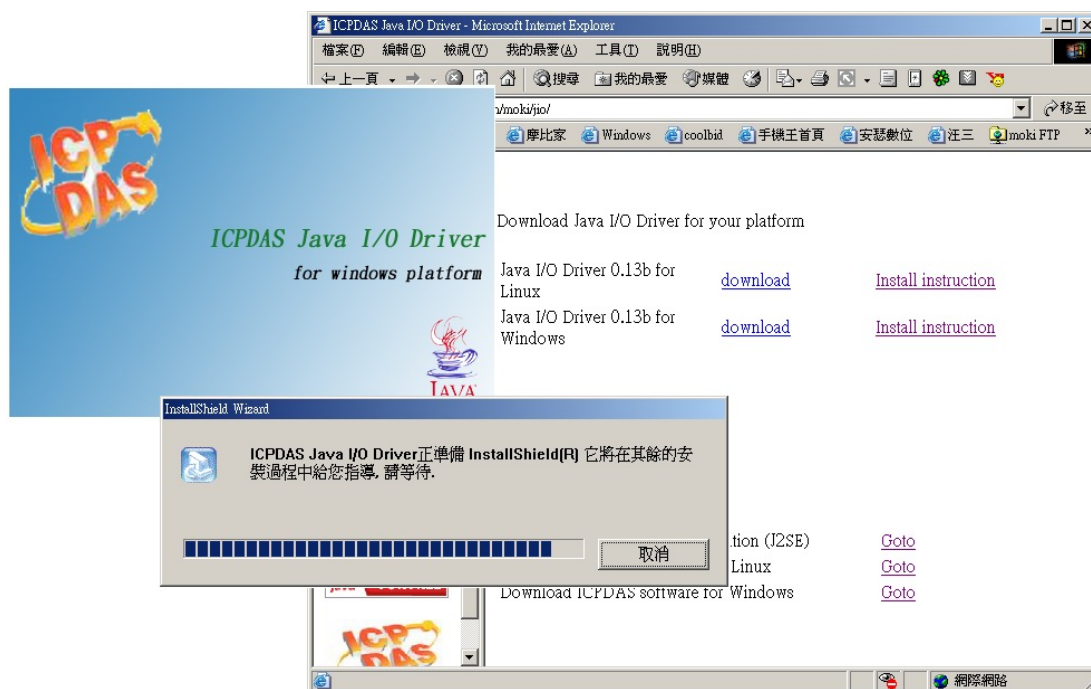


3. 建構遠端控制環境

本文將使用 Servlet 擴充伺服器端的功能來實現遠端工業控制，Servlet 是一個用 Java 寫的可以在是伺服器端執行的小程式，嚴格來說 Servlet 算是一種 CGI (Common Gateway Interface)都是用來擴充伺服器的功能，可是用 Servlet 比傳統的 CGI 更有效率且可以跨平台在不統的平台上使用。

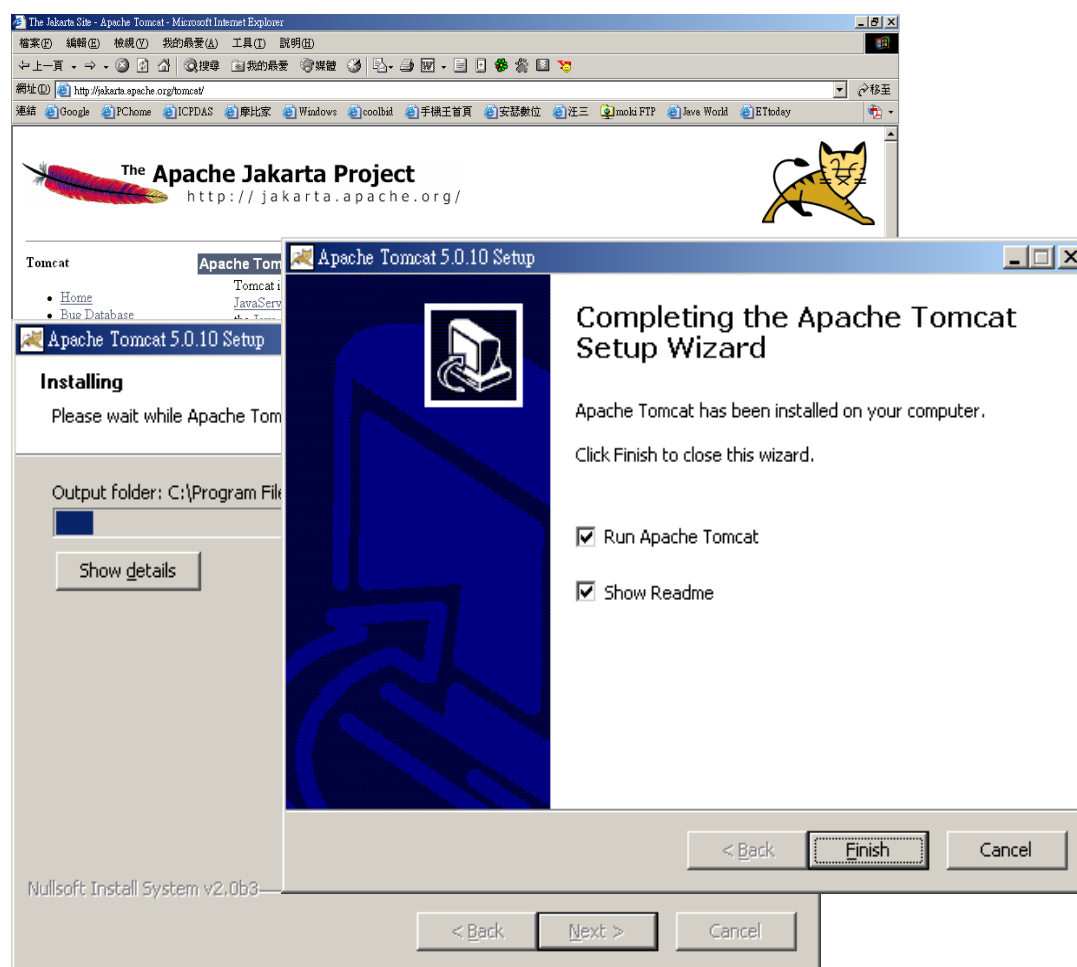
3.1 安裝 JIOD

Java I/O Driver (JIOD) 讓 Java 虛擬機器能支援安裝在電腦中的 I/O 介面卡及分散控制模組的支援套件，建立起 JAVA 與 I/O 卡之間的橋樑，JAVA 程式可以透過 JIOD 與週邊裝置作溝通，不受限於虛擬機器中，讓程式具有走出虛擬機器的能力，藉由 JAVA 天生完整的網路支援，設計出跨平台的遠端控制系統，提供完整的遠端網路控制解決方案。JIOD 目前已提供 Windows 及 Linux 平台讓 JAVA 套件可供免費下載，請至 Java I/O 下載頁下載適合您平台的 JIOD 並依說明完成安裝。



3.2 建構 Servlet 執行環境

建立 Servlet 執行環境有許多方法，本文以 Tomcat+IIS 來建立編譯及執行 Servlet 所需的環境，Tomcat 本身就是以 Java 寫成可以在不同的平台上執行而且還是免費的軟體，Tomcat 可以單獨使用也可以與其他的 Web Server 合作建構出適合自己需求的 Servlet 執行環境，最新版的 Tomcat 可以自 Tomcat 網站下載，按照內附說明完成安裝。



設定 CLASSPATH 環境變數

必須要正確的設定 CLASSPATH 環境變數才能讓 Java 找到我們所要使用的類別庫，以 JDK1.4.02 for windows 及 Tomcat4 為例，CLASSPATH 應設定成如下：

```
CLASSPATH=C:\j2sdk1.4.1_02\lib\dt.jar;C:\j2sdk1.4.1_02\lib\tools.jar;C:\j2sdk1.4.1_02\lib\icpdas.jar;C:\Program Files\Apache Tomcat 4.0\common\lib\servlet.jar;.
```

3.3 Servlet 程式架構

Servlet 可以處理來在表單的 Get 及 Post 請求，一個 Servlet 應該會有如下的架構

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SomeServlet extends HttpServlet {
    public void doDelete(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 DELETE 請求時會呼叫這個方法來處理請求
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 GET 請求時會呼叫這個方法來處理請求
    public void doHead(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 HTTP HEAD 請求時會呼叫這個方法來處理請求
    public void doOptions(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 OPTIONS 請求時會呼叫這個方法來處理請求
    public void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 POST 請求時會呼叫這個方法來處理請求
    public void doPut(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 PUT 請求時會呼叫這個方法來處理請求
    public void doTrace(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //當伺服器收到 TRACE 請求時會呼叫這個方法來處理請求
    public long getLastModified(HttpServletRequest req)
        throws ServletException, IOException { }
    //傳回請求物件被修改的時間,單位是千分之一秒自 1970 年一月一日 GMT 起算
    public void service(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException { }
    //接收來自客戶端的請求並把請求分發到這個類別定義中對應的方法(doXXX)
}
```

```
void doGet(HttpServletRequest request, HttpServletResponse response)
```

當伺服器收到來自表單的 Get 請求會呼叫 doGet 方法，並由 doGet 來處理這個請求。

```
void doPost(HttpServletRequest request, HttpServletResponse response)
```

當伺服器收到來自表單的 Post 請求會呼叫 doPost 方法，並由 doPost 來處理這個請求。

一個 Servlet 可以選擇實做 doGet 或 doPost 方法或兩種都實做出來，但最少要實做出其中一種，才算是個完整的 Servlet 程式。

3.4 撰寫第一個 Servlet 程式

平常我們在學習一種程式語言時的第一個程式不外乎是 HelloWorld，學習 Servlet 也不例外下面為 Servlet 版 HelloWorld 程式的原始碼，網頁中的表單會對 Servlet 發出一個 Get 請求，再由 Servlet 的 doGet 方法回應 HelloWorld 字串給瀏覽器

HelloWorld.java 列表

```
01 import java.io.*;
02 import javax.servlet.*;
03 import javax.servlet.http.*;
04
05 public class HelloWorld extends HttpServlet {
06
07     public void doGet(HttpServletRequest request, HttpServletResponse response)
08         throws IOException, ServletException
09     {
10         response.setContentType("text/html");
11         PrintWriter out = response.getWriter();
12         out.println("<html>");
13         out.println("<body>");
14         out.println("<head>");
15         out.println("<title>Hello World!</title>");
16         out.println("</head>");
17         out.println("<body>");
18         out.println("<h1>Hello World!</h1>");
19         out.println("</body>");
20         out.println("</html>");
21     }
22 }
```

01~03 引入撰寫 Servlet 所需的套件

07~22 處理 doGet 要求

11~21 產生 html 表頭

19~19 回應 HelloWorld 給瀏覽器

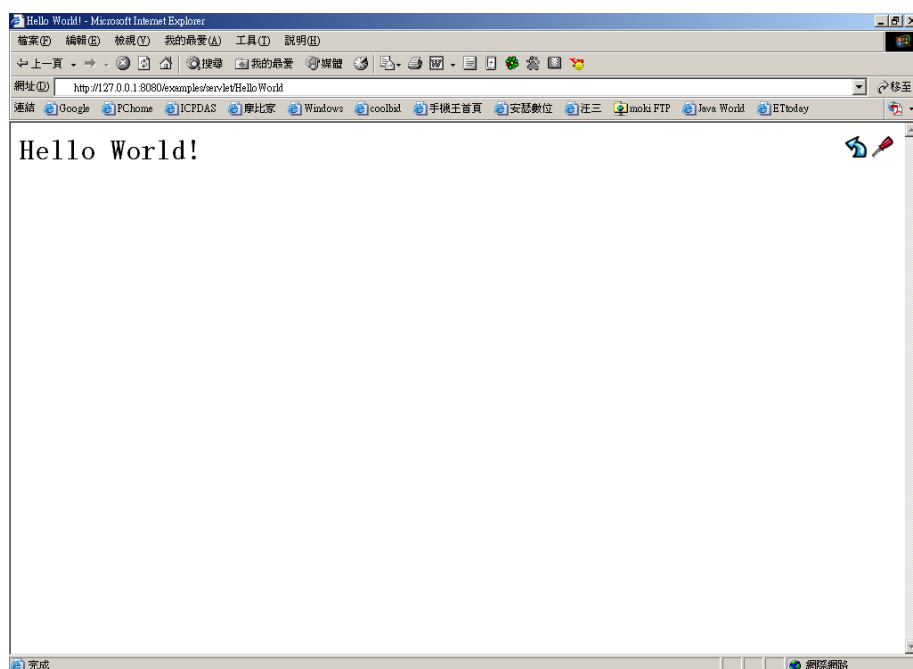
如果沒有錯誤的話應該可以順利編譯產生 class 檔將編譯好的 Class 檔放置在

Tomcat 安裝目錄\webapps\examples\WEB-INF\classes

執行結果

在瀏覽器中輸入

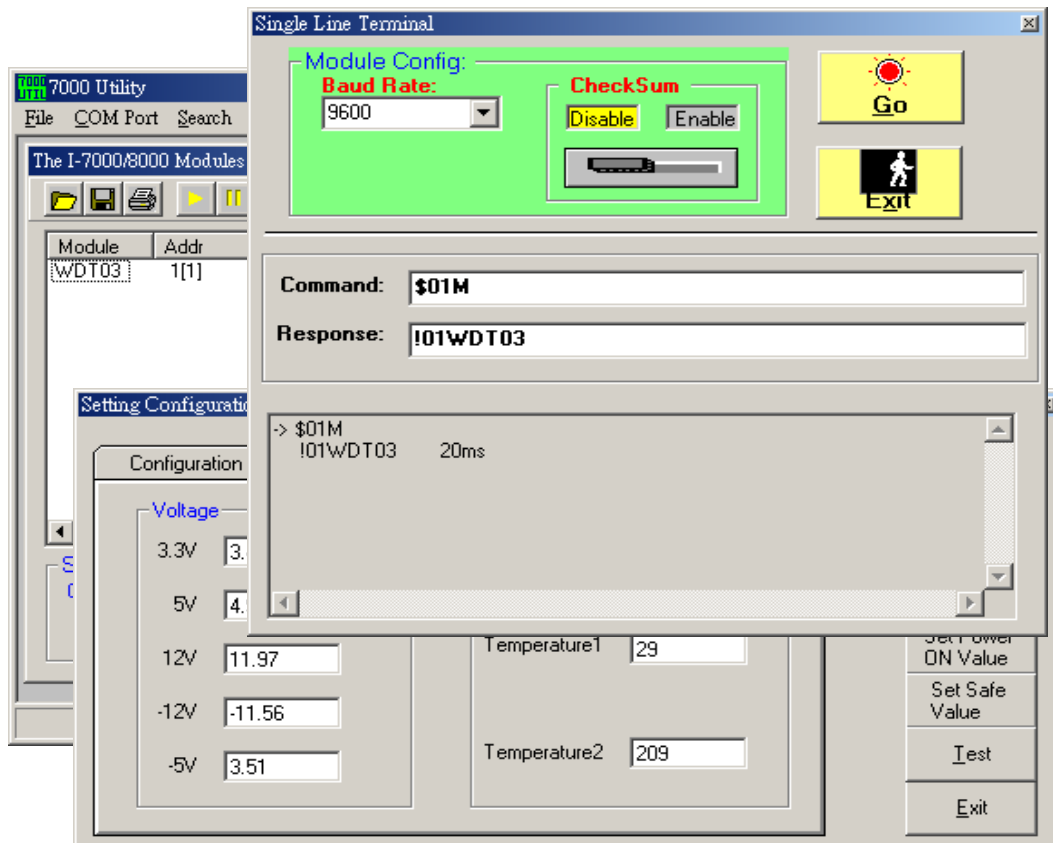
http://127.0.0.1:8080/examples/servlet/HelloWorld 並按下送出鍵應該會在瀏覽器中看到 Servlet 回應的 HelloWorld 字串



4. 遠端監控系統

4.1 遠端監控制模組介紹

工業控制實際應用時並非所有控制器都集中裝設在一起，相反的控制 器可能分散安裝在整個廠線路之中，i7k 系列模組便是屬於分散 控制用的模組，模組間以 485 網路相互連接幾乎可以無限擴充應用 範圍，485 網路具有使用容易，抗雜訊高及價格低廉等等優點，目 前已廣泛運用於工業控制場合，com.icpdas.comm 這個套件能支援 基於 485 網路的分散控制模組，讓 JAVA 能應用到分散控制場合。 ICPDAS 公司生產的分散控制模組使用 485 網路相互連接每個模組都 具有唯一的位址，主控端使用一問一答的方式(DCON Protocol)與模 組交換資料，例如\$01M 這個命令可以取得位址 1 模組的名稱。



(隨模組附的設定及診斷程式)

4.2 Send Receive Command 網路版

本節將要擴充送收命令的功能到乙太網路上，使用 Servlet 實做經由乙太網路對遠端模組交換資料

Servlet 端 SendReceiveCmd.java

```

01  import java.io.*;
02  import javax.servlet.*;
03  import javax.servlet.http.*;
04  import com.icpdas.comm.*;          //Import ICPDAS I/O Driver
05
06  public class SendReceiveCmd extends HttpServlet {
07
08      public void doGet(HttpServletRequest req, HttpServletResponse res)
09                          throws ServletException, IOException {
10
11          int rev;
12          res.setContentType("text/html");
13          PrintWriter out = res.getWriter();
14
15          out.println("<HTML>");
16          out.println("<HEAD><TITLE>Response from module</TITLE></HEAD>");
17          out.println("<BODY>");
18
19          Comm comm1 = new Comm();    //ICPDAS communication object
20          IoBuf i7kBuf = new IoBuf(); //control matrix
21          i7kBuf.szReceive = "";
22          rev = comm1.open(Integer.parseInt(req.getParameter("no")),
23                          Integer.parseInt(req.getParameter("baudrate")),comm1.DATABITS_8,
24                          comm1.PARITY_NONE,comm1.STOPBITS_1);
25          if(rev!=0) out.println("<BIG>Open Com port Error</BIG>");
26          else {

```

```

26         i7kBuf.szSend = req.getParameter("cmd");
27         i7kBuf.dwBuf[0] = Integer.parseInt(req.getParameter("no"));
28         //Serial Port no
29         if (Integer.parseInt(req.getParameter("checksum")) == 1)
30             i7kBuf.dwBuf[3] = 1;
31         else i7kBuf.dwBuf[3] = 0;           //check sum disable
32         i7kBuf.dwBuf[4] = Integer.parseInt(req.getParameter("timeout"));
33         //Timeout 100ms
34         rev = comm1.getSendReceiveCmd(i7kBuf);
35         //Send command to remote module
36         if (rev != 0) {
37             if (rev == 15 ) out.println("Time Out " + rev);
38             else out.println("Send Receive Command Error code : " +rev);
39         }
40         out.println(i7kBuf.szReceive);
41         comm1.close(1);
42     }
43     out.println("</BODY></HTML>");
44 }
45 }

```

Servlet 收到網頁以 Get 發法傳來的命令，將表單中的命令傳送到 485 網路上並等待回應將回應傳回到瀏覽器。

08~44 處理 Get 請求

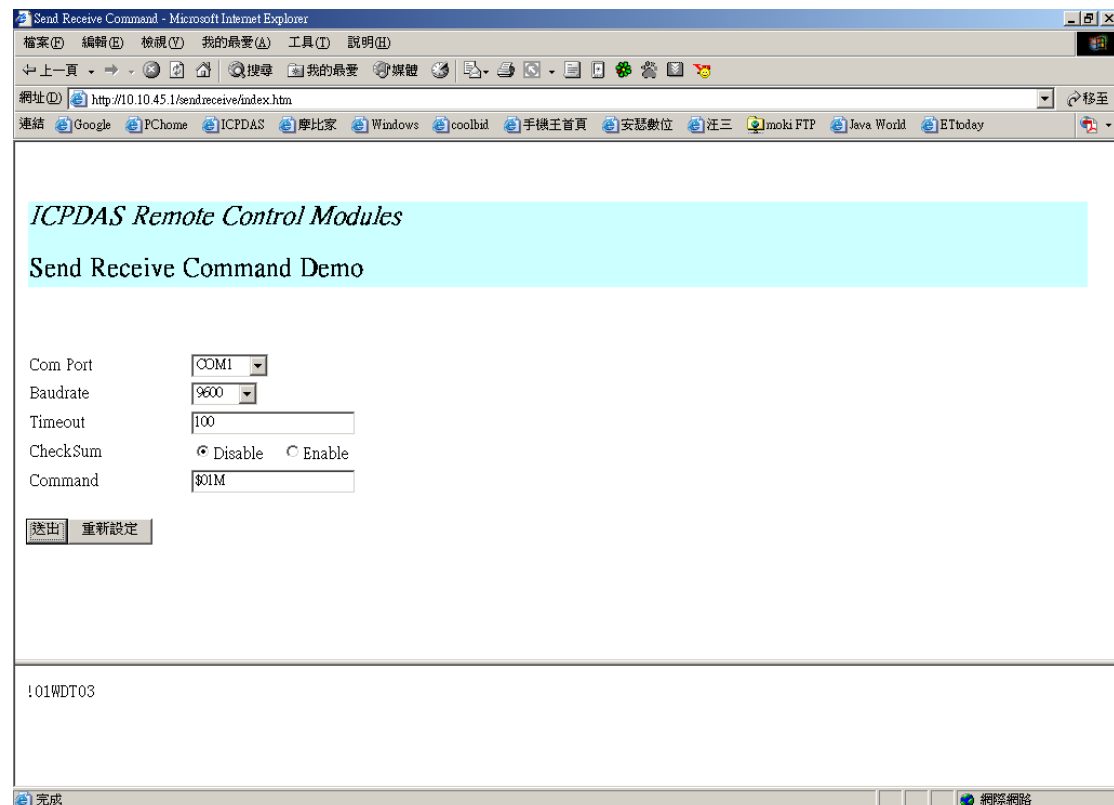
26~32 自請求中取得表單內容

34~34 將表單中命令輸出到序列埠並等待回應

36~40 將回應字串輸出到網頁中

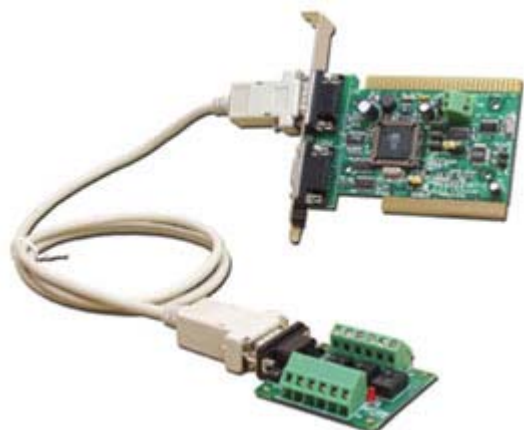
37~37 處理 timeout

執行結果



4.3 WDT03 多功能監控卡介紹

WDT03 是一張可以使用在 PCI 及 ISA 擴充槽的多功能監控卡片，採用與分散式模組相容的問答通訊協定及 RS232 電器規格作為通訊介面不需架設 485 網路便可以直接使用，很適合用來監控個人電腦及家電裝置。



WDT03 功能簡介

五組匯流排電壓：3.3V，5V，12V，-5V，-12V

三組風散轉速

三組溫度量測

三組 PWM 控制

三個數位輸入及三個數位輸出（可擴充成 Relay 輸出）

4.4 乙太網遠端監控實例

下面我們將要擴充 Send Receive Command 的功能來實做經由乙太網路做遠端監控，並增加一個參數來控制開關通訊埠增加 Servlet 效率，不需要在每次問答前重複做開關通訊埠的動作。網頁部分改用 applet 來呈現監控的資料讓網頁有更好的互動性，如果無法正確顯示 Applet 的話請至 Sun 網站下載最新版的 Applet plug-in 就可以正常顯示了。

Servlet 端 ModuleServlet.java

```
01 import java.io.*;
02 import javax.servlet.*;
03 import javax.servlet.http.*;
04 import com.icpdas.comm.*;          //Import Java I/O Driver
05
06 public class ModuleServlet extends HttpServlet {
07     int [] flag = new int[256];
08     Comm comm1 = new Comm();        //ICPDAS communication object
09     IoBuf i7kBuf = new IoBuf();     //control matrix
10     public void init() {
11         int rev = 0;
12         for (rev=1;rev<256;rev++) flag[rev]=0;
13     }
14     public void doGet(HttpServletRequest req, HttpServletResponse res)
15         throws ServletException, IOException {
16         int rev;
17         res.setContentType("text/plain");
18         PrintWriter out = res.getWriter();
19         if (req.getParameter("ioctl")!=null) {
20             if (req.getParameter("ioctl").equals("OPEN")) {
21                 if (flag[Integer.parseInt(req.getParameter("no"))]==0) {
22                     if ((rev = comm1.open(Integer.parseInt(req.getParameter("no")),
23                         Integer.parseInt(req.getParameter("baudrate")),comm1.DATABITS_8,
24                         comm1.PARITY_NONE,comm1.STOPBITS_1)) !=0) {
25                         out.println(rev);
26                     }
27                 }
28                 else {
29                     flag[Integer.parseInt(req.getParameter("no"))]=1;
30                 }
31             }
32             else out.println(20);
33         }
34         else if (req.getParameter("ioctl").equals("CLOSE")) {
35             comm1.close(Integer.parseInt(req.getParameter("no")));
36             flag[Integer.parseInt(req.getParameter("no"))]=0;
37         }
38         else if (req.getParameter("ioctl").equals("SR")) {
39             if ( flag[Integer.parseInt(req.getParameter("no"))]==1 ) {
40                 i7kBuf.szSend = req.getParameter("cmd");
41                 i7kBuf.dwBuf[0] = Integer.parseInt(req.getParameter("no"));
42                 //Serial Port no
43                 if (Integer.parseInt(req.getParameter("checksum")) == 1)
44                     i7kBuf.dwBuf[3] = 1;        //check sum enable
45                 else i7kBuf.dwBuf[3] = 0;        //check sum disable
46                 i7kBuf.dwBuf[4]=Integer.parseInt(req.getParameter("timeout"));
47                 //Timeout
48                 rev = comm1.getSendReceiveCmd(i7kBuf);
49                 //Send command to remote module
50                 if (rev != 0) out.println(rev);
51                 out.println(i7kBuf.szReceive);
52             }
53             else out.println("15");
54         }
55     }
56 }
```

10~13 初始化

14~55 處理 Get 請求

20~32 ioctl 等於 OPEN 開啓通訊埠

33~36 ioctl 等於 CLOSE 關閉通訊埠

37~53 ioctl 等於 SR 與 WDT03 交換資料

52~52 處理 timeout

Applet 程式 WdtDemo.java

```
01
02 import com.icpdas.comm.*; //include ICPDAS COMM Driver
03 import java.awt.*;
04 import java.applet.*; // import class Applet
05 import java.awt.event.*;
06 import java.net.*;
07 import java.io.*;
08
09 public class WdtDemo extends Applet implements ActionListener {
10
11     public static int rev,doVal;
12     public static boolean isrunning=false;
13     public static String hosts;
14     public static Thread pulling;
15     public static Label statusLabel = new Label();
16     public static Comm comm1;
17
18     public static Label v3ValueLabel = new Label();
19     public static Label v5ValueLabel = new Label();
20     public static Label v12ValueLabel = new Label();
21     public static Label vn5ValueLabel = new Label();
22     public static Label vn12ValueLabel = new Label();
23     public static Label t1ValueLabel = new Label();
24     public static Label t2ValueLabel = new Label();
25     public static Label t3ValueLabel = new Label();
26     public static Label f1ValueLabel = new Label();
27     public static Label f2ValueLabel = new Label();
28     public static Label f3ValueLabel = new Label();
29
30     public static Button[] outButton = new Button[4];
31     public static Button[] inButton = new Button[4];
32
33     public void paint( Graphics g ) {
34         super.paint(g); //更新畫面
35     }
36
37     public void init() { //人機介面的設定
38
39         int port0=0;
40         int buttonIndex;
41         Label[] bitLabel = new Label[4];
42         Label inLabel = new Label();
43         Label outLabel = new Label();
44         Label v3Label = new Label();
45         Label v5Label = new Label();
46         Label v12Label = new Label();
47         Label vn5Label = new Label();
48         Label vn12Label = new Label();
49         Label t1Label = new Label();
50         Label t2Label = new Label();
51         Label t3Label = new Label();
52         Label f1Label = new Label();
53         Label f2Label = new Label();
54         Label f3Label = new Label();
55
56         setLayout(new GridLayout(2,1));
57
58         Panel statusPanel = new Panel();
59         Panel monitorPanel = new Panel(new FlowLayout());
60         Panel voltagePanel = new Panel(new GridLayout(5, 2,0,0));
61         Panel temperaturePanel = new Panel(new GridLayout(5, 2,0,0));
62         Panel fanPanel = new Panel(new GridLayout(5, 2,0,0));
63         Panel dioPanel = new Panel(new GridLayout(3,4,0,0));
64
65         v3Label.setText("+3.3V");
66         v5Label.setText("+5V");
67         v12Label.setText("+12V");
68         vn5Label.setText("-5V");
69         vn12Label.setText("-12V");
70         t1Label.setText("T 1");
71         t2Label.setText("T 2");
72         t3Label.setText("T 3");
73         f1Label.setText("Fan 1");
74         f2Label.setText("Fan 2");
```

```

75     f3Label.setText("Fan 3");
76
77     v3ValueLabel.setText("0");
78     v5ValueLabel.setText("0");
79     v12ValueLabel.setText("0");
80     vn5ValueLabel.setText("0");
81     vn12ValueLabel.setText("0");
82     t1ValueLabel.setText("0");
83     t2ValueLabel.setText("0");
84     t3ValueLabel.setText("0");
85     f1ValueLabel.setText("0");
86     f2ValueLabel.setText("0");
87     f3ValueLabel.setText("0");
88
89
90     voltagePanel.add(v3Label);
91     voltagePanel.add(v3ValueLabel);
92     voltagePanel.add(v5Label);
93     voltagePanel.add(v5ValueLabel);
94     voltagePanel.add(v12Label);
95     voltagePanel.add(v12ValueLabel);
96     voltagePanel.add(vn5Label);
97     voltagePanel.add(vn5ValueLabel);
98     voltagePanel.add(vn12Label);
99     voltagePanel.add(vn12ValueLabel);
100    voltagePanel.setSize(100, 150);
101    monitorPanel.add(voltagePanel);
102
103    temperaturePanel.add(t1Label);
104    temperaturePanel.add(t1ValueLabel);
105    temperaturePanel.add(t2Label);
106    temperaturePanel.add(t2ValueLabel);
107    temperaturePanel.add(t3Label);
108    temperaturePanel.add(t3ValueLabel);
109    monitorPanel.add(temperaturePanel);
110
111    fanPanel.add(f1Label);
112    fanPanel.add(f1ValueLabel);
113    fanPanel.add(f2Label);
114    fanPanel.add(f2ValueLabel);
115    fanPanel.add(f3Label);
116    fanPanel.add(f3ValueLabel);
117    monitorPanel.add(fanPanel);
118
119    add(monitorPanel);
120
121    bitLabel[3] = new Label();
122    bitLabel[3].setText("Bits   :   ");
123    dioPanel.add(bitLabel[3]);
124    for (buttonIndex = 2;buttonIndex >= 0 ; buttonIndex-- ){
125        bitLabel[buttonIndex] = new Label();
126        bitLabel[buttonIndex].setText("           "+Integer.toString(buttonIndex)+"
127 ");
128        dioPanel.add(bitLabel[buttonIndex]);
129    }
130
131    inLabel.setText("Input :");
132    dioPanel.add(inLabel);
133    for (buttonIndex = 2;buttonIndex >= 0 ; buttonIndex-- ){
134        inButton[buttonIndex] = new Button();
135        inButton[buttonIndex].setLabel("OFF");
136        inButton[buttonIndex].setEnabled(false);
137        inButton[buttonIndex].setActionCommand("Output"+buttonIndex);
138        inButton[buttonIndex].addActionListener(this);
139        dioPanel.add(inButton[buttonIndex]);
140    }
141
142    outLabel.setText("Output:");
143    dioPanel.add(outLabel);
144    for (buttonIndex = 2;buttonIndex >= 0 ; buttonIndex--){
145        outButton[buttonIndex] = new Button();
146        outButton[buttonIndex].setLabel("OFF");
147        outButton[buttonIndex].setEnabled(true);
148        outButton[buttonIndex].setActionCommand("Output"+buttonIndex);
149        outButton[buttonIndex].addActionListener(this);
150        dioPanel.add(outButton[buttonIndex]);

```

```

151     }
152
153     add(dioPanel);
154     hosts = getParameter("baseip");
155     statusPanel.add(statusLabel);
156
157
158     }
159
160     /* menu and button message listener */
161     public void actionPerformed(ActionEvent e) {
162         int outVal = 0,bitIndex=0;
163         String cmd = e.getActionCommand();
164
165         if((cmd.substring(0,6)).equals("Output")){           //is output button been
166     press
167             bitIndex = Integer.parseInt(cmd.substring(6));    //which bit been press
168             if (outButton[bitIndex].getLabel().equals("OFF"))
169                 outButton[bitIndex].setLabel("ON");
170             else outButton[bitIndex].setLabel("OFF");
171             for(bitIndex = 0; bitIndex<3;bitIndex++){
172                 if (outButton[bitIndex].getLabel().equals("ON")) outVal = outVal + (int)
173     Math.pow(2,bitIndex);
174             }
175             doVal = outVal;
176         }
177     }
178
179
180     public void start() {
181         String line1="";
182         try {
183             URL url = new URL(hosts +
184     "no=1&baudrate=9600&timeout=100&checksum=0&cmd=$01M&iocctl=OPEN");
185             URLConnection con = url.openConnection();
186             con.setUseCaches(false);
187             InputStream in = con.getInputStream();
188             BufferedReader d = new BufferedReader(new InputStreamReader(in));
189             line1 = d.readLine();
190             if (line1 != null) statusLabel.setText("OPEN Error Code : " + line1);
191             else {
192                 statusLabel.setText("OPEN Success");
193                 isrunning = true;
194                 pulling=new pullthread(this);
195                 pulling.start();
196             }
197         }catch (Exception e) {statusLabel.setText(e.toString());}
198     }
199
200
201
202     public void stop() {
203         if (isrunning == true) {
204             isrunning = false;
205             try{
206                 pulling.join(); //wait thread terminal before end program
207             }catch(InterruptedException ie){}
208         }
209         try{
210             URL url = new URL(hosts +
211     "no=1&baudrate=9600&timeout=100&checksum=0&cmd=$01M&iocctl=CLOSE");
212             URLConnection con = url.openConnection();
213             con.setUseCaches(false);
214             InputStream in = con.getInputStream();
215             BufferedReader d = new BufferedReader(new InputStreamReader(in));
216             statusLabel.setText("CLOSED");
217         }catch (Exception e) {statusLabel.setText(e.toString());}
218     }
219 }
220
221 class pullthread extends Thread{
222     WdtDemo Wdt;
223     int rev,oldDi = 0,oldDo = 0;
224     int indexDI,nowDi,nowDo;
225     public pullthread(WdtDemo Wdt){}
226

```



```

227 public void run(){
228     Wdt.statusLabel.setText("Running");
229     double v3Value,v5Value,v12Value,vn5Value,vn12Value,
230         t1Value,t2Value,t3Value,f1Value,f2Value,f3Value;
231     while(Wdt.isrunning){
232         String line1="";
233         try {
234             Wdt.statusLabel.setText("");
235             URL url = new URL(Wdt.hosts +
236 "no=1&baudrate=9600&timeout=100&checksum=0&cmd=~018&ioc1=SR");
237             URLConnection con = url.openConnection();
238             con.setUseCaches(false);
239             InputStream in = con.getInputStream();
240             BufferedReader d = new BufferedReader(new InputStreamReader(in));
241             line1 = d.readLine();
242             if (line1==null) Wdt.statusLabel.setText("Get NULL !!");
243             else if (line1.equals("15")) Wdt.statusLabel.setText("Timeout !!");
244             else {
245                 try {
246                     v3Value = Integer.decode("0x"+line1.substring(3,5)).doubleValue() ;
247                     v3Value = (4.096 / 256) * v3Value;
248                     Wdt.v3ValueLabel.setText(Double.toString(v3Value).substring(0,4));
249                 }catch (Exception e) {Wdt.v3ValueLabel.setText("0.0");}
250
251                 try {
252                     v5Value = Integer.decode("0x"+line1.substring(6,8)).doubleValue();
253                     v5Value = (4.096 / 256) * v5Value;
254                     v5Value = v5Value / 0.6 ;
255                     Wdt.v5ValueLabel.setText(Double.toString(v5Value).substring(0,4));
256                 }catch (Exception e) {Wdt.v5ValueLabel.setText("0.0");}
257
258                 try {
259                     v12Value = Integer.decode("0x"+line1.substring(9,11)).doubleValue();
260                     v12Value = (4.096 / 256) * v12Value;
261                     v12Value = v12Value / 0.2232;
262                     Wdt.v12ValueLabel.setText(Double.toString(v12Value).substring(0,5));
263                 }catch (Exception e) {Wdt.v12ValueLabel.setText("0.0");}
264
265                 try {
266                     vn12Value =
267 Integer.decode("0x"+line1.substring(12,14)).doubleValue();
268                     vn12Value = (4.096 / 256) * vn12Value;
269                     vn12Value = (vn12Value - 3.6 * 0.866) / (1 - 0.866);
270
271 Wdt.vn12ValueLabel.setText(Double.toString(vn12Value).substring(0,6));
272                 }catch (Exception e) {Wdt.vn12ValueLabel.setText("0.0");}
273
274                 try {
275                     vn5Value =
276 Integer.decode("0x"+line1.substring(15,17)).doubleValue();
277                     vn5Value = (4.096 / 256) * vn5Value;
278                     vn5Value = (vn5Value - 3.6 * 0.6556) / (1 - 0.6556);
279                     Wdt.vn5ValueLabel.setText(Double.toString(vn5Value).substring(0,5));
280                 }catch (Exception e) {Wdt.vn5ValueLabel.setText("0.0");}
281
282                 try {
283                     t1Value = Integer.decode("0x"+line1.substring(18,20)).intValue();
284                     Wdt.t1ValueLabel.setText(Double.toString(t1Value));
285
286                     t2Value = Integer.decode("0x"+line1.substring(21,23)).intValue();
287                     Wdt.t2ValueLabel.setText(Double.toString(t2Value));
288
289                     t3Value = Integer.decode("0x"+line1.substring(24,26)).intValue();
290                     Wdt.t3ValueLabel.setText(Double.toString(t3Value));
291
292                     f1Value = Integer.decode("0x"+line1.substring(27,29)).intValue();
293                     if ( f1Value == 255 ) f1Value = 0;
294                     else if ( f1Value != 0 ) f1Value = 1350000 / (2 * f1Value);
295                     Wdt.f1ValueLabel.setText(Double.toString(f1Value));
296
297                     f2Value = Integer.decode("0x"+line1.substring(30,32)).intValue();
298                     if ( f2Value == 255 ) f2Value = 0;
299                     else if ( f2Value != 0 ) f2Value = 1350000 / (2 * f2Value);
300                     Wdt.f2ValueLabel.setText(Double.toString(f2Value));
301
302                     f3Value = Integer.decode("0x"+line1.substring(33,35)).intValue();

```

```

303         if ( f3Value == 255 ) f3Value = 0;
304         else if ( f3Value != 0 ) f3Value = 1350000 / ( 2 * f3Value);
305         Wdt.f3ValueLabel.setText(Double.toString(f3Value));
306
307     }catch (Exception e) {}
308     }
309
310
311     }catch (Exception e) {Wdt.statusLabel.setText(e.toString());}
312
313     if (oldDo != Wdt.doVal){ //Does Digital Output value have been change
314         oldDo = Wdt.doVal;
315         try {
316             Wdt.statusLabel.setText("");
317             URL url = new URL(Wdt.hosts +
318 "no=1&baudrate=9600&timeout=1000&checksum=0&
319 cmd=%2301000"+Integer.toString(oldDo)+"&ioc1=SR");
320             URLConnection con = url.openConnection();
321             con.setUseCaches(false);
322             InputStream in = con.getInputStream();
323             BufferedReader d = new BufferedReader(new InputStreamReader(in));
324             line1 = d.readLine();
325             if (line1!=null) Wdt.statusLabel.setText("Get NULL !!");
326             else if (line1.equals("15")) Wdt.statusLabel.setText("Timeout !!");
327         }catch (Exception e) {Wdt.statusLabel.setText(e.toString());}
328     }
329
330     try {
331         Wdt.statusLabel.setText("");
332         URL url = new URL(Wdt.hosts + "no=1&baudrate=9600&timeout=100&
333 checksum=0&cmd=$016&ioc1=SR");
334         URLConnection con = url.openConnection();
335         con.setUseCaches(false);
336         InputStream in = con.getInputStream();
337         BufferedReader d = new BufferedReader(new InputStreamReader(in));
338         line1 = d.readLine();
339         if (line1==null) Wdt.statusLabel.setText("Get NULL !!");
340         else if (line1.equals("15")) Wdt.statusLabel.setText("Timeout !!");
341         else {
342             try {
343                 nowDi = Integer.decode("0x"+line1.substring(3,5)).intValue();
344                 if(oldDi !=nowDi){
345                     oldDi = nowDi;
346                     for(indexDI=0;indexDI<3;indexDI++){
347                         if (nowDi%2 == 1) Wdt.inButton[indexDI].setLabel("ON");
348                         else Wdt.inButton[indexDI].setLabel("OFF");
349                         nowDi = nowDi / 2;
350                     }
351                 }
352                 nowDo = Integer.decode("0x"+line1.substring(1,3)).intValue();
353                 if(oldDo !=nowDo){
354                     for(indexDI=0;indexDI<3;indexDI++){
355                         if (nowDo%2 == 1) Wdt.outButton[indexDI].setLabel("ON");
356                         else Wdt.outButton[indexDI].setLabel("OFF");
357                         nowDo = nowDo / 2;
358                     }
359                 }
360             }
361         }catch (Exception e) {Wdt.statusLabel.setText(e.toString());}
362     }
363     }catch (Exception e) {Wdt.statusLabel.setText(e.toString());}
364     try{
365         Thread.sleep(500);
366     }catch(InterruptedExcepion ie){}
367 }
368 }
369 }
370
371

```

037~158 產生圖形介面

165~176 處理按鍵事件並轉換成數位輸出的值

- 235~241 使用~018 命令取得監控資料
- 242~305 分析監控資料並顯示
- 313~328 當數位輸出按鍵有變化時使用~01000D 來輸出 DO 值
- 330~363 使用\$016 命令取得數位輸入及數位輸出的讀回值並顯示

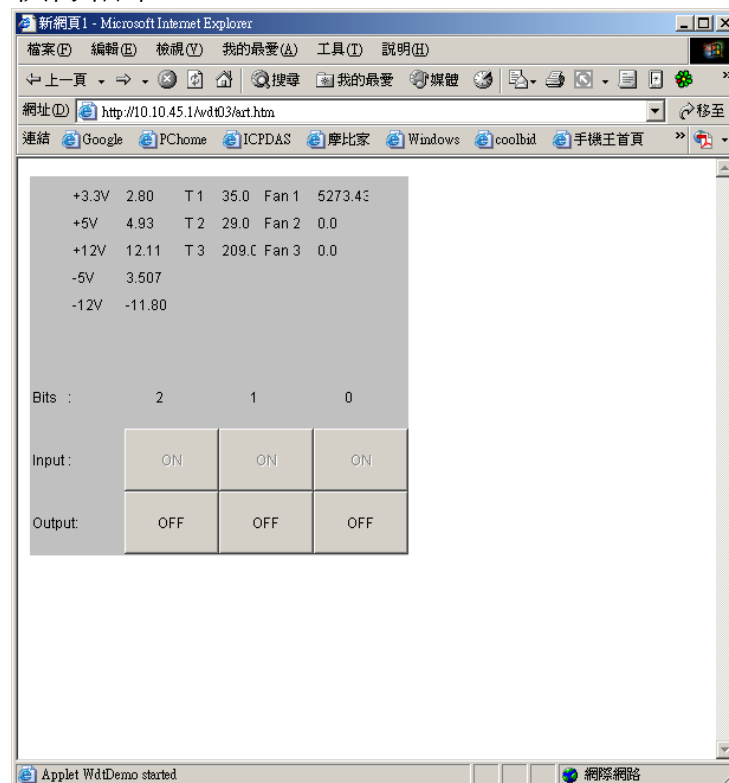
html 檔

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=big5">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>新網頁 1</title>
</head>
<body>
<p>
<applet width="320" height="320" code="WdtDemo.class">
  <param name="baseip" value="http://10.10.45.1:8080/examples/servlet/ModuleServlet?>The browser doesn't support
  java technique
</applet>
</p>
</body>
</html>

```

執行結果



Applet 執行時會建立一個執行緒持續的將遠端 WDT03 的監控結果顯示在本地的瀏覽器畫面。

5. 結論

雖然說遠端監控是那麼的方便可是遠端監控也意味著有可能會受到惡意入侵者的威脅，因此網路安全便成了一個重要的課題，使用 Tomcat+WebServer 的好處是可以選用自己喜歡的且安全性高的 WebServer 來加強遠端控制的安全性，也許網路技術越方便就會越不安全可是我們也不能因噎廢食而排斥網路為我們生活帶來的好處，您說是嗎？

參考資料：

- ICPDAS Website : <http://www.icpdas.com>
- ICPDAS JIOD : <http://w3.icpdas.com/moki/jio/>
- Apache Tomcat : <http://jakarta.apache.org/tomcat/>
- Sun Microsystems : <http://www.sun.com>
- Visual Basic 與分散式監控系統:使用 RS-232/485，范逸之，文魁圖書，2001.6
- Visual Basic 程式設計與圖形監控應用，陳正義、何坤鑫、程啓正，滄海書局，2001.2