**1.1 OVERVIEW OF THE PROJECT**

To transmit electric power from a small number of sources (the generators) to a large number of sinks (consumers) in the frequency range of 50-60 Hz power lines were design. Electrical power lines are usually classified into the high (>100kV), medium (1-100kV) and low (<1Kv) voltage network. Power line communication is an emerging home network technology that allows consumers to use their already existing wiring system to connect home appliances to each other and to the Internet.

For communication purpose electrical power supply network is used in power line communication. Reduction in operational costs and expenditures for communication is the main thing in power line communication. For internal communication of electrical utilities, remote measuring and control task high, medium and low voltage supply have been used. PLC is also used in internal electrical installation within buildings and homes called in home PLC for various communication application. PLC modems are used to make communication in power supply networks. Data signal from conventional communication devices, (computer, telephone) is converted by PLC modem in a form that is suitable for transmission over power lines.

The PLC transmission channel has some negative properties as frequency dependent attenuation, changing impedance, fading and unfavorable noise condition. However, to provide higher data rates PLC network has to operate in a frequency spectrum of up to 30 MHz. PLC network produces electromagnetic radiation and disturb other services operating in the same frequency range.

**1.2 NEED FOR THE PROJECT**

Reduction in operational costs and expenditures for communication is the main thing in power line communication. For internal communication of electrical utilities, remote measuring and control task high, medium and low voltage supply have been used. PLC is also used in internal electrical installation within buildings and homes called in home PLC for various communication application.PLC allows utility companies to reliably move data over an infrastructure that they own and control.

**There are four basic forms of power line communications:**

* **In-house networking**  
  In-house mains power wiring can be used for high speed data transmission for home networking;
* **Broadband over Power Line**The outdoor mains power wiring can be used to offer broadband internet access;
* **Narrowband in-house applications**In-house mains power wiring can be used for low bit rate services like home automation and intercoms.
* **Narrowband outdoor applications**  
  Narrowband outdoor applications are mainly used by the utility companies for automatic meter reading and remote surveillance and control.

**1.3 OBJECTIVE OF THE PROJECT**

PLC modems are used to make communication in power supply networks. Data signal from conventional communication devices, (computer, telephone) is converted by PLC modem in a form that is suitable for transmission over power lines. Although, power supply network is not designed for data communication.

The PLC transmission channel has some negative properties as frequency-dependent attenuation, changing impedance, fading and unfavorable noise condition. However, to provide higher data rates PLC network has to operate in a frequency spectrum of up to 30 MHz.

**1.4 SCOPE OF THE PROJECT**

PLC network produces electromagnetic radiation and disturb other services operating in the same frequency range. PLC is divided into two groups: narrowband PLC allowing data rates up to 100 kbps and broadband PLC allowing data rates beyond 2 Mbps.

With the inevitable arrival of broadband access, the demand for digital voice, video, internet data within the home increases continuously. PLC technology allows the uses of existing and widespread power distribution infrastructure to provide high speed networking capabilities along with many other benefits.

With current available data transmission speeds of 14 Mbps and a remarkable increase promised in the near-future, Power Line Carrier Communication Systems are a preferred choice over Wireless or other Home Network Technologies due to factors including ease of installation, availability of AC outlets, higher throughput, low cost, reliability and security.PLC Communications Systems are also a potential candidate for the deliverance of xDSL and Broadband Internet services (data, muli-media etc,.) along with electricity (and automation control signals) to the consumers by the energy utilities.

**2.1 Structured Compressed Sensing Based NBI Elimination For In-Home Power Line Communication**

**Authors:** Sicong Liu, Fang Yang

**Description:**

The structured compressed sensing based framework for the estimation of Narrow Band Interference (NBI) in power line communication is proposed, which facilitates in-home interconnection and prevents the wired consumer electronics devices from contamination of NBI. To recover the NBI accurately, the Structured Compressed Sensing (SCS) theory is introduced, and the method of SCS based Temporal Differential Measuring (SCS-TDM) is proposed, which fully exploits the

temporal correlation of NBI. By exploiting the repeated training sequences, the NBI measurements matrix is acquired. With the exploitation of the prior partial support information, a more effective greedy algorithm, structured prior aided sparsity adaptive matching pursuit, is proposed. The performance of the proposed algorithm is theoretically guaranteed, and simulation results validate that the proposed method significantly outperforms existing counterparts.

**2.2 A Spectral Compressive Resource Allocation Technique for PLC Systems**

**Authors:** H.J.Han Vinak, V.Ribeiro

**Description:**

This work focuses on the benefits of the existing relationship among consecutive sub channels (spectral information) in a multi-carrier modulation-based power line communication system for reducing the computational complexity and signaling overhead associated with resource allocation techniques. In this regard, we introduce the spectral compressive resource allocation technique, which reduces the computational complexity and the signaling overhead by grouping subcarriers into chunks, whose length/bandwidth is defined with basis on the so called normalized signal to noise ratio coherence bandwidth. Moreover, we address the combination of the proposal with another technique that exploits the existing temporal relationship within periodically time-varying power line channels. Based on a data set composed of in-home power line channel estimates and additive noise measurements, we quantify the trade-off between computational complexity reduction and data rate degradation under distinct scenarios and conditions, and indicate circumstances in which the proposed technique is more useful. Finally, we show that the exploitation of both temporal and spectral information altogether results in computational complexity reduction and less data rate degradation in comparison to other techniques.

**2.3 An Adaptive Impedance Matching System For Vehicular PLC**

**Authors:** Shahriar Mirabbasi, Lutz Lampe

**Description:**

Vehicular Power Line Communication (VPLC) is being considered as a potential solution to mitigate the increase of complexity and cost of the automotive wiring harness caused by the growth of the number of electronic devices and sensors deployed inside vehicles. This is because VPLC reuses power cables for data communication and thus avoids the need for additional communication wires. This reuse does not come without problems though. One of the challenges for VPLC is the time, frequency and location dependency of the access impedance, which can cause severe impedance mismatch for the communication signal. Impedance mismatch degrades the signal-to-noise-power ratio at communication receiver and thus affects transmission reliability. Due to the variable nature of the access impedance, a fixed matching circuit will be inefficient. Hence, in this work we present an adaptive impedance matching system which improves the

Communication signal transfer from the transmitting to the receiving device. The system is evaluated via simulations for a wide range of access impedance test points and S-parameters of VPLC networks obtained in previous measurement campaigns. Our simulation results demonstrate that the presented adaptive matching system is able to achieve VPLC signal-power transfer within 30% of the theoretical optimum. This translates to a power gain, up to a factor of 10, which is larger than that of other solutions reported in the literature.

**2.4 Power Line Communication Management Battery Energy Storage in a Small-Scale Autonomous Photo Voltaic System**

**Authors:** Jeremie Jousse, Nicolas Ginot

**Description:**

Today an increasing number of batteries are equipped with a digital battery management system (BMS) either for safety issues or lifetime improvement, or for both. In order to avoid the use of dedicated wiring for communicating with these

BMS, a power line communication (PLC) solution is proposed to communicate through the dc power line inherent in these systems.This solution is also designed to be directly compatible with the existing controller area network controllers integrated in a large number of commercially available microcontrollers. It uses a single carrier generator in order to generate a shared carrier signal on the bus. Each modem can modulate this common carrier signal by switching the impedance it presents on the bus. In a first part, this paper proposes an overview of the industrial

context of this PLC solution, including simulation results and discussion of the theoretical limits of the system. Then, the digitization stage interfacing the controller to the power line is described and a first prototype is validated through discussion of experimental results obtained within an actual industrial environment.

**2.5 FEASIBILITY STUDY**

A feasibility study is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation.

**2.5.1 Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

**2.5.2 Economic Feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as [cost/benefit analysis](http://en.wikipedia.org/wiki/Cost-benefit_analysis), the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. The present work focuses its attention on the definition of a home automation system based on power line communication. A low cost and non invasive system has been developed and tested with successful results. Combining the use of a microcontroller with a power line modem makes possible to manage sensors and actuators in the neighbors through the power grid and without modifying the electrical and network systems.

**2.5.3 Operational Feasibility**

Is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Any wrong input will not crash or affect the system. Even wrong data from hardware will not crash the system rather it shows an error message.

**2.5.4 Schedule Feasibility**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable.

**3.1 EXISTING SYSTEM**

In hospitals, medical equipment like ECG machine, ventilators, infusion pumps, heart beat and blood pressure monitors are placed near the patients who need medical assistance. Medical Intensive Care Unit (MICU) in some hospitals has automated patient monitoring system for their patient. In some cases these automated units are interconnected by networking for central monitoring and medical data storage. Recent year’s communication technologies are applied in healthcare for performing surgery and delivering assistance to the patients in the form of tele-surgery, telemedicine, biotelemetry using LAN, Radio Frequency (RF), ZigBee, WAN etc.Rural and urban sectors are targeted by the medical industries for assisting and delivering medical care.

**Disadvantages:**

* The drawbacks have to do with the high carrier frequency required for data communications and the sub-optimal design of power systems when you want them to effectively transport these high frequency signal, including attenuation and noise levels.
* It can only transmit during the zero crossing of the AC transmission. Meaning that it is limited in frequency of transmission.

**3.2 PROPOSED SYSTEM**

In Proposed system we present a feasible patient monitoring System in which ubiquitous power lines are mode of transmitting data over 220V/50v) to control the electrical devices. A power line communication modem is used in which FSK [Frequency shift keying] is used for modulating the signal. Sensors connected to microcontroller via PLCC sends the data and in turn controls the devices. There are Multiple receivers which contain the power line modem can be connected through the power line to the devices.PIC controller and Real Time Operating system whereas embedded web server technology is the combination of embedded device and PLCC. PLCC it is possible to monitor patients remotely. This paper presents a PC based temperature monitoring and control system using virtual instrumentation.

Temperature sensor measures the temperature and produce corresponding analog signal which is further processed by the microcontroller. The data will be displayed on the LCD in microcontroller and PC monitor. Monitoring and control can be done with the help of control circuitry.

**Advantages:**

* There will be no need to implement extra infrastructure as PLC will use existing Power lines.
* It provides Flexibility & Stability. It's easy to install.
* Power line communication can be used for many applications like Remote control, Emergency alarms, Security purpose, overall Industrial Automation.
* PLC solution is a complementary solution to traditional fixed line networks, wireless networks.

**3.3 PROPOSED SYSTEM ARCHITECTURE DESIGN**

**MICRO CONTROLLER**

**ADC**

**TEMP SENSOR**

**PRESSURE SENSOR**

**HEART BEAT SENSOR**

**LCD**

**BUZZER**

**PLCC-TX**

**UART**

**POWER SUPPLY**

**GSM**

**Fig 3.1 Architecture Diagram**

**MONITORING SECTION**

**PC**

**JAVA**

**PLCC**

**RX**

**UART**

**CLOUD**

**Fig 3.2 Patient Health Monitoring**

**3.4 DATAFLOW DIAGRAM FOR PROPOSED SYSTEM**

* A data flow diagram (DFD) is a graphical representation of the “flow” of datathrough an information system, modeling its process aspects. A DFD is often uses as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.
* DFD uses defined symbols like rectangles, circles and arrows, plus short text lables, to show data inputs, outputs, storage points and the routes between each destination.

**LEVEL 0:**

Inputs

Different Sensors

**Fig 3.3 Level 0 Data Flow Diagram**

**LEVEL 1:**

View

**Fig 3.4 Level 1 Data Flow Diagram**

**LEVEL 2:**

View DE

**Fig 3.5 Level 2 Data Flow Diagram**

**LEVEL 3:**

Inputs

Different Sensors

Abnormal condition condi

**Fig 3.6 Level 3 Data Flow Diagram**

**3.5 UML DIAGRAMS**

UML or Unified Modeling Language is “a standard notation for the modeling of real-world objects and systems”. This language has various types of diagrams which are divided into two categories: structural diagrams and behavior diagrams. Structural diagrams emphasizes element that must be present in the system that is being modeled whereas behavioral diagrams illustrates what must happen in the system that is being modeled.

**3.5.1 Use Case Diagram**



**Fig 3.7 Interaction between Patient and Doctor**

**3.5.2 Class Diagram**



**Fig 3.8 Overall Class Diagram**

**3.5.3 Activity Diagram**



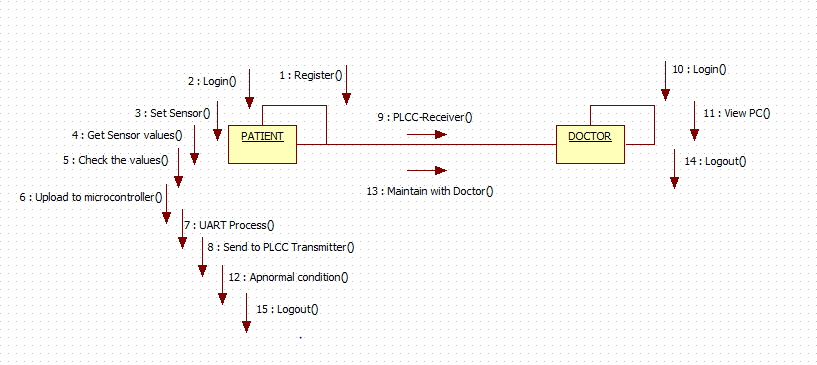
**Fig 3.9 State Flow Diagram**

**3.5.4 Sequence Diagram**



**Fig 3.10 Parallel Processing**

**3.5.5 COLLABORATION DIAGRAM**



**Fig 3.11 Step wise process**

**3.6 MODULE DESIGN**

* Interfacing Sensors
* Programming microcontroller
* Power Line Communication
* GSM
* Data Collection using Java
* Cloud Storage

**3.6.1 INTERFACING SENSORS**

1. Heart Rate Sensor
2. Pressure Sensor
3. Temperature Sensor

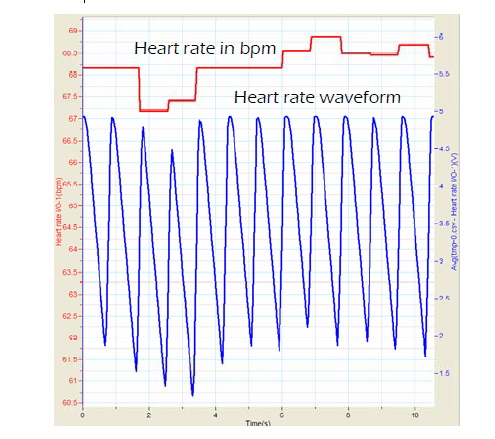
**3.6.1.1 Heart Rate Sensor**

The Heart Rate sensor monitors the light level transmitted through the vascular tissue of the fingertip and the corresponding variations in light intensities that occur as the blood volume change in the tissue. The ease of use makes it possible to measure everyone's heart rate, even in large classes. The Heart Rate sensor measures heart rate between 0 and 200 bpm (beats per minute).

**How it Works:**

Unlike an electrocardiograph (ECG) which monitors the electrical signal of the heart, the Heart Rate sensor measures heart rate by monitoring the change in infrared transmittance through blood vessels. As the heart forces blood through the blood vessels, the amount of blood changes with time and the corresponding variation in light intensities changes.

By plotting this signal, the heart rate can be determined, and some details of the pumping action of the rate can be seen on the graph. A sample graph is shown later in this data sheet.



**Fig 3.12 Monitoring both heart rate in bpm and waveform**

**Sensor Specification:**

Range of the sensor

* Waveform: 0 – 5 V
* Pulse: 0 – 200 bpm (beats per minute)
* Pulse: 0 – 200 bpm (beats per minute)
* Waveform Resolution (12-bit): 1.25 mV
* Pulse resolution: 1 bpm (beats per minute)
* Max. Sampling Rate: 100 samples/sec
* Data Logger Input Type: Digital

**3.6.1.2 Pressure Sensor**

This force sensitivity is optimized for use in human touch control of electronic devices such as automotive electronics, medical systems, and in industrial and robotics applications. The standard 400 sensor is a round sensor 7.62mm in diameter. Custom sensors can be manufactured in sizes ranging from 5mm to over 600mm. Female connector and short tail versions can also be ordered. When external force is applied to the sensor, the resistive element is deformed against the substrate. Air from the spacer opening is pushed through the air vent in the tail, and the conductive material on the substrate comes into contact with parts of the active area. The more of the active area that touches the conductive element, the lower the resistance.

****

**Fig 3.13 Pressure Sensor**

**Applications:**

• Enhance tool safety

• Detect presence, position, or motion

• Relative change in force or applied load

**3.6.1.3 Temperature Sensor**

The measurement of temperature is one of the fundamental requirements for environmental control, as well as certain chemical, electrical and mechanical controls. Many different types of temperature sensors are commercially available, and the type of temperature sensor that will be used in any particular application will depend on several factors. For example, cost, space constraints, durability, and accuracy of the temperature sensor are all considerations that typically need to be taken into account.



**Fig 3.14 Temperature Sensor**

Temperature sensors with a relatively high degree of accuracy are used in smart antenna systems for ground based applications (e.g., cellular antennas) and airborne applications (e.g., airplane or satellite antennas). Smart antenna systems, such as adaptive or phased array antennas, combine the outputs of multiple antenna elements with signal processing capabilities to transmit and/or receive communications signals.

With temperature sensors, such antenna systems can vary the transmission or reception pattern of the communications signals in response to the signal environment to improve performance characteristics.

Temperature sensors are also used in a variety of other applications. For instance, a temperature sensor has been used in petrochemical industry, process control industry, electrical, oil and gas exploration & mining industries whereas exact measurement of particular temperature has to be controlled for getting the desired products or composition. Temperature sensors are commonly utilized in cold rooms to monitor the temperature within the area to ensure that the temperature does not increase to a point where the goods may be at a risk of spoilage.

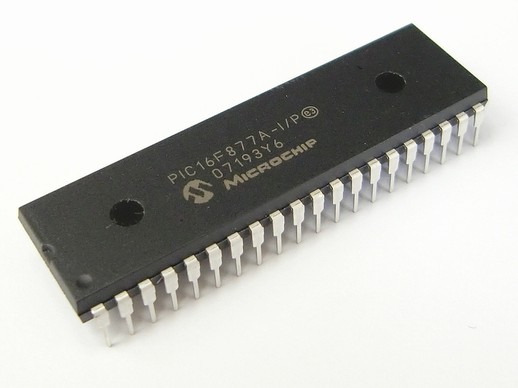
In such applications, a separate temperature sensor is connected to an alarm system to provide for a warning if the temperature increases beyond a particular limit. Storage areas where products are stored which must be kept from freezing are required to have the temperature monitored to ensure that it does not fall below a level at which the products may freeze.

Temperature sensors are also used in greenhouses to maintain the temperature within an acceptable growing range, while also ensuring that the temperature has not reached a level which would cause permanent damage to the crops. LIG thermometers are standard equipment at laboratories and surface weather stations. LIG thermometers have a fine glass bore and a fluid reservoir. Operation depends on the thermal expansion of the liquid contained in the glass envelope. The sensitivity of the LIG thermometer depends inversely on the diameter of the bore of the tube and on the relative expansion coefficients of the liquid and glass.

**3.6.2 MICROCONTROLLER**

**PIC microcontroller 16F877A:**

A PIC microcontroller is a processor with built in memory and RAM and you can use it to control your projects (or build projects around it). So it saves you building a circuit that has separate external RAM,ROM and peripheral chips. It has many useful built in modules E.g. EEPROM, timers, Analogue comparators, UART.



**Fig 3.15 PIC Microcontroller**

One of the most useful features of a PIC microcontroller is that you can re-program them as they use flash memory (if you choose a part with an F in the part number e.g. 12F675 not 12C509).

**Timer / Counters:**

Each PIC microcontroller has up to three timers that you can either use as a timer or a counter (Timer 1 & 2) or a baud clock (Timer 2).

**Timer 0:**

The original timer: Timer 0 was the first timer developed and you can find it in all the earliest devices e.g. 16F84 up to the most current e.g., 16F877A.  It is an 8 bit timer with an 8 bit prescaler that can be driven from an internal (Fosc/4) or external clock.  It generates an interrupt on overflow when the count goes from 255 to zero.  Timer 0 always synchronizes the input clock (when using external clock).

**Timer 1:**

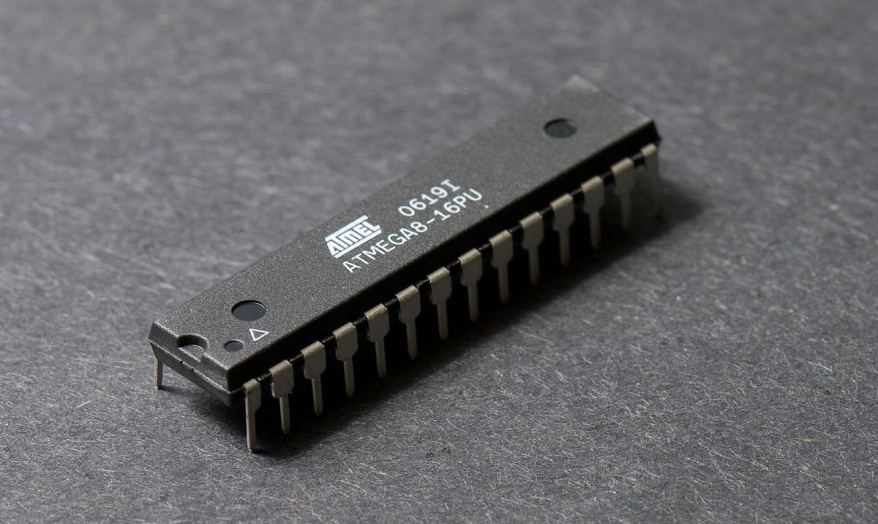
This is a 16 bit timer that generates an overflow interrupt when it goes from 65535 to zero.   It has an 8 bit programmable prescaler and you can drive it from the internal clock (Fosc/4) or an external pin. To eliminate false triggering it also has an optional input synchronizer for external pin input. This timer can be used in sleep mode and will generate a wakeup interrupt on overflow. Timer 1 is also read by the CCP module to capture an event time. In addition it can be used to drive a low power watch crystal.  This is something that sounds good but I don't recommend you do it as watch crystals are extremely difficult to drive correctly.  You should only use it if you are going to make a PCB and follow all the guidelines in making it noise free.

**Timer 2:**

This is an 8 bit timer with an 8 bit prescaler and an 8 bit post scaler.  It takes its input only from the internal oscillator (Fosc/4).This timer is used for the time base of a PWM when PWM is active and it can be software selected by the SSP module as a baud clock. It also has a period register that allows easy control of the period. When timer 2 reaches the PR2 register value then it resets.  This saves having to check the timer value in software and then reset the timer and since it is done in hardware the operation is much faster - so you can generate fast clocks with periods that are multiples of the main clocks

**ATMEL Microcontroller**

* Atmel introduced its first 8-bit first microcontroller AT89C51 in 1993, based on the 8051 core.
* The AT89C51 is a low-power high performance CMOS 8-bit microcontroller with 4k.bytes of flash programmable and erasable read only memory(PEROM).
* It is compatible with the INTEL’S industry standard MCS51 instruction set and pin-out.



**Fig 3.16 ATMEL Microcontroller**

* The device is manufactured using Atmel’s high-density nonvolatile memory technology.
* The on-chip flash allow the program memory to be reprogrammed in-system.
* By combining a versatile 8-bit CPU with flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost-effective solution to many embedded control application.

**3.6.3 POWER LINE COMMUNICATION**

PLC modems are used to make communication in power supply networks. Data signal from conventional communication devices, (computer, telephone) is converted by PLC modem in a form that is suitable for transmission over power lines. Power supply network is not designed for data communication. PLC network produces electromagnetic radiation and disturb other services operating in the same frequency range. PLC is divided into two groups: narrowband PLC allowing data rates up to 100 kbps and broadband PLC allowing data rates beyond 2 Mbps.



**Fig 3.17 Power Line Communication Modem**

Power-line communications systems operate by adding a modulated carrier signal to the wiring system. Different types of power-line communications use different frequency bands. The propagation problem is a limiting factor for each type of power-line communications.

The main issue determining the frequencies of power-line communication is laws to limit interference with radio services. Many nations regulate unshielded wired emissions as they have the potential to cause radio transmissions. These jurisdictions usually require unlicensed uses to be below 500 kHz or in unlicensed radio bands. Some jurisdictions (such as the EU, where longwave broadcast is historically common) regulate wire-line transmissions further. The U.S. is a notable exception, permitting limited-power wide-band signals to be injected into unshielded wiring, as long as the wiring is not designed to propagate radio waves in free space.

**3.6.4 GLOBAL SYSEM FOR MOBILE COMMUNICATION(GSM)**

Global system for mobile communication(GSM) is a globally accepted standard for digital cellular communication. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz. It is estimated that many countries outside of Europe will join the GSM partnership.

Cellular is one of the fastest growing and most demanding telecommunications applications. Today, it represents a continuously increasing percentage of all new telephone subscriptions around the worldwide, and nearby 50 percent of those subscribers are located in the United States. It is forecasted that cellular systems using digital technology will become the Universal method of telecommunications. The concept of cellular service is the use of low-power transmitters where frequencies can be reused within a geographic area.GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware.

**3.6.5 Data Collection using Java**

API by Java developers, it still provides all the necessary functionality for proper serial communications. In order to make the API portable across platforms, the API defines an abstract Serial Port class. This class is then sub classed and platform specific functionality is implemented in the sub classed object. Once a Serial Port object has been created, communications through the physical port are conducted through standard Input Stream and Output Stream objects. These streams send and receive information as bytes, integers or arrays of bytes.

**AES ENCRYPTION:**

* The Advanced Encryption Standard (AES), also known as Rijndael (its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.
* AES operates on a 4 × 4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.
* AES consists of several rounds of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher text.

**AES: Pseudocode**

Cipher(byte in[16], byte out[16], key\_array round\_key[Nr+1])

begin

byte state[16];

state = in;

AddRoundKey(state, round\_key[0]);

for i = 1 to Nr-1 stepsize 1 do

SubBytes(state);

ShiftRows(state);

MixColumns(state);

AddRoundKey(state, round\_key[i]);

end for

SubBytes(state);

ShiftRows(state);

AddRoundKey(state, round\_key[Nr]);

End

**3.6.5 CLOUD STORAGE**

CloudMe is a file storage service operated by CloudMe AB that offers cloud storage, file synchronization and client software. It features a blue folder that appears on all devices with the same content, all files are synchronized between devices. The CloudMe service is offered with a freemium business model and provides encrypted SSL connection with SSL Extended Validation Certificate. CloudMe features a Cloud storage that allows the users to store, access and share their content, both with each other and with people outside the service.

**4.1 HARDWARE REQUIREMENTS**

* PIC microcontroller
* ATMEL microcontroller
* Temperature sensor
* Heartbeat sensor
* Pressure sensor
* LCD
* PLCC modem
* GSM

**4.2 SOFTWARE REQUIREMENTS**

* Java
* HTML, CSS, JAVA SCRIPT
* MYSQL
* Embedded C
* MPLAB
* Web application

**4.3 LANGUAGE SPECIFICATION**

**4.3.1 The JAVA Programming Language**

Java is a Programming Language. It was first developed by James Gosling at Sun Microsystems, which is now a part of Oracle Corporation. It was released in 1995 as a part of Sun Micro system’s Java platform. The language has developed much of its syntax from C and C++. Java applications are usually compiled to byte code (class file) that can run on any Java Virtual Machine (JVM). Java is currently one of the most popular programming languages being used.

**4.3.2 Features of Java**

There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.

**Simple:**

According to Sun JAVA language is simple because:

* Syntax is based on C++ (so easier for the programmers to learn it after C++).
* Removed many confusing and/ or rarely used features e.g., explicit pointers, operator overloading etc.
* No need to remove unreferenced objects because there is Automatic Garbage Collection in JAVA.

**Object Oriented:**

Object oriented means we organize our software as a combination of different type of objects that incorporates both data and behaviour. Object Oriented Programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic concepts of OOPs are:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

**Portable:**

Java supports the feature portability. Java programs can be easily moved from one computer system to another and anywhere. Changes and upgrades in operating systems, processors and system resources will not force any alteration in Java programs. This is reason why Java has become a trendy language for programming on Internet which interconnects different kind of systems worldwide. Java certifies portability in two ways. First way is, Java compiler generates the byte code and that can be executed on any machine. Second way is, size of primitive data types are machine independent.

**Distributed:**

We can create distributed applications in JAVA. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

**Platform Independent:**

A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. JAVA provides software-based platform. The java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware based platforms.

**Multi-threaded:**

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Thread s are important for multi-media, Web applications etc.

**Robust:**

Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.

**Secure:**

Java is designed to be secure in a networked environment. The Java run-time environment uses a byte code verification process to ensure that code loaded over the network does not violate Java security constraints.

**Java is secured because:**

* No explicit pointer
* Java Programs run inside virtual machine sandbox
* **Class loader:** adds security by separating the package for the classes of the local file system from those that are imported from network sources.
* **Byte code Verifier:**checks the code fragments for illegal code that can violate access right to objects.
* **Security Manager:** determines what resources a class can access such as reading and writing to the local disk.

These security are provided by java language. Some security can also be provided by application developer through SSL, JAAS, Cryptography etc.

**4.3.3 SQL Server**

Microsoft SQL Server is a relation database management developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application)—which may run either on the same computer or on another computer across a network (including the Internet).

SQL is used to communicate with a database.SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](https://en.wikipedia.org/wiki/Concurrent_user).

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called [Tabular Data Stream](https://en.wikipedia.org/wiki/Tabular_Data_Stream) (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their [Sybase SQL Server](https://en.wikipedia.org/wiki/Sybase_SQL_Server) relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP), [named pipes](https://en.wikipedia.org/wiki/Named_pipe), and [shared memory](https://en.wikipedia.org/wiki/Shared_memory_(interprocess_communication)).

**4.3.4 IDE**

An integrated development environment (IDE) or interactive development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs offer intelligent code completion feature. Some IDEs contain a compiler, interpreter, or both, such as Net Beans and Eclipse; others do not, such as Sharp Develop and Lazarus. The boundary between an integrated development environment and other parts of the broader software development environment is not well-defined. Sometimes a version control system and various tools are integrated to simplify the construction of a Graphical User Interface (GUI). Many modern IDEs also have a class browser, an object browser, and class diagram, for use in object-oriented software development.

**4.3.5 Net Beans**

Net Beans IDE is the official IDE for Java 8. With its editors, code analyser’s, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyser’s and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, Net Beans IDE sets the standard for developing with cutting edge technologies out of the box.

**4.3.6 Apache Tomcat Server**

Tomcat is a Java servlet container and web server from the Jakarta project of the Apache Software Foundation (http://jakarta.apache.org). A web server is, of course, the program that dishes out web pages in response to requests from a user sitting at a web browser. But web servers aren’t limited to serving up static HTML pages; they can also run programs in response to user requests and return the dynamic results to the user’s browser. This is an aspect of the web that Apache’s Tomcat is very good at because Tomcat provides both Java servlet and Java Server Pages (JSP) technologies (in addition to traditional static pages and external CGI programming). The result is that Tomcat is a good choice for use as a web server for many applications. And it’s a very good choice if you want a free, open source (http://opensource.org/) servlet and JSP engine. Tomcat can be used stand-alone, but it is often used “behind” traditional web servers such as Apache http, with the traditional server serving static pages and Tomcat serving dynamic

Servlet and JSP requests.

**5.1 SAMPLE CODE**

**5.1.1 Source Code:**

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.Scanner;

import com.fazecast.jSerialComm.\*;

import java.io.FileOutputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.util.HashMap;

import java.util.Map;

import java.util.Set;

import org.apache.poi.hssf.record.formula.functions.Row;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.xssf.usermodel.XSSFCell;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileOutputStream;

import java.io.IOException;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class Smart {

public static void main(String[] args) throws IOException {

SerialPort[] ports = SerialPort.getCommPorts();

System.out.println("Select a port:");

int i = 1;

for(SerialPort port : ports)

System.out.println(i++ + ": " + port.getSystemPortName());

Scanner s = new Scanner(System.in);

int chosenPort = s.nextInt();

SerialPort serialPort = ports[chosenPort - 1];

if(serialPort.openPort())

System.out.println("Port opened successfully.");

else {

System.out.println("Unable to open the port.");

return;

}

serialPort.setComPortTimeouts(SerialPort.TIMEOUT\_SCANNER, 0, 0);

Scanner data = new Scanner(serialPort.getInputStream());

String value;

while(data.hasNext()){

FileWriter fw = new FileWriter("D:\\Port\_Read\\port.txt", true);

BufferedWriter bw= new BufferedWriter(fw);

PrintWriter out=new PrintWriter(bw);

ApachePOIExcelWrite a=new ApachePOIExcelWrite();

a.val(data.nextLine());

value = data.next();

//System.out.println("Gas"+(String)data.nextLine()+""+"Temp "+(String)data.nextLine());

//out.println( "Gas "+(String)data.nextLine()+""+"Temp "+(String)data.nextLine());

System.out.println((String)data.nextLine()+""+(String)data.nextLine());

out.println( (String)data.nextLine()+""+(String)data.nextLine());

//String a[]=data.nextLine().split(",");

//String temp=a[0];

//System.out.println(temp);

//String gas=a[1];

//System.out.println(gas);

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/manu","root","root");

// PreparedStatement ps=con.prepareStatement("insert into po(portid,tempid)values(?,?)");

// PreparedStatement ps=con.prepareStatement("insert into portval(heart\_temp,heart\_beat)values(?,?)");

PreparedStatement ps=con.prepareStatement("insert into data(valuess) values(?)");

ps.setString(1, (String) data.nextLine() + "" + (String) data.nextLine());

// ps.setString(2,(String)data.nextLine());

// ps.setString(1,temp);

// ps.setString(2,gas);

int x=ps.executeUpdate();

if(x!=0){

}

// try{value = Integer.parseInt(data.nextLine());}catch(Exception e){}

// slider.setValue(value);

}

catch(Exception e){

System.out.println(e);

}

out.close();

}

data.close();

System.out.println("Done.");

}

}

**5.1.2 Login**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import javax.swing.JOptionPane;

public class UserLog extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

String name=request.getParameter("name");

String pass=request.getParameter("pass");

HttpSession s=request.getSession();

s.setAttribute("name", name);

try{

int x=0;

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/manu","root","root");

PreparedStatement ps=con.prepareStatement("select \* from user\_reg where name=? and pass=?");

ps.setString(1, name);

ps.setString(2, pass);

ResultSet rs=ps.executeQuery();

while(rs.next()){

x++;

}

if(x!=0){

JOptionPane.showMessageDialog(null, "Login Success");

RequestDispatcher rd=request.getRequestDispatcher("UserHome.jsp");

rd.forward(request, response);

}else{

JOptionPane.showMessageDialog(null, "Login Failed");

RequestDispatcher rd=request.getRequestDispatcher("index.html");

rd.forward(request, response);

}

}catch(Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**5.1.3 Cloud Storage**

import Cloudme.CloudmeUser;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.swing.JOptionPane;

public class cloud extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

String path="D:/Port\_Read/port.txt";

String path1="D:/Port\_Read/enc/port.txt";

try{

encryption\_1 e=new encryption\_1();

e.encr(path);

}catch(Exception e){}

try{

CloudmeUser user=new CloudmeUser("dhanu","dhanusha");

user.getFileManager().uploadFile(path,"/port/");

user.killUser();

}catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

RequestDispatcher rd=request.getRequestDispatcher("UserHome.jsp");

rd.forward(request, response);

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**5.1.4 Embedded C Coding For PLC**

#include <htc.h>

#include "lcd16.h"

void txs(unsigned char val)

{

int i;

while(!TXIF)

continue;

TXREG=val;

for(i=0;i<600;i++);

}

unsigned char rxs(void)

{

int c=0;

while(!RCIF)

{

c++;

if(c>5000)

break;

}

return RCREG;

}

unsigned int val[6],flagx=0,val1,val2,val3,val4,val5,val6;

int j=0;

int x;

int sp,sp1,spt;

unsigned int d=0,e=0,f=0,flgg=0;

unsigned int m=0;

void Delay()

{

int i;

for(i=0;i<2000;i++)

{

}

}

void delay3(unsigned int val)

{

int i;

for(i=0;i<val;i++)

{

}

}

void sendtx(unsigned char val[],unsigned char length)

{

int i;

for(i=0;i<length;i++)

{

txs(val[i]);

}

}

/\* Sample code to set up the A2D module \*/

void init\_a2d(void){

ADCON0=0; // select Fosc/2

ADCON1=2; // select left justify result. A/D port configuration 0

ADON=1; // turn on the A2D conversion module

}

/\* Return an 8 bit result \*/

unsigned char read\_a2d(unsigned char channel){

channel&=0x07; // truncate channel to 3 bits

ADCON0&=0xC5; // clear current channel select

ADCON0|=(channel<<3); // apply the new channel select

Delay();

GO=1; // initiate conversion on the selected channel

while(GO)continue;

Delay();

return(ADRESH); // return 8 MSB of the result

}

void delay2()

{

long i;

for(i=0;i<200000;i++);

CLRWDT();

}

sendd1()

{

txs('A');txs('T');txs('+');txs('C');txs('M');txs('G');

txs('S');txs('=');

txs('"');

txs('8');txs('1');txs('2');txs('2');

txs('2');txs('8');txs('9');txs('9');txs('3');txs('7');

txs('"');

txs(13); txs(10);

delay2();

}

void sms1()

{

txs('T');txs('e');txs('m');txs('p');txs('e');

txs('r');txs('a');txs('t');txs('u');txs('r');

txs('e');txs(' ');txs('A');txs('l');txs('e');

txs('r');txs('t');

txs(' ');

txs('i');

txs('n');

txs(' ');

txs('E');txs('m');txs('e');txs('r');txs('g');

txs('e');txs('n');txs('c');txs('y');txs(' ');txs('i');

txs('n');txs(' ');txs('R');txs('o');

txs('o');

txs('m');

txs(':');

txs('1');txs('3');

txs(13); txs(10); delay(10000);

txs(26);

delay2();

delay2();

delay2();

delay2();

delay2();

}

void sms2()

{

txs('P');txs('r');txs('e');txs('s');txs('s');

txs('u');txs('r');txs('e');txs('A');txs('l');txs('e');

txs('r');txs('t');

txs(' ');

txs('i');

txs('n');

txs(' ');

txs('E');txs('m');txs('e');txs('r');txs('g');

txs('e');txs('n');txs('c');txs('y');txs(' ');txs('i');

txs('n');txs(' ');txs('R');txs('o');

txs('o');

txs('m');

txs(':');

txs('1');txs('3');

txs(13); txs(10); delay(10000);

txs(26);

delay2();

delay2();

delay2();

delay2();

delay2();

}

void sms3()

{

txs('H');txs('e');txs('a');txs('r');txs('t');

txs(' ');txs('B');txs('e');txs('a');txs('t');txs(' ');

txs('A');txs('l');txs('e');txs('r');

txs('t');

txs(' ');

txs('i');

txs('n');

txs(' ');

txs('E');txs('m');txs('e');txs('r');txs('g');

txs('e');txs('n');txs('c');txs('y');txs(' ');txs('i');

txs('n');txs(' ');txs('R');txs('o');

txs('o');

txs('m');

txs(':');

txs('1');txs('3');

txs(13); txs(10); delay(10000);

txs(26);

delay2();

delay2();

delay2();

delay2();

delay2();

}

unsigned char rval=2,rvalt=2,fl=0,val7;

int c;

void main() //Main entry

{

CMCON=0x07;

RCSTA=0x90; // receive enabling with the same speed

TXSTA =(0x24); // High speed selection baud rate 9600

SPBRG = 64;

CVRCON=0x00;

TRISD=0x00;

TRISE=0x00;

Lcdinit();

init\_a2d();

TRISB=0xFF;

TRISC=0x0F;

TRISD=0x00;

TRISE=0x00;

nRBPU=0;

while(1)

{

while(RCIF==0)

{

Delay();

val1=read\_a2d(0)+15;

Lcdcmd(0x80);

lcddata('T');

lcddata('e');

lcddata('m');

lcddata('p');

lcddata('e');

lcddata('r')

lcddata('a');

lcddata('t');

lcddata('u');

lcddata('r');

lcddata('e');

lcddata('=');

lcddata((val1/100)+0x30);

lcddata(((val1%100)/10)+0x30);

lcddata((val1%10)+0x30);

Delay();

val3=RC0;

if(val3==0)

{

if(fl==0)

{

val4++;

fl=1;

}

}

else

{

fl=0;

}

val5++;

if(val5>42)

{

if(val4>3)

{

val6=60+((val4%20)\*2);

}

else

{

val6=0;

}

val4=0;

val5=0;

}

Lcdcmd(0xC0);

lcddata('H');

lcddata('e');

lcddata('a');

lcddata('r');

lcddata('t');

lcddata(' ');

lcddata('B');

lcddata('e');

lcddata('a');

lcddata('t');

lcddata('=');

lcddata((val6/100)+0x30);

lcddata(((val6%100)/10)+0x30);

lcddata((val6%10)+0x30);

lcddata(' ');

lcddata((val4%10)+0x30);

sendtx("A",1);

txs((val1/100)+0x30);

txs(((val1%100)/10)+0x30);

txs((val1%10)+0x30);

sendtx("B",1);

txs((val6/100)+0x30);

txs(((val6%100)/10)+0x30);

txs((val6%10)+0x30);

sendtx("C",1);

Lcdcmd(0x80);

lcddata('P');

lcddata('r');

lcddata('e');

lcddata('s');

lcddata('s');

lcddata('u');

lcddata('r');

lcddata('e');

lcddata('=');

lcddata((val1/100)+0x30);

lcddata(((val1%100)/10)+0x30);

lcddata((val1%10)+0x30);

}

rval=RCREG;

RCIF=0;

OERR=0;

CREN=0;

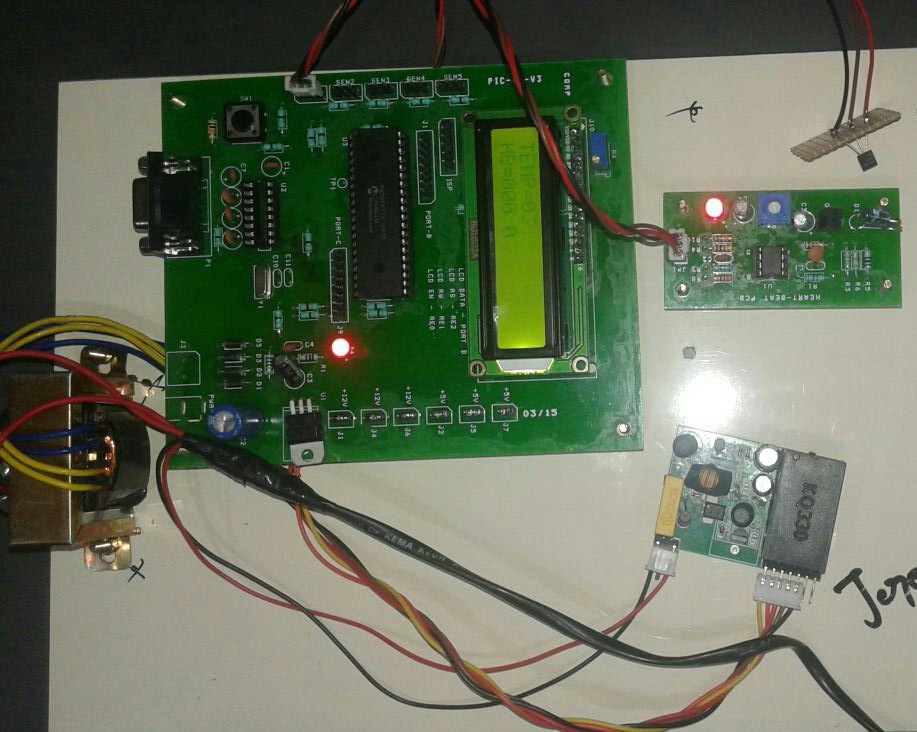
CREN=1;

}

}

**5.2 SAMPLE SCREEN SHOTS**

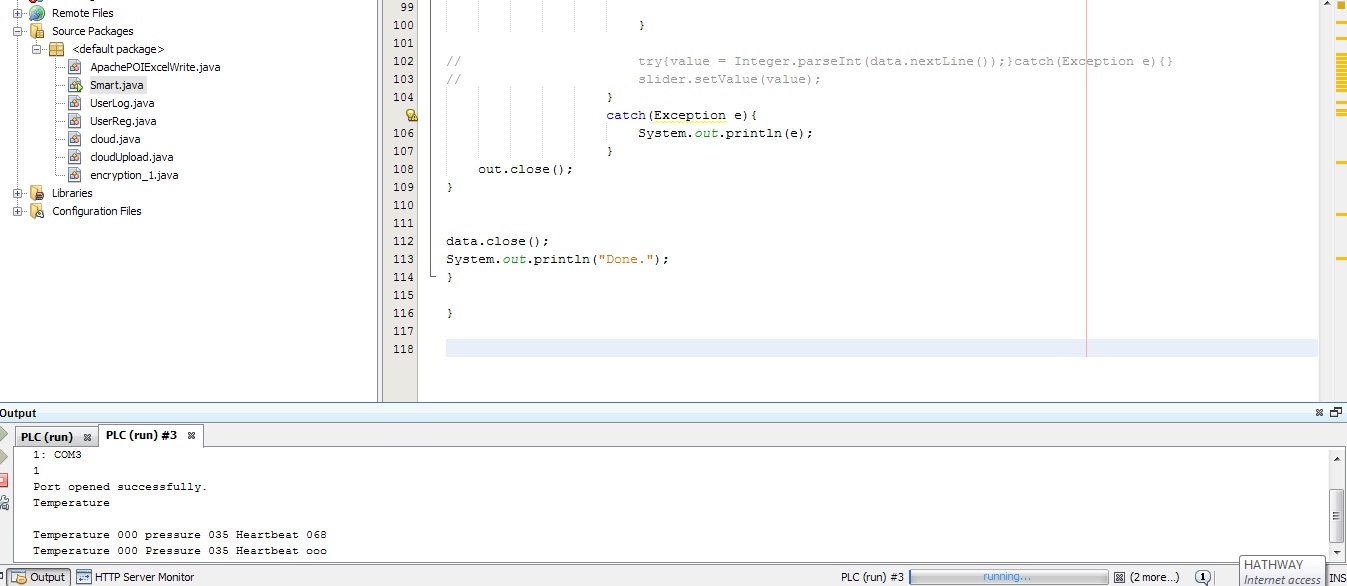
**5.2.1 Image of the Hardware Kit**

****

**Fig 5.1 Transmitter Kit**

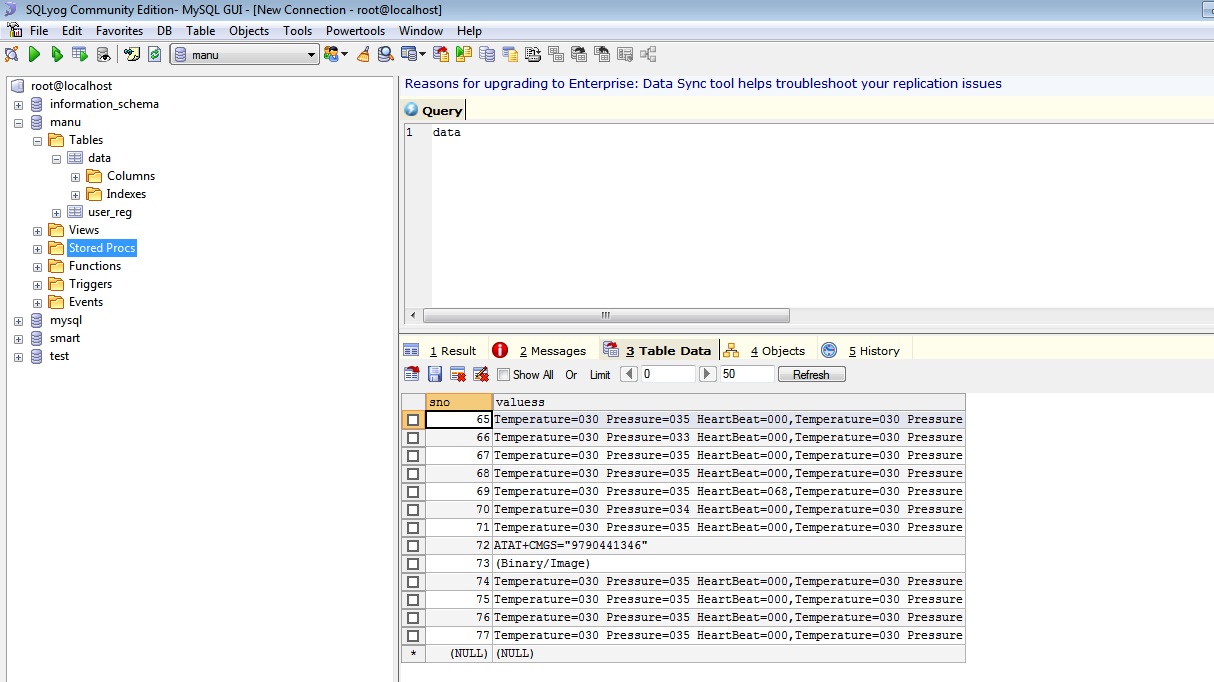
**Fig 5.2 Receiver Kit**

**5.2.2 Input Values**

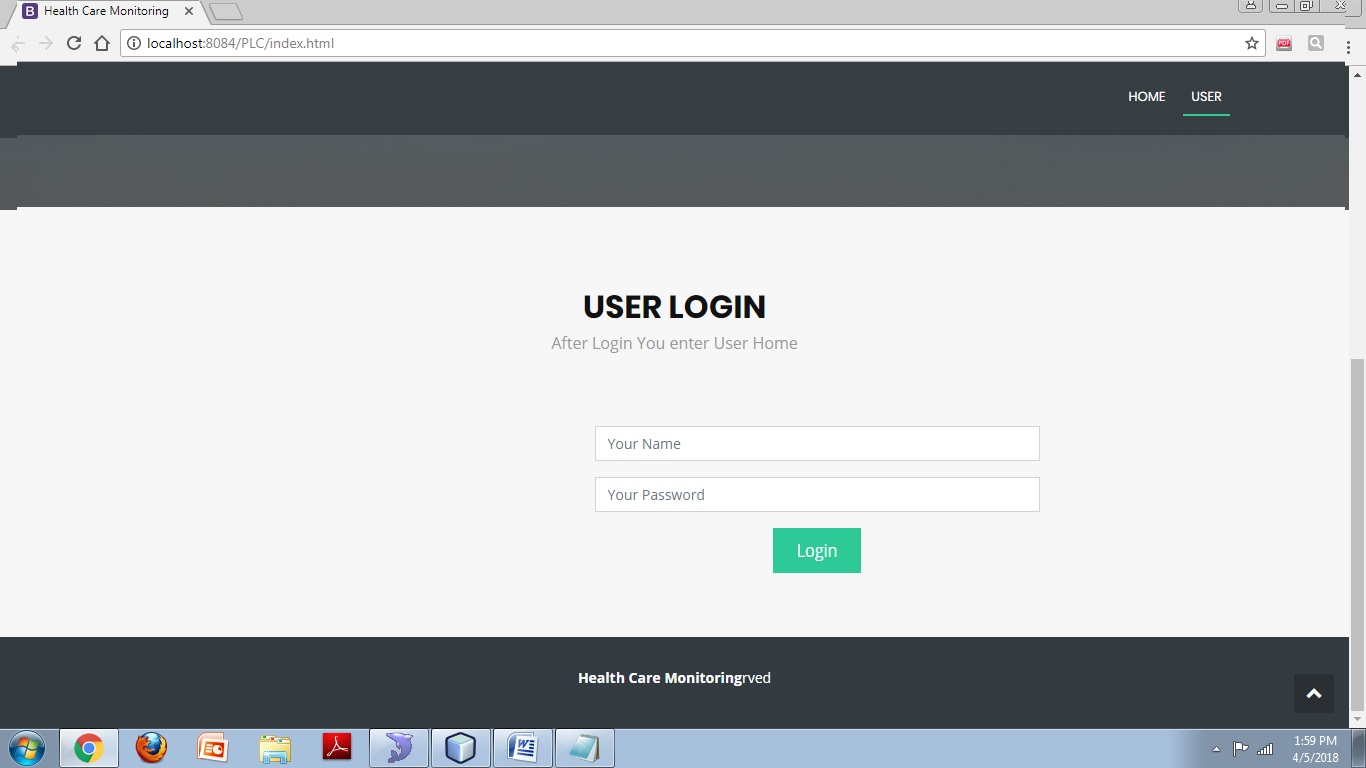
****

**Fig 5.3 Getting Input Values from Serial Port**

**5.2.3 Values Stored in Database**

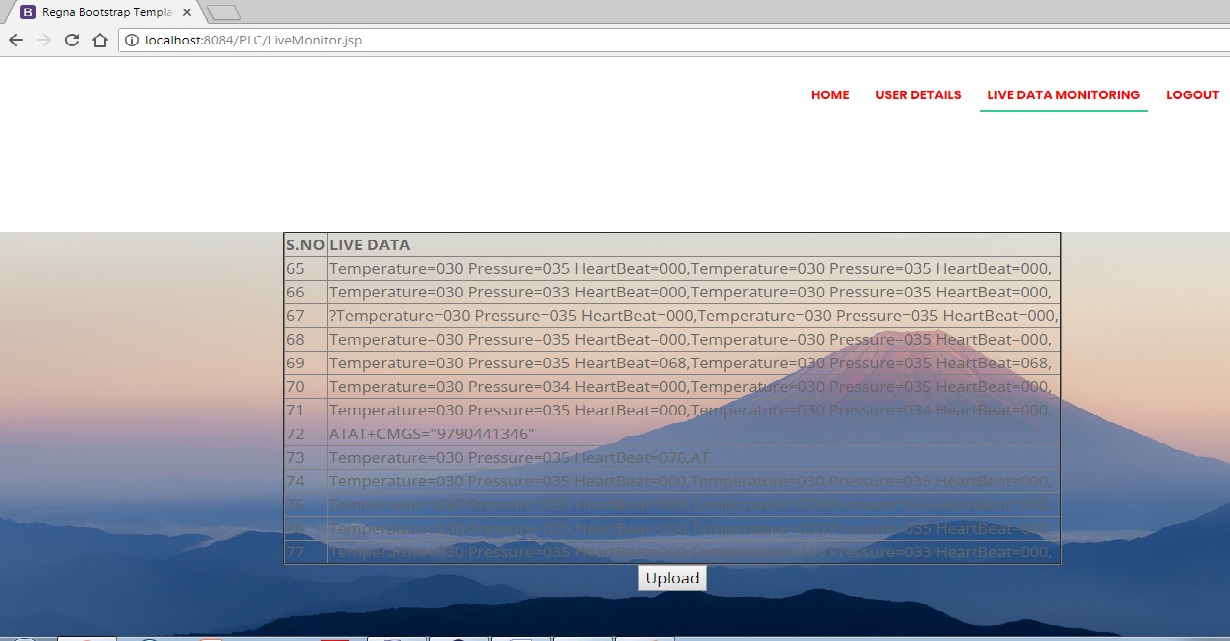
**Fig 5.4 Values Stored in Database Using MYSQL**

**5.2.4 User Login**

****

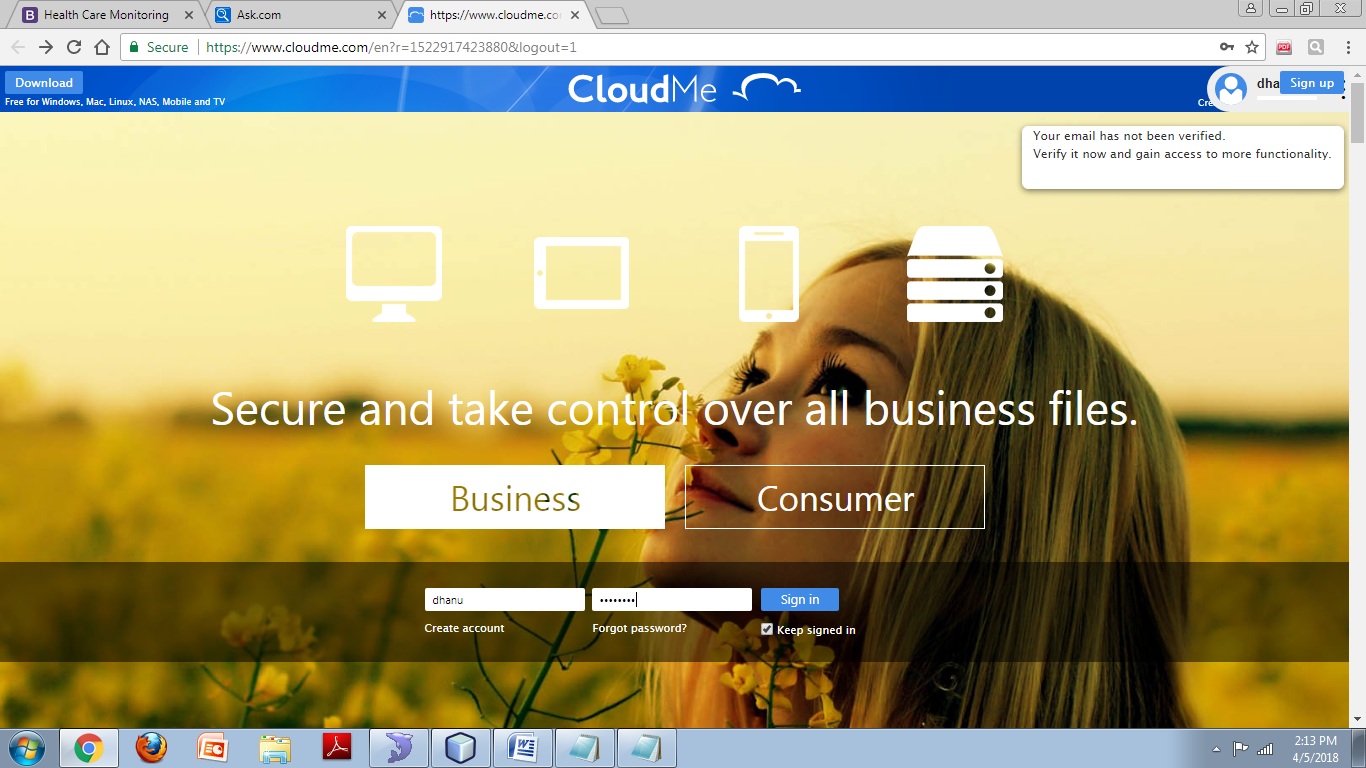
**Fig 5.5 User Login Page**

**5.2.5 Data Monitoring**

****

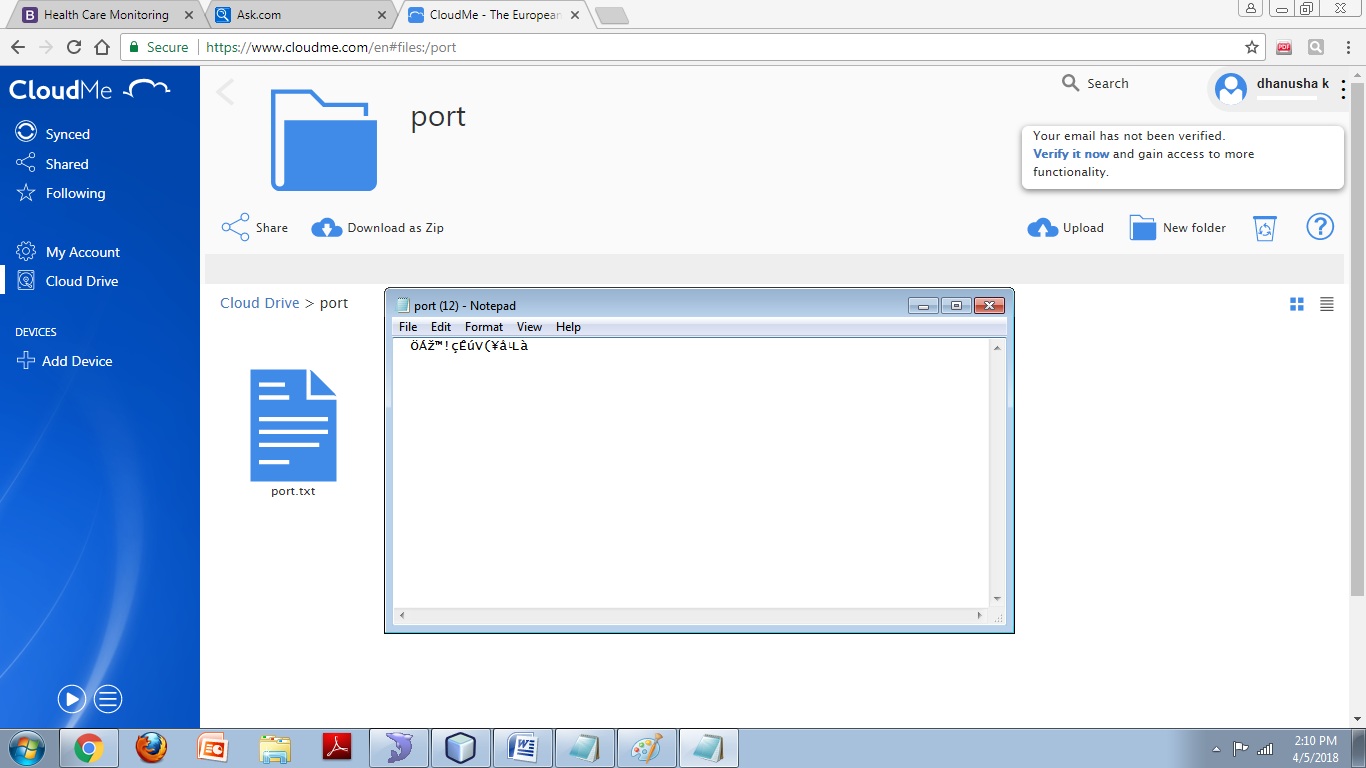
**Fig 5.6 Live Data Monitoring**

**5.2.6 Cloud Login**

****

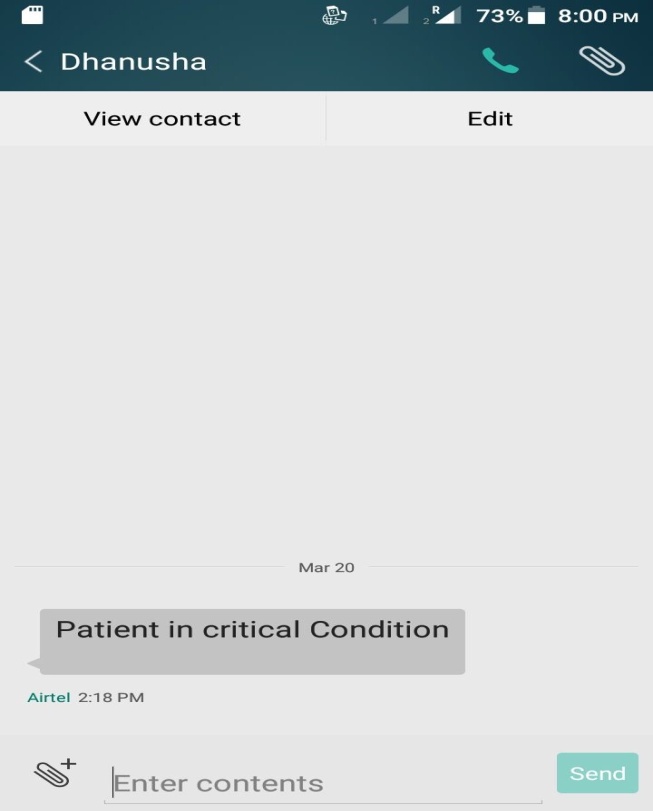
**Fig 5.7 Cloud Login Page**

**5.2.7 Data Encryption**

****

**Fig 5.8 Data Encryption**

**5.2.8 Alert Message**

****

**Fig 5.9 Alert Message to Phone**

**6.1 TESTING**

Software testing is a set of processes aimed at investigating, evaluating and ascertaining the completeness and quality of computer software. Software testing ensures the compliance of a software product in relation with regulatory, business, technical, functional and user requirements. Software testing is also known as application testing.

**6.1.1 Integration Testing**

By making unit testing for each module as explained above the process of integrated testing as a whole becomes simpler. This is because by correcting mistakes or bugs in each module the integration of all units as a system and testing process becomes easier. So one might think why the integration is testing needed. The answer is simple. It is needed because unit testing ad explained test and assures correctness of only each module. But it does not cover the aspects of how the system would behave or what error would be reported when modules are integrated. This is done in the level of integration Testing.

Integration testing in the software testing model comes before system testing and after the unit testing has been done. The way that integration testing works is by, getting the individual modules that have been through the unit testing phase and integrating each module into a group. The integration testing phase will make sure when the modules are being integrated together that any problems, for example errors or bugs, caused due to the integration of the modules are eliminated. Integration testing does not deal with the integration of the whole system but deals with the integration of a process in the system. In the integration testing stage there are three things that are created, to ensure that the integration of the modules is successful and that it runs successfully.

**Integration Test Plan**

When producing a test plan it must include the following information to be effective,

* A strategy to use when testing the integrated modules and how the tests will be conducted.
* What will be tested for example software features.
* What is the time scale and time management.
* Responsibilities, e.g. personnel.
* Testing pass and fail condition.
* Risk involved
* Approval from all important people involved.

**Integration Test Cases**

Test cases is created to make sure that the output, of the integrated modules are producing the expected output and is working exactly how it is supposed to work. This is simply a way to spot any errors or bugs that might have been made in the integration phase. The tester will then work through the program and document all the data using the test case that was created, the test case will test all inputs and outputs in the integrated modules.

**Integration Test Data**

Test data is simply data that is used in order to test the actual program or the integrated modules. Test data would normally be used in a test case as this would be used to check the inputs and expected outputs.

**Different types of Integration Testing**

There are some different types of integration testing that can be conducted, below is a list of the different integration testing types;

**Big Bang:**

The way this integration testing type works is, most or all of the modules are integrated together to form nearly a complete system. This is very similar to system testing as this basically has a whole system before starting the testing.

There are not many advantages of the big bang integration; the only advantage is that for a smaller system this will be the ideal integration testing technique.

The disadvantages is that you would have to wait for all the modules to be integrated in order to do big-bang testing so there will be quite a lot of delay. Any errors are identified at a very late stage and it is very hard to indentify the fault. It is very difficult to be sure that all testing has been done before product release.

**Top-Down testing:**

This is where the highest level components are tested first and then step by step start working downwards (lower components). Top-down testing mainly requires for the testing team to separate what is important and what is least important, then the most important modules are worked on first. The top-down approach is similar to a binary tree you would start off by integrating the top level before slowly working your way down the tree and integrating all the components at that level.

The advantage to this way of testing is that if a prototype is released or shown then most of the main functionality will already be working. It is also easy to maintain the code and there will be better control in terms of errors so most of the errors would be taken out before going to the next stage of testing.

The disadvantage is that it is hard to test the lower level components using test data. The other thing is that lower level modules may not be tested as much as the upper level modules.

**Bottom-up testing:**

This is the opposite to top-down where you test the lower level components and start testing your way upwards to the higher level components. The components will be separated into the level of importunacy and the least important modules will be worked on first, then slowly you would work your way up by integrating components at each level before moving upwards.

The advantage is that with this method you would be able to maintain code more easily and there is a more clear structure of how to do things.

The disadvantage is that when releasing a prototype you can not see a working prototype until nearly all the program has been completed so that may take a long time before this happens. There may be a lot of errors later on regarding the GUI and programming later on.

**Integration – Overall**

Integration testing is best to be used in an iterative process as this way it will save time in the long run and will help when trying to keep to a budget. The reason why a iterative process is the best for integration testing is simply because this allows for more feedback from the client, so if the client is involved with the project a lot it would be less likely in terms of having to make a lot of changes in the integration process, for example test plan.

**6.2 TEST CASES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case no** | **Description** | **Pre-conditions** | **Pass/Fail** | **Expected results** |
| PLC\_001 | Validate patient Registration | New patient only allowed | Pass | Registered Successfully |
| PLC\_002 | Validate patient login | Registered patient only allowed | Fail | Login Unsuccessfully |
| PLC\_002a | Validate patient login | Registered patient only allowed | Pass | Login successfully |
| PLC\_003 | Patient Set Sensor | Registered patient only allowed | Pass | Set Sensor process successfully |
| PLC\_004 | Patient Get the values of Sensor | Registered patient only allowed | Pass | Get Sensor Values successfully |
| PLC\_005 | Patient Check Values | Registered patient only allowed | Pass | Values Check successfully |
| PLC\_006 | Patient upload to Microcontroller | Registered patient only allowed | Pass | Upload Successfully |
| PLC\_007 | Send to PLCC Transmitter | Registered patient only allowed | Pass | Send Successfully |
| PLC\_008 | Send transmitter to receiver | Registered patient only  Allowed | Pass | Send Successfully |
| PLC\_009 | Doctor login | Authentication must | Pass | Login successfully |
| PLC\_010 | Doctor Get the values of Sensor | Doctor Only Allowed | Pass | Get Sensor Values successfully |
| PLC\_011 | View PC | Doctor Only Allowed | Pass | View PC successfully |
| PLC\_012 | Get Result | Registered patient only allowed | Pass | Buzzer ON Successfully |
| PLC\_013 | Result Maintain by Doctor | Doctor Only Allowed | Pass | Maintain Successfully |

**Class testcase program**

package Testing;

public class Registration {

int c;

public int Login()

{

String username = new String("DHANUSHA");

String password = new String("y");

if ((username.equals("DHANUSHA")) || (password.equals("y")))

{

c=2;

}

else

{

c=3;

}

return c;

**Junit program**

package pro;

import static org.junit.Assert.\*;

import org.junit.Test;

public class Pro1 {

@Test

public void test() {

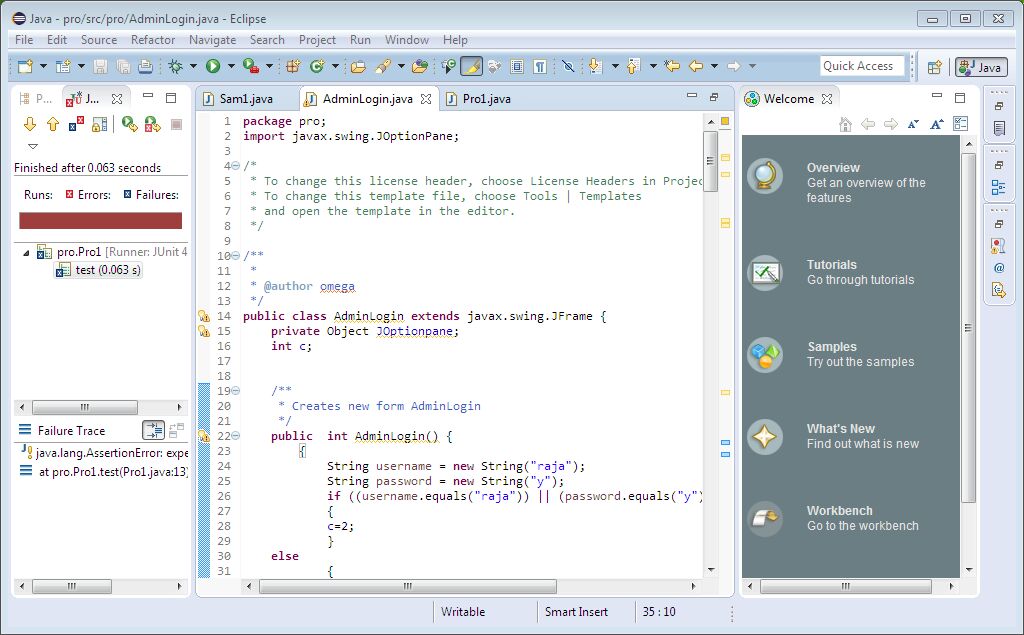
AdminLogin test=new AdminLogin ();

int out= test.AdminLogin();

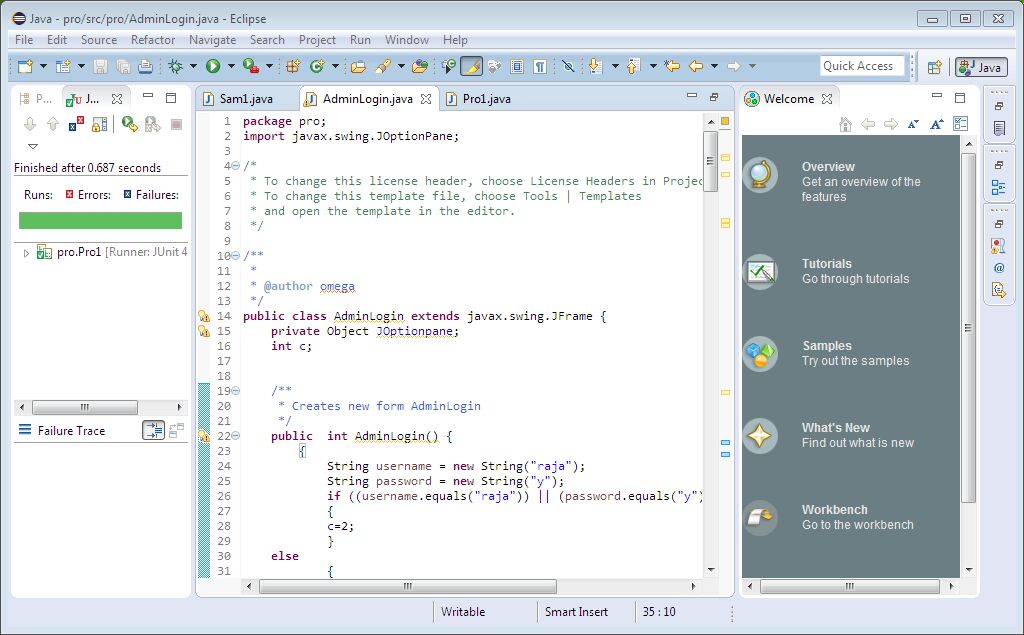
assertEquals(3,out);

}

}

****

**Fig 6.1 Screenshot for Login Failure**

****

**Fig 6.2 Screenshot for Login Success**

**6.3 MAINTENANCE**

After systems have been verified, tested and implemented, they must continue to be maintained to ensure that they continue to perform correctly and that they can adapt to new requirements if needed.

Ongoing monitoring or testing of systems may need to be systematised to ensure that maintenance needs are identified and met when necessary. Where systems are for extended use, a mechanism can be put in place to monitor feedback from users as another means to determine the need for maintenance and modification.

Maintenance routines vary depending on the type and complexity of the technology. Many items come with a maintenance schedule or program recommended by the manufacturer or supplier. For some hardware and software, maintenance is provided by the manufacturer or supplier as part of the purchase agreement.

Where modifications to hardware, software and/or communications are made as a result of maintenance or upgrades, it may be necessary to conduct further rounds of system verification and testing to ensure that requirements meet the same or updated specifications.

**7.1 CONCLUSION**

PLC is a technique that allows exchange of data by means of electric power supply network that are presented in every dwelling, office and in every building. Since the development and research on the subject of Power line communications is relatively new, the primary reason for this project is to serve as a reference for PLC technologies, products and standards. The power line technology is getting stabilized and attracting many industry leading vendors. Thus as a prototype level, we developed a healthcare based power line communication system integrating Internet of Things (IoT).

**7.2 FUTURE ENHANCEMENT**

After having tried with several practical X10 experiments, further steps will be explore other power line systems (like fibrenet lines) to test their potentialities, also conducting a comparative study on the power consumes of home and building automation systems.

**REFERENCE**

[1] Xavier Carcelle,Power line communication in practice.

[2] ETSI, [www.etsi.org/WebSite/Technologies/Powerline.aspx](http://www.etsi.org/WebSite/Technologies/Powerline.aspx)

[3] Galli, S., the Inter Inter-Co-PHY Protocol (IPP): A Simple Co Existence Protocol. IEEE International Symposium on Power Line Communications, Dresden, Germany, 31 March 2009.

[4] Galli, S. & Oleg, L., Recent developments in the standardization of power line communications in the IEEE. IEEE Communications Magazine, pp. 64–71, July 2008.

[5] Mains Network, <http://en.wikipedia.org/wiki/Mains_network>.

[6] IEEE P1901, Draft Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications, PUA and UPA contribution to IEEE 1901 call for submission on BPL access requirements Functional and Technical Requirements, pp. 1–41, February 2006. Online [www.upaplc.org](http://www.upaplc.org)

[7] Sathya, R., Broadband over power lines: a rapid way to join the knowledge society. Africa/Middle East Next Generation Network Summit, Johannesburg, South Africa, 17–19 February 2008. Online [www.istpowernet.org](http://www.istpowernet.org)

[8] PLC [www.plcmanual.com/plc.programming](http://www.plcmanual.com/plc.programming)

[9] Introduction to Programmable Controllers [www.plcdev.com](http://www.plcdev.com)

[10] Library <https://library.automationdirect.com/what-is-a-plc>