

# ICPDAS Java I/O Driver

---

Implement 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.

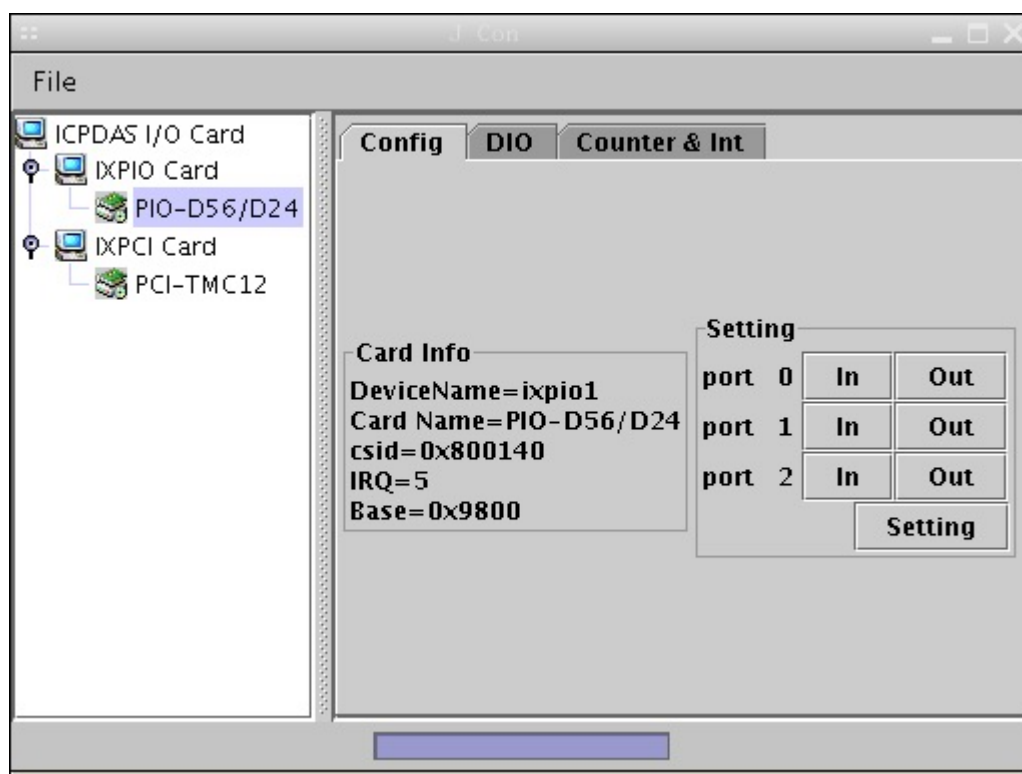


# 目錄

前言：	3
1. 安裝JDK	4
2. 環境設定	5
(1) PATH環境變數	6
(2) CLASSPATH環境變數	6
(3) LD_LIBRARY_PATH環境變數	6
3. 第一個程式 HelloWorld.java	7
4. I/O介面卡控制	7
5. 分散控制範例	12
結論：	16
參考資料：	16

## 前言：

資訊產業的進步快速，製造出許多功能強大且互不相容的硬體設備，然而不同廠商生產出的硬體中，或多或少都有些微的差異性，以至於軟體供應廠商須重覆投入大量的成本來開發，爲了幫不同平臺設計功能相似的程式，隨著平臺日新月異的多樣化，這種一在重覆投入的設計成本便顯得不合經濟效益，於是昇陽(Sun Microsystems)便研發推廣出 JAVA 技術，它不僅僅是一種語言，更完整的定義一套虛擬機器稱作 JVM，所有 JAVA 的應用程式都在 JVM 中來執行，無論你的平臺是 Windows 或 Linux，甚至 Power Mac，只要安裝適當的 JVM 便可以隨意執行，一但運用 JAVA 所寫的相關程式，甚至於硬體間的差異也能交給 JVM 來處理，相關的程式設計人員只須專攻於程式本身的功能來做設計，而不在須受平臺間的差異所困擾，達到只須寫一次而能到處執行的便利性。



(ICPDAS I/O 卡工具程式在 Linux 平台上執行的畫面)

跨平台的需求也限制了 Java 的應用，Java 虛擬機器並無法用於支援不同平台的特殊裝置列如 I/O 介面卡就不在支援的範圍之內，原始的虛擬機器只提供基本輸出輸入、運算網路等基本且共通的能力，不同平台特殊的需求須另行安裝支援套件。ICPDAS Java I/O Driver (JIOD) 讓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，不受限於虛擬機器的世界中，讓

程式具有走出虛擬機器的能力，利用 JAVA 天生完整的網路支援，便可設計出跨平台的遠端控制系統，提供完整的網路解決方案。JIOD 包含 com.icpdas.ixpio、com.icpdas.ixpci、com.icpdas.comm 三個套件目前已提供 Windows 及 Linux 平台讓 JAVA 套件可供免費下載，ixpio、ixpci 需配合 ICPDAS 公司的 I/O 卡使用，comm 套件則可用於 ICPDAS 的 i7k、i8k、i87k 分散控制模組或支援序列通訊的 PLC，請參照以下步驟一步一步建立您的 JAVA 程式。

## 1. 安裝 JDK

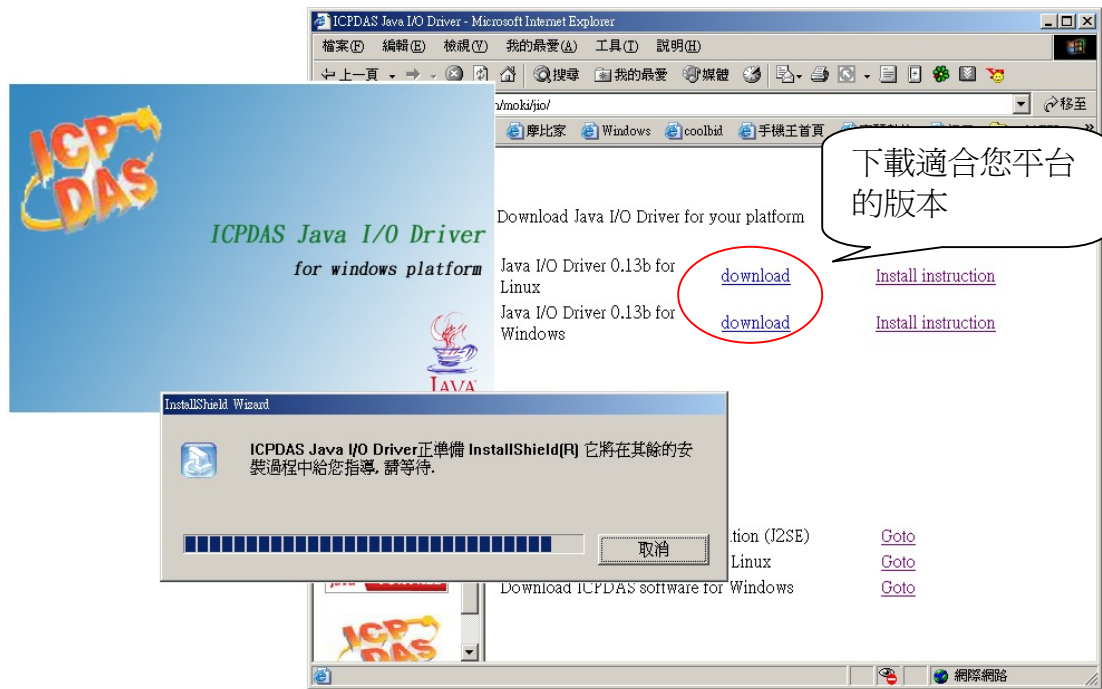
開發 JAVA 應用程式前需按裝編譯及執行 JAVA 程式所需的 JDK，本文所使用 JDK 是昇陽提供的 J2SE，J2SE 能免費自昇陽全球網站下載，請依您系統平台下載所需的 JDK。



安裝完成 JDK 後您的系統便擁有編譯 JAVA 程式的工具及執行 JAVA 程式所需的 JVM 了。

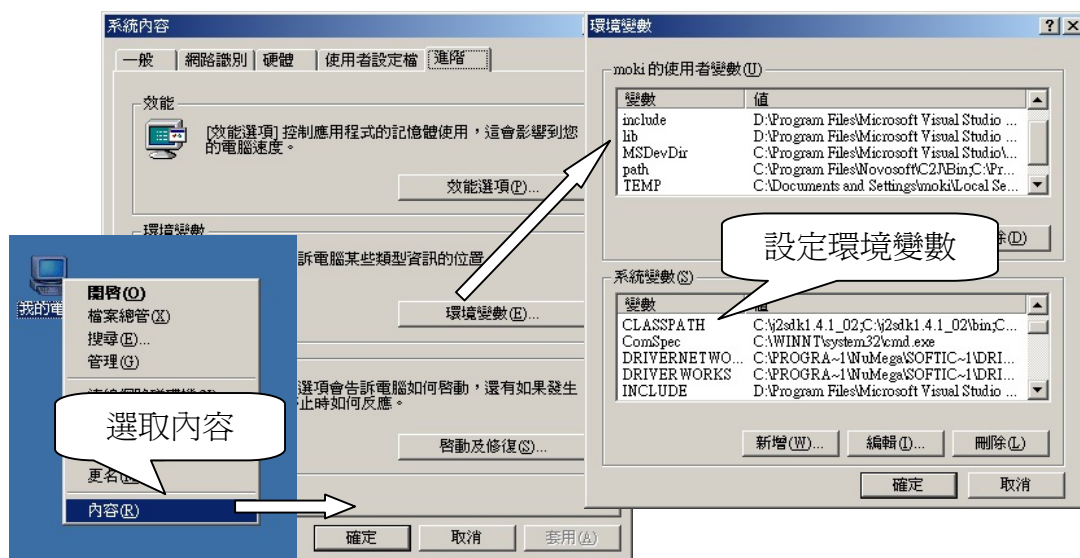
### 1 · 安裝 ICPDAS Java I/O Driver

在 Java I/O Driver 下載頁下載適合您平台的 JIOD 並依步驟完成安裝



## 2. 環境設定

JAVA 環境設定最爲一般初學 JAVA 的人所困擾，與高采烈下載了 JDK 並依說明安裝完成，卻連簡單的 HelloWorld 程式都無法如願完成，追究其原因大都是因爲環境設定不完整或不正確所造成，完整的 JAVA 環境設定包含執行路徑(PATH)、類別路徑(CLASSPATH)，Linux 平台須設定 LD\_LIBRARY\_PATH 環境變數才能正確使用 JIOD。Linux 平台可以加入環境設定到個人設定檔中例如 bash\_profile，Windows 平台需到<我的電腦>\<內容>\<進階>\<環境變數>中修改環境變數。



## (1)PATH 環境變數

PATH 指定執行檔的搜尋路徑幫助系統找到 JDK 所安裝的路徑，不正確得設定會使系統找不到 JDK 而出現如下錯誤訊息。以 JDK1.4.2 for windows 爲例，預設安裝到 C:\j2sdk1.4.1\_02 則 PATH 應設定成

```
PATH=C:\j2sdk1.4.1_02\Bin;
```

這樣的話就可以再任意目錄下執行 JDK 內的工具程式而不會出現找不到執行檔的錯誤訊息。

```
C:\>java
'java' 不是內部或外部指令、
可執行的程式或批次檔。

C:\>javac
'javac' 不是內部或外部指令、
可執行的程式或批次檔。
```

(找不到執行檔錯誤)

## (2)CLASSPATH 環境變數

JAVA 是物件導向的語言，程式中所使用到的類別大都是繼承自 JAVA，繼承的特性是讓 JAVA 能重複使用已經設計過的程式關鍵，例如 System.out.println 這個方法便是由 JAVA 所附類別庫提供，必須正確的設定 CLASSPATH 環境變數才能讓 JVM 找到我們所要使用的類別庫，以 JDK1.4.02 for windows 爲例，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;
```

## (3)LD\_LIBRARY\_PATH 環境變數

LD\_LIBRARY\_PATH 環境變數是 Linux 平台才需要的設定，Linux 需參考 LD\_LIBRARY\_PATH 才能找到 JIOD 所提供的動態連接庫，請在個人設定檔加入下列設定

```
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:
```

### 3. 第一個程式 HelloWorld.java

經過這一連串安裝與設定您是否已有點頭昏腦脹呢?先別急著睡著，接下來就要開始撰寫您的第一個 JAVA 程式 HelloWorld，請先使用文字編譯器完成如下的文字檔，並命名成 HelloWorld.java。注意大小寫需與 CLASS 的名稱一致。

```
//我的第一個 JAVA 程式
//檔案名稱：HelloWorld.java

public class HelloWorld{
    public static void main(String[] args){
        System.out.println("Hello World !!");
    }
}
```

( HelloWorld.java )

在命令列中輸入 `javac HelloWorld.java` 完成編譯，如果順利的話便能編譯完成您的第一個 JAVA 程式了，編譯完成後會輸出一個 JAVA 執行檔 `HelloWorld.class`，請輸入 `java HelloWorld` 來執行您的 JAVA 程式，如果還是出現錯誤訊息的話請檢查您的環境設定是否正確。

```
C:\pj\doc>
C:\pj\doc>
C:\pj\doc>
C:\pj\doc>
C:\pj\doc>
C:\pj\doc>javac HelloWorld.java

C:\pj\doc>java HelloWorld
Hello World !!

C:\pj\doc>
```

( 組譯及執行 HelloWorld.java )

### 4. I/O 介面卡控制

ICPDAS Java I/O Driver 的魅力就在於能讓 JAVA 程式控制電腦外真實世界的裝置，小到電燈開關、溫度的測量，大到機械手臂的控制都有可能。以 ICPDAS Piod24 控制卡為例，Piod24 為一具有 24 個可程式化的數位輸出輸入控制點，可控制及量測開關的狀態，完成硬體及驅動程式安裝後請執行 `<JIODDIR>\<examples>\List.java`，來尋找安裝在電腦中的 I/O 控制卡。

```
C:\DAQPro\JIOD\examples>java List
Ixp101 CSID = 800130 IRQ = 9 base = 9c00 NAME = PIOD48 NO= 1
Ixp102 CSID = 800830 IRQ = 9 base = 9800 NAME = PISO_P8R8 NO= 2
End of program

C:\DAQPro\JIOD\examples>
```

( 搜尋電腦上的卡片 )

由 List 程式我們可以得知 piod24 這張卡片的裝置名稱爲 ixpio1 等相關資訊，接下來撰寫我們的 I/O 卡控制程式時會用到這些資訊，請輸入如下的程式碼並命名成 Piod24Demo.java

```
/* Piod48Demo.java
 *
 * Piod48 數位輸出輸入範例
 *
 * v 0.0 2003.8.9 by moki matsushima
 *
 * Copyright (c) 2003 ICPDAS, Inc. All Rights Reserved.
 */

import java.io.*;      //System.in.read() 所用
import com.icpdas.cardio.ixpio.*; //引入 ICPDAS PIO Driver
import javax.swing.*;  //圖形介面所需
import java.awt.*;
import java.awt.event.*;

public class Piod48Demo extends Frame implements ActionListener {
    public static int fd;
    public static boolean isrunning=false;
    public static JLabel statusLabel;
    public static Ixp101 icpdasio1;
    public static Thread pulling;
    public JButton[] outButton = new JButton[8];
    public static JButton[] inButton = new JButton[8];

    public static void main(String[] args){
        byte a[] = new byte[100];
        Piod48Demo Piod48 = new Piod48Demo(args);
        Piod48.setVisible(true);
        Piod48.repaint();
    }

    public Piod48Demo(String[] args) { //畫面及 Piod48 初始化
        super("Piod48Demo 範例程式");
        MenuBar mb;
        Menu fileMenu;
        MenuItem exitItem;
        int rev;
        int port0=0;
        int buttonIndex;
        JPanel inPanel = new JPanel();
```



```

JPanel outPanel = new JPanel();
JPanel dioPanel = new JPanel();
JPanel statusPanel = new JPanel();
JPanel labelPanel = new JPanel();
JLabel[] bitLabel = new JLabel[9];

JLabel inLabel = new JLabel();
JLabel outLabel = new JLabel();
JLabel digitalLabel = new JLabel();
statusLabel = new JLabel();

addWindowListener(new CloseHandler(this));
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
setLocation(screenSize.width/4, screenSize.height/4);
setSize((screenSize.width/2),screenSize.height/4);

mb = new MenuBar();
fileMenu = new Menu("File");
exitItem = new MenuItem("Exit");
exitItem.addActionListener(this);
fileMenu.add(exitItem);
mb.add(fileMenu);
setMenuBar(mb);

GridBagLayout gridBag = new GridBagLayout();
setLayout(new BorderLayout());

labelPanel.setLayout(gridBag);
bitLabel[8] = new JLabel();
bitLabel[8].setText("Bits : ");
labelPanel.add(bitLabel[8]);
for (buttonIndex = 7;buttonIndex >= 0 ; buttonIndex-- ){
    bitLabel[buttonIndex] = new JLabel();
    bitLabel[buttonIndex].setText(" "+Integer.toString(buttonIndex)+" ");
    labelPanel.add(bitLabel[buttonIndex]);
}

inPanel.setLayout(gridBag);
inLabel.setText("Input :");
inPanel.add(inLabel);
for (buttonIndex = 7;buttonIndex >= 0 ; buttonIndex-- ){
    inButton[buttonIndex] = new JButton();
    inButton[buttonIndex].setText("OFF");
    inButton[buttonIndex].setEnabled(false);
    inButton[buttonIndex].setActionCommand("Output"+buttonIndex);
    inButton[buttonIndex].addActionListener(this);
    inPanel.add(inButton[buttonIndex]);
}
dioPanel.setLayout(new BorderLayout());
outPanel.setLayout(gridBag);
outLabel.setText("Output:");
outPanel.add(outLabel);
for (buttonIndex = 7;buttonIndex >= 0 ; buttonIndex--){
    outButton[buttonIndex] = new JButton();
    outButton[buttonIndex].setText("OFF");
    outButton[buttonIndex].setEnabled(true);
    outButton[buttonIndex].setActionCommand("Output"+buttonIndex);
    outButton[buttonIndex].addActionListener(this);
    outPanel.add(outButton[buttonIndex]);
}

```

```

dioPanel.add(inPanel,"North");
dioPanel.add(outPanel,"Center");
add(dioPanel,"Center");

statusPanel.add(statusLabel);
add(statusPanel,"South");
add(labelPanel,"North");

IxprioReg rdo = new IxprioReg();
icpdasio1=new Ixprio();
fd=icpdasio1.open("ixprio1"); // Open Piod48
if(fd== -1) statusLabel.setText("Open Piod48 Error Code:"+ fd);
else {
    isrunning = true;
    /* setup 8255-1 control word */
    /* PA as input port PB as output*/
    rev=icpdasio1.setPort(fd,"IXPIO_82551CW",0x90);
    pulling=new pullthread(this);
    pulling.start();
}
}

/* 選單及按鍵的訊息處理函數 */
public void actionPerformed(ActionEvent e) {
    int outVal = 0,bitIndex,rev;
    IxprioReg rdo = new IxprioReg();
    String cmd = e.getActionCommand();
    if (cmd.equals("Exit")) {
        if(isrunning == true){
            isrunning = false;
            try{
                pulling.join(); //等待執行緒停止後才停止程式
            }catch(InterruptedException ie){}
        }
        System.exit(1);
    }
    if((cmd.substring(0,6)).equals("Output")){ //當 Output 按鍵被壓下會執行
        int doValue = Integer.parseInt(cmd.substring(6));//取得按下哪一個位元
        if (outButton[doValue].getText() == "OFF") outButton[doValue].setText("ON");
        else outButton[doValue].setText("OFF");
        for(bitIndex = 0; bitIndex<8;bitIndex++){
            if (outButton[bitIndex].getText() == "ON") outVal = outVal + (int) Math.pow(2,bitIndex);
        }
        rdo.value = outVal;
        if((rev=icpdasio1.setReg(fd,"IXPIO_P1",rdo))!=0) { //輸出到 Port1
            statusLabel.setText("Write P1 Error Code:"+ rev);
        }
    }
}

class pullthread extends Thread{
    Piod48Demo Piod48;
    int port0,oldDi = 0;
    int indexDI;
    public pullthread(Piod48Demo Piod48){}
}

```

```

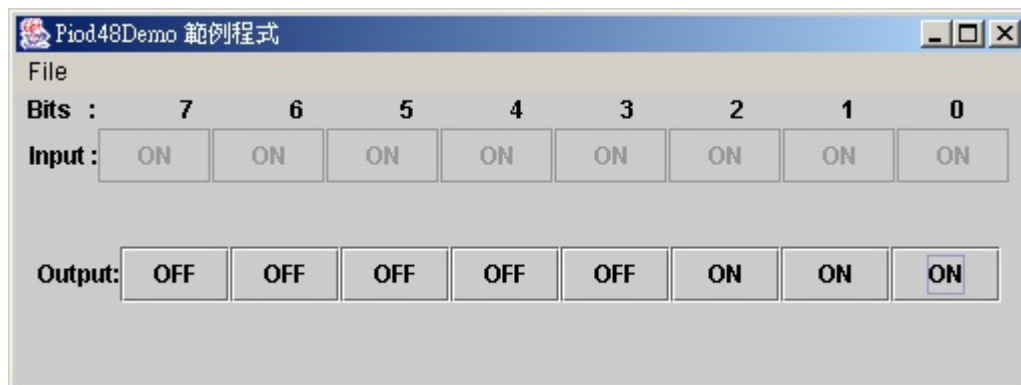
public void run(){
    while(Piod48.isrunning){
        Piod48.statusLabel.setText("");
        //讀取 Port0 的值
        if((port0=Piod48.icpdasio1.getReg(Piod48.fd,"IXPIO_82551PA"))== -1) {
            Piod48.statusLabel.setText("Read P0 Error Code:"+ port0);
        }
        else{
            if(oldDi != port0){ //Port 0 的值是否有變化
                oldDi = port0; //有變化才更新 DI 值
                for(indexDI=0;indexDI<8;indexDI++){
                    if (port0%2 == 1) Piod48.inButton[indexDI].setText(" ON ");
                    else Piod48.inButton[indexDI].setText("OFF");
                    port0 = port0 / 2;
                }
            }
        }
        try{
            Thread.sleep(100);
        }catch(InterruptedException ie){}
    }
}

class CloseHandler extends WindowAdapter {
    Piod48Demo Piod48;
    public CloseHandler(Piod48Demo Piod48) {}

    public void windowClosing(WindowEvent e) {
        if(Piod48.isrunning == true){
            Piod48.isrunning = false;
            try{
                Piod48.pulling.join(); //等待執行緒停止後才停止程式
            }catch(InterruptedException ie){}
        }
        System.exit(1);
    }
}

```

編譯後執行結果如下：



Pio24Demo 初始畫會將 PA 設定成輸入埠 PB 設定成輸出埠，程式中會建立一個執行緒來讀取 PA 輸入的值及數位輸出到 PB，將結果顯示到螢幕。

## 5. 分散控制範例

工業控制實際應用時並非所有控制器都集中裝設在一起，相反的控制器也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的優點，目前已廣泛運用於工業控制場合，com.icpdas.comm 這個 packages 能支援基於 485 網路的分散控制模組，讓 JAVA 的應用擴充到分散控制場合。安裝完 JIOD 在<JIODDIR>\<examples>\<comm> 目錄中附有許多分散控制範例，下列程式碼以 I7060 模組為例，經由 i7060 模組做數位輸出輸入的控制，請用文字編輯器輸入後存成 I7060Demo.java 並完成編譯並執行。

程式碼列表：

```
/* I7060Demo.java
 *
 * I7060 數位輸出輸入範例
 *
 * v 0.0 2003.8.9 by moki matsushima
 *
 * Copyright (c) 2003 ICPDAS, Inc. All Rights Reserved.
 */

import java.io.*;           //System.in.read() 所需要的套件
import com.icpdas.comm.*;  //引入 ICPDAS COMM Driver
import javax.swing.*;      //圖形介面所需
import java.awt.*;
import java.awt.event.*;

public class I7060Demo extends Frame implements ActionListener {
    public static int rev,doVal;
    public static boolean isrunning=false;
    public static JLabel statusLabel;
    public static Comm comm1;
    public static Thread pulling;
    public JButton[] outButton = new JButton[4];
    public static JButton[] inButton = new JButton[4];

    public static void main(String[] args){
        byte a[] = new byte[100];
        I7060Demo I7060 = new I7060Demo(args);
        I7060.setVisible(true);
        I7060.repaint();
    }

    public I7060Demo(String[] args) { //畫面及 I7060 初始化
        super("I7060 範例程式");
        MenuBar mb;
        Menu fileMenu;
        MenuItem exitItem;
        int port0=0;
```

```

int buttonIndex;
JPanel inPanel = new JPanel();
JPanel outPanel = new JPanel();
JPanel dioPanel = new JPanel();
JPanel statusPanel = new JPanel();
JPanel labelPanel = new JPanel();
JLabel[] bitLabel = new JLabel[5];

JLabel inLabel = new JLabel();
JLabel outLabel = new JLabel();
JLabel digitalLabel = new JLabel();
statusLabel = new JLabel();

addWindowListener(new CloseHandler(this));
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
setLocation(screenSize.width/4, screenSize.height/4);
setSize((screenSize.width/3),screenSize.height/4);

mb = new MenuBar();
fileMenu = new Menu("File");
exitItem = new MenuItem("Exit");
exitItem.addActionListener(this);
fileMenu.add(exitItem);
mb.add(fileMenu);
setMenuBar(mb);

GridBagLayout gridBag = new GridBagLayout();
setLayout(new BorderLayout());

labelPanel.setLayout(gridBag);
bitLabel[4] = new JLabel();
bitLabel[4].setText("Bits : ");
labelPanel.add(bitLabel[4]);
for (buttonIndex = 3;buttonIndex >= 0 ; buttonIndex-- ){
    bitLabel[buttonIndex] = new JLabel();
    bitLabel[buttonIndex].setText(" "+Integer.toString(buttonIndex)+" ");
    labelPanel.add(bitLabel[buttonIndex]);
}

inPanel.setLayout(gridBag);
inLabel.setText("Input :");
inPanel.add(inLabel);
for (buttonIndex = 3;buttonIndex >= 0 ; buttonIndex-- ){
    inButton[buttonIndex] = new JButton();
    inButton[buttonIndex].setText("OFF");
    inButton[buttonIndex].setEnabled(false);
    inButton[buttonIndex].setActionCommand("Output"+buttonIndex);
    inButton[buttonIndex].addActionListener(this);
    inPanel.add(inButton[buttonIndex]);
}
dioPanel.setLayout(new BorderLayout());
outPanel.setLayout(gridBag);
outLabel.setText("Output:");
outPanel.add(outLabel);
for (buttonIndex = 3;buttonIndex >= 0 ; buttonIndex--){
    outButton[buttonIndex] = new JButton();
    outButton[buttonIndex].setText("OFF");
    outButton[buttonIndex].setEnabled(true);
    outButton[buttonIndex].setActionCommand("Output"+buttonIndex);
    outButton[buttonIndex].addActionListener(this);
}

```

```

        outPanel.add(outButton[buttonIndex]);
    }
    dioPanel.add(inPanel,"North");
    dioPanel.add(outPanel,"Center");
    add(dioPanel,"Center");

    statusPanel.add(statusLabel);
    add(statusPanel,"South");
    add(labelPanel,"North");

    comm1 = new Comm(); //ICPDAS 通訊物件
    //open serial port
    rev =
comm1.open(1,9600,comm1.DATABITS_8,comm1.PARITY_NONE,comm1.STOPBITS_1);
    if(rev===-1) statusLabel.setText("Open Port Error Code:"+ rev);
    else {
        isrunning = true;
        pulling=new pullthread(this);
        pulling.start();
    }
}

/* 選單及按鍵的訊息處理函數 */
public void actionPerformed(ActionEvent e) {
    int outVal = 0,bitIndex;
    String cmd = e.getActionCommand();
    if (cmd.equals("Exit")) {
        if ( isrunning == true ){
            isrunning = false;
            try{
                pulling.join(); //等待執行緒停止後才停止程式
            }catch(InterruptedException ie){}
        }
        System.exit(1);
    }
    if((cmd.substring(0,6)).equals("Output")){ //當 Output 按鍵被壓下會執行
        int doValue = Integer.parseInt(cmd.substring(6)); //取得按下哪一個位元
        if (outButton[doValue].getText() == "OFF") outButton[doValue].setText("ON");
        else outButton[doValue].setText("OFF");
        for(bitIndex = 0; bitIndex<4;bitIndex++){
            if (outButton[bitIndex].getText() == "ON") outVal = outVal + (int) Math.pow(2,bitIndex);
        }
        doVal = outVal;
    }
}
}

class pullthread extends Thread{
    I7060Demo I7060;
    IoBuf i7kBuf = new IoBuf(); //控制矩陣
    int rev,oldDi = 0,oldDo = 0;
    int indexDI;
    public pullthread(I7060Demo I7060){
        i7kBuf.dwBuf[0] = 1; //通訊附編號
        i7kBuf.dwBuf[1] = 1; //模組位址
        i7kBuf.dwBuf[2] = 0x7060 ; //0x7060; //模組名稱
    }
}

```

```

i7kBuf.dwBuf[3] = 0;    //不要錯誤碼檢查
i7kBuf.dwBuf[4] = 10;   //Timeout 設為 100ms
i7kBuf.dwBuf[6] = 1;   //傳回除錯字串
}

public void run(){
    while(I7060.isrunning){
        I7060.statusLabel.setText("");

        //讀取 Digital Input 的值
        if ((rev = I7060.comm1.getDigitalIn(i7kBuf)) != 0) {
            I7060.statusLabel.setText("DigitalIn Error Code:"+ rev);
        }
        else{
            if(oldDi != i7kBuf.dwBuf[5]){    //Digital Input 的值是否有變化
                oldDi = i7kBuf.dwBuf[5]; //有變化才更新 DI 值
                for(indexDI=0;indexDI<4;indexDI++){
                    if (i7kBuf.dwBuf[5]%2 == 1) I7060.inButton[indexDI].setText(" ON ");
                    else I7060.inButton[indexDI].setText("OFF");
                    i7kBuf.dwBuf[5] = i7kBuf.dwBuf[5] / 2;
                }
            }
        }

        if (oldDo != I7060.doVal){    //Digital Output 的值是否有變化
            oldDo = I7060.doVal;    //有變化才更新 DO 值
            for(indexDI=0;indexDI<4;indexDI++){
                i7kBuf.dwBuf[5] = oldDo ;
                if((rev = I7060.comm1.setDigitalOut(i7kBuf)) != 0) {
                    I7060.statusLabel.setText("DigitalOut Error Code:"+ rev);
                }
            }
        }
        try{
            Thread.sleep(100);
        }catch(InterruptedException ie){}
    }
}

class CloseHandler extends WindowAdapter {
    I7060Demo I7060;
    public CloseHandler(I7060Demo I7060) {}

    public void windowClosing(WindowEvent e) {
        if (I7060.isrunning == true ) {
            I7060.isrunning = false;
            try{
                I7060.pulling.join(); //等待執行緒停止後才停止程式
            }catch(InterruptedException ie){}
        }
        System.exit(1);
    }
}
}

```

執行結果：



I7060Demo 開啓第一個通訊埠並操作 baudrate 9600，address 爲 1 的 i7060 模組，程式中產生一個執行緒來讀取 DI 值及輸出 DO 值將結果顯示於視窗中。

## 結論：

JAVA 並不是跨平台唯一的解決方案，微軟也提出 .NET 架構試圖提供跨平台的解決方案，至今卻都不若 JAVA 能這麼廣泛的運用於各種平台。不知曾幾何時 JAVA 這個名詞已經充斥在我們的生活中，瀏覽執行網頁中的 Java script 或 applet 新上市手機也大都會標榜支援 JAVA 能力，試想有一天您能坐在咖啡廳中拿著無線上網 PDA 連上廠線網頁得知工廠運作情形或遙控家中電器的未來生活，這就是 JAVA 的魔力。

## 參考資料：

ICPDAS Website : <http://www.icpdas.com>

ICPDAS JIOD : <http://w3.icpdas.com/moki/jio/>

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