

用 JAVA 小程序实现工业监控环境下 WEB 页面的实时更新

钟寒梅 张永华 董海鹰 西安交通大学电气工程学院(710049)

Abstract

In this paper, how to use Java Applet to implement data auto-refreshing on Web pages in industrial environment is discussed. I elaborate on how to program and I give emphasis on database objects ADO.

Keywords: ADO, Java Applet, thread

摘要

本文就如何利用 JAVA 的小程序实现工业监控环境下 WEB 页面的实时更新作了的讨论。具体介绍如何进行程序的编制,其中重点介绍了数据库对象 ADO。

关键词: ADO, JAVA 小程序, 线程

本文所依赖的开发环境是微软提供的 VJ++, 它界面友好, 功能完备, 是目前功能最强大的 Java 开发环境之一。和微软提供的其他集成编程工具一样, Visual J++ 也是通过项目(project)来管理其所开发的程序。

本文所建立的项目是 Applet on HTML 类型, 也就是嵌入网页的小应用程序。小应用程序就是可执行内容, 而 Java 的最根本作用就是对可执行内容执行与平台无关的访问。

Web 现在可以传送用 Java 程序写的可执行内容, 当与 Java 兼容的浏览器访问这些页面时, 它就可以自动下载和执行所有的嵌入式 Java 程序。这种模式的发展潜力是巨大的, 因为小程序可写成任何类型的程序。又由于这些程序是与平台无关的, 所以它们可以在安装了 Java 的任何计算机上运行。Web 有了它做保证后, 即呈现勃勃生机。

Java 改进了 Web 的不少缺点, 使它具有动态性, 真正的交互性。在变电站系统的实时监控问题上, 我们利用 Java 小程序实现从数据库中读数, 并实时刷新 Web 页面的功能。下面将具体介绍小程序是如何实现这一功能的。

1 Java 中常用的数据库对象

Java 中常用到的数据库对象有 Data Access (DAO)(数据库访问对象), Remote Data Objects (RDO)(远程数据对象), ActiveX Data Object (ADO) (ActiveX 数据对象)以及 JDBC。

DAO 和 RAO 出现较早, 这两种技术互相补充并被设计用于两种不同的程序设计情况。当需要与一个 Access 数据库相连接并从中访问数据时用 DAO。另一方面, RDO 用于与 ODBC (开放数据库连接)数据源相连接的情况。

用 DAO 与 RDO 连接数据库有很多相似之处, 实现起来也比较麻烦。所以在使用到它们的时候最好用数据连接向导。这样可以避免源程序的编制。

使用 ADO 进行连接编程简单, 程序容易编制。当然也可以用数据连接向导实现 ADO。ADO 的出现解决了 ODBC 的问题。ODBC 提供了一整套应用程序的接口 API, 我们的程序可以通过这个 API 系统来调用 ODBC 驱动器实现数据库处理。但是, ODBC API 相对来说比较复杂, 解决这种复杂性的问题就是 ADO 接口。下面通过具体例子说明连接的实现。

本例中是以 java.sql.Connection 类实现和数据库的连接。当创建一个 Connection 对象之后, 通过其 setConnectionString 方法设定连接字符串, 以 setCursorLocation 方法设定当前光标位置。当设定了上述两个 Connection 对象的属性之后, 就可以打开这个连接并建立了和数据库的联系。

代码如下:

```
import java.awt.*;  
import java.applet.*;  
import com.ms.wfc.app.*  
import com.ms.wfc.data.* //本类提供通过 ADO 对
```

数据库访问的途径

```
public class ManualADO extends Applet implements Runnable
{
    Connection con =new Connection();//创建一个新
连接
    String conString="DSN=test;DATABASE=mytest;"
    Con.setConnectionString(conString);//设置连接字符串
    Con.setCursorLocation (AdoEnums.CursorLocation.
CLIENT);
    Con.open(); //打开连接
    .....
}
```

建立连接时要用到连接字符串,本例中的连接字符串是提供了连接文件的名称,该文件包含了连接所需的所有信息,必须提前在 ODBC 中建立。通过控制面板→管理工具→数据源(ODBC)来建立。建立时提供数据库连接的几个关键字段:数据库的驱动程序,数据库的路径,用户名及密码。

上面说明了如何建立数据库的连接,数据库连接打开后,要对其中数据进行操作还需采用 ADO 的记录集对象(Recordset 对象)进行数据库的数据操作:

代码如下:

```
Recordset rs=new Recordset();//创建一个新记录集
.....
rs.setActiveConnection (con);//以 SQL 查询打开一个记录集
rs.setSource(queryString);
rs.setCursorType(AdoEnums.CursorType.STATIC);
rs.setCursorLocation(AdoEnums.CursorLocation.CLIENT);
rs.setLockType(AdoEnums.LockType.OPTIMISTIC);
rs.open();
```

记录集对象打开后就可以通过字段名获得数据库字段中保存的数据:

```
Field f=rs.getField(fieldName);
```

上面讨论的数据库连接方法,无论是 DAO, RDO 还是 ADO,都是特定操作系统平台上的数据库设计的,这和 Java 的跨平台特性并不一致。SUN 采用 JDBC(Java Database Connectivity)来解决这个问题。JDBC 是一套设计 Java 数据库应用程序的 API 系统,JDBC 提供数据库应用的几乎所有功能。

下面简要说明如何用 JDBC 建立连接。

以 JDBC 建立和数据库的连接之前一般是建立一个 JDBC-ODBC 连接,通过这个连接利用 ODBC

驱动操作数据库。建立一个和 Sybase SQL Anywhere 数据库的连接。

```
//以 JDBC 连接一个 ODBC 数据库
String url="jdbc:odbc:Zoo";//通过 JDBC-ODBC 桥
建立一个和数据库 Zoo 的连接
//创建连接
Connection con =DriverManager.getConnection
(url,"dba","sql");
```

当建立和数据库的连接后,可以进行如下查询以获得结果集。以:

```
Statement stmt=con.createStatement()
```

建立一个 Statement 对象,在 Statement 对象中进行简单查询,将其结果返回为一个结果集,这个结果集就是一份能够被操作的数据库数据了。

2 用 Java 小程序实现自动更新

实时监控对工业监控是非常重要的,现在已成为其不可缺少的一部分。我们所用的 Java 小程序是基于某个实时数据库的,工业现场的数据通过相应的采集程序送入数据库,Java 小程序通过访问该数据库将现场数据反映给用户。对数据库的访问可以利用 Java 所提供的 ADO。

下面用电力监控的一个例子来说明。电力监控常常要求图形和数据的紧密结合,例如监视某个电力一次系统的相关数据,需要将实时数据同一次系统图紧密结合在一起,在图中反映出各个开关的变化和相关的电压电流值。这可以用 Java 小程序的多线程和绘图优势得到很好的解决。在考虑如下的一幅一次系统页面时,将背景图与和实时数据相关的图像,字符区分开。背景图和实时图分别绘制。

2.1 背景图的载入

背景图可以提前做成 bmp 或 jpg 等形式的图像,放在 Java 工程的目录下,在程序中调用图像 API 来装载。下面是装在图像的一段代码:

```
public Image waitForImage(String filename)
{
    Image img;
    MediaTracker tracker=new MediaTracker(this);
    Img=getImage(getCodeBase(),filename);
    Tracker.addImage(img,0);
    try
    {tracker.waitForID(0);}
    catch(InterruptedException e){}
    return(img);
}
```

(下转第 58 页)

完成上述计算后,将转速送显示,再回到转速采集程序的开始,重复上面的工作。

该程序用到了定时器 T1 和键盘中断,关于中断优先级的控制编程可参阅相关文献^[4]。

当然,上面只是说明了实现转速采集及计算的原理,适当地选取每次转速采集过程完成后的时间延时,便可以取得很好的适时显示效果。另外,通过改变所记录的时间发生次数,能获得更高的显示精度。

3 实验结果分析

上面实验采集的数据可以通过 80C196KC 的串口,用 RS-232C 标准电缆送到 PC 机上显示出来。图 5 为电机起动和停机时的转速变化曲线,该实验曲线从整体上反映了两个动态过程,与理论值基本相符。

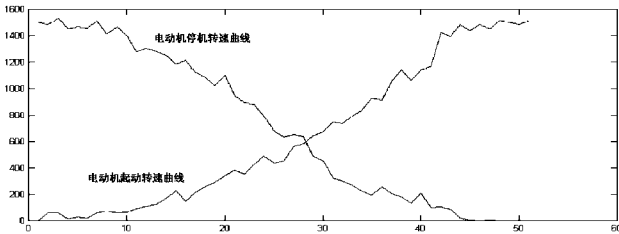


图 5 电动机起动和停机的转速变化曲线

4 结束语

在单片机的两片系统中,就微处理器而言,可以

选用 Intel 8098 系列中的专门用于电机高速控制的 80C196MC,甚至 Intel 80286 也可以很好地与 PSD 系列芯片接口。就 PSD 芯片而言,该芯片已发展到 PSD9XX 系列,它们具有更大的存储容量和更加灵活、丰富的接口功能,其中的 PSD4XX 和 PSD5XX 系列都可以与 16 位机直接接口,并且,随着产品的不断更新,其价格差距也不断缩小,大有取而代之。

通过以上分析可以看出,单片机两片系统结构简单明了,使用灵活方便,不失为一很好的单片机设计思路和发展方向,在今后单片机开发中具有很好的应用前景。

参考文献

- 1 颜荣江,朱元清,侯文军.PSD3 系列可编程单片机外围接口芯片原理编程及应用.北京:人民邮电出版社,1995
- 2 孙涵芳.单片机现场可编程外围芯片 PSD 的原理及应用.北京:北京航空航天大学出版社,1998
- 3 孙涵芳.Intel 16 位单片机.北京:北京航空航天大学出版社,1998
- 4 汪建,章育群.MCS-96 系列单片机的中断优先级的控制方法.微处理机,1999(1)

[收稿日期:2001.10.14]

(上接第 42 页)

067 基地变电站一次系统

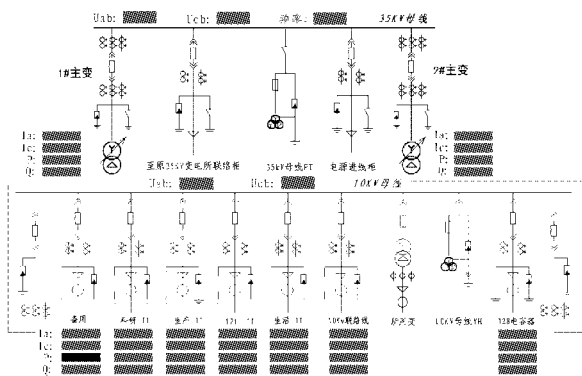


图 1 电力监控界面图

2.2 实时图像的绘制

图像的实时性是通过线程来实现的,每次线程运行时,对数据库重新查询的到新数据,并根据查得的新数据对图像进行重画,这样就能将时刻变化的一个电力一次系统反映给用户。线程可以用 Thread 类实现,也可以通过 Runnable 接口来实现。不论哪种实现方法你都需在线程的 run()方法中利用数据库对象

实现对数据库的查询,用 repaint()实现图像的重画。需要注意的是,应在线程的在 Java 小程序的 Paint()函数中编写与实时数据相关的图像,这样字符的显示代码,这样才可以用 repaint()重画。另外,线程的休眠时间可以由 Sleep(int time)函数决定(单位为毫秒),这决定了图像的刷新速率。关于这部分代码的编写,其中 paint()函数的编写应由具体的要求来定,调用 API 函数很容易实现。线程的定义也很容易,线程 run()函数的编写主要是对数据库的访问,这部分代码在前面已经给出。

3 结束语

本文具体讨论了对工业监控环境中对实时数据在 Web 页面上的自动定时刷新问题的一种实现方法,希望对该问题的解决起到抛砖引玉的作用。

参考文献

- 1 Richard 著.VJ++宝典.电子工业出版社,1999
- 2 闪四清著.Microsoft SQL Server 2000 实用教程.人民邮电出版社,2000

[收稿日期:2001.10.9]