HOME AUTOMATION AND AGRICULTURAL MONITORING SYSTEM

A Mini Project Report

Submitted by

ANUBHAV TEWARI (2012EEC06)
ANKIT TYAGI (2012EEC07)
RAHUL SHARMA (2012EEC22)
SHIVAM KHARE (2012EEC24)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

At

विज्ञानं ब्रह्म
Shri Mata Vaishno Devi University

SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING
SHRI MATA VAISHNO DEVI UNIVERSITY
KATRA

MAY 2016

STUDENT DECLARATION

ABSTRACT

Today's generation of electronic devices are more enhanced and capable than the previous ones with exciting changes in technology has seen to control a variety of home devices with the help of a home automation system. These devices can include lights, fans, doors, surveillance systems and consumer electronics. However along with the smartness and intuitiveness we want a system which is economic as well as low power consuming.

ZigBee technology collects and monitors different types of measurements that reflect energy consumption and environment parameters. This paper details the designing of a protocol to monitor various environmental conditions in a home. We are using advanced technology of Micaz motes (which have their own routing capabilities), NESC language programming and Moteworks (used as a data acquisition platform).

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**ii**
	ACKNOWLEDGMENT	**iii**
1.	**INTRODUCTION**	
	1.1 GENERAL	1
	1.2 HOME AUTOMATION	2
	1.3 WSN	13
	1.4 NS3	15
2.	**LITERATURE REVIEW**	16
	2.1 MOTEWORKS KIT	17
	2.2 NS3	
	2.3 ROUTING PROTOCOLS	19
	2.4 ROUTING ALGORITHMS	
3.	**WORK DONE**	
	3.1 Installation and Configuration	
	3.2 NS3 code design	
	3.3 SIMULATION	
4.	**HARDWARE AND SOFTWARE REQUIREMENTS**	
	4.1 MOTEWORKS & MOTEVIEW	
	4.2 NS3	

INTRODUCTION

1.1 GENERAL

Life nowadays is full of desire of comfortability. A perfect system can be which obeys your command or does it automatically using voice commands or Apps in the mobile phones.

In the home environment there are several home networks, such as: Internet access, computers and devices interconnection, audio and video distribution, security and surveillance system, energy saving systems, automation and control systems which are increasingly becoming a requirement.

A wireless sensor network plays a very important part in these scenarios related to band width, data rate, maintenance and installation requirements, being usually a solution with lower implementation costs. The home environment automation are developed with the intention to automate processes pertaining to home appliances and various processes associated with them for facilitating higher comfort and safety to the inhabitants.

New generation electromechanical automation systems aim at real time data processing and transmission to a remote location for monitoring and sensing which aids in the domain of knowledge to the end user about the respective region where the system is currently deployed. Various communication standards and platforms have been devised in previous decades to turn such concepts into reality along with tremendous progress in semiconductor and MEMS technologies over the past decade has led to production of inexpensive sensors and microcontroller platforms which can be easily interfaced with each other. Many such startups are now also dealing with these technologies and rolling out products which have unique features like insignificant power consumption resulting in prolonged battery life, low maintenance modules with added short circuit prevention, encrypted data transmission standards for software level security and use of biometric data of intended user for 0 counterfeiting rate.

Present day approaches for home automation systems includes improvement of basic device interoperability through research and standards as well as monolithic system to integrate multiple devices for specific tasks. This was known through extensive paper readings from other respective journals and conference publications. The remote sensing options were already available but we have pursued the aim of creating automation and manual control too. The project was developed keeping account of current barrier, providing an adaptable, low cost system that the end user can oneself install, configure, upgrade and control, consisting of server and diverse systems. Contemporary approaches include remote controlling and sensing which can be implemented using convectional devices like smartphones and personal computers. These solutions provide facility to end user to remotely access their home. Additionally, latest features can be introduced using future update roll outs to make system self-adaptive.

1.2 HOME AUTOMATION

The "Home Automation" concept has existed for many years. The terms "Smart Home", "Intelligent Home" followed and has been used to introduce the concept of networking appliances and devices in the house. Home automation Systems (HASs) represents a great research opportunity in creating new fields in engineering, and Computing.

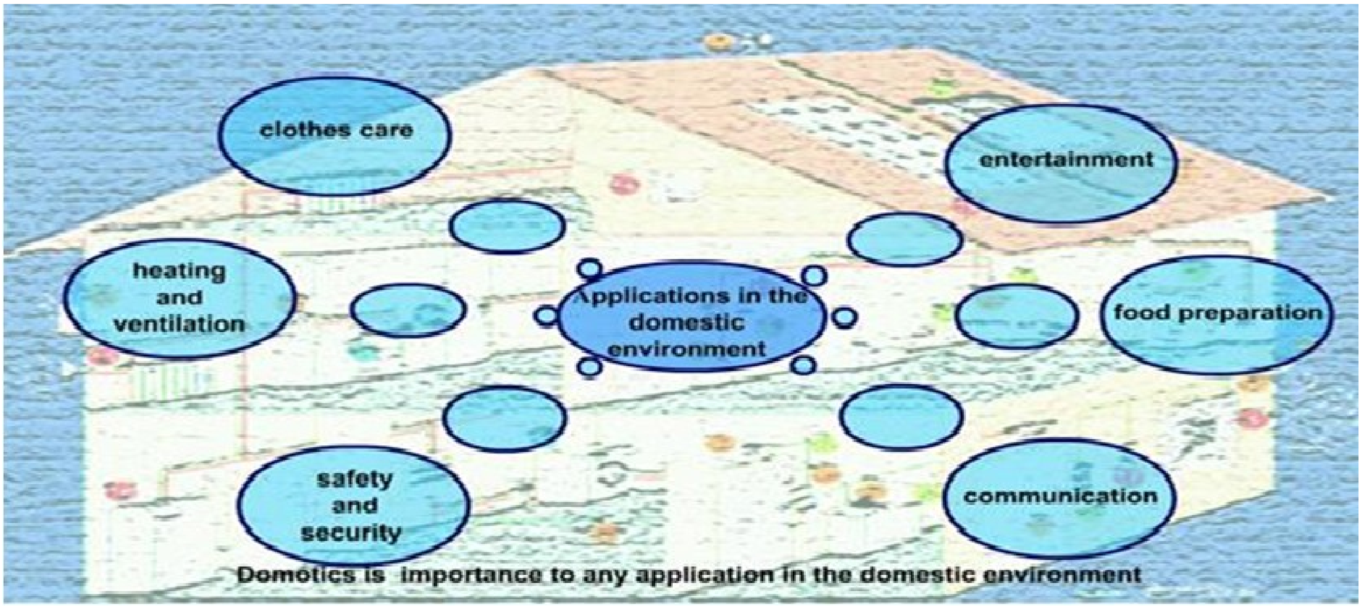


Figure 1

HASs includes centralized control of lighting, appliances, security locks of gates and doors and other systems, to provide improved comfort, energy efficiency and security system. HASs becoming popular nowadays and enter quickly in this emerging market. Home automation involves introducing a degree of computerized or automatic control to certain electrical and electronic systems in a building. These include lighting, temperature control, etc.

This project demonstrates automated system based on Wireless Sensor Networking which can be accessed and controlled at any remote location resulting in real time responses. This purposes is achieved using MOTEWORKS, a data acquisition platform developed by Memsic Inc. which supports mesh networking and mote monitoring to store the data into a PSQL ODBC database.

In this project an energy efficient and self-adaptive environment is being implemented to reduce the manual requirement and enhance the output efficiency by providing accurate results in a regular span of time resulting in better analysis and precise actions. Here in case of a system failure an additional feature remote sensing and controlling using smart phones has been added to secure the system working. System is self-efficient to gather data from displaced nodes at the base station where MDA300CA has been interfaced which operates upon Tiny OS Software and respond accordingly in form of appliance controlling on the basis of the values obtained and the threshold value already set without any external assessment. Our prime target is to establish an environment in which any kind of break in, theft or any malfunction such as fire, gas leakage and water leak can be easily detected and the requisite actions can be taken instantly.

1.3 AGRICULTURAL MONITORING AND CONTROL

The advanced development in wireless sensor networks can be used in monitoring various parameters in agriculture. Due to uneven natural distribution of rain water it is very difficult for farmers to monitor and control the distribution of water to agriculture field in the whole farm or as per the requirement of the crop. There is no ideal irrigation method for all weather conditions, soil structure and variety of crops cultures. Farmers suffer large financial losses because of wrong prediction of weather and incorrect irrigation methods. In this context, with the evolution of miniaturized sensor devices coupled with wireless technologies, it is possible remotely monitor parameters such as moisture, temperature and humidity. In this project it is proposed to design, develop and implement a wireless sensor network connected to a central node using ZigBee, which in turn is connected to a Central Monitoring Station (CMS) through General Packet Radio Service (GPRS) or Global System for Mobile (GSM) technologies. The system also obtains Global Positionting System (GPS) parameters related to the field and sends them to a central monitoring station. This system is expected to help farmers in evaluating soil conditions and act accordingly.

The system consists of wireless sensor network nodes and network management platform. Zigbee node (1 to n) respectively transmits acquisition of the temperature and humidity data to the Zigbee stations of gateways node. The automatic networking realizes through the many jump routing consumption, flexible automatic networking temperature humidity monitoring system of soil. And the system is a complete set of wireless sensor network induction, acquisition, storage, application, reporting, solution, has a good man computer exchange interface. Users need not go into farmland, in a corner anywhere in the world, could prompt understand the changing condition of farmland soil temperature and humidity, and scientifically guide agricultural production.

1.4 WIRELESS SENSOR NETWORKS

Wireless sensor network (WSN) is a wireless network which consists of equally distributed autonomous devices using sensors capable of monitoring the physical or environmental conditions such as temperature, sound, vibration, pressure, motion or pollutants, at various remote locations especially for buildings in campus. It comprises of an ad-hoc network composed by hundreds, even thousands, of wireless sensor nodes.In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery.

Depending on the specific application, sensor nodes are deployed into some sensing field so that they can collaborate with each other and form a wirelessnetwork and this sensor network is connected to the outside network through a base station. Generally a WSN consists of a base station (or “gateway”) that can communicate with a number of sensors via a radio network Data is collected at the wireless sensor nodes or motes, compressed, and transmitted to the gateway directly or, if required, uses other wireless sensor nodes to forward data to the gateway. The transmitted data is then presented to the system by the gateway connection to user PC.

1.4 .a ZIGBEE

The 802.15.4 IEEE platform is a non-infrastructure based protocol serves the foundation for ZigBee platform which specifies the physical and media access control layer standards for LR-WPAN networks. It’s maintained and published by ZigBee Alliance which provides support for upper networking layers according to the SoC intended.

This mesh networking targeted standard works on ISM band frequency range (2.4 GHz, 714 MHz, 868 MHz, 915 MHz) with data rate encompassing 20 kbps (868 MHz) to 250 kbps (2.4 GHz). Numerous routing protocols like flooding, TEEN, APTEEN, LEACH have been proposed but till now no perfect one has been accepted by the industry. The device class is divided into physical and logical types with further into FFD and RFD which serve unique purposes individually. The nodes act as whether in sensing or control mode then it’s under RFD and FFD’s are responsible for routing functions catering child devices under the cluster with FFD as cluster head.It finds wide applications in fields of animal tracking, home automation, industrial control, medical health monitoring and numerous others.

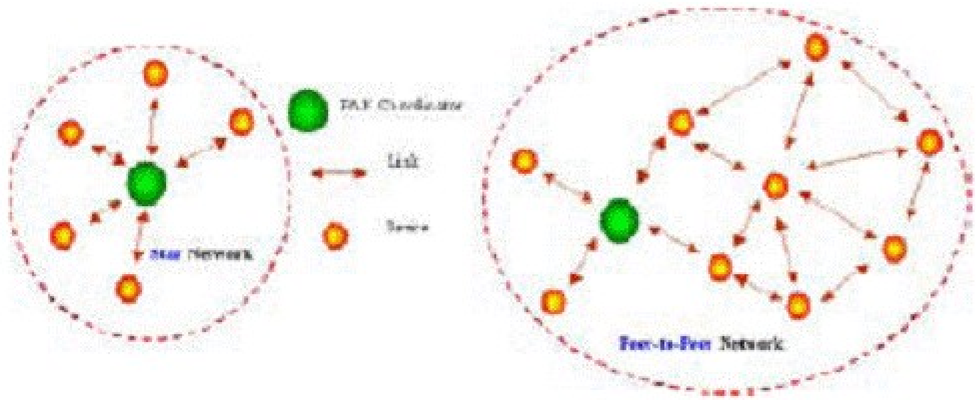


Figure 1.2: ZIGBEE NETWORK TOPOLOGY(reference ??????????????????)

2. LITERATURE REVIEW

2.1 ROUTING PROTOCOLS

2.1.a LEACH

LEACH Protocol is the first protocol of hierarchical routings which proposed data fusion, it is of milestone significance in clustering routing protocols. When wireless sensor networks gradually go into our lives, it is of great significance to research on LEACH protocol.

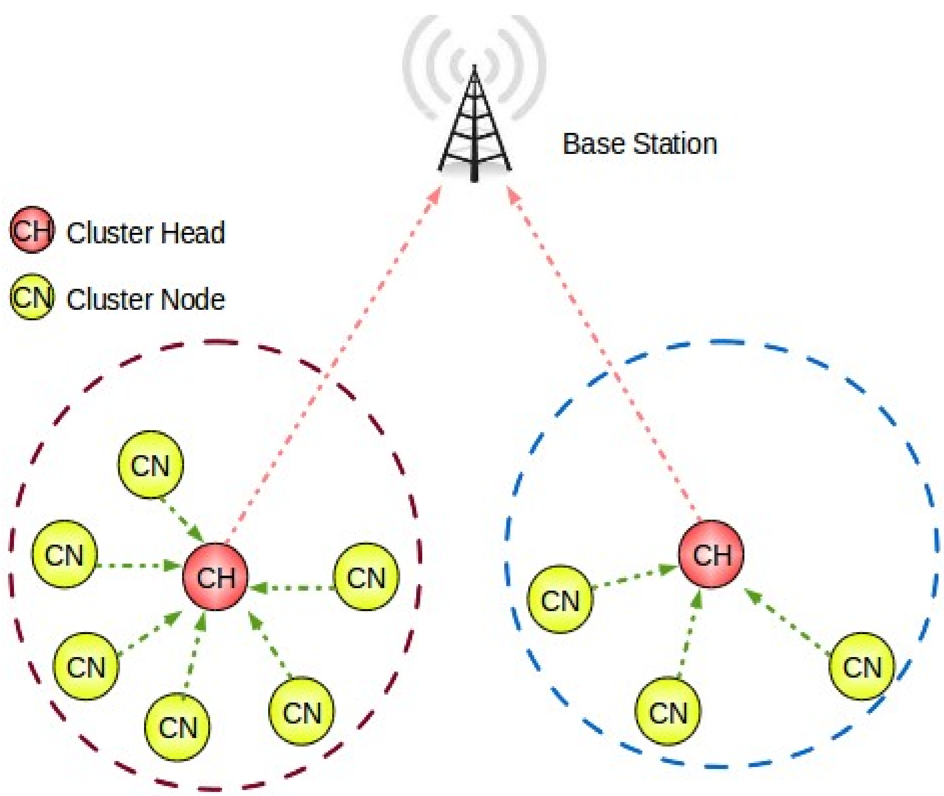


Figure 3 : Leach protocol (ref ??????????)

LEACH Protocol is a typical representative of hierarchical routing protocols. It is self adaptive and self-organized. LEACH protocol uses round as unit, each round is made up of cluster set-up stage and steady-state stage, for the purpose of reducing unnecessary energy costs, the steady state stage must be much longer than the set-up stage.

$$t(n) = \begin{cases} \dfrac{p}{1 - p \bullet (r \bmod \frac{1}{p})} & if\ n \in G \\ 0 & if\ n \notin G \end{cases}$$

Where p is the percentage of the cluster head nodes in all nodes, r is the number of the round, G is the collections of the nodes that have not yet been head nodes in the first 1/P rounds. Using this threshold, all nodes will be able to be head nodes after 1/P round s. The analysis is as follows: Each node becomes a cluster head with probability p when the round begins, the nodes which have been head nodes in this round will not be head nodes in the next 1/P rounds, because the number of the nodes which is capable of head node will gradually reduce, so, for these remain nodes, the probability of being head nodes must be increase d. After 1/P-1 round, all nodes which have not been head nodes will be selected as head nodes with probability 1, when 1/P rounds finished, all nodes will return to the same starting line. When clusters have formed, the nodes start to transmit the inspection data. Cluster heads receive data sent from the other nodes, the received data was sent to the gateway after fused. This is a frame data transmission. In order to reduce unnecessary energy cost, steady stage is composed of multiple frames and the steady stage is much longer than the set-up stage.

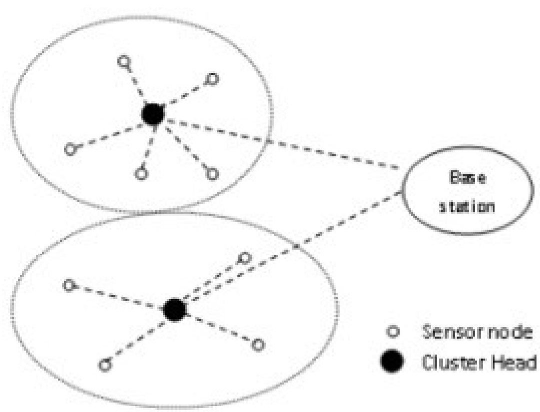


Figure 1.4: Clustering in LEACH protocol

(ref ?????????????????)

The different advantages that the LEACH protocols having are as follows:

• It provides scalability in the network by means of limiting most of the communication inside the different clusters of the network.

- The cluster heads aggregates or fuses the information that is been collected by the sensor nodes and this helps in to limit high amount of traffic generated within the network. By this means, a large-scalable network without traffic overload can be deployed and by this also better energy efficient network topology can be achieved as compared to the flat-topology.

- Single-hop routing is possible from sensor node to cluster head, and by this means we can able to save the energy of the network.

- Distributiveness property within the cluster, where it distributes the role of CH to the other cluster members within the cluster.

- It increases the lifetime of network in three phases. First, it distributes the role of CH (which consumes more energy than normal sensor nodes) to the other nodes in the cluster. Second, it aggregates the sensed information by the CHs. Finally, by the process of TDMA, (which is been assigned by the CH to its members) puts most of the sensor nodes in the sleep mode. This is done especially in event-based applications only. By this means, it is able to increase the network lifetime and also able to achieve a more than 7- fold reduction of energy dissipation compared to direct communication.

- It does not require the information of location of the sensor nodes in the network to create the clusters. Therefore it is very powerful routing protocol and it is very much simple also.

- It gives the dynamic clustering approach. It is well-suited for applications where constant monitoring of the environmental information is needed and data collection process occurs periodically to a centralized location of the network.

The different disadvantages that the LEACH protocols having are as follows:

- It significantly relies on cluster heads rather than cluster members of the cluster for communicating to the sink. Due to this it incurs robustness issues like failure of the cluster heads.

- It incurs additional overheads due to the process of cluster head changes in each iterations of the communication of information. It also incurs overhead due to calculations which leads to the energy inefficiency for dynamic clustering in large scale networks.

- There is no inter-cluster communication in the network because CHs directly communicate with sink. This process requires high range of transmission power in the network. For this only, LEACH is not best suited for large- scale networks that interns require single hop communication with sink.

- In LEACH CHs are not uniformly distributed within the cluster that means CHs can be located at the edges of the cluster.

- In LEACH, CH selection is random process, which does not take energy consumption of the different nodes within the cluster along with CH into account and this leads to reselecting of CH as the sane node in many simultaneous iteration of data processing in the network.

- It does not work well with the applications that require large area coverage along with multi-hop inter-cluster communication.

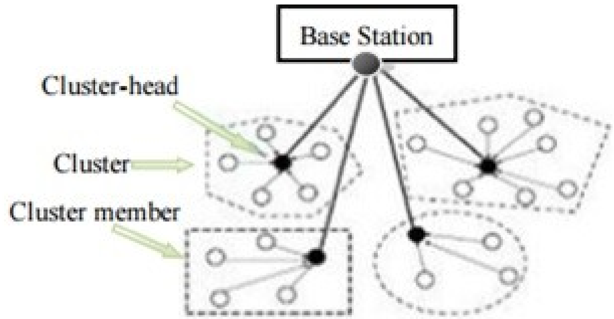


Figure 1. 5 : LEACH protocol(ref ???????)

2.1.b SPIN Protocol

Sensor Protocols for Information via Negotiation (SPIN) proposed a family of adaptive protocols called Sensor Protocols for Information via Negotiation (SPIN) that disseminate all the information at each node to every node in the network assuming that all nodes in the network are potential base-stations. This enables a user to query any node and get the required information immediately.

These protocols make use of the property that nodes in close proximity have similar data, and hence there is a need to only distribute the data that other nodes do not possess. The SPIN family of protocols uses data negotiation and resource-adaptive algorithms. Nodes running SPIN assign a high-level name to completely describe their collected data (called meta-data) and perform meta-data negotiations before any data is transmitted. This assures that there is no redundant data sent throughout the network. The semantics of of the meta-data format is application-specific and is not specified in SPIN. The SPIN family is designed to address the deficiencies of classic flooding by negotiation and resource adaptation.

The SPIN family of protocols is designed based on two basic ideas:

1. Sensor nodes operate more efficiently and conserve energy by sending data that describe the sensor data instead of sending all the data; for example, image and sensor nodes must monitor the changes in their energy resources.
2. Conventional protocols like flooding or gossiping based routing protocols waste energy and bandwidth when sending extra and un-necessary copies of data by sensors covering overlapping areas. The drawbacks of flooding include implosion, which is caused by duplicate messages sent to the same node, overlap when two nodes sensing the same region will send similar packets to the same neighbor, and resource blindness by consuming large amounts of energy without consideration for the energy constraints. Gossiping avoids the problem of implosion by just selecting a random node to send the packet to rather than broadcasting the packet blindly. However, this causes delays in propagation of data through the nodes.

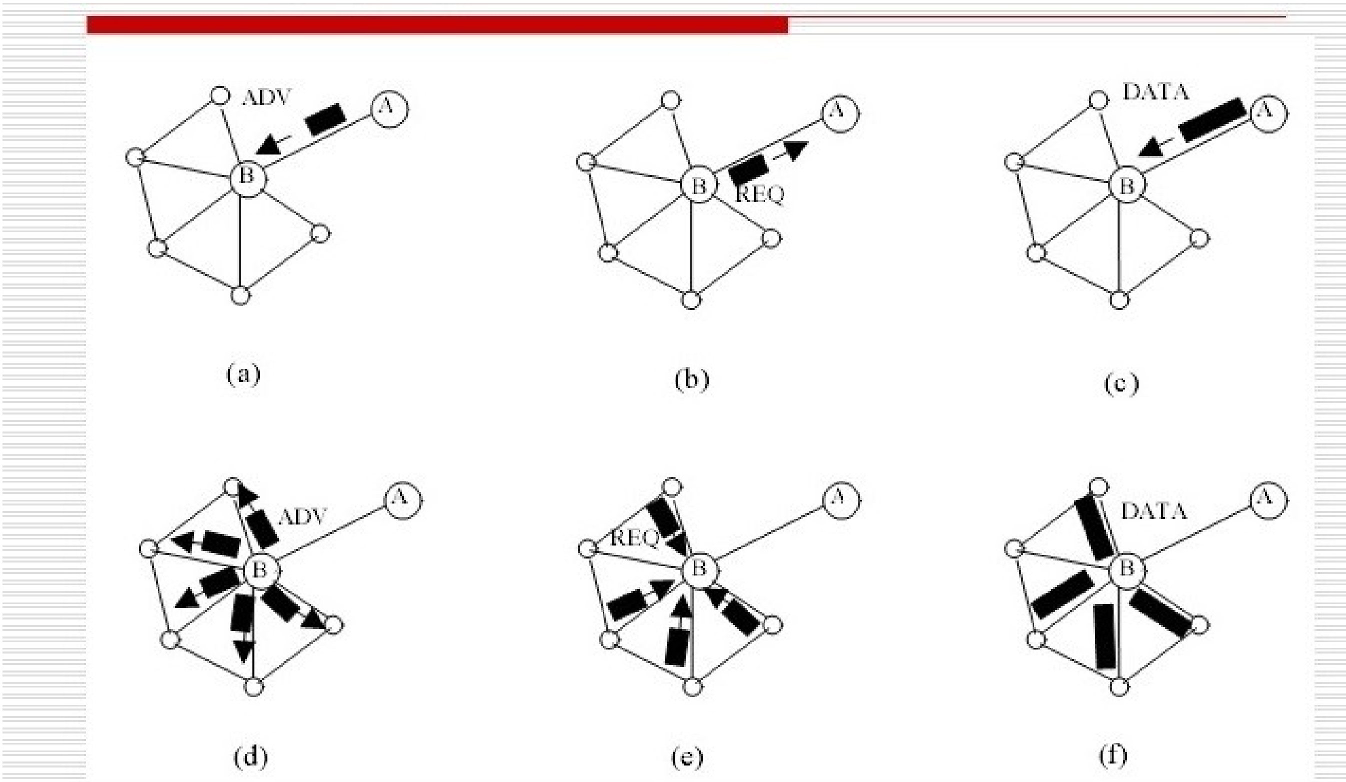


Figure 1.6: SPIN protocol (ref ?????????????????)

SPIN's meta-data negotiation solves the classic problems of flooding, and thus achieving a lot of 8 energy efficiency. SPIN is a 3-stage protocol as sensor nodes use three types of messages ADV, REQ and DATA to communicate. ADV is used to advertise new data, REQ to request data, and DATA is the actual message itself. The protocol starts when a SPIN node obtains new data that it is willing to share. It does so by broadcasting an ADV message containing meta-data. If a neighbor is interested in the data, it sends a REQ message for the DATA and the DATA is sent to this neighbor node. The neighbor sensor node then repeats this process with its neighbors. As a result, the entire sensor area will receive a copy of the data.

The SPIN family of protocols includes many protocols. The main two protocols are called SPIN-1 and SPIN-2, which incorporate negotiation before transmitting data in order to ensure that only useful information will be transferred. Also, each node has its own resource manager which keeps track of resource consumption, and is polled by the nodes before data transmission.

The SPIN-1 protocol is a 3-stage protocol, as described above. An extension to SPIN-1 is SPIN-2, which incorporates threshold-based resource awareness mechanism in addition to negotiation. When energy in the nodes is abundant, SPIN-2 communicates using the 3-stage protocol of SPIN-1. However, when the energy in a node starts approaching a low energy threshold, it reduces its participation in the protocol, i.e., it participates only when it believes that it can complete all the other stages of the protocol without going below the low-energy threshold. In conclusion, SPIN-1 and SPIN-2 are simple protocols that efficiently disseminate data, while maintaining no per-neighbor state. These protocols are well-suited for an environment where the sensors are mobile because they base their forwarding decisions on local neighborhood information. Other protocols of the SPIN family:

– SPIN-BC: This protocol is designed for broadcast channels.

– SPIN-PP: This protocol is designed for a point to point communication, i.e., hop-by-hop routing. – SPIN-EC: This protocol works similar to SPIN-PP, but with an energy heuristic added to it.

– SPIN-RL: When a channel is lossy, a protocol called SPIN-RL is used where adjustments are added to the SPIN-PP protocol to account for the lossy channel.

One of the advantages of SPIN is that topological changes are localized since each node needs to know only its single-hop neighbors. SPIN provides much energy savings than flooding and metadata negotiation almost halves the redundant data. However, SPINs data advertisement mechanism cannot guarantee the delivery of data. To see this, consider the application of intrusion detection where data should be reliably reported over periodic intervals and assume that nodes interested in the data are located far away from the source node and the nodes between source and destination nodes are not interested in that data, such data will not be delivered to the destination at all.

2.2 ROUTING ALGORITHMS

A **routing algorithm** specifies how routers communicate with each other, disseminating information that enables them to select routes between any two nodes on a computer network. Routing algorithms determine the specific choice of route. Each router has a pre knowledge only of networks attached to it directly. A routing protocol shares this information first among immediate neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network.

Routing protocols were created for routers. These protocols have been designed to allow the exchange of routing tables, or known networks, between routers. There are a lot of different routing protocols, each one designed for specific network sizes.

Two main types of routing:

- **Static routing**
- **Dynamic routing**

The router learns about remote networks from neighbor routers or from an administrator. The router then builds a routing table. If the network is directly connected then the router already knows how to get to the network. If the networks are not attached, the router must learn how to get to the remote network with either static routing (administrator manually enters the routes in the router's table) or dynamic routing (happens automatically using routing protocols like EIGRP,OSPF,etc.).

The routers then update each other about all the networks they know. If a change occurs e.g a router goes down, the dynamic routing protocols automatically inform all routers about the change. If static routing is used, then the administrator has to update all changes into all routers and therefore no routing protocol is used.

Only Dynamic routing uses routing protocols, which enable routers to:

- Dynamically discover and maintain routes
- Calculate routes
- Distribute routing updates to other routers
- Reach agreement with other routers about the network topology

Statically programmed routers are unable to find routes, or send routing information to other routers. They send data over routes defined by the network Admin.A Stub network is so called because it is a dead end in the network. There is only one route in and one route out and, because of this, they can be reached using static routing, thus saving valuable bandwidth.

Dynamic versus Static Routing

	Dynamic routing	Static routing
Configuration Complexity	Generally independent of the network size	Increases with network size
Required administrator knowledge	Advanced knowledge required	No extra knowledge required
Topology changes	Automatically adapts to topology changes	Administrator intervention required
Scaling	Suitable for simple and complex topologies	Suitable for simple topologies
Security	Less secure	More secure
Resource usage	Uses CPU, memory, link bandwith	No extra resources needed
Predictability	Route depends on the current topology	Route to destination is always the same

Dynamic Routing Protocols

There are 3 types of Dynamic routing protocols, these are differ by the way that discover and make calculations about routes;

1. Distance Vector

2. Link State

3. Hybrid

- Distance Vector routers find the best path from information send from neighbors
- Link State routers each have a copy of the entire network map
- Link State routers find best routes from this local map

The Table below shows the main characteristics of a few different types of dynamic routing protocols:

Routing Protocols Features	**RIPv1**	**RIPv2**	**IGRP**	**OSPF**	**EIGRP**
Classful/Classless	Classful	Classless	Classful	Classless	Classless
Metric	Hop	Hop	Composite	Cost	composite
Time Period	30 sec	30 sec	90 sec	None	30 sec
Administrative Distance(AD)	120	120	100	110	External: 170 Internal:90
Type	Distance vector	Distance vector	Distance vector	Link state	Hybrid

Table 1.1 : Dynamic routing protocols

3. DETAILED PROBLEM STATEMENT

Home Automation Primer

A home automation system is a means that allow users to control electric appliances of varying kind. Home automation is also known as domotics, a contraction of the words “domestic robotics”. When home automation principles are applied to buildings not falling in the “home” category, building automation system is a commonly used term.The control unit communicates with the actuator to tell how much current to let through to the lamp. . The wireless communication between the remote, control unit, and the actuator is done using a home automation communications protocol, e.g. ZigBee or Z-Wave.

3.1 Benefits and Obstacles

This section presents the potential benefits from home automation and then looks at some of the obstacles, that are somewhat hindering these benefits.

3.1.a Potential Benefits

The potential benefits we can gain from home automation are almost only limited by imagination and as such it would be infeasible to create a comprehensive list of them. The short list below exemplifies potential benefits in four areas of home automation. The examples are meant to spark the imagination.

Energy Savings Through user tracking both in- and outdoors, a home automation system would potentially be able to make sure that, for example, no unnecessary light or heat is turned on in individual rooms.

Convenience Trough Web based access to the home automation system a forgetful user will potentially no longer have to worry about if the coffee machine was left on when he left for work. Simply go to a Web page, check it, and turn it off if necessary.

Security Tracking user locations can assist in automatic alarm system arming. Also, security cameras might be accessed from a vacation to check that the house is alright.

Home Entertainment When engaging in movie watching, the lights might be set to an appropriate dimming level. When listening to music, speakers might be changing from room to room for your listening pleasure throughout the house. Digital paintings on the wall might change according to persons currently occupying the room.

3.1.b Obstacles

For most of the examples the technology required to realize them already exists: People can be tracked with Bluetooth, RFID chips, or digital people counters, used at some supermarkets and conferences. Some home automation systems feature Web based access to domestic appliances and alarm systems. Other home automation systems come with home entertainment integration, featuring control of television and stereo sets. With the technology being available, the question is what obstacles are hindering all of the above examples from being common in a setup.

Proprietorship Many of the systems TV, stereo, surveillance camera, etc. are proprietary and as such each have their own programmatic interface that control them, or none at all. Thus to obtain a system able to handle the examples, the buyer has to seek out a home automation vendor that specializes in custom home automation solutions and likely has to buy a whole range of appliances that the vendor is endorsing. This introduces a high cost due to the amount of work required to realize these systems. High cost means that home automation is less likely to become a common household system, unfortunate for both home automation vendors and households.

Extensibility Even if the buyer has acquired such a custom system, there is no guarantee that it can be extended with completely new, yet home automation related, features. For instance, the buyer might later purchase a system able to keep track of his refrigerator by means of a camera enabled mobile phone and software able to recognize bar codes. This might be a functionality that the buyer would like to add to his home automation system, much like he would install a new program on his computer, but most likely will be unable to due to a complicated, or even completely incompatible, system structure.

Standardization To obtain an extensible system of home automation related devices, that system must provide an agreed upon standard for device communication. One that companies providing proprietary systems are willing to implement. ZigBee, which is an open source communication protocol, is said to have boosted wireless sensor network standardization. Another problem with these kinds of protocols is that they require special hardware to function. In the case with the mobile phone application for keeping track of the refrigerator, a ZigBee, or equivalent, chip might not be available.

3.2 Hypothesis

Summing up the problems described in the previous section into one word yields: communication. Domestic appliances such as TVs, stereo systems, lights, heaters, etc. have no standardized common communication platform and consequently it becomes difficult to extend a home automation system to include new features. It is possible to create a standardized communication platform, that is able to handle communication between many different kinds of home automation related systems.

The overall idea for the hypothesized platform is to lift the abstraction level for communicating with domestic devices, thus lift the level for inter-device communication as well as human to device communication. By lifting the abstraction level for communication it is possible to overcome standardization problems with low-level protocols, thus facilitating extensibility and (hopefully) promoting adoption. The major challenge in lifting the abstraction level is to do it in an already standardized way, otherwise the platform will be too hard to use. The way that this project attempts to meet this challenge is described in following chapters.

Project Goals

The goals for this project can be divided into the following steps:

1. Define requirements for a system that accommodates hypothesis.

2. Identify more than one way in which the system can me implemented.

3. Compare the alternatives and make an informed choice between them.

4. Confirm that the implemented system fulfills the defined requirements through testing.

5. Evaluate the system with established requirements in mind and state possible limitations and suggest future work in the area.

4. PROPOSED SOLUTION

With an upward trend in Home Automation we require a system capable of simultaneously sensing and monitoring the environment and acting accordingly in real time to provide safe and secure surroundings. EarlyHome Automation Systems specifically relied on appliance interfacing and controlling but now the prime focushas shifted to get a secure and energy efficient environment which is being realized using WSN and ZigBee technology.

In this project, ZigBee platform along with MICAZ motes + MPR 2400CA sensor module has been implemented to obtain a self-forming non infrastructure based network which can use various kinds of topologies as suited and capable of monitoring surroundings on a continuous time lapse and can take decisions as per requirement. In this project the major concern of “manual control” has been eliminated as the system is smart enough to act on its own in time of need and take required decisions. MPR 2400CA sensor module acts as media gateway and receives all the data from the respective nodes and transfer that data to the server using Wi-Fi network, only user can have access to the server and will be full aware about the happenings taking place at his place and can continuously control the system by giving commands in real time, being present anywhere in the world and also get push notifications on smartphone for any activities. Server allows user to act instantaneously in case of anyunwanted activity and can also notify authorities in case of emergency if required.

The other requirement is to have energy efficient and cost effective system so that user won’t have to worry about maintenance and power consumption cost. MICAZ motes being used here are specifically designed to consume less power and have an extended working life. These motes are specifically designed to be deployed in remote locations and can bear any weather conditions and provides measured output to base station in real time.

The forthcoming substantial advancement in sensor nodes will help to overcome the issues related to wireless sensor network with improved fault tolerance, context awareness, power management, Quality of Service and security aspects. The upcoming developments in sensor nodes will produce relevant devices which may be used in applications like cognitive sensing, spectrum management, time-crucial systems, mobile micro-machines,smart rotating building control, structural health monitoring, environment friendly and adaptive systems, cold chain management.

5. ROAD MAP OF THE PROJECT

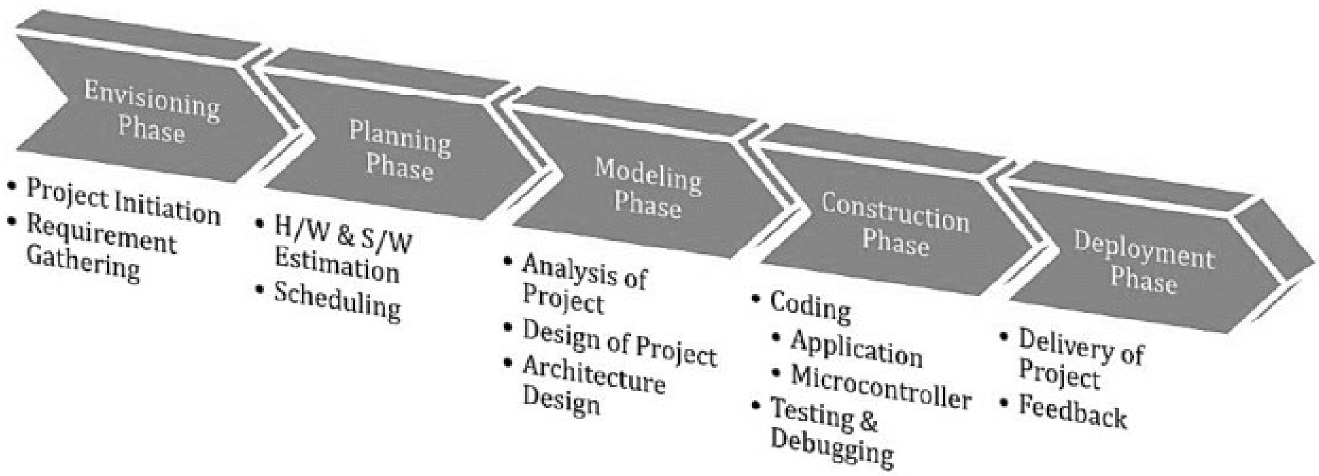


Figure 9 : (ref ???????????????????)

In the first month of the project prime focus was to study and analyze the preexisting systems in the market to get a clear picture about the parameters and features that will be taken care of. Various websites were browsed and different systems were compared and a list was prepared to mention their pros and cons and a model was designed covering all the necessary features that are going to be implemented in the project. Most of the home automation systems primarily concerned about the appliance interfacing and controlling, giving less importance to security and remote access to user so in this project security issue was given top priority.

In the second month the platform on which the project is going to be implemented was decided as there were many adaptive platforms were available in the market but the MEMSIC kit was most comprehensive one consisting all the required hardware and software which supports mesh networking and mote monitoring to store the data into a PSQL ODBC database. The overall software kit is divided into 3 separate tiers: mote, server and client. Mote tier supports multi-hop, non-infrastructure, ad-hoc, mesh networking protocol for LR-WPAN wireless networks based on ZigBee. The motes communicate via multi hop communication for improved reliability and radio coverage which are connected to PC via MIB520CB gateway which is equipped with antenna to remotely program motes and the comes under Xmesh Mote Tier.

In the third month project architecture was constructed comprising of all the scenarios undertaken to take care of such as appliance interfacing and controlling, automated gate control, access to server, secure access to home using password, notification on smart phone via ZigBee and remote access to system. Project was designed to cover all the basic as well as user defined requirements keeping in mind the cost efficiency and power regulations. MICAZ motes are designed to work in any sort of surroundings and seemed apt for the project.

In the consecutive months all the components were programmed and interfaced with the MIB520CB using various software such as Mote View, Mote Config, Tiny OS provided by the MEMSIC KIT and for simulation purpose used NS3 .Earlier project was focused on Home Automation but in last phase the projects application were expanded in to agriculture field to measure the quality of soil and to check the water level. Different configuration of motes were deployed using wireless sensor network topologies such as SPIN, LEACH and the simulation results were analyzed to decide the most suitable for project.

In the final phase a web server was developed using Arduino and it helps to monitor the entire system constantly by keeping the user aware of the surroundings. Server stores all the data transmitted by the motes and enables the user to act from any remote location. In the final phase a fully automated Home Automation System has been designed fulfilling all the needs of the user and acting in real time.

6. HARDWARE

A. MEMSIC WSN KIT

A.1 MICAZ Mote

MICAZ mote is specifically designed for wireless sensor networks having its own router capabilities to communicate with the surrounding nodes and use IEEE 802.15.4 (specifies the physical layer and media access control for low rate wireless personal area network) protocol for setting up a low manufacturing cost and power efficient, battery-operated networks.

Figure ?????????????????

MICAZ mote comprises of MPR2400CA platform based on the Atmel ATmega128L microcontroller which uses its internal flash memory to run mote works. MPR2400CA platform simultaneously communicates with surrounding nodes and runs sensor applications. MICAZ motes uses wireless ad hoc networking using mesh topology to set up an autonomous network and provides data rate of 250 kbps among nodes and base station to interface wide range of external peripherals.

A.2 MIB520CB

MIB520CB acts as an USB connector for MICAZ motes for communication and in system programming purposes. MICAZ motes when connected with MIB520CB acts as a base station and helps to collect data from all other motes present in the system. MIB520CB extend two different kind of ports one for in system mote programming and another for data communication over USB with baud rate of 57.6K and when connected with USB port doesn't require external power source.

A.3 MTS420

MTS420 is a surrounding monitoring sensor which can be easily deployed in remote locations as it requires very low maintenance and have an extended battery life. MTS420 provides wide range of features such as temperature/ humidity/ pressure (300mbar to 1100mbar) sensors and light intensity, along with dual axis accelerometer. Along with above mentioned features MTS420 also have GPS module to get coordinates of the motes.

Fig ????????????????

A.4 MDA300CA

MDA300CA is an extremely flexible data acquisition board having temperature and humidity sensors embedded on its platform and offers a wide range of features on board such as ADC channels, digital I/O channels, two relay channels (one open and one closed) and supports external I2C interface. MDA300CA have 64K EEPROM to store the data measured by the sensors. MDA300CA operates using Tiny OS software.

Fig ???????????????????

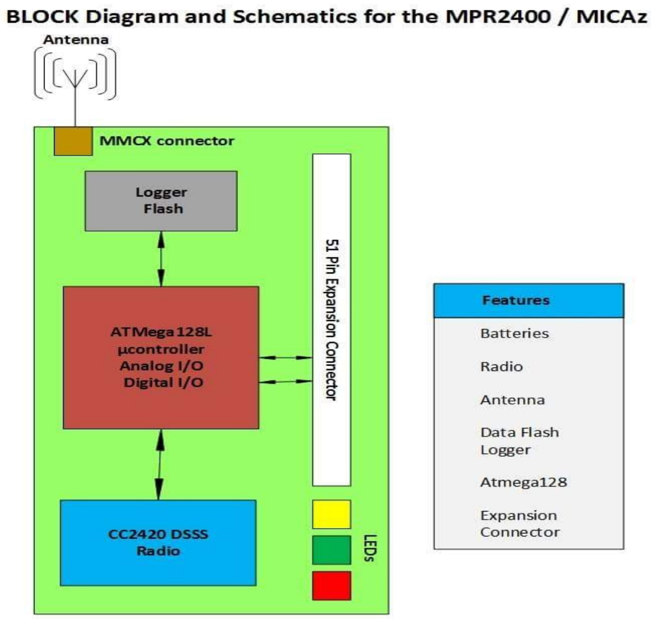


Figure : MPR2400 component block view (ref ???????????)

SOFTWARES AND TOOLS

A. MOTEWORKS

This is a data acquisition platform developed by Memsic Inc. which supports mesh networking and mote monitoring to store the data into a PSQL ODBC database. The overall software kit is divided into 3 separate tiers: mote, server, client. Mote tier supports multi-hop, non-infrastructure, ad-hoc, mesh networking protocol for LR-WPAN wireless networks based on ZigBee. The motes communicate via multi hop communication for improved reliability and radio coverage which are connected to PC via MIB520CB gateway which is equipped with antenna to remotely program motes and the comes under Xmesh Mote Tier.

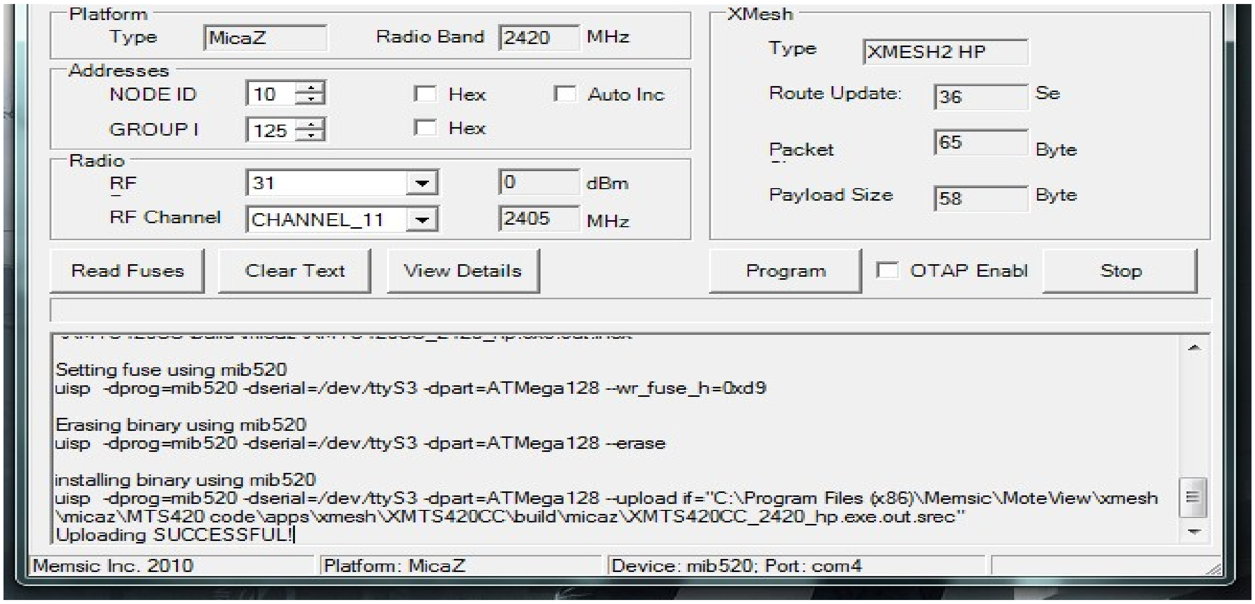


Fig. 2 MoteConfig(ref ???????????)

Server tier manages SQL database under applications interfacing with mesh to higher level layers and outside apps via terminal exchange.

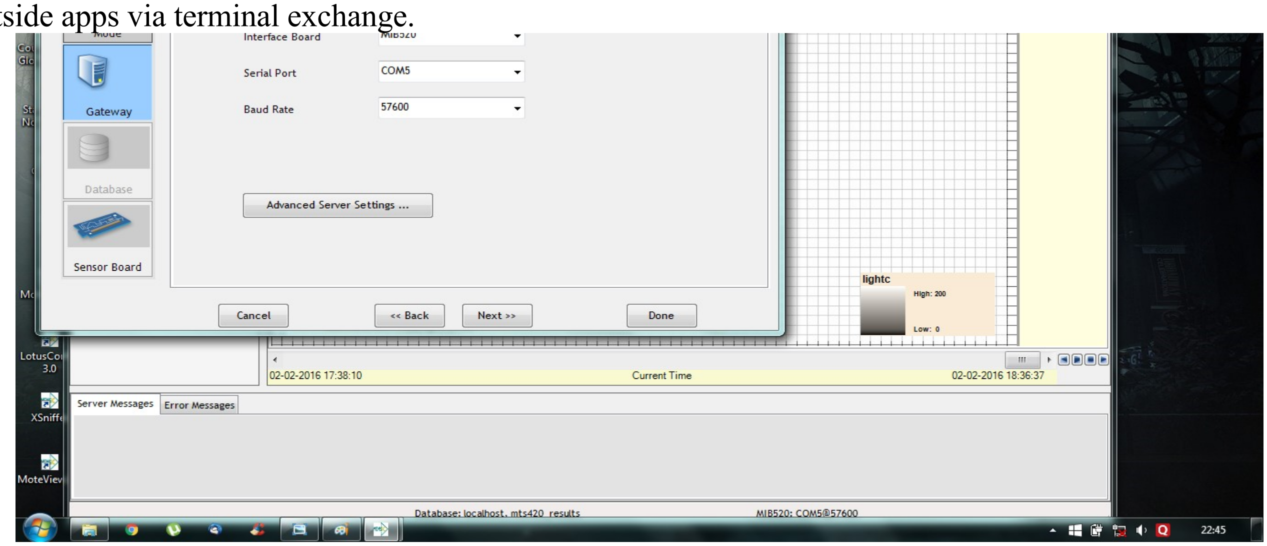


Fig. 3 MoteView(ref ???????????????????)

For end to end solution across all the tiers to the user or developer, the Client tier comes to the service which displays statistical information straight from sensors in form of text or graphic charting with the capability of rendering past event readings through fetching database. Individually node can be updated and configured based on sensor board attached and communication channel attached respectively through programming via gateway.

The software package provided by Memsic can be subdivided into the respective components as follows:

TinyOS and MoteWorks tools	An event-driven OS for wireless sensor networks; tools for debugging.
nesC compiler	An extension of C-language designed for TinyOS
Cygwin	A Linux-like enivreonment for Windows
AVR tools	A suite of software development tools for Atmel's AVR processors
Programmer's notepad	IDE for code compilation and debugging
XSniffer	Network monitoring rool for RF environment
Moteconfig	GUI environment for Mote Programming and OTAP
LotusConfig	GUI environment for Lotus Programming
Graphviz	To view files made from make docs

Table no. 1

B. TinyOS

The environment selected for the current work in TinyOS which is supports by the MEMSIC kit. The protocol shown here suits the network and routing requirements of the design and the deployment of motes. The whole operation is shown by the application layer which is in direct contact with the user or administrator. The advantage of using the TinyOS is that it is open source and is easily available in the internet. The language used here is NesC which is used to program the motes. The TinyOs is easily compatible with the motes, is event-driven in nature. The libraries used here are available default and are used in the code by direct programming and including them.

As per the data of 2010, three IDEs (integrated development environments) are available for the TinyOS, to be used in Eclipse:

- YETI 2, ETH Zurich
- XPairtise
- TinyDT

These plugins are included in Eclipse in order for them to run.

The version of TinyOS used for this project is 2.1.2.

The range of using microprocessors in TinyOS is right from 8-bit architecture to 32-bit architecture and from 2KB RAM to 32MB RAM (or more) respectively.

C. NESC

For programming motes, a new language ecosystem has been created which is focused on a component and interface driven methodology. It provides an easier linking model than C and derives it's features from C, C++ and Java with the aim of building components just like in Java objects for which can be compiled into complete concurrent systems for robust embedded network systems.

Program writing involves writing components and wiring them often known as interfaces occurring at compile time and bidirectional in nature. It builds up into a concurrency model for monitoring hardware event handlers and tasks as per the respective sensor module.

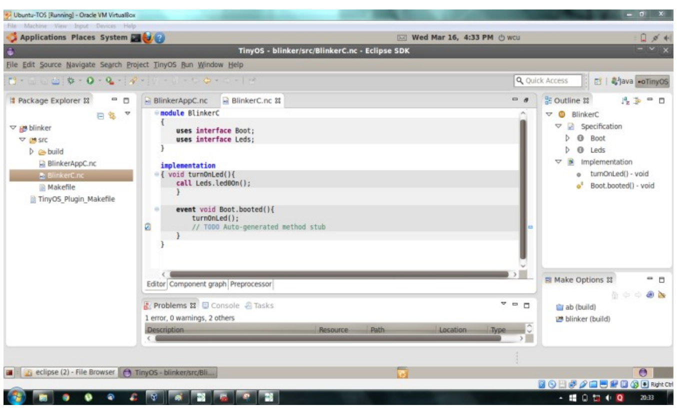


Fig. 4 Eclipse + Yeti2 running in UbunTos virtual machine

The NesC was run into a UbunTos (Ubuntu + TinyOS) virtual machine with Eclipse installed alongwith Yeti2 Plugin installed for NesC syntax recognition and compilation.

MATHEMATICAL MODELLING

Numerous parameters are dealt with during discussion about Zigbee transmission technology. Some of them are :

1. Battery Consumption of onboard radio – power usage of a Zigbee radio can be broken down into a combination of it's unique states.

$p_{con} = p_{tx} + p_{rx} + p_{Sleep} + p_{idle}$ (1)

Where p= E/t

2. The battery lifetime in hours could be calculated using the following formula:

$t = \frac{i_c}{i^n}$

3. Power loss which is crucial in designing battery extensive systems, can be calculated for communication technologies which utilize ISM band is:

$PL(dB) = -10\log\left(\frac{G_T G_R \lambda^2}{(4\pi)^2 d^2}\right)$

Symbol Table:

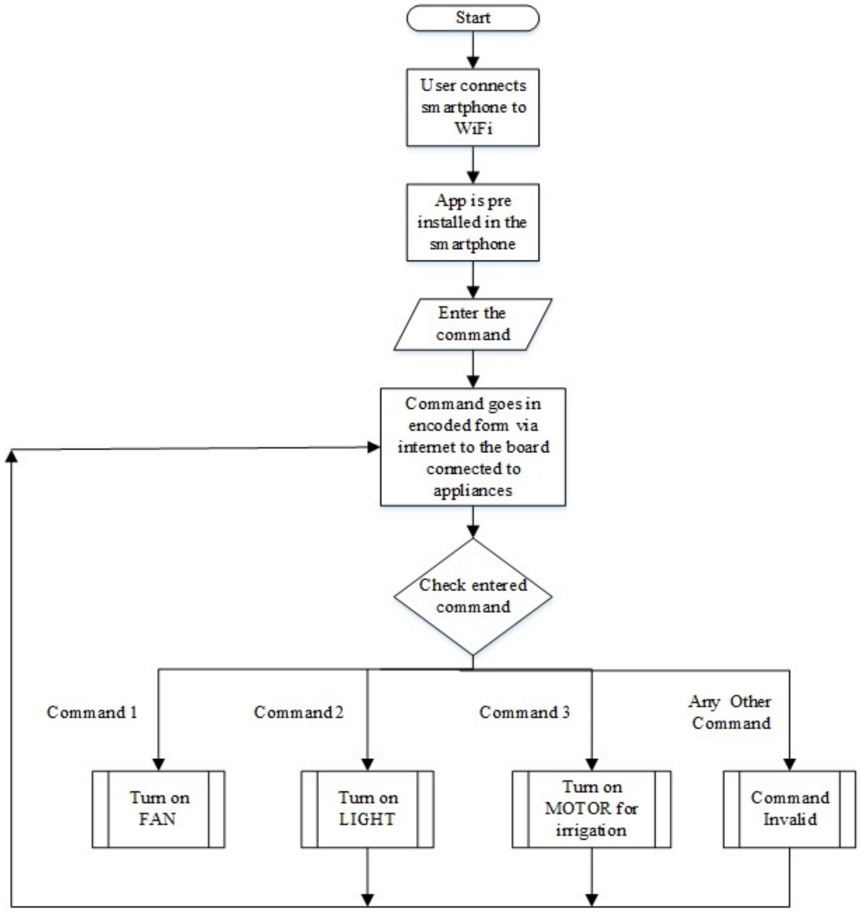
p_{Con}	total power consumption
p_{tx}	transmitted signal power
p_{rx}	received signal power
p_{Sleep}	sleep state power
p_{idl}	idle state power is no packets are transmitted or received
T	battery life time in Hours
i_c	battery Capacity in mAh
I	load current in mA
N	peukert's exponent, it ranges from 1 to 1.3, where 1 is the nominal value
PL	path loss
G_T	transmitter antenna gain
G_R	receiver antenna gain
λ	frequency of wave

d	transmission distance
E	Energy consumption

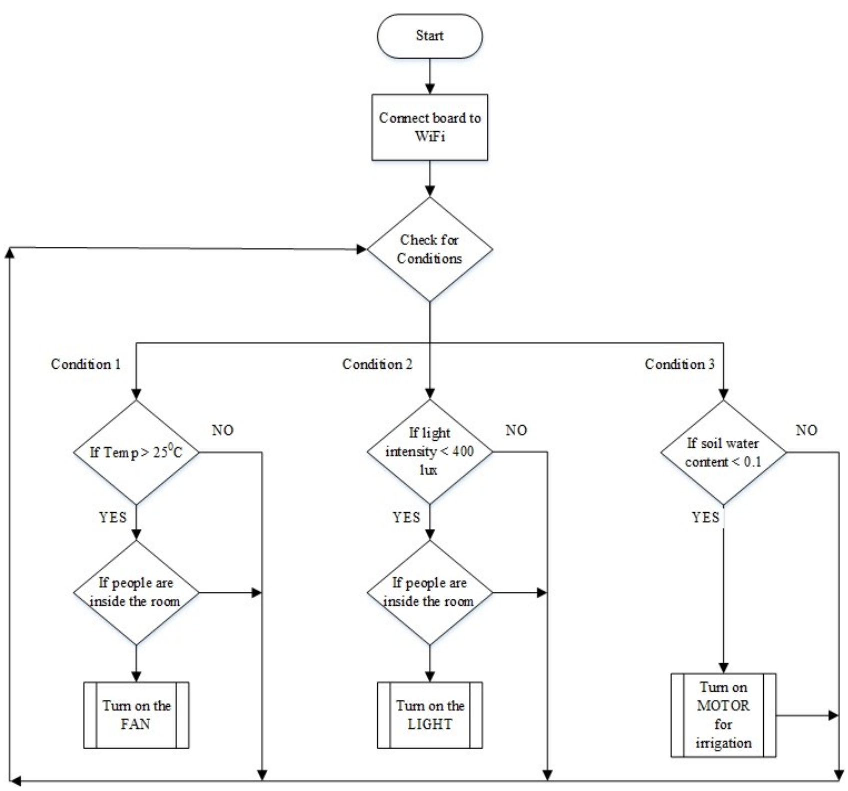
Table no. 2

PSEUDO CODE :

```
if (device state==1)
{
    //code for connection establishment
    // Fan control
    if (room temp > 25⁰C)
    {
        if(no of people in room > 0)
        { turnonfan(); }
    }
    // Light control
    if (light intensity < 400 lux)
    {
        if(no of people in room > 0)
        { turnonlight(); }
    }
    //Irrigation
    if (soil water content < 0.1)
    { turnonmotor(); }
```



Flow chart for manual control



Flow chart for auto control

SIMULATION PARAMETERS

Simulated result is benchmarked upon various parameters which may vary as per respective sensor module type, some of them can be listed for MTS 420CC used are:

1. Humidity / Temperature (C) – It uses SHT 11 single chip sensor module for calibrated digital output values.
2. Barometric Pressure (mba) – Intersema M55ER SMD Hybrid peizoresistive sensor and 3-wire ADC interface.
3. Light intensity (Lux) – TLS2250 digital sensor with dual photodiode provides effective 12 – Bit dynamic range.
4. Acceleration (g) – MEMS micromachined 2 – Axis, +/- 2g capable of tilt detection, movement, vibration and seismic mesaurements.
5. GPS – Leadtek GPS – 9546 or uBlox LEA-4A provides antenna power and serial data at USART1 for positional detection.

Conclusion

Routing in sensor networks is a new area of research, with a limited, but rapidly growing set of research results. In this project, we presented a comprehensive survey of routing techniques in wireless sensor networks which have been presented in the literature. They have the common objective of trying to extend the lifetime of the sensor network, while not compromising data delivery. Overall, the routing techniques are classified based on the network structure into three categories: flat, hierarchical, and location based routing protocols. Furthermore, these protocols are classified into multipath-based, query-based, negotiation-based, or QoS-based routing techniques depending on the protocol operation. We also highlight the design tradeoffs between energy and communication overhead savings in some of the routing paradigm, as well as the advantages and disadvantages of each routing technique. Although many of these routing techniques look promising, there are still many challenges that need to be solved in the sensor networks. We highlighted those challenges and pinpointed future research directions in this regard.

LITERATURE REVIEW

2.1 MOTEWORKS WSN KIT

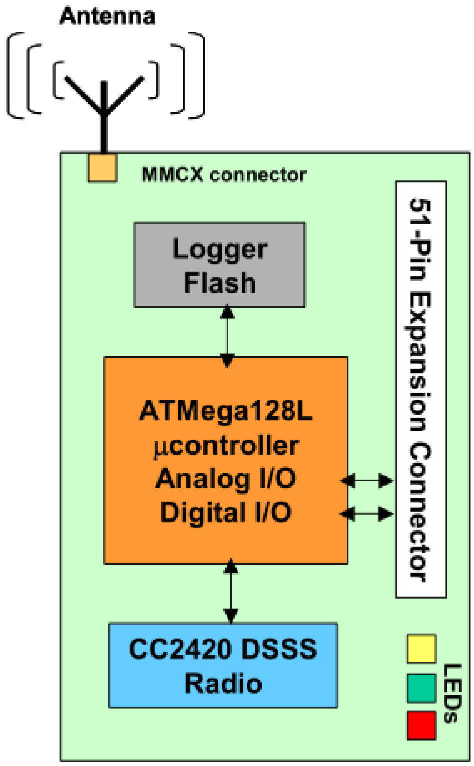
We implemented wireless sensor network using MEMSICS Professional Kit for Wireless Sensor Networks. Wireless sensor hardware and modules, in conjunction with the MoteWorks TinyOS based software platform, enable automation applications.

MEMSIC's IRIS, MICAz and MICA2 are range of wireless sensor network which are used often for setting up networks modules which enable the low-power wireless sensor network measurement system, available in **2.4 GHz**. We have used **MICAz series** boards for WSN deployment in home automation applications.

The **MPR2400** from MicaZ series of WSN boards are equipped with following features :

- IEEE 802.15.4 compliant RF Transceiver
- 2.4 to 2.48 GHz, a globally compatible ISM band
- Direct sequence spread spectrum radio which is resistant to RF interference and provides inherent
- data security
- 250 kbps data rate
- Supported by MoteWorks ™ wireless sensor network platform for reliable, ad-hoc mesh networking
- Plug and play with Crossbow's sensor boards, data acquisition boards, gateways, and software

Block Diagram and Schematics for the MPR2400 / MICAz



Feature	Chapter
Batteries	6
Radio	7
Antenna	8
Data Flash Logger	9
Atmega128	10
Expansion Connector	11

MoteWorks enables the development of custom sensor applications and is specifically optimized for low-power, battery-operated networks. **MoteWorks** is based on the open-source **TinyOS** operating system and provides reliable, ad-hoc mesh networking, over-the-air-programming capabilities, cross development tools, server middleware for enterprise network integration and client user interface for analysis and a configuration.

MoteWorks includes TinyOS, the open-source operating system originally developed by **University of California, Berkeley**. It is also the most widely-deployed wireless sensor network operating system for commercial applications. TinyOS is a **component-based, event-driven** operating system designed from the ground up for low-power devices with small memory footprint requirements. TinyOS supports microprocessors ranging from 8-bit architectures with as little as 2 KB of RAM to 32-bit processors with 32 MB of RAM or more. It provides a well-defined set of APIs for application programming. These APIs provide access to the computing capabilities of the sensor node, allowing for intelligence within the network. Using these capabilities, sensor data can be preprocessed on the node, optimizing both network throughput and battery life by avoiding unnecessary send and receive messages.

A **base station** allows the aggregation of sensor network data onto a PC or other computer platform. Any MICAz Mote can function as a base station when it is connected to a standard PC interface or gateway board. The **MIB510 or MIB520** provides a serial/USB interface for both programming and data communications. Crossbow also offers a stand-alone gateway solution, the MIB600 for TCP/IP-based Ethernet networks.

WORK DONE

3.1 Installation and Configuration

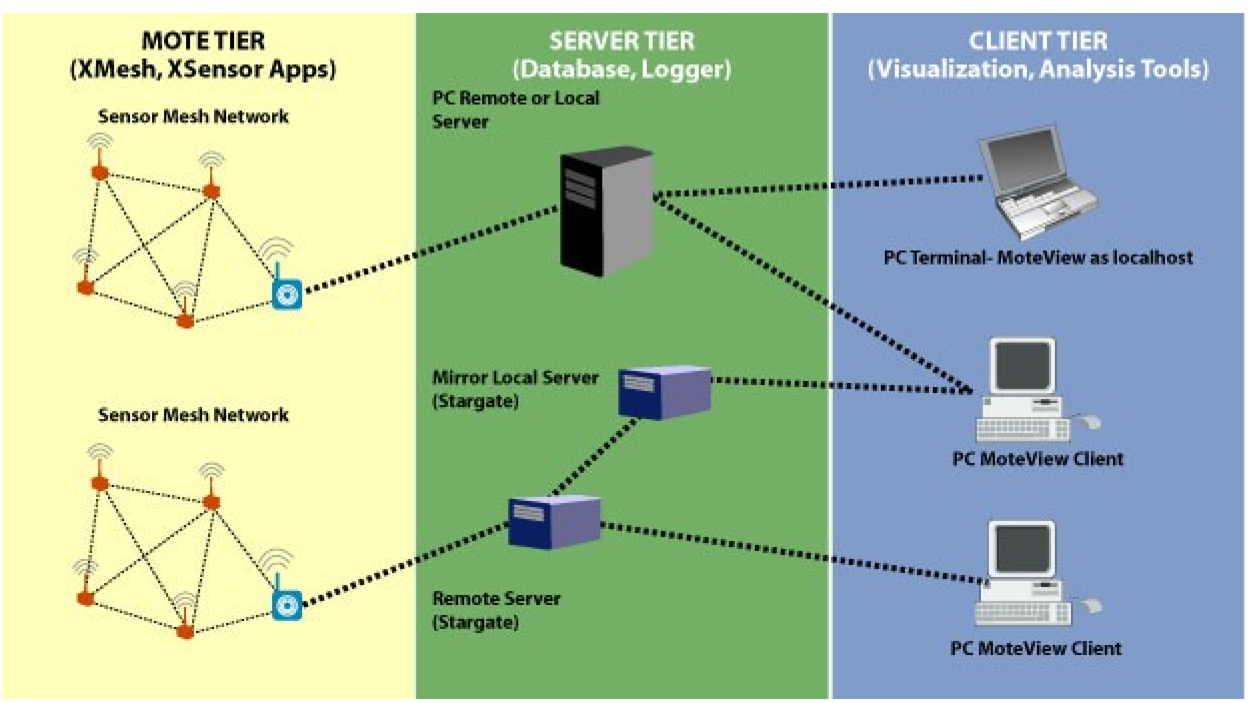
MoteWorks is the end-to-end enabling platform for the creation of wireless sensor networks. The optimized processor/radio hardware, industry-leading mesh networking software, gateway server middleware and client monitoring and management tools support the creation of reliable, easy-to-use wireless OEM solutions. OEMs are freed from the detailed complexities of designing wireless hardware and software enabling them to focus on adding unique differentiation to their applications while bringing innovative solutions to market quickly.

A wireless network deployment is composed of the three distinct software tiers:

- The Mote Tier, where XMesh resides, is the software that runs on the cloud of sensor nodes forming a mesh network. The XMesh software provides the networking algorithms required to form a reliable communication backbone that connects all the nodes within the mesh cloud to the server. (Refer to XMesh User's Manual).

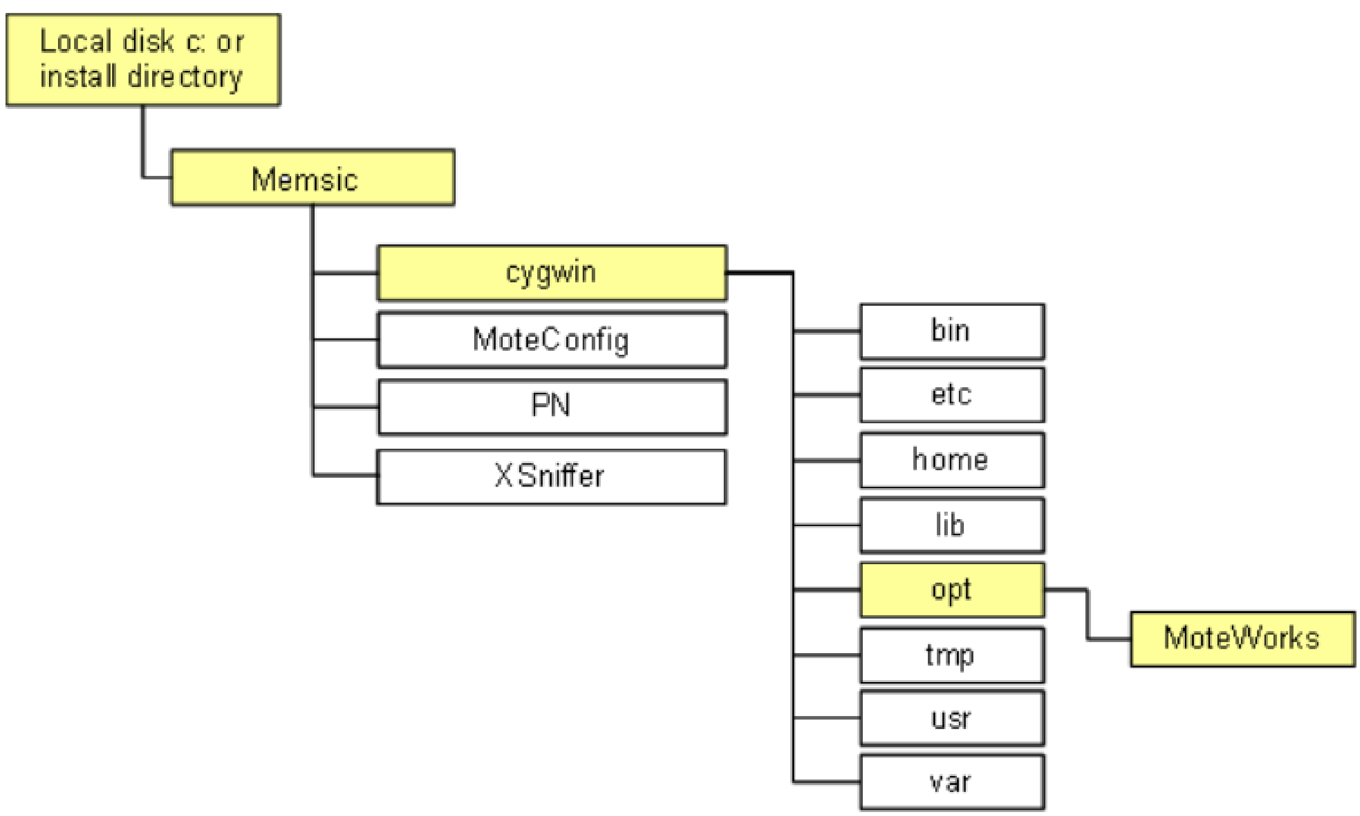
- The Server Tier is an always-on facility that handles translation and buffering of data coming from the wireless network and provides the bridge between the wireless Motes and the internet clients. XServe and XOtap are server tier applications that can run on a PC or Stargate. (Refer to XServe User's Manual).

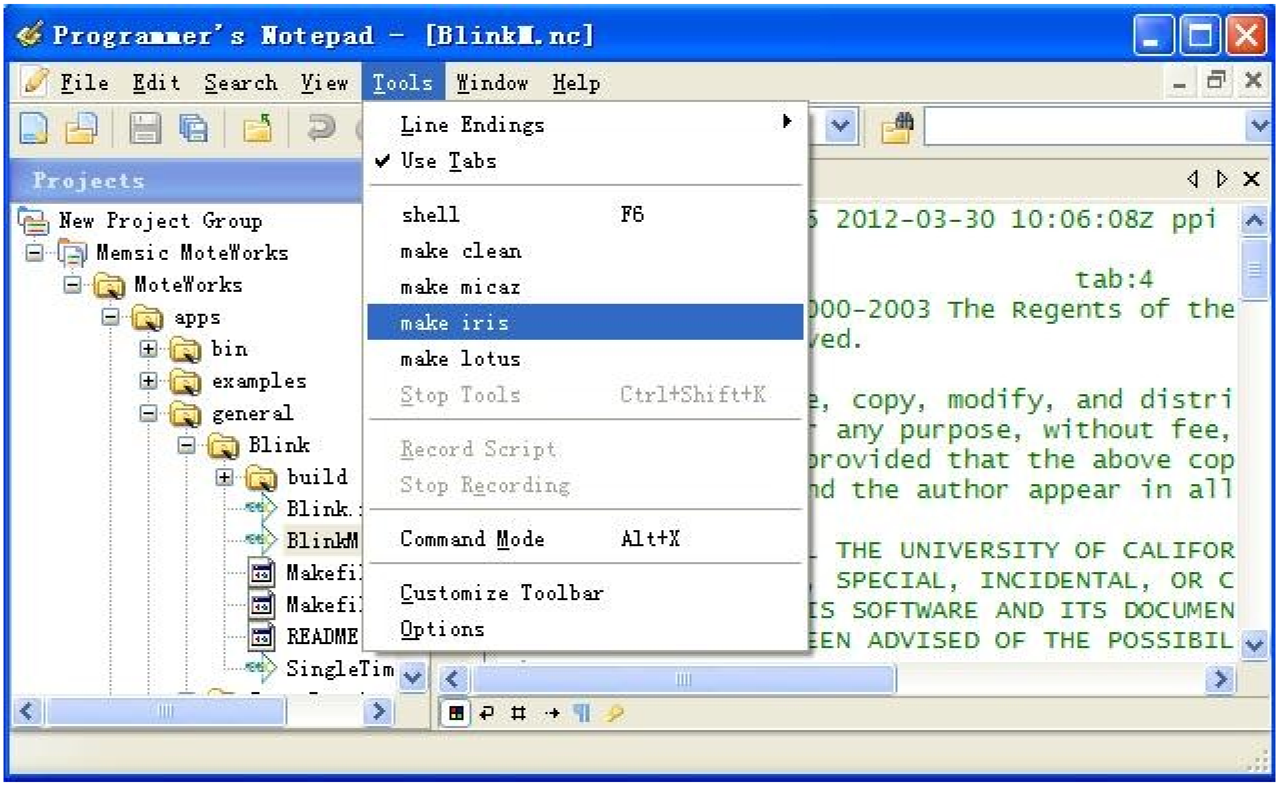
- The Client Tier provides the user visualization software and graphical interface for managing the network. Memsic provides free client software called MoteView, but XMesh can be interfaced to custom client software as well. (Refer to MoteView User's Manual)



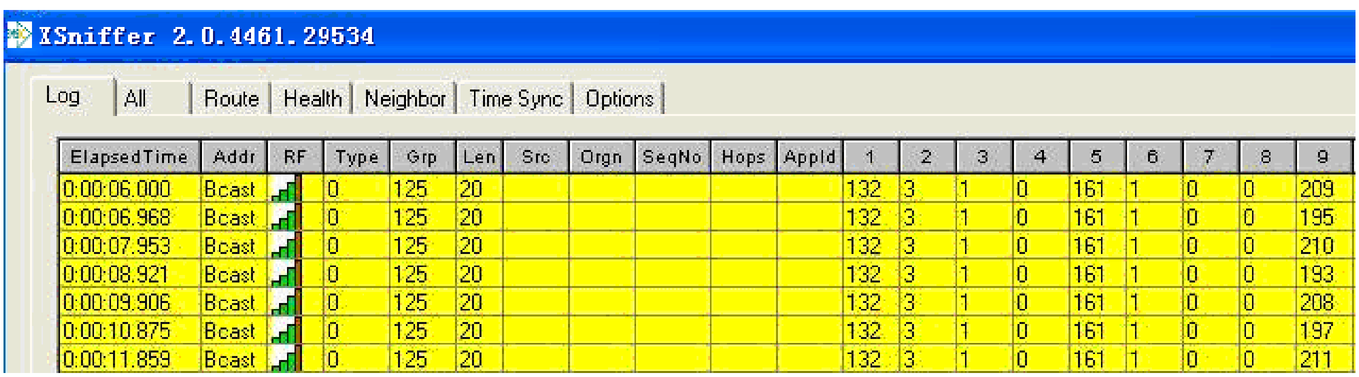
MoteWorks InstallShield Wizard setup offers the following software packages:

- **TinyOS and MoteWorks Tools** - An event-driven OS for wireless sensor networks; tools for debugging
- **nesC compiler** - An extension of C-language designed for TinyOS
- **Cygwin** - A Linux-like environment for Windows
- **AVR Tools** - A suite of software development tools for Atmel's AVR processors
- **Programmer's notepad** - IDE for code compilation and debugging
- **XSniffer** - Network Monitoring Tool for the RF environment
- **MoteConfig** - GUI environment for Mote Programming and OTAP
- **LotusConfig** - GUI environment for Lotus Programming
- **Graphviz** - To view files made from make documents

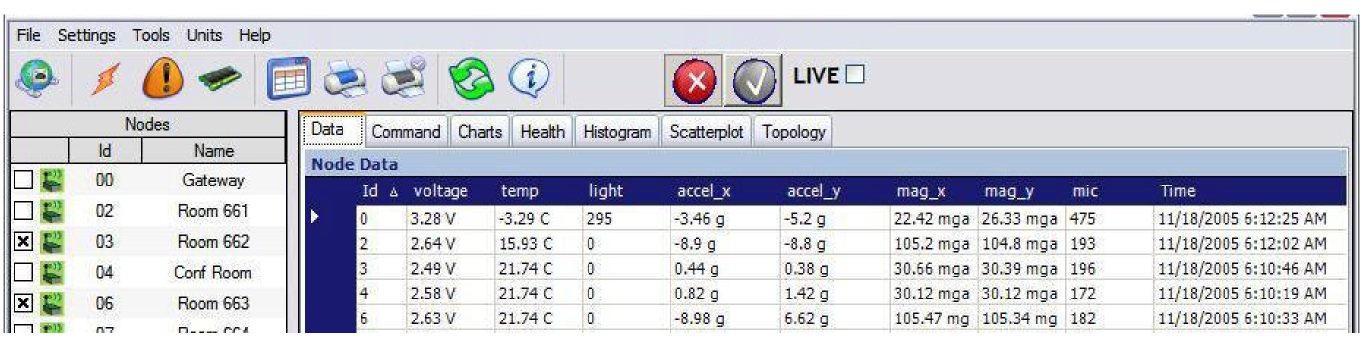




Programmer's Notepad - [BlinkM.nc]
File Edit Search View Tools Window Help
Projects
New Project Group
Memsic MoteWorks
MoteWorks
apps
bin
examples
general
Blink
build
README
Line Endings
Use Tabs
shell F6
make clean
make micaz
make iris
make lotus
Stop Tools Ctrl+Shift+K
Record Script
Stop Recording
Command Mode Alt+X
Customize Toolbar
Options
2012-03-30 10:06:08Z ppi
tab:4
THE UNIVERSITY OF CALIFOR
SPECIAL, INCIDENTAL, OR C



XSniffer 2.0.4461.29534
Log
All
Route
Health
Neighbor
Time Sync
Options
ElapsedTime Addr RF Type Grp Len Src Orgn SeqNo Hops AppId 1 2 3 4 5 6 7 8 9
0:00:06.000 Bcast 0 125 20 132 3 1 0 161 1 0 0 209
0:00:06.968 Bcast 0 125 20 132 3 1 0 161 1 0 0 195
0:00:07.953 Bcast 0 125 20 132 3 1 0 161 1 0 0 210
0:00:08.921 Bcast 0 125 20 132 3 1 0 161 1 0 0 193
0:00:09.906 Bcast 0 125 20 132 3 1 0 161 1 0 0 208
0:00:10.875 Bcast 0 125 20 132 3 1 0 161 1 0 0 197
0:00:11.859 Bcast 0 125 20 132 3 1 0 161 1 0 0 211



File Settings Tools Units Help
LIVE
Nodes
Id Name
00 Gateway
02 Room 661
03 Room 662
04 Conf Room
06 Room 663
Data
Command
Charts
Health
Histogram
Scatterplot
Topology
Node Data
Id voltage temp light accel_x accel_y mag_x mag_y mic Time
0 3.28 V -3.29 C 295 -3.46 g -5.2 g 22.42 mga 26.33 mga 475 11/18/2005 6:12:25 AM
2 2.64 V 15.93 C 0 -8.9 g -8.8 g 105.2 mga 104.8 mga 193 11/18/2005 6:12:02 AM
3 2.49 V 21.74 C 0 0.44 g 0.38 g 30.66 mga 30.39 mga 196 11/18/2005 6:10:46 AM
4 2.58 V 21.74 C 0 0.82 g 1.42 g 30.12 mga 30.12 mga 172 11/18/2005 6:10:19 AM
6 2.63 V 21.74 C 0 -8.98 g 6.62 g 105.47 mg 105.34 mg 182 11/18/2005 6:10:33 AM

4.2 NS3

The ns3 is installed in UBUNTU. We have chosen UBUNTU 14.04 LTS.

For installation we have used the command line (terminal of UBUNTU), and used 20 commands for the same. The commands are

- sudo apt-get install gcc g++ python
- sudo apt-get install gcc g++ python-dev
- sudo apt-get install qt4-dev-tools
- sudo apt-get install mercurial
- sudo apt-get install bzr
- sudo apt-get install cmake libc6-dev libc6-dev-i386 g++-multilib
- sudo apt-get install gdb valgrind
- sudo apt-get install gsl-bin lingsl0-dev libsl0l-dev
- sudo apt-get install tcpdump
- sudo apt-get install sqlite sqlite3 libsqlite3-dev
- sudo apt-get install libxml2 libxml2-dev
- sudo apt-get install libgtk2.0-dev
- sudo apt-get install vtun lxc
- sudo apt-get install uncrustify
- sudo apt-get install doxygen graphviz imagemagick
- sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font utils dvipng
- sudo apt-get install python-sphinx dia
- sudo apt-get install python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev
- sudo apt-get install libboost-signals-dev libboost-filesystem-dev
- sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev

```
Waf: Leaving directory `/home/user/ns-allinone-3.22/ns-3.22/build'
'build' finished successfully (20.334s)

Modules built:
antenna                   aodv                      applications
bridge                    buildings                 config-store
core                      csma                      csma-layout
dsdv                      dsr                       energy
fd-net-device             flow-monitor              internet
lr-wpan                   lte                       mesh
mobility                  mpi                       netanim (no Python)
network                   nix-vector-routing        olsr
pftr (no Python)          point-to-point            point-to-point-layout
propagation               sixlowpan                 spectrum
stats                     tap-bridge                test (no Python)
topology-read             uan                       virtual-net-device
visualizer                wave                      wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite                     click                     openflow
```

REFERENCES