

IoT ARCHITECTURE FOR SMART CITIES

A major project report submitted for partial fulfillment of the requirements for

the award of the degree of

MASTER OF TECHNOLOGY

in

SOFTWARE ENGINEERING



Submitted by:

Kshitij Saini - 14/ICS/024

Shikha Dwivedi - 14/ICS/048

Submitted to:

Dr. Vidushi Sharma

HOD (School of ICT, Computer Science)

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

GAUTAM BUDDHA UNIVERSITY

GREATER NOIDA-201312, UTTAR PRADESH, INDIA

MAY, 2018

ACKNOWLEDGEMENT

We would like to take this opportunity to express our humble gratitude to **Dr. Vidushi Sharma HOD (SOICT, Computer Science Department)** under whom we are explaining this topic. Her constant guidance and willingness to share her vast knowledge made us understand this topic and its manifestations in great depths and helped us to complete the assigned tasks timely.

We would also like to thank all the faculty members and staff of the department of Computer Science and Engineering (SoICT), Gautam Buddha University for their contribution in the development of this topic of major project.

CERTIFICATE

This is to certify that this major project report entitled “**IoT Architecture For Smart Cities**” by **SHIKHA DWIVEDI (14/ICS/048), KSHITIJ SAINI (14/ICS/024)**, is an authentic work carried out under my supervision and guidance in fulfillment of the requirements for the purpose of the Major Project in **Master of Technology in Computer Science and Engineering**

SUPERVISED BY:

DR.VIDUSHI SHARMA

Head of Department, Computer Science Department(SOICT)

GAUTAM BUDDHA UNIVERSITY



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

GAUTAM BUDDHA UNIVERSITY, GREATER NOIDA, 201 312, U. P., (INDIA)

Candidate's Declaration

We, hereby, certify that the work embodied in this project report entitled “IOT Architecture For Smart Cities” by us in partial fulfillment of the requirements for the award of the degree of M.Tech. in Information and Communication Technology (ICT) with Specialization in Software Engineering submitted to the School of Information and Communication Technology, Gautam Buddha University, Greater Noida is an authentic record of our own work carried out under the supervision of Dr. Vidushi Sharma School of ICT. The matter presented in this report has not been submitted by me in any other University / Institute for the award of any other degree or diploma. Responsibility for any plagiarism related issue stands solely with us.

Names:

Signatures:

Shikha Dwivedi (14/ICS/048)

Kshitij Saini (14/ICS/024)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief. However, responsibility for any plagiarism related issue solely stands with the students.

Signature of the Supervisor: Dr. Vidushi Sharma

Name with Designation: HOD CSE

Date:

Place: Greater Noida

ABSTRACT

The Internet of Things (IoT) might have the capacity to combine straightforwardly and flawlessly a substantial number of various and heterogeneous end frameworks, while giving open access to chosen subsets of information for the improvement of a plenty of computerized benefits in the coming years. Making and developing a general design for the IoT is henceforth an exceptionally complex undertaking, for the most part in view of the to a great degree substantial assortment of gadgets, diverse conventions, connect layer advancements, and administrations that might be engaged with such a framework. In this report, we are concentrating on the development and executing the IoT design and will attempt to demonstrate the outcomes on nearby and in addition remote areas. Utilizing the devices and manuals given by MEMSIC Inc. we should have the capacity to achieve the execution of the center structure of Urban IoT. Urban IoTs, truth be told, are intended to help the Smart City vision, which goes for misusing the most exceptional correspondence advances to help included esteem administrations for the organization of the city and for the subjects. Moreover, this report speaks to and indicates execution of Urban IoT Smart city structure by talking about the devices, conventions, innovations utilized as a part of actualizing the case of Smart Environment Temperature checking framework and Smart Theft observing framework.

List Of Abbreviations

RISC	Reduced Instruction Set Computer
CMOS	Complementary Metal Oxide Semiconductor
RAM	Random Access Memory
MIPS	Million Instructions Per Second
OQPSK	Offset Quadrature Phase Shift Keying
IEEE	Institute of Electrical and Electronics Engineers
ELP	Extended Low Power
HP	High Power
LP	Low Power
ISP	Internet Service Provider
USB	Universal Serial Bus
JTAG	Joint Test Action Group
6LOWPAN	Internet Protocol (IPv6) and Low-power Wireless Personal Area Networks
WSN	Wireless Sensor Networks
SMTP	Simple Mail Transfer Protocol
MTA	Mail Transfer Agent
POP	Post Office Protocol
LDR	Light Dependent Resistor
CdSe	Cadmium-Selenide

TABLE OF CONTENT

I.	Acknowledgement.....	ii
II.	Certificate.....	iii
III.	Candidate Declaration.....	iv
IV.	Abstract.....	v
V.	List of Abbreviations	vi
1.	Chapter 1: Introduction.....	2-9
	1.1 Features.....	2-4
	1.2 Challenges.....	4-6
	1.3 Application.....	6-8
	1.4 Limitation	8-9
2.	Chapter 2: Framework of IoT.....	10-27
	2.1 Introduction.....	10
	2.2 Topologies.....	10-13
	2.3 XMesh.....	13-19
	2.4 MoteConfig.....	19-20
	2.5 MoteView.....	20-21
	2.6 MIB520 USB Interface Board.....	22-24
	2.7 MDA 100.....	25
	2.8 XM2110(IRIS).....	25-26
	2.9 6Lowpan.....	27
3.	Chapter 3: Environment Temperature Monitoring System.....	28-39
	3.1 Introduction.....	28
	3.2 Tools.....	28
	3.3 Circuit Diagram.....	29
	3.4 Conversion Unit.....	30
	3.5 Implementation.....	30-32
	3.6 Visualization.....	32-35
	3.7 Sending Alert Message.....	36-39
	3.8 Conclusion.....	39
4.	Chapter 4: Anti-Theft Monitoring System.....	40-46

4.1 Introduction.....	40
4.2 Tools.....	40-41
4.3 Visualization.....	41-43
4.4 Sending Alert Message.....	44-45
4.5 Conclusion.....	46
5. Chapter 5: Conclusion.....	47
References.....	48

CHAPTER 1: INTRODUCTION

IoT implies interfacing regular things installed with gadgets, programming and sensors to the web empowering them to gather and trade information. IoT is valuable since it gives productive asset use, limiting human endeavors, spares time, advancement of Artificial Intelligence through IOT, enhanced security, and many more.

1.1 Features

- IoT connects various objects to the IoT platform.
- Analyze the data collected and use it to build business intelligence.
- Integrate various models to improve user experience.

IoT idea, thus, goes for making the Internet considerably more immersive and unavoidable. Besides, by empowering simple access and communication with a wide assortment of gadgets, for example, for example, home machines, reconnaissance cameras, observing sensors, actuators, presentations, vehicles, etc. The IoT will cultivate the improvement of various applications that make utilization of the conceivably tremendous sum and assortment of information produced by such protests give new administrations to residents, organizations, and open organizations.

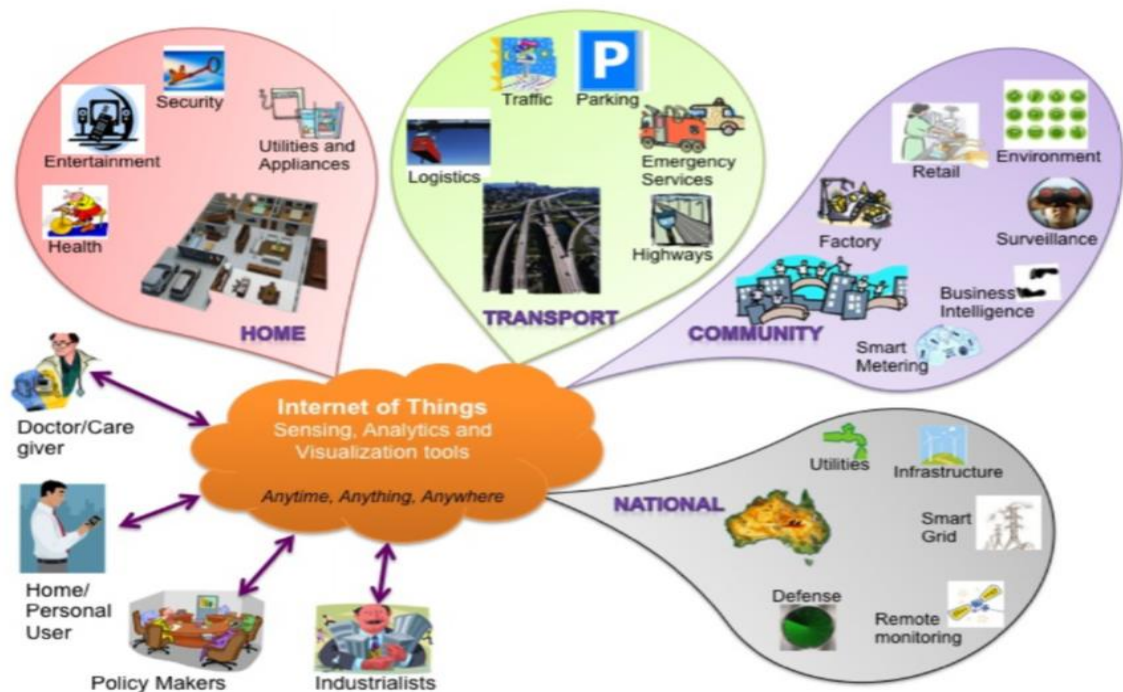


Fig1.1 Internet of Things in Daily Life

A shrewd city is a group that is effective, practical and liveable. The term keen city has turned out to be increasingly famous in the field of urban arranging. The six measurements of a savvy city are Smart Economy, Smart Mobility, Smart Environment, Smart People, Smart Living and Smart Governance. Each city can end up more astute by concentrating on any of the above measurements. Keen urban areas can fill in as an apparatus for controlling the quick urbanization and different issues caused by the consistently expanding urban populace. The usage of the brilliant advances can build the estimation of the city. Savvy city idea presents new practices and administrations that profoundly impacts approach making and arranging.

Keen urban communities are relied upon to enhance the personal satisfaction of nationals by depending on new ideal models, for example, the IoT and its ability to oversee and interconnect a huge number of sensors and actuators scattered over the city. Shrewd urban communities are mind boggling situations where a few regions of advancement meet keeping in mind the end goal to generously enhance financial improvement and personal satisfaction. Monetary developments, innovative instruments that urge individuals to take an interest in administration procedures and Internet-empowered city foundation administrations and utilities shape a flourishing examination field. The expanding requirement for adaptability and conveyed control has prompted the idea of the Internet of Things. The IoT worldview, which changes any question or machine into part of an associated organize, in this manner dissolving the boundaries between the physical and virtual universes, has turned into the foundation of numerous administrations in a keen city, including more efficient upkeep undertakings.

To make a city brilliant the primary point is to favorable circumstances of present day data and correspondence advances to improve utilization of open assets increment the nature of offered administrations decrease operational expenses of government.

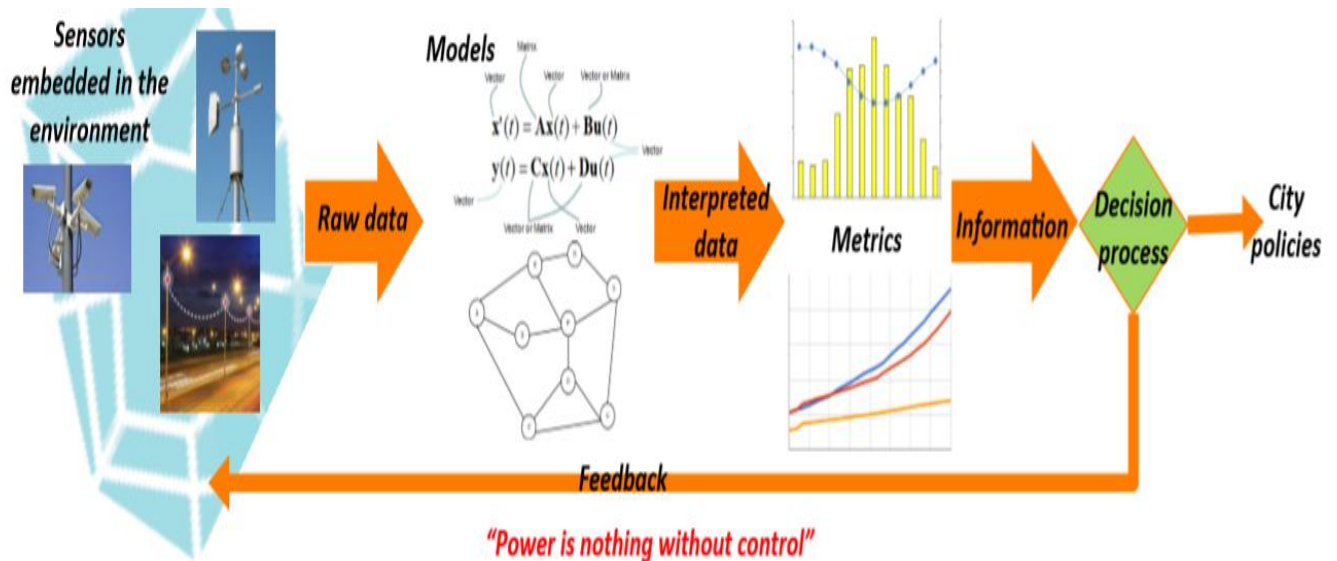


Fig 1.2. How to turn dumb data into smart services

1.2 Challenges of Iot

Analysts estimate that 20+ Billion devices will get connected to the Internet by 2020*. In this exploding Internet of Things (IoT), users, things and cloud services connect using the Internet to enable new use cases and new business models across multiple markets and applications. Texas Instruments is the only semiconductor company with all of the building blocks to enable the IoT. There are many challenges and few of them are listed below.

1. Low power is paramount

For the IoT to evolve from a niche market to a pervasive network that connects virtually every aspect of our lives, power consumption is vital. Many of the connected devices within the IoT are nodes containing microcontrollers (MCUs), sensors, wireless devices and actuators that collect data. In many cases, these nodes will be battery operated or have no batteries at all, gathering power through energy harvesting. Particularly in industrial settings, these nodes will be placed in hard-to-access or no-access areas. This means they must be able to operate and transmit data for years at a time on a single, coin-cell battery.

2. Sensing is essential

Without sensing, there would be no IoT. The entire IoT system starts with sensors, the tiny devices or nodes measuring anything and everything to create data that is sent to

other nodes or to the cloud. Whether sensing that a door is open at your house, that your car's oil needs to be changed or that a piece of equipment is about to fail on an assembly line, sensors gather crucial information.

3. Connectivity options: Simplifying the complex is critical

Once the sensor data is collected by low-powered nodes, it must be sent somewhere. In most cases, it goes to a gateway, which is a midpoint between the Internet/cloud or other nodes in an IoT system. Today, there are multiple wired and wireless options to connect devices with unique use cases and different needs. Each of the 14 different connectivity standards and technologies serve valuable purposes, but being able to take on all of those standards from Wi-Fi to Bluetooth® to Sub-1 GHz to Ethernet is a huge undertaking

4. Security is crucial for widespread adoption

Security of the entire system as the biggest barrier to widespread adoption of the IoT. With more devices becoming 'smart,' it will enable more potential security breach entry points. teams are looking at ways to build the most advanced hardware security mechanisms while keeping them small, low cost and low power. On top of that, we are investing heavily in integrated security protocols and security software to make security implementation as simple as possible for customers.

5. IoT needs to be made easy for inexperienced developers

At first, IoT technology was predominantly used by technology companies. But today – and even more so in the future – the IoT will be included in industries with limited technological background. For example, take a faucet manufacturing company. Until now, electrical engineers may never have worked at a faucet manufacturing company because there was no need. But if the company wanted to make Internet-connected shower heads, the investment in manpower and time would be significant. Thus, IoT technologies must be easy to add to existing and future customer products without the need to have network and security engineers on staff. While more and more aspects of our lives are being connected, and as the IoT continues to proliferate, a lot more work must be done.

6. Managing cloud connectivity is key

Once the data passes through a gateway, in most cases it heads straight to the cloud where that data can be analyzed, reviewed and put into action. The value of IoT comes

from data running on cloud services. Just like with connectivity, there are lots of cloud service options – yet another point of complexity in the IoT world.

1.3 Application of iot

Some applications of Iot are:

Structural Health of Buildings: Legitimate upkeep of the authentic structures of a city requires the ceaseless observing of the real states of each building and recognizable proof of the zones that are most subject to the effect of outside specialists. The urban IoT may give a disseminated database of building auxiliary respectability estimations, gathered by appropriate sensors situated in the structures, for example, vibration and distortion sensors to screen the building pressure, air operator sensors in the encompassing territories to screen contamination levels, and temperature and moistness sensors to have an entire portrayal of the ecological conditions. This database ought to diminish the requirement for costly intermittent auxiliary testing by human administrators and will permit focused on and proactive upkeep and reclamation activities. At long last, it will be conceivable to consolidate vibration and seismic readings keeping in mind the end goal to better investigation and comprehend the effect of light quakes on city structures. This database can be made openly available keeping in mind the end goal to make the residents mindful of the care taken in protecting the city chronicled legacy.

Waste Management: Waste management is a primary issue in many modern cities, due to both the cost of the service and the problem of the storage of garbage in landfills. A deeper penetration of ICT solutions in this domain, however, may result in significant savings and economical and ecological advantages. For instance, the use of intelligent waste containers, which detect the level of load and allow for an optimization of the collector trucks route, can reduce the cost of waste collection and improve the quality of recycling. To realize such a smart waste management service, the IoT shall connect the end devices, i.e., intelligent waste containers, to a control center where an optimization software processes the data and determines the optimal management of the collector truck fleet.

Air Quality: The European Union officially adopted a 20-20-20 Renewable Energy Directive setting climate change reduction goals for the next decade.⁴ The targets call for a 20% reduction in greenhouse gas emissions by 2020 compared with 1990 levels, a 20% cut in energy

consumption through improved energy efficiency by 2020, and a 20% increase in the use of renewable energy by 2020. To such an extent, an urban IoT can provide means to monitor the quality of the air in crowded areas, parks, or fitness trails. In addition, communication facilities can be provided to let health applications running on joggers' devices be connected to the infrastructure. In such a way, people can always find the healthiest path for outdoor activities and can be continuously connected to their preferred personal training application. The realization of such a service requires that air quality and pollution sensors be deployed across the city and that the sensor data be made publicly available to citizens.

Smart Lighting: In order to support the 20-20-20 directive, the optimization of the street lighting efficiency is an important feature. In particular, this service can optimize the street lamp intensity according to the time of the day, the weather condition, and the presence of people. In order to properly work, such a service needs to include the street lights into the Smart City infrastructure. It is also possible to exploit the increased number of connected spots to provide WiFi connection to citizens. In addition, a fault detection system will be easily realized on top of the street light controllers.

Noise Monitoring: Noise can be seen as a form of acoustic pollution as much as carbon oxide (CO) is for air. In that sense, the city authorities have already issued specific laws to reduce the amount of noise in the city centre at specific hours. An urban IoT can offer a noise monitoring service to measure the amount of noise produced at any given hour in the places that adopt the service. Besides building a space-time map of the noise pollution in the area, such a service can also be used to enforce public security, by means of sound detection algorithms that can recognize, for instance, the noise of glass crashes or brawls. This service can hence improve both the quiet of the nights in the city and the confidence of public establishment owners, although the installation of sound detectors or environmental microphones is quite controversial, because of the obvious privacy concerns for this type of monitoring.

Traffic Congestion: On the same line of air quality and noise monitoring, a possible Smart City service that can be enabled by urban IoT consists in monitoring the traffic congestion in the city. Even though camera-based traffic monitoring systems are already available and deployed in many cities, low-power widespread communication can provide a denser source of information.

Traffic monitoring may be realized by using the sensing capabilities and GPS installed on modern vehicles , and also adopting a combination of air quality and acoustic sensors along a given road. This information is of great importance for city authorities and citizens: for the former to discipline traffic and to send officers where needed and for the latter to plan in advance the route to reach the office or to better schedule a shopping trip to the city Centre.

City Energy Consumption: Together with the air quality monitoring service, an urban IoT may provide a service to monitor the energy consumption of the whole city, thus enabling authorities and citizens to get a clear and detailed view of the amount of energy required by the different services (public lighting, transportation, traffic lights, control cameras, heating/ cooling of public buildings, and so on). In turn, this will make it possible to identify the main energy consumption sources and to set priorities in order to optimize their behavior. This goes in the direction indicated by the European directive for energy efficiency improvement in the next years. In order to obtain such a service, power draw monitoring devices must be integrated with the power grid in the city. In addition, it will also be possible to enhance these services with active functionalities to control local power production structures (e.g., photovoltaic panels).

Smart Parking: The smart parking service is based on road sensors and intelligent displays that direct motorists along the best path for parking in the city. The benefits deriving from this service are manifold: faster time to locate a parking slot means fewer CO emission from the car, lesser traffic congestion, and happier citizens. The smart parking service can be directly integrated in the urban IoT infrastructure, because many companies in Europe are providing market products for this application. Furthermore, by using short-range communication technologies, such as Radio Frequency Identifiers (RFID) or Near Field Communication (NFC), it is possible to realize an electronic verification system of parking permits in slots reserved for residents or disabled, thus offering a better service to citizens that can legitimately use those slots and an efficient tool to quickly spot violations.

1.4 Limitations

Internet of Things or IoT is the interconnection of several physical objects including everything such as home appliances (washing machine, toaster, refrigerator, microwave oven, coffee maker, etc.), mobile phones, laptops, television, etc. All the connected “things” are embedded with sensors, softwares, electronics and Internet connectivity in order to exchange information with

each other. It has so many advantages since it prevents extra time and energy consumption. But it too has flaws. Some of them are listed below that are undoubtedly thoughtful.

1. Privacy – This is a great concern when it comes to exchanging valuable information regarding anything. Since everything will be connected breaching inside the network would be easy by the hackers. By entering into just a part of network would reveal everything regarding an individual or organization or both (maybe). What if your office colleagues know what medicines you take or where did you go last night?

2. Safety – If a situation comes like a notorious hacker changes your medical prescription and you are supplied expired medicines or those medicinal drugs to which you are allergic to, then there would be a health disaster. Since the consumer that time would be dependent entirely on the technology there would be least probability that he would bother checking anything. The verification today is done manually by the consumer himself but no one knows what will happen later.

3. Compatibility – At present there is no international standard for device compatibility. For example, home based appliances and equipments may be getting problems in connecting with laptops or mobile phones. Also, Apple devices don't accept the connectivity with any other device. Likewise, different manufacturers need to agree upon this else people will prefer buying only one brand and there would be monopoly.

4. Complexity – If there comes a bug all of a sudden or electricity goes off there would be so much of problem since everyone will be dependent on the IoT technology only. The bugs will lead to certain tasks wrongly accomplished or not done at all.

5. Unemployment – Since the technology would do its task all by itself there will be no need of any manpower, ultimately leading to unemployment. Today's computers need to be operated by someone but later the lists of tasks would be done by the machine itself.

6. Being dependent entirely – Today's generation kids are so used to having mobile phones, tablets, Internet, computers, elevators, air conditioners, etc. that they get bored without these things. The older generation cannot imagine the term "getting bored" actually exists since they somehow managed to get busy by doing something productive that actually needed manual work.

CHAPTER 2: FRAMEWORK OF IOT

2.1 Introduction

In this chapter, we introduce you with the topologies, protocols, xmesh, xmesh network landscape, MoteConfig, MoteView, MIB520 USB interface board, MDA100 and XM2100(iris) Wi-Fi module.

Topology is the arrangement of a network including its nodes and connecting lines. There are two ways of defining a network geometry: the physical topology and the logical topology. The physical topology of a network is the actual geometry of workstations.

Protocols are a set of rules in which computers communicate with each other. The protocol says what part of the conversation comes at which time. It also says how to end the communication.

XMesh is a full featured multi-hop, ad-hoc, mesh networking protocol developed by MEMSIC for wireless networks. More details about Xmesh are mentioned in this chapter.

Xmesh network Landscape is a wireless network deployment is composed of the three distinct software tiers:

- 1.Mote Tier,
- 2.Server Tier,
- 3.Client Tier.

MoteView and MoteConfig are Software interfaces that allow us to program and view the data collected by our sensor nodes. More details about other components are described in this chapter.

2.2 Topologies

There are several architectures that can be used to implement wireless sensor network applications, including star, mesh, and star-mesh hybrid.

- Endpoints — Incorporate with sensors and actuators to catch the sensor information. For ZigBee systems these are generally alluded to RFDs (Reduced Functional Devices). RFDs can't forward system messages upstream or downstream.

- Routers — Extend network area coverage, route around obstacles, and provide backup routes in case of network congestion or device failure. In some cases, routers can also act as endpoints. Routers are also referred to as full-function devices (FFD) in a ZigBee network.
- Gateways — Aggregate the data from the network, interface to the host, LAN, or the Internet, and act as a portal to monitor performance and configure network parameters.
- System Software — provides the networking protocol to enable the self-configuring, self-healing ad hoc network

2.2.1 Star Topology -A star topology is a solitary jump framework in which all remote sensor hubs are inside direct correspondence extend — as a rule 30 to 100 meters — to the gateway. All sensor hubs are indistinguishable — they are generally endpoints — and the passage serves to impart information and summons to the sensor endpoints. The gateway is additionally used to transmit information to a more elevated amount control or observing framework. The endpoints don't pass information or summons to each other's; they utilize the gateway as a coordination point

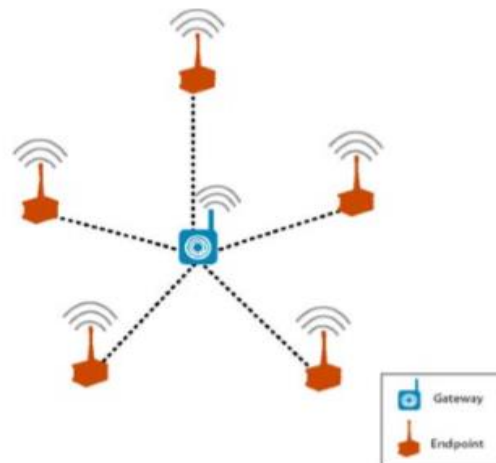


Fig 2.1. Star Topology

The star topology delivers the lowest overall power consumptions but is limited by the transmission distance of the radio in each endpoint back to the gateway. There are also no alternate communication paths to the endpoints. If path becomes obstructed, communication with the associated endpoint will be lost

2.2.2 Mesh Topology

Mesh topologies are multi-hopping frameworks in which all remote sensor nodes are indistinguishable — they are for the most part routers — and speak with each other to hop information to and from the sensor nodes and the gateway. This is the standard XMesh arrangement. Dissimilar to the star design, where the nodes can just converse with the gateway, the nodes in a mesh topology can likewise hop messages among other router nodes. A mesh arrange is additionally exceptionally blame tolerant since every sensor node has different ways back to the gateway and to different nodes. In the event that a sensor node falls flat, the system will reconfigure itself around the fizzled node naturally. Contingent upon the quantity of nodes and the separations between them, the system may likewise encounter expanded inertness as sensor information is hopped from node to node on its way to the gateway.

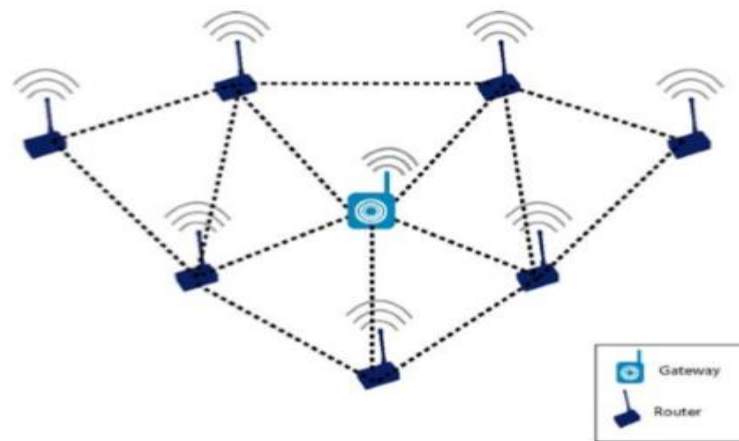


Fig 2.2. Mesh Topology

2.2.3 Star-Mesh Hybrid

A star-mesh hybrid seeks to take advantage of the low power and simplicity of the star topology, as well as the extended range and self-healing nature of a mesh topology. A star-mesh hybrid organizes sensor nodes in a star topology around routers which, in turn, organize themselves in a mesh network. The routers serve both to extend the range of the network and to provide fault tolerance. Since wireless sensor nodes can communicate with multiple routers, the network reconfigures itself around the remaining routers if one fails or if a radio link experiences interference.

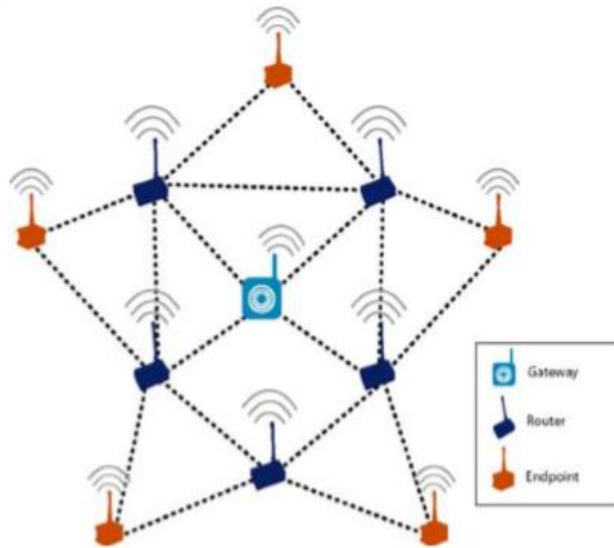


Fig 2.3. Star-Mesh Hybrid

2.3 XMesh

2.3.1 XMesh Overview

XMesh is a full featured multi-hop, ad-hoc, mesh networking protocol developed by MEMSIC for wireless networks. An XMesh network consists of nodes (Motes) that wirelessly communicate to each other and are capable of hopping radio messages to a base station where they are passed to a PC or other client. The hopping effectively extends radio communication range and reduces the power required to transmit messages. By hopping data in this way, XMesh can provide two critical benefits: improved radio coverage and improved reliability.

XMesh is a software library, using the TinyOS operating system that runs on embedded devices called Motes. Motes consist of a:

- a. Microprocessor:** Atmel ATmega1281 for IRIS - ATmega1281 has 128K of flash memory, 8K of RAM, and 4K of EEPROM. The ATmega 1281 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega 1281 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

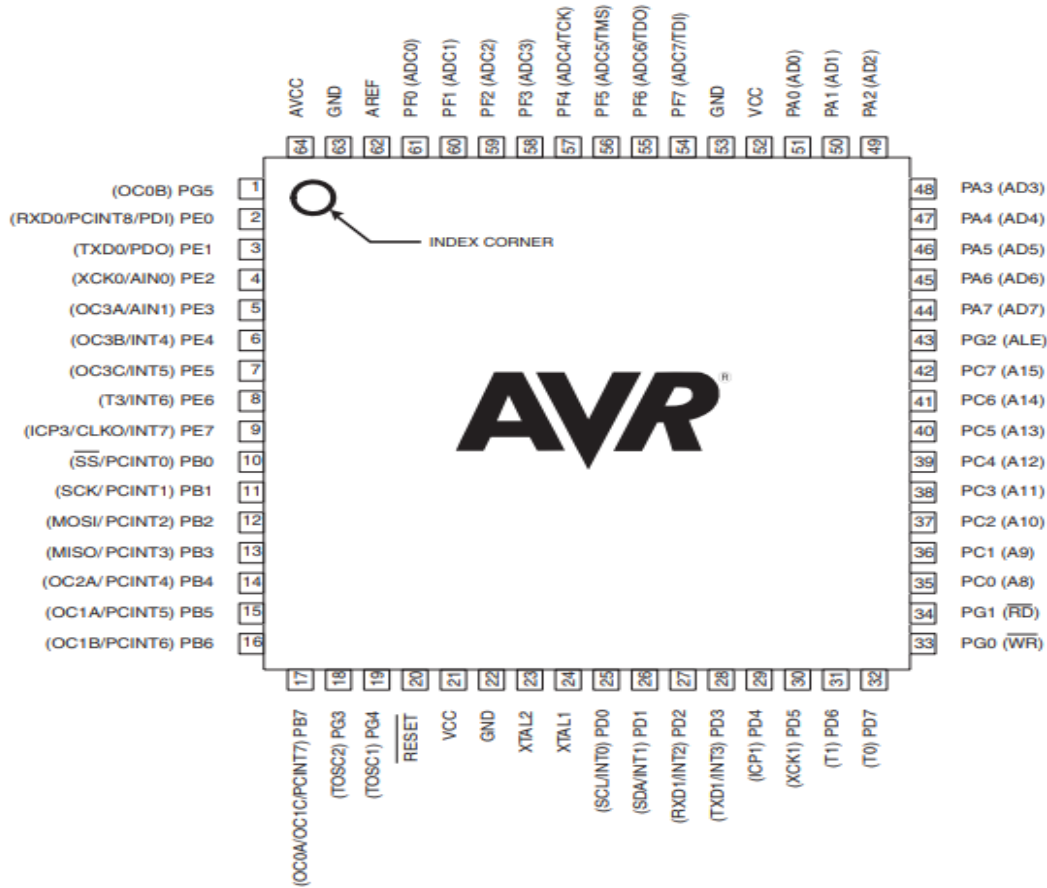


Fig 2.4. ATmega1281 Microprocessor Pinout

b. Radio: The radio used by the IRIS is an IEEE 802.15.4 compliant RF transceiver designed for low power and low-voltage wireless applications. It uses Atmel's AT86RF230 radio that employs OQPSK ("offset quadrature phase shift keying") with half sine pulse shaping. The 802.15.4 radio includes a DSSS (digital direct sequence spread spectrum) baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbps. The radio is a highly integrated solution for wireless communication in the 2.4 GHz unlicensed ISM band. It complies with worldwide regulations covered by ETSI EN 300 328 and EN 300 440 class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan).

c. Serial Flash: Serial flash is a small, low-power flash memory that provides only serial access to the data - rather than addressing individual bytes, the user reads or writes large contiguous groups of bytes in the address space serially. Serial Peripheral Interface Bus (SPI) is a typical protocol for accessing the device. When incorporated into an embedded system, serial flash requires fewer wires on the PCB than parallel flash memories, since it transmits

and receives data one bit at a time. This may permit a reduction in board space, power consumption, and total system cost. It is external flash storage memory to support OTAP (over-the-air programming) and data logging.

- d. UID:** An integrated circuit that is programmed with a unique 64-bit identifier (for MICA2 and MICA2DOT).

The entire XMesh network consists of (see Fig 2.5):

1. One or more Motes which participate in the network
2. A base station node. This is another MICA2 or MICAz mounted on a MEMSIC MIB510/520/600 interface board and programmed with XMeshBase application. It manages the network and forwards data messages into and out of the mesh.
3. A PC, Stargate or other client which receives data and sends commands into the network.

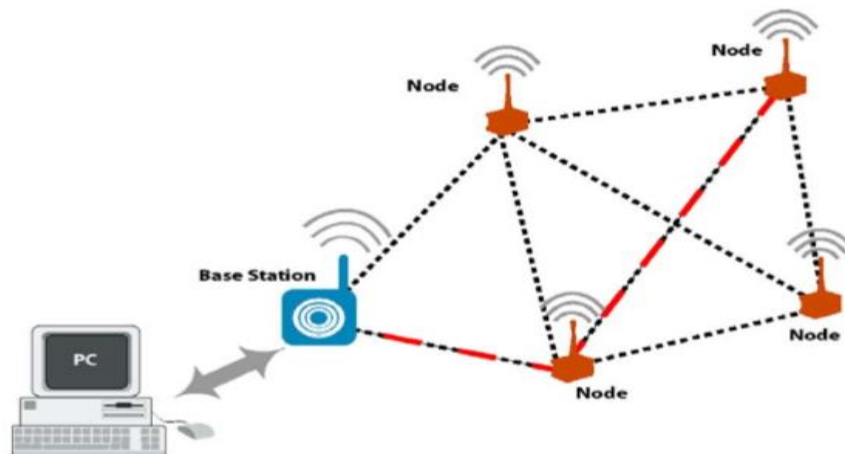


Fig2.5. XMesh Network Diagram

XMesh provides a TrueMesh networking service that is both self-organizing and self-healing. XMesh can be configured into various power modes including HP (high power), LP (low power), and ELP (extended low power). The XMesh networking protocol has various options including low-power listening, time synchronization, sleep modes, any-to-base and base-to-any routing.

The XMesh network has the following features:

- MICA2, MICA2DOT, and MICAz support.

- Low power (typically less than 220 μ A average current (without sensor board)).
- Network time synchronization to ± 1 msec.
- Low power listening with an 8 times per second wake-up interval, allowing for rapid message transfer across the network. The default sampling period is 3 minutes, although many other sampling intervals are allowed.

2.3.2 XMesh Network Landscape

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

1. 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.
2. 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.
3. 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.

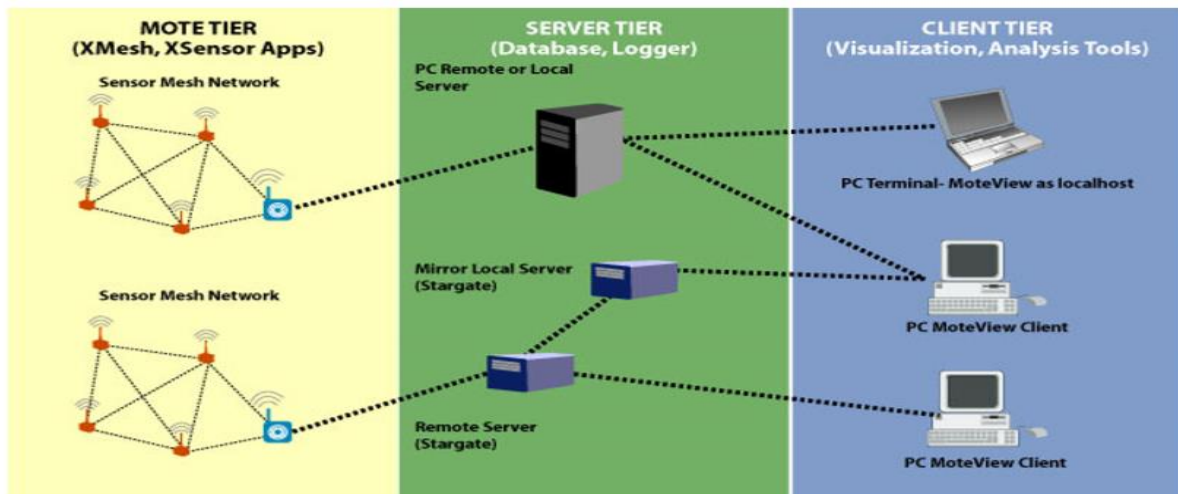


Fig 2.6. Software Framework of Internet of Things

An XMesh sensor network system consists of multiple motes (MICA2/MICAz) and a base station unit (MICA2/MICAz) installed on an interface board (e.g. MIB520). This base station Mote serves two purposes:

1. It acts as the Gateway between the Mote Tier and Server Tier. The base station communicates with other motes over the radio, and with the server using serial communication. In this way, the base station forms a bridge to send and receive messages between a host system (PC and/or Stargate) and the rest of the mesh network.
2. It forms the network and directs all data messages from the Motes to itself. To other Motes in the network, this base station Mote can forward messages to the PC (host) with zero energy cost. The base station Mote is always identified as “node 0” in a single base station system.

2.3.3 XMesh Features and Benefits

XMesh has many features and benefits. They include:

- **TrueMesh** - TrueMesh technology refers to the ability of the nodes to dynamically seek new routes for delivering packets when parts of the network go offline due to radio interference or power duty cycling
- **Multiple Transport Services** - XMesh provides multiple transport services for communication between nodes. They are:
 - Upstream – Delivers packets from a node to the base station Mote.
 - Downstream – Delivers packets from base station Mote to node(s)
 - Single Hop – Delivers packets to neighboring nodes only.
- **Multiple Quality of Service (QoS) Modes** - XMesh provides multiple qualities of service modes. They are:
 - **Best Effort** – Link level acknowledgement where motes will try multiple times to transmit a message to its immediate neighbor.
 - **Guaranteed Delivery** – Provides end-to-end acknowledgement where a message is transmitted through the mesh to the base station (or downstream) and an acknowledge message is then sent back to the originator.

- **Multiple Power Modes** - XMesh can be configured to run in one of the several power modes.

The modes are:

1. High Power (HP) – The HP mode provides:

- TrueMesh capability
- Every node in the network can route data
- High bandwidth, low latency (full channel utilization)
- Mote radios are always powered.

2. Low Power (LP) – The LP mode provides:

- TrueMesh capability
- Every node in the network can route data
- Low bandwidth, high latency (ideal for low data rate applications)
- Mote radios are normally in a low power sleep state and wake periodically to check for radio traffic.

3. Extended Low Power (ELP) – The ELP mode provides:

- Used only for end nodes of the network
- Nodes cannot route data
- Uses hybrid star mesh configuration

- **Health Diagnostics-** Within the XMesh network, nodes can automatically transmit health information to the base station. The health information includes data on how well the node is performing in the network with regards to radio traffic, battery voltage, and parent's node Radio Signal Strength Indicator (RSSI). The base station Mote will forward the health information data to MoteView and XSniffer to monitor and diagnose the health of XMesh.

- **Time Synchronization-** XMesh-LP support a network global time synchronization to ± 1 msec. The time stamping is used to synchronize radio messages but is also available to users to synchronize sensor measurements.

- **Over-the-Air-Programming (OTAP)-** OTAP XMesh supports Over-the-Air-Programming which allows users to reprogram all nodes in the mesh with new code. OTAP uses a directed downstream strategy that allows different code images to be sent to different Motes. This allows users to deploy networks of multiple sensor boards and only reprogram the units of

interest. OTAP also uses a promiscuous listening mode; motes that can overhear the new code download, and know that they also need the same image, will store the code transmissions.

MoteConfig

MoteConfig is a Windows-based GUI utility for programming Motes.

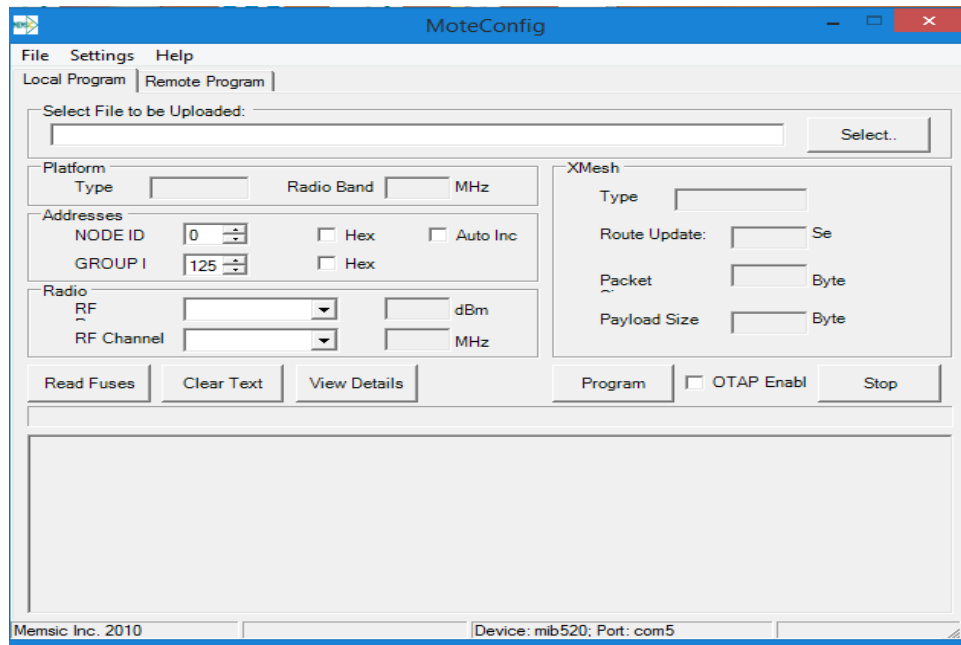


Fig 2.7 MoteConfig GUI

This utility provides an interface for configuring and downloading pre-compiled XMesh/TinyOS firmware applications onto Motes. MoteConfig allows the user to configure the Mote ID, Group ID, RF channel and RF power. The user can also enable the over-the-air-programming feature present on all XMesh - based firmware. High-power and low-power XMesh applications are available for each sensor board and platform.

Over – The – Air – Programming (OTAP)- The Over-The-Air-Programming (OTAP) feature allows users to reprogram a Mote over a wireless link. OTAP allows one or more Motes in the XMesh network to receive new firmware images from XServe (via the XOtapi service). Each Mote has a 512kB external non-volatile flash divided into 4 slots. These slots have a default size of 128 kB. Slot 0 is reserved for the OTAP image. Slots 1, 2 and 3 can be used for user-specified firmware. During the OTAP process, the server sends a command to the Mote to reboot into the OTAP image (slot 0). A user-specified firmware image is

broken up into fragments and transmitted to the Mote and stored into Slot 1, 2 or 3. The server can send a message to transfer the newly uploaded firmware into the program flash and reboot the Mote.

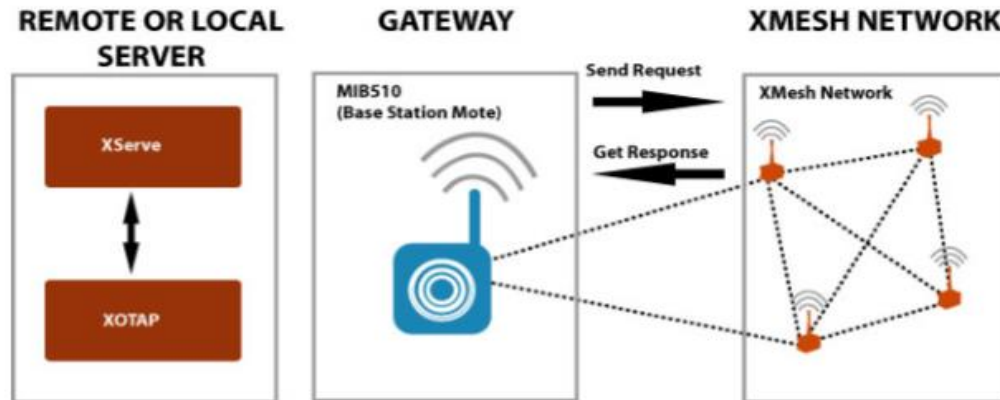


Fig 2.8: XOTAP Architecture

The following components are required for OTAP to work:

- XServe and XOTap running on the server
- Firmware applications that include the XOTAPLiteM component (this is automatically included when the firmware is built with XMesh)
- The Mote needs to have pre-configured with a bootloader in the program flash, and the OTAP image in slot 0 of the external flash.

2.4 MoteView

MoteView is designed to be an interface (“client tier”) between a user and a deployed network of wireless sensors. MoteView provides the tools to simplify deployment and monitoring. It also makes it easy to connect to a database, to analyze, and to graph sensor readings.

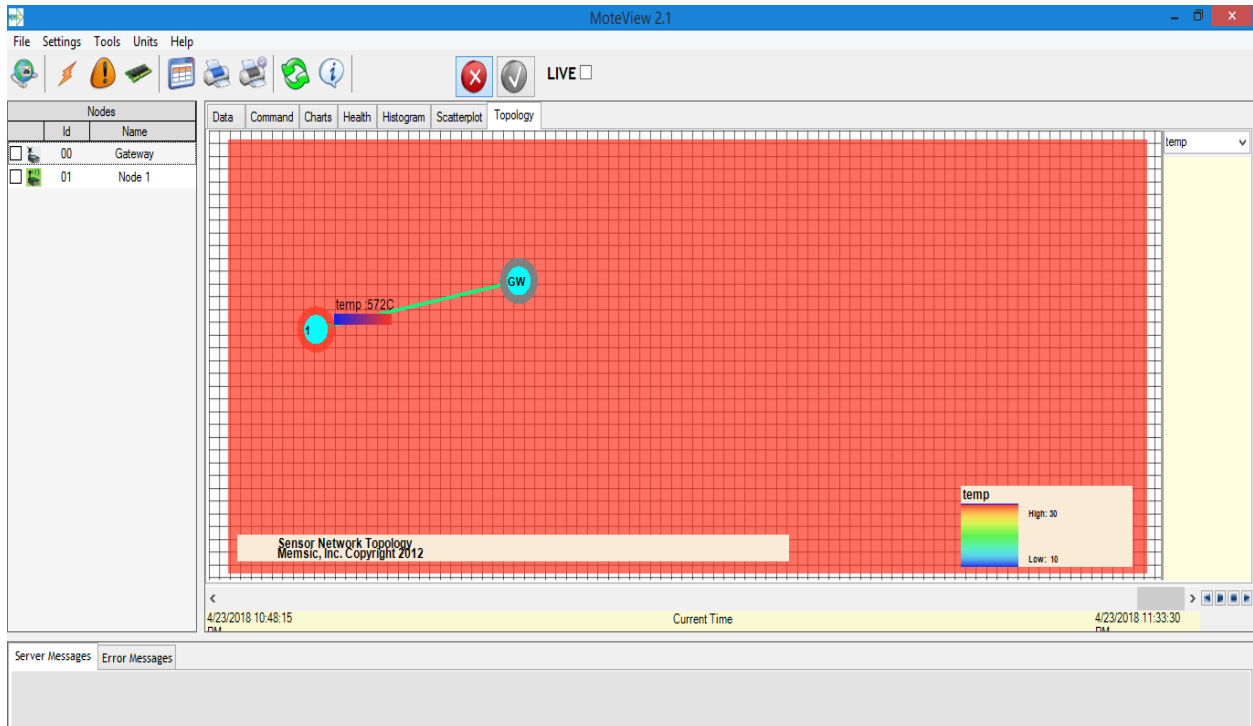


Fig 2.9. MoteView GUI

Wireless Mesh Networking Overview- Wireless sensor networks have attracted a wide interest from industry due to their diversity of applications. A key to realizing their potential is multi-hop mesh networking which enables scalability and reliability. A mesh network is a generic name for a class of networked embedded systems that share several characteristics including:

- **Multi-Hop**—the capability of sending messages peer-to-peer to a base station, thereby enabling scalable range extension;
- **Self-Configuring**—capable of network formation without human intervention;
- **Self-Healing**—capable of adding and removing network nodes automatically without having to reset the network; and
- **Dynamic Routing**—capable of adaptively determining the route based on dynamic network conditions (e.g., link quality, hop-count, gradient, or other metric).

When combined with battery power management, these characteristics allow sensor networks to be long-lived, easily deployed, and resilient to the unpredictable wireless channel. With mesh networking, the vision of pervasive and fine-grained sensing becomes reality.

2.5 MIB520 USB Interface Board

The MIB520 provides USB connectivity to the IRIS and MICA family of Motes for communication and in-system programming. It supplies power to the devices through USB bus. MIB520CB has a male connector while MIB520CA has female connector.

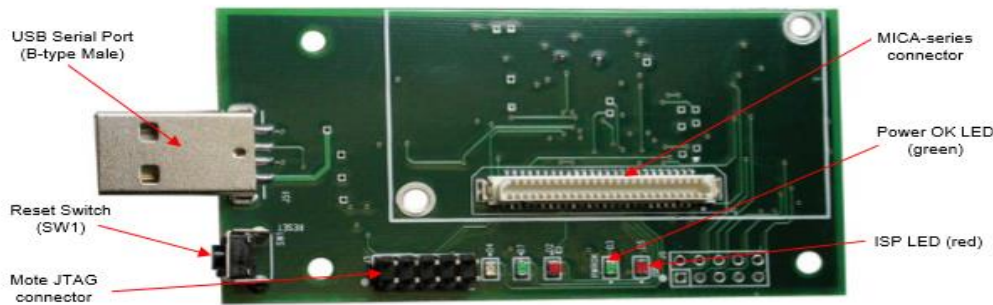


Fig 2.10. Photo of top view of an MIB520CB

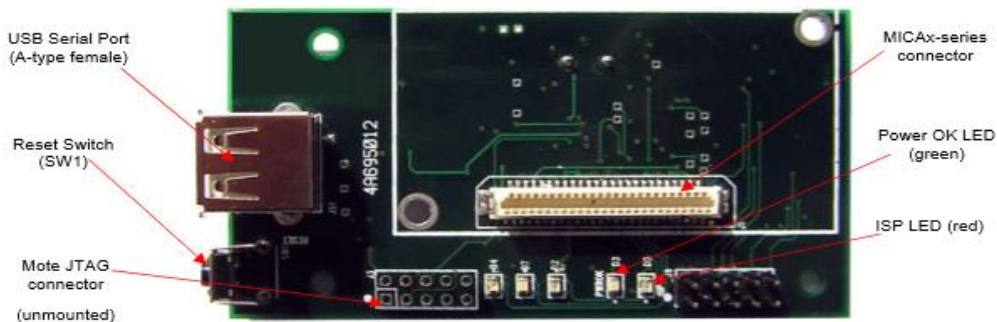


Fig 2.11. Photo of top view of an MIB520CA

ISP- The MIB520 has an on-board in-system processor (ISP)—an Atmega16L located at U14—to program the Motes. Code is downloaded to the ISP through the USB port.

Reset- The “RESET” push button switch resets both the ISP and Mote processors. It also resets the monitoring software which runs on the host PC.

JTAG - The MIB520 has a connector, J3 which connects to an Atmel JTAG pod for in-circuit debugging. This connector will supply power to the JTAG pod; no external power supply is required for the pod.

Power- The MIB520 is powered by the USB bus of the host.

USB- Interface the MIB520 offers two separate ports: one dedicated to in-system Mote programming and a second for data communication over USB.

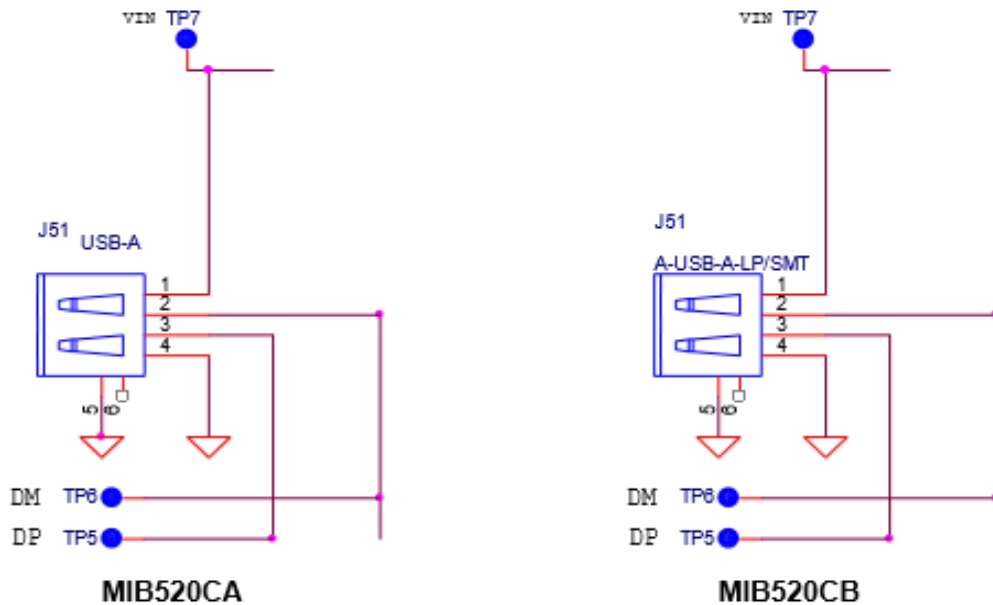
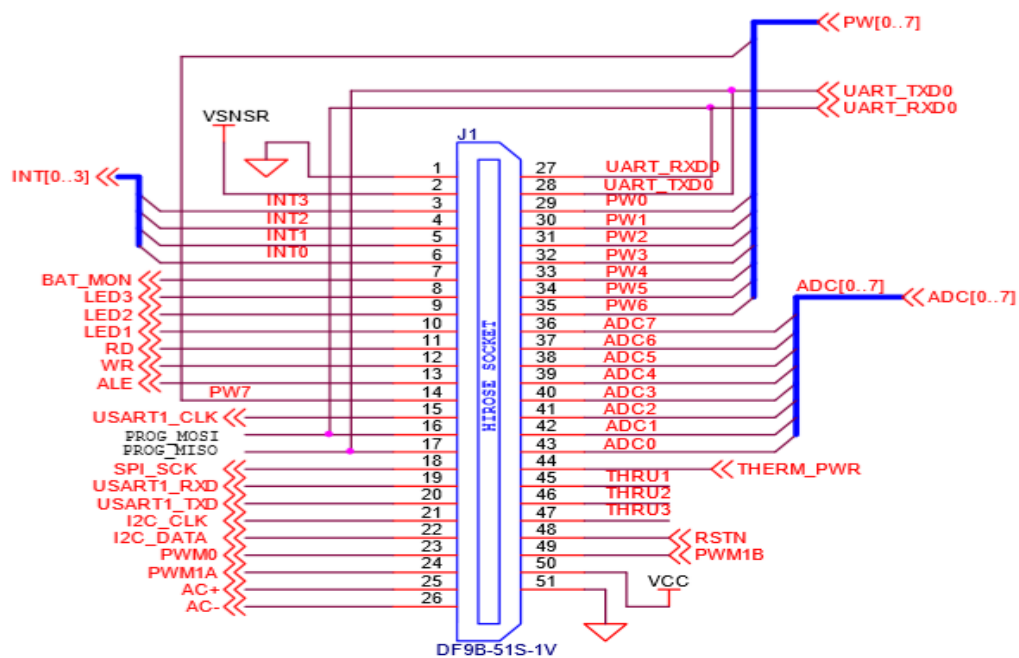


Fig 2.12. Pin Out for a USB Connection

51-Pin Mote Connector Interface



PIN	NAME	DESCRIPTION
1	GN	GROUND
2	VSN	SENSOR SUPPLY
3	INT3	GPIO
4	INT2	GPIO
5	INT1	GPIO
6	INT0	GPIO
7	BAT_MON	BATTERY VOLTAGE MONITOR ENABLE
8	LED3	LED3
9	LED2	LED2
10	LED1	LED1
11	RD	GPIO
12	WR	GPIO
13	ALE	GPIO
14	PW7	POWER CONTROL 7
15	USART1_CLK	USART1 CLOCK
16	PROG_MOSI	SERIAL PROGRAM MOSI
17	PROG_MISO	SERIAL PROGRAM MISO
18	SPI_SCK	SPI SERIAL CLOCK
19	USART1_RXD	USART1 RX DATA
20	USART1_TXD	USART1 TX DATA
21	I2C_CLK	I2C BUS CLOCK
22	I2C_DATA	I2C BUS DATA
23	PWM0	GPIO/PWM0
24	PWM1A	GPIO/PWM1A
25	AC+	GPIO/AC+
26	AC-	GPIO/AC-

PIN	NAME	DESCRIPTION
27	UART_RXD0	UART_0 RECEIVE
28	UART_TXD0	UART_0 TRANSMIT
29	PW0	POWER CONTROL 0
30	PW1	POWER CONTROL 1
31	PW2	POWER CONTROL 2
32	PW3	POWER CONTROL 3
33	PW4	POWER CONTROL 4
34	PW5	POWER CONTROL 5
35	PW6	POWER CONTROL 6
36	ADC7	ADC INPUT 7 - BATTERY MONITOR/JTAG TDO
37	ADC6	ADC INPUT 6 / JTAG TDO
38	ADC5	ADC INPUT 5 / JTAG TMS
39	ADC4	ADC INPUT 4 / JTAG TCK
40	ADC3	ADC INPUT 3
41	ADC2	ADC INPUT 2
42	ADC1	ADC INPUT 1
43	ADC0	ADC INPUT 0 / RSSI MONITOR
44	THERM_FWR	TEMP SENSOR ENABLE
45	THRU1	THRU CONNECT 1
46	THRU2	THRU CONNECT 2
47	THRU3	THRU CONNECT3
48	RSTN	RESET (NEG)
49	PWM1B	GPIO/PWM1B
50	VCC	DIGITAL SUPPLY
51	GND	GROUND

2.6 MDA100

The MDA100 series sensor boards have a precision thermistor, a light sensor/photocell, and general prototyping area. The prototyping area supports connection to all eight channels of the Mote's analog to digital converter (ADC0–7), both USART serial ports and the I2C digital communications bus. The prototyping area also has 45 unconnected holes that are used for breadboard of circuitry.



Fig 2.13. Pin Out of MDA100

2.7 XM2110 (IRIS)

The IRIS is the most recent age of Motes from Memsic. The XM2110 (2400 MHz to 2483.5 MHz band) utilizes the Atmel RF230, IEEE 802.15.4 agreeable, ZigBee prepared radio recurrence handset incorporated with an Atmega1281 smaller scale controller. These upgrades give up to three times enhanced radio range and double the program memory over past age

MICA Motes. A similar MICA family, 51 stick I/O connector, and serial Flash memory is utilized; all application programming and sensor sheets are good with the XM2110.



Fig 2.14. Photo of the XM2110—IRIS with standard antenna

Block Diagram and Schematics for the XM2110 / IRIS

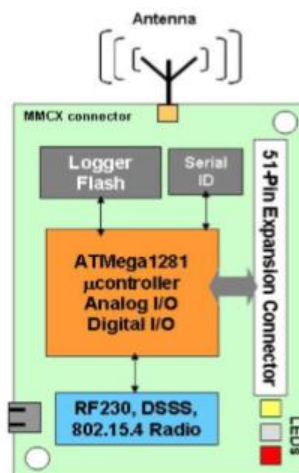


Fig 2.15. Block diagram of the IRIS

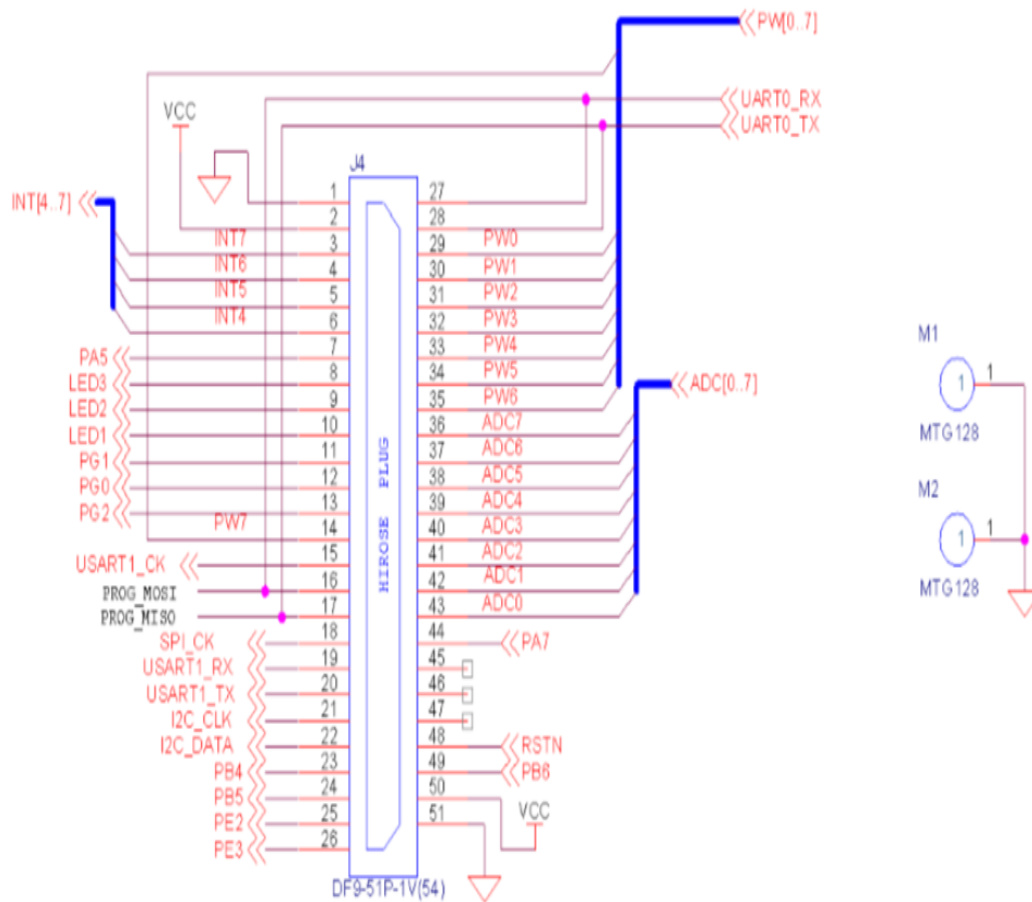


Fig 2.16. 51-pin Expansion Connector

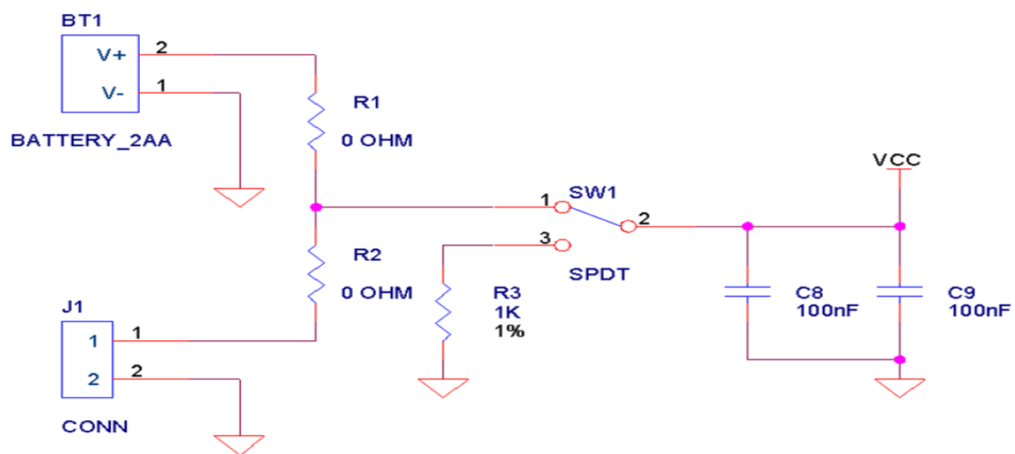


Fig 2.17. Circuit Diagram of Battery

2.8 6LoWPAN

The 6LoWPAN innovation uses IEEE 802.15.4 to give the lower layers to this low power remote system framework. While this appears a direct way to deal with the improvement of a packet information remote system or remote sensor arrange, there are incongruencies between IPv6 organize and the configurations allowed by IEEE 802.15.4. This distinctions are overcome inside 6LoWPAN and this allows the framework to be utilized as a layer over the fundamental 802.15.4.

Keeping in mind the end goal to send packet information, IPv6 more than 6LowPAN, it is important to have a technique for changing over the packet information into an arrangement that can be taken care of by the IEEE 802.15.4 lower layer framework.

IPv6 requires the greatest transmission unit (MTU) to be no less than 1280 bytes long. This is significantly longer than the IEEE802.15.4's standard packet size of 127 octets which was set to keep transmissions short and along these lines diminish power utilization.

To conquer the address determination issue, IPv6 nodes are given 128 piece addresses in a various leveled way. The IEEE 802.15.4 gadgets may utilize both of IEEE 64 bit expanded addresses or 16 bit addresses that are interesting inside a PAN after gadgets have related. There is additionally a PAN-ID for a gathering of physically co-found IEEE802.15.4 gadgets.

CHAPTER 3: ENVIRONMENT TEMPERATURE MONITORING SYSTEM

3.1 Introduction

The surroundings or conditions in which a person, animal, or plant lives or operates is Environment. **Measuring temperature** is one of the most common technique used because it is important for many operations and tasks to be performed like in any industries where heaters are used, heat up to a certain temperature is required. When it comes to sensing temperature, a temperature sensor is used that is installed at a place whose temperature is to be sensed. The temperature of that place can be **monitored through internet using internet of things**. the web-based temperature monitoring system that can be access anywhere and anytime through the Internet is build. With this system a user can remotely monitor the room temperature from anywhere which could save the human expenses.

IoT Web Based Temperature Monitoring is one type of temperature recorder that monitors a temperature in a room and stores the data into a database and display the current temperature through a web server. The system will continuously monitor the temperature condition of the room and the data can be monitored at anytime and anywhere from the Internet. The temperature monitoring is widely used in various processes like in automotive industries, air conditioning, power plant and other industries that need the data to be saved and analyzed. The main purpose of this system model is to make it easy for the user to view the current temperature.

3.2 Tools-

Software: MoteView, Moteconfig

Hardware: MIB520 USB Interface Board, XM2110 (IRIS),

Sensors: Thermistor, CdSe photocell

Language: NesC

3.3 Circuit Diagram of Temperature Monitoring System

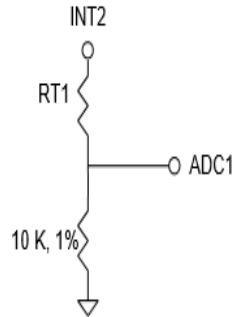


Fig 3.1. Schematic of the Thermistor on MDA100CA

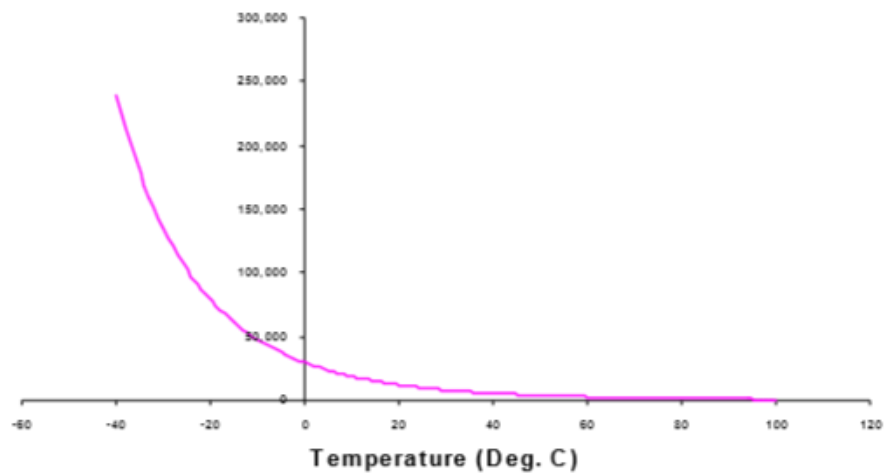


Fig 3.2. Resistance vs. Temperature Graph

3.4 Conversion to Engineering Units

The Mote's ADC output can be converted to Kelvin using the following approximation over 0 to 50 °C: $1/T(K) = a + b \times \ln(R_{thr}) + c \times [\ln(R_{thr})]^3$

Where: $R_{thr} = R1 (ADC_FS - ADC) / ADC$

$a = 0.001010024$

$b = 0.000242127$

$c = 0.000000146$ $R1 = 10 \text{ k}\Omega$

ADC_FS = 1023, and

ADC = output value from Mote's ADC measurement.

3.5 Implementation:

Local Programming:

1. Open MoteConfig 2.0 and Click on Settings > Interface Board... to select the correct gateway and port settings.
2. Select MIB520 Gateway and choose low numbered port as it is used for programming and the high-numbered port is used for communication.

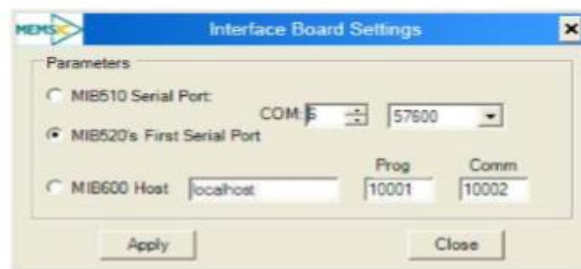


Fig 3.3. MIB520 Gateway Settings

3. The pre-compiled XMesh applications installed with MoteView are located in C:\Program Files (x86)\Memsic\MoteView\xmesh\iris

We have to program Base station Mote and Remote nodes

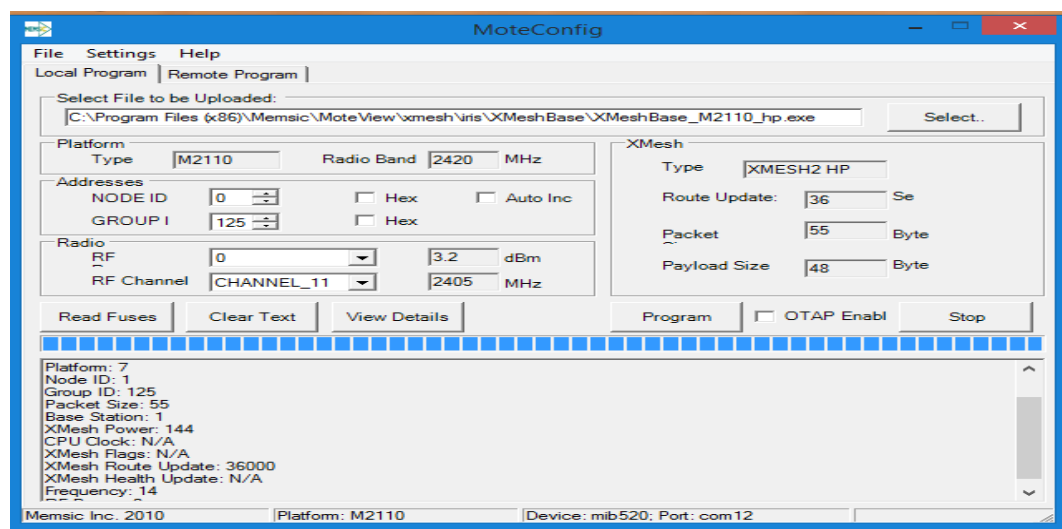


Fig 3.4. Programming Base station mote with Node Id 0

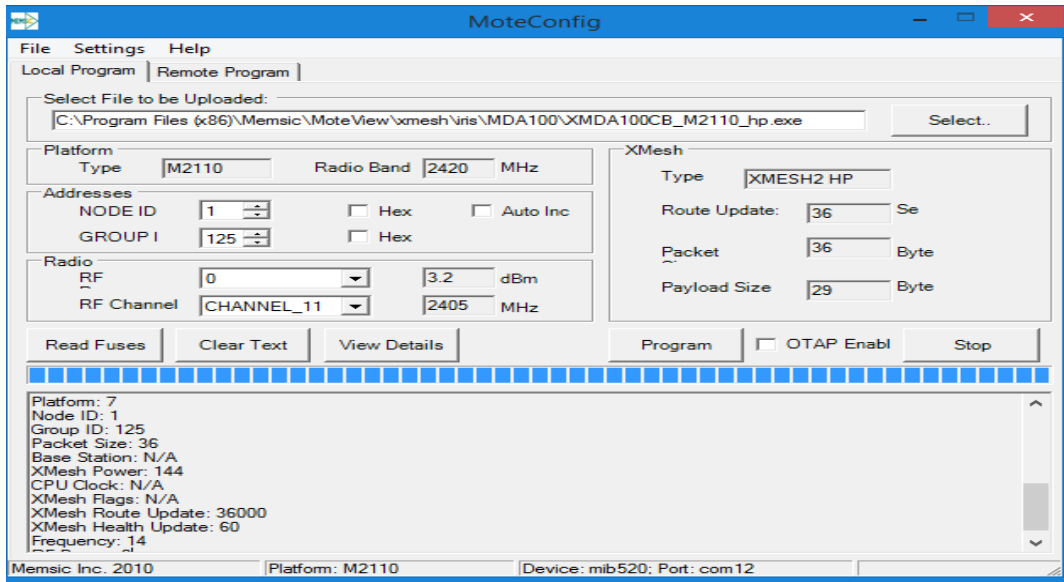


Fig 3.5. Programming Remote nodes with Node Id 1

Connecting to a Live Sensor Network on our local PC:

1. Open MoteView and Select File > Connect to WSN... from the menu. Select the Mode tab, check on Acquire Live Data as operation mode and Local as acquisition type and click on Next >>.
2. In the Gateway tab, specify the Interface Board type, Port/Host Name etc. as described below. Since we are using MIB520, enter the higher of the 2 COM ports installed by the MIB520's driver and set the baud rate to 57600 and click next

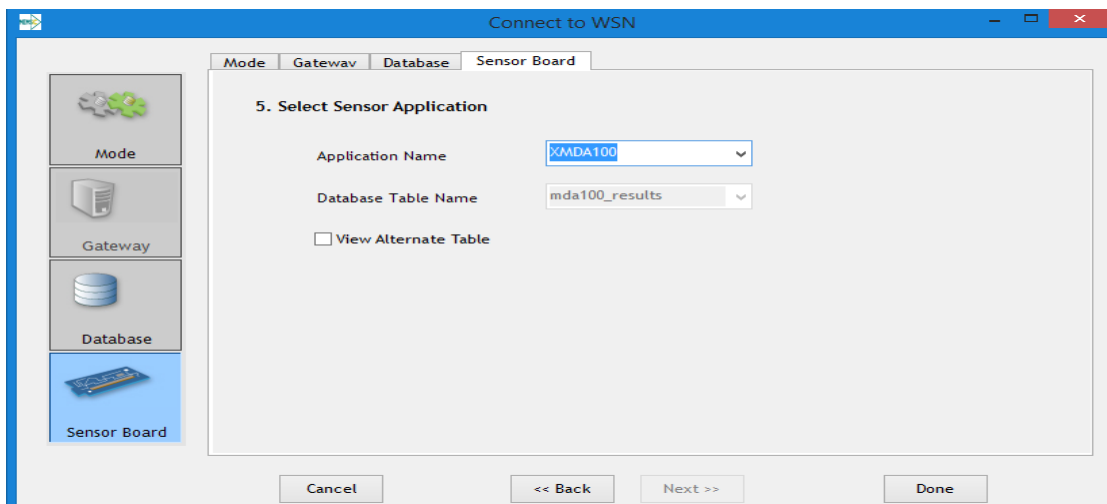


Fig 3.6 Connecting sensor Network

3. Select available database and click Next
4. In the Sensor Board tab, uncheck the View Alternate Table checkbox and choose the XMesh Application Name that matches the firmware programmed into the Mote (i.e. XMDA100) from Application Name dropdown. Click on Done

3.6 Visualization

Seven visualization tabs (Data, Command, Charts, Health, Histogram, Scatterplot and Topology) provide different methods of viewing our sensor data.

a). Topology

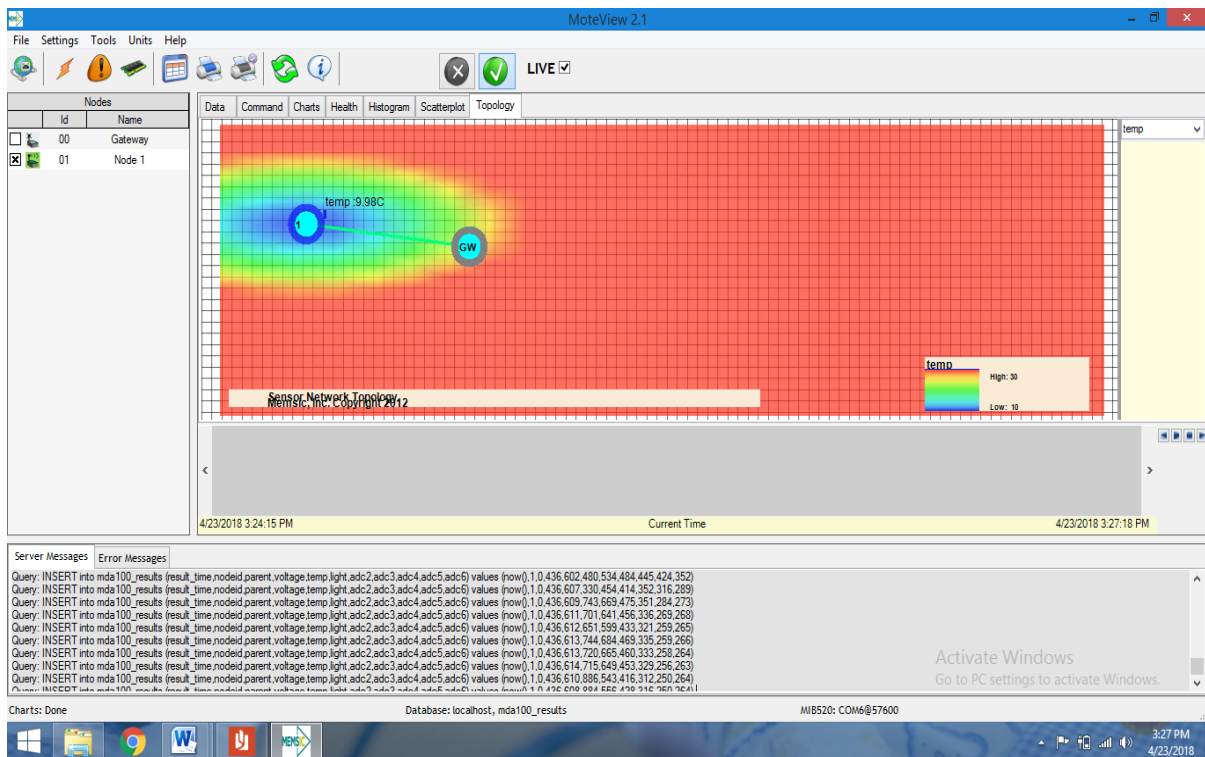


Fig 3.7. Temperature Variation of Sensor Nodes

Figure shows one sensor node is held in hand and we can see temperature of the surrounding increasing. Snapshot of the reading is taken at 9.98°C which is still increasing. Showing Blue color for minimum temperature which is changing as the temperature increases in the surrounding.

b). Histogram

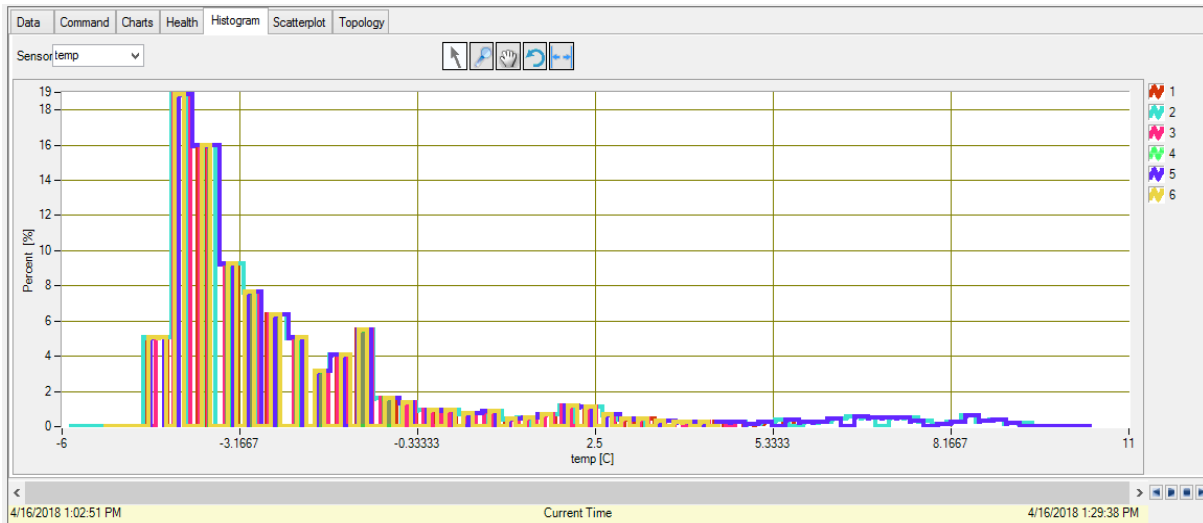
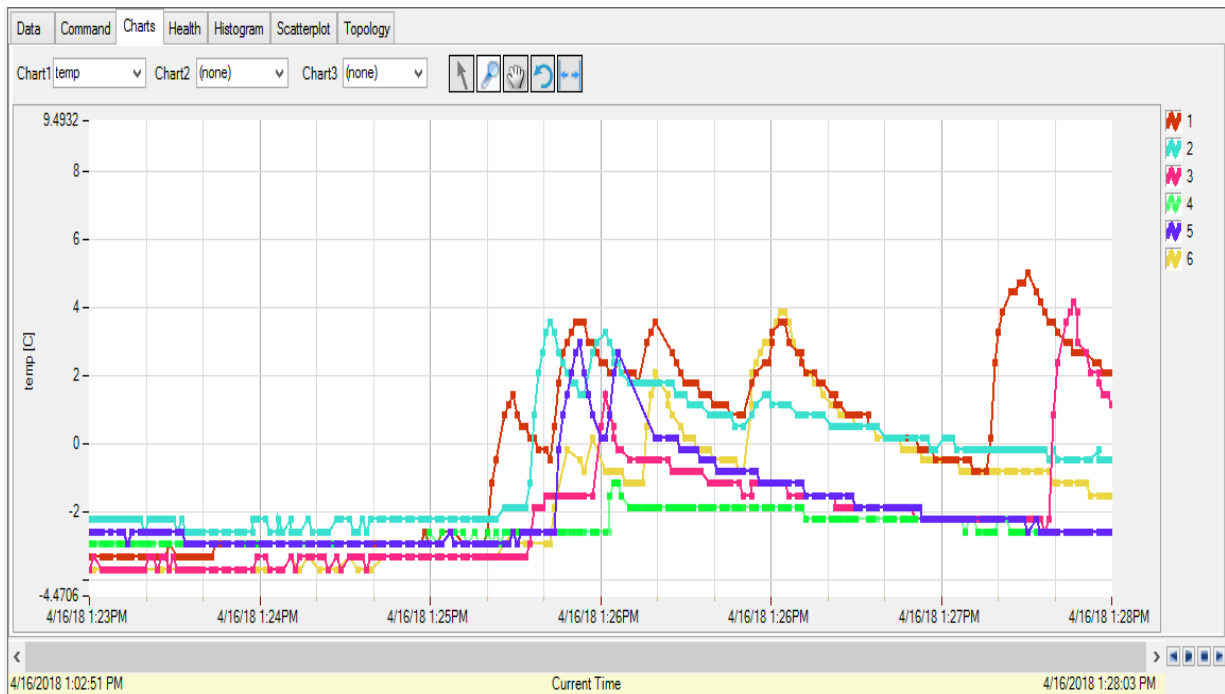


Fig 3.8. Histogram of temperature variation of different nodes

Figure shows histogram of temperature of different nodes. Node 5 is having the maximum temperature of as compared to all other nodes. Meanwhile node 4 has touched 19% i.e the maximum on percentage chart. Temperature of node 6 vary continuously, at starting it reached to 19% and then it is continuously decreasing.

c). Charts



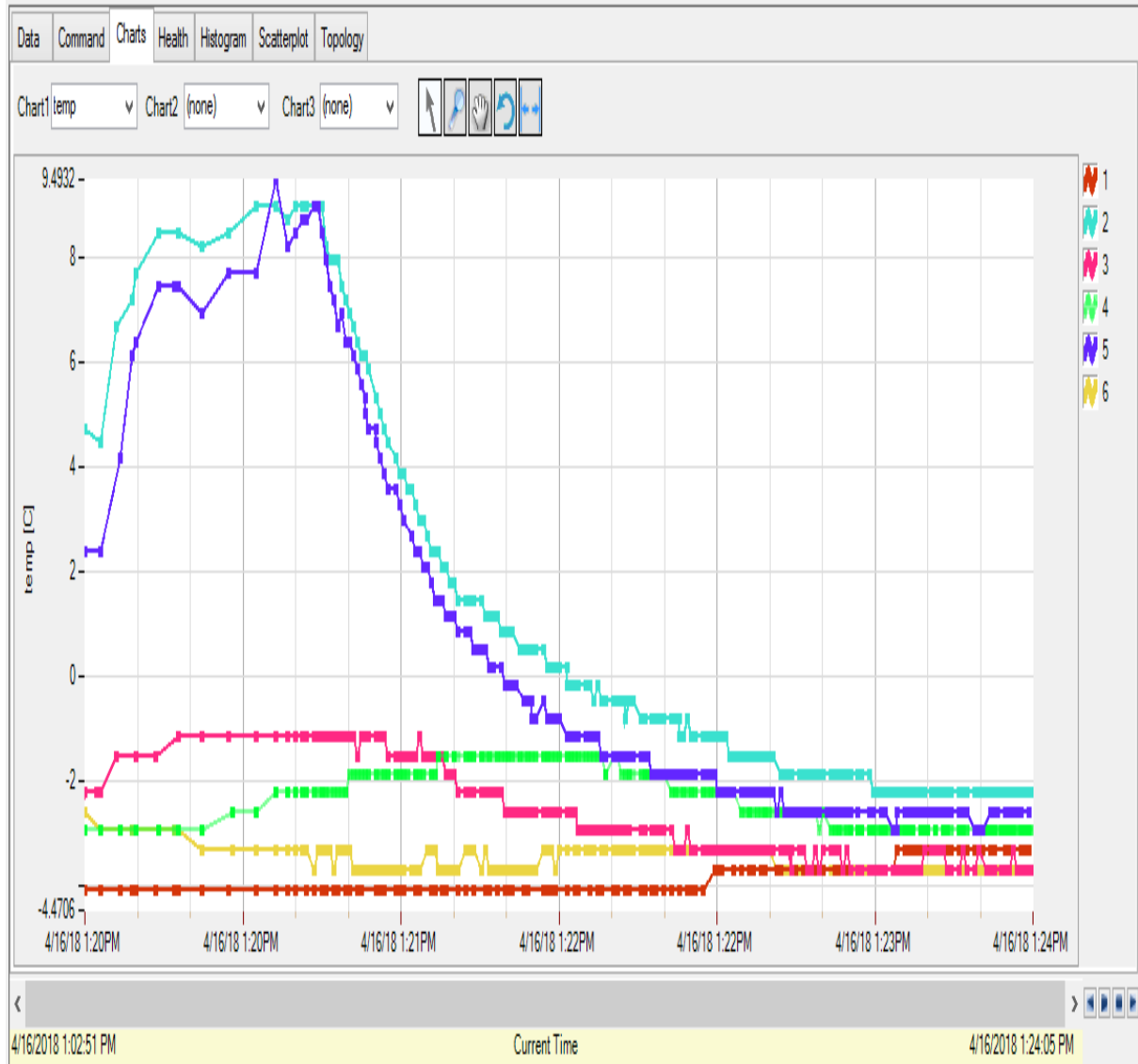


Fig 3.9 Temperature variation of nodes with respect to time

Figure shows temperature of different surroundings when kept at different positions on our Information communication technology building. The maximum temperature reached is 5°C. of node 1. Similarly ode 4 and node 2 have drastic changes in their readings.

Chart showing temperature variation for node 5 and node 2 vary as we move them in the sunlight. As node 2 and node 5 were subjected to sunlight node 5 reached a max temptemperature of 9.4932 at time 1:20:17pm (approx). since the sensor nodes arent calibrated, nodes are showing low temperature readings.

d). Topology

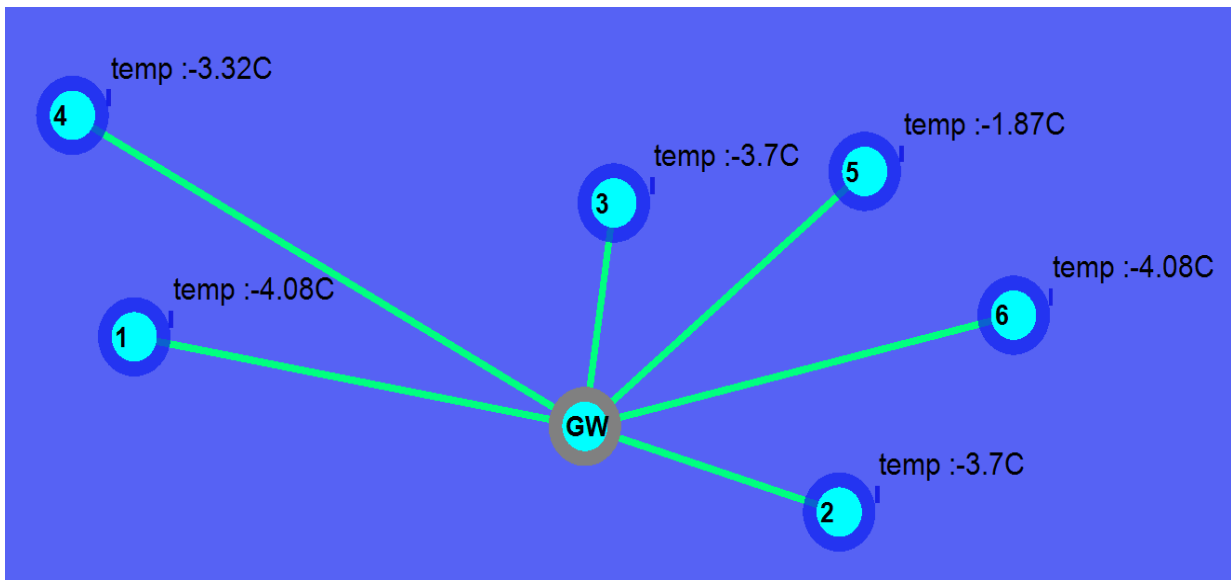


Fig 3.10 Star Topology of different nodes

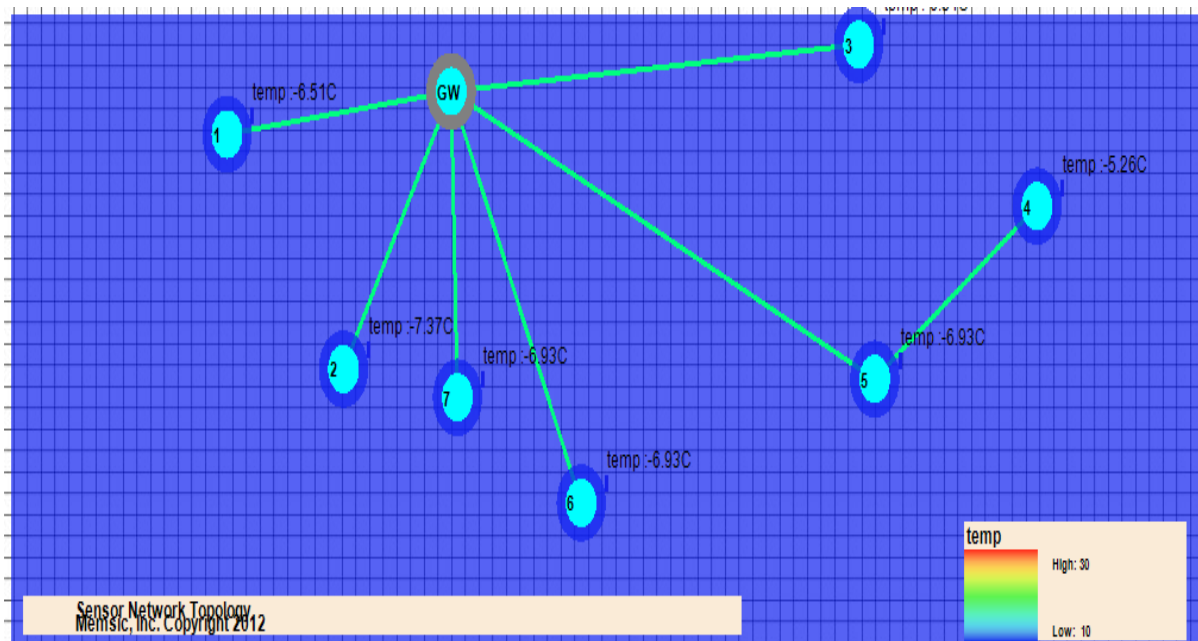


Fig 3.10 Star-Mesh Topology of different nodes

Figure shows star topology is followed by our wireless sensor gateway. Nodes are operating at different temperatures as shown in the figure.

3.7 Sending Alert Message

The acronym **SMTP** stands for **Simple Mail Transfer Protocol**, the procedure behind the email flow on the internet. The process of email delivery is actually quite similar to classical mail. Basically, the journey of a message from computer to the recipient's is like this:

- We send an email with our webmail or **mail client** from our address (e.g. mark@website.com) to a given recipient (e.g. jane@domain.com).
- The message is sent normally via **port 25** to an SMTP server (named for instance mail.website.com) which is given to our client when we set it up and acts as a Message Transfer Agent or MTA. Client and server start a brief "conversation" where the latter checks all the data concerning the message's transmission (sender, recipient, domains, etc.).
- Then, if the domain where our recipient has his account is directly connected to the server, the email is immediately delivered. If it's not the case, the SMTP hands it to another **incoming server** closer to the recipient.
- What if the recipient's server is down or busy? The SMTP host simply drops the message to a **backup server**: if none of them is available, the email is queued and the delivery is retried periodically. After a determined period, however, the message is returned as undelivered.
- If there are no issues, however, the final segment is controlled by **POP**, another protocol that picks up the email from the receiving server and puts it into the recipient's inbox.

1. Setting up the SMTP for mail notification

Open MoteView. On top left corner look for tools>>alerts>>Alert Mail Configure.

Write our SMTP server information in the corresponding edit text. Then click on Test Mail.

MailConfig

Mail Server Information

SMTP Server: smtp.mailtrap.io

User Name: ec826050e15d88 Password:

Send To: kshitzsaini1941996@gmail.com

Message: fire in the city

OK Cancel Test Mail

Fig 3.11 SMTP Mail Configuration

2. Open Alert Manager from tools>>alerts>>Alert Manager Then Set the Alert Threshold ,Alert Action, Interval etc.

[illegible]

Fig 3.12 Setting the conditions for Alert Message

3. When the Temperature reached the Alert Threshold then mail is send to SMTP server or Pop-up Alert is shown based on the Actions set.

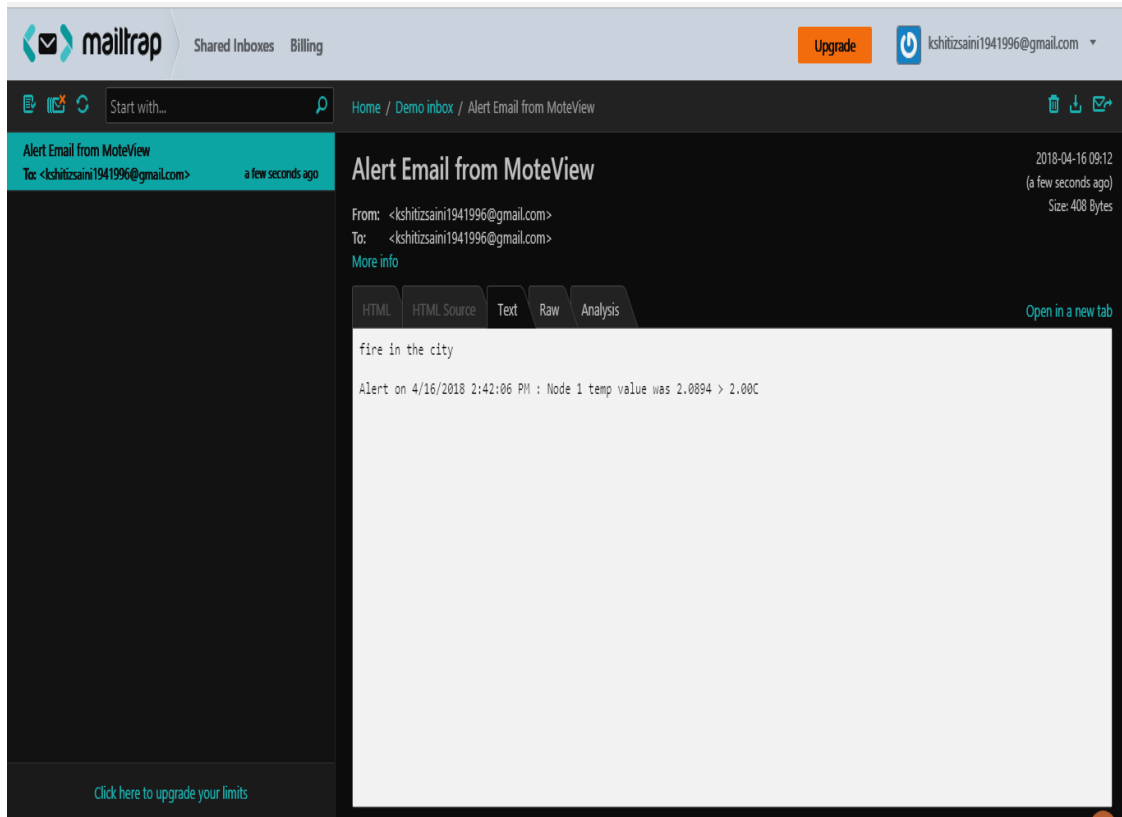


Fig 3.13. Alert message on SMTP server

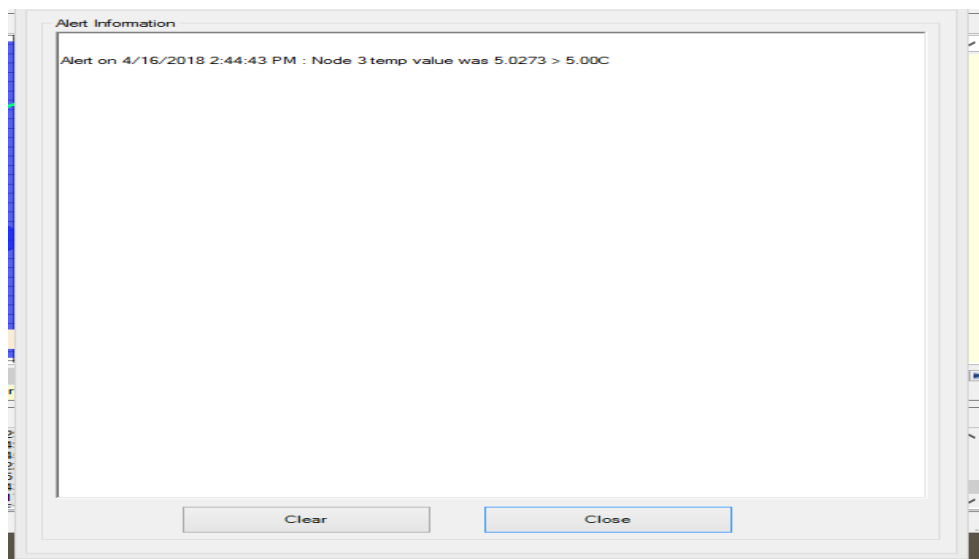


Fig 3.14 Pop up Alert Message

3.8 Conclusion

The IoT nodes are providing us the data of the temperature of the surroundings of the node. We can monitor the data and extract essential information about the surrounding. If we are building a large-scale project in which environment plays an important role in construction and sustainment of the project, this system will be of extreme help in the above scenario. So, we can conclude that the Smart Environmental Temperature Monitoring System is constructed and implemented successfully.

CHAPTER 4: ANTI-THEFT MONITORING SYSTEM

4.1 Introduction

An anti-theft system is any device or method used to prevent or deter the unauthorized appropriation of items considered valuable. Theft is one of the most common and oldest criminal behaviours. From the invention of the first lock and key to the introduction of RFID tags and biometric identification, anti-theft systems have evolved to match the introduction of new inventions to society and the resulting theft by others. In this chapter, we will discuss and implement the anti-theft monitoring system using photoresistor.

A photoresistor (or light-dependent resistor, LDR, or photo-conductive cell) is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. A photoresistor can be applied in light-sensitive detector circuits, and light-activated and dark-activated switching circuits. The light sensor we are using is a CdSe (Cadmium selenide) photocell. The maximum sensitivity of the photocell is at the light wavelength of 690 nm. Typical on resistance, while exposed to light, is 2 k Ω . Typically off resistance, while in dark conditions, is 520 k Ω

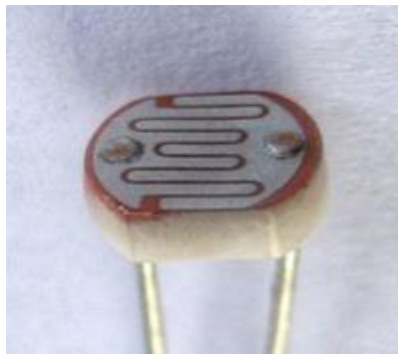


Fig 4.1. Photoresistor

4.2 Tools

Software: MoteView, Moteconfig.

Hardware: MIB520 USB Interface Board, XM2110 (IRIS), Laser Light or other Source of light like LED.

Sensors: Thermistor, CdSe photocell.

Language: NesC.

4.3 Visualization

Seven visualization tabs (Data, Command, Charts, Health, Histogram, Scatterplot and Topology) provide different methods of viewing our sensor data.

a) Topology

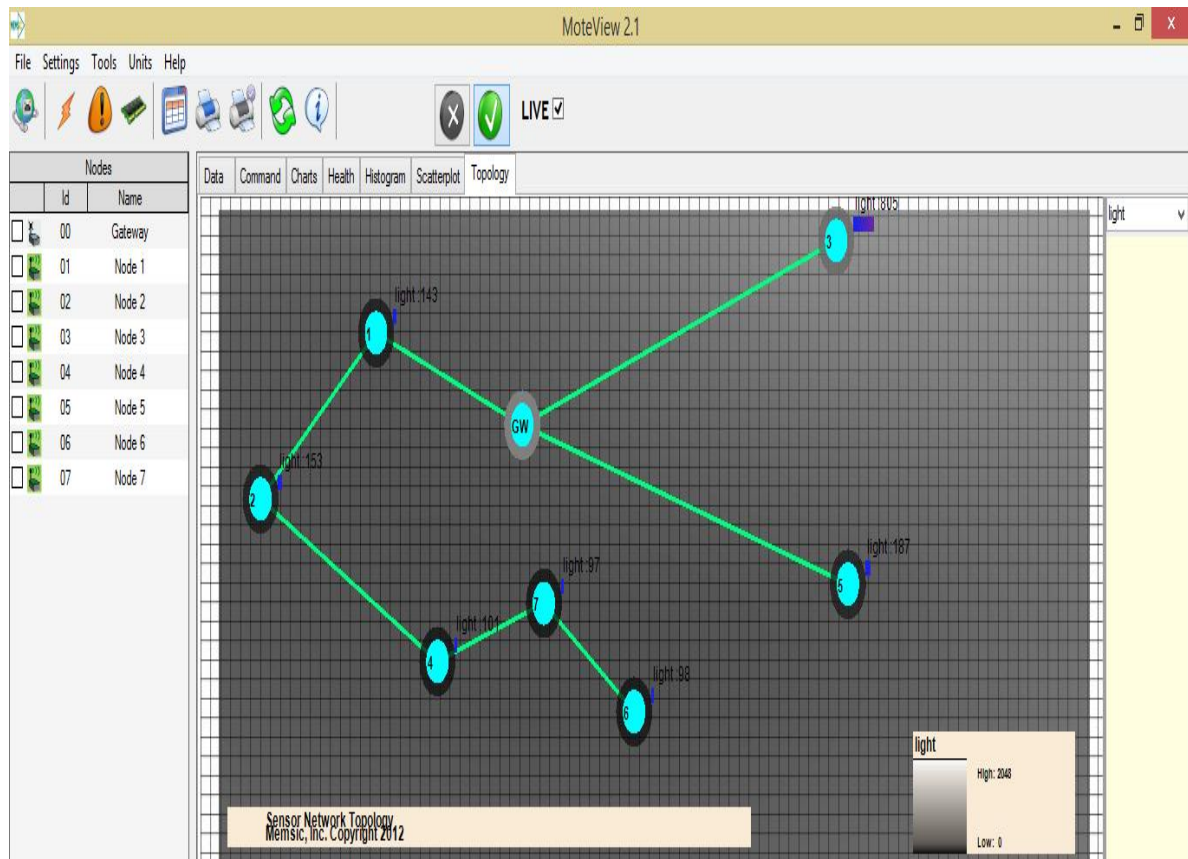


Fig 4.2 Hybrid Topology

Figure shows the hybrid topology is followed by the network. Since node 1,2,4,7,6 are in bus and node 3 and 5 are connected to the gateway directly. These nodes are currently sensing the intensity of light. Node 3 is getting has a highest value of light intensity as compared to all other nodes.

b) Data

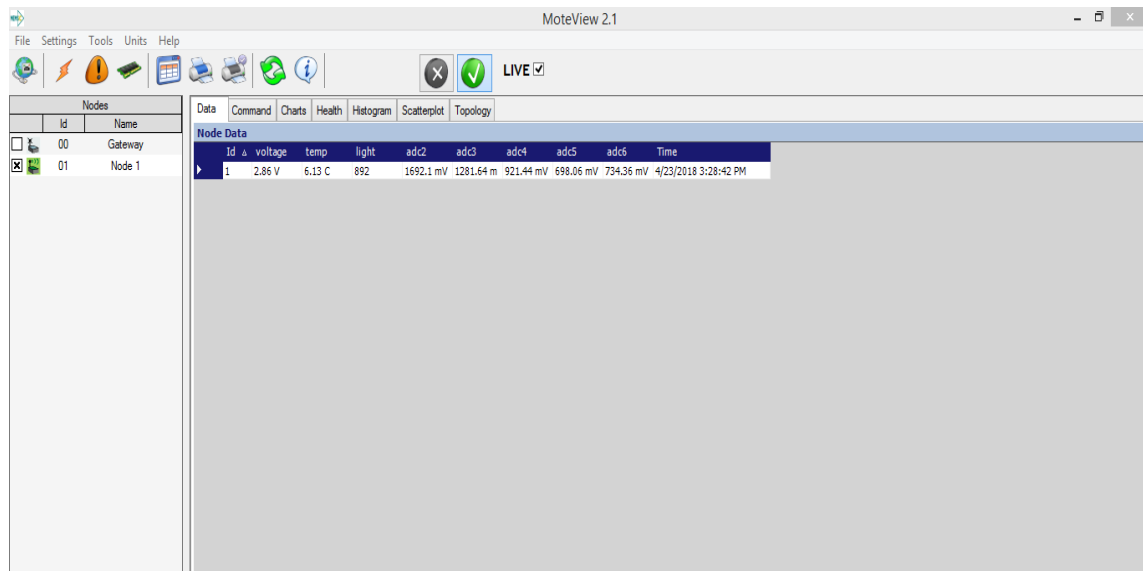


Fig 4.3. Data received through light sensor

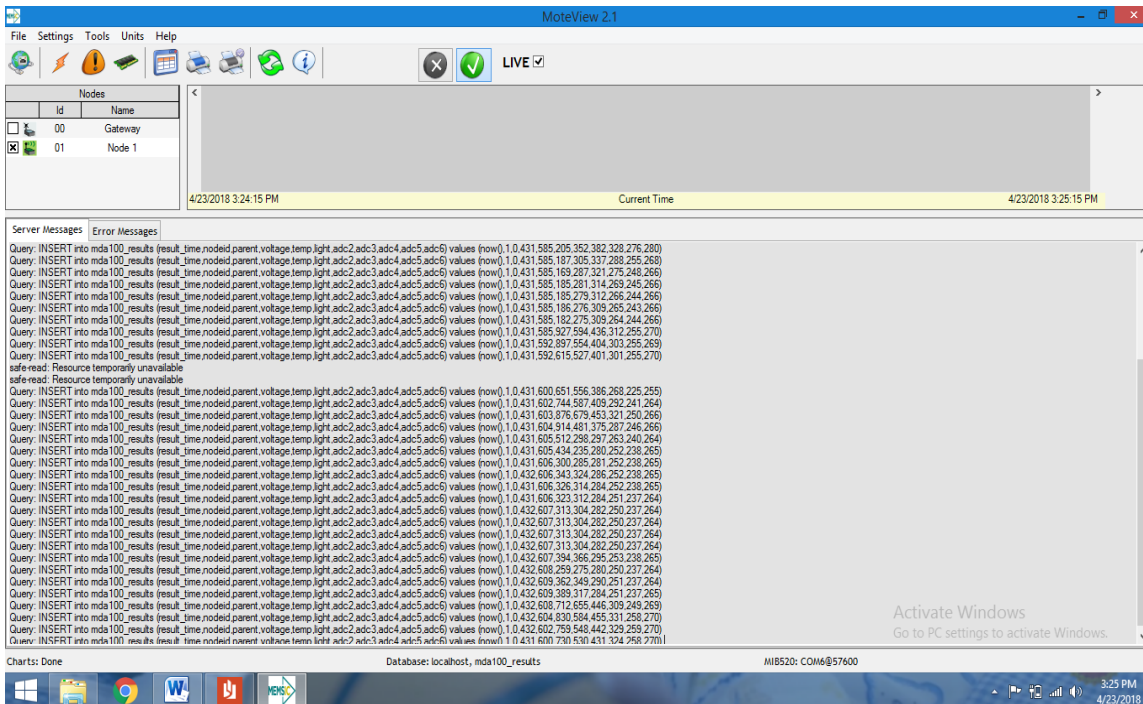


Fig 4.4. Query inserted into MDA 100 per second

Figure shows that our node is successfully connected with our gateway and is transmitting the sensed data. Gateway on the other end is also configured correctly. Xserve.exe application is running, since its received the heartbeat signal from the node.

c) Charts

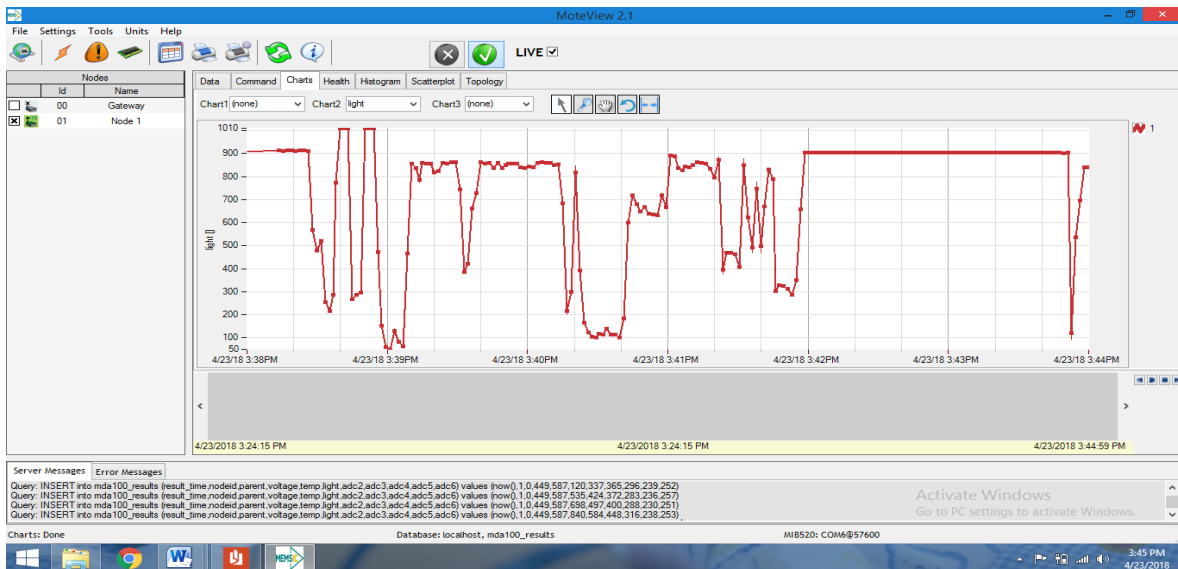


Fig 4.5. Variation of light intensity

Figure showing chart of intensity of light. The maximum intensity is 1010 and the minimum is 50. Intensity 1010 means someone has opened the locker. Intensity 50 means the locker is closed with a very little amount of light entering inside the locker.

d) Histogram

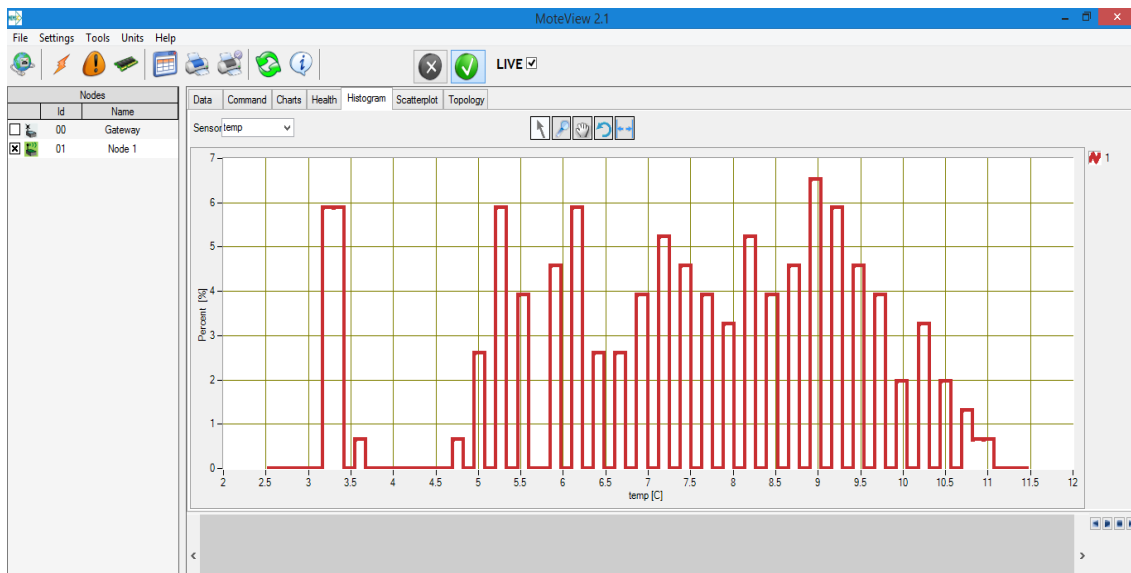
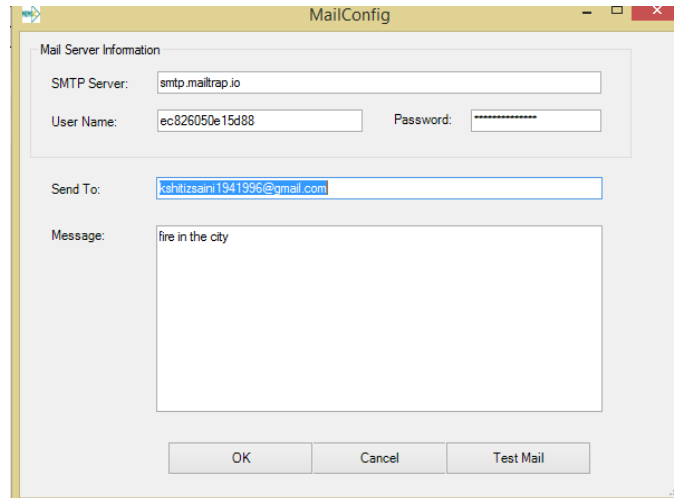


Fig 4.6. Histogram of nodes

4.4 Sending Alert Message

1. Setting up the SMTP for mail notification

Open MoteView. On top left corner look for tools>>alerts>>Alert Mail Configure.



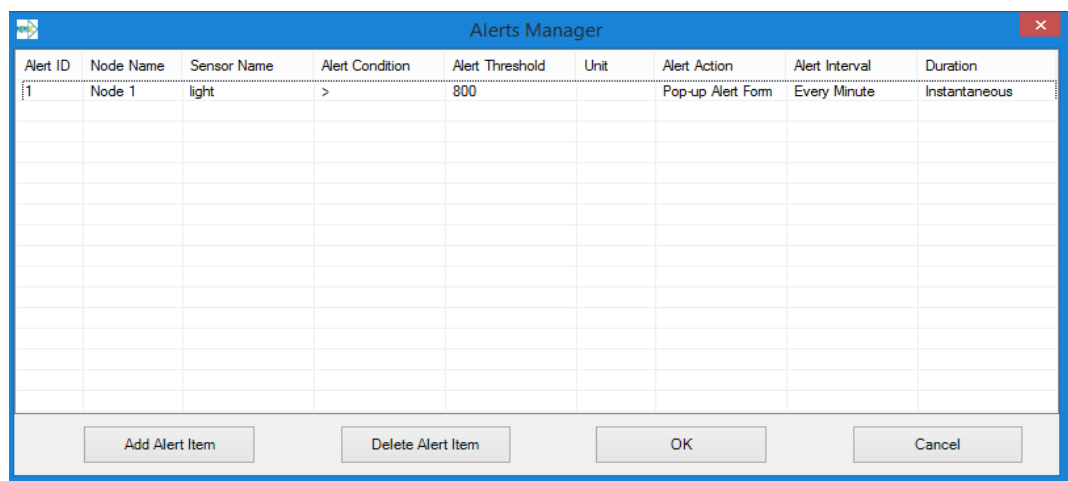
The MailConfig dialog box is used for setting up SMTP mail notification. It contains the following fields and buttons:

- Mail Server Information:**
 - SMTP Server: smtp.mailtrap.io
 - User Name: ec826050e15d88
 - Password: (masked with asterisks)
- Send To:** shritzsaini1941996@gmail.com
- Message:** fire in the city
- Buttons:** OK, Cancel, Test Mail

Fig 4.7. Alert Mail Configuration

Write our SMTP server information in the corresponding edit text. Then click on Test Mail.

2. Open Alert Manager from tools>>alerts>>Alert Manager Then Set the Alert Threshold, Alert Action, Interval etc.



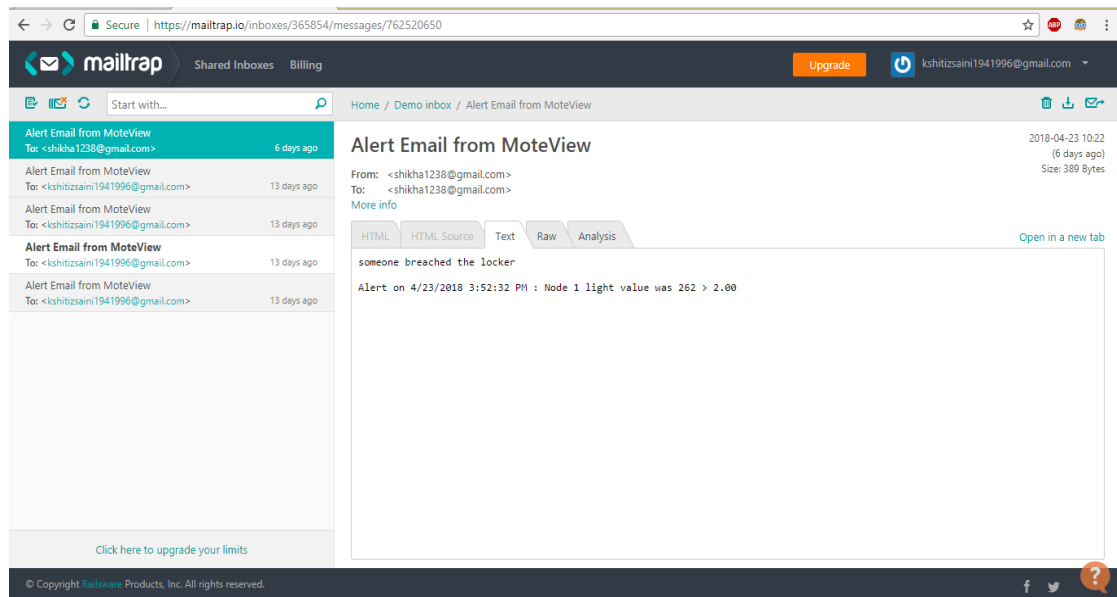
The Alerts Manager dialog box is used for setting up alert conditions. It contains a table with the following columns and data:

Alert ID	Node Name	Sensor Name	Alert Condition	Alert Threshold	Unit	Alert Action	Alert Interval	Duration
1	Node 1	light	>	800		Pop-up Alert Form	Every Minute	Instantaneous

Buttons: Add Alert Item, Delete Alert Item, OK, Cancel

Fig 4.8. Setting the Alert message condition

- When the Temperature reached the Alert Threshold then mail is send to SMTP server or Pop-up Alert is shown based on the Actions set



4.

Fig 4.9 Alert Message through SMTP Server

Figure above is the screenshot of the mail sent to shikha123@gmail.com stating someone has breached the locker. This email was generated by MoteView application.

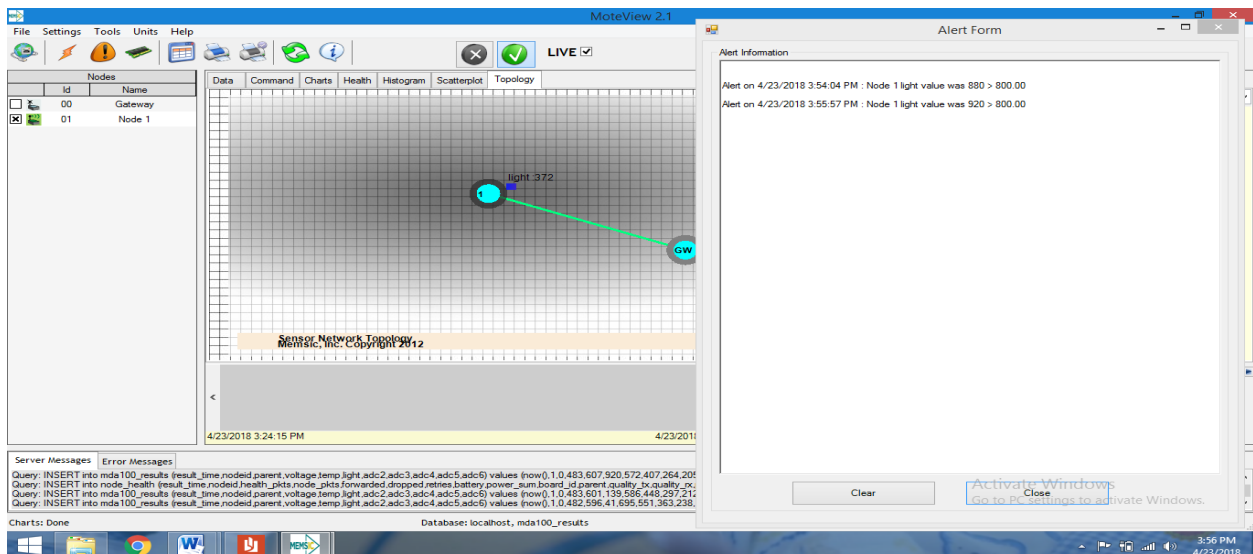


Fig 4.10. Alert Message through Pop-up Box

Figure showing alert form box popped as someone has opened the locker. Since we have set a condition i.e. if light intensity is greater than 800 then the alert should be provided to us.

4.5 Conclusion

As we can see that if light intensity in the chamber increases the IoT node's reading of light intensity increases. This means that the chamber is has been opened by some unauthorized person. Furthermore, the node sends and email and a notification to the user about the event. So, we can conclude that our Smart theft monitoring system is implemented successfully.

CHAPTER 5: CONCLUSION

Urban IoTs, in fact, are designed to support the Smart City vision, which aims at exploiting the most advanced communication technologies to support added-value services for the administration of the city and for the citizens. In the coming years, maximum of our everyday objects will be connected to the internet. While the range of design options for IoT systems is rather wide, the set of open and standardized protocols is significantly smaller. In this paper, we analyzed and implemented the structure of urban IoTs by using the tools provided by Memsic Inc. The discussed technologies are close to being standardized, and industry players are already active in the production of devices that take advantage of these technologies to enable the applications of interest. Parts of Urban Smart City are implemented in the form of Smart Environment Monitoring System which monitors the surrounding environmental factors like temperature, humidity, moisture etc. Another part of urban smart city is smart theft monitoring system that enables a user to detect that there is a breach in the area where nodes are applied. These systems work perfectly and provide us with notification features like pop up message and email alert.

References

- [1] Internet of Things for Smart Cities Andrea Zanella, Senior Member, IEEE, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, Senior Member, IEEE, and Michele Zorzi, Fellow, IEEE <https://ieeexplore.ieee.org/document/6740844/>
- [2] Build a Cloud based temperature Monitoring system IOT
<https://www.pantechsolutions.net/fpga-projects/build-a-cloud-based-temperature-monitoring-system-iot-using-spartan3an-starter-kit>
- [3] SMTP, and its working: <https://tools.ietf.org/html/rfc821>
- [4] Working of Moteconfig, MDA100, MIB520 <http://www.memsic.com/userfiles/files/User-Manuals/moteview-users-manual.pdf>
- [5] Xmesh wireless network http://www.memsic.com/userfiles/files/User-Manuals/xmesh-user-manual-7430-0108-02_a-t.pdf