



# Java Based Serial Data Logging System

Anupama.B<sup>#1</sup>, Sukesh Rao.M<sup>#2</sup>

<sup>#1</sup>M.Tech Student, Department of Digital Electronics and Communication, N.M.A.M.I.T Nitte, Karkala, India

<sup>#2</sup>Assistant Professor, Department of Electronics and Communication, N.M.A.M.I.T Nitte, Karkala, India

<sup>1</sup>anupamagk.gk@gmail.com

<sup>2</sup>sukesh@nitte.edu.in

**Abstract**— *The main objective of this project is to develop a Java based GUI (Graphical User Interface) for database entry through serial port. This paper proposes direct database entry through Java API (Application Program Interface) and database entry through serial port using Java software environment. Since the programming language is Java, this document deals with features and configuration required for the Java project.*

*The technological aspects of this project are database connectivity, serial port communication and developing GUI. The project “Java Based Data Logging System” demonstrates a prototype that can be developed at server side within an integrated campus or pool. The back-end installed architectural independent nature of Java can be useful for quick data entry and analysis between client-server in a secure and easy manner.*

*Nowadays most users already own the requisites such as a mobile phone or desktop PC; hence the cost of the system considerably reduced. In more and more healthcare facilities, patient files are being kept in databases that can be accessed from anywhere in the facility. This is not only a time saver but it also results in better data coordination and management. The widespread adoption of electronic health records has resulted in significant savings in health care costs as well as improved patient health and safety.*

**Keywords** — *Java, database, serial port, API, GUI*

## I. INTRODUCTION

The technology today affects every single aspect of modern society. From widespread adoption of electronic medical records towards advances in bio-medical engineering and technology, modern healthcare and its delivery methods are changing at an ever increasing rate. Java based data logging system is one such method helpful for updating user records in databases. The user can access the database located at remote access using SQL(Structured Query Language) query and can insert values for the same using Java programming. Due to its rich library and API(Application Program Interface) support Java is used as the main programming language.

Java is a popular third-generation programming language, which can be used to perform any of the thousands of things that computer software can do. With the features it offers, Java has become the language of choice for Internet and Intranet applications. Java plays an important role for the proper functioning of many software- based devices attached to a network. The kind of functionality the Java offers, has contributed a lot towards the popularity of Java.

Java application development is supported by many different tools. One of the most powerful and helpful tool is the free eclipse IDE(Integrated Development Environment).Eclipse requires Java to run, therefore Java SE Development Kit 6 (32-bit) is required to be installed prior to the IDE. We need to select the versions of JDK and JRE according to our system specification

## II. DATABASE ACCESS MANAGEMENT

A database is an integrated collection of data, organized to meet the needs of multiple users in an organization. A database system involves the data itself, the hardware on which the data resides, the software that controls the storage and retrieval of data, and the users themselves.

JDBC(Java Database Connectivity) is a Java API(Application Program Interface) that provides Java programmers with a uniform interface for accessing and manipulating a wide range of relational databases. Since a JDBC driver serves as the interface to facilitate communications between JDBC and a proprietary database, JDBC drivers are database-specific.

### A. Developing JDBC Applications

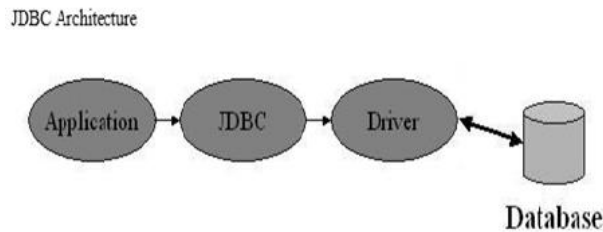


Fig.1 Database Connectivity.

### B. Steps

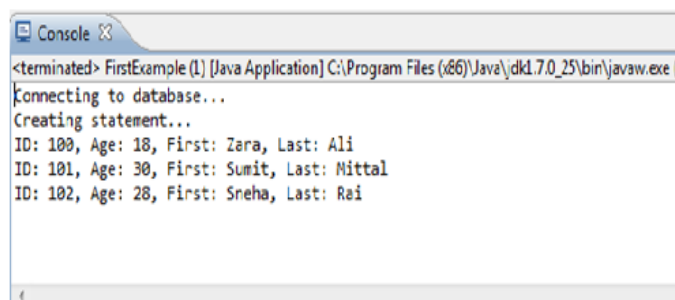
- Load the Driver : JDBC allows multiple drivers to be loaded in one program. A program can load a JDBC driver at any time. For example *MyDriver* is loaded with the following statement:  
*Class.forName("MyDriver")*
- Establish connections: Once the driver is loaded and registered with the driver manager, the *DriverManager* is capable of choosing an appropriate one from a list of registered drivers to fulfil the connection to the database. We simply invoke *DriverManager.getConnection()* to establish a session with the database.
- Create statements: We can create statements for executing SQL in three types: *Statement*, *PreparedStatement*, or *CallableStatement*. The *Statement* class can be used to execute static SQL. The *PreparedStatement* and *CallableStatement* classes are used to construct dynamic SQL.
- Execute statements : The statements can be executed using the methods *executeQuery*, *executeUpdate*, or *execute*. The result of the query is returned in *ResultSet*.
- Process *ResultSet* : The *ResultSet* maintains a table whose current row can be retrieved. The initial row position is *null*. The methods such as *absolute*, *first*, *last*, *next*, and *previous* can be used to set the row position. We can use various *get* methods to retrieve values from a current row.

```

mysql> use testdb;
Database changed
mysql> show tables;
+-----+
| Tables_in_testdb |
+-----+
| analysis          |
| employees         |
+-----+
2 rows in set (0.09 sec)

mysql> select * from employees;
+----+-----+-----+-----+
| id  | age  | first | last  |
+----+-----+-----+-----+
| 100 | 18   | Zara  | Ali   |
| 101 | 30   | Sumit | Mittal |
| 102 | 28   | Sneha | Rai   |
+----+-----+-----+-----+
3 rows in set (0.22 sec)

mysql>
  
```



```

Console
<terminated> FirstExample (1) [Java Application] C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javaw.exe
Connecting to database...
Creating statement...
ID: 100, Age: 18, First: Zara, Last: Ali
ID: 101, Age: 30, First: Sumit, Last: Mittal
ID: 102, Age: 28, First: Sneha, Last: Rai

```

Fig. 2,3 Working with MySQL database

### C. Serial Communication In Java

Peripheral devices are usually external to the computer. Printers, video cameras, scanners, data/fax modems, plotters, robot, light switches etc. Serial communication is a way that enables different equipments to communicate with their outside world. It is called serial because the data bits will be sent in a serial way over a single line.

A personal computer has a serial port known as communication port or COM Port used to connect a modem or any other device, there could be more than one COM Port in a PC. Serial ports are controlled by a special chip called UART (Universal Asynchronous Receiver Transmitter). Different applications use different pins on the serial port and this basically depend upon the functions required. The Java Communications API not only gives us that, but cleverly unifies the programming model for dealing with a range of external devices. It supports both serial (RS232/434, COM or tty) and parallel ports.

### D. Java API

The Communications API is centered around the abstract class *CommPort* and its two subclasses, *SerialPort* and *ParallelPort*, which describe the two main types of ports found on desktop computers. *CommPort* represents a general model of communication and has general methods like *getInputStream()* and *getOutputStream()* that allow us to access information from serial port.

The official API for serial communication in Java is the *JavaComm* API. This API is not part of the standard Java 2 version. Therefore they are separately installed as follows;

- comm.jar should be placed in java-home/jre/lib/ext
- Win32com.dll should be placed in java\_home/jre/bin
- Javax.comm.properties should be placed in java\_home/jre/lib

### E. Preliminaries

- Test the serial port working, connect Rx and Tx of serial cable in loopback method.
- Start *HyperTerminal*, and configure the connection for 9600 Baud rate with 1 stop bit and 8 data bits and no flow control.
- With nothing plugged into the serial port, typing will show nothing in the terminal, now connect Rx and Tx using null modem or loop back method, now we can see whatever we typed can be appeared in the terminal.

F. *Project Setup and Program flow using Eclipse IDE*

- Create a new eclipse project for the lab. (File-> New -> Project ->Java project).
- Download the JAR files and import them.(File->Project->properties->Java Build path->add external library).
- Import serial port library, libraries for i/o, libraries for database, libraries for GUI, libraries for events.
- Declare the database driver address, url , username and password as different variables.
- Open the available serial port and configure the port parameters.
- Check if there is any data available using standard class *CommPortIdentifier*.
- Whenever there is data available event read them and convert as array of characters.
- Declare character array writer and append each character of particular array using functions supported by the *CharArrayWriter*.
- Access the table and database created using *MySQL*.
- Print each array of character in separate character array and insert it separately using SQL query.
- Design GUI to show the user inputs and inserted values into database successfully.

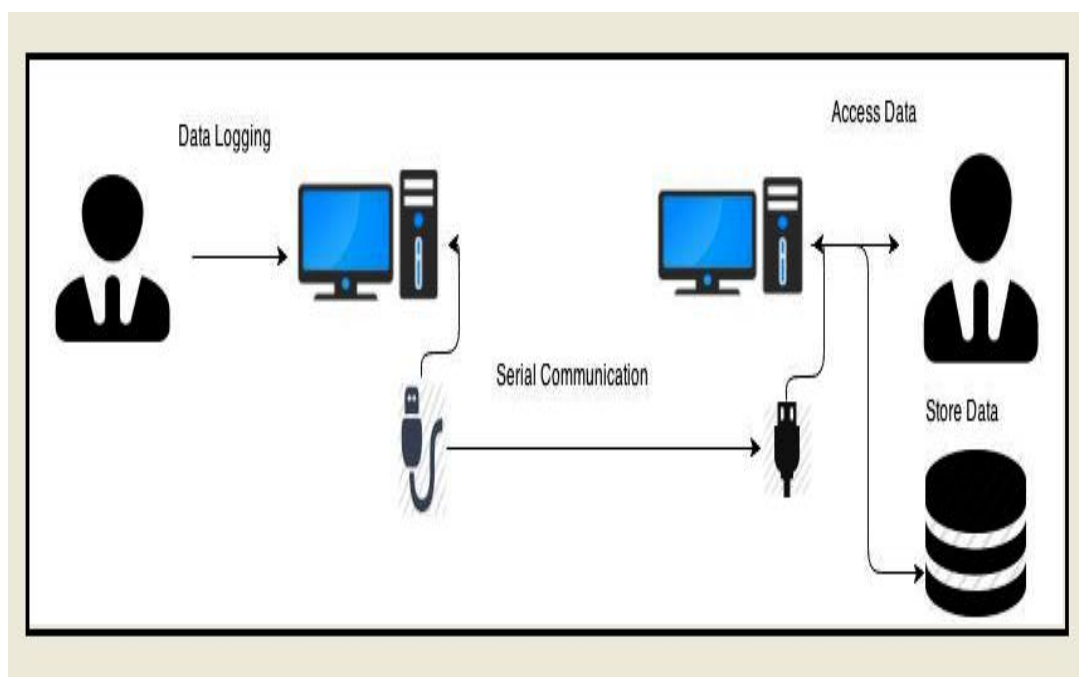
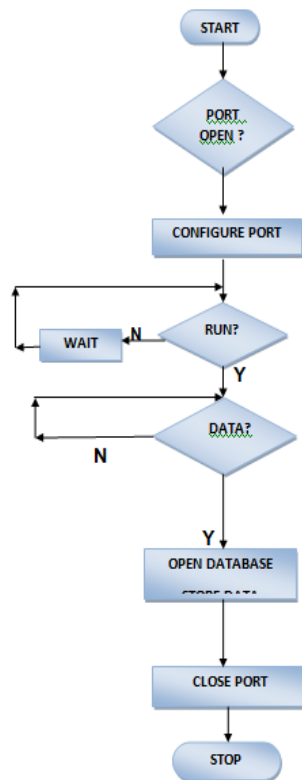


Fig. 4 Diagrammatic Representation of project-setup

G. Flow-Chart



```

MySQL 5.5 Command Line Client
mysql> create table patient_rec(id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,first
name VARCHAR(100),details VARCHAR(100),cur_timestamp TIMESTAMP NOT NULL);
Query OK, 0 rows affected (0.13 sec)

mysql> DESCRIBE patient_rec;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | PRI | NULL | auto_increment |
| first | varchar(100) | YES | | NULL | |
| name  |          |     |   |      | |
| details | varchar(100) | YES | | NULL | |
| cur_timestamp | timestamp | NO | | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+-----+
  
```

Fig.5 Creating patient\_rec table in patient\_data database which is used as an application for this project.

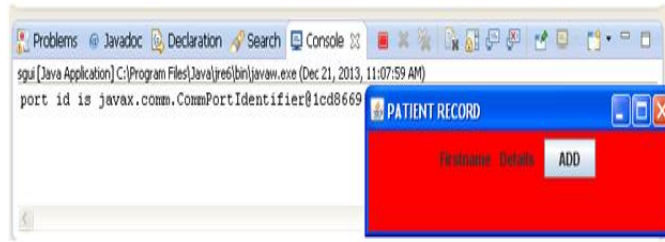


Fig.6 Screen shot after compilation before giving data.

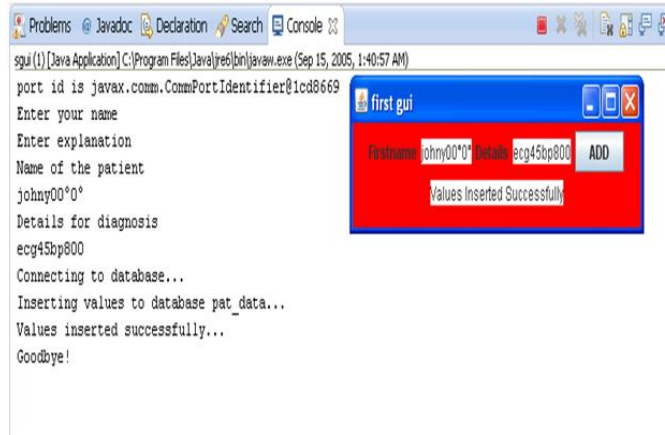


Fig.7 Showing result after compilation and transferring data through serial port via Java program.

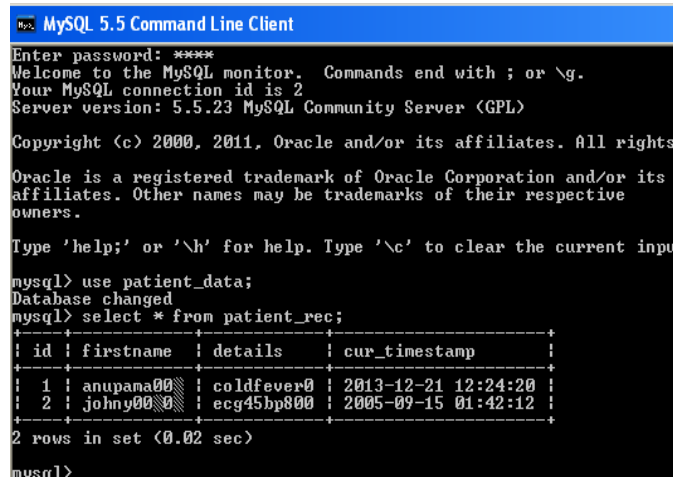


Fig.8. Showing patient\_data updated soon after compilation of sgui.java program.

### III. CONCLUSION

This work is an effort of analysing the support of java language towards database entry through serial programming. Efforts are underway to standardize API of real-time Java and maintain the portability of real-time Java. The current system developed in integrated campus or buildings where the client/server communication through bluetooth data. The coverable region can be extended by developing this prototype for wireless network.

### ACKNOWLEDGMENT

Our thanks to Department of Electronics & Communication and Department of Computer Science for their help and guidance.

#### REFERENCES

- [1] Noureddine BELGACEM , Said ASSOUS, Fethi BEREKSI-REGUIG, "Bluetooth Portable Device and Matlab-Based GUI For ECG Signal Acquisition and Analysis", 2011 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA).
- [2] Mrutyunjaya Swain, South Carolina State University, Orangeburg, SC James A. Anderson, South Carolina State University, Orangeburg, SC Raghu Korrapati, Webster University, Columbia, SC Nikunja K. Swain, South Carolina State University, Orangeburg, SC, "Database programming using Java", Proceedings IEEE Southeastcon 2002.
- [3] B. Raman, M. Wilson, I. Benche and P. Soliz Kestrel Corporation, Albuquerque, NM 87109, "A Java based System for Segmentation and Analysis of retinal Images" Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems (CBMS'03)1063-7125/03 \$17.00 © 2003 IEEE.
- [4] In-Gook Chun, In-Sik Hong, " The implementation of knowledge-based recommender system for electronic commerce using Java expert system library", Division of Information Technology Engineering, SoonChunHyang University , 0-7803-~09C-2/01/\$0100 @ 2001 IEEE.
- [5] Huiwei Guan, Horace H.S.Ip and Yanchun Zhang, "Java-based approaches for accessing databases on the Internet and a JDBC-ODBC implementation", Computing & Control Engineering Journal April 1998.
- [6] Java website Available: <http://www.javatpoint.com/swing>