DESIGN OF WIRELESS SENSOR NETWORK BASED SMART UNIVERSITY

A Dissertation submitted in partial fulfillment of the requirements

For the award of the Degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

MOHD. EHTHESHAMUDDIN

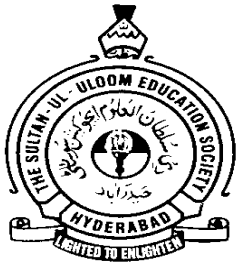
Roll No: 1604-13-735-046

B. SHAFAT MOHAMMED KHAN

Roll No: 1604-13-735-045

SYED ABDUL WAHAB

Roll No: 1604-13-735-055



MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

BANJARA HILLS, HYDERABAD-500 034

(AFFILIATED TO OSMANIA UNIVERSITY)

2017

DESIGN OF WIRELESS SENSOR NETWORK BASED SMART UNIVERSITY

A Dissertation submitted in partial fulfillment of the requirements

For the award of the Degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

MOHD. EHTHESHAMUDDIN

Roll No: 1604-13-735-046

B. SHAFAT MOHAMMED KHAN

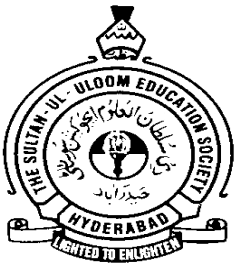
Roll No: 1604-13-735-045

SYED ABDUL WAHAB

Roll No: 1604-13-735-055

Under the Supervision of
Ms. SALMA FAUZIA
Assistant Professor
ECE Department

Under the Co-Supervision of
Ms. MALIHA NAAZ
Assistant Professor
ECE Department



MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

BANJARA HILLS, HYDERABAD-500 034

(AFFILIATED TO OSMANIA UNIVERSITY)

2017

MUFFAKHAM JAH
COLLEGE OF ENGINEERING & TECHNOLOGY
(Est. by Sultan-Ul-Uloom Education Society in 1980)
(Affiliated to Osmania University, Hyderabad)
Approved by the AICTE & Accredited by NBA

CERTIFICATE

This is to certify that the dissertation titled '**Design of Wireless Sensor Network Based Smart University'** has been submitted by **B. Shafat Mohammed Khan** bearing Roll No: 1604-13-735-045, **Mohd Ehtheshamuddin** bearing Roll No: 1604-13-735-046, and **Syed Abdul Wahab** bearing Roll No: 1604-13-735-055, in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Engineering,** is a bona fide record of work carried out by them under my supervision during the year 2016-2017. The results embodied in this report have not been submitted to any University or Institute for the award of any Degree or Diploma.

SALMA FAUZIA
Project Supervisor
Asst. Prof, ECED
MJCET, Hyderabad

MALIHA NAAZ
Project Co-Supervisor
Asst. Prof, ECED
MJCET, Hyderabad

Dr. Mohammed Arifuddin Sohel
Project coordinator
Prof, ECED
MJCET, Hyderabad

Dr. Kaleem Fatima
Head of Department
Prof, ECED
MJCET, Hyderabad

External Examiner

8-2-249, Mount Pleasant, Road No.3, Banjara Hills, Hyderabad – 500 034
Phone: 040-23350523, 23352084, Fax: 040-2335 3428
Website: www.mjcollege.ac.in, e-mail: principal@mjcollege.ac.in

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this dissertation entitled **'Design of Wireless Sensor Network Based Smart University'**, which has been submitted in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Engineering**, in the Department of Electronics and Communication Engineering, **Muffakham Jah College of Engineering and Technology**, Hyderabad, is an authentic record of my own work carried out from August 2016 to April 2017 under the supervision of Ms. Salma Fauzia Asst. Prof, ECED, MJCET, along with the co- supervision of Ms. Maliha Naaz Asst. Prof, ECED, MJCET.

I have not submitted the matter embodied in this report for the award of any other degree or diploma.

Date:

Place: Hyderabad

B. Shafat Mohammed Khan

Mohammed Ehtheshamuddin

Syed Abdul Wahab

ACKNOWLEDGEMENTS

As we come close to finishing our under graduate studies, which has been a great learning experience, we would like to take this as an opportunity to thank a lot of people who have helped us throughout our journey.

We are very much thankful to **Dr. Kaleem Fatima,** Head of the Department, Electronics and Communication Engineering, for encouragement and providing of lab facilities for our project work and guidelines on documentation of thesis.

We would first like to express our sincere gratitude to our Project Guide **Ms.Salma Fauzia** Asst. Professor, Department of Electronics and Communication Engineering, who has been a great source of inspiration and support throughout our studies. The endless doubts from our side never bothered her as we were patiently answered all our questions. The long discussions regarding our project have helped us understand different aspects of communications and sensor nodes and we are greatly indebted to her for all her guidance during our studies. We would always be grateful to her for her co-operation in completion of this project. As it was her unflinching support, guidance and encouragement that drove us towards the successful completion of our project. Without her guidance and endless support in giving feedback, this project along with the thesis would have been impossible.

We would like to express our deepest appreciation to **Ms. Maliha Naaz** for her valuable support and also thank **Ms. Nazeer Unnisa** and **Ms. Sucharitha** for providing invaluable inputs. We also express our sincere and deep gratitude to whole heartedly thank our Project coordinator **Dr. Arifuddin Sohel,** Professor, Programme coordinator Department of ECE, for his strong support and encouragement throughout the project.

We also extend our deepest sense of gratitude and thanks towards our Advisor cum Director, **Prof. Basheer Ahmed and Principal, Dr. Krishnan** for their support and encouragement.

We also thank the entire non-teaching staff of the Department of Electronics and Communication Engineering for their help and support during the course of the project and throughout the engineering career.

Our special thanks also goes to our beloved family members; our kind-hearted parents and our sisters for their strong support and encouragement during our studies.

We would also hereby like to thank and acknowledge the support of our dear friend **Mr. Huzaifa Nayeem,** Department of Electronics and Communication Engineering, for his valuable suggestions during the course of our project, without which it would not have been possible to give our project the structure it gained.

Above all we thank Almighty for constantly motivating us with His love and giving us courage at each difficult stride to step forward with confidence and self –belief.

B. Shafat Mohammed Khan
Mohammed Ehtheshamuddin
Syed Abdul Wahab

Table of Contents

ABSTRACT

Attendance is one of the important factors that determine the students rendering. There is a necessity to build a smart system that decrease load in managing the attendance as it improves the performance of colleges, universities and any educational institute. The most common actions in educational organizations involve identification of student, maintenance of student attendance, security of attendance. In this thesis, the following concepts are considered and present how evolving technologies of radio frequency identification (RFID) and wireless sensor network (WSN) can be used for building a smart university have been discussed. Prototype is developed using various WSN like Passive Infra Red (PIR), ZigBee and Arduino based controller boards while considering main cases concerned in a smart university. The system includes smart student attendance, smart automation of taking the attendance in authenticated manner and storing it at the database server with complete possible authentication of the attendance as well as the database.

List of Figures

List of Tables

1.1 Introduction

This chapter deals with our motivation behind taking this project, a description of issues that were addressed by us in designing our project, the literature survey was carried out and various IEEE papers and journals were studied to get an idea about Design of Wireless Sensor Network Based Smart University. Designing smart home [1] [2] [3] is an evolving issue that enclose ubiquitous pervasive computing concept of Mark Weiser [4] that favors the seamless integration of technologies in human lives. The stimulus of this is to concentrate on the advantages of using RFID and WSN technology in development of smart university. The developed prototype shows evolving technologies of RFID and WSN can add in improving student's attendance method and power conservation. RFID technology provides the answer for person's identification and forms the basis of:

- *Student attendance record.*
- *Employee attendance record.*
- *Authentication of attendance.*
- *Automation of electrical appliances.*

This was done so that we could come up with effective RFID system detecting the tags and processing it with better performance. Then technical approach was taken up to proceed with project from the Design of Wireless Sensor Network Based Smart University.

1.2 Problem Statement

Attendance is an important factor that determines the students rendering. Necessity to build a system that decreases load in managing the attendance, improves the performance of colleges, universities and any educational institute. Hence, there is a necessity to build a smart system that

- *Decrease load in managing the attendance and*
- *Improves the performance of colleges, universities and any educational institute.*

The most common actions in educational organizations involve identification of student, maintenance of student attendance, security of attendance at the place where the attendance is taken.

Prototype is developed considering main cases concerned in a smart university. The system is interesting of smart student attendance, smart automation of attendance and also taking attendance in authenticated manner.

1.3 Literature Survey

S.NO	TITLE	AUTHOR	YEAR	TECHNOLOGY REFERRED
1.	Designing of an Smart University using RFID and WSN [1]	Mahmood K. Al Shimmary, Muna M. Al Nayar, Abbas R. Kubba	2015	RFID, WSN and ZigBee
2.	Implementation of PIC16F877A Based Intelligent Smart Home System [2]	M. Shanmugasundaram, G. Muthuselvi, S. Sundar	2013	RFID and ZigBee
3.	Implementation of WSN based Home/Office Automation (HOA) [3]	Usha Sharma, S.R.N. Reddy	2014	WSN and ZigBee
4.	RFID Based Smart Home Architecture for improving lives [5]	Xuemei Li, Gang Xu, Li Li	2008	RFID
5.	Design of Home Automation System based on ZigBee Wireless Sensor Network [6]	Li Yang, Maorong Ji, Zhenru Gao	2010	RFID, WSN and ZigBee

1.4 Aim of the Thesis

The aim of this thesis is to design an Attendance Monitoring system for continuous marking of attendance as well as keeping it authentic. Two motion sensors are used to sense the presence/entry of a person through the entrance, this sensed count from the motion sensors are constantly compared with the number of RFID Tags read. After authentication of the tags the data is transferred through the XBee module and the data is fed to the system at the central attendance server. The data is stored continuously and is updated at regular basis and is available to view through Infra-net service. This report

describes the design of different blocks that are needed for Design of Wireless Sensor Network Based Smart University.

1.5 Motivation of the Thesis

The motivation for this research springs from the common problem for the faculty members i.e. Attendance Monitoring. Generally the faculty members are distracted by frequent movement of the students during the class and also the delayed arrival of the students and whether to allow them or not is another issue. Our objective is to develop more accurate way of determining attendance and a proper automated solution that would enable the faculty to focus more on the quality of class and the attendance would be monitored continuously and in an authenticated way.

Our project has a completely different solution compared to the previous work. This project involves use of software databases along with the hardware databases, which has not been used in the previous works.

1.6 Technical Approach

The technical approach of our thesis is as follows:

During the course of literature survey it is found that some of the projects used the Arduino UNO board, MF-RC 522 RFID Reader, PIR and XBee and hence after referring various previous works the basic technology was analyzed. In the first attempt, MF-RC522 was interfaced with Arduino UNO, along with two PIR. Post this interface we interfaced Arduino UNO with XBee S2B Pro. And later on, all the hardware modules were clubbed and the desired code was generated and UNO was replaced with MEGA at the class room end due to lack of dynamic memory.

Referring to this output data, in the second attempt we created a software database along with the servlet application for the attendance monitoring purpose which has given us the satisfying results.

1.7 Organization of Thesis

The thesis is divided into chapters. An introduction to the thesis is given in Chapter 1. This has been followed by problem statement, literature survey, aim of the thesis, motivation for research and technical approach to the problems solved in the thesis.

In chapter 2, the brief review of Wireless Sensor Networks has been discussed; which includes block diagram of Wireless Sensor Networks, applications, organization of WSN, Structure of Wireless Sensor Nodes and Networks, which is followed by Issues, Challenges, Security of Wireless Sensor Networks. Various Sensor Network Scenarios are also discussed in this chapter.

In Chapter 3 the architectural details and design aspects of single components are described. It includes Block diagram and working and it has been discussed for each component.

In Chapter 4 the software's used have been discussed, which include the in detail requirements of the software along with the basic overview of the working of the software.

Chapter 5 gives details about exact working of all the hardware components giving the general structure followed by the exact working of the software programs forming the whole project. This has been well supported by relevant flowcharts and block diagrams.

Chapter 6 gives the program codes which have been used in the various IDE and using various languages.

Chapter 7 gives relevant simulation results for each model.

Finally, conclusive remarks and future scope are presented in chapter 8, which is followed by the References.

2.1 WSN ORGANIZATION

Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructure- less wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analyzed. A sink or base station acts like an interface between users and the network. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically a wireless sensor network contains hundreds of thousands of sensor nodes. The sensor nodes can communicate among themselves using radio signals. A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components. The individual nodes in a wireless sensor network (WSN) are inherently resource constrained: they have limited processing speed, storage capacity, and communication bandwidth. After the sensor nodes are deployed, they are responsible for self-organizing an appropriate network infrastructure often with multi-hop communication with them. Then the onboard sensors start collecting information of interest. Wireless sensor devices also respond to queries sent from a "control site" to perform specific instructions or provide sensing samples. The working mode of the sensor nodes may be either continuous or event driven. Global Positioning System (GPS) and local positioning algorithms can be used to obtain location and positioning information. Wireless sensor devices can be equipped with actuators to "act" upon certain conditions. These networks are sometimes more specifically referred as Wireless Sensor and Actuator Networks as described in (Akkaya et al., 2005).

Wireless sensor networks (WSNs) enable new applications and require non-conventional paradigms for protocol design due to several constraints. Owing to the requirement for low device complexity together with low energy consumption (i.e. long network lifetime), a proper balance between communication and signal/data processing capabilities must be found. This motivates a huge effort in research activities, standardization process, and industrial investments on this field since the last decade (Chiara et. al. 2009). At present time, most of the research on WSNs has concentrated on the design of energy- and computationally efficient algorithms and protocols, and the application domain has been restricted to simple data-oriented monitoring and reporting applications

(Labrador et. al. 2009). The authors in (Chen et al., 2011) propose a Cable Mode Transition (CMT) algorithm, which determines the minimal number of active sensors to maintain K-coverage of a terrain as well as K-connectivity of the network. Specifically, it allocates periods of inactivity for cable sensors without affecting the coverage and connectivity requirements of the network based only on local information. In (Cheng et al., 2011), a delay-aware data collection network structure for wireless sensor networks is proposed. The objective of the proposed network structure is to minimize delays in the data collection processes of wireless sensor networks which extends the lifetime of the network. In (Matin et al., 2011), the authors have considered relay nodes to mitigate the network geometric deficiencies and used Particle Swarm Optimization (PSO) based algorithms to locate the optimal sink location with respect to those relay nodes to overcome the lifetime challenge. Energy efficient communication has also been addressed in (Paul et al., 2011; Fabbri et al. 2009). In (Paul et al., 2011), the authors proposed a geometrical solution for locating the optimum sink placement for maximizing the network lifetime. Most of the time, the research on wireless sensor networks have considered homogeneous sensor nodes. But nowadays researchers have focused on heterogeneous sensor networks where the sensor nodes are unlike to each other in terms of their energy. In (Han et al., 2010), the authors addresses the problem of deploying relay nodes to provide fault tolerance with higher network connectivity in heterogeneous wireless sensor networks, where sensor nodes possess different transmission radii. New network architectures with heterogeneous devices and the recent advancement in this technology eliminate the current limitations and expand the spectrum of possible applications for WSNs considerably and all these are changing very rapidly.

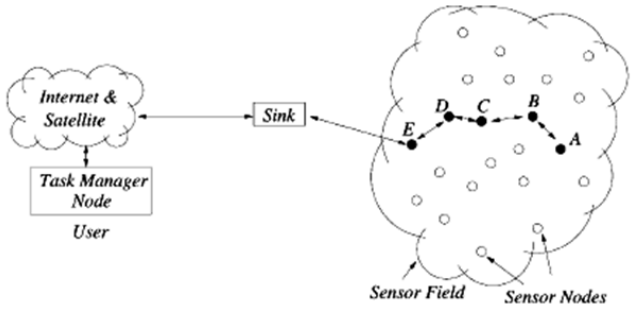


Fig 2.1-1. A typical Wireless Sensor Network

In WSN the number of sensor nodes can be in the order of hundreds or even thousands. In comparison with sensor networks, Ad Hoc networks will have less number

of nodes without any infrastructure. The differences between WSN and Ad hoc Networks are presented in the Table 2-1 [7].

Now- a- days wireless network is the most popular services utilized in industrial and commercial applications, because of its technical advancement in processor, communication, and usage of low power embedded computing devices. Sensor nodes are used to monitor environmental conditions like temperature, pressure, humidity, sound, vibration, position etc. In many real time applications the sensor nodes are performing different tasks like neighbor node discovery, smart sensing, data storage and processing, data aggregation, target tracking, control and monitoring, node localization, synchronization and efficient routing between nodes and base station [8].

Parameters	**Wireless Sensor Networks**	**Ad Hoc Networks**
Number of sensor nodes	Large	Medium
Deployment	Densely deployed	Scattered
Failure rate	Prone to failures	Very rare
Topology	Changes very frequently	Very rare
Communication paradigm	Broadcast communication	Point-to-Point communications
Battery	Not replaceable / Not rechargeable	Replaceable
Identifiers	No unique identifiers	Unique identifiers
Centric	Data centric	Address centric
Fusion / aggregation	Possible	Not suitable
Computational capacities, and memory	Limited	Not limited
Data rate	Low	High
Redundancy	High	Low

Table 2-1 Wireless Sensor Networks v/s Ad hoc Networks

Wireless sensor nodes are equipped with sensing unit, a processing unit, communication unit and power unit [9]. Each and every node is capable to perform data gathering, sensing, processing and communicating with other nodes. The sensing unit senses the environment, the processing unit computes the confined permutations of the sensed data, and the communication unit performs exchange of processed information among neighboring sensor nodes. The basic building block of a sensor node is shown in Fig 2-2.

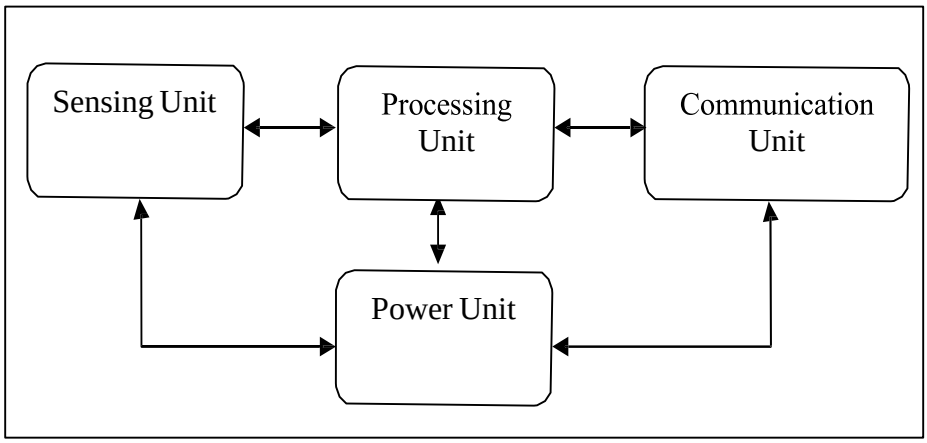


Fig 2.1-2 Basic Building Blocks of Sensor Node

The sensing unit of sensor nodes integrates different types of sensors like thermal sensors, magnetic sensors, vibration sensors, chemical sensors, bio sensors, and light sensors. The measured parameters from the external environment by sensing unit of sensor node are fed into the processing unit. The analog signal generated by the sensors are digitized by using Analog to Digital converter (ADC) and sent to controller for further processing.

The processing unit is the important core unit of the sensor node. The processor executes different tasks and controls the functionality of other components. The required services for the processing unit are pre-programmed and loaded into the processor of sensor nodes. The energy utilization rate of the processor varies depending upon the functionality of the nodes. The variation in the performance of the processor is identified by the evaluating factors like processing speed, data rate, memory and peripherals supported by the processors. Mostly ATMEGA 16, ATMEGA 128L, MSP 430 controllers [10] are used in commercial motes. The

computations are performed in the processing unit and the acquired result is transmitted to the base station through the communication unit.

In communication unit, a common transceiver act as a communication unit and it is mainly used to transmit and receive the information among the nodes and base station and vice versa. There are four states in the communication unit: transmit, receive, idle and sleep.

The major characteristics of the sensor node used to evaluate the performance of WSN are [11]

1. **Fault tolerance:** Each node in the network is prone to unanticipated failure. Fault tolerance is the capability to maintain sensor network functionalities without any break due to sensor node failures
2. **Mobility of nodes:** In order to increase the communication efficiency, the nodes can move anywhere within the sensor field based on the type of applications.
3. **Dynamic network topology:** Connection between sensor nodes follows some standard topology. The WSN should have the capability to work in the dynamic topology.
4. **Communication failures:** If any node in the WSN fails to exchange data with other nodes, it should be informed without delay to the base station or gateway node.
5. **Heterogeneity of nodes:** The sensor nodes deployed in the WSN may be of various types and need to work in a cooperative fashion.
6. **Scalability:** The number of sensor nodes in a sensor network can be in the order of hundreds or even thousands. Hence, WSN designed for sensor networks is supposed to be highly scalable.
7. **Independency:** The WSN should have the capability to work without any central control point.
8. **Programmability:** The option for reprogramming or reconfiguring should be available for the WSN to become adaptive for any dynamic changes in the network.
9. **Utilization of sensors:** The sensors should be utilized in a way that produces the maximum performance with less energy.
10. **Impracticality of public key cryptosystems:** The limited computation and power resources of sensor nodes often make it undesirable to use public key algorithms.
11. **Lack of a prior knowledge of post-deployment configuration:** If a sensor network

is deployed via random distribution, the protocols will not be aware of the communication status between each nodes after deployment.

The following metrics are used to evaluate the performance of a WSN [12]: network coverage, node coverage, efficiency in terms of system lifetime, effortless deployment, data accuracy, system response time, fault tolerance, scalability, network throughput, sample rate, security, the cost of the network and network architecture used. The individual sensor node in the WSN is evaluated using flexibility, robustness, computation, communication, security, synchronization, node size and cost.

The components of WSN system are sensor node, relay node, actor node, cluster head, gateway and base station which are explained below [13].

Sensor node: Capable of executing data processing, data gathering and communicating with additional associated nodes in the network. A distinctive sensor node capability is about 4-8 MHz, having 4 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

Relay node: It is a midway node used to communicate with the adjacent node. It is used to enhance the network reliability. A rely node is a special type of field device that does not have process sensor or control equipment and as such does not interface with the process itself. A distinctive rely node processor speed is about 8 MHz, having 8 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency

Actor node: It is a high end node used to perform and construct a decision depending upon the application requirements. Typically these nodes are resource rich devices which are outfitted with high quality processing capabilities, greater transmission powers and greater battery life. A distinctive actor node processor capability is about 8 MHz, having 16 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency [11].

Cluster head: It is a high bandwidth sensing node used to perform data fusion and data aggregation functions in WSN. Based on the system requirements and applications, there will be more than one cluster head inside the cluster. A distinctive cluster head processor is about 4-8 MHz, having 512 KB of RAM, 4 MB flash and preferably 2.4 GHz of radio frequency [11]. This node assumed to be highly reliable, secure and is trusted by all the nodes in the sensor network.

Gateway: Gateway is an interface between sensor networks and outside networks. Compared with the sensor node and cluster head the gateway node is most powerful in terms of program and data memory, the processor used, transceiver range and the possibility of expansion through external memory. A distinctive gateway processor speed is about 16 MHz, having 512 KB of RAM, 32 MB flash and preferably 2.4 GHz of radio frequency.

Base station: It is an extraordinary type of nodes having high computational energy and processing capability.

Attractive functionality of sensor nodes in a WSN includes effortlessness installation, fault indication, energy level diagnosis, highly reliability, easy coordination with other nodes in the network, control protocols and simple network interfaces with other smart devices. In WSN, based on the sensing range and environment, the sensor nodes are classified into four groups, namely specialized sensing node, generic sensing node, high bandwidth sensing node and gateway node. The radio bandwidth for the sensor nodes are <50 Kbps, <100 Kbps, ≈500 Kbps and >500 Kbps respectively. On board processing, computational rate and communication ranges differ from node to node in WSN. Particularly for some dedicated application sensor nodes with different capabilities are used. For example, smart specialized sensing nodes are preferred for special purpose devices, intelligent generic sensing node preferred for generic functions. For interconnectivity functions high end smart bandwidth sensing node and gateway nodes are preferred.

Sensor networks are clustered with gateway, relay node, actor node and cluster head, and every other node within the communication range. Cluster is a collection of group of sensor nodes in that particular sensor field. There may be more than one cluster in WSN. Based on the parameters like computation rate, processing speed, storage, and communication range, sensor nodes are identified and selected for WSN formation [7]. Based on the node properties the sensor networks are classified into two types, homogenous sensor networks and heterogeneous sensor networks. In homogenous sensor networks, all sensor nodes have the same property in terms of computation, communication, memory, energy level and reliability. In heterogeneous sensor networks, the nodes are of different capabilities in terms of computation, communication, memory, energy level and reliability. If all the sensor nodes within the cluster are having the same properties (homogenous) it is referred as distributed

WSN (DWSN). Otherwise if the sensor nodes have different properties (heterogeneous) it is called Hierarchical WSN (HWSN). The distributed & hierarchical WSN is shown in Fig 2-3.

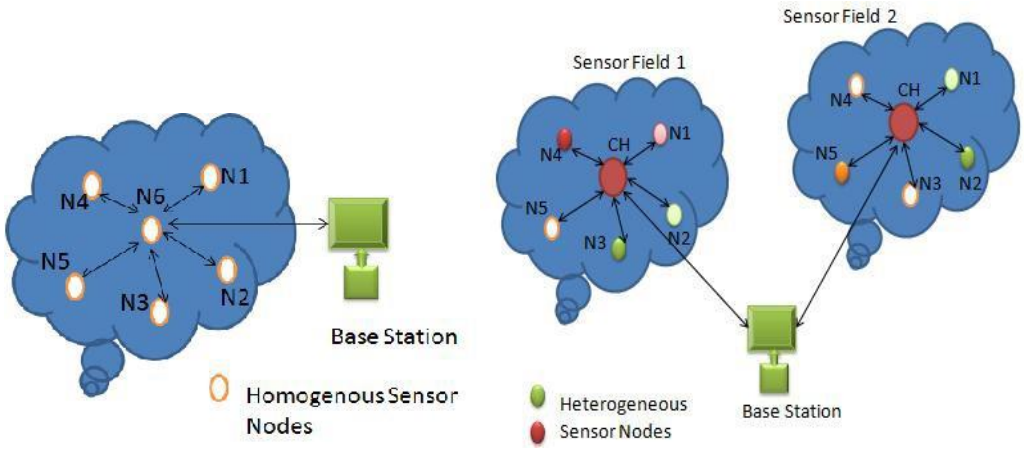


Fig 2.1-3 Distributed and Hierarchical WSN

Sensor nodes in an open environment regularly sense the physical and environmental changes and transmit the information to the centralized server called a gateway. The computational rate and interaction of sensor nodes with the physical environment is different for different nodes in the network. In real time, sensor nodes are more constrained in its computational energy and storage resources.

The sensor nodes are intelligent to observe an extensive diversity of ambient circumstances that includes flow, temperature, pressure, humidity, moisture, noise levels, mechanical stress, speed, etc. Many novel applications are being developed due to the new concept of micro sensing and wireless networking for these smart sensing devices. Some of the possible assorted applications [14] of WSN 's temperature control, inventory management, physiological monitoring, habitat monitoring, precision agriculture, forest fire detection, nuclear, chemical, and biological attack detection, military, transportation, disaster relief, and environmental monitoring.

Any WSN can be configured [14] as a five layered architecture as explained below

- The physical layer is responsible for frequency selection, modulation and data

encryption.

- The data link layer functions as a pathway for multiplexing of data streams, data frame detection, Medium Access control (MAC) and error control.
- The network layer is used to route the data supplied by the transport layer using special multi-hop wireless routing protocols between sensor nodes and sink nodes.
- The transport layer maintains the flow of data if the application layer requires it.
- The application layer makes the hardware and software of the lower layers transparent to the end user

2.2 Sensor node hardware overview

When choosing the hardware components for a wireless sensor node, evidently the application's requirements play a decisive factor with regard mostly to size, costs, and energy consumption of the nodes – communication and computation facilities as such are often considered to be of acceptable quality, but the trade-offs between features and costs is crucial. In some extreme cases, an entire sensor node should be smaller than 1 cc, weigh (considerably) less than 100 g, be substantially cheaper than US$1, and dissipate less than 100 micro W. In even more extreme visions, the nodes are sometimes claimed to have to be reduced to the size of grains of dust. In more realistic applications, the mere size of a node is not so important; rather, convenience, simple power supply, and cost are more important [15].

These diversities notwithstanding, a certain common trend is observable in the literature when looking at typical hardware platforms for wireless sensor nodes. While there is certainly not a single standard available, nor would such a standard necessarily be able to support all application types, this section will survey these typical sensor node architectures. In addition, there are a number of research projects that focus on shrinking any of the components in size, energy consumption, or costs, based on the fact that custom off-the-shelf components do currently not live up to some of the more stringent application requirements. But as this book focuses on the networking aspects of WSNs, these efforts are not discussed here.

A basic sensor node comprises of five main components (Fig 2.2):

Controller: A controller to process all the relevant data, capable of executing arbitrary code.

Memory: Some memory to store programs and intermediate data; usually, different types of memory are used for programs and data.

Sensors and actuators: The actual interface to the physical world: devices that can observe or control physical parameters of the environment.

Communication: Turning nodes into a network requires a device for sending and receiving information over a wireless channel

Power supply: As usually no tethered power supply is available, some form of batteries are necessary to provide energy. Sometimes, some form of recharging by obtaining energy from the environment is available as well (e.g. solar cells).

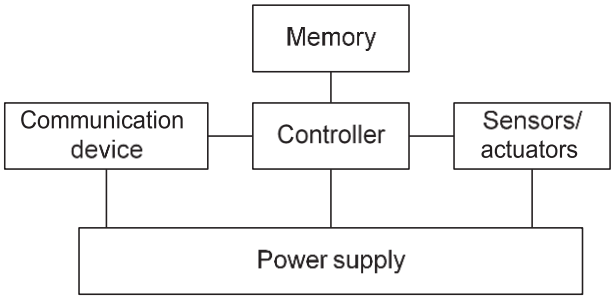


Fig 2.2 Overview of Main Sensor Node Hardware Components

Each of these components has to operate balancing the trade-off between as small an energy consumption as possible on the one hand and the need to fulfill their tasks on the other hand. For example, both the communication device and the controller should be turned off as long as possible. To wake up again, the controller could, for example, use a preprogrammed timer to be reactivated after some time. Alternatively, the sensors could be programmed to raise an interrupt if a given event occurs – say, a temperature value exceeds a given threshold or the communication device detects an incoming transmission.

Supporting such alert functions requires appropriate interconnection between individual components. Moreover, both control and data information have to be exchanged along these interconnections. This interconnection can be very simple – for example, a sensor could simply report an analog value to the controller – or it could be endowed with some intelligence of its own, preprocessing sensor data and only waking up the main controller if an actual event has been detected – for example, detecting a threshold crossing for a simple temperature sensor. Such preprocessing can be highly customized to the specific sensor yet remain simple enough to run continuously, resulting in improved energy efficiency [16].

2.3 Structure of a Wireless Sensor Network

Structure of a Wireless Sensor Network includes different topologies for radio communications networks. A short discussion of the network topologies that apply to wireless sensor networks are outlined below:

Star network (single point-to-multipoint) (Wilson, 2005)

A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks includes simplicity, ability to keep the remote node's power consumption to a minimum. It also allows low latency communications between the remote node and the base station. The disadvantage of such a network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

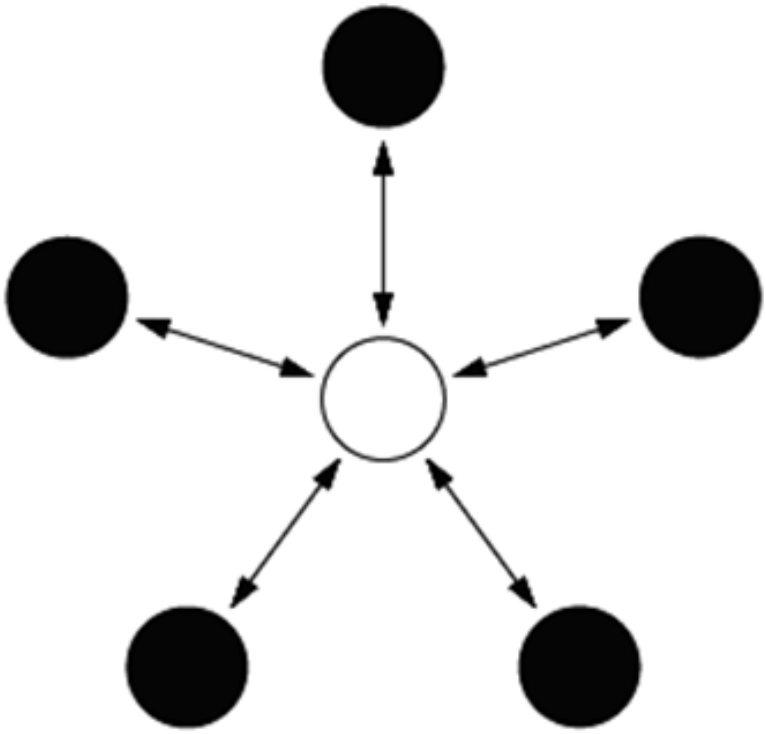


Fig 2.3-1 A Star network topology

Mesh network (Wilson, 2005)

A mesh network allows transmitting data to one node to other node in the network that is within its radio transmission range. This allows for what is known as multi-hop communications, that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes; it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multi-hop communications are generally higher than for the nodes that don't have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

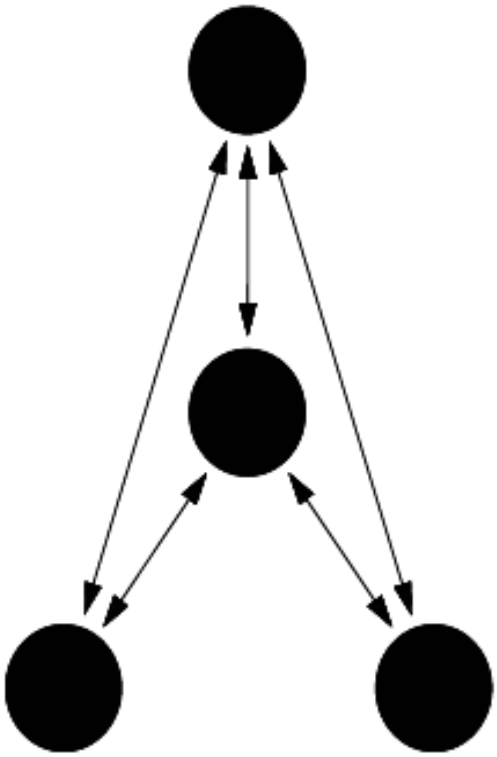


Fig 2.3-2 A Mesh network topology

Hybrid star – Mesh network (Wilson, 2005)

A hybrid between the star and mesh network provides a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the sensor nodes with lowest power are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multi-hop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multi-hop capability are higher power, and if possible, are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

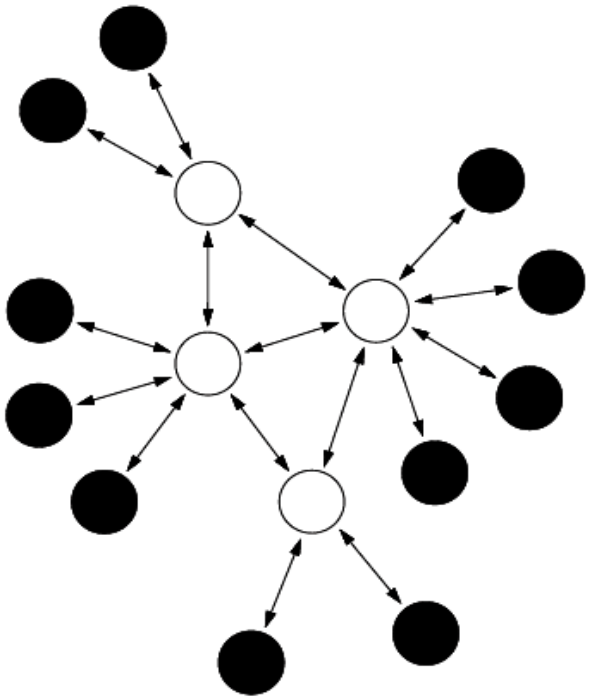


Fig 2.3-3 A Hybrid Star – Mesh network topology

2.4 Structure of a Wireless Sensor Node

A sensor node is made up of four basic components such as sensing unit, processing unit, transceiver unit and a power unit which is shown in Fig. 2.4. It also has application dependent additional components such as a location finding system, a power generator and a mobilizer. Sensing units are usually composed of two subunits: sensors and analogue to digital converters (ADCs) (Akyildiz et al., 2002). The analogue signals produced by the sensors are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit is generally associated with a small storage unit and it can manage the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units can be supported by a power scavenging unit such as solar cells. The other subunits, of the node are application dependent.

Modular design approach provides a flexible and versatile platform to address the needs of a wide variety of applications. For example, depending on the sensors to be deployed, the signal conditioning block can be re-programmed or replaced.

This allows for a wide variety of different sensors to be used with the wireless sensing node. Similarly, the radio link may be swapped out as required for a given applications' wireless range requirement and the need for bidirectional communications.

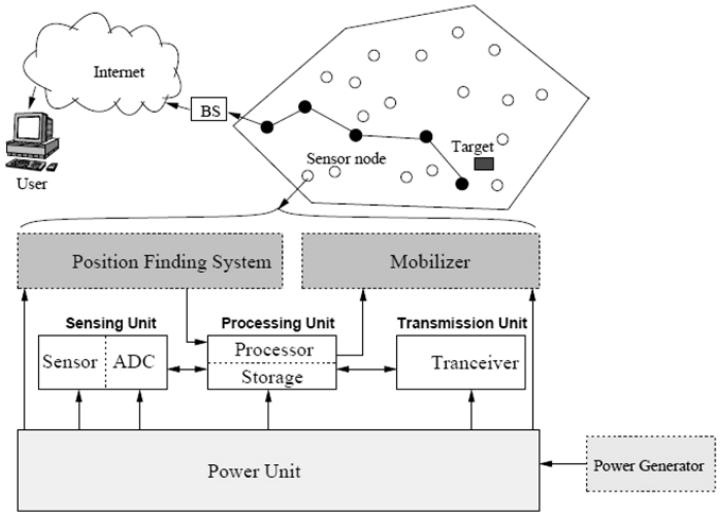


Fig 2.4: The components of a sensor node

The microprocessor has a number of functions including:

- Managing data collection from the sensors
- Performing power management functions
- Interfacing the sensor data to the physical radio layer
- Managing the radio network protocol

A key aspect of any wireless sensing node is to minimize the power consumed by the system. Usually, the radio subsystem requires the largest amount of power. Therefore, data is sent over the radio network only when it is required. An algorithm is to be loaded into the node to determine when to send data based on the sensed event. Furthermore, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microprocessor to judiciously control power to the radio, sensor, and sensor signal conditioner (Akyildiz et al., 2002).

2.5 Communication structure of a wireless sensor network

The sensor nodes are usually scattered in a sensor field. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructure-less architecture through the sink. The sink may communicate with the task manager node via Internet or Satellite.

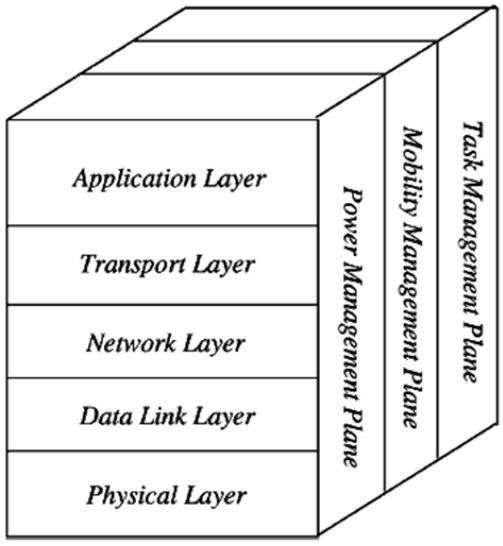


Fig 2.5. Wireless Sensor Network protocol stack

The protocol stack used by the sink and the sensor nodes is given in Fig 2.5. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium and promotes cooperative efforts of sensor nodes. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane (Akyildiz et al., 2002). Different types of application software can be built and used on the application layer depending on the sensing tasks. This layer makes hardware and software of the lowest layer transparent to the end-user. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer, specific multi-hop wireless routing protocols

between sensor nodes and sink. The data link layer is responsible for multiplexing of data streams, frame detection, Media Access Control (MAC) and error control. Since the environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbor broadcast. The physical layer addresses the needs of a simple but robust modulation, frequency selection, data encryption, transmission and receiving techniques.

In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower the overall energy consumption.

2.6 Issues and Challenges in designing Wireless Sensor Network

- Sensor networks do not fit into any regular topology, because while deploying the sensor nodes they are scattered [7] [12] [17]
- Very limited resources
  - Limited memory,
  - Limited computation
  - Limited power
- It comes under fewer infrastructures and also maintenance is very difficult.
- Conflicts and latency
- Sensor node relies only on battery and it cannot be recharged or replaced. Hardware design for sensor node should also be considered.
- Unattended operations
  - Exposure to physical attack
  - Remotely managed
  - No central control point
- Achieving synchronization between nodes is also another issue.
- Node failure, topology changes and adding of nodes and deletion of nodes is another challenging issue.

- Because of its transmission nature and hostile environment, security is a challenging issue. Based on the applications, sensor node has to be chosen

2.7 Security in Wireless Sensor Network

Sensor networks presents exclusive challenges, so conventional security techniques used in traditional networks cannot be applied directly for WSN. The sensor devices are inadequate in their energy, computation, and communication capabilities. When sensor networks are deployed in a hostile environment, security becomes extremely important, as they are prone to different types of malicious attacks. For example, an adversary can easily listen to the traffic, impersonate one of the network nodes, or intentionally provide misleading information to other nodes. WSN works together closely with their corporal environments, posing new security troubles [18]. As a result, existing security mechanisms are insufficient, and novel ideas are needed.

- Sensor nodes are randomly deployed in an open and unattended environment, so security is critical for such networks
- WSN uses wireless communication, which is predominantly easy to eavesdrop on.
- An attacker can easily inject malicious node in the network.
- WSN involves a large number of nodes in the network. Enforcing security in all the levels is important and also too complex.
- Sensor nodes are resource constraints in terms of memory, energy, transmission range, processing power. Hence asymmetric cryptography is too expensive and symmetric cryptography is used as alternatives.
- Cost of implementing tamper resistant software is very high.

WSN's general security goals [19] are confidentiality, integrity, authentication, availability, survivability, efficiency, freshness and scalability as described in Table 2.7 WSN is susceptible to many attacks because of its transmission nature, resource restriction on sensor nodes and deployment in uncontrolled environments. To ensure the security services in WSN many crypto mechanisms like symmetric and asymmetric methods are proposed. To achieve security in wireless sensor networks, it is important to be able to encrypt and authenticate messages sent between sensor nodes.

Confidentiality	Keeping node information secret from others but authorized users see it.
Integrity	Possible for the receiver node of a message to confirm that it has not been customized in transit.
Device authentication	Justification of the identity of the device.
Message authentication	Justification the source of information
Validation	To provide correctness of authorization to use or manipulate resources.
Access control	Restricting access to resources.
Revocation	Renunciation of certification or authorization.
Survivability	The lifetime of the sensor node must be extended even the node is compromised.
Nonrepudiation	Preventing the denial of a previous commitment.
Availability	High availability systems in sensor node is aim to remain available at all times preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks.
Data freshness	Data freshness objective ensures that messages are fresh, meaning that they are in proper order and have not been reused.

Table 2.7 Security Services

While designing the key management schemes, the important metrics [20] to be evaluated are

1. **Local / global connectivity:** Each node communicates with every other node in the sensor field region.

2. Resilience: Whenever a sensor node is compromised, the key management scheme assures in securing the remaining communication link against node capture.

3. Scalability: Capability to support when large numbers of nodes are added to the

sensor network.

4. **Efficiency:** In terms of storage, communication and computation

2.8 Sensor Network Scenarios

Types of sources and sinks

There are several typical interaction patterns found in WSNs – event detection, periodic measurements, function approximation and edge detection, or tracking – it has also already briefly touched upon the definition of "sources" and "sinks". A source is any entity in the network that can provide information, that is, typically a sensor node; it could also be an actuator node that provides feedback about an operation.

A sink, on the other hand, is the entity where information is required. There are essentially three options for a sink: it could belong to the sensor network as such and be just another sensor/actuator node or it could be an entity outside this network. For this second case, the sink could be an actual device, for example, a handheld or PDA used to interact with the sensor network; it could also be merely a gateway to another larger network such as the Internet, where the actual request for the information comes from some node "far away" and only indirectly connected to such a sensor network. These main types of sinks are illustrated by Figure 2.8-1, showing sources and sinks in direct communication.

This distinction between various types of sinks is actually fairly irrelevant. It is important, whether sources or sinks move, but what they do with the information is not a primary concern of the networking architecture. There are some consequences if a sink is considered as a gateway node.

Single-hop versus multihop networks

From the basics of radio communication and the inherent power limitation of radio communication follows a limitation on the feasible distance between a sender and a receiver. Because of this limited distance, the simple, direct communication between source and sink is not always possible, specifically in WSNs, which are intended to cover a lot of ground (e.g. in environmental or agriculture applications) or that operate in difficult radio environments with strong attenuation (e.g. in buildings).

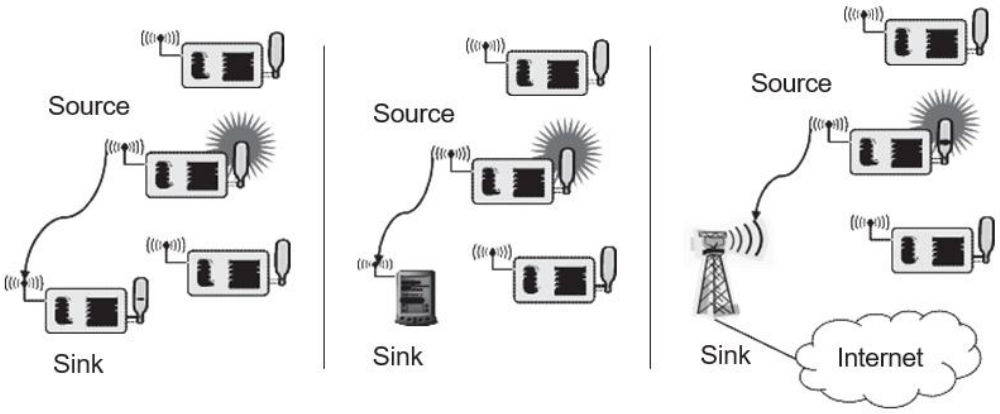


Fig 2.8-1: Three types of sinks in a very simple, single-hop sensor network

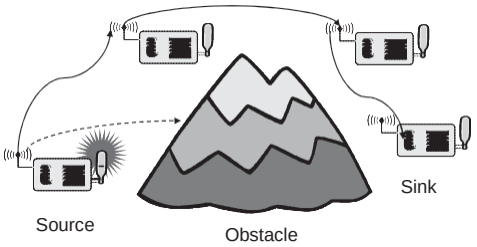


Figure 2.8-2: Multihop networks: As direct communication is impossible because of distance and/or obstacles, multihop communication can circumvent the problem

To overcome such limited distances, an obvious way out is to use relay stations, with the data packets taking multi hops from the source to the sink. This concept of multihop networks (illustrated in Figure 2.8-2) is particularly attractive for WSNs as the sensor nodes themselves can act as such relay nodes, foregoing the need for additional equipment. Depending on the particular application, the likelihood of having an intermediate sensor node at the right place can actually be quite high – for example, when a given area has to be uniformly equipped with sensor nodes anyway – but nevertheless, there is not always a guarantee that such multihop routes from source to sink exist, nor that such a route is particularly short.

While multihopping is an evident and working solution to overcome problems with large distances or obstacles, it has also been claimed to improve the energy efficiency of communication. The intuition behind this claim is that, as attenuation of

radio signals is at least quadratic in most environments (and usually larger), it consumes less energy to use relays instead of direct communication.

2.9 Applications

Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. WSNs have been successfully applied in various application domains such as:

Military applications: Wireless sensor networks be likely an integral part of military command, control, communications, computing, intelligence, battlefield surveillance, reconnaissance and targeting systems.

Area monitoring: In area monitoring, the sensor nodes are deployed over a region where some phenomenon is to be monitored. When the sensors detect the event being monitored the event is reported to one of the base stations, which then takes appropriate action.

Transportation: Real-time traffic information is being collected by WSNs to later feed transportation models and alert drivers of congestion and traffic problems.

Health applications: Some of the health applications for sensor networks are supporting interfaces for the disabled, integrated patient monitoring, diagnostics, and drug administration in hospitals, tele-monitoring of human physiological data, and tracking & monitoring doctors or patients inside a hospital.

Environmental sensing: The term Environmental Sensor Networks has developed to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, glaciers, forests etc. Some other major areas are listed below:

- Air pollution monitoring
- Forest fires detection
- Greenhouse monitoring
- Landslide detection

Structural monitoring: Wireless sensors can be utilized to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc. enabling Engineering practices to monitor assets remotely without the need for costly site visits.

Industrial monitoring: Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionalities. In wired systems, the installation of enough sensors is often limited by the cost of wiring.

Agricultural sector: Using a wireless network frees the farmer from the maintenance of wiring in a difficult environment. Irrigation automation enables more efficient water use and reduces waste.

2.10 Conclusion

This chapter can be summarized in the following manner:

Each sensor node consists of five main components, which are a microcontroller unit, a transceiver unit, a memory unit, a power unit and a sensor unit. Each one of these components is determinant in designing a WSN for deployment.

- The **Microcontroller unit** is in charge of the different tasks, data processing and the control of the other components in the node.
- Through the **Transceiver unit** a sensor node performs its communication with other nodes and other parts of the WSN. It is the most power consuming unit.
- The **Memory unit** is for temporal storage of the sensed data and can be RAM, ROM and their other memory types (SDRAM, SRAM, EPROM etc.) flash or even external storage devices such as USB.
- The **Power unit,** which is one of the critical components, is for node energy supply. Power can be stored in batteries (most common) rechargeable or in capacitors. For extra power supply and recharge, there can be use of natural sources such as solar power in forms of photovoltaic panels and cells etc.
- Lastly the **Sensor unit,** which includes one or more different types of sensors for parameter measurements.

In this chapter it has been briefly discussed about the definition of Wireless Sensor Network, Organization of WSN. Further the detailed overview for Sensor Node Hardware is discussed. Then, structure of Wireless Sensor Nodes and Network, Communication structure of WSN was discussed, followed by the various sensor network scenarios and issues, challenges in design of WSN and security in WSN has been discussed. Finally, the applications of WSN were discussed.

3.1 ARDUINO

This chapter introduces the reader to Arduino, the electronics prototyping platform which has been used during the developments described in this thesis.

3.1.1 BASICS OF ARDUINO

Arduino, according to Massimo Banzi, one of its creators, is an open source physical computing platform based on a simple input/output (I/O) board and a development environment that implements the Processing language. Hence we need to refer such software, instead of using the misleading Open Source Software naming, with the term Libre Software or Free Software so that the reader can clearly understand that the importance is given to the freedoms, not only to the access of the source code. Given the considerations made for the software and porting them into the hardware world, I'd rather prefer using the term Libre Hardware rather than Open Source Hardware so that it's clear that we are more concerned about the freedoms given to the user by using Libre Hardware rather than the open access to the hardware designs. I'd also would like to note the fact that most of the success of Arduino is due to a thrilling community of developers, hackers, hobbyists which contribute code, documentations, guides on the arduino.cc and other websites. So, in my opinion, a better definition of Arduino would be: a libre hardware physical computing platform based on a simple input/output (I/O) board, a development environment that implements the Processing language and a community of users which share their efforts and knowledge in their Arduino based projects

3.1.2 NEED FOR ARDUINO?

There are many hardware prototyping platforms available but Arduino is a good choice as:

- It is a libre hardware and software project, so both software and hardware are extremely accessible and very flexible and they can easily be customized and extended.
- It is flexible, offers various digital and analog inputs, SPI, I2C, a serial interface and digital and PWM outputs
- It is easy to use, it connects to a computer via USB and communicates using the standard serial protocol, runs in standalone mode and as an interface connected to PC/Macintosh

- It is inexpensive, less than 30 euro per board and comes with free development environment
- It is backed up by a growing on-line community, lots of source code is already available and ready to be used.

3.1.3 STUFF POSSIBLE WITH ARDUINO

Arduino is a great tool for developing interactive objects, taking inputs from a variety of switches or sensors and controlling a variety of lights, motors and other outputs. Arduino projects can be stand-alone or they can be connected to a computer using USB. The Arduino will be seen by the computer as a standard serial interface. There are serial communication APIs on most programming languages so interfacing Arduino with a software program running on the computer is pretty straightforward.

3.1.4 ARDUINO HARDWARE

The Arduino board is a microcontroller board [21], which is a small circuit (the board) that contains a whole computer on a small chip (the microcontroller). There are different versions of the Arduino board: they are different in components, aim and size, etc. Some examples of Arduino boards are: Arduino Duemilanove/UNO, Arduino Mega, Arduino Nano, Arduino Mini. Arduino schematics are distribute using an open license so anyone is free to build his own Arduino compatible board. The Arduino name is a registered trademark so it's not possible to call a cloned board Arduino: that's why it's very common to reference on Arduino boards like Seeeduino, FreeDuino, Japanino, Zigduino, iDuino, etc.

Fig 3.1.4: Few Arduino boards - From top left to bottom right: Lylipad, Mini, Nano (two), Pro, Duemilanove, Mega.

3.1.5 ARDUINO SHIELDS

Arduino boards functionalities can be extended by using shields, ad hoc designed PCBs having the same pin layout of Arduino, which can be stacked above each other thereby adding few additional functionalities. Fig 3.1.5 shows a quite few examples of Arduino shield usage.

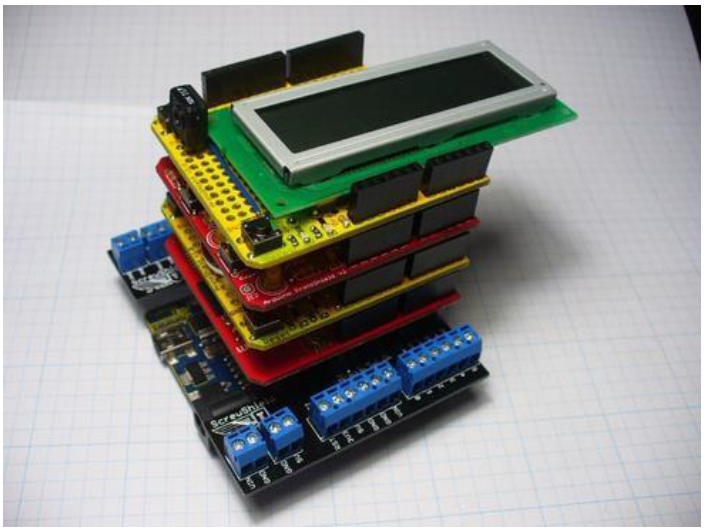


Fig 3.1.5: Arduino shielding - Arduino it's placed on the bottom and the different shields are stacked above of it. Picture by John Boxall CC-BY-NC-SA 3.0

There is a huge amount of shields available, each one of them especially designed for one application. Some are being developed by the Arduino team while most of them have been developed by third party companies or individuals. There are shields for Motor controlling, Ethernet communication, MP3 playing, Analog video output, LCD displays, etc. The idea is that using a shield is possible to add a specific feature to Arduino without the hassle of developing an ad hoc circuit or PCB trying to implement such feature. Moreover, some shields comes with easy to use libraries which allows fast and straightforward application development.

3.1.6 ARDUINO UNO

The UNO [22] is the best board to get started with electronics and coding, as it is the most robust board. The UNO is the most used and documented board of the whole Arduino. The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Fig 3.1.6: Arduino UNO

The ATmega328 on the Uno comes preprogrammed with a bootloader that allows to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated.

Power

The UNO board can be powered via the USB connection or with an external Power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. The board can operate on an external supply from 6 to 20 volts. If supplied

with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **Vin:** The input voltage to the Uno board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- **IOREF:** This pin on the Uno board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

3.1.7 ARDUINO MEGA

The Arduino Mega 2560 [23] is microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

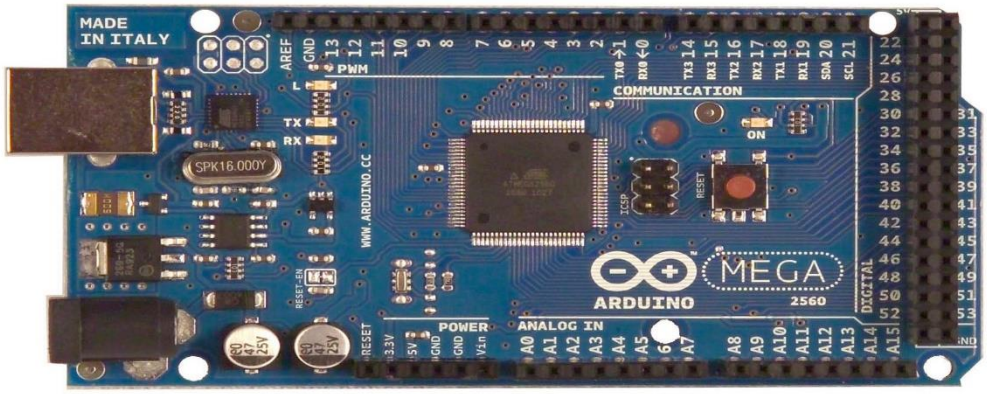


Fig 3.1.7 Arduino Mega

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). Voltage can be supplied through this pin or, if supplying voltage via the power jack, access it through this pin.

5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3V. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin

can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions.

Serial:

0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts:

2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM: 0 to 13. Provides 8-bit PWM output with the analogWrite() function.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library.

The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

I2C: 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library. Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference()

There are a couple of other pins on the board:

AREF: Reference voltage for the analog inputs. Used with analogReference().

Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

The ATmega2560 on the Arduino Mega comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

Even in Arduino Mega by, bypassing the bootloader programing of the microcontroller through the ICSP (In-Circuit Serial Programming) header is possible.

The ATmega8U2 firmware source code is available in the Arduino repository. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board and then resetting the 8U2. Atmel's FLIP software

(Windows) or the DFU programmer (Mac OS X and Linux) can be used to load a new firmware. Or ISP header can be used with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, it should be made sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable poly fuse that protects the computer's USB

ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

3.1.8 ARDUINO SOFTWARE

The other component of the Arduino platform is the Arduino IDE. This contains all the software which will run a computer in order to program and communicate with an Arduino board. The Arduino IDE contains an editor which we can use to write sketches (that's the name of Arduino programs) in a simple programming language modeled after the Processing language. Using the IDE, the program written is converted to a C program and then compiled using avr-gcc, a free libre and open source compiler based on the GNU C Compiler (gcc) especially designed for AVR microcontrollers. This process produces binary code which the microcontroller on the Arduino board will be able to understand and execute the binary code. It's then uploaded to the Arduino microcontroller through the USB connection. This is done using the program avrdude which implements the communication protocol used to store programs into the Arduino program memory

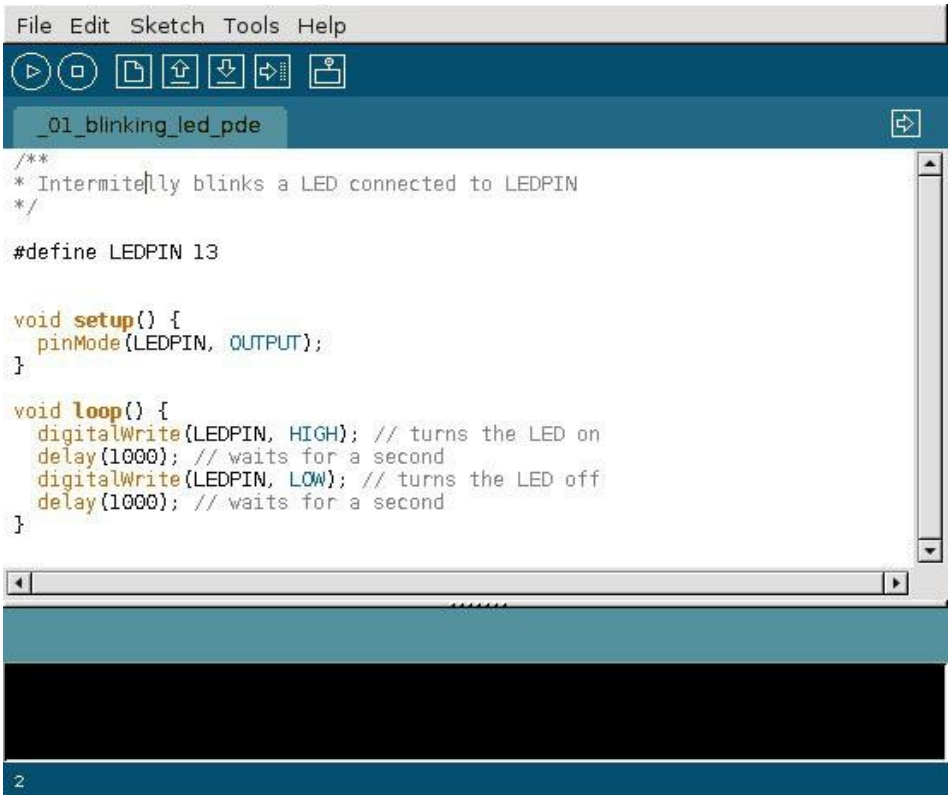


Fig 3.1.8: Arduino Programming IDE

3.2 RFID

3.2.1 Introduction to RFID?

Radio Frequency Identification (RFID) is a flexible, wireless, automatic identification technology that transmits the identity (in the form of a unique serial number) of an object or person wirelessly, using radio waves [24] [25]. It comes under the category of automatic identification technologies.

Auto-ID technologies include optical character readers, bar codes and some biometric technologies, such as retinal scans [24] [25]. These technologies are mainly used to reduce time and labor needed for manually data entry and to enhance data accuracy. Some auto-ID technologies, like bar code systems, often require a person to manually scan a label or tag to capture the data [24] [25]. While bar code tags and bar code systems are much less expensive than RFID at present, RFID provides many benefits than barcode system, which are listed below.

3.2.2 Technical characteristics of RFID

- Data read and write – RFID reader can read the data to the database without contact, and process multiple tags once, and write the logistic processing state into the tag for the logistic processing in the next stage [26].
- Miniaturized and diverse form – RFID will not be limited by the size or form when it reads data, so it needs not to use the paper with fixed size or print quality to fit for the precision. In addition, E-tag of RFID can be applied in different products by small size, so we can more flexibly control the production of the products, especially the application on the production line [26].
- Anti-pollution – RFID possesses strong anti-pollution nature for water, oil or drugs. And in the dark or polluted environment, RFID also can read data [26].
- Repetitive use – Because RFID is electric data which can be written repetitively, so the tag can be used repetitively [26].
- Penetrability – If RFID is covered by the paper, wood, plastics or non-metal or non-transparent materials, it can communicate through these materials except for the irons or other metals [26].
- Big memory capacity of data – The data capacity will be extended with the development of the memory scale, and the quality of the material carried by the goods is larger, the requirement of the capacity for the volume label also increase, and RFID will not be restrained [26].
- System safety – The system stores the data from the central computer to the work piece which will largely enhance the safety of the system [26].
- Data safety – The checkout method or the cycle redundancy checkout method will be used to ensure the data veracity stored in the radio frequency tag [26].

The RFID concept is not new but has been around for decades; in fact, it was introduced to the world for the first time during World War II by the British Air Force to distinguish Allied aircraft from enemy aircraft using radar (table 1 provides a brief overview of the history of RFID technology) [27]. Since then, this technology has been used for various applications. RFID technology has been used by thousands of companies

in many different ways for a decade or more to create value [25]. Here are some of the business applications where this technology is used

- Asset Tracking – It is one of the most common uses of RFID. RFID tags can be put on assets that are lost or stolen [25] [28].
- Supply Chain Management – It is used in closed loop supply chains or to automate parts of the supply chain within a company [25] [28].

Year	Event
1930 – 1940	American navy research laboratories developed a system known as IFF (Identify Friend or Foe).
1940 – 1950	The first application of RFID consisted of identifying Allied or enemy planes during WWII through the use of the IFF system.
1950 – 1960	IFF technology was used to develop the modern air traffic control system. First RFID applications in the military sector, in research laboratories and in major commercial enterprises.
1960-1970	Sensormatic and Checkpoint Systems introduced new applications for RFID, such as electronic article surveillance (EAS) equipment.
1970 – 1980	Technological advancements led to the creation of the passive tag, and the first initiatives for animal tracking and factory automation took place.
1980 – 1990	Many American and European companies started to manufacture RFID tags. First RFID application for automatic toll payment.
1990 – 2000	Standards for RFID equipment interoperability were developed.
2003	The Auto-ID Centre from MIT became EPCglobal, an organization whose objective is to promote the use and adoption of EPC technology.
2005	Wal-Mart launched an EPC pilot.

Table 3.2-1: RFID's History

(Source: AIM Publication (2001), Manish (2005), EPCglobalinc.org)

- Retailing – It is used by retailers to improve supply chain efficiency and making sure product is on the shelf when customers want to buy it [25] [28].
- Payment Systems – One of the most popular uses of RFID today is to pay for road tolls without stopping. It can also be used in a convenient way to pay for bus, subway and train ticket [25] [28].
- Security and Access Control – It can be used as an electronic key to control who has

access to office buildings or areas within office building [25] [28].

RFID technology is cheap and many new applications are being developed to solve common and unique business problems.

3.2.3 Architecture & Operation of a RFID System

A RFID system is composed of three basic components: a tag, a reader, and a host computer.

3.2.3.1 RFID tags

It contains tiny semiconductor chips and miniaturized antennas inside some form of packaging [29]. They can be uniquely identified by the reader/host pair and, when applied or fastened to an object or a person, that object or person can be tracked and identified wirelessly and on the move [29]. RFID tags come in many forms. For example, some look like paper labels and are applied to boxes and packaging; others are incorporated into the walls of injection moulded plastic containers; and still others are built into wristbands and worn by people [30] [31].

Types of RFID tags

i. **Active RFID Tags:** Include on-board power source (miniature batteries) that are used to power the tag, and can transmit signals autonomously
ii. **Passive RFID tags** don't include an on-board power source and have power beamed to them by the reader.
iii. **Battery Assisted Passive (BAP) or Semi-Passive RFID Tags:** Require an external source to wake up but have significant higher forward link capability providing greater range.

Smart Tags

i. **Read only tags:** Information is programmed onto chip during manufacturing, no overwriting, and information is constant, least expensive.
ii. **Write Once Read Many (WORM) tags:** Information added only once along with unique identifier but can be read many times.

iii. **Read-Write tags:** Open to data manipulation by user's system without restrictions. It contains a unique identifier but carries an updateable memory for that to be added. It is expensive also.

The following are the commonly used frequencies:

i. **Microwave** works on 2.45 GHz, it has good reader rate even faster than UHF tags. Although at this frequency the reading rate results are not the same on wet surfaces and near metals: The frequency produces better results in applications such as vehicle tracking (in and out with barriers), with approximately 1 meter of tags read range [32].
ii. **Ultra High Frequency** works within a range of 860-930 MHz, it can identify large numbers of tags at one time with quick multiple read rate at a given time. So, it has a considerable good reading speed. It has the same limitation as Microwave when is applied on wet surface and near metal. However, it is faster than high frequency data transfer with a reading range of 3 meters [32].
iii. **High Frequency** works on 13.56MHz and has less than one meter reading range but is inexpensive and useful for access control, items identifications on sales points etc. as it can be implanted inside thin things such as paper [33]
iv. **Low Frequency** works on 125 kHz, it has approximately half a meter reading range and mostly used for short reading range applications such as shops, manufacturing factories, inventory control through in and out counts, access control through showing a card to the reader. These low frequency tags are mostly not affected when applied on wet and near metal surfaces [32] [34].

3.2.3.2 RFID Readers

RFID Readers are composed of an antenna and an electronic module. The antenna is used for communicating with RFID tags wirelessly. The electronic module is most often networked to the host computer through cables and relay message between the host computer and all the tags within the antenna's range. The electronic module also perform a number of security functions such as

encryption/decryption and user authentication, and another critical function called anti-collision, which enables a reader to communicate with multiple tags simultaneously [29]. The reader is also called the coupler. The coupler can send information in two directions: It can read information from a tag and send it to the PC (read mode), or it can read information from the PC and send to an RFID tag (write mode) [35]

3.2.3.3 Host Computer or PC

It provides an interface between the RFID hardware and application based system, which is the "brain" of any RFID system. They are used to network multiple RFID interrogators together and to centrally process information. The controller in any network is most often a PC or a workstation running database or application software, or a network of these machines [36].

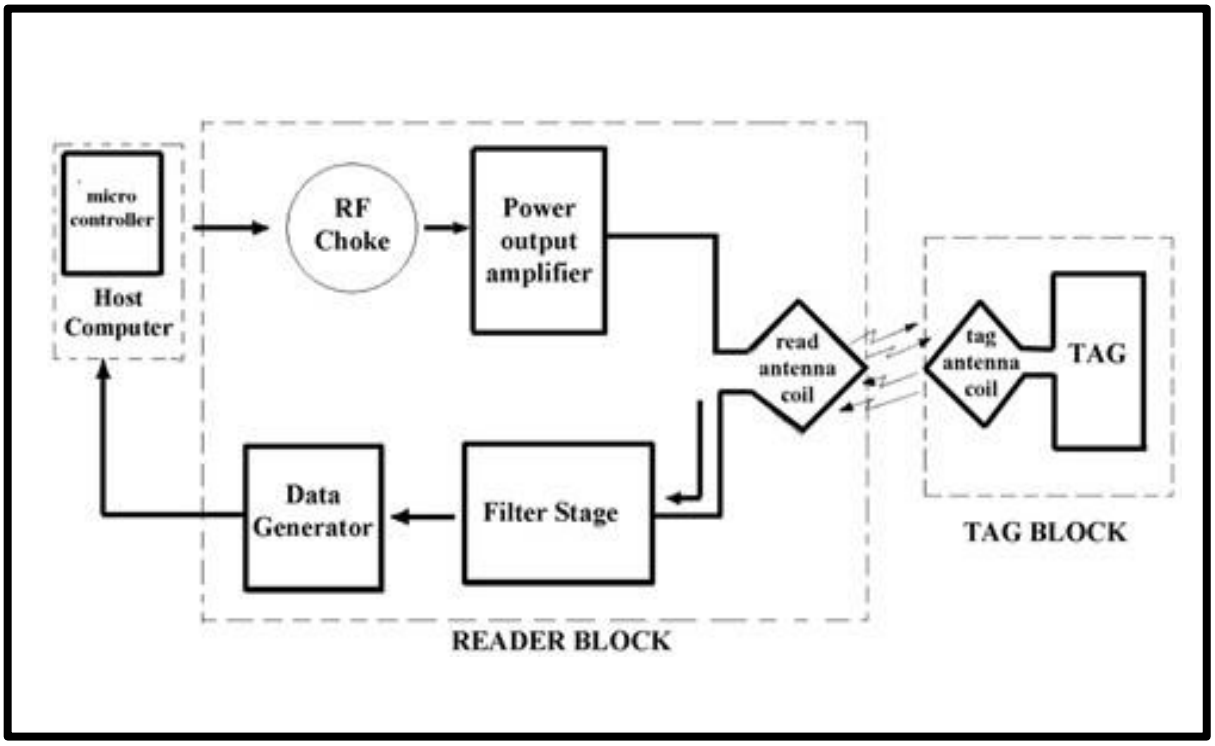


Fig 3.2.3.3 Basic architecture of a RFID system

Parameter	RFID	Barcode	Smart card
Line of Sight	Not required (in most cases)	Required	Required (exposed to reader)
Memory	Small<= 2KB	No memory	Large
Cost	Medium	Low	High
Range	Inches to100's of feet	Inches to feet	Inches
Reusability	Yes	No	Yes
Read Rate	Multiple (simultaneously)	One at a time	One at a time
Security	Medium	Very Low	High(Encryption)
Technology	RF	Optical Laser	RF
Read/Write	Both	Read only	Both

Table 3.2-2 Comparison of RFID with other Technologies

3.3 PIR Motion Sensor

PIR sensors allows to sense motion [37]. They are used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

Fig 3.3-1 PIR Motion Sensor

PIRs are basically made of a pyroelectric sensor (which can be see above as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

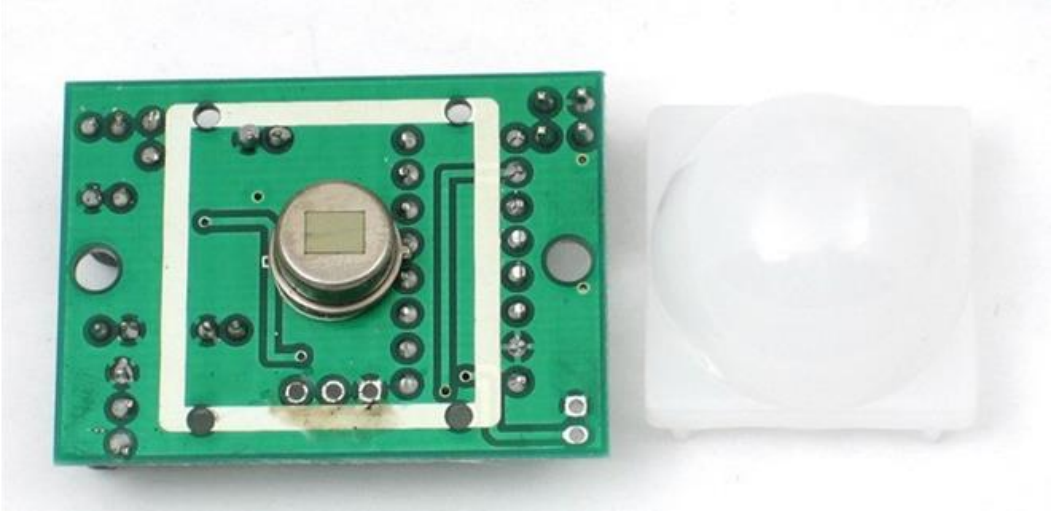


Fig 3.3-2 PIR Sensor Details

Along with the pyroelectic sensor is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the BISS0001 ("Micro Power PIR Motion Detector IC"), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

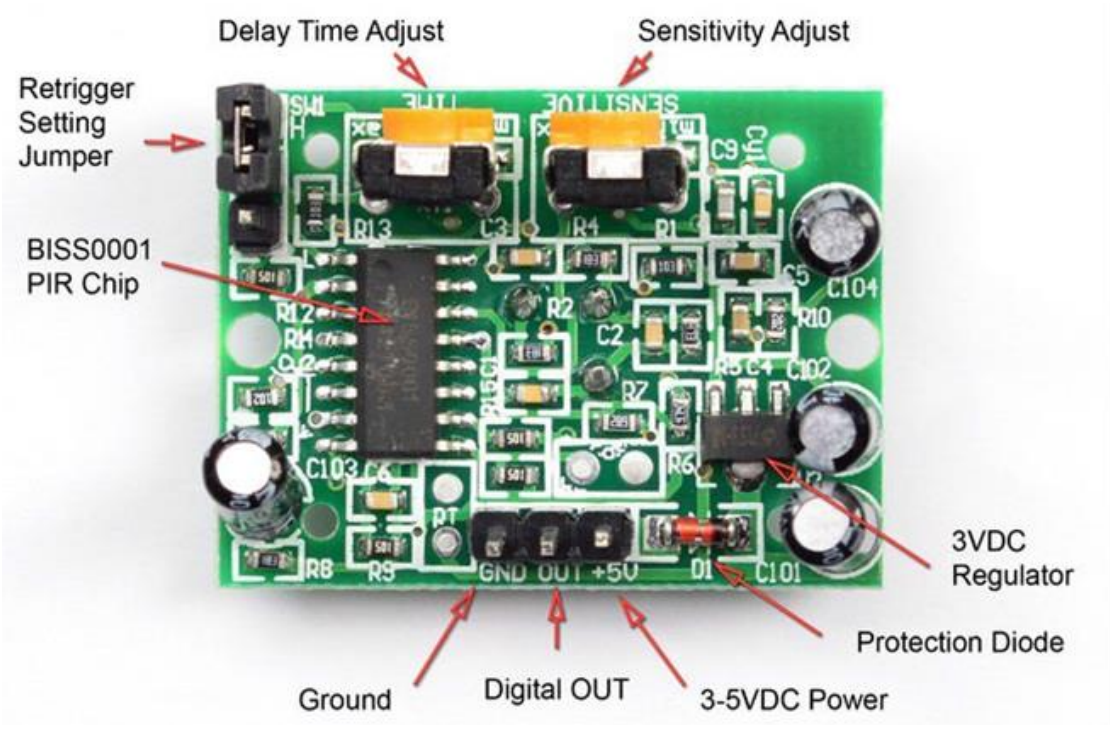


Fig 3.3-3 PIR Detailed Structure

For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. It is worth noting that PIRs won't provide information as to how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance.

3.3.1 Basic Specifications

Nearly all PIRs will have slightly different specifications, although they all pretty much work the same.

- **Output:** Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor.
- **Sensitivity range:** up to 20 feet (6 meters) 110° x 70° detection range
- **Power supply:** 5V-12V input voltage for most modules (they have a 3.3V regulator), but 5V is ideal in case the regulator has different specs
- **Sensing range:** 7 meters
- **Block time:** 2 Sec
- **Sensor Angle:** $<110^{o}$ cone angle

3.3.2 How PIR Works

PIR sensors are more complicated than many of the other sensors (Like photocells, FSRs and Tilt Switches) because there are multiple variables that effect the sensors input and output.

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so there are two slots that can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first

intercepts one half of the PIR sensor, which causes a *positive differential* change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These changes pulses are what is detected.

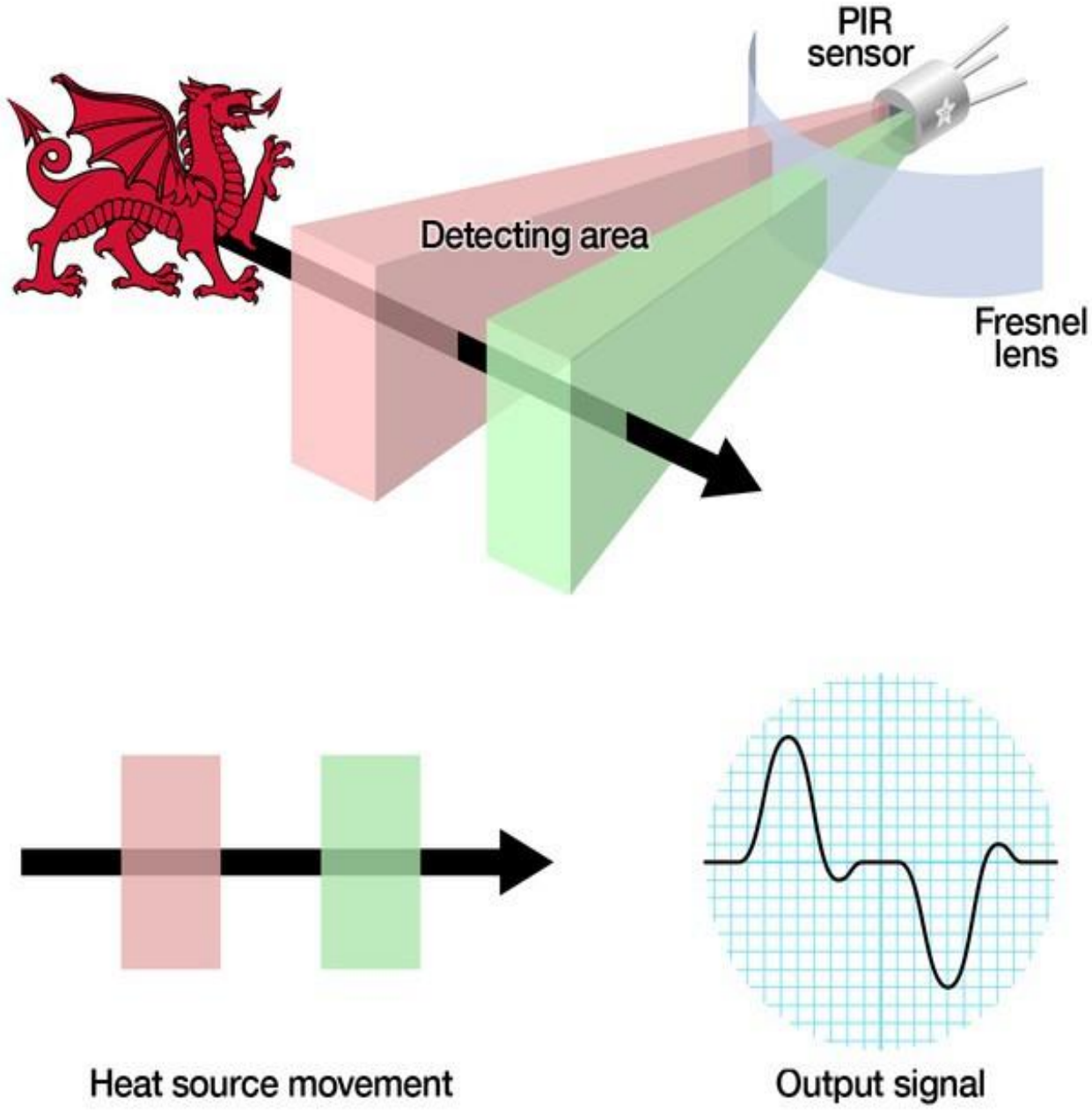


Fig 3.3.2 Working of PIR Sensor

3.3.3 PIR Connection

Connecting PIR sensors to a Arduino is simple. The PIR acts as a digital output, hence we only need to connect to pin to be detecting high or low.

Power the PIR with 5V and connect ground to ground. Then connect the output to a digital pin. Below is basic connection with digital pin two connected.

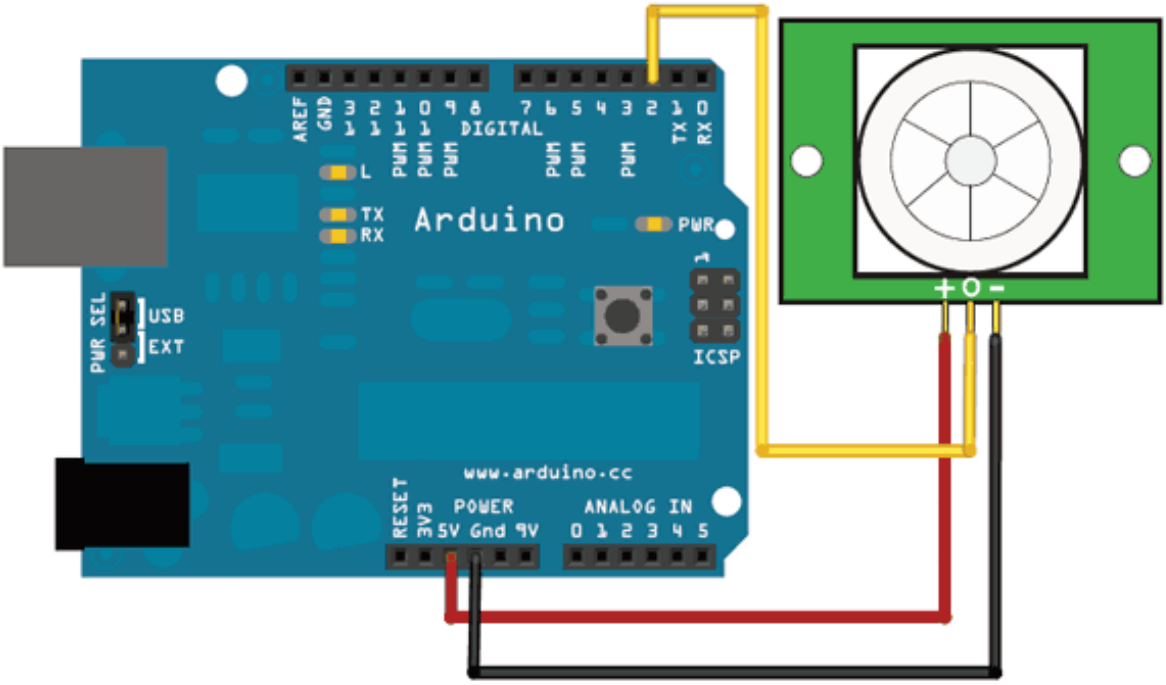


Fig 3.3.3 PIR Connection

3.4 ZigBee

3.4.1 Introduction to ZigBee

ZigBee is specification for a suite of high level communication protocols using small, low power digital radios based on the IEEE 802.15.4 – 2003 standard for wireless personal area networks (WPAN's) [38], such as wireless headphones connecting with cellphones via short range radio. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPAN's, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life and secure networking.

3.4.2 Overview

ZigBee is a low cost, low power, wireless mesh networking proprietary standard [39]. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the lower power usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range.

The ZigBee Alliance, the standards body that defines ZigBee, also publishes application profiles either published or in the works are:

- Home Automation
- ZigBee Smart Energy
- Commercial Building Automation
- Telecommunication Applications
- Personal, Home and Hospital care
- Toys

3.4.3 Uses

ZigBee protocols are intended for use in embedded applications requiring low data rates and low power consumption. ZigBee's current focus is to define a general-purpose, inexpensive self-organizing mesh network that can be used for industrial

control, embedded sensing, medical data collection, smoke and intruder warning, building automation, home automation etc. The resulting network will use very small amounts of power – individual devices must have a battery life of at least two years to pass ZigBee certification.

3.4.4 ZigBee Specifications

ZigBee is the specification of a low-cost, low-power wireless communication solution, meant to be integrated as the main building block of the ubiquitous networks. It is maintained by the ZigBee Alliance, which develops the specification and certifies its proper implementation [40].

ZigBee Alliance was established in August 2001. The ZigBee specification, officially named ZigBee 2007. It offers full wireless mesh networking capable of supporting more than 64,000 devices on a single network. It's designed to connect the widest range of devices, in any industry, into a single control network. The ZigBee specification has two implementation options or Feature Sets: ZigBee and ZigBee PRO.

The ZigBee Feature Set is designed to support smaller networks with hundreds of devices in a single network. The ZigBee Feature Set is the most popular choice of developers and the specification used for most Alliance developed ZigBee Feature Set, plus facilitates ease-of-use and advanced support for larger network comprised of thousands of devices. Both Feature Sets are designed to interoperate with each other, ensuring long-term use and stability. The ZigBee specification enhances the IEEE 802.15.4 standard by adding network and security layers and an application frame work. From this foundation, Alliance developed standards, technically referred to as Public Application Profiles, can be used to create a multi-vendor interoperable solutions. For custom application where interoperability is not required, manufacturers can create their own manufacturer specific profiles.

Some of the characteristics of ZigBee include:

- Global operation in the 2.4GHz frequency band according to IEEE 802.15.4
- Regional operation in the 915 MHz (Americas) and 868 MHz (Europe).

- Frequency agile solution operating over 16 channels in the 2.4GHz frequency.
- Incorporates power saving mechanisms for all device classes.
- Discovery mechanism with full application confirmation.
- Pairing mechanism with full application confirmation.
- Multiple star topology and inter-personal area network (PAN) communication.
- Various transmission options including broadcast.
- Security key generation mechanism.
- Supports Alliance standards (public application profiles) or manufacturer profile

3.4.5 ZigBee Standard Device Types

ZigBee devices are the combination of application (such as light sensor, lighting control etc.), ZigBee logical (Co-Ordinator, Router, End Devices) and ZigBee physical device type (Full Function Device and Reduced Function Device)

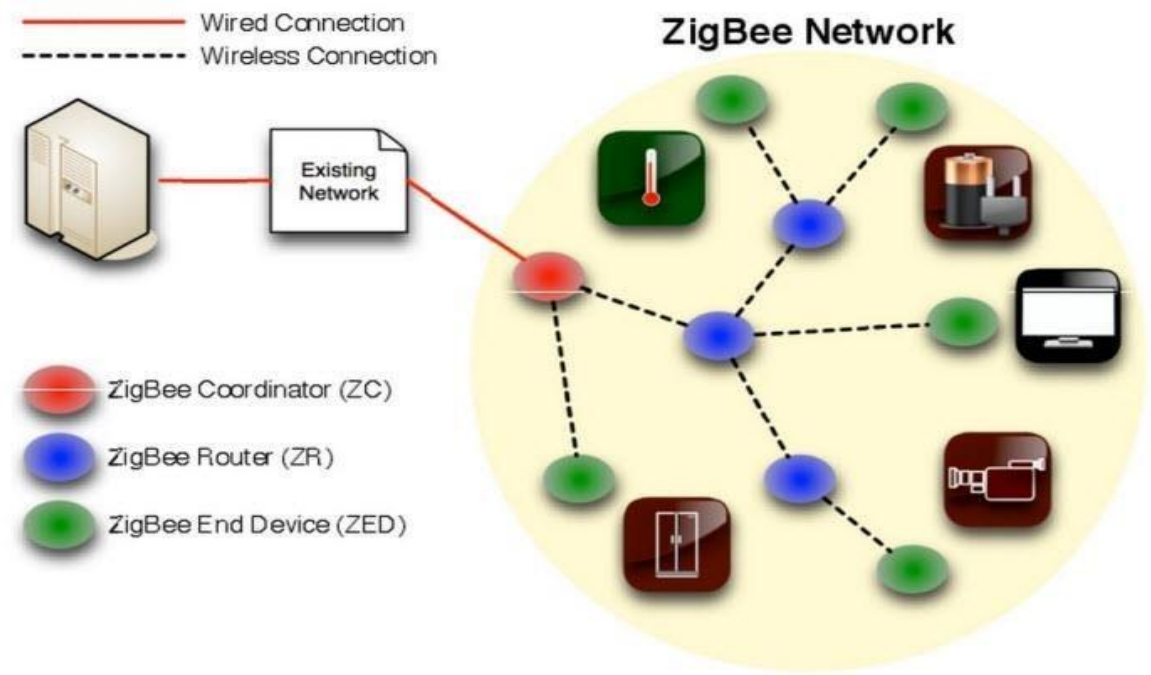


Fig 3.4.5 ZigBee Network

3.4.6 Application Areas

- Home Entertainment and Control — Smart lighting, advanced temperature control, safety and security, movies and music
- Home Awareness — Water sensors, power sensors, energy monitoring, smoke and fire detectors, smart appliances and access sensors
- Mobile Services — m-payment, m-monitoring and control, m-security and access control, m-healthcare and tele-assist
- Commercial Building — Energy monitoring, HVAC, lighting, access control
- Industrial Plant — Process control, asset management, environmental management, energy management, industrial device control

3.4.7 Device Types and Networking Topologies

There are three different types of ZigBee devices:

- ZigBee coordinator (ZC): The most capable device, the coordinator forms the root of the network tree and might bridge to other networks. There is exactly one ZigBee coordinator in each network since it is the device that started the network originally. It is able to store information about the network, including acting as the Trust Centre & repository for security keys.
- ZigBee Router (ZR): As well as running an application function, a router can act as an intermediate router, passing on data from other devices.
- ZigBee End Device (ZED): Contains just enough functionality to talk to the parent node (either the coordinator or a router); it cannot relay data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving long battery life. A ZED requires the least amount of memory, and therefore can be less expensive to manufacture than a ZR or ZC.

ZigBee Networking Layer Topologies

ZigBee networking layer (NWK) manages the network formation. The network should be either star connected or peer to peer connected. When all routers and end devices in a network directly communicate to the coordinator it is known as star connection. For any device to be able to communicate with the coordinator it has to be within the radio sphere of influence of the coordinator. This is a problem for the star connected network because it is not possible to grow the network beyond the coverage of coordinator.

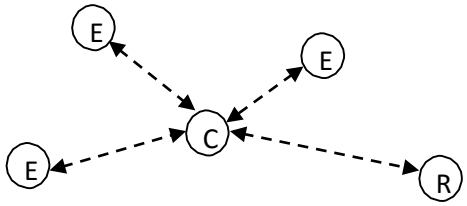


Fig 3.4.7-1: Star Connected Network

Generally in peer to peer communication every node is able to communicate with every other node. There would be one coordinator and rest of the node should be able to relay message which means they have to be router. This type of peer to peer topology is known as mesh network where everyone talks with everyone.

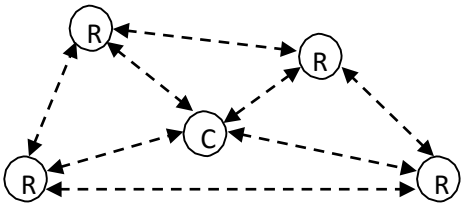


Fig 3.4.7-2: Mesh Connected Network

Another type of peer to peer network is supported by ZigBee known as tree topology. First of all coordinator establishes the initial network. Routers form the

branches and relay the messages. The end devices act as leaves of the tree and do not participate in message routing. Router can grow the network beyond the initial network established by the coordinator.

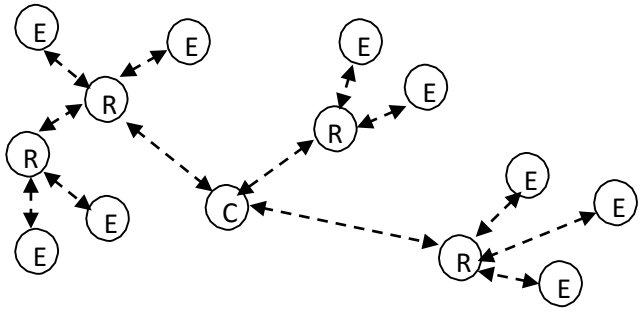


Fig 3.4.7-3: Tree Connected Network

No matter which topology is used coordinator controls the network and does the following minimum tasks:

1. Allocate a unique address (16-bit or 64-bit) to each device in the network
2. Initiate, terminate and route the messages throughout the network
3. Select a unique PAN for the network.

3.4.8 Software and Hardware

The software is designed to be easy to develop on small, inexpensive microprocessors. The radio design used by ZigBee has been carefully optimized for low cost in large scale production. It has few analog stages and uses digital circuits wherever possible.

Even though the radio themselves are inexpensive, the ZigBee Qualification Process involves a full validation of the requirements of the physical layer. This amount of concern about the Physical Layer has multiple benefits, since all radios derived from that semiconductor mask set would enjoy the same RF characteristics. On the other hand, an uncertified physical layer that malfunctions could cripple the battery life span

of other devices on a ZigBee network. Where other protocols can mask other poor sensitivity or other esoteric problems in a fade compensation response, ZigBee radios have very tight engineering constraints: they are both power and bandwidth constrained. Thus, radios are tested to the ISO 17025 standard with guidance given by Clause 6 of the 802.15.4-2006 Standard. Most vendors plan to integrate the radio and microprocessor onto a single chip.

Serial Interface Protocols

The XBee module has support for transparent (AT mode) and Application Programming Interface (API mode) serial interfaces. Below is a short description for both modes.

AT mode: In this mode the XBee module behaves as a serial line replacement. All the UART data received from the microcontroller is queued up for RF transmission. When data is received it is sent out through the DOUT pin. This is the simplistic version of serial communication using XBee modules.

API mode: In this mode all data entering and leaving the module is contained in frames that define operations or events within the module. Using this mode it is possible to configure module and route data at the host application layer. Transmitting RF data to multiple remotes only requires changing the address in the API frame. This process is much faster than in transparent operation where the application must enter AT command mode, change the address, exit command mode, and then transmit data. Each API transmission can return a transmit status frame indicating the success or reason for failure. Some of the advantage of these modes are mentioned below:

- Supports both broadcast and multicast transmission
- Packet include checksum for data integrity
- RSSI can be obtained from a RX packet
- Remote radio can be configured with the remote AT feature.
- Received packets contain source address.

ZigBee Parameters

There are a few important ZigBee parameters. Most of these parameters can be configured through XCTU.

Function Set: From this option user define the device role and select mode of operation.

BD: Baud Rate is the speed of the serial interface which should be same for all the devices.

PAN ID: All the devices within a network must have the same PAN ID.

SH and SL: Every XBee device has a unique 64-bit address. This parameter gives the source address.

DH and DL: When user want to send data to another XBee the 64-bit address of the receiver should be set to DH and DL of the sender so that sender gets the information regarding where to send data.

MY: This parameter is known as 16-bit Network Address which is automatically assigned by the coordinator.

PL: This is Power Level, using this parameter it's possible to select the transmit power

3.4.9 Application layer

The application layer is the highest-level layer defined by the specification, and is the effective interface of the ZigBee system to its end users. It comprises the majority of components added by the ZigBee specification: both ZDO and its management procedures, together with application objects defined by the manufacturer, are considered part of this layer.

3.4.10 Communication Models

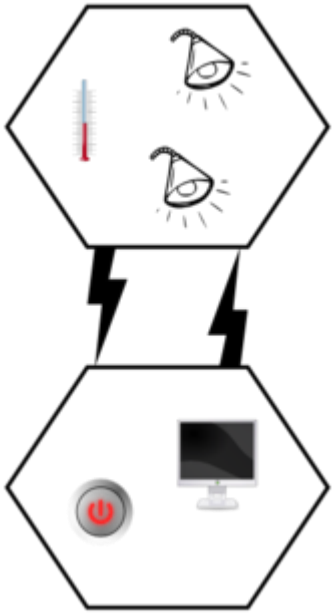


Fig 3.4.10 ZigBee communication model

An application may consist of communicating objects which cooperate to carry out the desired tasks. The focus of ZigBee is to distribute work among many different devices which reside within individual ZigBee nodes which in turn form a network (said work will typically be largely local to each device, for instance the control of each individual household appliance). The collection of objects that form the network communicate using the facilities provided by APS, supervised by ZDO interfaces. The application layer data service follows a typical request – confirm/indication – response structure. Within a single device, up to 240 application objects can exist, numbered in the range 1-240. 0 is reserved for the ZDO data interface and 255 for broadcast; the 241-254 range is not currently in use, but may be in the future.

There are two services available for application objects to use (in ZigBee 1.0)

- The key-value pair service (KPV) is meant for configuration purposes. It enables description, request and modification of object attributes through a simple interface based on get/set and event primitives, some allowing a request for response. Configuration uses compressed XML (full XML can be used) to

provide an adaptable and elegant solution.

- The message service is designed to offer a general approach to information treatment, avoiding the necessity to adapt application protocols and potential overhead incurred on by KPV. It allows arbitrary payloads to be transmitted over APS frames.

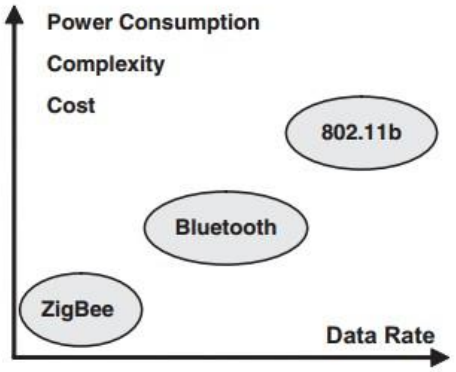
Addressing is also part of the application layer. A network node consists of an 802.15.4-conformant radio transceiver and one or more device descriptions (basically collections of attributes which can be polled or set, or which can be monitored through events). The transceiver is the base for addressing, and devices within a node are specified by an *endpoint identifier* in the range 1-240.

Parameter	**ZigBee**	**Bluetooth**	**Wibree**	**Wi-Fi**
Frequency	2.4GHz	2.4GHz	2.4GHz	2.4GHz
Range	30m – 1.6km	30- 300ft	Upto10ft.	100 – 150 ft.
Data Rate	250kbps	1 Mbps	1 Mbps	11- 54 Mbps
Power Consumption	Low	Medium	Low	High
Cost	Low	Low	Low	High
Modulation Protocol	DSSS,CSMA/CA	FHSS	FHSS	DSSS/CCK,OFDM

Table 3.4 Comparison of ZigBee with Other Technologies

3.5 XBee

For the wireless communication between sensor nodes and the gateway node ZigBee RF modules were used. All the ZigBee devices are based on ZigBee standard which has adopted IEEE 802.15.4 for its physical layer and MAC protocols. The wireless devices based on this standard operate in 868 MHz, 915 MHz and 2.4 GHz frequency bands having a maximum data rate 250Kbps.



	Data Rate	Typical Range	Application Examples
ZigBee	20 to 250 Kbps	10–100 m	Wireless Sensor Networks
Bluetooth	1 to 3 Mbps	2–10 m	Wireless Headset Wireless Mouse
IEEE 802.11b	1 to 11 Mbps	30–100 m	Wireless Internet Connection

Fig 3.5-1: Comparison between ZigBee, Bluetooth and IEEE 802.11b [41]

Bluetooth and Wi-Fi is the other two examples for short range wireless networking. Bluetooth has a typical data rate up to 3 Mbps with indoor range between 2 to 10m. Wi-Fi based on IEEE 802.11 family has typical range somewhere between 30 to 100 meters having data rate up to 100 Mbps. Communication protocol for ZigBee has been kept simple compared to Wi-Fi to reduce power consumption at the expense of lower data rate but the typical indoor range is quite similar to Wi-Fi. So it addresses the need for very low cost implementation of low data rate wireless networks with ultra-low power consumption. The comparison between these three protocols is summarized in Fig 3.5-1.

There are quite a few variants of ZigBee module available from Digi. I used XBee-PRO S2B XBP24BZ7WIT-004. There are 2 version of this XBee available which are known as XBee pro series1 and series2. It is important to note that they don't work together. It's not possible to use both versions in the same network. All of our XBee used for the design of the system were XBee pro series2 model. There is no basic hardware difference between the gateway XBee and those deployed in sensor network.

The difference lies mainly in their software configuration. Digi already has software named XCTU for configuring XBee, details of which is going to be discussed in later chapters.

Fig 3.5-2: XBee Pro S2B [42]

The detail specification for XBee is provided in the following table.

Specification	**XBee Pro (S2B)**
RF Data Rate	250 Kbps
Indoor Range	300 ft
Outdoor Range	2 miles
Configuration Method	API or AT
Interference Immunity	DSSS
Channel Access	CSMA-CA
Encryption	128-bit AES
Supply Voltage	2.7 - 3.6 V dc
Transmit Current	220 mA
Receive Current	62 mA
Power-Down Current	4 uA
Receiver sensitivity	-102 dBm

Table 3.5 XBee Pro Specification [42]

3.6 Conclusion

Thus in this chapter we have discussed briefly about the following:

Arduino:

Here we have discussed about brief details of Arduino, Why is it required, What can be done with Arduino, The brief Hardware of various Arduino and Shields of it, The Arduino Specifications of UNO and MEGA followed by the brief details of the Arduino Software.

RFID:

Here we have discussed briefly about details of RFID, followed by the Technical characteristics of RFID, Architecture and Operation and discussed briefly about the RFID tags and RFID Readers.

PIR:

Here we have discussed briefly about the Construction and working of the PIR Motion sensor along with its Basic Specifications and How to Use a PIR.

ZigBee:

Here we have discussed about Uses, Typical Applications, Types of ZigBee devices, Protocol Layers, Application Layers, Specification Layers, Origin of ZigBee, History of ZigBee and the Security and Communication Architectures provided by ZigBee modules.

XBee:

Here we have discussed briefly about, the basic setup of XBee, software required to configure XBee, its specifications and compared it with other devices.

4.1 ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and hardware to upload programs and communicate with them [43].

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allows to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Verify

Checks the code for errors compiling it.

Upload

Compiles the code and uploads it to the configured board.

Note: If an external programmer is used with the board, the "shift" key on the computer can be used when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in the sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll. To open a sketch late in the list, the File | Sketch book menu can be used instead.

Save
Saves your sketch.

Serial Monitor
Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- *New*

  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

- *Open*

  Allows to load a sketch file browsing through the computer drives and folders.

- *Open Recent*

  Provides a short list of the most recent sketches, ready to be opened.

- *Sketchbook*

  Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

- *Examples*

  Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

- *Close*

  Closes the instance of the Arduino Software from which it is clicked.

- *Save*

  Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

- *Save as*

  Allows to save the current sketch with a different name.

- *Page Setup*

  It shows the Page Setup window for printing.

- *Print*

  Sends the current sketch to the printer according to the settings defined in Page Setup.

- *Preferences*

  Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

- *Quit*

  Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- *Undo/Redo*

  Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

- *Cut*

  Removes the selected text from the editor and places it into the clipboard.

- *Copy*

  Duplicates the selected text in the editor and places it into the clipboard.

- *Copy for Forum*

  Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

- *Copy as HTML*

  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

- *Paste*

  Puts the contents of the clipboard at the cursor position, in the editor.

- *Select All*

  Selects and highlights the whole content of the editor.

- *Comment/Uncomment*

  Puts or removes the // comment marker at the beginning of each selected line.

- *Increase/Decrease Indent*

  Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

- *Find*

  Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

- *Find Next*

  Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

- *Find Previous*

  Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- *Verify/Compile*

  Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

- *Upload*

  Compiles and loads the binary file onto the configured board through the configured Port.

- *Upload Using Programmer*

  This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn Bootloader* command must be executed.

- *Export Compiled Binary*

  Saves a .hex file that may be kept as archive or sent to the board using other tools.

- *Show Sketch Folder*

  Opens the current sketch folder.

- *Include Library*

  Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

- *Add File...*

  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

Tools

- *Auto Format*

  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

- *Archive Sketch*

  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

- *Fix Encoding & Reload*

  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

- *Serial Monitor*

  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

- *Board*

  Select the board that you're using. See below for descriptions of the various boards.

- *Port*

  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

- *Programmer*

  For selecting a harware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

- *Burn Bootloader*

  The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find a comparison table between the various boards here.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

- Arduino Yùn

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino/Genuino Uno

  An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

- Arduino Diecimila or Duemilanove w/ ATmega168 An ATmega168 running at 16 MHz with auto-reset.
- Arduino Nano w/ ATmega328

  An ATmega328 running at 16 MHz with auto-reset. Has eight analog inputs.
- Arduino/Genuino Mega 2560

  An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega

  An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega ADK

  An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Leonardo

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Micro

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Esplora

  An ATmega32u4 running at 16 MHz with auto-reset.
- Arduino Mini w/ ATmega328

  An ATmega328 running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Fio

  An ATmega328 running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATmega328, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino BT w/ ATmega328

  ATmega328 running at 16 MHz. The bootloader burned (4 KB) includes codes to initialize the on-board bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM..
- LilyPad Arduino USB

  An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.

- LilyPad Arduino

  An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328

  An ATmega328 running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328; 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino NG or older w/ ATmega168

  An ATmega168 running at 16 MHz without auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Robot Control

  An ATmega328 running at 16 MHz with auto-reset.
- Arduino Robot Motor

  An ATmega328 running at 16 MHz with auto-reset.
- Arduino Gemma

  An ATtiny85 running at 8 MHz with auto-reset, 1 Analog In, 3 Digital I/O and 2 PWM

4.2 XCTU

Before using any XBee module it is required to set some parameters for each of the modules. There is a software named XCTU from digi international which helped to set these parameters. Using the adapter kit and the FTDI cable it is possible to connect the XBee module to any computer [44].

PC settings of XCTU is used to test the connectivity of XBee with the computer. The baud rate can also be set from here. Terminal window in XCTU helped to see incoming and outgoing packet both in character and hex mode. Using the window Modem configuration parameters for XBee module can be configured.

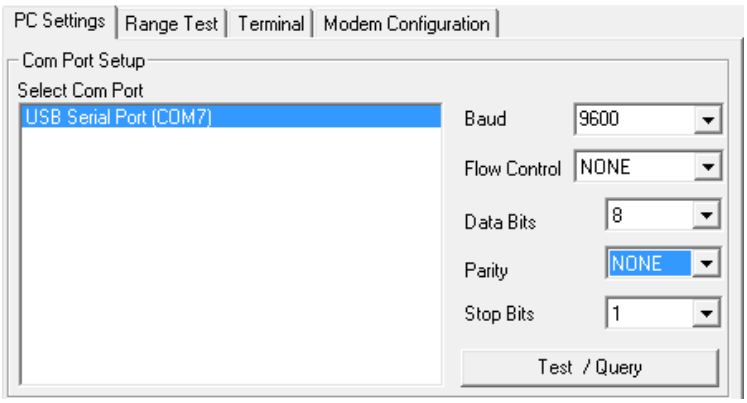


Fig 4.2: XCTU Software

4.3 MySQL Server

MySQL Server is used in the JDBC layer for storing all the readings from the sensor network and present them on the front end in a user-friendly manner. MySQL Server is a Relational Database Management System (RDMS) owned by Oracle Corporation [45]. It supports a superset of Structured Query Language SQL, the most common database language. It is commonly used by businesses for small to medium sized databases, and in the past 5 years large enterprise databases have been implemented and it competes with other relational database products for this market segment.

MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, a database management system such as MySQL Server is needed. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns,

offers a flexible programming environment. Rules can be setup that governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on programming environment, one can enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. "The SQL standard" means the current version of the SQL Standard at any time.

MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. The source code can be studied and can be change to suit the needs. The MySQL software uses the GPL (GNU General Public License), to define what one may and may not do with the software in different situations.

The MySQL Database Server is very fast, reliable, scalable, and easy to use.

MySQL Server can run comfortably on a desktop or laptop, alongside other applications, web servers, and so on, requiring little or no attention. If an entire machine is dedicated to MySQL, settings can be adjusted to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

There are many features of MySQL, they are:

Internals and Portability

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms.
- Uses multi-layered server design with independent modules.
- Designed to be fully multi-threaded using kernel threads, to easily use multiple CPUs if they are available.
- Uses very fast B-tree disk tables (MyISAM) with index compression.
- Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
- Executes very fast joins using an optimized nested-loop join.
- Implements in-memory hash tables, which are used as temporary tables.
- Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.

Data Types

- Many data types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, FLOAT, DOUBLE, CHAR, VARCHAR, BINARY, VARBINARY, TEXT, BLOB, DATE, TIME,

DATETIME, TIMESTAMP, YEAR, SET, ENUM, and OpenGIS spatial types.

- Fixed-length and variable-length string types.

Statements and Functions

- Full operator and function support in the SELECT list and WHERE clause of queries. For example:
- Full support for SQL GROUP BY and ORDER BY clauses. Support for group functions (COUNT(), AVG(), STD(), SUM(), MAX(), MIN(), and GROUP_CONCAT()).
- Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with standard SQL syntax.
- Support for aliases on tables and columns as required by standard SQL.
- Support for DELETE, INSERT, REPLACE, and UPDATE to return the number of rows that were changed (affected), or to return the number of rows matched instead by setting a flag when connecting to the server.

Security

- A privilege and password system that is very flexible and secure, and that enables host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

Connectivity

- Clients can connect to MySQL Server using several protocols:
- Clients can connect using TCP/IP sockets on any platform.
- On Windows systems, clients can connect using named pipes if the server is started with the --enable-named-pipe option. Windows servers also support shared-memory connections if started with the --shared-memory option. Clients can connect through shared memory by using the --protocol=memory option.
- On Unix systems, clients can connect using Unix domain socket files.
- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs

that use ODBC (Open Database Connectivity) connections. For example, MS Access can be used to connect to MySQL server. Clients can be run on Windows or Unix. Connector/ODBC source is available.

- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix.
- MySQL Connector/Net enables developers to easily create .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET aware tools. Developers can build applications using their choice of .NET languages. MySQL Connector/Net is a fully managed ADO.NET driver written in 100% pure C#. See MySQL Connector/Net Developer Guide.

Localization

- The server can provide error messages to clients in many languages.
- Full support for several different character sets, including latin1 (cp1252), german, big5, ujis, several Unicode character sets, and more. For example, the Scandinavian characters “å”, “ä” and “ö” are permitted in table and column names.
- All data is saved in the chosen character set.
- Sorting and comparisons are done according to the default character set and collation. is possible to change this when the MySQL server is started. To see an example of very advanced sorting, look at the Czech sorting code. MySQL Server supports many different character sets that can be specified at compile time and runtime.
- The server time zone can be changed dynamically, and individual clients can specify their own time zone.

Clients and Tools

- MySQL includes several client and utility programs. These include both command-line programs such as **mysqldump** and **mysqladmin**, and graphical programs such as MySQL Workbench.

Here for the project we are using MySQL server through an open source GUI of MySQL which is called as SQLyog**,** by the help of this SQLyog the data is entered in simple table format without any usage of other programming languages.

4.4 NetBeans IDE

NetBeans IDE is free and open source software which helps us in quick and easy development of Java desktop, mobile, and web applications, as well as HTML5 applications with HTML, JavaScript, and CSS. The IDE is a Java Enterprise Edition (Java EE) which also provides a great set of tools for PHP and C/C++ developers. It has a large community of users and developers around the world [46].

NetBeans IDE is used to develop an HTML5 web application using HTML, JavaScript and CSS along with the help of GlassFish web server.

HTML5 is a web development support kit which has responsive design capability, provided with enhanced JavaScript and CSS editing and styling. This is an inbuilt feature of the NetBeans IDE, used in the project to develop a front-end servlet.

The Database Explorer helps in connecting to databases and view their content. You can create, modify, and delete tables, columns, and indices in databases directly from the IDE. There are three different databases:

1. Oracle Database
2. JavaDB Database
3. MySQL Database

MySQL Database explorer is used, which is connected through a GUI called as SQLyog and the data can be updated or deleted from the database, through both SQLyog and MySQL Database.

The four vital features of NetBeans IDE are:

1. Best support for latest Java Technologies
2. Fast and smart code editing

3. Easy and efficient project management
4. Rapid user interface development

NetBeans IDE is used with Java programming, MySQL JDBC connector and have utilized the glassfish server along with the HTML5 web application.

The key elements of database connectivity are the following:

Database: The repository where data is stored for an enterprise. Java EE applications access relational databases through the JDBC API. For administration procedures.

JDBC Connection Pool: A JDBC connection pool is a group of reusable connections for a particular database. For administration procedures, Administering JDBC Connection Pools can be referred.

JDBC Resource: A JDBC resource (data source) provides applications with a means of connecting to a database. To create a JDBC resource, specify the connection pool with which it is associated. Multiple JDBC resources can specify a single connection pool. A JDBC resource is identified by its Java Naming and Directory Interface (JNDI) name.

JDBC Driver: A database driver is a software component that enables a Java application to interact with a database connectivity API. Each database requires its own driver.

At runtime, the following sequence occurs when an application connects to a database:

1. The application gets the JDBC resource associated with the database by making a call through the JNDI API.

   Using the JNDI name of the resource, the naming and directory service locates the JDBC resource. Each JDBC resource specifies a connection pool.

2. Using the JDBC resource, the application gets a database connection.

GlassFish Server retrieves a physical connection from the connection pool that corresponds to the database. The pool defines connection attributes such as the database name (URL), user name, and password.

3. After the database connection is established, the application can read, modify, and add data to the database.

   The application accesses the database by making calls to the JDBC API. The JDBC driver translates the application's JDBC calls into the protocol of the database server.

4. When the application is finished accessing the database, the application closes the connection and returns the connection to the connection pool

GlassFish Server

Oracle GlassFish Server provides a server for developing and deploying Java Platform Enterprise Edition (Java EE) applications and web Java Web Services [47].

As an administrator of GlassFish Server, the main responsibilities are to establish a secure GlassFish Server environment and to oversee the services, resources, and users that participate in that environment. The key tasks include configuring resources and services, managing GlassFish Server at runtime, and fixing problems that are associated with the server.

It uses a slew of default data:

Item	Default
Domain Name	domain1
Administration Password	Admin
Administration Server Port	4848
HTTP Port	8080
HTTPS Port	8181
Pure JMX Clients Port	8686
Message Queue Port	7676
IIOP Port	3700
IIOP/SSL Port	3820
IIOP/SSL Port With Mutual Authentication	3920

Table 4.5 Default Administration Values

4.5 Conclusion

All the various applications/software's, used in the project have been discussed briefly along with the, general setup of the applications /software's and their features have also been discussed.

And the connectivity between the databases with the sensor node(s) has also been discussed along with the basic setup for the XBee and Arduino boards.

The need for the server has been discussed along with its default values.

5.1 Introduction

This section deals with the general working of the hardware components, which forms the complete structure of the project, followed by the implementation of project and concluding with the deployment of the sensors, which is done for the sole purpose of the proper authentication.

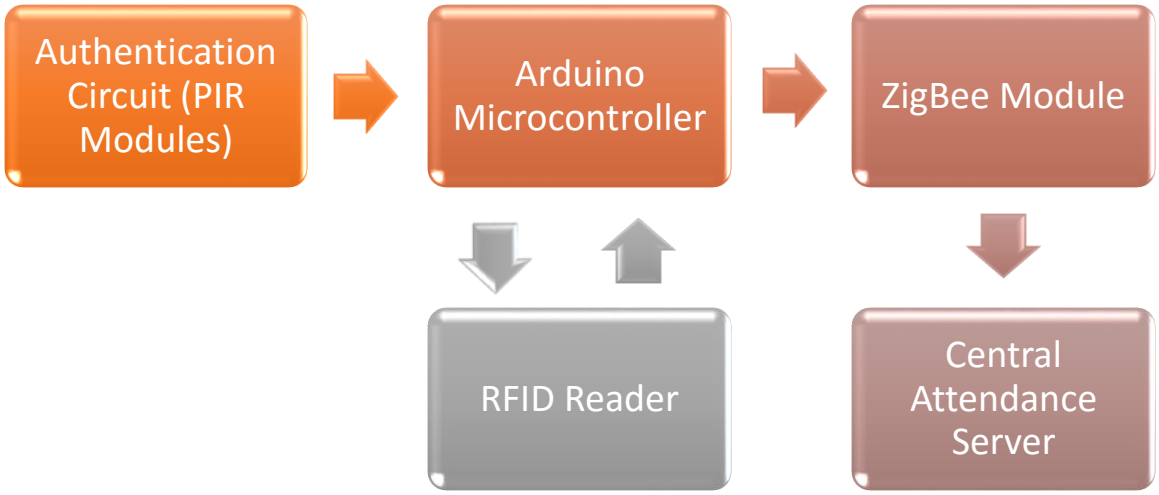


Fig 5.1-1: Basic Structure

The system provides multi node environment (ZigBee module can reach to more than 64,000 nodes) and it is capable of dealing with an unlimited number of rooms by increasing the number of reading nodes. Every room can store up to 45,000 card IDs. It is easily scalable by extending the tag ID of multiple bytes to single byte and also by increasing memory size of Arduino board, so that each room can handle up to 65,000 card IDs. The network block diagram is shown in fig 5-2.

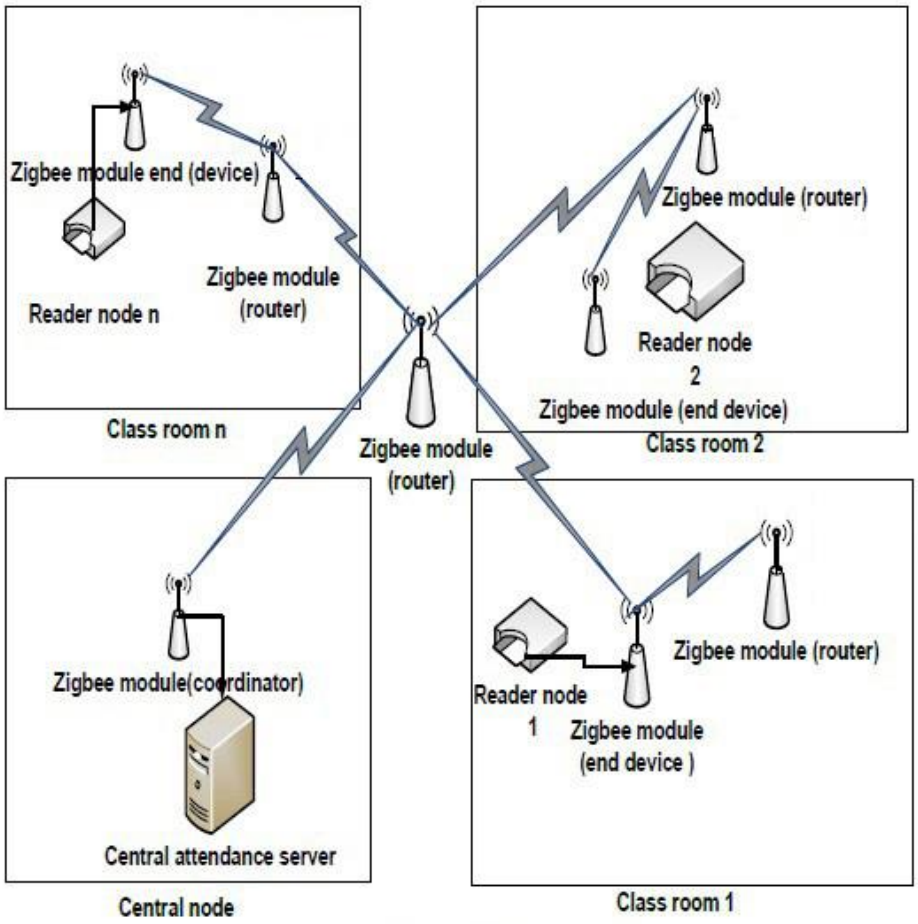


Fig 5.1-2: Network Block diagram

The general working of the system is: First Microcontroller reads the number of people and then Reader detects RFID card and forward that ID to Arduino microcontroller, which after the authentication of the ID, generates a specific number (1byte code) against that ID. This specific number is then forwarded to the ZigBee transceiver (via serial link) from where it is sent to the receiving nodes. One of the receiving nodes is the central attendance server where attendance record is managed. The detailed working of the system is discussed in this chapter.

5.2 Student Identification

This is the most important task that the system would do in order to carry out the other use of the system. When a user enters a room, his unique ID is sent to the authentication circuit and the central attendance server. The person is identified and a log is made according to the time and location. If the person identified has a valid ID,

then his presence is marked for the required time and then control present in the program performs the automation function which has been described in its profile.

When a student enters in the classroom, he passes through the door, by cutting of two PIR motion sensors; i.e. entrance sensor and exit sensor. When a student enters in the vicinity area of the entrance sensor, the PIR entrance sensor enables the counter through an Arduino Mega board, and the action is performed by the PIR exit sensor when the person passes by the vicinity area of it.

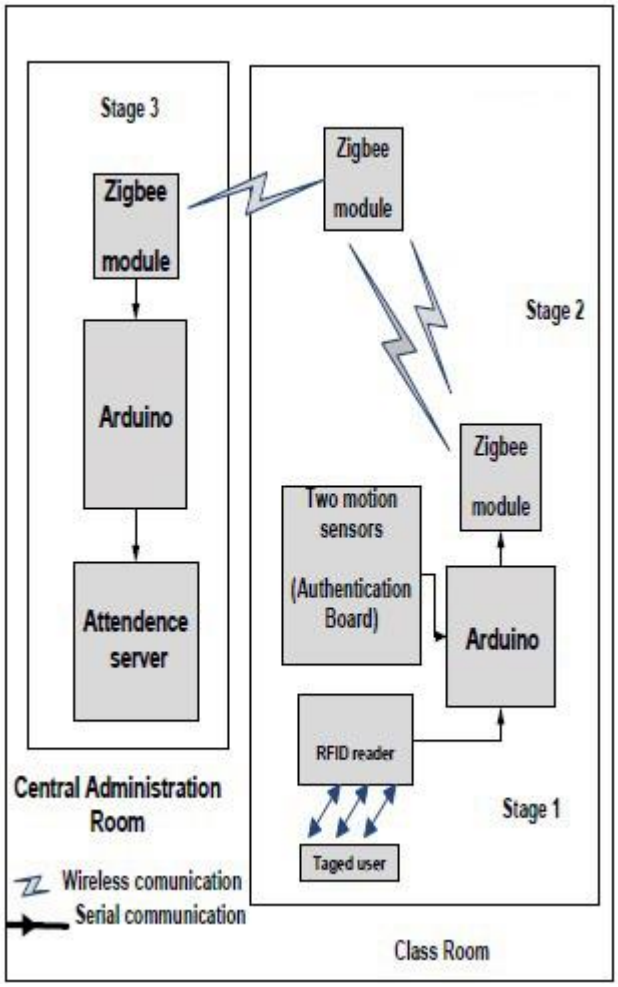


Fig 5.2-1 System Block Diagram

After both PIR sensors sense and give data high to the Arduino Mega board, then RFID reader is activated and it reads the number of cards being scanned. Now, the basic authentication is performed i.e. if the number of cards scanned are equal to the number

of persons entering into the class. If they are equal then a green LED glows, and if they are not equal then a red/orange LED glows and a buzzer sounds, making sure the unauthenticated entry is noticed by the faculty.

The unauthenticated entry can also be such that, a person from another class/section tries to enter into a classroom to which he has no access. Then the person coming into the classroom irrespective of the PIR count gives rise to the buzzer sound along with a red/orange LED being in high state. The flowchart present below gives us a clear understanding of the working of the student identification process:

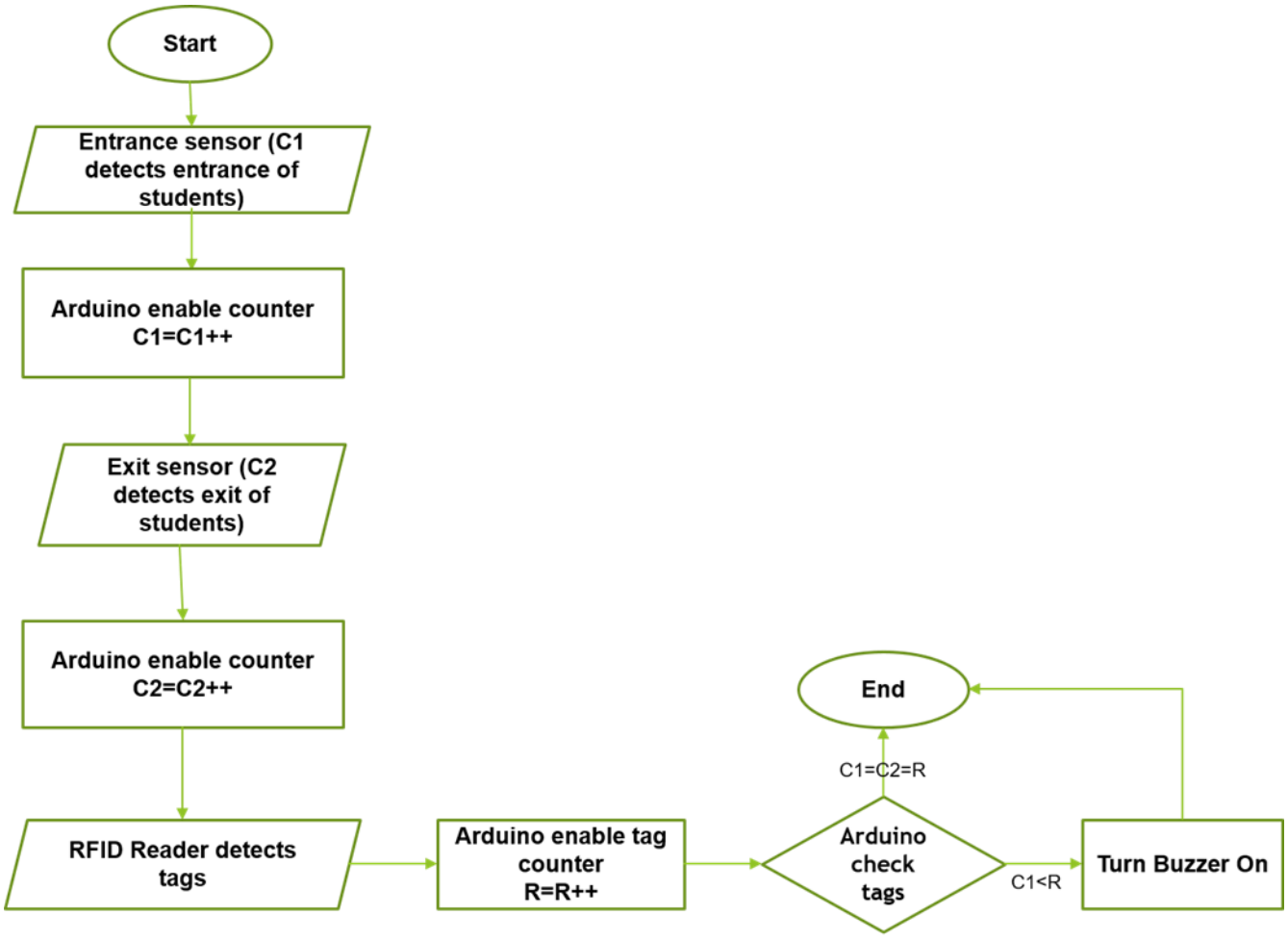


Fig 5.2-2 Flow Chart for Authentication Circuit

5.3 Student Data Authentication and Transfer

After sensing, reading of the cards and basic authentication through the sensors, the data is now present with the Arduino Mega board. The data (Tag) of the students who belong to the class is present in the Arduino Mega board. When the card is read it is mapped with a string of 8 characters and then this string of characters is sent to ZigBee end device (Serial Communication), which transmits it to the central router (Wireless Communication). From this central router the data is received at the ZigBee coordinator (Wireless Communication). All these ZigBee devices are setup initially in AT command mode rather than API command mode. The flowchart present below gives us a clear understanding of the working of the student data authentication and transfer process:

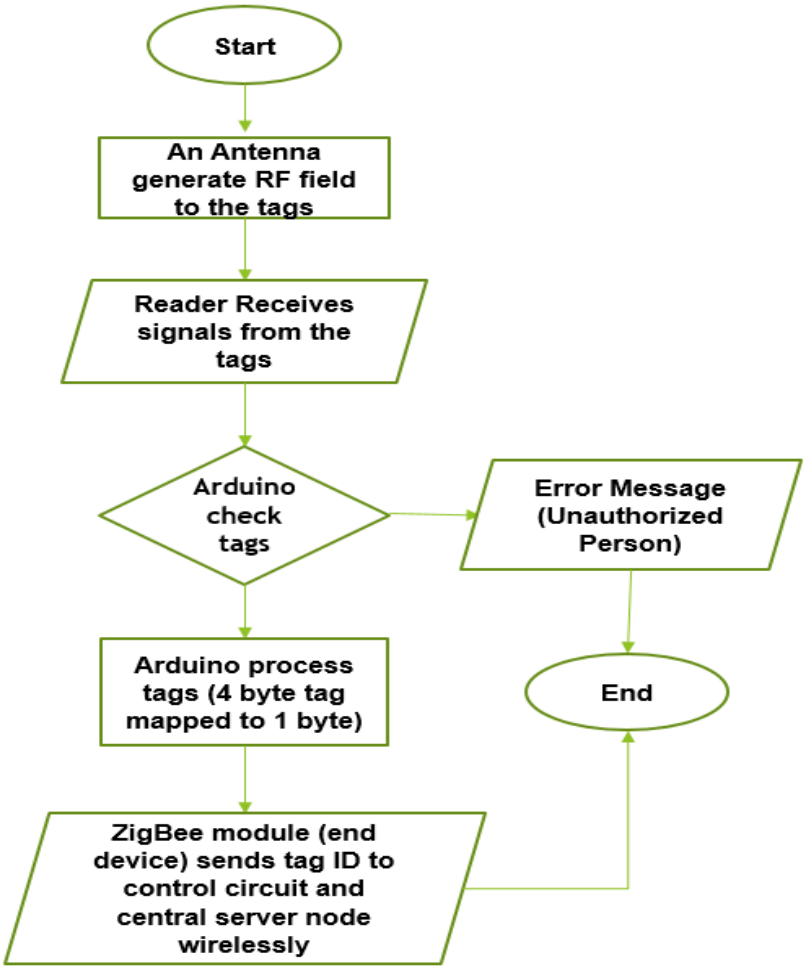


Fig 5.3-1 RFID Reader and ZigBee End Device Node

Then this information from ZigBee is read by Arduino Uno board (Serial Communication). After reading of this data, it is arranged as a string of 8 characters. This string of 8 characters is unique for every card and is different from the code present with each RFID tag. This string of 8 characters can be assigned in any way and can be used to easily differentiate the students of various sections, by keeping a fix set of few characters reserved specifically for differentiation purpose.

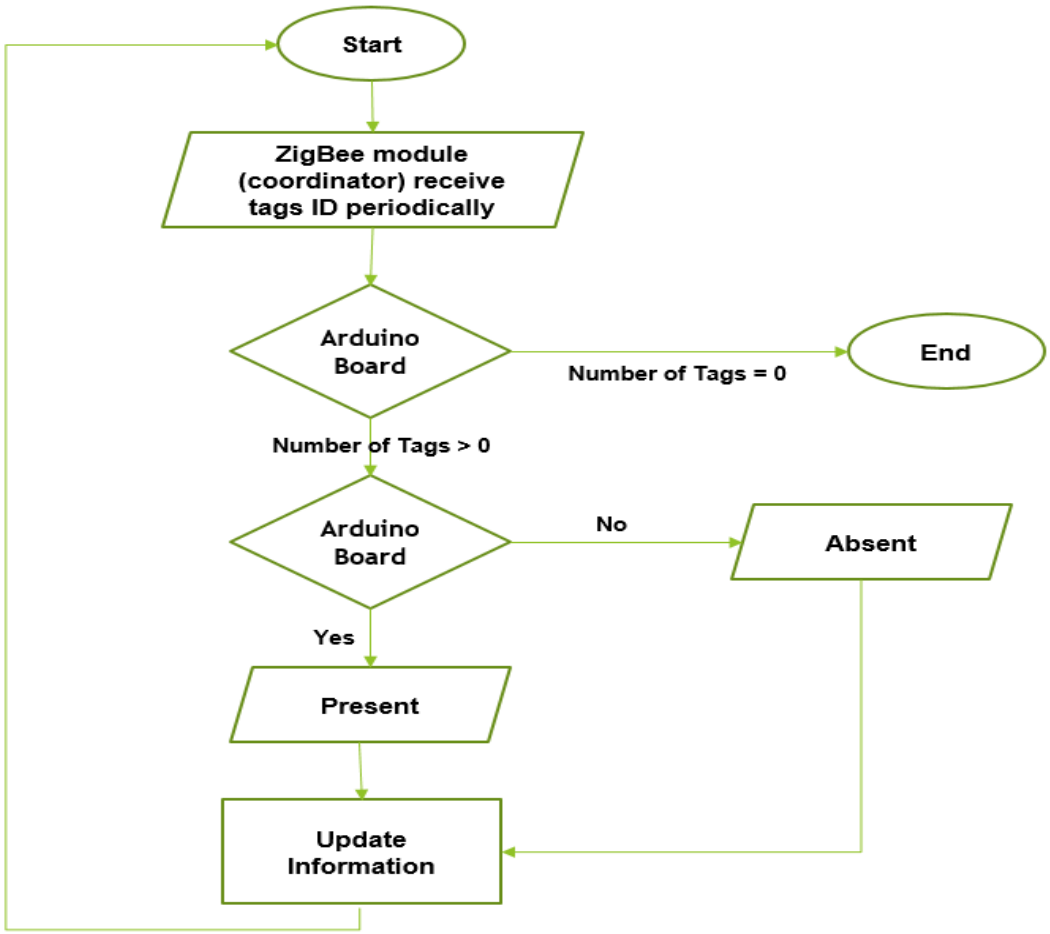


Fig 5.3-2: ZigBee node at Attendance server

This string of 8 characters acts as an input to the software part of the project serially. And then further actions are performed at this end.

5.4 Student Database Management

The predefined strings of 8 bytes are now referred to as card links. These unique card links are made as the reference and all these card links are assigned to the designated users i.e. the students or the faculty. The database is created with these card links and other important details such as Name, Roll Number and Flag. This is the main database which updates itself on regular basis (only flag updates) and is named as asection. There are two other databases, which are vital for the whole software part to be running successfully, properly and without any bugs.

The second one is called as apartial which contains completely different data, is active and the data is updated whenever a student enters as well as leaves the class. The data present in this database is the card link, in time, out time and the most important data which is updated according to the logic is named as PA (which marks whether the student is present or absent).

The third one is called as areport which contains some additional information when compared to asection database such as attendance percentage, number of classes attended and number of classes marked as present. This database is the critical one as it self-updates the data based on the previous two databases and is used while the JSP is used to display the main attendance at the servlet front.

All these databases are created using the software called as MySQL, with the help of SQLyog GUI.

5.5 Student Attendance Marking

When the Arduino UNO board is connected serially to NetBeans software, the crucial working of the software begins and the databases discussed above are constantly updated. When the data is made available to NetBeans program of SimpleRead, the data is accepted and based on the differentiation criteria the respective database of asection/bsection is called for.

If a person's data is read for the first time then the flag counter present in the database of a section is set to as high and if it is read for the second time then the flag counter is

set to as low. Now, when a person belonging to "**A Section"** has entered and the person's data is read for the first time, then the following actions take place accordingly for the program of **SimpleRead**:

- *The flag value in the database asection is set to 1, and the data of the same person is collected from the database of areport.*
- *The classes attended tab is incremented by one number of the same person in the database of areport and the same is updated in the database.*
- *Now, the in time of the person is calculated. This is converted to minutes as the default time received is in milliseconds.*
- *This in time is updated in the database of apartial, and the database is successfully updated with other columns being set as NULL.*

This is continuously performed every first time a card is read. If the card is read for the second time, then the following actions take place accordingly for the same program:

- *The flag value in the database asection is set to 0, and the data of the same person is collected from the database of areport.*
- *Now, the out time of the person is calculated. This is converted to minutes as the default time received is in milliseconds.*
- *This out time is updated in the database of apartial, and the database is successfully updated with other columns being set other values depending on the logic below.*
    - From the information of in time and out time, the total time present in the class is calculated.
    - If this time is more than 150 minutes and less than or equal to 180 minutes, then the attendance of the person is marked as present; if the condition is not true then it is marked as absent.
- *If the person is marked as present in apartial then, the areport database of the*

same person is called for through the card link.

- *Now, the number of classes' present column is incremented, and along with it the percentage column of the person is updated after calculation of the percentage attendance based on the data present in database after the update.*
- *If the person has been marked as absent in apartial then, nothing will be updated and the classes present, percentage attendance columns display the previous values.*

The same actions are performed for a person belonging to **"B Section"** as have been discussed and mentioned above.

After the successful update of all the data is completed, the other programs i.e. the programs required for the proper display, make use of this data present in the *areport/breport* database and display it using the Glassfish server on any browser installed in the server system. This is run on Infra-Net service i.e. any other mobile/device present in the vicinity and having access to the same internet service can access the information that is displayed on the servlet of the server, provided proper address is typed in the address bar of the device which wants to access the data on the servlet, for display purpose.

5.6 Conclusion

Here we have briefly discussed the working of the hardware and the software applications.

The flowcharts, block diagrams, algorithm has also been covered with optimal description of the method of the update of the attendance at the server end.

6.1 Arduino Mega Program

```
#include <SoftwareSerial.h>

#include<SPI.h>

#include<MFRC522.h>

#define RSTPIN 8

#define SSPIN 9

#define BUZZ 4

#define PIR 2

#define PIR1 7


MFRC522 rc(SSPIN,RSTPIN);

SoftwareSerial XBee(19,18); //Rx and Tx

int x=0;

int PIRState=LOW;

int PIR1State=LOW;

int readsuccess;

byte defcard[][4]=

{//DEFINE CARD LOOP STARTS

{0x05,0xEF,0xE7,0x2B},{0x65,0x3D,0xED,0x2B},{0x24,0x0A,0xB6,0x5B},{0x94,0xCD
,0xB3,0x5B},{0x35,0x89,0x15,0x2C},{0x95,0x80,0xC5,0x2A},{0xD5,0xF6,0x1A,0x2C},
{0xC5,0x8A,0x1B,0x2C},{0x05,0x26,0x1F,0x2C},{0x15,0xDE,0x16,0x2C},{0xF4,0x66,0
xB6,0x5B},{0x94,0x46,0xB8,0x5B},{0x64,0x12,0xB3,0x5B},{0xB4,0xA0,0xB2,0x5B},{
0xF4,0xA2,0xB4,0x5B},{0xE5,0x09,0x14,0x2C},{0x35,0xBC,0x1D,0x2C},{0xA5,0x6C,0
x1D,0x2C},{0x84,0xFD,0xB9,0x5B},{0x54,0x74,0xBB,0x5B},{0x65,0xC0,0xD1,0x2B},{
```

0x45,0x20,0x18,0x2C},{0x95,0x24,0xDA,0x2B},{0x75,0x48,0xDB,0x2B},{0x25,0x19,0xD8,0x2B},{0x75,0xE8,0xD0,0x2B},{0x35,0x4A,0xD6,0x2B},{0xB5,0xE7,0xD7,0x2B},{0xF5,0xFC,0xD5,0x2B},{0x65,0x19,0xD0,0x2B},{0x75,0x9B,0xD3,0x2B},{0x94,0x4E,0xB9,0x5B},{0x34,0xC0,0xB5,0x5B},{0x14,0xAF,0xB8,0x5B},{0xD4,0x61,0xB3,0x5B},{0xB4,0xC7,0xBE,0x5B},{0x75,0xEE,0xD2,0x2B},{0xF5,0x02,0x19,0x2C},{0x25,0xAB,0x1B,0x2C},{0x55,0x41,0x1B,0x2C},{0x55,0x52,0xD4,0x2B},{0x05,0xB4,0xD0,0x2B},{0x34,0x97,0xB3,0x5B},{0xF4,0x13,0xB5,0x5B},{0x44,0x2D,0xB5,0x5B},{0xA4,0x34,0xB5,0x5B},{0x45,0x0F,0xEF,0x2B},{0xB4,0xEE,0xBD,0x5B},{0x14,0x85,0xB3,0x5B},{0x74,0x75,0xB2,0x5B},{0x54,0x6D,0xBC,0x5B},{0x75,0x84,0xAB,0x2A},{0xC5,0x21,0xE3,0x2B},{0x55,0xEA,0xC5,0x2A},{0x35,0x93,0xDD,0x2B},{0x45,0x8B,0xEB,0x2B},{0xE4,0xCA,0xB9,0x5B},{0x35,0xA7,0xEB,0x2B},{0x05,0xD1,0xF1,0x2B},{0xD4,0x57,0xB0,0x5B},

{0xB5,0xAE,0xA9,0x2A},{0xF5,0x8B,0xE4,0x2B},{0xC4,0x2C,0xBD,0x5B},{0x25,0x7A,0xAF,0x2A},{0x74,0xEF,0xA0,0x5B},{0x84,0x4B,0x9F,0x5B},{0x84,0x09,0x9F,0x5B},{0xC4,0x00,0x99,0x5B},{0x94,0xFB,0x9D,0x5B},{0x54,0xC5,0x9A,0x5B},{0x95,0x77,0xE4,0x2B},{0x65,0xA9,0x13,0x2C},{0xE5,0xD8,0xD2,0x2B},{0xD5,0x44,0x14,0x2C},{0xB5,0x78,0xCF,0x2B},{0x15,0x94,0xD7,0x2B},{0xD5,0x73,0xF4,0x2B},{0xA5,0xCA,0xD4,0x2B},{0xC5,0x83,0xD5,0x2B},{0xD5,0x43,0x1F,0x2C},{0x55,0x3F,0x1E,0x2C},{0x55,0x2D,0x17,0x2C},{0x85,0x16,0x18,0x2C},{0xB5,0x91,0xE2,0x2B},{0xE5,0xC8,0x1F,0x2C},{0x05,0x07,0x1D,0x2C},{0x75,0x4E,0xE3,0x2B},{0xE5,0x14,0x17,0x2C},{0xC5,0x90,0x14,0x2C},{0x35,0xD6,0x14,0x2C},{0x7C,0xD6,0x6E,0x85},{0x06,0xDC,0x6E,0x85},{0x41,0xE6,0x6F,0x85},{0x00,0x36,0x6F,0x85},{0xDA,0x59,0x26,0x3B},{0xAB,0x3A,0x26,0x3B},{0xFC,0xFD,0x6F,0x85},{0xDD,0xC5,0x6F,0x85},{0x76,0xBA,0x6E,0x85},{0xA7,0x39,0x6F,0x85},{0x57,0x73,0x6F,0x85},{0x20,0xF6,0x6F,0x85},{0x60,0xF8,0x6F,0x85},{0xFC,0x41,0x6F,0x85},{0x26,0xEB,0x6F,0x85},{0x60,0x1D,0x27,0x3B},{0xD1,0x3F,0x6F,0x85},{0xB6,0x1D,0x27,0x3B},{0x2E,0x01,0x6F,0x85},{0x0A,0

```
x7E,0x6F,0x85},{0x90,0xA1,0x26,0x3B},{0xF1,0x38,0x6F,0x85},{0x1D,0x85,0x26,0x3
B},{0xCD,0xBE,0x6E,0x85},{0x4D,0x7C,0x6F,0x85},{0xCB,0xB2,0x6F,0x85},{0x94,0x
65,0x6F,0x85},{0x13,0x68,0x26,0x3B},{0xDC,0x3F,0x6F,0x85},{0x52,0x09,0x26,0x3B}
};
//DEFINE CARD LOOP ENDS


int N=120;
byte readcard[4];


void setup()
{//SETUP LOOP BEGINS
  Serial.begin(9600);
  XBee.begin(9600);
  SPI.begin();
  rc.PCD_Init();
  rc.PCD_DumpVersionToSerial();
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);
  digitalWrite(4,LOW);
  //{//DISPLAY OF AUTHORISED CARDS INFORMATION STARTS
  //Serial.println(F("THE AUTHORISED CARDS ARE"));
    //for(int i=0;i<N;i++)
```

```
    //{//SETUP FOR LOOP1 STARTS
      //Serial.print(i+1);
      //Serial.print("  ");
      //for (int j=0;j<4;j++)
      //{//SETUP FOR LOOP2 STARTS
        //Serial.print(defcard[i][j],HEX);
      //}//SETUP FOR LOOP2 ENDS
      //Serial.println("");
    //}//SETUP FOR LOOP1 ENDS
    //Serial.println("");
  //}//DISPLAY OF AUTHORISED CARDS INFORMATION ENDS
}//SETUP LOOP ENDS

void loop()
{//VOID LOOP STARTS
  readsuccess=getid();
  if(readsuccess)
  {//IF READSUCCESS LOOP STARTS
    int match=0;
    for (int i=0;i<N;i++)
    {//FOR LOOP1 STARTS
      if(!memcmp(readcard,defcard[i],4))
```

```
  {//IF LOOP OF FOR LOOP1 STARTS
    match++;
  }//IF LOOP OF FOR LOOP1 ENDS
}//FOR LOOP1 ENDS
if (match)
{//MATCH IF LOOP STARTS
  x=0;
  if(digitalRead(PIR)==HIGH)
  {//PIR HIGH DETECTION IF LOOP STARTS
    if(digitalRead(PIR1)==HIGH)
    {//PIR1 HIGH DETECTION IF LOOP STARTS
      {x++;}
       Serial.println(F("Person has entered"));
      if (x==1)
      {//WHEN X=1 IF LOOP STARTS
        authorised();
      }//WHEN X=1 IF LOOP ENDS
      else if(x>1)
      {//ELSE IF LOOP STARTS
        Serial.println(F("MORE MEMBERS ARE ENTERING"));
        digitalWrite(4,HIGH);
        digitalWrite(5,HIGH);
```

```
            delay(1500);
            digitalWrite(4,LOW);
            digitalWrite(5,LOW);
            delay(500);
          }//ELSE IF LOOP ENDS
        }//PIR1 HIGH DETECTION IF LOOP ENDS
      }//PIR HIGH DETECTION IF LOOP ENDS
}//MATCH IF LOOP ENDS
          else
          {//ELSE LOOP STARTS
            unauthorised();
            }//ELSE LOOP ENDS
  }//IF READSUCCESS LOOP ENDS
}//VOID LOOP ENDS



int getid()
{//GETID LOOP STARTS
  if (!rc.PICC_IsNewCardPresent())
  {//IF LOOP1 STARTS
    return 0;
  }//IF LOOP1 ENDS
```

```
  if (!rc.PICC_ReadCardSerial())
  {//IF LOOP2 STARTS
    return 0;
  }//IF LOOP2 ENDS

  for (int i=0;i<4;i++)
  {//GETID FOR LOOP1 STARTS
    readcard[i]=rc.uid.uidByte[i];
  }//GETID FOR LOOP1 STARTS
  rc.PICC_HaltA();
  return 1;
}//GETID LOOP ENDS

void authorised()
{//VOID AUTHORISED STARTS
  Serial.println(F("THE UID OF THE SCANNED AUTHORISED CARD IS"));
  for (int i=0;i<4;i++)
  {//FOR LOOP1 STARTS
    readcard[i]=rc.uid.uidByte[i];
    Serial.print(readcard[i],HEX);
  }//FOR LOOP1 STARTS
  Serial.println("");
```

```
        Serial.println(F("CARD AUTHORISED"));
        digitalWrite(6,HIGH);
        disp();
        digitalWrite(4,LOW);
        delay(1500);
        digitalWrite(6,LOW);
        delay(500);
}//VOID AUTHORISED ENDS

void unauthorised()
{//VOID UNAUTHORISED STARTS
  Serial.println(F("THE UID OF THE SCANNED UNAUTHORISED CARD IS"));
  for (int i=0;i<4;i++)
  {//FOR LOOP1 STARTS
    readcard[i]=rc.uid.uidByte[i];
    Serial.print(readcard[i],HEX);
  }//FOR LOOP1 ENDS
  Serial.println("");
        Serial.println(F("CARD NOT AUTHORISED"));
        digitalWrite(4,HIGH);
        digitalWrite(5,HIGH);
        delay(1500);
```

```
        digitalWrite(4,LOW);
        digitalWrite(5,LOW);
        delay(500);
}//VOID UNAUTHORISED ENDS



void disp()
{//DISP STARTS
  String content= "";
  byte letter;
  for (byte i = 0; i < rc.uid.size; i++)
{//FOR BYTE STARTS
    content.concat(String(rc.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(rc.uid.uidByte[i], HEX));
  }//FOR BYTE ENDS

  content.toUpperCase();
  if (content.substring(1) == "05 EF E7 2B")
  {
    Serial.println(F("Name of the Authorised Person is Ghazala Anjum \nRoll Number is 1604-13-735-001 \nCard Link is AAAAAAAA"));
    XBee.print("AAAAAAAA");    Serial.println("");
  }
```

```
 if (content.substring(1) == "65 3D ED 2B")

 {

  Serial.println(F("Name of the Authorised Person is Samiya Firdose \nRoll Number is 1604-13-735-002 \nCard Link is AAAAAAAB"));

  XBee.print("AAAAAAAB");    Serial.println("");

 }

  if (content.substring(1) == "24 0A B6 5B")

 {

  Serial.println(F("Name of the Authorised Person is Arfa Shazli \nRoll Number is 1604-13-735-003 \nCard Link is AAAAAAAC"));

  XBee.print("AAAAAAAC");    Serial.println("");

 }

  if (content.substring(1) == "94 CD B3 5B")

 {

  Serial.println(F("Name of the Authorised Person is Syed Rizwan Hashmi \nRoll Number is 1604-13-735-004 \nCard Link is AAAAAAAD"));

  XBee.print("AAAAAAAD");    Serial.println("");

 }

  if (content.substring(1) == "35 89 15 2C")

 {

  Serial.println(F("Name of the Authorised Person is Amena Najeeb \nRoll Number is 1604-13-735-005 \nCard Link is AAAAAAAE"));

  XBee.print("AAAAAAAE");    Serial.println("");
```

```
 }
 if (content.substring(1) == "95 80 C5 2A")
 {
  Serial.println(F("Name of the Authorised Person is Amreen Sultana \nRoll Number is 1604-13-735-006 \nCard Link is AAAAAAAF"));
  XBee.print("AAAAAAAF");    Serial.println("");
 }
 if (content.substring(1) == "D5 F6 1A 2C")
 {
  Serial.println(F("Name of the Authorised Person is Samreen Sultana \nRoll Number is 1604-13-735-007 \nCard Link is AAAAAAAG"));
  XBee.print("AAAAAAAG");    Serial.println("");
 }
 if (content.substring(1) == "C5 8A 1B 2C")
 {
  Serial.println(F("Name of the Authorised Person is Syeda Rumaan Hussaini \nRoll Number is 1604-13-735-008 \nCard Link is AAAAAAAH"));
  XBee.print("AAAAAAAH");    Serial.println("");
 }
 if (content.substring(1) == "05 26 1F 2C")
 {
  Serial.println(F("Name of the Authorised Person is Nisma Imroze \nRoll Number is 1604-13-735-009 \nCard Link is AAAAAAAI"));
```

```
  XBee.print("AAAAAAAI");    Serial.println("");
 }
 if (content.substring(1) == "15 DE 16 2C")
 {
  Serial.println(F("Name of the Authorised Person is Saadiya Fathima \nRoll Number is 1604-13-735-010 \nCard Link is AAAAAAAJ"));
  XBee.print("AAAAAAAJ");    Serial.println("");
 }
 if (content.substring(1) == "F4 66 B6 5B")
 {
  Serial.println(F("Name of the Authorised Person is Asmaa Mohammed Afzal \nRoll Number is 1604-13-735-011 \nCard Link is AAAAAAAK"));
  XBee.print("AAAAAAAK");    Serial.println("");
 }
 if (content.substring(1) == "94 46 B8 5B")
 {
  Serial.println(F("Name of the Authorised Person is Mahrukh S Hasan \nRoll Number is 1604-13-735-012 \nCard Link is AAAAAAAL"));
  XBee.print("AAAAAAAL");    Serial.println("");
 }
 if (content.substring(1) == "64 12 B3 5B")
 {
  Serial.println(F("Name of the Authorised Person is Maru Sai Ramya Reddy \nRoll Number
```

```
is 1604-13-735-013 \nCard Link is AAAAAAAM"));
    XBee.print("AAAAAAAM");    Serial.println("");
  }
   if (content.substring(1) == "B4 A0 B2 5B")
  {
    Serial.println(F("Name of the Authorised Person is Syeda Nemath Unnisa \nRoll Number
is 1604-13-735-014 \nCard Link is AAAAAAAN"));
    XBee.print("AAAAAAAN");    Serial.println("");
  }
   if (content.substring(1) == "F4 A2 B4 5B")
  {
    Serial.println(F("Name of the Authorised Person is Erum Abdul Razzak \nRoll Number is
1604-13-735-015 \nCard Link is AAAAAAAO"));
    XBee.print("AAAAAAAO");    Serial.println("");
  }
 if (content.substring(1) == "E5 09 14 2C")
  {
    Serial.println(F("Name of the Authorised Person is Shaik Sameera \nRoll Number is 1604-
13-735-016 \nCard Link is AAAAAAAP"));
    XBee.print("AAAAAAAP");    Serial.println("");
  }
   if (content.substring(1) == "35 BC 1D 2C")
  {
```

```
  Serial.println(F("Name of the Authorised Person is Ksheera Surapaneni \nRoll Number is 1604-13-735-017 \nCard Link is AAAAAAAQ"));
  XBee.print("AAAAAAAQ");    Serial.println("");
 }
 if (content.substring(1) == "A5 6C 1D 2C")
 {
  Serial.println(F("Name of the Authorised Person is Rafi Imran \nRoll Number is 1604-13-735-018 \nCard Link is AAAAAAAR"));
  XBee.print("AAAAAAAR");    Serial.println("");
 }
 if (content.substring(1) == "84 FD B9 5B")
 {
  Serial.println(F("Name of the Authorised Person is Mohaammad Kashif Moiz \nRoll Number is 1604-13-735-019 \nCard Link is AAAAAAAS"));
  XBee.print("AAAAAAAS");    Serial.println("");
 }
 if (content.substring(1) == "54 74 BB 5B")
 {
  Serial.println(F("Name of the Authorised Person is Shaik Ahamed Ali Riyaz \nRoll Number is 1604-13-735-020 \nCard Link is AAAAAAAT"));
  XBee.print("AAAAAAAT");    Serial.println("");
 }
 if (content.substring(1) == "65 C0 D1 2B")
```

```
{
  Serial.println(F("Name of the Authorised Person is Amit Sharma \nRoll Number is 1604-13-735-021 \nCard Link is AAAAAAAU"));
  XBee.print("AAAAAAAU");    Serial.println("");
}
 if (content.substring(1) == "45 20 18 2C")
{
  Serial.println(F("Name of the Authorised Person is Humair Ahmed Khan Aqib \nRoll Number is 1604-13-735-022 \nCard Link is AAAAAAAV"));
  XBee.print("AAAAAAAV");    Serial.println("");
}
 if (content.substring(1) == "95 24 DA 2B")
{
  Serial.println(F("Name of the Authorised Person is Mohammed Azeeth \nRoll Number is 1604-13-735-023 \nCard Link is AAAAAAAW"));
  XBee.print("AAAAAAAW");    Serial.println("");
}
 if (content.substring(1) == "75 48 DB 2B")
{
  Serial.println(F("Name of the Authorised Person is Shaik Pasha \nRoll Number is 1604-13-735-024 \nCard Link is AAAAAAAX"));
  XBee.print("AAAAAAAX");     Serial.println("");
}
```

```
 if (content.substring(1) == "25 19 D8 2B")

 {

  Serial.println(F("Name of the Authorised Person is Azher Rashid Hussain \nRoll Number is 1604-13-735-025 \nCard Link is AAAAAAAY"));

  XBee.print("AAAAAAAY");     Serial.println("");

 }

 if (content.substring(1) == "75 E8 D0 2B")

 {

  Serial.println(F("Name of the Authorised Person is Sabeel Ahmed Syed \nRoll Number is 1604-13-735-026 \nCard Link is AAAAAAAZ"));

  XBee.print("AAAAAAAZ");     Serial.println("");

 }

 if (content.substring(1) == "35 4A D6 2B")

 {

  Serial.println(F("Name of the Authorised Person is Mohd Farooq \nRoll Number is 1604-13-735-027 \nCard Link is AAAAAABA"));

  XBee.print("AAAAAABA");     Serial.println("");

 }

 if (content.substring(1) == "B5 E7 D7 2B")

 {

  Serial.println(F("Name of the Authorised Person is Syed Kashif \nRoll Number is 1604-13-735-028 \nCard Link is AAAAAABB"));

  XBee.print("AAAAAABB");     Serial.println("");
```

```
  }
   if (content.substring(1) == "F5 FC D5 2B")
  {
    Serial.println(F("Name of the Authorised Person is K.Sai Nithin Bharadwaj \nRoll Number is 1604-13-735-029 \nCard Link is AAAAAABC"));
    XBee.print("AAAAAABC");     Serial.println("");
  }
   if (content.substring(1) == "65 19 D0 2B")
  {
    Serial.println(F("Name of the Authorised Person is Shaik Sthanka Babu \nRoll Number is 1604-13-735-030 \nCard Link is AAAAAABD"));
    XBee.print("AAAAAABD");     Serial.println("");
  }
  if (content.substring(1) == "75 9B D3 2B")
  {
    Serial.println(F("Name of the Authorised Person is Mohammed Abdul Aziz \nRoll Number is 1604-13-735-031 \nCard Link is AAAAAABE"));
    XBee.print("AAAAAABE");     Serial.println("");
  }
   if (content.substring(1) == "94 4E B9 5B")
  {
    Serial.println(F("Name of the Authorised Person is Mohammed Afroz \nRoll Number is 1604-13-735-032 \nCard Link is AAAAAABF"));
```

```
  XBee.print("AAAAAABF");     Serial.println("");
 }
 if (content.substring(1) == "34 C0 B5 5B")
 {
  Serial.println(F("Name of the Authorised Person is Syed Luqman \nRoll Number is 1604-13-735-034 \nCard Link is AAAAAABG"));
  XBee.print("AAAAAABG");     Serial.println("");
 }
 if (content.substring(1) == "14 AF B8 5B")
 {
  Serial.println(F("Name of the Authorised Person is Huzaifa Nayeem \nRoll Number is 1604-13-735-036 \nCard Link is AAAAAABH"));
  XBee.print("AAAAAABH");     Serial.println("");
 }
 if (content.substring(1) == "D4 61 B3 5B")
 {
  Serial.println(F("Name of the Authorised Person is J.Manav Reddy \nRoll Number is 1604-13-735-037 \nCard Link is AAAAAABI"));
  XBee.print("AAAAAABI");     Serial.println("");
 }
 if (content.substring(1) == "B4 C7 BE 5B")
 {
  Serial.println(F("Name of the Authorised Person is Syed Abdul Hannan \nRoll Number is
```

```
1604-13-735-038 \nCard Link is AAAAAABJ"));
   XBee.print("AAAAAABJ");     Serial.println("");
 }
  if (content.substring(1) == "75 EE D2 2B")
 {
   Serial.println(F("Name of the Authorised Person is Md.Yousuf Ahmed \nRoll Number is
1604-13-735-039 \nCard Link is AAAAAABK"));
   XBee.print("AAAAAABK");     Serial.println("");
 }
  if (content.substring(1) == "F5 02 19 2C")
 {
   Serial.println(F("Name of the Authorised Person is Shaik Rahmath \nRoll Number is 1604-
13-735-040 \nCard Link is AAAAAABL"));
   XBee.print("AAAAAABL");     Serial.println("");
 }
  if (content.substring(1) == "25 AB 1B 2C")
 {
   Serial.println(F("Name of the Authorised Person is Syed Rizwan \nRoll Number is 1604-
13-735-042 \nCard Link is AAAAAABM"));
   XBee.print("AAAAAABM");     Serial.println("");
 }
  if (content.substring(1) == "55 41 1B 2C")
 {
```

```
  Serial.println(F("Name of the Authorised Person is Mohd Akber Ali \nRoll Number is 1604-13-735-043 \nCard Link is AAAAAABN"));
  XBee.print("AAAAAABN");     Serial.println("");
 }
  if (content.substring(1) == "55 52 D4 2B")
 {
  Serial.println(F("Name of the Authorised Person is Syed Faizaan Mustafa \nRoll Number is 1604-13-735-044 \nCard Link is AAAAAABO"));
  XBee.print("AAAAAABO");     Serial.println("");
 }
  if (content.substring(1) == "05 B4 D0 2B")
 {
  Serial.println(F("Name of the Authorised Person is B.Shafat Mohammed Khan \nRoll Number is 1604-13-735-045 \nCard Link is AAAAAABP"));
  XBee.print("AAAAAABP");     Serial.println("");
 }
  if (content.substring(1) == "34 97 B3 5B")
 {
  Serial.println(F("Name of the Authorised Person is Mohd.Ehtheshamuddin \nRoll Number is 1604-13-735-046 \nCard Link is AAAAAABQ"));
  XBee.print("AAAAAABQ");     Serial.println("");
 }
  if (content.substring(1) == "F4 13 B5 5B")
```

```
{
  Serial.println(F("Name of the Authorised Person is G.Vineeth \nRoll Number is 1604-13-735-047 \nCard Link is AAAAAABR"));
  XBee.print("AAAAAABR");     Serial.println("");
}
 if (content.substring(1) == "44 2D B5 5B")
{
  Serial.println(F("Name of the Authorised Person is Mohammad Mazharuddin \nRoll Number is 1604-13-735-048 \nCard Link is AAAAAABS"));
  XBee.print("AAAAAABS");     Serial.println("");
}
if (content.substring(1) == "A4 34 B5 5B")
{
  Serial.println(F("Name of the Authorised Person is Mohammed Imran Afzal \nRoll Number is 1604-13-735-049 \nCard Link is AAAAAABT"));
  XBee.print("AAAAAABT");     Serial.println("");
}
 if (content.substring(1) == "45 0F EF 2B")
{
  Serial.println(F("Name of the Authorised Person is Mohammed Abdul Samad \nRoll Number is 1604-13-735-050 \nCard Link is AAAAAABU"));
  XBee.print("AAAAAABU");     Serial.println("");
}
```

```
 if (content.substring(1) == "B4 EE BD 5B")

 {

  Serial.println(F("Name of the Authorised Person is Mohammed Obeid \nRoll Number is 1604-13-735-051 \nCard Link is AAAAAABV"));

  XBee.print("AAAAAABV");    Serial.println("");

 }

 if (content.substring(1) == "14 85 B3 5B")

 {

  Serial.println(F("Name of the Authorised Person is Mohd Khaja \nRoll Number is 1604-13-735-052 \nCard Link is AAAAAABW"));

  XBee.print("AAAAAABW");    Serial.println("");

 }

 if (content.substring(1) == "74 75 B2 5B")

 {

  Serial.println(F("Name of the Authorised Person is B.Nihal Reddy \nRoll Number is 1604-13-735-053 \nCard Link is AAAAAABX"));

  XBee.print("AAAAAABX");    Serial.println("");

 }

 if (content.substring(1) == "54 6D BC 5B")

 {

  Serial.println(F("Name of the Authorised Person is Mohammed Abdul Raheem Khan \nRoll Number is 1604-13-735-054 \nCard Link is AAAAAABY"));

  XBee.print("AAAAAABY");    Serial.println("");
```

```
}
 if (content.substring(1) == "75 84 AB 2A")
{
  Serial.println(F("Name of the Authorised Person is Syed Abdul Wahab \nRoll Number is 1604-13-735-055 \nCard Link is AAAAAABZ"));
  XBee.print("AAAAAABZ");     Serial.println("");
}
 if (content.substring(1) == "C5 21 E3 2B")
{
  Serial.println(F("Name of the Authorised Person is Rashid Hussain \nRoll Number is 1604-13-735-056 \nCard Link is AAAAAACA"));
  XBee.print("AAAAAACA");     Serial.println("");
}
 if (content.substring(1) == "55 EA C5 2A")
{
  Serial.println(F("Name of the Authorised Person is Battula Geetha Praseeda \nRoll Number is 1604-13-735-057 \nCard Link is AAAAAACB"));
  XBee.print("AAAAAACB");     Serial.println("");
}
 if (content.substring(1) == "35 93 DD 2B")
{
  Serial.println(F("Name of the Authorised Person is M.Sindhu \nRoll Number is 1604-13-735-059 \nCard Link is AAAAAACC"));
```

```
  XBee.print("AAAAAACC");     Serial.println("");
 }
 if (content.substring(1) == "45 8B EB 2B")
 {
  Serial.println(F("Name of the Authorised Person is Alapati Sita Hasita \nRoll Number is 1604-13-735-060 \nCard Link is AAAAAACD"));
  XBee.print("AAAAAACD");     Serial.println("");
 }
 if (content.substring(1) == "E4 CA B9 5B")
 {
  Serial.println(F("Name of the Authorised Person is Ashjan Khan \nRoll Number is 1604-13-735-061 \nCard Link is AAAAAACE"));
  XBee.print("AAAAAACE");     Serial.println("");
 }
 if (content.substring(1) == "35 A7 EB 2B")
 {
  Serial.println(F("Name of the Authorised Person is Sumaiyya Raahman \nRoll Number is 1604-13-735-062 \nCard Link is AAAAAACF"));
  XBee.print("AAAAAACF");     Serial.println("");
 }
 if (content.substring(1) == "05 D1 F1 2B")
 {
  Serial.println(F("Name of the Authorised Person is Shaik Asma \nRoll Number is 1604-
```

```
13-735-063 \nCard Link is AAAAAACG"));
   XBee.print("AAAAAACG");     Serial.println("");
  }
  if (content.substring(1) == "D4 57 B0 5B")
  {
   Serial.println(F("Name of the Authorised Person is Mohammed Salman Ahmed \nRoll Number is 1604-13-735-064 \nCard Link is AAAAAACH"));
   XBee.print("AAAAAACH");     Serial.println("");
  }
  if (content.substring(1) == "F5 8B E4 2B")
  {
   Serial.println(F("Name of the Authorised Person is Usamah Ahmed Khan \nRoll Number is 1604-13-735-065 \nCard Link is BBBBBBBA"));
   XBee.print("BBBBBBBA");     Serial.println("");
  }
  if (content.substring(1) == "C4 2C BD 5B")
  {
   Serial.println(F("Name of the Authorised Person is Mohammed Hesham Ahmed \nRoll Number is 1604-13-735-066 \nCard Link is BBBBBBBB"));
   XBee.print("BBBBBBBB");     Serial.println("");
  }
  if (content.substring(1) == "B5 AE A9 2A")
  {
```

```
  Serial.println(F("Name of the Authorised Person is Sarah Semeen \nRoll Number is 1604-13-735-067 \nCard Link is BBBBBBBC"));
  XBee.print("BBBBBBBC");     Serial.println("");
 }
 if (content.substring(1) == "25 7A AF 2A")
 {
  Serial.println(F("Name of the Authorised Person is Fariha Amreen \nRoll Number is 1604-13-735-068 \nCard Link is BBBBBBBD"));
  XBee.print("BBBBBBBD");     Serial.println("");
 }
 if (content.substring(1) == "74 EF A0 5B")
 {
  Serial.println(F("Name of the Authorised Person is D.Aishwarya \nRoll Number is 1604-13-735-069 \nCard Link is BBBBBBBE"));
  XBee.print("BBBBBBBE");     Serial.println("");
 }
if (content.substring(1) == "84 4B 9F 5B")
 {
  Serial.println(F("Name of the Authorised Person is Nazia Sultana \nRoll Number is 1604-13-735-070 \nCard Link is BBBBBBBF"));
  XBee.print("BBBBBBBF");     Serial.println("");
 }
 if (content.substring(1) == "84 09 9F 5B")
```

```
{
  Serial.println(F("Name of the Authorised Person is Syeda Maseera \nRoll Number is 1604-13-735-071 \nCard Link is BBBBBBBG"));
  XBee.print("BBBBBBBG");     Serial.println("");
}
 if (content.substring(1) == "C4 00 99 5B")
{
  Serial.println(F("Name of the Authorised Person is Fizza Moosavi \nRoll Number is 1604-13-735-072 \nCard Link is BBBBBBBH"));
  XBee.print("BBBBBBBH");     Serial.println("");
}
 if (content.substring(1) == "94 FB 9D 5B")
{
  Serial.println(F("Name of the Authorised Person is Fathima Liyaqath \nRoll Number is 1604-13-735-073 \nCard Link is BBBBBBBI"));
  XBee.print("BBBBBBBI");     Serial.println("");
}
 if (content.substring(1) == "54 C5 9A 5B")
{
  Serial.println(F("Name of the Authorised Person is Asma \nRoll Number is 1604-13-735-074 \nCard Link is BBBBBBBJ"));
  XBee.print("BBBBBBBJ");     Serial.println("");
}
```

```
 if (content.substring(1) == "95 77 E4 2B")
 {
  Serial.println(F("Name of the Authorised Person is Mohammed Rukhaya Siddiqua \nRoll Number is 1604-13-735-075 \nCard Link is BBBBBBBK"));
  XBee.print("BBBBBBBK");     Serial.println("");
 }
  if (content.substring(1) == "65 A9 13 2C")
 {
  Serial.println(F("Name of the Authorised Person is Hafsa Fathima \nRoll Number is 1604-13-735-077 \nCard Link is BBBBBBBL"));
  XBee.print("BBBBBBBL");     Serial.println("");
 }
  if (content.substring(1) == "E5 D8 D2 2B")
 {
  Serial.println(F("Name of the Authorised Person is Shaik Shereen \nRoll Number is 1604-13-735-078 \nCard Link is BBBBBBBM"));
  XBee.print("BBBBBBBM");     Serial.println("");
 }
  if (content.substring(1) == "D5 44 14 2C")
 {
  Serial.println(F("Name of the Authorised Person is Saniya Ansari \nRoll Number is 1604-13-735-079 \nCard Link is BBBBBBBN"));
  XBee.print("BBBBBBBN");     Serial.println("");
```

```
}
 if (content.substring(1) == "B5 78 CF 2B")
{
  Serial.println(F("Name of the Authorised Person is Sana Sabahath \nRoll Number is 1604-13-735-081 \nCard Link is BBBBBBBO"));
  XBee.print("BBBBBBBO");     Serial.println("");
}
if (content.substring(1) == "15 94 D7 2B")
{
  Serial.println(F("Name of the Authorised Person is Firdous \nRoll Number is 1604-13-735-082 \nCard Link is BBBBBBBP"));
  XBee.print("BBBBBBBP");     Serial.println("");
}
 if (content.substring(1) == "D5 73 F4 2B")
{
  Serial.println(F("Name of the Authorised Person is Mohd Owais Ahmed Fahad \nRoll Number is 1604-13-735-083 \nCard Link is BBBBBBBQ"));
  XBee.print("BBBBBBBQ");     Serial.println("");
}
 if (content.substring(1) == "A5 CA D4 2B")
{
  Serial.println(F("Name of the Authorised Person is Samreen Begum \nRoll Number is 1604-13-735-084 \nCard Link is BBBBBBBR"));
```

```
  XBee.print("BBBBBBBR");     Serial.println("");
 }
 if (content.substring(1) == "C5 83 D5 2B")
 {
  Serial.println(F("Name of the Authorised Person is Syed Hafeez Pasha \nRoll Number is 1604-13-735-085 \nCard Link is BBBBBBBS"));
  XBee.print("BBBBBBBS");     Serial.println("");
 }
 if (content.substring(1) == "D5 43 1F 2C")
 {
  Serial.println(F("Name of the Authorised Person is Syed Latheef \nRoll Number is 1604-13-735-086 \nCard Link is BBBBBBBT"));
  XBee.print("BBBBBBBT");     Serial.println("");
 }
 if (content.substring(1) == "55 3F 1E 2C")
 {
  Serial.println(F("Name of the Authorised Person is Yemineni Shiva Teja \nRoll Number is 1604-13-735-087 \nCard Link is BBBBBBBU"));
  XBee.print("BBBBBBBU");     Serial.println("");
 }
 if (content.substring(1) == "55 2D 17 2C")
 {
  Serial.println(F("Name of the Authorised Person is Mohd Shahid Zeeshan \nRoll Number
```

```
is 1604-13-735-088 \nCard Link is BBBBBBBV"));
    XBee.print("BBBBBBBV");     Serial.println("");
  }
   if (content.substring(1) == "85 16 18 2C")
  {
    Serial.println(F("Name of the Authorised Person is Mohammed Sohail S \nRoll Number is 1604-13-735-089 \nCard Link is BBBBBBBW"));
    XBee.print("BBBBBBBW");     Serial.println("");
  }
   if (content.substring(1) == "B5 91 E2 2B")
  {
    Serial.println(F("Name of the Authorised Person is Mohd Aqeel Shareef \nRoll Number is 1604-13-735-090 \nCard Link is BBBBBBBX"));
    XBee.print("BBBBBBBX");     Serial.println("");
  }
  if (content.substring(1) == "E5 C8 1F 2C")
  {
    Serial.println(F("Name of the Authorised Person is Rahul Raj Chopra \nRoll Number is 1604-13-735-091 \nCard Link is BBBBBBBY"));
    XBee.print("BBBBBBBY");     Serial.println("");
  }
 if (content.substring(1) == "05 07 1D 2C")
  {
```

```
  Serial.println(F("Name of the Authorised Person is Mohd Shabbiruddin \nRoll Number is 1604-13-735-092 \nCard Link is BBBBBBBZ"));
   XBee.print("BBBBBBBZ");     Serial.println("");
 }
  if (content.substring(1) == "75 4E E3 2B")
 {
  Serial.println(F("Name of the Authorised Person is Mohammed Sohail \nRoll Number is 1604-13-735-093 \nCard Link is BBBBBBCA"));
   XBee.print("BBBBBBCA");     Serial.println("");
 }
  if (content.substring(1) == "E5 14 17 2C")
 {
  Serial.println(F("Name of the Authorised Person is Mohd Abdul Hasan Imroze \nRoll Number is 1604-13-735-094 \nCard Link is BBBBBBCB"));
   XBee.print("BBBBBBCB");     Serial.println("");
 }
  if (content.substring(1) == "C5 90 14 2C")
 {
  Serial.println(F("Name of the Authorised Person is Z.Syed Noman \nRoll Number is 1604-13-735-095 \nCard Link is BBBBBBCC"));
   XBee.print("BBBBBBCC");     Serial.println("");
 }
  if (content.substring(1) == "35 D6 14 2C")
```

```
{
  Serial.println(F("Name of the Authorised Person is Mohammed Saif Khan \nRoll Number is 1604-13-735-096 \nCard Link is BBBBBBCD"));
  XBee.print("BBBBBBCD");     Serial.println("");
}
 if (content.substring(1) == "7C D6 6E 85")
{
  Serial.println(F("Name of the Authorised Person is Abdul Moiz Md \nRoll Number is 1604-13-735-097 \nCard Link is BBBBBBCE"));
  XBee.print("BBBBBBCE");     Serial.println("");
}
 if (content.substring(1) == "06 DC 6E 85")
{
  Serial.println(F("Name of the Authorised Person is Mirza Razaa Ali Baig \nRoll Number is 1604-13-735-098 \nCard Link is BBBBBBCF"));
  XBee.print("BBBBBBCF");     Serial.println("");
}
 if (content.substring(1) == "41 E6 6F 85")
{
  Serial.println(F("Name of the Authorised Person is Sameer Khan \nRoll Number is 1604-13-735-100 \nCard Link is BBBBBBCG"));
  XBee.print("BBBBBBCG");     Serial.println("");
}
```

```
 if (content.substring(1) == "00 36 6F 85")

 {

  Serial.println(F("Name of the Authorised Person is Mohd Ershad \nRoll Number is 1604-13-735-101 \nCard Link is BBBBBBCH"));

  XBee.print("BBBBBBCH");     Serial.println("");

 }

 if (content.substring(1) == "DA 59 26 3B")

 {

  Serial.println(F("Name of the Authorised Person is Imran Asad \nRoll Number is 1604-13-735-102 \nCard Link is BBBBBBCI"));

  XBee.print("BBBBBBCI");     Serial.println("");

 }

if (content.substring(1) == "AB 3A 26 3B")

 {

  Serial.println(F("Name of the Authorised Person is Zeeshan Ahmed \nRoll Number is 1604-13-735-103 \nCard Link is BBBBBBCJ"));

  XBee.print("BBBBBBCJ");     Serial.println("");

 }

 if (content.substring(1) == "FC FD 6F 85")

 {

  Serial.println(F("Name of the Authorised Person is Aiman Mohiddin Mohammed \nRoll Number is 1604-13-735-104 \nCard Link is BBBBBBCK"));

  XBee.print("BBBBBBCK");     Serial.println("");
```

```
  }

   if (content.substring(1) == "DD C5 6F 85")

  {

    Serial.println(F("Name of the Authorised Person is Ali Osman Khaja \nRoll Number is 1604-13-735-105 \nCard Link is BBBBBBCL"));

    XBee.print("BBBBBBCL");     Serial.println("");

  }

   if (content.substring(1) == "76 BA 6E 85")

  {

    Serial.println(F("Name of the Authorised Person is Mohammed Muneer Akram \nRoll Number is 1604-13-735-106 \nCard Link is BBBBBBCM"));

    XBee.print("BBBBBBCM");     Serial.println("");

  }

   if (content.substring(1) == "A7 39 6F 85")

  {

    Serial.println(F("Name of the Authorised Person is Mohammad Fauzan \nRoll Number is 1604-13-735-107 \nCard Link is BBBBBBCN"));

    XBee.print("BBBBBBCN");     Serial.println("");

  }

  if (content.substring(1) == "57 73 6F 85")

  {

    Serial.println(F("Name of the Authorised Person is Khaja Moizuddin \nRoll Number is 1604-13-735-108 \nCard Link is BBBBBBCO"));
```

```
  XBee.print("BBBBBBCO");     Serial.println("");
 }
 if (content.substring(1) == "20 F6 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Mohammed Abdul Khadeer \nRoll Number is 1604-13-735-110 \nCard Link is BBBBBBCP"));
  XBee.print("BBBBBBCP");     Serial.println("");
 }
 if (content.substring(1) == "60 F8 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Md Abrar Ali \nRoll Number is 1604-13-735-111 \nCard Link is BBBBBBCQ"));
  XBee.print("BBBBBBCQ");     Serial.println("");
 }
 if (content.substring(1) == "FC 41 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Ayub Shareef \nRoll Number is 1604-13-735-112 \nCard Link is BBBBBBCR"));
  XBee.print("BBBBBBCR");     Serial.println("");
 }
 if (content.substring(1) == "26 EB 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Syed Jawwad Mohiuddin \nRoll
```

```
Number is 1604-13-735-113 \nCard Link is BBBBBBCS"));
   XBee.print("BBBBBBCS");     Serial.println("");
  }
 if (content.substring(1) == "60 1D 27 3B")
  {
   Serial.println(F("Name of the Authorised Person is Nisamuddin Mohd \nRoll Number is 1604-13-735-114 \nCard Link is BBBBBBCT"));
   XBee.print("BBBBBBCT");     Serial.println("");
  }
  if (content.substring(1) == "D1 3F 6F 85")
  {
   Serial.println(F("Name of the Authorised Person is Shaikh Mohammed Simraan \nRoll Number is 1604-13-735-115 \nCard Link is BBBBBBCU"));
   XBee.print("BBBBBBCU");     Serial.println("");
  }
  if (content.substring(1) == "B6 1D 27 3B")
  {
   Serial.println(F("Name of the Authorised Person is Mohd Kashif Hussain \nRoll Number is 1604-13-735-116 \nCard Link is BBBBBBCV"));
   XBee.print("BBBBBBCV");     Serial.println("");
  }
  if (content.substring(1) == "2E 01 6F 85")
  {
```

```
  Serial.println(F("Name of the Authorised Person is Karam Ilahi Khan \nRoll Number is 1604-13-735-117 \nCard Link is BBBBBBCW"));
   XBee.print("BBBBBBCW");     Serial.println("");
 }
  if (content.substring(1) == "0A 7E 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Mohammed Khadeer \nRoll Number is 1604-13-735-118 \nCard Link is BBBBBBCX"));
   XBee.print("BBBBBBCX");     Serial.println("");
 }
 if (content.substring(1) == "90 A1 26 3B")
 {
  Serial.println(F("Name of the Authorised Person is Shaikh Zameeruddin \nRoll Number is 1604-13-735-119 \nCard Link is BBBBBBCY"));
   XBee.print("BBBBBBCY");     Serial.println("");
 }
  if (content.substring(1) == "F1 38 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Mohammed Abdullah \nRoll Number is 1604-13-735-120 \nCard Link is BBBBBBCZ"));
   XBee.print("BBBBBBCZ");     Serial.println("");
 }
  if (content.substring(1) == "1D 85 26 3B")
```

```
{
  Serial.println(F("Name of the Authorised Person is Saud Habeeb Mohammed Ba Aqeel \nRoll Number is 1604-13-735-123 \nCard Link is BBBBBBDA"));
  XBee.print("BBBBBBDA");     Serial.println("");
}
 if (content.substring(1) == "CD BE 6E 85")
{
  Serial.println(F("Name of the Authorised Person is Salam Basha Pathan \nRoll Number is 1604-13-735-124 \nCard Link is BBBBBBDB"));
  XBee.print("BBBBBBDB");     Serial.println("");
}
 if (content.substring(1) == "4D 7C 6F 85")
{
  Serial.println(F("Name of the Authorised Person is Mohammed Kazim Ahmed \nRoll Number is 1604-13-735-125 \nCard Link is BBBBBBDC"));
  XBee.print("BBBBBBDC");     Serial.println("");
}
if (content.substring(1) == "CB B2 6F 85")
{
  Serial.println(F("Name of the Authorised Person is Ibrahim Khan \nRoll Number is 1604-13-735-126 \nCard Link is BBBBBBDD"));
  XBee.print("BBBBBBDD");     Serial.println("");
}
```

```
 if (content.substring(1) == "94 65 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Syed Khalid Ashraf \nRoll Number is 1604-13-735-127 \nCard Link is BBBBBBDE"));
  XBee.print("BBBBBBDE");    Serial.println("");
 }
 if (content.substring(1) == "13 68 26 3B")
 {
  Serial.println(F("Name of the Authorised Person is Amaan Aslam Ashraf \nRoll Number is 1604-13-735-128 \nCard Link is BBBBBBDF"));
  XBee.print("BBBBBBDF");    Serial.println("");
 }
 if (content.substring(1) == "DC 3F 6F 85")
 {
  Serial.println(F("Name of the Authorised Person is Syed Shafi \nRoll Number is 1604-13-735-301 \nCard Link is BBBBBBDG"));
  XBee.print("BBBBBBDG");    Serial.println("");
 }
 if (content.substring(1) == "52 09 26 3B")
 {
  Serial.println(F("Name of the Authorised Person is Sana Noureen \nRoll Number is 1604-13-735-302 \nCard Link is BBBBBBDH"));
  XBee.print("BBBBBBDH");    Serial.println("");
```

```
  }

}//DISP ENDS
```

6.2 Arduino UNO Program

```
#include <SoftwareSerial.h>
SoftwareSerial XBee(7,8); //Rx and Tx
void setup()
{//Setup Loop Starts
  XBee.begin(9600);  //XBee/UART1/pins 0 and 1
  Serial.begin(9600);   //USB
  pinMode(12,OUTPUT);
}//Setup Loop Ends
void loop()
{//Void Loop Starts
  String content = "";  char character;
  while (XBee.available()>0)   //XBee/UART1/pins 0 and 1
  { // If data comes in from XBee, send it out to serial monitor
    character = XBee.read();
    content.concat(character);
  }
  if (content != "")
  {
    digitalWrite(12,HIGH);
    Serial.println(content);
    delay(1500);
    digitalWrite(12,LOW);
    delay(500);
  }
}//Void Loop Ends
```

6.3 NetBeans Programs

a. SimpleRead

```
import com.commondb.Common_DB;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.*; //import gnu.io.*;
import javax.comm.*;

public class SimpleRead implements Runnable, SerialPortEventListener
{
   static CommPortIdentifier portId;
   static Enumeration portList;

InputStream inputStream;
SerialPort serialPort;
Thread readThread;
byte[] readBuffer;
int z=0;
String data=null;
public  String recvdData="aa";
public static void main(String[] args)
{
   portList = CommPortIdentifier.getPortIdentifiers();
   System.out.println("portList... " + portList);
```

```
   while (portList.hasMoreElements())
   {
      portId = (CommPortIdentifier) portList.nextElement();
      if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL)
      {
         System.out.println("port identified is Serial.. "+ portId.getPortType());
         if (portId.getName().equals("COM4"))
{
            System.out.println("port identified is COM4.. "+ portId.getName());
            // if (portId.getName().equals("/dev/term/a"))
{
            SimpleRead reader = new SimpleRead();
         }
         else
         {
            System.out.println("unable to open port");
         }
      }
   }
}

public SimpleRead()
{
   try {
      System.out.println("In SimpleRead() contructor");
      serialPort = (SerialPort) portId.open("SimpleReadApp1111",2000);
      System.out.println(" Serial Port.. " + serialPort);
   }
```

```
catch (PortInUseException e)
{
   System.out.println("Port in use Exception");
}
try
{
   inputStream = serialPort.getInputStream();
   System.out.println(" Input Stream... " + inputStream);
}
catch (IOException e)
{
   System.out.println("IO Exception");
}
try
{
   serialPort.addEventListener(this);

}
catch (TooManyListenersException e)
{
   System.out.println("Tooo many Listener exception");
}
serialPort.notifyOnDataAvailable(true);
try
{
   serialPort.setSerialPortParams(9600, SerialPort.DATABITS_8,
         SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
   // no handshaking or other flow control
```

```
            serialPort.setFlowControlMode(SerialPort.FLOWCONTROL_NONE);
            // timer on any read of the serial port
            serialPort.enableReceiveTimeout(500);
            System.out.println(".....");
        }
        catch (UnsupportedCommOperationException e)
        {
            System.out.println("UnSupported comm operation");
        }
        readThread = new Thread(this);
        readThread.start();
    }

    public void run()
    {
        try
        {
            System.out.println("In run() function ");
            Thread.sleep(100);
        }
        catch (InterruptedException e)
        {
            System.out.println("Interrupted Exception in run() method");
        }
    }
```

```
public void serialEvent(SerialPortEvent event)
{
 switch (event.getEventType())
  {
   case SerialPortEvent.DATA_AVAILABLE:
   // System.out.println("DATA_AVAILABLE");

    byte[] readBuffer = new byte[1024];
    int numBytes=1024;
    data="";
    String x="";
          Connection con=null;
       Statement st=null;
       ResultSet rs=null;
    try
    {
       Thread.sleep(100);
       while (inputStream.available() > 0)
       {
          numBytes = inputStream.read(readBuffer);//count of reading data
          data=data+new String(readBuffer,0,numBytes);
          data=data.trim();
          this.recvdData+=data;
       }
       System.out.println(data);
```

```
    if(data.charAt(0)=='A')
    {
       //System.out.println("class a-" +data);
 try
     {
        Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/wahab","root",
"password");
        st=con.createStatement();
        rs=st.executeQuery("Select * from asection where Cardlink='"+data+"'");
        if(rs.next())
        {
           String n=rs.getString(2);
           int Flag=rs.getInt(5);
        System.out.println("Swiped person is-"+n);
        if(Flag==0)
        {

           Common_DB.UpdateTable("wahab","UPDATE asection set Flag='1'
WHERE Cardlink='"+data+"' ");
           int Classes=0;
           rs=st.executeQuery("Select * from areport where Cardlink='"+data+"'");
        if(rs.next())
        {
           Classes=rs.getInt(6);
           Classes=Classes+1;
```

```
            Common_DB.UpdateTable("wahab","UPDATE areport set
Classes='"+Classes+"' WHERE Cardlink='"+data+"' ");
        }
          long milliseconds = System.currentTimeMillis();
          int Intime = (int) ((milliseconds / 1000) / 60);
           System.out.println("Intime-"+Intime);
             String Outtime="NULL";
             String Dummy="NULL";
           st.executeUpdate("Insert into apartial(Cardlink,Intime,Outtime,PA)
Values('"+data+"','"+Intime+"','"+Outtime+"','"+Dummy+"')");
        }
        else if(Flag==1)
        {
           Common_DB.UpdateTable("wahab","UPDATE asection set Flag='0'
WHERE Cardlink='"+data+"' ");
           long milliseconds = System.currentTimeMillis();
          int Outtime = (int) ((milliseconds / 1000) / 60);
           System.out.println("Outtime-"+Outtime);
           String Dummy="NULL";
           rs=st.executeQuery("Select * from apartial where Cardlink='"+data+"' and
Outtime='"+Dummy+"'");
           if(rs.next())
        {
           int Intime=rs.getInt(3);
           int time=Outtime-Intime;
           System.out.println("time-"+time);
           Common_DB.UpdateTable("wahab","UPDATE apartial set
Outtime='"+Outtime+"' WHERE Cardlink='"+data+"' and Intime='"+Intime+"' ");
          String PA="NULL";
```

```
            if(time>150 && time<=180)
{
              PA="P";
            Common_DB.UpdateTable("wahab","UPDATE apartial set PA='"+PA+"'
WHERE Cardlink='"+data+"' and Intime='"+Intime+"' ");

             int Present=0;
            rs=st.executeQuery("Select * from areport where Cardlink='"+data+"'");
          if(rs.next())
          {
            Present=rs.getInt(7);
            Present=Present+1;
          }

            Common_DB.UpdateTable("wahab","UPDATE areport set
Present='"+Present+"' WHERE Cardlink='"+data+"' ");
            System.out.println("Present-"+Present);

            int c=0;
             rs=st.executeQuery("Select * from areport where Cardlink='"+data+"'");
          if(rs.next())
          {
            c=rs.getInt(6);
          }
            int Percent=Present*100/c;
            Common_DB.UpdateTable("wahab","UPDATE areport set
Percent='"+Percent+"' WHERE Cardlink='"+data+"' ");
            System.out.println("Percent-"+Percent);
             }
```

```
            else
            {
              PA="A";
            Common_DB.UpdateTable("wahab","UPDATE apartial set PA='"+PA+"'
WHERE Cardlink='"+data+"' and Intime='"+Intime+"' ");
            }
          }

          }

          }

       }
       catch(Exception ex)
       {
          ex.printStackTrace();
       }

      }
```

```
else if(data.charAt(0)=='B')
        {


        try
      {


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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/wahab","root","pa
ssword");
        st=con.createStatement();
        rs=st.executeQuery("Select * from bsection where Cardlink='"+data+"'");
        if(rs.next())
        {
           String n=rs.getString(2);
           int Flag=rs.getInt(5);
        System.out.println("Swiped person is-"+n);
        if(Flag==0){


           Common_DB.UpdateTable("wahab","UPDATE bsection set Flag='1'
WHERE Cardlink='"+data+"' ");
           int Classes=0;
           rs=st.executeQuery("Select * from breport where Cardlink='"+data+"'");
        if(rs.next())
        {
           Classes=rs.getInt(6);
           Classes=Classes+1;
```

```
            Common_DB.UpdateTable("wahab","UPDATE breport set
Classes='"+Classes+"' WHERE Cardlink='"+data+"' ");
        }
          long milliseconds = System.currentTimeMillis();
          int Intime = (int) ((milliseconds / 1000) / 60);
           System.out.println("Intime-"+Intime);

             String Outtime="NULL";
             String Dummy="NULL";
           st.executeUpdate("Insert into bpartial(Cardlink,Intime,Outtime,PA)
Values('"+data+"','"+Intime+"','"+Outtime+"','"+Dummy+"')");

        }
        else if(Flag==1)
        {
            Common_DB.UpdateTable("wahab","UPDATE bsection set Flag='0'
WHERE Cardlink='"+data+"' ");
            long milliseconds = System.currentTimeMillis();
           int Outtime = (int) ((milliseconds / 1000) / 60);
            System.out.println("Outtime-"+Outtime);
            String Dummy="NULL";
            rs=st.executeQuery("Select * from bpartial where Cardlink='"+data+"' and
Outtime='"+Dummy+"'");
            if(rs.next())
        {
            int Intime=rs.getInt(3);
            int time=Outtime-Intime;
            System.out.println("time-"+time);
            Common_DB.UpdateTable("wahab","UPDATE bpartial set
Outtime='"+Outtime+"' WHERE Cardlink='"+data+"' and Intime='"+Intime+"' ");
```

```
String PA="NULL";


if(time>150 && time<180)
{
  PA="P";
Common_DB.UpdateTable("wahab","UPDATE bpartial set PA='"+PA+"' WHERE Cardlink='"+data+"' and Intime='"+Intime+"' ");


 int Present=0;
rs=st.executeQuery("Select * from breport where Cardlink='"+data+"'");
if(rs.next())
{
Present=rs.getInt(7);
Present=Present+1;
}


Common_DB.UpdateTable("wahab","UPDATE breport set Present='"+Present+"' WHERE Cardlink='"+data+"' ");
System.out.println("Present-"+Present);


int c=0;
 rs=st.executeQuery("Select * from breport where Cardlink='"+data+"'");
if(rs.next())
{
c=rs.getInt(6);


}
```

```
            int Percent=Present*100/c;
            Common_DB.UpdateTable("wahab","UPDATE breport set
Percent='"+Percent+"' WHERE Cardlink='"+data+"' ");
            System.out.println("Percent-"+Percent);
             }
            else
            {
              PA="A";
            Common_DB.UpdateTable("wahab","UPDATE bpartial set PA='"+PA+"'
WHERE Cardlink='"+data+"' and Intime='"+Intime+"' ");

            }

          }

          }

          }

       }
       catch(Exception ex)
       {
          ex.printStackTrace();
       }

          }
    }
```

```
catch (Exception e)
    {
        System.out.println("Exception in serial event-->"+e);
    }

    break;//break from switch case 1:
    }//end of switch

}

}
```

b. ASection

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->


<html>
    <link rel="stylesheet" href="style.css" type="text/css" />
    <link rel="stylesheet" href="styles.css" type="text/css" />
<body align="center">
<div id="content">


    <h1 >Section-A attendance report </h1>


    <table border="1" cellpadding="10" class="rwd-table">


        <thead>
```

```
<tr>
    <th>Name</th>
    <th>Rollno</th>
    <th>Percent</th>

</tr>
</thead>
<tbody>


<%
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = null;
    conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/wahab", "root",
"password");
    Statement stmt = null;
    stmt = conn.createStatement();
    String query = "select * from areport";
    ResultSet rs = null;
    rs = stmt.executeQuery(query);
    while(rs.next()){
%>
<tr>
    <%



    %>
    <td><%=rs.getString("Name") %></td>
```

```
            <td><%=rs.getString("Rollno") %></td>
            <td><%=rs.getInt("Percent") %></td>

        </tr>

        <%
            }
        %>

    </tbody>
  </table>
</div>
</body>
</html>
```

c. BSection

```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.DriverManager"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <link rel="stylesheet" href="style.css" type="text/css" />
   <link rel="stylesheet" href="styles.css" type="text/css" />
<body align="center">
<div id="content">


   <h1 >Section-B attendance report </h1>

   <table border="1" cellpadding="10" class="rwd-table">

      <thead>
         <tr>
            <th>Name</th>
            <th>Rollno</th>
            <th>Percent</th>
```

```
        </tr>
    </thead>
    <tbody>


        <%
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = null;
            conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/wahab", "root",
"password");
            Statement stmt = null;
            stmt = conn.createStatement();
            String query = "select * from breport";
            ResultSet rs = null;
            rs = stmt.executeQuery(query);
            while(rs.next()){
        %>
        <tr>
            <%



            %>
            <td><%=rs.getString("Name") %></td>
            <td><%=rs.getString("Rollno") %></td>
            <td><%=rs.getInt("Percent") %></td>

        </tr>
```

```
            <%
                }
            %>

        </tbody>
    </table>
</div>
</body>
</html>
```

d. Condonation

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.DriverManager"%>

<html>
    <link rel="stylesheet" href="style.css" type="text/css" />
    <link rel="stylesheet" href="styles.css" type="text/css" />
<body align="center">
<div id="content">


    <h1 >Condonation report </h1>

    <table border="1" cellpadding="10" class="rwd-table">

        <thead>
            <tr>
                <th>S.No</th>
                <th>Name</th>
                <th>Rollno</th>
                <th>Percent</th>
```

```
    </tr>
  </thead>
  <tbody>


    <%
      Class.forName("com.mysql.jdbc.Driver");
      Connection conn = null;
      conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/wahab", "root",
"password");
      Statement stmt = null;
            Statement stmt1 = null;


      stmt = conn.createStatement();
            stmt1 = conn.createStatement();


      String query = "select * from areport where Percent<'70'";
      String query1 = "select * from breport where Percent<'70'";


      ResultSet rs = null;
      ResultSet rs1 = null;
      rs = stmt.executeQuery(query);
      rs1 = stmt1.executeQuery(query1);
      int flag=0;
      while(rs.next()){
        flag++;
    %>
```

```
<tr>
   <%



   %>
   <td><%=flag %></td>
   <td><%=rs.getString("Name") %></td>
   <td><%=rs.getString("Rollno") %></td>
   <td><%=rs.getInt("Percent") %></td>

</tr>

<%
   }
 while(rs1.next()){
flag++;
<%>
<tr>
   <%



   %>
   <td><%=flag %></td>
   <td><%=rs1.getString("Name") %></td>
   <td><%=rs1.getString("Rollno") %></td>
   <td><%=rs1.getInt("Percent") %></td>

</tr>
```

```
            <%
                }
            %>


        </tbody>
    </table>
</div>
</body>
</html>
```

e. Index

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->



<html>
   <head>
<div id="content">


      <style>
.button {
   background-color: #4CAF50; /* Green */
   border: none;
   color: white;
   padding: 15px 32px;
   text-align: center;
   text-decoration: none;
   display: inline-block;
   font-size: 16px;
```

```
    margin: 4px 2px;
    cursor: pointer;
}
.button2 {background-color: #008CBA;} /* Blue */
.button3 {background-color: #f44336;} /* Red */


</style>
</head>
<body align="center">


<img src="logo.png"  alt="Logo" >
<h2>Report</h2>


<button class="button"  ><a href="http://localhost:8080/Rfiddisplay/areport.jsp">A section</a></button>
<button class="button button2" ><a href="http://localhost:8080/Rfiddisplay/breport.jsp">B section</a></button>
<button class="button button3"><a href="http://localhost:8080/Rfiddisplay/condonation.jsp">Condonation</a></button>


</div>
</body>
</html>
```

f. Style.css

```
@import "http://fonts.googleapis.com/css?family=Montserrat:300,400,700";
.rwd-table {
  margin: 1em 0;
  min-width: 300px;
}
.rwd-table tr {
  border-top: 1px solid #ddd;
  border-bottom: 1px solid #ddd;
}
.rwd-table th {
  display: none;
}
.rwd-table td {
  display: block;
}
.rwd-table td:first-child {
  padding-top: .5em;
}
.rwd-table td:last-child {
  padding-bottom: .5em;
}
```

```
.rwd-table td:before {
  content: attr(data-th) ": ";
  font-weight: bold;
  width: 6.5em;
  display: inline-block;
}
@media (min-width: 480px) {
  .rwd-table td:before {
    display: none;
  }
}
.rwd-table th, .rwd-table td {
  text-align: left;
}
@media (min-width: 480px) {
  .rwd-table th, .rwd-table td {
    display: table-cell;
    padding: .25em .5em;
  }
  .rwd-table th:first-child, .rwd-table td:first-child {
    padding-left: 0;
  }
```

```
  .rwd-table th:last-child, .rwd-table td:last-child {
    padding-right: 0;
  }
    .rwd-table {
    margin-left:auto;
    margin-right:auto;
  }
     margin-left:auto;
    margin-right:auto;
}

body {
  padding: 0 2em;
  font-family: Montserrat, sans-serif;
  -webkit-font-smoothing: antialiased;
  text-rendering: optimizeLegibility;
  color: #444;
  background: #eee;
}
h1 {
  font-weight: normal;
  letter-spacing: -1px;
```

```
  color: #34495E;
}
.rwd-table {
  background: #34495E;
  color: #fff;
  border-radius: .4em;
  overflow: hidden;
}
.rwd-table tr {
  border-color: #46637f;
}
.rwd-table th, .rwd-table td {
  margin: .5em 1em;
}
@media (min-width: 480px) {
  .rwd-table th, .rwd-table td {
    padding: 1em !important;
  }
}
.rwd-table th, .rwd-table td:before {
  color: #dd5;
}
```

Deployment:

The prototype developed has been setup at Electronic Devices Lab of ECE Department (1116 Room No.) and Electronic Workshop and Basic Electronics Lab (1109 Room No.). The deployment is such that the PIR Entrance is outside the lab and PIR Exit is inside the lab. The RFID Reader is placed in such a way inside the lab that the data is read later, this is possible solely because of the fact that PIR has a hold time of 2 seconds.

The central attendance server has been setup at the ED Lab (1116 Room No.), all the attendance that is taken into account i.e. Present or Absent is noted at this end and the other information i.e. update of cards need to be done at this end itself.

The prototype developed has been tested for basic to higher level of hacks, i.e.

- If one student enters in the classroom and swipes two cards, then both of them are not read and processed only one among them is read and used for processing.
- If one student enters the classroom and leaves without swiping the card, then based on the flag condition after 180 minutes the person is marked as absent.

  Thus the prototype is properly setup, along with the database and has been checked for its proper working. The images below support the results:

  Servlet:

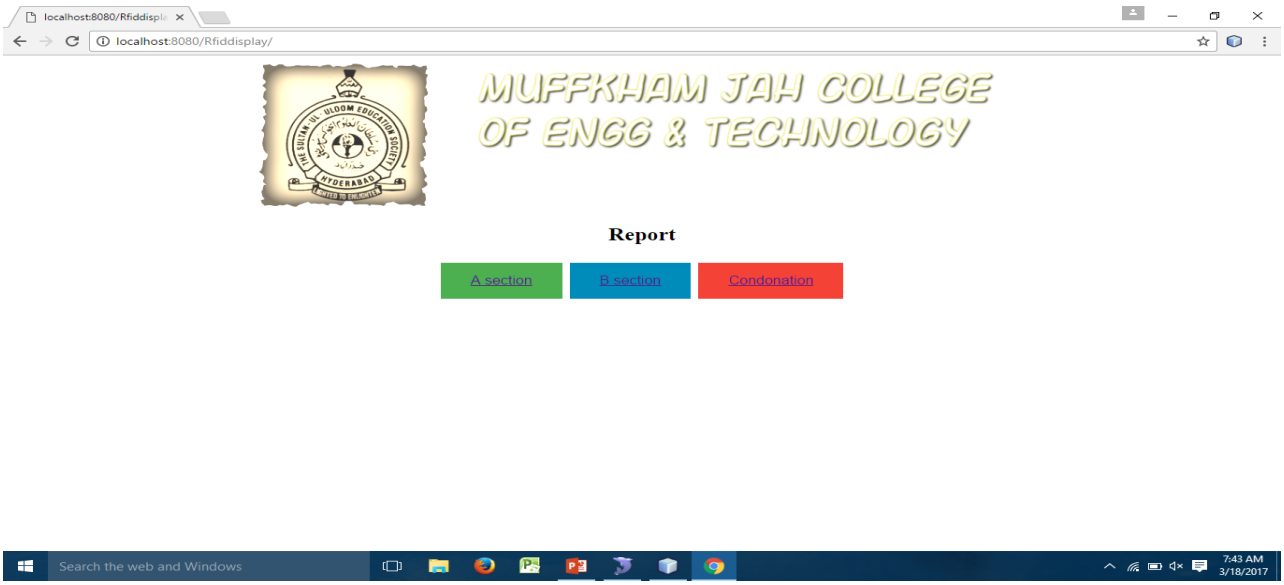


Fig 7-1: Welcome Screen

Fig 7-2: Section A data display

Fig 7-3: Section B data display

Fig 7-4: Condonation display

Database:

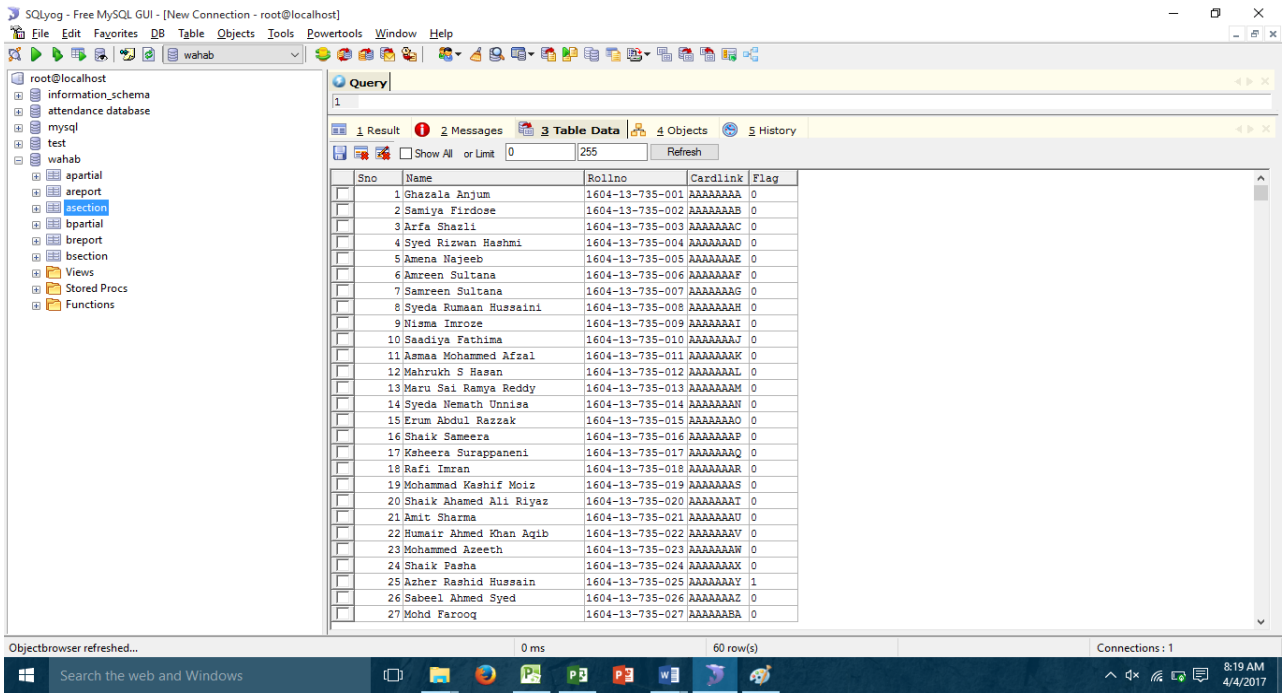


Fig 7-5: A Section Database

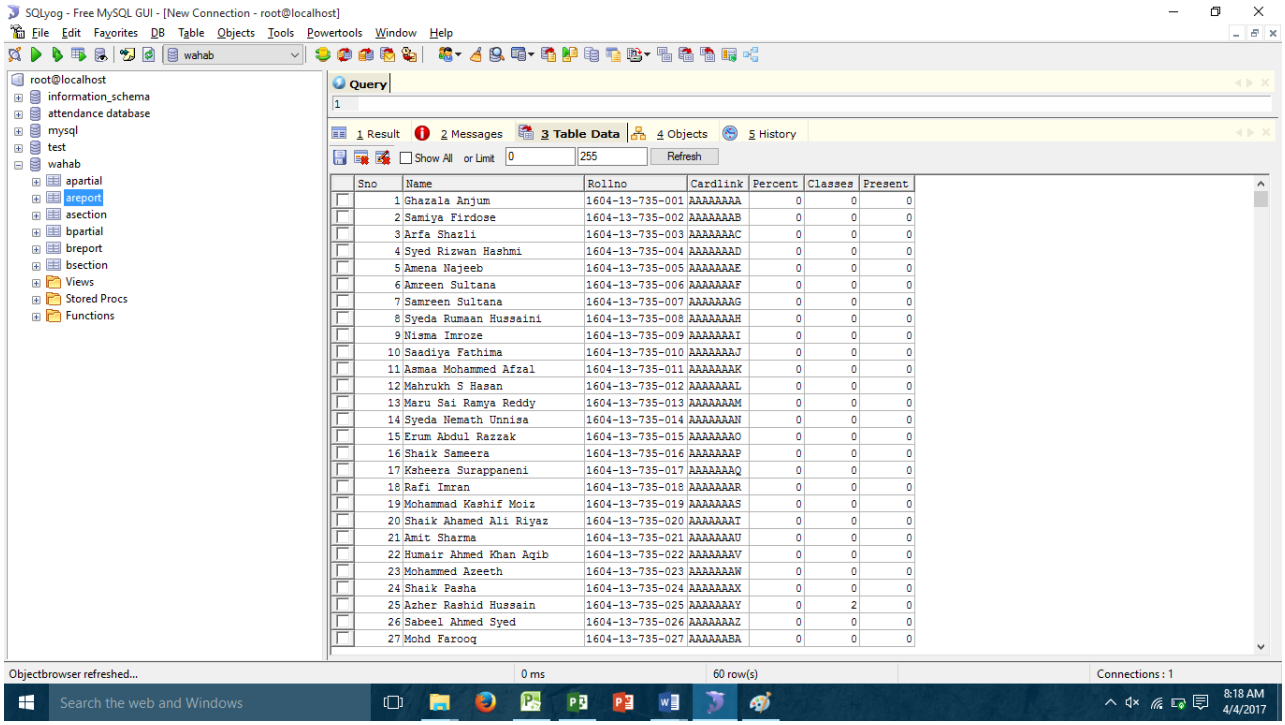


Fig 7-6: A Report Database

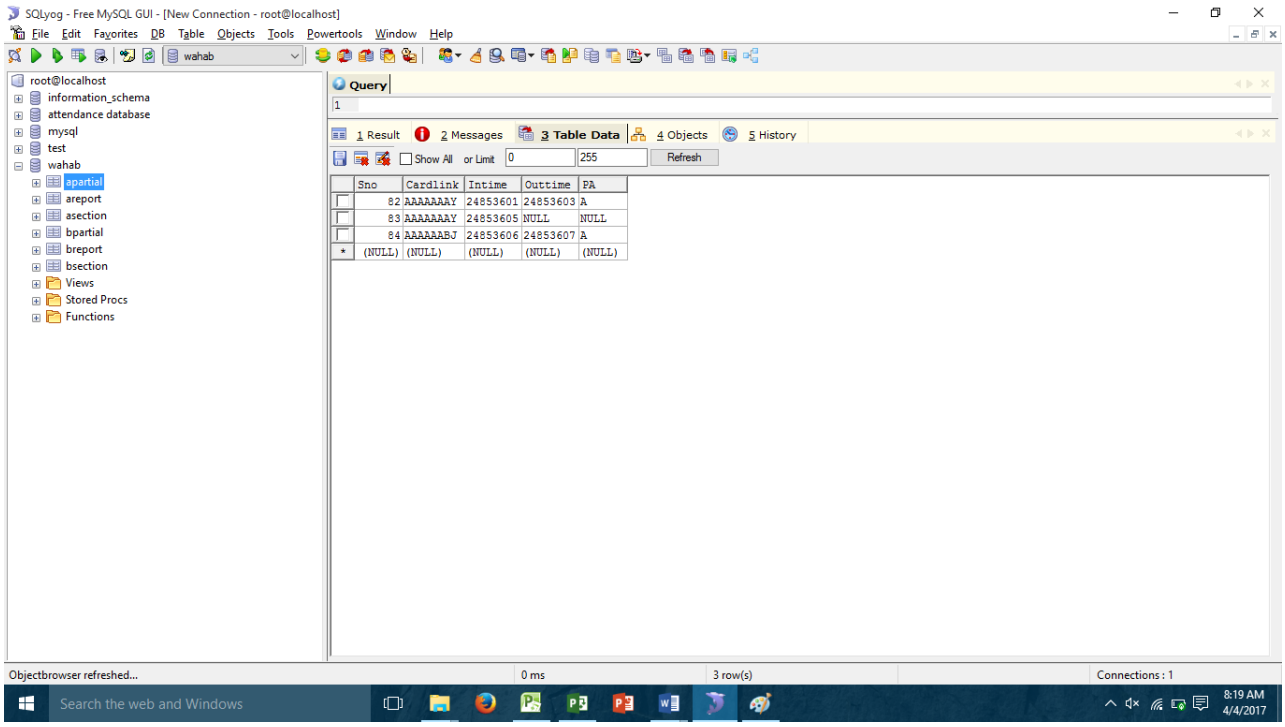


Fig 7-7: A Partial Database

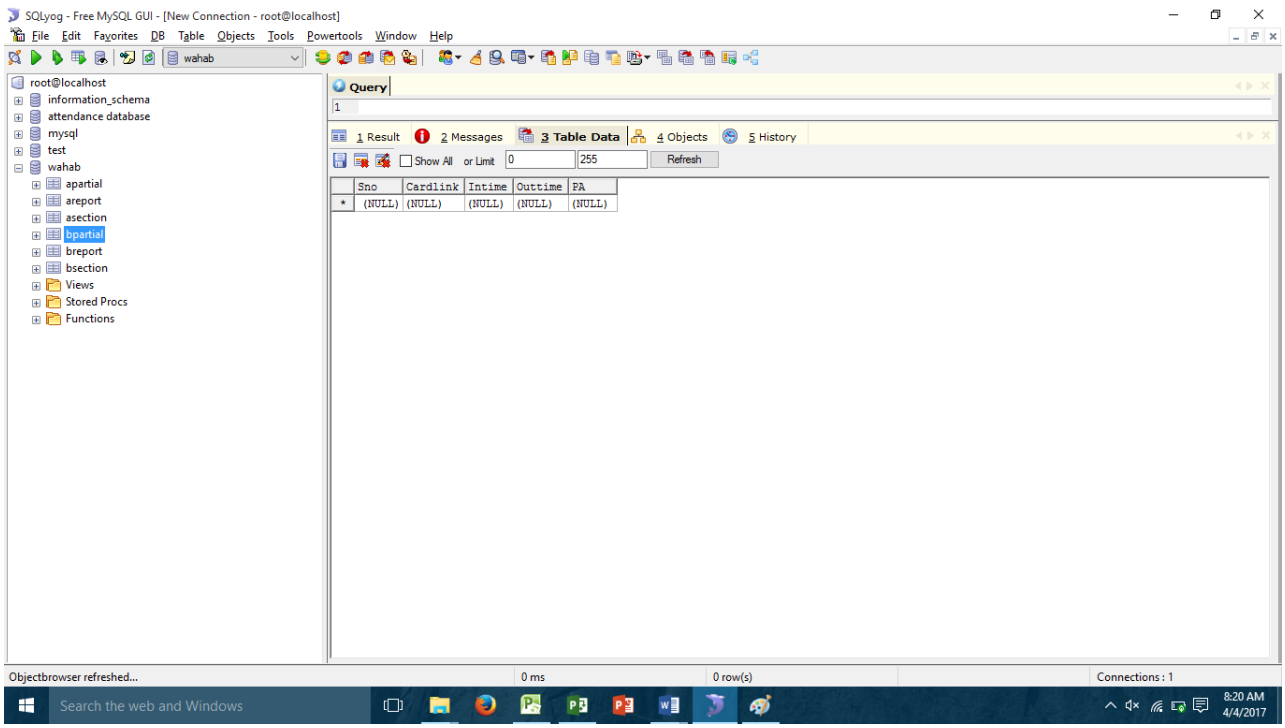


Fig 7-8: B Partial Database

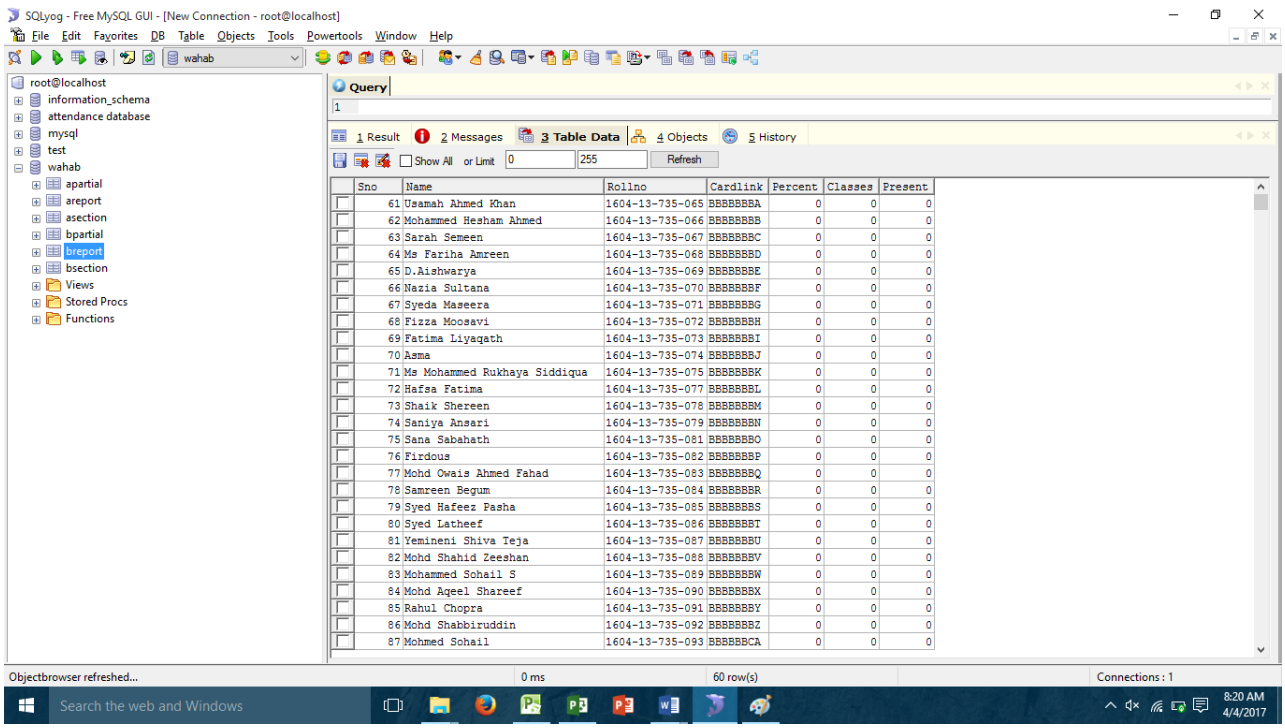


Fig 7-9: B Report Database

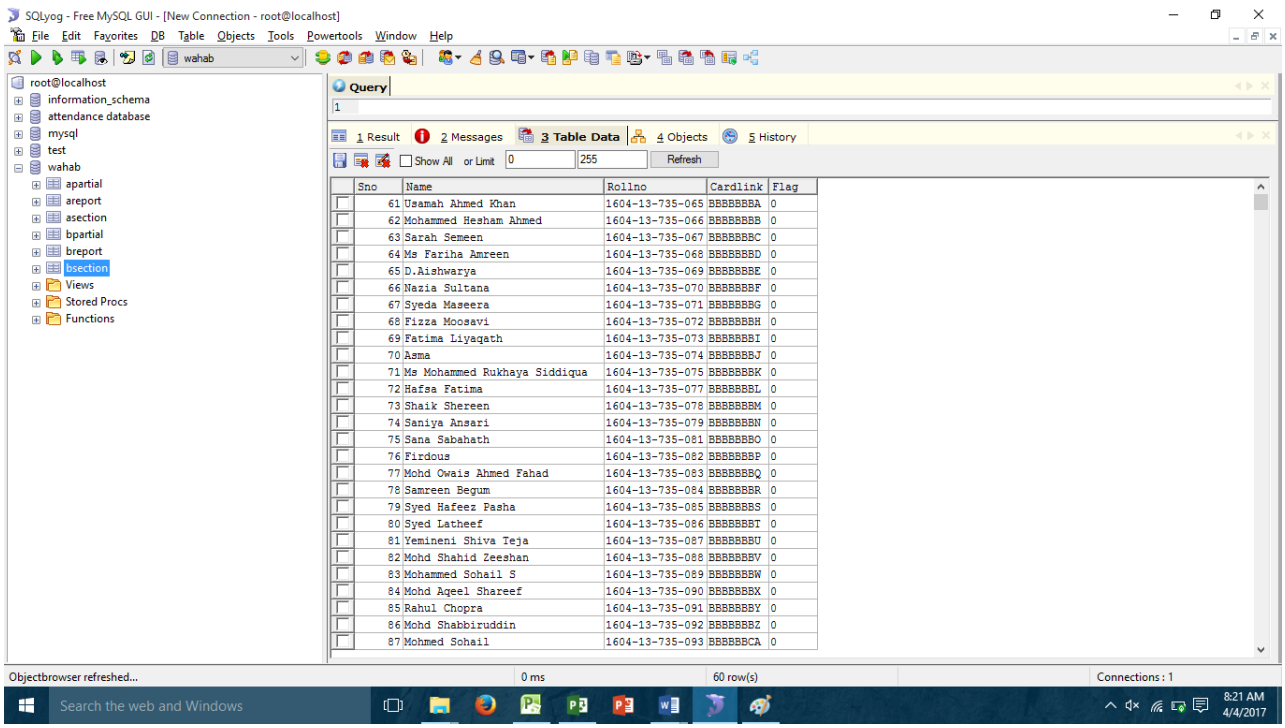


Fig 7-10: B Section Database

A smart university prototype developed using long range RFID technology, ZigBee communication standard protocol, motion sensors and Arduino microcontroller. These technologies are integrated together to improve creditability, authenticity of attendance in a smart manner. The beauty of the system is providing these features inexpensively. The using of Arduino board opens the way to integrate the system with wireless ZigBee camera to enforce the authentication approach. In addition to that, electrical power sensor (current/voltage) should be applied over the control circuit to send the sensing values to the central server wirelessly, in order to observe the power consumption over the campus and how can it be automated smartly.

If any person tries to manipulate the attendance at the server end, then the attendance is locked at the specific percentage for the whole semester and hence the student can be caught hold of.

The developed prototype covers basically all areas for proper authentication, except on one front. The person entering in with the card can also be the brother, cousin or sister of the actual student. To cover this front the project can be extended by connecting an IRIS Scanner along with the RFID Reader or even RFID Reader can be replaced with IRIS Scanner. The room automation control circuit can also be attached to make it even more enhanced.

[1] Mahmood K. Al, Muna M Al Nayar, Abbas R Kubba, "Designing of Smart University using RFID and WSN", International Journal of Computer Applications (0975 – 8887) Volume 112 – No 15, February 2015.

[2] M. Shanmugasundaram, G. Muthuselvi, S. Sundar, "Implementation of PIC 16F877A Based Intelligent Smart Home System", Volume 5 International Journal of Engineering and Technology (IJET), Apr-May 2013.

[3] Usha Sharma, S.R.N.Reddy, "Implementation of a WSN based Home/Office Automation (HOA)", Volume 3 International Journal of Engineering and Advanced Technology (IJEAT), Issue 3, February 2014.

[4] Mark Weiser, "The computer of the 21st century", Scientific American, Volume 265(3), pp.66--75, September 1991.

[5] Xuemi Li, Gang Xu, Li Li "RFID based Smart Home architecture for improving lives", 2nd International Conference Anti-counterfeiting, Security and Identification, 2008. ASID 2008.

[6] Li Yang, Maorong Ji, Zhenru Gao, "Design of Home Automation System Based on ZigBee Wireless Sensor Network", First International Conference on Information Science and Engineering, 2009. ICISE, 2009.

[7] Agnius Liutkevicius, Arunas Vrubliauskas, Egidijus Kazanavicius, "A Survey of Wireless Sensor Network Interconnection to External Networks", Springer Science, B.V. 2010

[8] Chong, C.Y. and Kumar, S.P. (2003) Sensor Networks: Evolution, Opportunities and Challenges. IEEE Transactions on Consumer Electronics, 8, August 2003, pp. 1247-1256.

[9] Chen, M., Chen W., Cui, W., Wen, V. and Woo, A. Security and Deployment Issues in a Sensor Network. Berkeley, California, USA, University Press, 2000.

[10] Crossbow Technologies Inc. Wireless Sensor Networks: getting started guide. [Online] Available at http://www.xbow.com/Support/Support_pdf_files/Getting_Started _Guide_7430-0022-05_B.pdf, 2004.

[11] Ganesan, D., Cerpa, A., Yu, Y. and Estrin, D. Networking Issues in Wireless Sensor Networks. Journal of Parallel and Distributed Computing (JPDC), Special issue on Frontiers in Distributed Sensor Networks, 64, 2004.

[12] Stajano, F.and Anderson, The resurrecting duckling: Security Issues for Ubiquitous Computing. In: Security & Privacy 2002, Supplement to IEEE Computer, Vol. 35, No. 4, April 2002, 22-26. http://www.computer.org/security/supplement1/sta/

[13] Elaine Shi and Adrian Perrig," Designing Secure Sensor Networks," IEEE Wireless Communications, December 2004

[14] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[15] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat Monitoring: Application Driver for Wireless Communications Technology. In Proceedings of the ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, 2001.

[16] G. Asada, M. Dong, T. S. Lin, F. Newberg, G. Pottie, and W. J. Kaiser. Wireless Integrated Network Sensors: Low Power Systems on a chip. In Proceedings of the 1998 European Solid State Circuits Conference, The Hague, Netherlands, 1998.

[17] Yi-Wei Ma 1, Jiann-Liang Chen 2, Yueh-Min Huang 1,3,* and Mei-Yu Lee , " An Efficient Management System for Wireless Sensor Networks", Sensors 2010, 11400-11413

[18] Perrig, A., Stankovic, J. and Wagner, D. Security in Wireless Sensor Network. Communication of the ACM, 47, 2004.

[19] D. Djenouri, L. Khelladi, and A. N. Badache, "A Survey of Security Issues in Mobile Ad Hoc and Sensor Netowkrs", In Proceedings of Communications Surveys and Tutorials, Vol. 7, No. 4, pp. 2-28, 2005

[20] Camtepe.S, Yener.B, "Key Distribution Mechanisms for Wireless Sensor Networks: A Survey" in Technical report, Rensselaer Polytechnic Institute. TR -05-07, 2005.

[21] Arduino. Introduction to Arduino [Online]. Italy: Arduino URL: http://arduino.cc/en/Guide/Introduction Accessed July 15 2013

[22] Arduino. Arduino board Uno [Online]. Italy: Arduino URL: http://arduino.cc/en/Main/ArduinoBoardUno Accessed 17 July 2013

[23] Arduino. Arduino board Mega [Online]. Italy: Arduino URL: http://arduino.cc/en/Main/ArduinoBoardMega Accessed 17 July 2013

[24] http://www.rfidjournal.com/article/view/1339/

[25] http://www.aimglobal.org/technologies/RFID/what_is_rfid.asp

[26] Zhi Chang, Zhangeng Sun & Junbao Gu, "Study on the Technology of the Coal Mining Safety Monitoring System". Modern Applied Science. Vol.3,No.8,(August 2009)

[27] Castro, L. and Fosso Wamba, S. (2007). "An Inside Look at RFID Technology". Journal of Technology Management & Innovation, Vol. 2, No. 1, pp. 128 – 141, (ERA: C)

[28] http://www.rfidjournal.com/article/view/1334/

[29] Debdoot Sheet, Atul Kumar, Agnibesh Dutta, Samrat Das Gupta, Tirthankar Datta, Subir Kr. Sarkar. "Realization and Simulation of the Hardware for RFID System and its Performance study." Proc. Of ICTES 2007. Dr. M.G.R. University, Chennai, Tamil Nadu, India. December, 20-22, 2007. Pp.697-700.

[30] Klaus Finkenzeller, RFID Handbook. John Wiley and Sons, 2003.

[31] IBM WebSphere RFID Handbook: A Solution Guide (http://www.ibm.com/redbooks)

[32] Application Notes, "Introduction to RFID Technology" CAENRFID: The Art of Identification (2008)

[33] L. Srivastava, RFID: Technology, Applications and Policy Implications, Presentation, International Telecommunication Union, Kenya, (2005)

[34] T. Frank, H. Brad, M. Anand, B. Hersh, C. Anita, K. John, "RFID Security", (2006) ISBN: 1-59749-047-4

[35] www.slais.ubc.ca/courses/libr500/04-05-wt2/www/A_Farrell/HowDoesRFIDWork.htm

[36] V. Daniel Hunt, Albert Puglia, Mike Puglia. RFID-A Guide to Radio Frequency Identification. Wiley-Interscience. 2007

[37] https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor

[38] http://www.informit.com/articles/article.aspx?p=1409785&seqNum=3 Introduction to the ZigBee Wireless Sensor and Control Network

[39] https://en.wikipedia.org/wiki/ZigBee

[40] http://www3.nd.edu/~mhaenggi/ee67011/zigbee.pdf ZigBee Document 053474r06, Version 1.0 December 14th, 2004

[41] ZigBee Wireless Networks and Transceivers – Shahin Farahani

[42] XBee module: [Online] Available: http://www.digi.com/

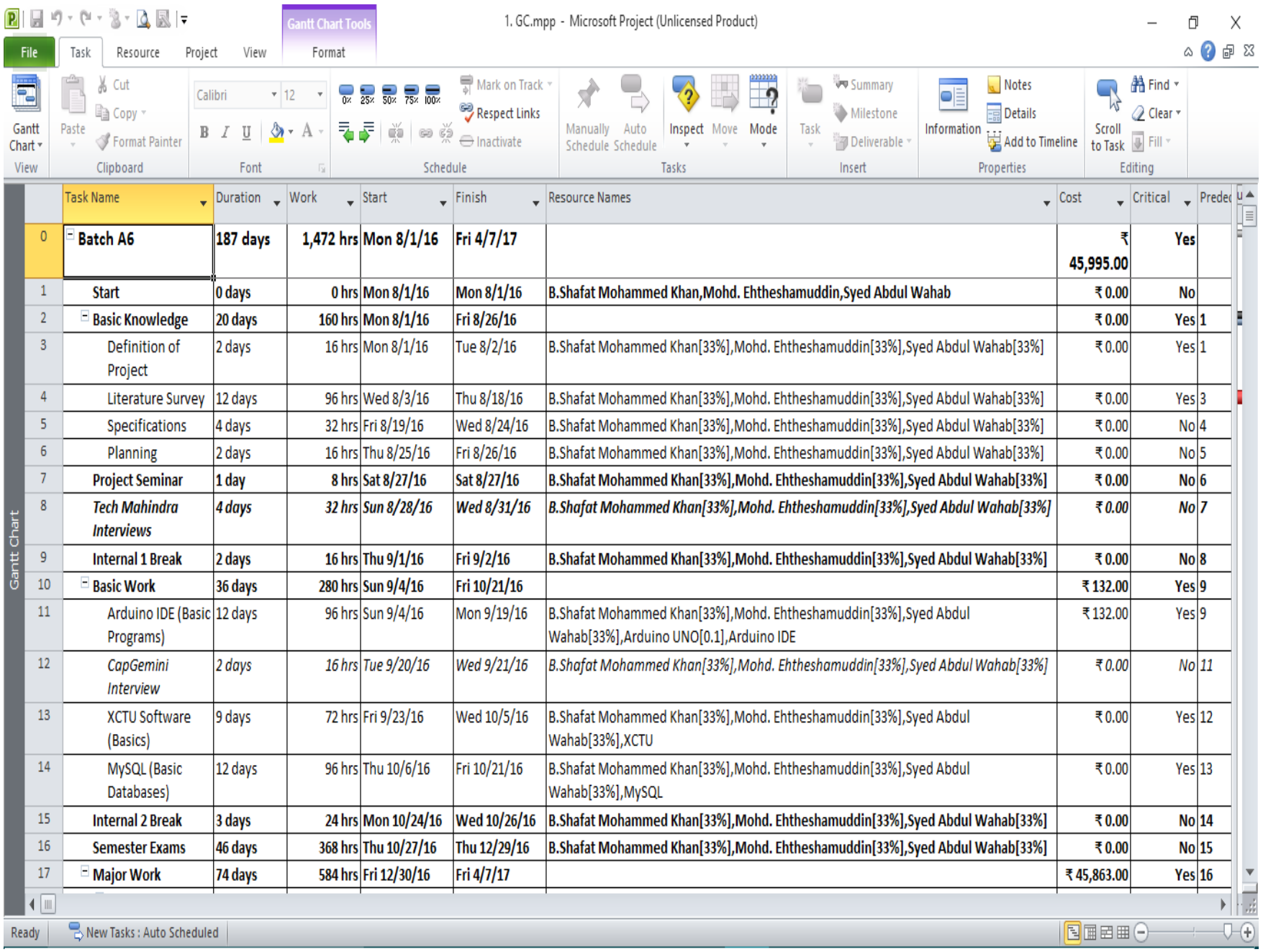
[43] Arduino. [Online] Available: https://www.arduino.cc/en/Main/Software

[44] XCTU. [Online] : https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

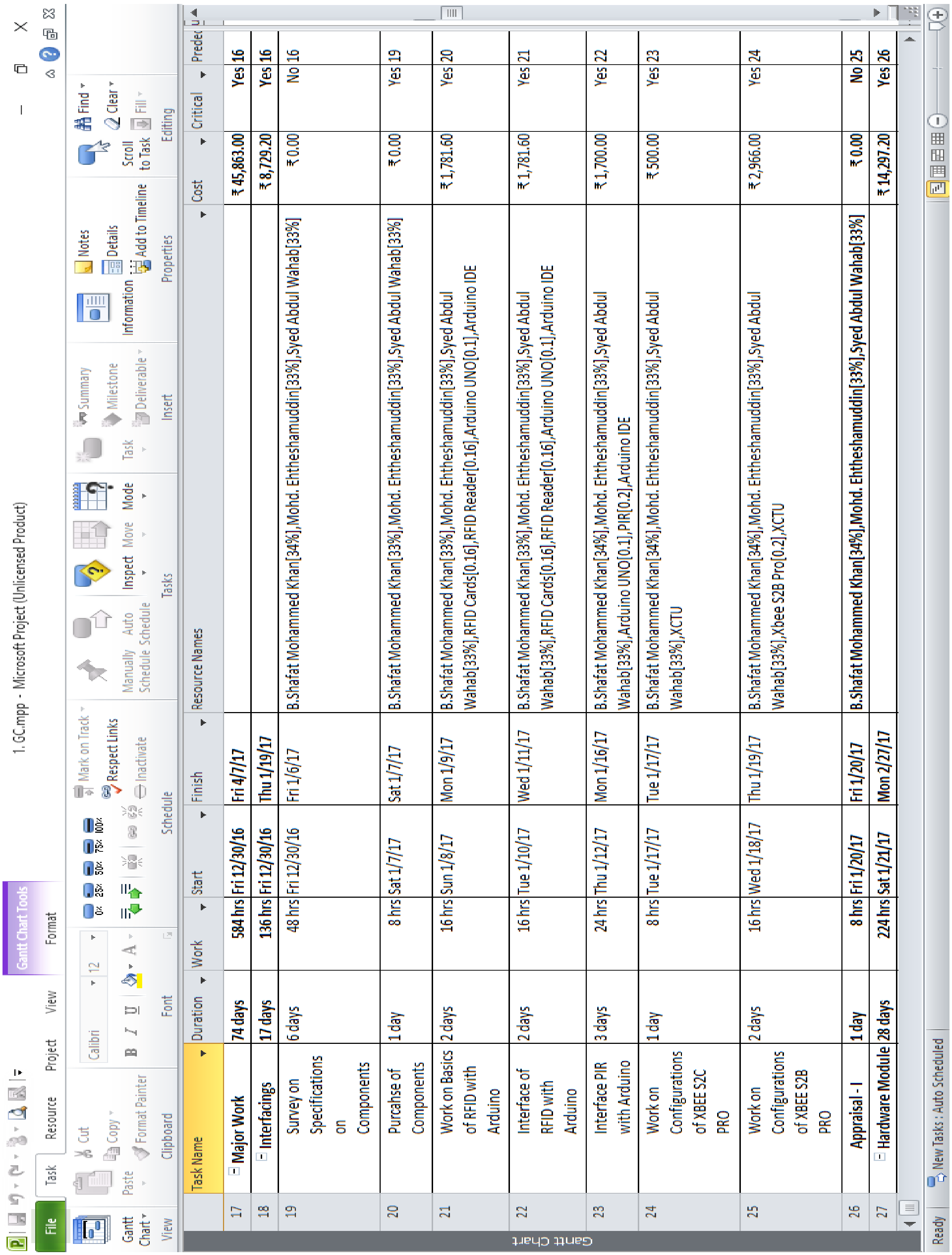
[45] MySQL [Online] Available: https://dev.mysql.com/doc/refman/5.5/en/introduction.html

[46] NetBeans [Online] Available: https://netbeans.org/features/index.html

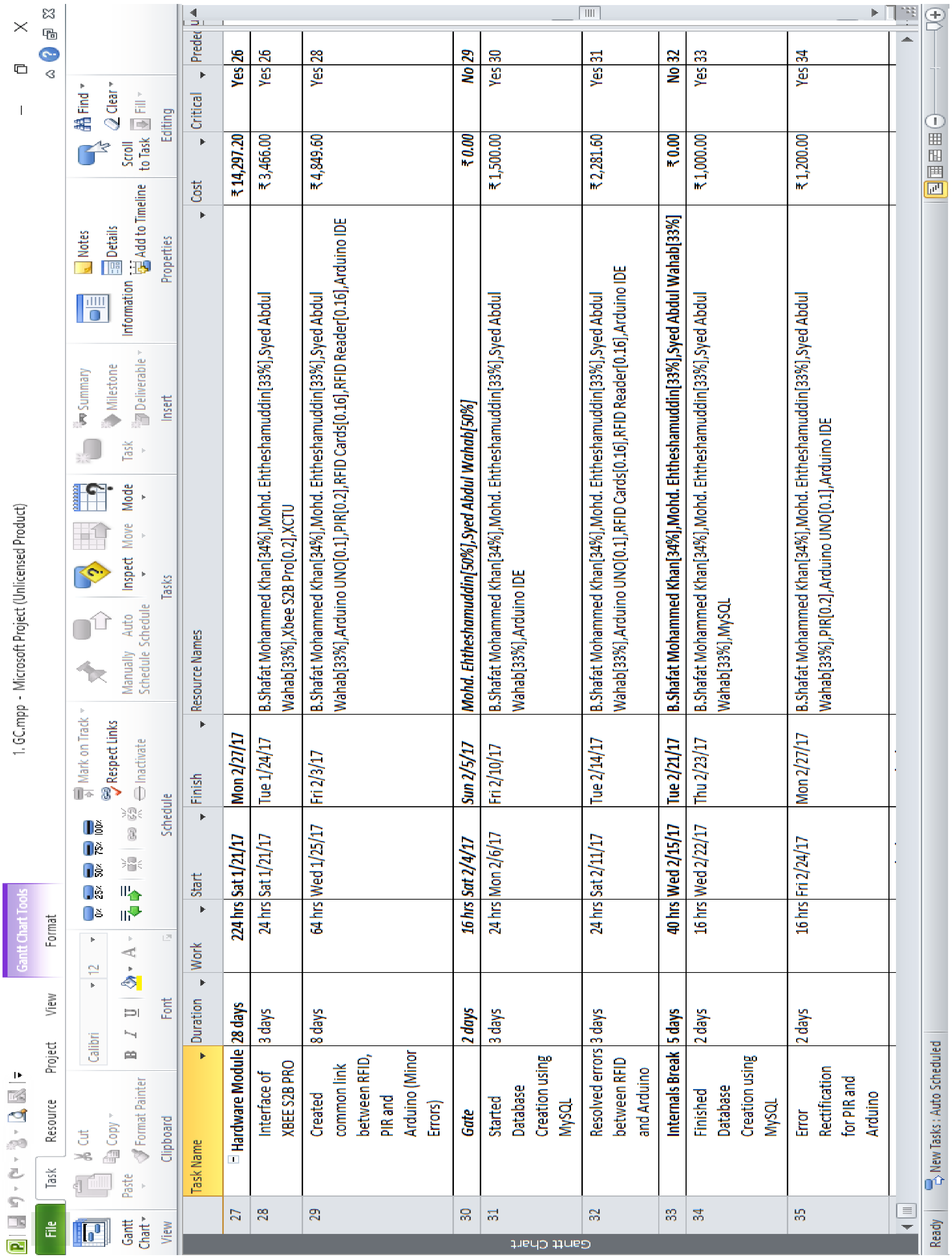
[47] GlassFish Server [Online]:https://docs.oracle.com/cd/E18930_01/html/821-2416/giubb.html



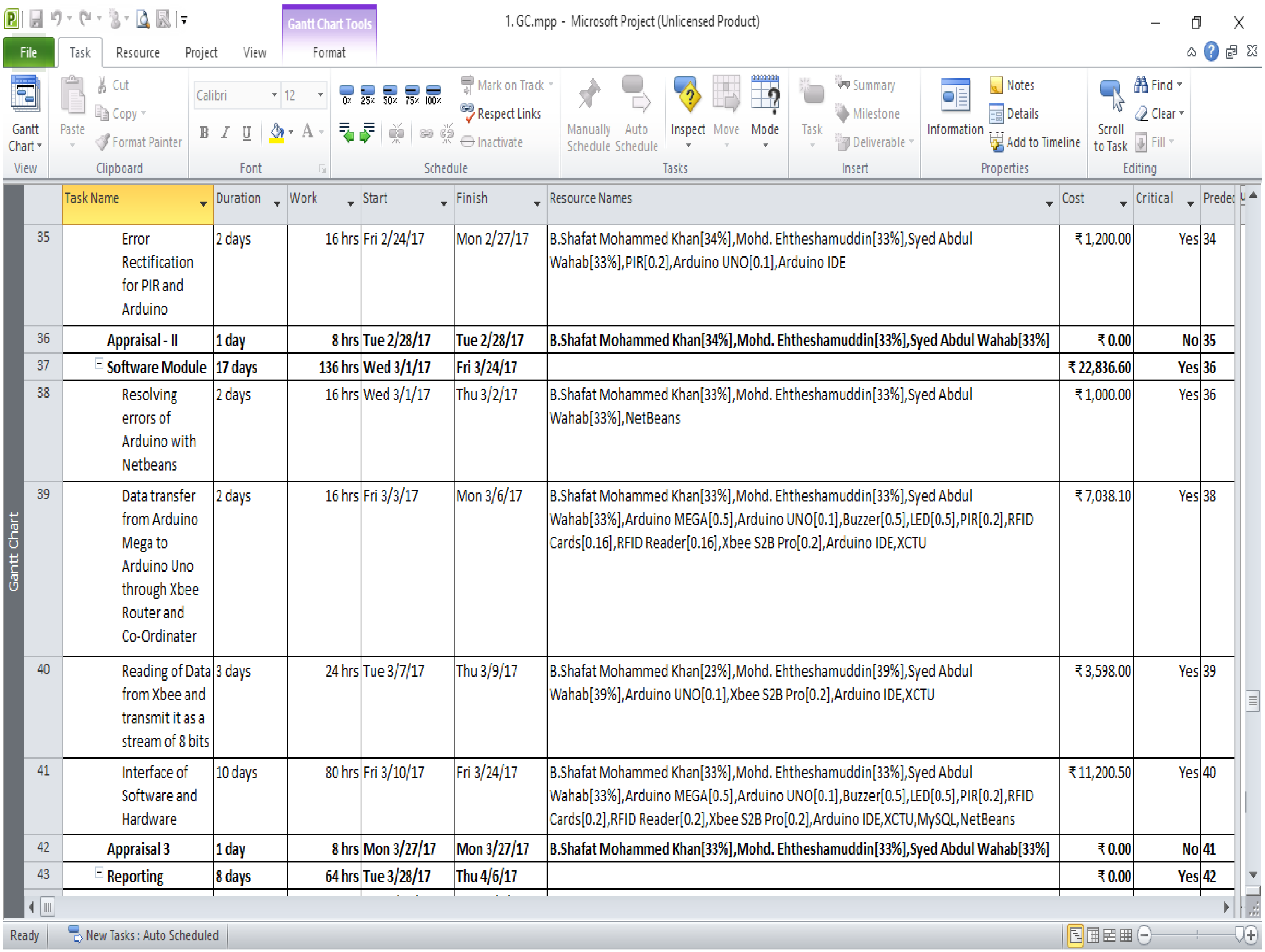
	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predec
0	**Batch A6**	**187 days**	**1,472 hrs**	**Mon 8/1/16**	**Fri 4/7/17**		**₹ 45,995.00**	**Yes**	
1	**Start**	**0 days**	**0 hrs**	**Mon 8/1/16**	**Mon 8/1/16**	**B.Shafat Mohammed Khan,Mohd. Ehtheshamuddin,Syed Abdul Wahab**	**₹ 0.00**	**No**	
2	**Basic Knowledge**	**20 days**	**160 hrs**	**Mon 8/1/16**	**Fri 8/26/16**		**₹ 0.00**	**Yes**	**1**
3	Definition of Project	2 days	16 hrs	Mon 8/1/16	Tue 8/2/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	1
4	Literature Survey	12 days	96 hrs	Wed 8/3/16	Thu 8/18/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	3
5	Specifications	4 days	32 hrs	Fri 8/19/16	Wed 8/24/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	No	4
6	Planning	2 days	16 hrs	Thu 8/25/16	Fri 8/26/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	No	5
7	**Project Seminar**	**1 day**	**8 hrs**	**Sat 8/27/16**	**Sat 8/27/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**6**
8	***Tech Mahindra Interviews***	***4 days***	***32 hrs***	***Sun 8/28/16***	***Wed 8/31/16***	***B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]***	***₹ 0.00***	***No***	***7***
9	**Internal 1 Break**	**2 days**	**16 hrs**	**Thu 9/1/16**	**Fri 9/2/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**8**
10	**Basic Work**	**36 days**	**280 hrs**	**Sun 9/4/16**	**Fri 10/21/16**		**₹ 132.00**	**Yes**	**9**
11	Arduino IDE (Basic Programs)	12 days	96 hrs	Sun 9/4/16	Mon 9/19/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],Arduino IDE	₹ 132.00	Yes	9
12	*CapGemini Interview*	*2 days*	*16 hrs*	*Tue 9/20/16*	*Wed 9/21/16*	*B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]*	*₹ 0.00*	*No*	*11*
13	XCTU Software (Basics)	9 days	72 hrs	Fri 9/23/16	Wed 10/5/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],XCTU	₹ 0.00	Yes	12
14	MySQL (Basic Databases)	12 days	96 hrs	Thu 10/6/16	Fri 10/21/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],MySQL	₹ 0.00	Yes	13
15	**Internal 2 Break**	**3 days**	**24 hrs**	**Mon 10/24/16**	**Wed 10/26/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**14**
16	**Semester Exams**	**46 days**	**368 hrs**	**Thu 10/27/16**	**Thu 12/29/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**15**
17	**Major Work**	**74 days**	**584 hrs**	**Fri 12/30/16**	**Fri 4/7/17**		**₹ 45,863.00**	**Yes**	**16**



	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predecessors
17	**Major Work**	74 days	584 hrs	Fri 12/30/16	Fri 4/7/17		₹45,863.00	Yes	16
18	**Interfacing**	17 days	136 hrs	Fri 12/30/16	Thu 1/19/17		₹8,729.20	Yes	16
19	Survey on Specifications on Components	6 days	48 hrs	Fri 12/30/16	Fri 1/6/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹0.00	No	16
20	Purcahse of Components	1 day	8 hrs	Sat 1/7/17	Sat 1/7/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹0.00	Yes	19
21	Work on Basics of RFID with Arduino	2 days	16 hrs	Sun 1/8/17	Mon 1/9/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],RFID Cards[0.16],RFID Reader[0.16],Arduino UNO[0.1],Arduino IDE	₹1,781.60	Yes	20
22	Interface of RFID with Arduino	2 days	16 hrs	Tue 1/10/17	Wed 1/11/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],RFID Cards[0.16],RFID Reader[0.16],Arduino UNO[0.1],Arduino IDE	₹1,781.60	Yes	21
23	Interface PIR with Arduino	3 days	24 hrs	Thu 1/12/17	Mon 1/16/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],PIR[0.2],Arduino IDE	₹1,700.00	Yes	22
24	Work on Configurations of XBEE S2C PRO	1 day	8 hrs	Tue 1/17/17	Tue 1/17/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],XCTU	₹500.00	Yes	23
25	Work on Configurations of XBEE S2B PRO	2 days	16 hrs	Wed 1/18/17	Thu 1/19/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Xbee S2B Pro[0.2],XCTU	₹2,966.00	Yes	24
26	**Appraisal - I**	1 day	8 hrs	Fri 1/20/17	Fri 1/20/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹0.00	No	25
27	**Hardware Module**	28 days	224 hrs	Sat 1/21/17	Mon 2/27/17		₹14,297.20	Yes	26



	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predecessors
27	**Hardware Module**	**28 days**	**224 hrs**	**Sat 1/21/17**	**Mon 2/27/17**		**₹ 14,297.20**	**Yes**	**26**
28	Interface of XBEE S2B PRO	3 days	24 hrs	Sat 1/21/17	Tue 1/24/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Xbee S2B Pro[0.2],XCTU	₹ 3,466.00	Yes	26
29	Created common link between RFID, PIR and Arduino (Minor Errors)	8 days	64 hrs	Wed 1/25/17	Fri 2/3/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],PIR[0.2],RFID Cards[0.16],RFID Reader[0.16],Arduino IDE	₹ 4,849.60	Yes	28
30	***Gate***	***2 days***	***16 hrs***	***Sat 2/4/17***	***Sun 2/5/17***	***Mohd. Ehtheshamuddin[50%],Syed Abdul Wahab[50%]***	***₹ 0.00***	***No***	***29***
31	Started Database Creation using MySQL	3 days	24 hrs	Mon 2/6/17	Fri 2/10/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino IDE	₹ 1,500.00	Yes	30
32	Resolved errors between RFID and Arduino	3 days	24 hrs	Sat 2/11/17	Tue 2/14/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],RFID Cards[0.16],RFID Reader[0.16],Arduino IDE	₹ 2,281.60	Yes	31
33	**Internals Break**	**5 days**	**40 hrs**	**Wed 2/15/17**	**Tue 2/21/17**	**B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**32**
34	Finished Database Creation using MySQL	2 days	16 hrs	Wed 2/22/17	Thu 2/23/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],MySQL	₹ 1,000.00	Yes	33
35	Error Rectification for PIR and Arduino	2 days	16 hrs	Fri 2/24/17	Mon 2/27/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],PIR[0.2],Arduino UNO[0.1],Arduino IDE	₹ 1,200.00	Yes	34

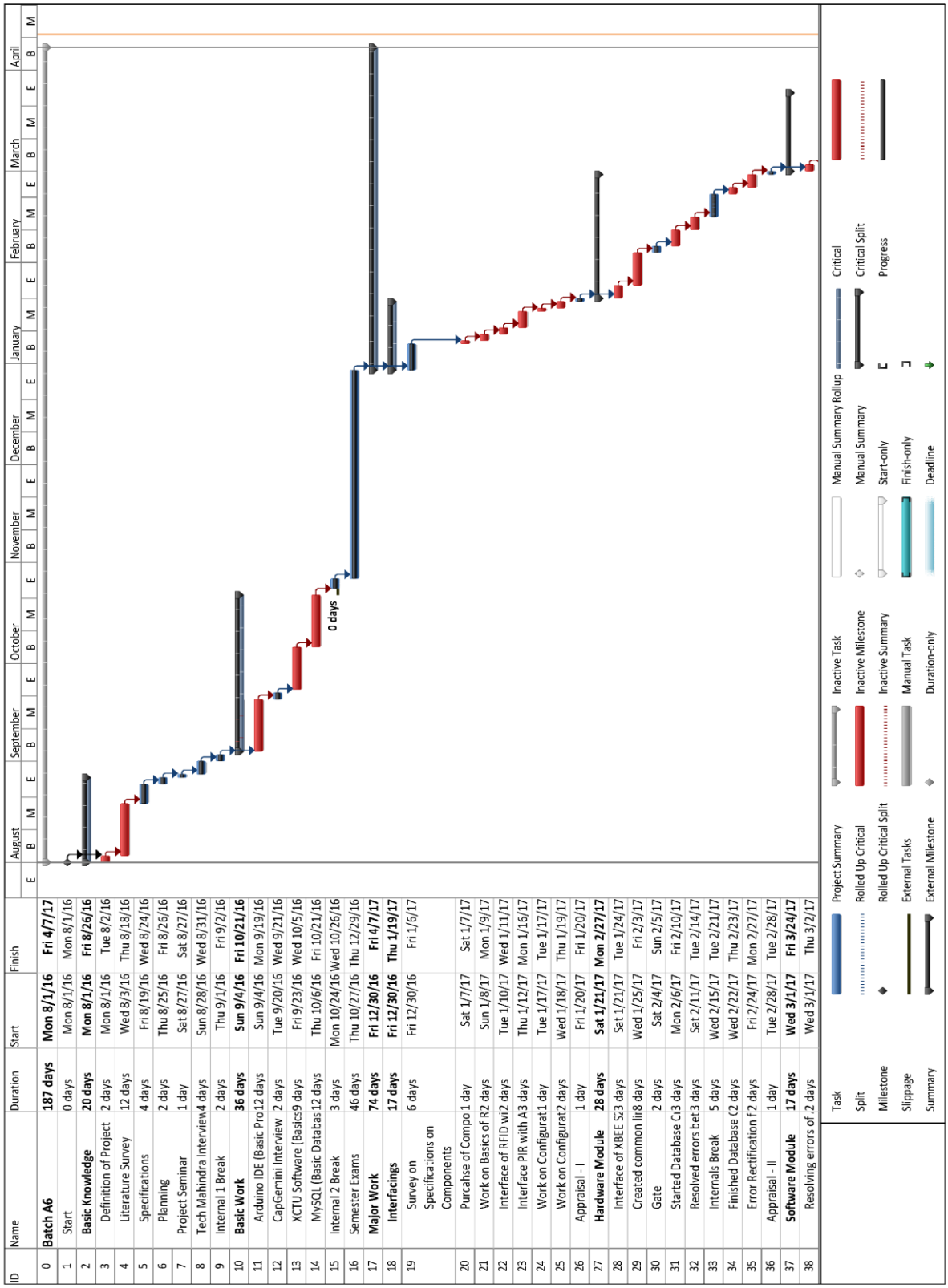


	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predec
35	Error Rectification for PIR and Arduino	2 days	16 hrs	Fri 2/24/17	Mon 2/27/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],PIR[0.2],Arduino UNO[0.1],Arduino IDE	₹1,200.00	Yes	34
36	**Appraisal - II**	**1 day**	**8 hrs**	**Tue 2/28/17**	**Tue 2/28/17**	**B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹0.00**	**No**	**35**
37	**Software Module**	**17 days**	**136 hrs**	**Wed 3/1/17**	**Fri 3/24/17**		**₹22,836.60**	**Yes**	**36**
38	Resolving errors of Arduino with Netbeans	2 days	16 hrs	Wed 3/1/17	Thu 3/2/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],NetBeans	₹1,000.00	Yes	36
39	Data transfer from Arduino Mega to Arduino Uno through Xbee Router and Co-Ordinater	2 days	16 hrs	Fri 3/3/17	Mon 3/6/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino MEGA[0.5],Arduino UNO[0.1],Buzzer[0.5],LED[0.5],PIR[0.2],RFID Cards[0.16],RFID Reader[0.16],Xbee S2B Pro[0.2],Arduino IDE,XCTU	₹7,038.10	Yes	38
40	Reading of Data from Xbee and transmit it as a stream of 8 bits	3 days	24 hrs	Tue 3/7/17	Thu 3/9/17	B.Shafat Mohammed Khan[23%],Mohd. Ehtheshamuddin[39%],Syed Abdul Wahab[39%],Arduino UNO[0.1],Xbee S2B Pro[0.2],Arduino IDE,XCTU	₹3,598.00	Yes	39
41	Interface of Software and Hardware	10 days	80 hrs	Fri 3/10/17	Fri 3/24/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino MEGA[0.5],Arduino UNO[0.1],Buzzer[0.5],LED[0.5],PIR[0.2],RFID Cards[0.2],RFID Reader[0.2],Xbee S2B Pro[0.2],Arduino IDE,XCTU,MySQL,NetBeans	₹11,200.50	Yes	40
42	**Appraisal 3**	**1 day**	**8 hrs**	**Mon 3/27/17**	**Mon 3/27/17**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹0.00**	**No**	**41**
43	**Reporting**	**8 days**	**64 hrs**	**Tue 3/28/17**	**Thu 4/6/17**		**₹0.00**	**Yes**	**42**

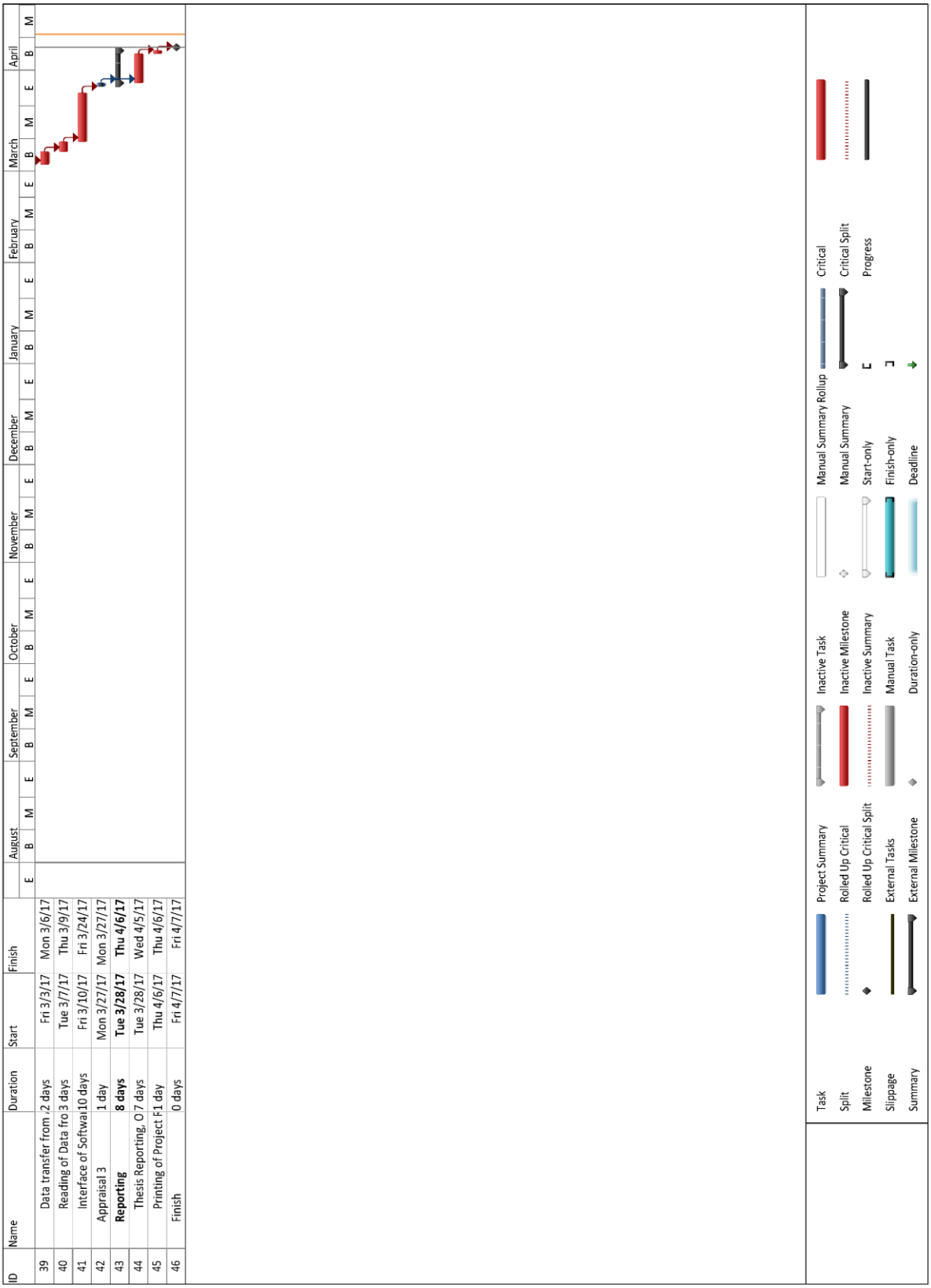
1. GC.mpp - Microsoft Project (Unlicensed Product)

	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Prede
40	Reading of Data from Xbee and transmit it as a stream of 8 bits	3 days	24 hrs	Tue 3/7/17	Thu 3/9/17	B.Shafat Mohammed Khan[23%],Mohd. Ehtheshamuddin[39%],Syed Abdul Wahab[39%],Arduino UNO[0.1],Xbee S2B Pro[0.2],Arduino IDE,XCTU	₹3,598.00	Yes	39
41	Interface of Software and Hardware	10 days	80 hrs	Fri 3/10/17	Fri 3/24/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino MEGA[0.5],Arduino UNO[0.1],Buzzer[0.5],LED[0.5],PIR[0.2],RFID Cards[0.2],RFID Reader[0.2],Xbee S2B Pro[0.2],Arduino IDE,XCTU,MySQL,NetBeans	₹11,200.50	Yes	40
42	**Appraisal 3**	**1 day**	**8 hrs**	**Mon 3/27/17**	**Mon 3/27/17**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹0.00**	**No**	**41**
43	**Reporting**	**8 days**	**64 hrs**	**Tue 3/28/17**	**Thu 4/6/17**		**₹0.00**	**Yes**	**42**
44	Thesis Reporting, Organising Correction and	7 days	56 hrs	Tue 3/28/17	Wed 4/5/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹0.00	Yes	42
45	Printing of Project Report	1 day	8 hrs	Thu 4/6/17	Thu 4/6/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹0.00	Yes	44
46	**Finish**	**1 day**	**0 hrs**	**Fri 4/7/17**	**Fri 4/7/17**		**₹0.00**	**No**	**45**

Ready | New Tasks : Auto Scheduled



Detail Gantt Chart


ID
Name
Duration
Start
Finish
August
September
October
November
December
January
February
March
April
39 Data transfer from ,2 days Fri 3/3/17 Mon 3/6/17
40 Reading of Data fro 3 days Tue 3/7/17 Thu 3/9/17
41 Interface of Softwai 10 days Fri 3/10/17 Fri 3/24/17
42 Appraisal 3 1 day Mon 3/27/17 Mon 3/27/17
43 Reporting 8 days Tue 3/28/17 Thu 4/6/17
44 Thesis Reporting, O 7 days Tue 3/28/17 Wed 4/5/17
45 Printing of Project Fi 1 day Thu 4/6/17 Thu 4/6/17
46 Finish 0 days Fri 4/7/17 Fri 4/7/17
Task
Split
Milestone
Slippage
Summary
Project Summary
Rolled Up Critical
Rolled Up Critical Split
External Tasks
External Milestone
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
Deadline
Critical
Critical Split
Progress

Batch A6

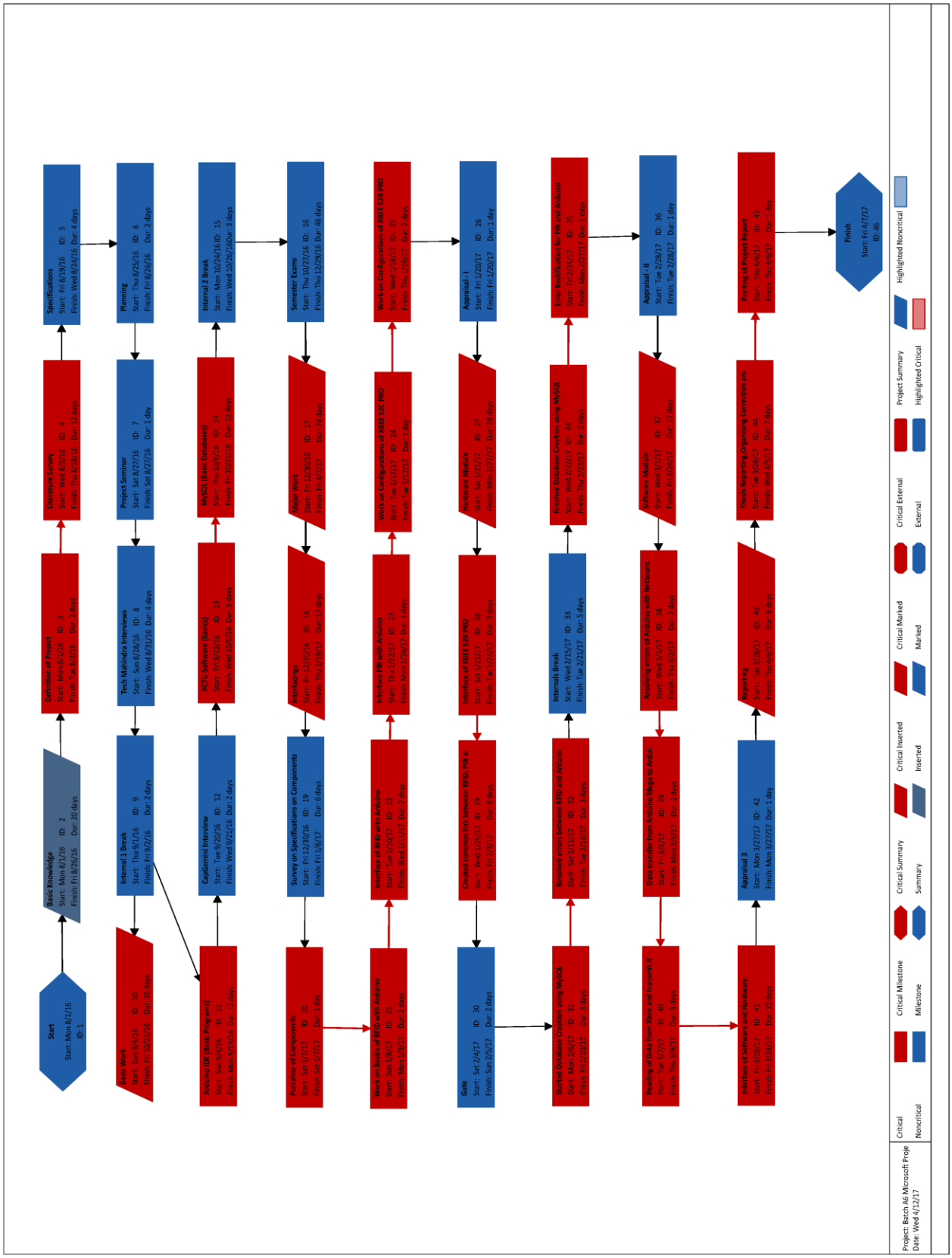
ID	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predecessors
0	**Batch A6**	**187 days**	**1,472 hrs**	**Mon 8/1/16**	**Fri 4/7/17**		**₹ 45,995.00**	**Yes**	
1	**Start**	**0 days**	**0 hrs**	**Mon 8/1/16**	**Mon 8/1/16**	**B.Shafat Mohammed Khan,Mohd. Ehtheshamuddin,Syed Abdul Wahab**	**₹ 0.00**	**No**	
2	**Basic Knowledge**	**20 days**	**160 hrs**	**Mon 8/1/16**	**Fri 8/26/16**		**₹ 0.00**	**Yes**	**1**
3	Definition of Project	2 days	16 hrs	Mon 8/1/16	Tue 8/2/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	1
4	Literature Survey	12 days	96 hrs	Wed 8/3/16	Thu 8/18/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	3
5	Specifications	4 days	32 hrs	Fri 8/19/16	Wed 8/24/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	No	4
6	Planning	2 days	16 hrs	Thu 8/25/16	Fri 8/26/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	No	5
7	**Project Seminar**	**1 day**	**8 hrs**	**Sat 8/27/16**	**Sat 8/27/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**6**
8	***Tech Mahindra Interviews***	***4 days***	***32 hrs***	***Sun 8/28/16***	***Wed 8/31/16***	***B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]***	***₹ 0.00***	***No***	***7***
9	**Internal 1 Break**	**2 days**	**16 hrs**	**Thu 9/1/16**	**Fri 9/2/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**8**
10	**Basic Work**	**36 days**	**280 hrs**	**Sun 9/4/16**	**Fri 10/21/16**		**₹ 132.00**	**Yes**	**9**
11	Arduino IDE (Basic Programs)	12 days	96 hrs	Sun 9/4/16	Mon 9/19/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],Arduino IDE	₹ 132.00	Yes	9
12	***CapGemini Interview***	***2 days***	***16 hrs***	***Tue 9/20/16***	***Wed 9/21/16***	***B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]***	***₹ 0.00***	***No***	***11***
13	XCTU Software (Basics)	9 days	72 hrs	Fri 9/23/16	Wed 10/5/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],XCTU	₹ 0.00	Yes	12
14	MySQL (Basic Databases)	12 days	96 hrs	Thu 10/6/16	Fri 10/21/16	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],MySQL	₹ 0.00	Yes	13
15	**Internal 2 Break**	**3 days**	**24 hrs**	**Mon 10/24/16**	**Wed 10/26/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**14**
16	**Semester Exams**	**46 days**	**368 hrs**	**Thu 10/27/16**	**Thu 12/29/16**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**15**
17	**Major Work**	**74 days**	**584 hrs**	**Fri 12/30/16**	**Fri 4/7/17**		**₹ 45,863.00**	**Yes**	**16**
18	**Interfacings**	**17 days**	**136 hrs**	**Fri 12/30/16**	**Thu 1/19/17**		**₹ 8,729.20**	**Yes**	**16**
19	Survey on Specifications on Components	6 days	48 hrs	Fri 12/30/16	Fri 1/6/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	No	16
20	Purcahse of Components	1 day	8 hrs	Sat 1/7/17	Sat 1/7/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	19
21	Work on Basics of RFID with Arduino	2 days	16 hrs	Sun 1/8/17	Mon 1/9/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],RFID Cards[0.16],RFID Reader[0.16],Arduino UNO[0.1],Arduino IDE	₹ 1,781.60	Yes	20
22	Interface of RFID with Arduino	2 days	16 hrs	Tue 1/10/17	Wed 1/11/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],RFID Cards[0.16],RFID Reader[0.16],Arduino UNO[0.1],Arduino IDE	₹ 1,781.60	Yes	21
23	Interface PIR with Arduino	3 days	24 hrs	Thu 1/12/17	Mon 1/16/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],PIR[0.2],Arduino IDE	₹ 1,700.00	Yes	22
24	Work on Configurations of XBEE S2C PRO	1 day	8 hrs	Tue 1/17/17	Tue 1/17/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],XCTU	₹ 500.00	Yes	23

Batch A6

ID	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predecessors
25	Work on Configurations of XBEE S2B PRO	2 days	16 hrs	Wed 1/18/17	Thu 1/19/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Xbee S2B Pro[0.2],XCTU	₹ 2,966.00	Yes	24
26	**Appraisal - I**	**1 day**	**8 hrs**	**Fri 1/20/17**	**Fri 1/20/17**	**B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**25**
27	**Hardware Module**	**28 days**	**224 hrs**	**Sat 1/21/17**	**Mon 2/27/17**		**₹ 14,297.20**	**Yes**	**26**
28	Interface of XBEE S2B PRO	3 days	24 hrs	Sat 1/21/17	Tue 1/24/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Xbee S2B Pro[0.2],XCTU	₹ 3,466.00	Yes	26
29	Created common link between RFID, PIR and Arduino (Minor Errors)	8 days	64 hrs	Wed 1/25/17	Fri 2/3/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],PIR[0.2],RFID Cards[0.16],RFID Reader[0.16],Arduino IDE	₹ 4,849.60	Yes	28
30	***Gate***	***2 days***	***16 hrs***	***Sat 2/4/17***	***Sun 2/5/17***	***Mohd. Ehtheshamuddin[50%],Syed Abdul Wahab[50%]***	***₹ 0.00***	***No***	***29***
31	Started Database Creation using MySQL	3 days	24 hrs	Mon 2/6/17	Fri 2/10/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino IDE	₹ 1,500.00	Yes	30
32	Resolved errors between RFID and Arduino	3 days	24 hrs	Sat 2/11/17	Tue 2/14/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino UNO[0.1],RFID Cards[0.16],RFID Reader[0.16],Arduino IDE	₹ 2,281.60	Yes	31
33	**Internals Break**	**5 days**	**40 hrs**	**Wed 2/15/17**	**Tue 2/21/17**	**B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**32**
34	Finished Database Creation using MySQL	2 days	16 hrs	Wed 2/22/17	Thu 2/23/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],MySQL	₹ 1,000.00	Yes	33
35	Error Rectification for PIR and Arduino	2 days	16 hrs	Fri 2/24/17	Mon 2/27/17	B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],PIR[0.2],Arduino UNO[0.1],Arduino IDE	₹ 1,200.00	Yes	34
36	**Appraisal - II**	**1 day**	**8 hrs**	**Tue 2/28/17**	**Tue 2/28/17**	**B.Shafat Mohammed Khan[34%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**35**
37	**Software Module**	**17 days**	**136 hrs**	**Wed 3/1/17**	**Fri 3/24/17**		**₹ 22,836.60**	**Yes**	**36**
38	Resolving errors of Arduino with Netbeans	2 days	16 hrs	Wed 3/1/17	Thu 3/2/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],NetBeans	₹ 1,000.00	Yes	36
39	Data transfer from Arduino Mega to Arduino Uno through Xbee Router and Co-Ordinater	2 days	16 hrs	Fri 3/3/17	Mon 3/6/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino MEGA[0.5],Arduino UNO[0.1],Buzzer[0.5],LED[0.5],PIR[0.2],RFID Cards[0.16],RFID Reader[0.16],Xbee S2B Pro[0.2],Arduino IDE,XCTU	₹ 7,038.10	Yes	38

Batch A6

ID	Task Name	Duration	Work	Start	Finish	Resource Names	Cost	Critical	Predecessors
40	Reading of Data from Xbee and transmit it as a stream of 8 bits	3 days	24 hrs	Tue 3/7/17	Thu 3/9/17	B.Shafat Mohammed Khan[23%],Mohd. Ehtheshamuddin[39%],Syed Abdul Wahab[39%],Arduino UNO[0.1],Xbee S2B Pro[0.2],Arduino IDE,XCTU	₹ 3,598.00	Yes	39
41	Interface of Software and Hardware	10 days	80 hrs	Fri 3/10/17	Fri 3/24/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%],Arduino MEGA[0.5],Arduino UNO[0.1],Buzzer[0.5],LED[0.5],PIR[0.2],RFID Cards[0.2],RFID Reader[0.2],Xbee S2B Pro[0.2],Arduino IDE,XCTU,MySQL,NetBeans	₹ 11,200.50	Yes	40
42	**Appraisal 3**	**1 day**	**8 hrs**	**Mon 3/27/17**	**Mon 3/27/17**	**B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]**	**₹ 0.00**	**No**	**41**
43	**Reporting**	**8 days**	**64 hrs**	**Tue 3/28/17**	**Thu 4/6/17**		**₹ 0.00**	**Yes**	**42**
44	Thesis Reporting, Organising Correction and Approval	7 days	56 hrs	Tue 3/28/17	Wed 4/5/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	42
45	Printing of Project Report	1 day	8 hrs	Thu 4/6/17	Thu 4/6/17	B.Shafat Mohammed Khan[33%],Mohd. Ehtheshamuddin[33%],Syed Abdul Wahab[33%]	₹ 0.00	Yes	44
46	**Finish**	**0 days**	**0 hrs**	**Fri 4/7/17**	**Fri 4/7/17**	**B.Shafat Mohammed Khan,Mohd. Ehtheshamuddin,Syed Abdul Wahab**	**₹ 0.00**	**No**	**45**


Start
Start: Mon 8/1/16
ID: 1
Basic Knowledge
Start: Mon 8/1/16 ID: 2
Finish: Fri 8/26/16 Dur: 20 days
Definition of Project
Start: Mon 8/1/16 ID: 3
Finish: Tue 8/2/16 Dur: 2 days
Specifications
Start: Fri 8/19/16 ID: 5
Finish: Wed 8/24/16 Dur: 4 days
Literature Survey
Start: Wed 8/3/16 ID: 4
Finish: Thu 8/18/16 Dur: 12 days
Basic Work
Start: Sun 9/4/16 ID: 10
Finish: Fri 10/21/16 Dur: 35 days
Internal 1 Break
Start: Thu 9/1/16 ID: 9
Finish: Fri 9/2/16 Dur: 2 days
Tech Mahindra Interviews
Start: Sun 8/28/16 ID: 8
Finish: Wed 8/31/16 Dur: 4 days
Planning
Start: Thu 8/25/16 ID: 6
Finish: Fri 8/26/16 Dur: 2 days
Project Seminar
Start: Sat 8/27/16 ID: 7
Finish: Sat 8/27/16 Dur: 1 day
Arduino IDE (Basic Programs)
Start: Sun 9/4/16 ID: 11
Finish: Mon 9/19/16 Dur: 12 days
CapGemini interview
Start: Tue 9/20/16 ID: 12
Finish: Wed 9/21/16 Dur: 2 days
XCTU Software (Basics)
Start: Fri 9/23/16 ID: 13
Finish: Wed 10/5/16 Dur: 9 days
MySQL (Basic Databases)
Start: Thu 10/6/16 ID: 14
Finish: Fri 10/21/16 Dur: 12 days
Internal 2 Break
Start: Mon 10/24/16 ID: 15
Finish: Wed 10/26/16 Dur: 3 days
Purchase of Components
Start: Sat 1/7/17 ID: 20
Finish: Sat 1/7/17 Dur: 1 day
Survey on Specifications on Components
Start: Fri 12/30/16 ID: 19
Finish: Fri 1/6/17 Dur: 6 days
Interfacings
Start: Fri 10/30/16 ID: 18
Finish: Thu 1/19/17 Dur: 57 days
Major Work
Start: Fri 12/30/16 ID: 17
Finish: Fri 4/7/17 Dur: 74 days
Semester Exams
Start: Thu 10/27/16 ID: 16
Finish: Thu 12/29/16 Dur: 46 days
Work on Basics of RFID with Arduino
Start: Sun 1/8/17 ID: 21
Finish: Mon 1/9/17 Dur: 2 days
Interface of RFID with Arduino
Start: Tue 1/10/17 ID: 22
Finish: Wed 1/11/17 Dur: 2 days
Interface PIR with Arduino
Start: Thu 1/12/17 ID: 23
Finish: Mon 1/16/17 Dur: 3 days
Work on Configurations of XBEE S2C PRO
Start: Tue 1/17/17 ID: 24
Finish: Tue 1/17/17 Dur: 1 day
Work on Configurations of XBEE S2B PRO
Start: Wed 1/18/17 ID: 25
Finish: Thu 1/19/17 Dur: 2 days
Gate
Start: Sat 2/4/17 ID: 30
Finish: Sun 2/5/17 Dur: 2 days
Created common link between RFID, PIR a
Start: Wed 1/25/17 ID: 29
Finish: Fri 2/3/17 Dur: 8 days
Interface of XBEE S2B PRO
Start: Sat 1/21/17 ID: 28
Finish: Tue 1/24/17 Dur: 3 days
Hardware Module
Start: Sat 1/21/17 ID: 27
Finish: Mon 2/27/17 Dur: 28 days
Appraisal - I
Start: Fri 1/20/17 ID: 26
Finish: Fri 1/20/17 Dur: 1 day
Started Database Creation using MySQL
Start: Mon 2/6/17 ID: 31
Finish: Fri 2/10/17 Dur: 3 days
Resolved errors between RFID and Arduin
Start: Sat 2/11/17 ID: 32
Finish: Tue 2/14/17 Dur: 3 days
Internals Break
Start: Wed 2/15/17 ID: 33
Finish: Tue 2/21/17 Dur: 5 days
Finished Database Creation using MySQL
Start: Wed 2/22/17 ID: 34
Finish: Thu 3/23/17 Dur: 2 days
Error Rectification for PIR and Arduino
Start: Fri 2/24/17 ID: 35
Finish: Mon 3/27/17 Dur: 2 days
Reading of Data from Xbee and transmit it
Start: Tue 3/7/17 ID: 40
Finish: Thu 3/9/17 Dur: 3 days
Data transfer from Arduino Mega to Arduii
Start: Fri 3/3/17 ID: 39
Finish: Mon 3/6/17 Dur: 2 days
Resolving errors of Arduino with Netbeans
Start: Wed 3/1/17 ID: 38
Finish: Thu 3/2/17 Dur: 2 days
Software Module
Start: Wed 3/1/17 ID: 37
Finish: Fri 3/24/17 Dur: 17 days
Appraisal - II
Start: Tue 2/28/17 ID: 36
Finish: Tue 2/28/17 Dur: 1 day
Interface of Software and Hardware
Start: Fri 3/10/17 ID: 41
Finish: Fri 3/24/17 Dur: 10 days
Appraisal 3
Start: Mon 3/27/17 ID: 42
Finish: Mon 3/27/17 Dur: 1 day
Reporting
Start: Tue 3/28/17 ID: 43
Finish: Thu 4/6/17 Dur: 8 days
Thesis Reporting,Organising Correction anc
Start: Tue 3/28/17 ID: 44
Finish: Wed 4/5/17 Dur: 7 days
Printing of Project Report
Start: Thu 4/6/17 ID: 45
Finish: Thu 4/6/17 Dur: 1 day
Finish
Start: Fri 4/7/17
ID: 46
Project: Batch A6 Microsoft Proje
Date: Wed 4/12/17
Critical
Noncritical
Critical Milestone
Milestone
Critical Summary
Summary
Critical Inserted
Inserted
Critical Marked
Marked
Critical External
External
Project Summary
Highlighted Critical
Highlighted Noncritical

Details of project and relevance to environment, safety, ethics, cost and

Mapping with PO and PSO with justification.

Title of Project	Roll No of Students	Project Super-visor	Relev-ance to Enviro-nment	Relev-ance to Human Safety	Relev-ance to Ethics	Cost(hard ware or software cost)	Type (Application , Product, Research, Review)
Design of Wireless Sensor Network Based Smart University	1604-13-735-045 1604-13-735-046 1604-13-735-055	Salma Fauzia	3*	3*	3*	24,500 (Hardware and software project)	Application Based Product

3 – Highly Relevant 2- Moderately Relevant 1 – Less Relevant - Not Relevant

Mapping with POs and PSOs

Implementation Details:	The Project is based on the Concept of Wireless Sensor Networks in which Monitoring and Controlling of Attendance is carried out using Arduino board with ZigBee, PIR Sensor, RFID Reader and RFID Cards along with the Web Server designed by us.
PO1	Project deals with the deployment of a Wireless Sensor Network involving the basic concepts of Wireless and Data Communication.
PO2	The problem of marking attendance at universities was investigated and also existing methods of attendance monitoring have been analyzed.
PO3	The project deals with reducing the burden of faculty in marking and maintaining the paper records of attendance.
PO4	**NA**
PO5	We have used the latest technologies available in the market like ZigBee, RFID etc.
PO6	Better and efficient way of marking, retrieving and maintaining the attendance.
PO7	**NA**
PO8	The attendance marked using this system cannot be tampered with. Hence, provides a flavor of morals and ethics.
PO9	The project involves team work and hence we have learnt on how to lead the group and support each other in the working of the project.
PO10	Project report was prepared and submitted which enhanced our technical writing abilities and PowerPoint presentation was given to improve communication (Spoken and Written) skills.
PO11	Project Management Fundamentals were utilized. And the project was developed with minimum cost and at maximum efficiency.
PO12	The project contributes to lifelong learning in the domain of Wireless Sensor Networks as there is wide scope for masters and research in the said field. It also provides jobs opportunities in pertinent fields which involve the usage of ZigBee and RFID along with Arduino.

PSO1	**NA**
PSO2	The project is based on ZigBee concept using Arduino board which is the latest technology utilizing AtMega Processors which comes under embedded systems domain and hence is aligned to PSO 2.
PSO3	The hardware module after authenticating the attendance of the person and sends it to ZigBee module wirelessly which utilizes the wireless communication protocols and hence is aligned to PSO3.
PSO4	The hardware and software have been installed. And the software has been configured using the JAVA programming language hence enhancing our software programming skills.

Program Outcomes

PO1: **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2: **Problem analysis**: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12: Life-long learning: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1: The ECE Graduates will be Equipped with knowledge of complete design flow from specification to silicon in areas of both digital and analog VLSI Design and will be able to work in IC Design companies.
PSO2: the ECE Graduates will be Equipped with microprocessor and microcontroller based system design skills and can work as design and verification engineers in the area of Embedded Systems Design
PSO 3: The ECE Graduates will be able to apply engineering knowledge for design and implementation of projects pertaining to signal processing and Communications
PSO 4: The ECE Graduates will be Equipped with necessary soft skills, aptitude and technical skills to work in the software industry and IT sector.