IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.
PP 16-21
www.iosrjournals.org

Remote Data Monitoring System with Android Application

Neelam V Tambe ,Prof.R.R.Itkarkar
ME-VLSI & ES, Department of Electronics & Telecommunication Engineering,RSCOE,PUNE
Department of Electronics & Telecommunication Engineering,RSCOE,PUNE

Abstract: *A data monitoring system is a complete set of hardware and software that helps to obtain knowledge of the system and helps to access the data to the user. Remote monitoring system is used in different control applications, engineering systems. Many of the monitoring systems make use of the internet and internet protocols and technologies. Using these systems leads to save resources exploitation and manpower as well as can be used in the remote areas where human cannot reach. A simple definition of remote data monitor is collecting information about a system or a process. The data collected is used to examine various parameters of a system and the implications it would have on the dependent processes/systems. The systems that enable supervision of remote processes for data collection are termed remote data monitor or remote data collection systems. These systems are designed using personal computers and other processor-based input/output modules conforming to RS-232 and RS-485 standards. Our aim is to design, develop and execute an application which can be used in the various fields of engineering to monitor the data with a remote access.*

Keywords: *Android, monitoring, validation, enterprises*

I. Introduction

Remote Data collection is very crucial in process engineering applications. It assists evaluating and enhancing the effectiveness, performance, accuracy, consistency, and energy consumption of a system and/or process.
It is used in the following applications:

- Temperature sensors – remote data acquisition
- Pressure sensors and strain gauges
- Flow and speed sensors – remote data acquisition
- Current loop transmitter – remote data acquisition
- Weather and hydrological sensors
- Laboratory analytical instruments

Data Monitor Systems are the basis for building monitoring tools that enable the supervision of local and remote systems. It is difficult for developers to compare proprietary generic DMS products and/or standards, and the design of a specific DMS is expensive. In this, we propose an implementation independent specification, based on conceptual and role based use case modelling, of a generic architecture for DMSs. This generic DMS specification gives DMS developers an abstraction of DMSs; it enables them to compare existing DMS products and standards; and it provides the DMS developers that aim to develop a specific DMS with a starting point for the design of a specific DMS.

Remote monitoring systems have been developed in many business areas such as building, power engineering and transportation systems to provide condition-monitoring systems with information about the state of equipment. The kernel of any remote monitoring system is a data acquisition system which enables the collection of relevant data. There are many standards for DASs such as OLE for Process & Control (OPC), Interchangeable Virtual Instrument (IVI) and Open Data Acquisition Standard (ODAS).

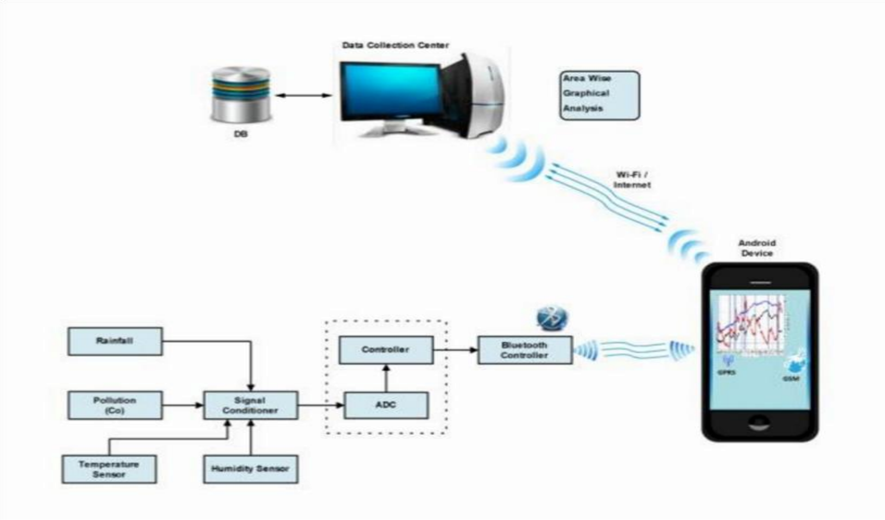


Fig.1. Proposed System

II. TECHNOLOGY FRAMEWORK

2.1.SENSORS

IR Sensor:An Infrared (IR) sensor is an electronic sensor that emits IR light so as to sense objects in its field of view. These consist of an IR transmitter and an IR receiver; the IR transmitter is an IR LED and the receiver is a photodiode which is sensitive to IR light. When the IR light falls on the photodiode the output voltage change in proportion to the magnitude of the IR light received.

Potentiometer: Potentiometer is a three terminal resistor which is used as adjustable voltage divider. Two terminals are connected to both ends of a resistive element; the third terminal is connected to the slider or wiper that moves over the resistive element. The position of the wiper tells the output voltage of the potentiometer.

III. MICROCONTROLLER

Microcontroller is a small computer on a single IC that contains memory, processor, and input/output peripherals.Whether a particular requirement needs to be implemented using discrete ICs or PLDs or a microprocessor must be determined by the designer. However, many applications can be suitably implemented using microcontrollers, and a great many of them would benefit from using the AVR as outlined briefly below.

Atmel’s AVR RISC family of controllers has the following features:

- RISC architecture with mostly fixed-length instruction, load-store memory access, and 32 general-purpose registers.
- A two-stage instruction pipeline that speeds up execution.
- Majority of instructions take one clock cycle.
- Up to 10-MHz clock operation. In-system programmable.
- Available in 8-pin to 64-pin package size to suit wide variety of applications.
- Up to 12 times performance speedup over conventional CISC controllers.
- Wide operating voltage from 2.7 V to 6.0 V.

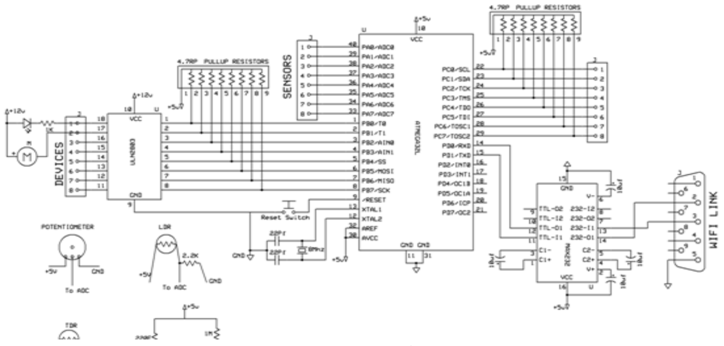


Fig.2. Circuit Diagram of the System

3.1.BLUETOOTH MODULE

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices and building personal area networks (PANs). It can connect various devices, overcoming problems of synchronization. Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 25,000 member companies in the areas of telecommunication, networking, computing, and consumer electronics.

In our research work, Bluetooth is used to connect our input section (sensors, microcontroller) with the android device. Thus with the help of Bluetooth module we can remote access the data acquired.

3.2.SOFTWARE SYSTEM

3.2.1.JAVA:

Java Platform, Enterprise Edition or Java EE is a widely used platform for server programming in the Java programming language. The Java platform (Enterprise Edition) differs from the Java standard Edition Platform (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

General APIs

The Java EE APIs includes several technologies that extend the functionality of the base Java SE APIs.

- Java EE 6 Platform Packages
- Java EE 5 Platform Packages

javax.faces.* This package defines the root of the JavaServer Faces (JSF) API. JSF is a technology for constructing user interfaces out of components.

javax.faces.component.* This package defines the component part of the JavaServer Faces (JSF) API. Since JSF is primarily component oriented, this is one of the core packages. The package overview contains a UML diagram of the component hierarchy.

javax.servlet.* The servlet specification defines a set of APIs to service mainly HTTP requests. It includes the JavaServer Pages specification.

javax.enterprise.inject .* These packages define the injection annotations for the contexts and Dependency Injection (CDI) API.

javax.enterprise.context .* These packages define the context annotations and interfaces for the Contexts and Dependency Injection (CDI) API.

javax.ejb .* The Enterprise JavaBeans (EJB) specification defines a set of lightweight APIs that an object container (the EJB container) will support in order to provide transactions (using JTA), remote procedure calls (using RMI or RMI-IIOP), concurrency control, dependency injection and access control for business objects. This package contains the Enterprise JavaBeans classes and interfaces that define the contracts between the enterprise bean and its clients and between the enterprise bean and the ejb container.

javax.validation This package contains the annotations and interfaces for the declarative validation support offered by the Bean Validation API. Bean Validation provides a unified way to provide constraints on beans (e.g. JPA model classes) that can be enforced cross-layer. In Java EE, JPA honors bean validation constraints in the persistence layer, while JSF does so in the view layer. **javax.persistence**This package contains the classes and interfaces that define the contracts between a persistence provider and the managed classes and the clients of the Java Persistence API (JPA).

javax.transaction This package provides the Java Transaction API (JTA) API that contains the interfaces to interact with the transaction support offered by Java EE. Even though this API abstracts from the really low-level details, it is itself also considered somewhat low-level and the average application developer in Java EE is assumed to be relying on transparent handling of transactions by the higher level EJB abstractions.

javax.jms .* This package defines the Java Message Service (JMS) API. The JMS API provides a common way for Java programs to create, send, receive and read an enterprise messaging system's messages.

javax.xml.stream This package contains readers and writers for XML streams.
javax.resource .* This package defines the Java EE Connector Architecture (JCA) API. Java EE Connector Architecture (JCA) is a Java-based technology solution for connecting application servers and enterprise information systems (EIS) as part of enterprise application integration (EAI) solutions. This is a low-level API aimed at vendors that the average application developer typically does not come in contact with.

3.2.2.Net Beans:

NetBeans refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, Scala, Clojure, and others.

The NetBeans IDE is written in Java and can run anywhere a JVM is installed, including Windows, Mac OS, Linux, and Solaris. A JDK is required for Java development functionality, but is not required for development in other programming languages.

The NetBeans platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans platform (including the NetBeans IDE) can be extended by third party developers.[3]

The NetBeans Platform is a reusable framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required. Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated Development Tools

Modularity: All the functions of the IDE are provided by modules. Each module provides a well defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

3.3.COMMUNICATION TECHNOLOGY

3.3.1.GSM/GPRS

Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The reason for this is to limit the designers as little as possible but still to make it possible for the operators to buy equipment from different suppliers. The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS). There are four types of communication methods in GSM network data mode, voice mode, GPRS mode, SMs mode. Voice mode is just a telephone work, and cannot be used for data transmission, and the other three can all be used. Data mode is just like both sides having a talk with data, and it is a real time data transmission mode, but it is also a dear mode especially multi-province work. GPRS

GPRS has the advantages of high speed, immediacy, and packet switching. But GPRS still has a few limitations, such as transit delays and no storage and forward. As the system requires slave stations to send data to sub stations timely, if there is some interrupt in the communication between slave and sub-stations, it is possible to miss important data by GPRS in this condition.

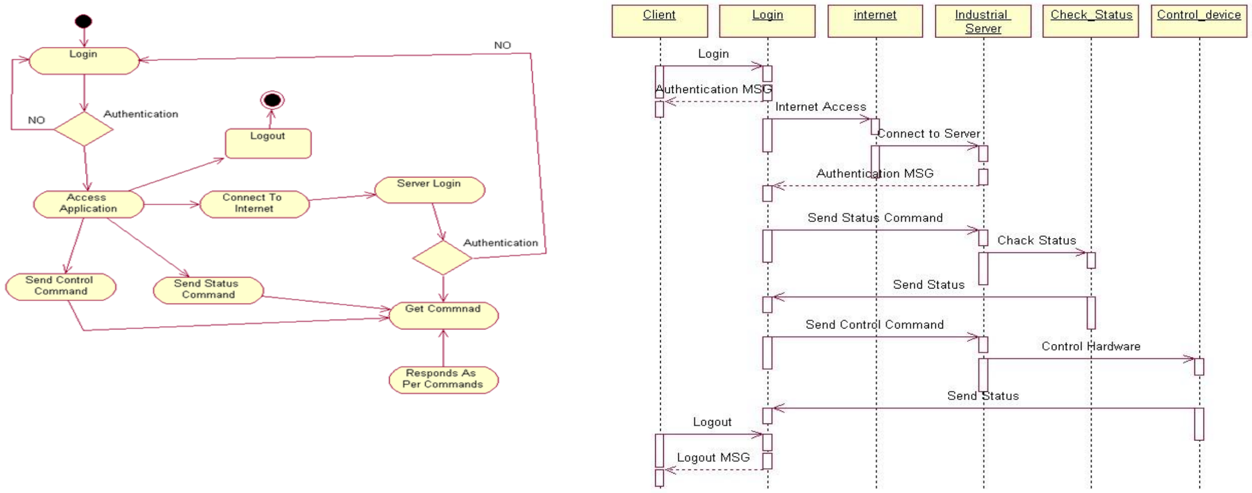


Fig 3. Communication flow(Server)

3.4.WEB SERVER

3.4.1.Apache Tomcat:

Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment for Java code to run in. Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.

Components *Tomcat 4.x was released with Catalina (a servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).*

Catalina *Catalina is Tomcat's servlet container. Catalina implements Sunmicrosystems' specifications for servlet and JavaServer Pages (JSP). In Tomcat, a Realm element .*

Coyote *Coyote is a Connector component for Tomcat that supports the HTTP 1.1 protocol as a web server. This allows Catalina, nominally a Java Servlet or JSP container, to also act as a plain web server that serves local files as*
HTTP documents.

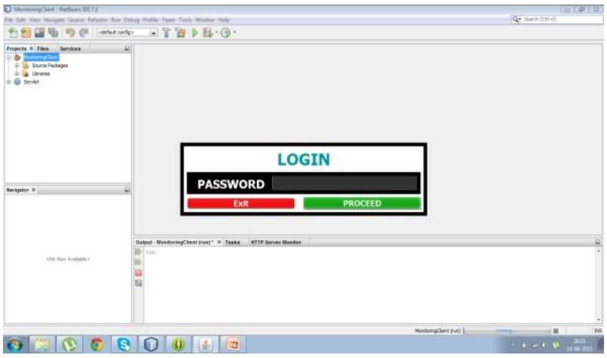
Jasper *Jasper is Tomcat's JSP Engine. Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them.*

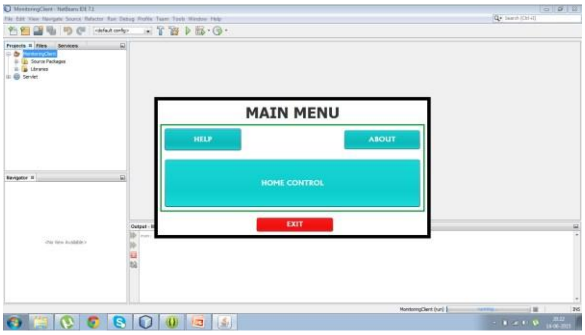
As of version 5, Tomcat uses Jasper 2, which is an implementation of the Sun Microsystems's JSP 2.0 specification. From Jasper to Jasper 2, important features were added.
JSP Tag library pooling - Each tag markup in JSP file is handled by a tag handler class. Tag handler class objects can be pooled and reused in the whole JSP servlet.
Background JSP compilation - While recompiling modified JSP Java code, the older version is still available for server requests. The older JSP servlet is deleted once the new JSP servlet has finished being recompiled.

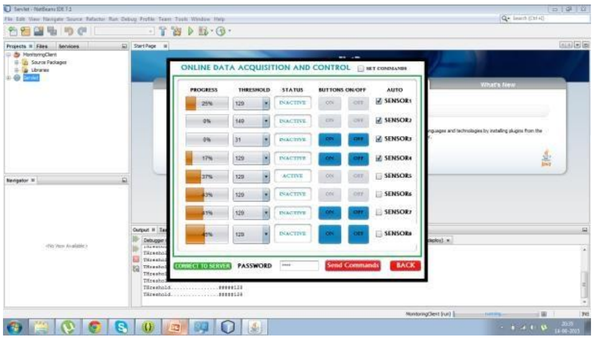
- Recompile JSP when included page changes - Pages can be inserted and included into a JSP at runtime. The JSP will not only be recompiled with JSP file changes but also with included page changes.
- JDT Java compiler - Jasper 2 can use the Eclipse JDT (Java Development Tools) Java compiler instead of Ant and Javac.

IV. RESULT:

represents a "database" of usernames, passwords, and roles (similar to UNIX groups) assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.







PROGRESS	THRESHOLD	STATUS	BUTTONS ON/OFF		AUTO
25%	129	INACTIVE	ON	OFF	☑ SENSOR1
0%	149	INACTIVE	ON	OFF	☑ SENSOR2
0%	31	INACTIVE	ON	OFF	☑ SENSOR3
17%	129	INACTIVE	ON	OFF	☑ SENSOR4
37%	129	ACTIVE	ON	OFF	□ SENSOR5
43%	129	INACTIVE	ON	OFF	□ SENSOR6
41%	129	INACTIVE	ON	OFF	□ SENSOR7
45%	129	INACTIVE	ON	OFF	□ SENSOR8

```
Threshold...............######128
Threshold...............######128
```

V. Conclusion:

In this paper a remote data acquisition system based on GSM/GPRS is presented. This remote data acquisition system has the advantage of low cost, good expansibility, high reliability and easy maintenance, and suits for long time run outdoors. The paper presents a versatile remote measurement and data acquisition system for monitoring applications. In case of remote monitoring over long distances internet connection can be used for data transfer. Applications can be implemented using classical TCP/IP programming techniques.

REFERENCES:

[1] Haitao JIA, Li CA0, “A Remote Data Acquisition System Based on SMs”, IEEE International Conference on Systems, Man and Cybernetics,2004.

[2] Jean Paul TalledoVilela, Jose Carlos Miranda Valenzuela, “Design and Implementation of a Wireless Remote Data Acquisition System for Mobile Applications”, Research Center for Automotive Mechatronics, 2005.

[3] IoanLita, Ion BogdanCioc, IliePopa, “Technologies for Remote Data Acquisition Systems in Environmental Monitoring”, IEEE Conference, 2006.

[4] Chen Xiaorong , Shi Zhan ,GeZhenhua, “Research on Remote Data Acquisition System Based on GPRS”, IEEE Conference The Eighth International Conference on Electronic Measurement and Instruments, 2007.

[5] DaogangPeng, Hao Zhang, Kai Zhang, Hui Li, Fei Xia, “Research and Development of the Remote I/O Data Acquisition System Based on Embedded ARM Platform”, IEEE Conference International Conference on Electronic Computer Technology, 2009.

[6] Shouxian WEN, Lei XU, Xingguo SUN, Xiaohui LI, “A Design of “Remote Monitoring System based on 3G and Internet Technology”, IEEE National Conference on Information Technology and Computer Science, 2012.