Boğaziçi University
Electrical & Electronics Engineering
Fall 2003 EE 497 Java Programming
by Prof. Dr. Işıl Bozma

Term Project Report

Çağdaş Kayra Akman
9902929

14.01.2004

Project Title: Communication over Serial Port

Problem Statement:

There are many electronic devices which are connected to some kind computer for various purposes such as control, signal processing, measurement, etc. Serial/parallel port or USB port on computers are points of access to external devices.

In this project, a Java application to access the serial port of a PC for information transfer is developed.

API's used:

- javax.comm (Java Communications API)
- java.io
- java.util

Implementation

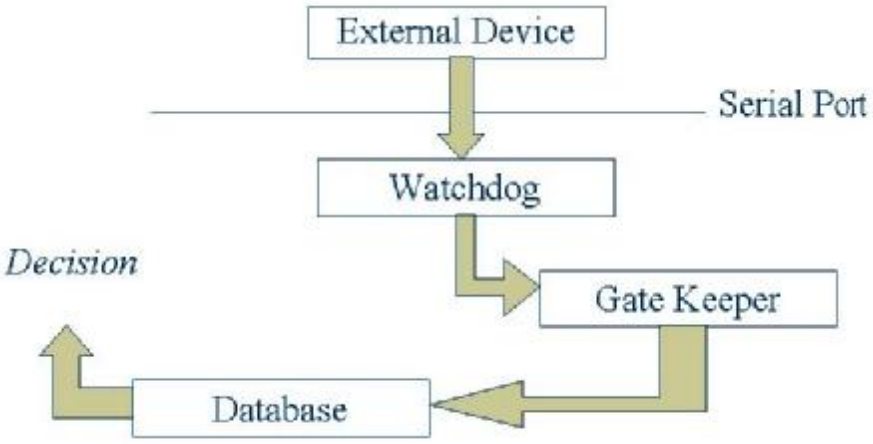
An application called "Gate Keeper" is developed. An external device, a magnetic card reader, is connected to serial port (COM1) of a PC. Each time a card is read, the data stord in the magnetic strip at the back of the card is sent to PC in asynchronous mode of communication with the following parameters. Baud rate : 9600, 8 data bits, 1 stop bit an no parity genertaed.

There are three classes: GateKeeper, Watchdog and Database. Brief descriptions of these classes are as follows:

1. **Watchdog.java :** This class handles the serial port connection of the application. First COM1 port is found among available ports using CommPortIdentifier class of Java Communications API. Then the COM1 port's reference is passd to SerialPort object, whose **open()** method is called to open the port, **setSerialPortParams(int, int, int, int)** method is called to initialize the port to settings given above and **notifyOnDataAvailable(boolean)** method is called to activate the event Listener to listen for new data arrivals. **getInputStream()** method is used to access the data provided by the external device. **portOpened()** method of this class returns the SerialPort object that is the reference to the port that it owned by the application.

2. **GateKeeper.java :** This class has the **main(String [])args** method. It creates new instances of the other classes, adds Enevt Listener for the serial port, implements **SerialPortEventListener** interface for serial port event handling. The serial port is listened only for new data arrivals, so teh impleemnted **serialEvent(SerialPortEvent event)** method is called once new data is received through serial port. This method cleans teh ';' character at the beginning and '?' character at the end of the of the data received, which act as framing characters. They are used th extract the data. While doing this, single bytes are converted to characters and concatenated to form strings. The final string is passed to **data(String data)** method of the **Database** class.
3. **Database.java :** This class contains a hastable that is of size 20 and holds the user data and assigned user names such as "user1", user2", etc. There are two modes of operation: Normal and Database Updating modes. When "099" or "100" is received as input data, Database Updating mode is entered, in ehich next data obtained is added as a new user in case of a preceding "099", or as a user to be removed in case of a preceding "100". To keep track of these modes, two flags, namely **flag** and **registerFlag**, are used. The first one is determined when to use that part of the code which adds the new user data into the hashtable. The other one prevents the code that implements Normal mode operations to be executed after registering or removing a user, i.e. it is prevented that the data that is used in Database Updating mode to be reprocessed by the Normal mode code. Various **if** and **else if** operations are performed to distinguish diffrent possible combinations of decisions and errors.

   The following diagram depicts the data flow from the external device through the application:



References:

- Java 2 SDK API Documentation
- Java Communications API Documentation
- H. M. Deitel, P. J. Deitel, *JAVA How To Program*, Upper Saddle River, New Jersey: Prentice Hall, 2003
- Java Tech, http://www.particle.kth.se/~lindsey/JavaCourse/Book/index.html