Chapter 1

INTRODUCTION TO 6LoWPAN BASED LEAK DETECTION

Water is an abundant resource and one of the most essential to the living organism if not the most essential. However, water scarcity is a prevalent issue in many a places across the world. Reasons for water scarcity maybe draining up of rivers, reduction in the level of ground water or water leakages in households and so on. The system focuses on one aspect of the above mentioned negative factors and that is the water leak. Water leakages in household as well as industry results in an imbalance in the amount of water received and the amount of water used. Experts reveal that 30% of the rain water is absorbed by the system while there is a 15% drop in the amount of water used because of the water leakages. Also water leaks can quickly cause subsidence of land including footpaths, driveways and gardens. Internal water leaks from pipes, tanks and taps cause an untold amount of damage to the home including damage to walls and floors; costing thousands to rectify. The main challenge posed to the system is the communication between the transmitting and the receiver mote.

The system aims to be implemented with almost no input from the user. The system that has been implemented is a small-scale version of the fully blown system. This system uses a concept called 6LoWPAN that is a communication protocol given in the IEEE 802.15.4 standard to ensure communication occurs over the air with low power consumption. This system shows how a leak would be detected and how the system would respond to the user given such a scenario. To ensure no damage is caused, high precision requirements are paramount.

Using the input from the sensor, the transmitting mote continually sends the data if a leak is detected; the receiver mote using the internet connection sends a notification to the user through email and through text message. Upon a leak, an alarm is also triggered to provide auditory notification.

A website is also developed for the users to analyze the statistics of the water leakages. The website is responsive and can be accessed from any device. Each user can view his/her statistics using the login credentials given to him. Monthly and yearly statistics of the leakages along with the entire leakage history is provided.

1.1 Terminology

Terminologies related to the system have been explained in this section.

- **6LoWPAN:** 6LoWPAN stands for IPv6 over Low power Wireless Personal Area Network. It is a low power communication protocol defined in IEEE 802.15.4 standard. Devices in the 6LoWPAN network communicate with each other through IPv6 and can be arranged in either star network or mesh network. Edge router in this network acts as the gateway for internet and as the master node in star network. This node is responsible for the mapping of each node in the network with an identifier. If a device outside the network wants to access a device in the network, the edge router provides the connection between them.

- **MSP430:** The MSP430 is a mixed-signal microcontroller family from Texas Instruments. MSP430 is designed for low cost and, specifically, low power consumption embedded applications.

1.2 Purpose

The purpose of the system is to detect a water leak in a pipe(s). A new idea has been proposed here to make this an automated system. This system takes input as the rate of water flow into the flow sensor and the output from the system is the notification to the user that a water leak has been detected near the flow sensor.

1.3 Scope

The scope of the system mainly lies within household. It mainly deals with water leak detection which is the future of connected household. The system reduces the human effort for searching where the leakage is and reduces the amount of water wasted before leakage is fixed. This approach is simpler and faster than human reactions. The other industries where this system can be useful is small and heavy industries where water usage is at a maximum or otherwise.

1.4 Motivation

Water although being an existential need for humans, is one of the most under prioritised and over abused commodity. Leakage of water is a major issue which needs to be given due attention as water is becoming an increasingly scarce resource.

Water leakage not only wastes the resource but also increases the cost incurred. It also has several other consequences:

- Water main supply leaks can quickly cause subsidence of land including footpaths, driveways and gardens.
- Internal water leaks from pipes, tanks and taps cause an untold amount of damage to the home, wastage of water. While there are millions without water, leakage – should be of paramount importance.
- Water leakage creates a risk of a multitude of health problems affected by dampness.

Hence water leakage conservation is of extreme importance and when the system developed is scaled for entire household, both water and money can be saved. Also providing the users with the statistics can make them take decisions as to whether just stop the leakage or replace the whole pipe.

1.5 Literature Survey

The book [1] is the first released book on 6LoWPAN technology. This book explains the technology in depth, the history and standardization, its relation to other standards, architecture, topologies, addressing format and so on. It provides complete theoretical knowledge of the 6LoWPAN technology.

In the paper [2] clear emphasis is provided towards the use of 6LoWPAN technology over other alternatives. It explains how different features of 6LoWPAN like interoperability, security, neighbour discovery, header compression, fragmentation and reassembly and stateless auto configuration are implemented and why it is better than its alternatives. It also discusses the advantages of the usage of IPv6 technology for communication providing it a long lifetime.

Further research in different applications in 6LoWPAN studied [5, 6] explains the 6LoWPAN technology from a practical perspective. A comparative study of the power line communication and 6LoWPAN technology is also provided to know the dominance of 6LoWPAN. Also the energy management perspective where the need for batteries can be removed completely and security perspective using intrusion detection system is also explained.

The manual [3] provides complete description of the hardware C-Mote used in the system. The hardware architecture, different components used in the device and their uses have been explained along with the pin configuration. Different sources of power that can be used with the device are also explained.

The paper [4] provides an approach very similar to the system that is implemented. The paper focuses on detection of quality of water using the pH of water and communicating that data using Constrained Application Protocol. It also explains the detection of consumption rate of water.

1.6 Problem Statement

The project focuses on tackling the severe issues of water leakage by means of using sensors and C-Motes. The sensors are used to indicate the occurrence of the leakage of water and send the information to the motes. The motes communicate the information to the edge router which sends the information to the user and uploads the data in the database. User can analyze this data from the web application using the statistics provided.

For providing seamless connectivity of these low power devices to the Internet, an upcoming wireless networking technology known as 6LoWPAN is being used.

1.7 Objectives

Objectives of the system are,

- The detection of the water leak.
- Transmitting the data to the receiver communicating via the IEEE 802.15.4 standard.
- The system must also provide an alert to the user, through email and phone message.

1.8 Methodology

The system methodology consists of four major modules- Input Module, Communication Module, Coordinator Module, User module

- **Input module:** Reads the data from the flow sensor, converts the data into digital form and transmits it to the mote**.**

- **Communication module:** The sensor readings are interpreted and packets are created and transmitted to the corresponding motes.
- **Coordinator mote module:** Gathers data from the transmitter mote and transmits it to the cloud/server and sends notification to the user via email and text message.
- **User module:** Acts as an interface for the users to receive notifications, monitor the state of the system and provides data visualization.

1.9 Organization of the Report

This report focuses on the development of the water leak detection and conservation system using 6LoWPAN technology. The main body of the report is preceded by a table of contents including a list of figures, tables and glossary. The body of the report contains an introduction to 6LoWPAN based leak detection system and a literature survey done at the beginning of the project to collect requirements and product use cases. This is then followed by high-level design which focuses on developing the system architecture and a detailed design where the system is broken down into modules. The report also contains details of development and deployment environment used during the implementation of the project. The main body of the report is followed by reference section where the papers used for building the system are mentioned. The organization of the rest of the report is as follows:

Chapter 1 includes the Introduction, which provides an overview of the 6LoWPAN based Leak Detection, purpose, scope, problem statement, literature survey, objectives, brief methodology and the organization of the report.

Chapter 2 consists of Theory and Concepts, which explains different concepts that play a crucial role in the development of the system.

Chapter 3 includes a Software Requirements Specification, which explains the user characteristics, assumptions, dependencies, constraints and functional requirements.

Chapter 4 consists of the Design, which explains architectural strategies, system architecture and flow of data in the system through the data flow diagrams.

Chapter 5 consists of the Implementation, which explains the programming language, development environment and code conventions followed during the project. This chapter also has various difficulties encountered in the project and how they were overcome.

Chapter 6 is Testing, which explains the test environment and briefly explains the test cases which were executed during unit testing, functional testing and integration testing. The entire system testing is also done and all related details are listed in this chapter.

Chapter 7 is Results and Discussions, which focuses on the output of the project along with the screenshots and the pictures of the implementation. It is also helpful in assessing the expected output to the actual output. This chapter also includes snapshots of the system that is used.

Chapter 8 is the Conclusion, which gives the outcome of the project work carried out and also brings out the limitations of the project and elaborates on future enhancements to improve the Water Leak Detection system.

Chapter 2

THEORY AND CONCEPTS OF 6LoWPAN BASED LEAK DETECTION

The project makes use of different types of models used for efficiently programming the C-Motes. Few of the parameters used for leak detection and transmission are MSP430-GCC compiler, the Makefile, interrupt service routines, serial port communication etc. Combination of all these types are used for leak detection, transmission and notification.

2.1 MSP430-GCC

MSP430-GCC is a cross compiler for MSP430 embedded devices from Texas Instrument. It can be installed within Code Composer Studio or as a standalone package. Upon installation, a various number of header files are downloaded into the system for each of the device in the MSP family. Each header file contains various macros defined to access and manage the components and registers within the device. Macros for using the built-in functions are also provided in the header files. The header file corresponding to the device being used should be included in the program that is to be uploaded to the device. When the program is compiled into a device, the header included in the program is matched with the components in the device and if a mismatch is present an error is prompted.

The hardware device used in the system is C-Mote which uses msp430f2618 system on chip from Texas Instruments which has 8KB RAM with 116KB flash memory. Hence the C-Mote has to be programmed using the macros defined for msp430f2618. MSP430-GCC version required for the system is gcc-msp430 4.6 or higher.

2.1.1 Makefile

To compile and upload programs to MSP devices like C-Mote, a file called Makefile is used which contains a set of directives used with the build automation tool called ‘make’. Makefile is used to compile programs which uses various number of header files with a single command. If the main program uses functions from several other files, only the header files should be included in the main program. Commands for

compiling those files should be included in the Makefile. Also the device name (example: msp430f2618) and the file path should be included in each command.
Makefile has its own syntax style and the file should be named ‘Makefile’ without any extension. Makefile also provides option to include other Makefiles into it just like any other programming language. Identifiers can be declared in the Makefile to store static data. Each variable can be accessed using “$(variable_name)” notation. Makefile provides option to declare labels under which commands to compile other files should be mentioned. When uploading a program into the device, the label name should be specified which will include all the commands defined under it.

For example, to upload a program named ‘tx.c’, the command “make flash” should be used in the terminal. The ‘make’ command will search the Makefile for compilation and finds the program name, label name and other details needed for compiling and uploading the program to the device.

2.2 Interrupt Service Routine

Interrupt Service Routine also called as Interrupt Handler is a callback subroutine which is triggered upon the occurrence of an interrupt. Interrupt service routines are defined to handle the occurrence of certain hardware events. When an interrupt occurs, the operating system saves the current state in the status registers and executes the interrupt service routine. Once the interrupt handler is successfully executed, the system comes back to its original state using status registers.

In this system, interrupt service routine is used to count the number of pulses from the flow sensor. When a leakage is detected, the sensor sends pulses of signals as output. This acts as interrupts in the C-Mote and the interrupt handler counts these pulses. If this count crosses a specific value, data is transmitted to the coordinator mote.

2.3 Minicom

Minicom is a terminal emulation program for Unix-like operating systems. During the development of the system, minicom can be extensively used for testing and debugging purposes. To obtain the output of the devices with no display capabilities, minicom is used and serial data from the devices are printed. Minicom provides various options like changing the user interface, setting the port to read the data from, setting the baud rate at which data is to be read. This rate varies depending on the device specifications. C-Mote writes data at a baud rate of 4800 bits per second. Also minicom provides an option to save the configuration and the output obtained as files.

2.4 Serial Communication

Serial Communication is a common way to communicate between a computer and various electronic devices. The device connected to the computer or laptop through USB can transfer data through the serial port. To read or write data from or to the external devices, the port and baud rate of the device should be available

2.4.1 Port

Port acts as an interface between the computer and the device connected to it. Communication port used when a device like C-Mote is connected to the computer is /dev/ttyUSBx where x is a number. At any time, only one program can access a particular port.

2.4.2 Baud Rate

Baud Rate is the rate of transfer of information in a communication channel. Baud rate is measured in terms of bits per second. C-Mote transfers data at a baud rate of 4800 bits per second.

Chapter 3

REQUIREMENTS SPECIFICATION OF 6LoWPAN BASED LEAK DETECTION

Requirement process is defined as a list of activities which are performed in the requirement phase to produce a high quality document containing the Software Requirement Specification (SRS). This phase is crucial and critical to the success of any project. Requirements must be testable, quantifiable, defined to a level of detail sufficient for the system design related to the identified business opportunities and needs. The Software Requirement Specification is a complete description of the behavior of the system to be developed. This lays down the foundation for continued product evaluation.

3.1 Overall Description

This section provides a description of the general factors that affect the system and its requirements. A background for the set of system requirements is provided, which are defined in detail in section 3.2, making them easier to understand.

3.1.1 System Perspective

In this project, a scheme has been proposed to bring into use 6LoWPAN, which is a low-power wireless networking technology that adds the potential for end-to-end communication, control, and monitoring of devices from anywhere on the globe.
The project focuses on tackling the severe issues of water leakage by means of using C-Motes which are programmed to operate on 802.15.4 link layer using 6LoWPAN.

3.1.2 System Functions

The primary function of the system is to detect water leak(s) anywhere in the pipes used in common households and notify the owner through mail and text message.

3.1.3 Constraints

- The range up to which the motes can communicate is limited to approximately 100m and is subject to obstruction.
- Only two flow sensor can be used per C-Mote.
- The flow sensor works at a voltage of 5V whereas C-Mote works at 3.3V and hence a logic level converter is required for every flow sensor. Also a 5V supply should always be connected to the slow sensor.

3.1.4 Assumptions and Dependencies

The assumptions and dependencies are listed below:

- The system is assumed to work without taking into account environmental factors such as temperature, pressure, humidity etc.
- The overhead tank is assumed to be at a height greater than the heights at which the valve and the flow sensor are placed.
- The C-Mote attached to a gateway such as laptop, mobile or Beagle bone black is assumed to be connected to the Internet.

3.2 Specific Requirements

This section contains software requirements specified to a level of detail sufficient to aid the designers to design a system such that these requirements are satisfied. It also helps design test cases to confirm the system satisfies those requirements.

3.2.1 Functionality Requirements

- The speed at which the water flows through the pipes, valves or the sensor is dependent on the height at which the overhead tank is placed.
- The output of the flow sensor is the input to the transmitter C-Mote which determines the rate at which the water is flowing through the sensor.
- Flow sensor gives output in the form of digital pulses which is fed to the C-Mote through the GPIO pin P2.4 and converted into a numerical value.
- The transmitter mote triggers a buzzer and transmit the data to receiver mote.
- The data is uploaded to database and user is alerted via email and message.

3.2.2 Performance Requirements

The performance characteristics of the system have been outlined in this section. The resources which are being utilised have been explained in detail. A power supply source giving voltage of 5V must be used to power the flow sensor. In addition to this an LLC must be used to connect the flow sensor to the C-Mote. Also, the following requirements should be embraced by the system:

- **Reliability:** The system should be extremely accurate in terms of generating the trigger.
- **Ease of Use:** Since the owner or the user is assumed to be a non-technical person, the system should be handy and easy to use.

3.2.3 Supportability

The system is designed for domestic household. The diameter of the flow sensor should be proportionate to that of the pipe used for water supply. Also the system can be scaled by adding more hardware depending on the household requirement.

3.2.4 Hardware Requirements

- C-Mote v1.1 by CDAC
- Flow Sensor (YF-S201)
- Logic Level Converter
- 5V supply with breadboard
- UbiSense Buzzer
- Pipes, valves and trough

3.2.5 Software Requirements

- MSP430-GCC compiler and libraries
- Code Composer Studio IDE
- Minicom emulation tool
- Programming Language C, Python, MySQL

3.3 External Interface Requirements

The external Interface Requirements include User, Hardware, Software and Communication interfaces. These are as follows:

3.3.1 User Interfaces

The user interacts with the system through the web application Leakbabu, a website which has been designed to provide the leakage statistics for each user.

3.3.2 Hardware interfaces

The system can be deployed in any household environment. The flow sensor and the motes form the main portion of the system.

3.3.3 Software interfaces

The whole system is programmed using C on C-Motes. The output of the code run on the C-Motes can be checked on the terminal using the minicom emulation program. The system makes use of SMTP and Twilio services to send email and text message.

3.3.4 Communication interfaces

Communication between the motes takes place only in case of the event of leak detection.

Chapter 4

DESIGN OF 6LoWPAN BASED LEAK DETECTION

The Design phase is an integral part of software development process. It is a creative process in which the system organization is established in a manner which satisfies both functional as well as non-functional system requirements. Large Systems are decomposed into sub-systems and these sub-systems provide a related set of services. The output of this phase is a description of the Software architecture.

Several challenges faced during development of design modules such as Design considerations, assumptions, dependencies etc. are elaborated in the following sections.

The High Level and Low Level or Detailed Designs which have been used in the implementation of the "6LoWPAN Based Leak Detection and conservation" system have been discussed in detail in this chapter. The identification of any event of water leakage and notifying the owner/user subsequently is the most essential goal, however the performance of the system under real time constraints must also be achieved.

4.1 High Level Design

The high level design of the system includes design considerations, architectural strategies, system architecture, Data Flow Diagrams and Context Flow Diagrams. These are explained as below:

4.1.1 Design Considerations

Before starting to design a complete solution for the problem statement, several design issues that need to be considered. These include the assumptions and dependencies with respect to the software, any global limitations or constraints that have substantial impact on the system, the architectural strategies and the method or approach used for the development. The following section comprises all the above along with the system design that provides a high level overview of the functionality of the system.

4.1.1.1 Assumptions and Dependencies

The dependencies related to the system have been stated as below:

- The operating system which has been used for implementation is ContikiOS, which is open sourced operating system similar to Ubuntu.

- The programming languages used are C and Python.
- The hardware requirements for implementation are C-Mote (8KB RAM, 116 KB Flash, 8Mbit Serial Flash), laptop (512 MB RAM, Core 2 duo processor or higher).

4.1.1.2 General Constraints

The following constraints were considered while building the system:

- The system operates without the consideration of environmental factors.
- The range up to which the motes can communicate is limited to approximately 100m and is subject to obstruction.
- Only two flow sensors can be used per C-Mote.
- The flow sensor operates at a voltage level of 5V whereas the C-Mote operates at 3.3V, so an electronic component called LLC has to be used to connect the flow sensor to the C-Mote.

4.1.1.3 Development Methods

The development methods employed play a vital role in determining a large section of how the final system functions, and thus care is taken to ensure that the best practices, tools and equipments are being made use of. The system has been developed upon C-Motes by means of using C programming language. The main modules would be input, transmitter, receiver and a GUI for leakage statistics.

4.1.2 Architectural Strategies

This section describes the design decisions and strategies that affect the overall organization of the system and its higher-level structures. These strategies will provide insight into the key abstractions and mechanisms used in the system architecture.

4.1.2.1 Programming Language

- The whole system is dependent on programming in an open source environment. The MSP430 libraries are readily available to be installed for use. Code Composer Studio IDE from Texas Instruments has been used for development.
- MSP430-GCC based C language has been used extensively for the development of this project. C is a type less, structured and an imperative programming language. C allows the use of a number of libraries required for

operating sensor nodes. C has been used primarily for reading sensor data and establishing communication between the motes.

- Python has also been used for this project. It is an interpreted language having a design philosophy which emphasizes readability of code. It provides syntax which enables programmers to express concepts in fewer lines of code than possible in other contemporary languages like C++ or Java. Python has been used primarily for reading the program output of the motes from the terminal, to send the data from the system to the server and to notify the user through email and text message.
- MySQL is used for creating and managing the database. It is an open source relational database management system. phpMyAdmin is used to handle the administration of MySQL. Also HTML/CSS is used for creating a responsive web application for the users PHP is used to provide the connection between the web application and the database.

4.1.2.2 Future Plans

- Currently the system is implemented using a single flow sensor for a pipe in a small prototype which can be scaled to entire household.
- More sensors to be added to a single mote so as reduce the number of motes required.
- Solenoid Valve, a programmable valve to be used for shutting down the water supply in future.

4.1.3 System Architecture

Leak Detection system is decomposed into sub-systems that provide some related set of services. The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called Architecture design and the output of this design process is a description of the software architecture. The architectural design process is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communications between these components. The system architecture shows the blocks required for the system. Figure 4.1 shows the existing system architecture. Flow Sensor detects the rate of water flow. The transmitter mote (mote 1) collects the sensor data and generates a trigger in case of a leak. The receiver mote (mote 2) sends a notification to the user in the form of a text message as well as email via laptop.

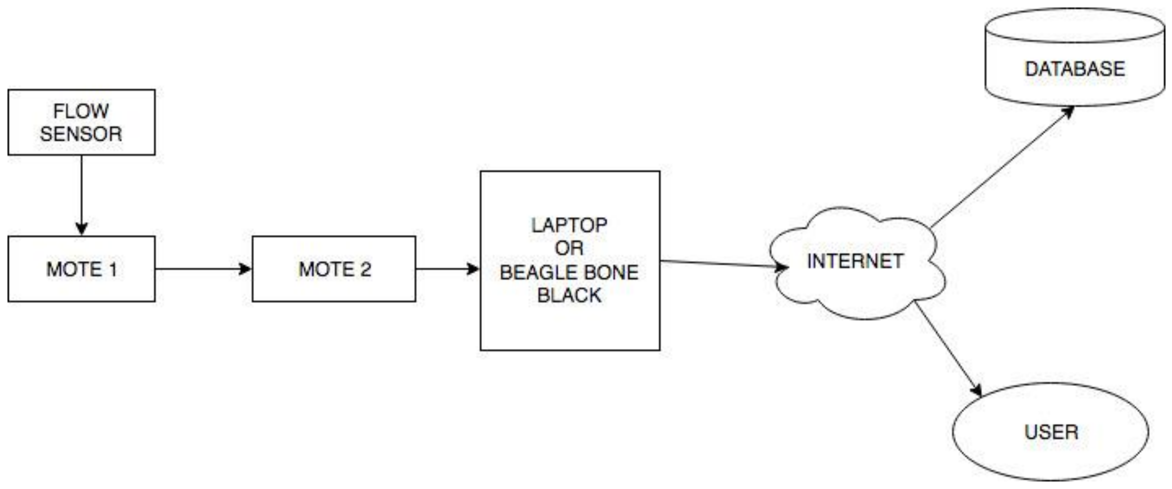


Figure 4.1 System Architecture

4.1.4 Data Flow Diagrams

A data flow diagram (DFD) is a pictorial representation of the flow of data through an information system. As opposed to a flowchart which depicts the control flow of the program, a DFD shows the flow of data through the course of the program. A DFD makes the process of picturing the structured design, less cumbersome. Invented by Larry Constantine, the DFDs are based on Martin and Estrin's "data flow graph" model of computation.

It is a common practice to draw a context-level Data flow diagram first, which depicts the interaction between the external entities and the system. The DFD is designed to show the different smaller components of the system and is also used to highlight the flow of data between those parts. This context-level DFD is then blown up to show more detail of the system being modelled.

The DFDs show the input to and output from a system. This includes information related to what the data is and its storage location. It does not indicate whether the processes will be carried out sequentially or in parallel, nor any information about the timing of processes. It also does not deal with control flow of information.

4.1.4.1 Data Flow Diagram - Level 0

The Level 0 diagram is a context-level data flow diagram, which shows the entire system and its interaction with external agents which are usually data sources or sinks. The context diagram indicates the whole system as a single process with no further information about the internal functioning of sub processes.

The Figure 4.2 shows the Level 0 Data Flow Diagram. In this level 0 DFD, two external entities have been identified i.e. Source and Destination. It shows the interaction in the Leak detection system.

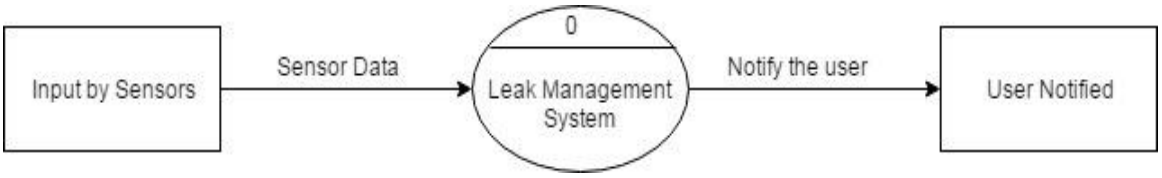


Figure 4.2 Data Flow Diagram Level - 0

4.1.4.2 Data Flow Diagram – Level 1

The context-level DFD is exploded into a Level 1 diagram that shows how the system is divided into constituent parts with the exact relationship between them.

The Figure 4.3 shows the DFD - Level 1 for System Process from the DFD - Level 0. It shows how the system is divided into separate sub-systems.

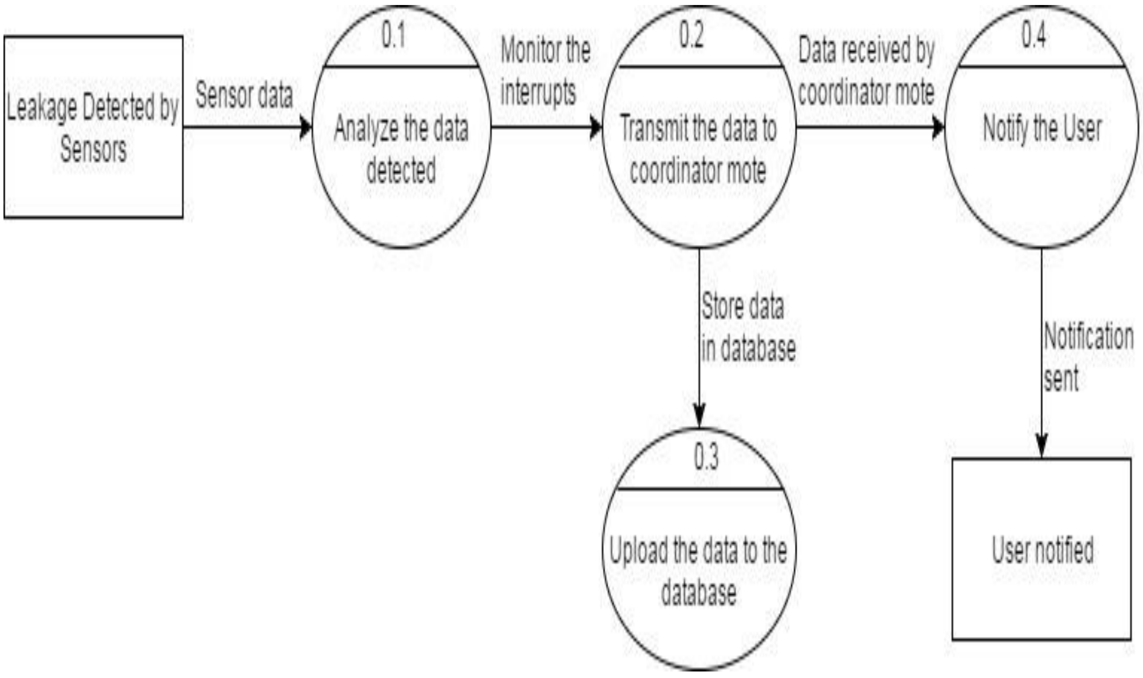

Leakage Detected by Sensors
Sensor data
0.1
Analyze the data detected
Monitor the interrupts
0.2
Transmit the data to coordinator mote
Data received by coordinator mote
0.4
Notify the User
Store data in database
0.3
Upload the data to the database
Notification sent
User notified

Figure 4.3 Data Flow Diagram Level - 1

4.1.4.3 Data Flow Diagram – Level 2

The Level 2 Data Flow Diagram gives the complete picture of all the sub modules in this system. Figure 4.4 shows the level 2 data flow diagram of 6LoWPAN based the leak detection system.

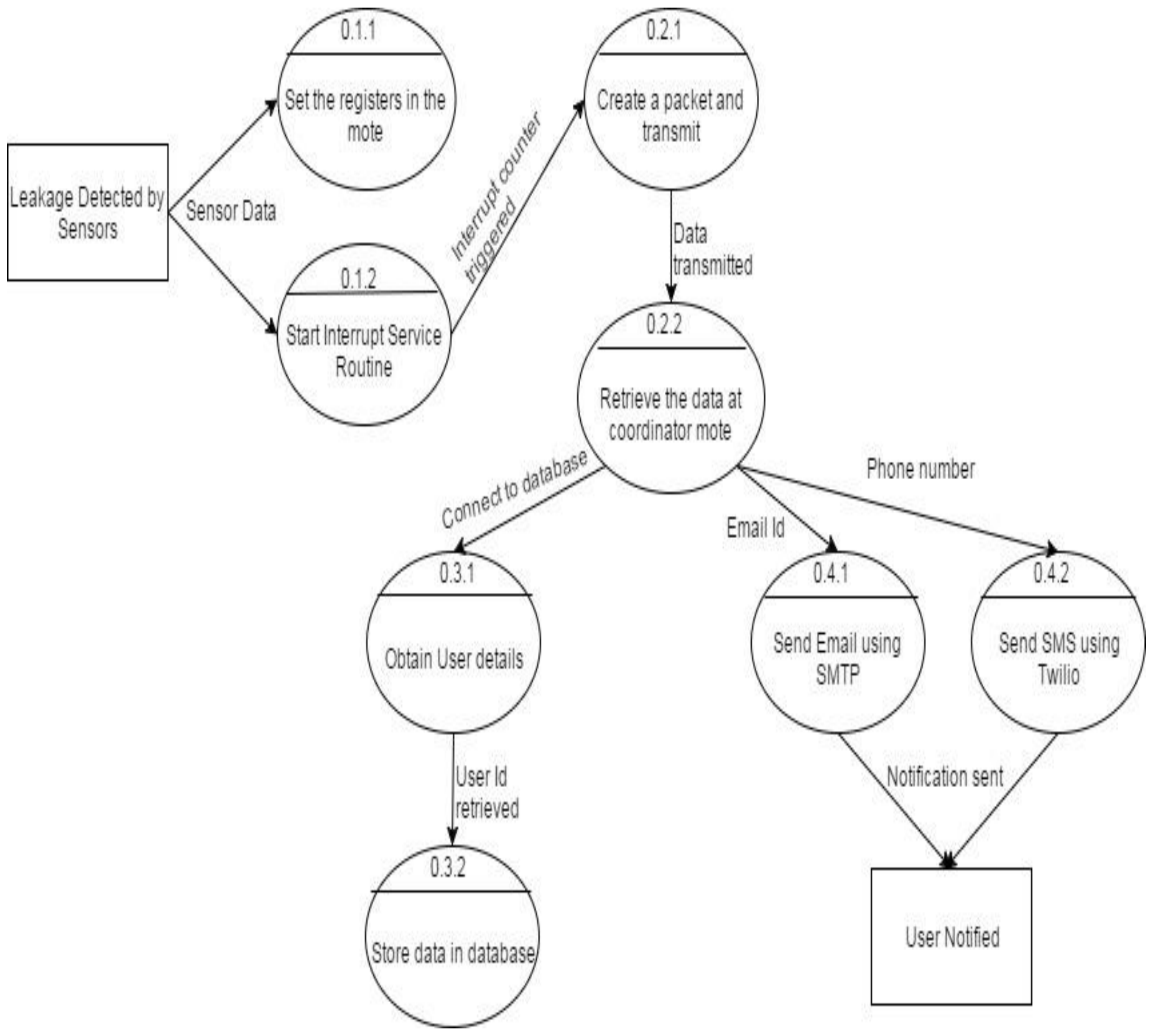


Figure 4.4 Data Flow Diagram Level - 2

4.1.5 Context Flow Diagram

In software engineering and systems engineering, there are diagrams that are used to represent all external entities that may interact with a system. These diagrams are referred to as Context Flow Diagrams. Figure 4.5 shows the highest level view of a system showing a possibly software-based system as a whole and its inputs and outputs from/to external factors.

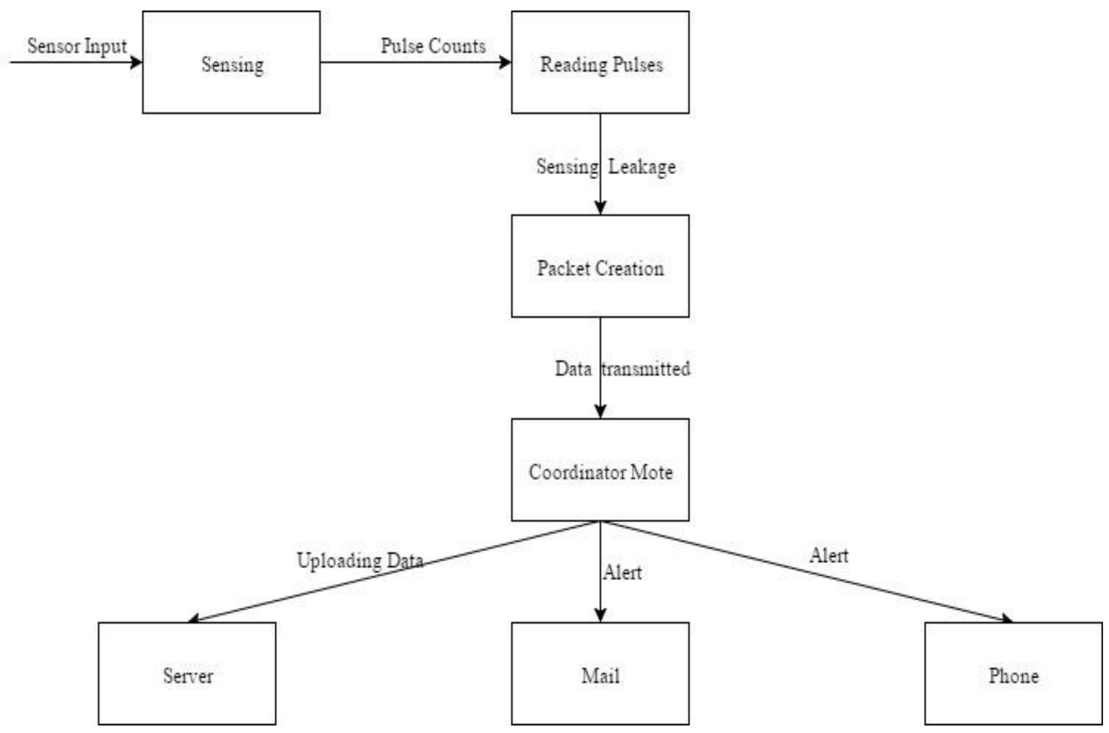


Figure 4.5 Context Flow Diagram of system

The system context diagram shows the interactions between the system, the actors with which the system is designed to interact and the context of the environment in which it operates. These diagrams are usually included in a requirements document. These diagrams must be read by all system stakeholders and should thus be written in plain language, so that the stakeholders can understand the items with the document. The contextual diagram of the overall system working is displayed in Figure 4.5.

4.2 Detailed Design

The detailed design of the system consists of the detailed system design and the functional description of each module.

4.2.1 Detailed System Design

This phase deals with the internal logic of each of the modules specified in high-level design in the previous section. In this phase further details, algorithmic design and in depth explanation of each of the modules is provided. Other low-level components and sub-components are also described. The structure chart provides the control flow amongst all the modules present in the system. This chapter also provides more details about the system modules by clarifying the details about each function with functionality, purpose, input, and output.

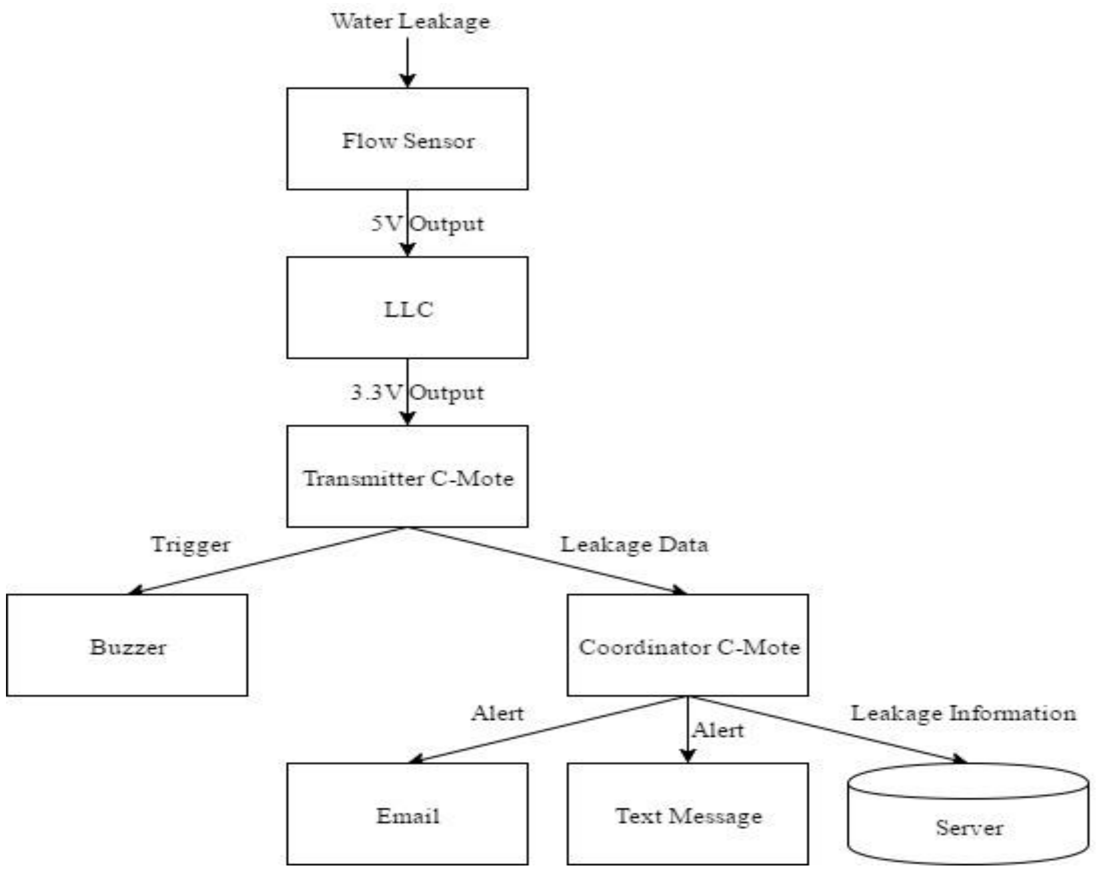


Figure 4.6 Structure Chart

An overall view of the structure of the system is presented in the Figure 4.6 which shows the interactions between each module and at what level each module are initiated.

4.2.2 Functional Description of Modules

This description of modules includes the description of sub-modules and other components that play a vital role in the system.

4.2.2.1 Input Module

This section is responsible for taking in real-time values from the flow sensor.

- **Purpose**

  The purpose of this module is to obtain the value of flow rate from the flow sensor and feed it to the transmitter C-Mote.
- **Functionality**

  The functionality of this module is to obtain the flow rate from flow sensor, which is in the form of digital pulses and convert it into a form suitable to be interpreted for further computation.

- **Input**

  The input to this module are real-time values of the flow-rate of the water through the pipe(s).
- **Output**

  The output of this module is used to interpret if there is an occurrence of any event of a leakage or not.

4.2.2.2 Transmission module

This section takes the output from the input module and transmits the data to the receiver C-Mote in case of occurrence of a leak.

- **Purpose**

  The purpose of this module is to create a packet and initialize it with the data that is to be transmitted.
- **Functionality**

  This module initializes the IEEE 802.15.4 header along with the transmitter and receiver node id and creates a packet and initializes it with the data that is to be transmitted to the receiver mote to indicate the occurrence of leakage.
- **Input**

  The input is the pulse count which is interpreted to indicate leakage.
- **Output**

  The output from this module are packets of data which are to be transmitted to the receiver mote.

4.2.2.3 Coordinator module

This section receives the data from the transmitter C-Mote, uploads it to the database server and notifies the user through SMS and mail.

- **Purpose**

  The purpose of this module is to upload the data to the server and notify the user.
- **Functionality**

  This modules takes the packets transmitted from the transmitter C-Mote and through an internet gateway such as a laptop or a mobile or a Beagle Bone Black, uploads the data to the server, sends mail to user' personal mail and uses Twilio to send an SMS to the user.

- **Input**

  The input to this module are the packets sent by the transmitter C-Mote.

- **Output**

  The output of this module are the mail and SMS sent to the user along with the data to be uploaded to the server.

4.2.2.4 User Module

This section serves the purpose of a GUI which basically shows the user the leakage statistics.

- **Purpose**

  The purpose of this module is to provide a GUI to the user which indicates the occurrence of any leak as well as the leakage statistics ordered in weekly and yearly manners.

- **Functionality**

  It is used to display to the user the real-time as well as pre-stored data or leakage statistics.

- **Input**

  The input to this module is the data sent from the receiver mote.

- **Output**

  Output of this module is the analytical data in the form of charts and graphs, which is the final output of the system.

Chapter 5
IMPLEMENTATION OF 6LoWPAN BASED LEAK DETECTION

The proposed system detects the leakages in the pipe and sends a trigger to the coordinator mote which contains a buzzer to provide audio notifications. The model uses 6LoWPAN technology to establish communication between the motes. The serial data from the coordinator mote is read and is uploaded to the database along with sending email and text message notifications to the user. Data visualization is provided for each user in a website based on the history of water leakages.

5.1 Details of language used

In this system C programming language is used in integration with the Code Composer Studio IDE by Texas Instruments to program the motes. C is the broadly used language to program the msp430 microcontroller based hardware devices. C utilizes the functional programming style and hence is suitable for structured programming. C provides constructs that map effectively to typical machine instructions, and thus it is very useful in applications like operating systems. C also provides syntax for defining interrupt service routines which plays a key role in this system.

Python programming language is used in addition to C to perform serial communication with the coordinator mote. Python is an interpreted and cross platform programming language with a very simple syntax definition. Python provides an easy way to do cross platform serial communication using pySerial module. This module hides the operating system specific peculiarities and provides a simple interface for programming the serial port. The code written with the pySerial module can run unmodified on Windows as well as Linux systems.

Code Composer Studio (CCS) is an integrated development environment (IDE) to develop applications for Texas Instruments embedded processors. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger and many other features. Code Composer Studio combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from Texas Instrument.

5.2 Details of Platform: C-Mote

The C-Mote is a Wireless Sensor Node (WSN) designed with a robust radio frequency front end and highly flexible power management system. This mote also supports many features which are suitable for WSN outdoor deployments & Solar Energy Harvesting. It is designed by the Center for Development of Advancement Computing (CDAC).

The main elements of the Mote are:

- TI's MSP430F2618 16-bit Ultra Low Power Microcontroller with 8KB RAM and 116KB Flash.
- TI's CC2520 2nd generation 2.4GHz IEEE 802.15.4 compliant transceiver.
- Connector for interfacing 2.4 GHz antenna.
- 8Mbit Serial Flash memory for storing the golden image of the hex file.
- Sensor Connector for interfacing UbiSense.
- DAC connector for interfacing the Ubi-DAC.
- USB interface for programming and data transfers with Host Computer
- Power through Li ion battery, solar connector or USB Connector.

5.2.1 Components of C-Mote

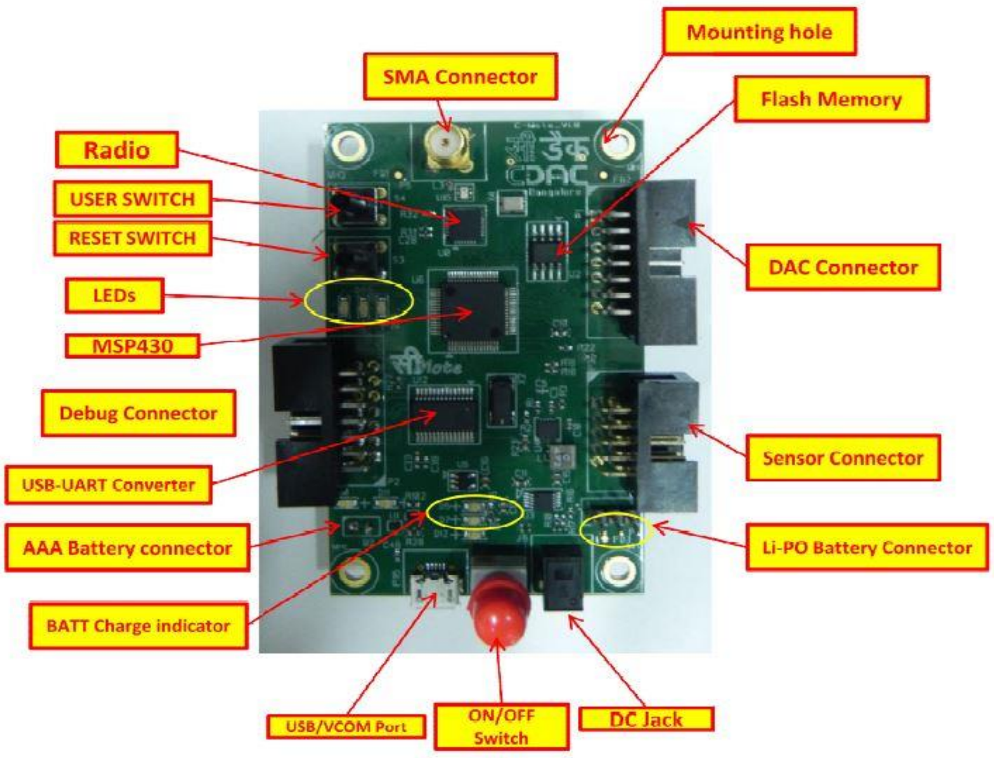


Figure 5.1 Components of C-Mote

USB Connector: This standard Micro USB connector (P15) can be used for connecting the Mote with PC for supplying power & data transfer. The USB interface can be used to transfer data between the mote and the computer with the help of UART to USB Bridge.

Sensor Connector: C-Mote provides two ports with 10-pin and 14-pin headers for sensor connection as shown in Figure 5.1. Each port provides connection for UART, General Purpose Input Output (GPIO), Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI).

User LEDs: Three user LEDs have been provided for configurable output indication from the MSP430 microcontroller. The default State of these GPIO pins are High (LED off). Driving these Pins low through program enables the LEDs. These LEDs can be used for any application as per user requirement.

5.2.2 Logic Level Converter

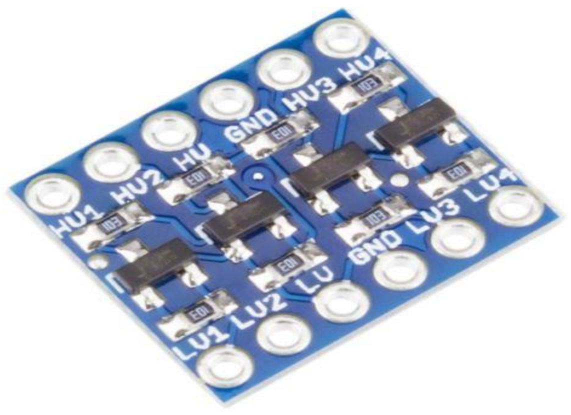


Figure 5.2 Bi-Directional Logic Level Converter

C-Mote operates at 3.3V whereas the flow sensor which is used in the system requires 5V. Hence there is a need to perform level shifting/conversion to protect the 3.3V C-Mote from 5V. The system uses Bi-Directional Logic Level Converter (BD-LLC) for this purpose which is powered by 5V through the sensor and 3.3V through the C-Mote. The output from the sensor is brought down from 5V to 3.3V and transferred to the C-Mote. There are four separate data channels on the BD-LLC, each capable of shifting data to and from high and low voltages. These pins are labelled HV1, LV1, HV2, LV2, HV3, LV3, HV4, and LV4. The number at the end of each label designates the channel of the pin, and the HV or LV prefix determines whether it’s on the high or low side of the channel.

5.2.3 UbiSense

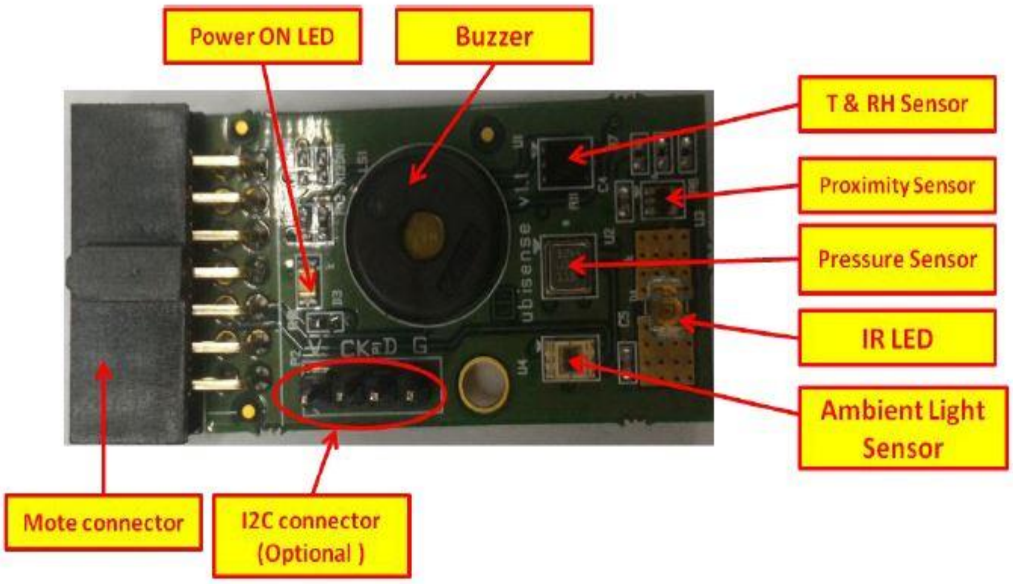


Figure 5.3 UbiSense

UbiSense is the sensor board developed by CDAC for testing and integrating various sensors with the C-Mote as shown in Figure 5.3. UbiSense contains various sensors that suits for WSN applications where the users can make use of sensor data by plugging in the module directly to the C-Mote. The sensors on UbiSense are I2C compatible and user can avail the advantage in interfacing without much trouble.

In the system, the UbiSense is used with the receiver mote to provide the audio notification to the user by triggering the buzzer when a leakage is detected.

5.2.4 Water Flow Sensor

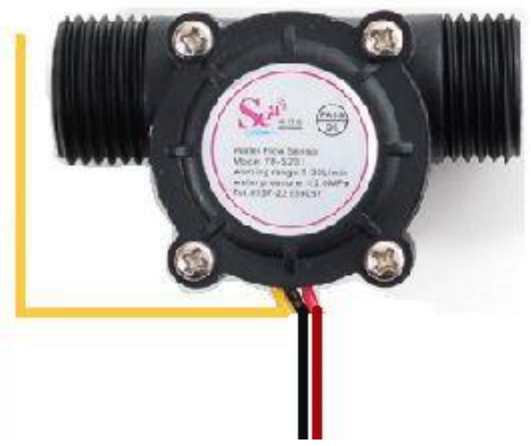


Figure 5.4 YF-S201 Water Flow Sensor

YF-S201 water flow sensor acts as a bridge that can be fir between the pipes and it contains a pinwheel sensor to measure how much liquid has moved through it. There is an integrated magnetic Hall Effect sensor that outputs an electrical pulse with every revolution. The Hall Effect sensor is sealed from the water pipe and allows the sensor to stay safe and dry.

The sensor comes with three wires: red (5V DC power), black (ground) and yellow (Hall Effect pulse output). By counting the pulses from the output of the sensor, water flow can be calculated. Each pulse is approximately 2.25ml and sensor transmits 450 pulses per litre and has a diameter of 0.78 inch. This is not a precision sensor, and the pulse rate does vary a bit depending on the flow rate, fluid pressure and sensor orientation.

5.3 Task Implementation

Implementation procedure of this system is split into various steps. Each step is necessary in order to achieve the end result. Steps are interdependent on each other. Step-wise approach gives a better clarity for development and implementation.

a. Sensing Leakage

Flow sensor used to detect water leakage acts as bridge between two pipes. When there is no leakage, the water inside the pipe would be stagnant. Hence the sensor does not send any signal to the C-Mote connected to it.

Step 1: When there is a leakage, there is a flow of water in the pipe and the pinwheel of the sensor rotates and sends the pulses as output.
Step 2: The pulses from the sensor will be at a voltage level of 5V which the C-Mote cannot process. Hence the pulses are sent to the logic level converter which reduces the voltage level to 3.3V and transmit it to the C-Mote.

b. Reading the pulses

Pin (P2.4) of the C-Mote is a GPIO pin which takes interrupts as input. The output signal from the logic level converter is connected to P2.4.

Step 1: Interrupts at the GPIO pin of the C-Mote are enabled and the CPU is halted till an interrupt occurs. This ensures that the device will be in sleep mode and do not consume much power when there is no leakage.

Step 2: When an interrupt occurs, the interrupt service routine programmed into the C-Mote is called which increments the pulse counter. When the count reaches specific value, the value is set to zero and the leakage data is transmitted. Also the buzzer is triggered to provide auditory notification.

c. Wireless Communication

Main focus of the project is the wireless communication of data between the devices. This is implemented in following steps.

Step 1: In this step, the IEEE 802.15.4 header is initialized along with the transmitter and receiver node identifier.

Step 2: In the second step, a packet is created and initialized with the data that has to be transmitted.

Step 3: In the final step, the packet is transmitted to the receiver mote. Also the LEDs in the C-Mote are toggled. The interrupt service routine is restarted and the C-Mote halts again.

d. Coordinator Mote

The Coordinator C-Mote is responsible for connecting the entire network to the internet and hence acts as a gateway. When a packet is received, it is parsed to obtain the data. This data acts as a trigger and the coordinator C-Mote writes this data onto the serial port of the laptop or any other alternative device connected to it at a baud rate of 4800.

e. User Notification

A python script runs infinitely on the laptop or the device connected to the coordinator C-Mote. Open the port where the coordinator C-Mote is communicating. If there is any data at this port, following steps are implemented.

Step 1: The current timestamp is uploaded to the database with the id of the user.

Step 2: A text message is sent to the user using Twilio web service.

Step 3: Also an email is sent to the user using simple mail transfer protocol.

Since this is an infinite process, loss of internet connection does not stop the program. Once internet is available the above steps are carried out.

f. Data Analysis

The website is built with the focus to provide users with their entire leakage information in the form of statistics and tables. The website is built such that it can be accessed by any device.

Every user has to register to the website and can login using those credentials. Once the user logs in, the dashboard displays the recent few leakages with the date and time. Also the user can view monthly and yearly statistics of leakages. A table containing entire leakage history is also provided.

Chapter 6

TESTING OF 6LoWPAN BASED LEAK DETECTION

In this section the results of the tests cases on the leakage detection, communication between the motes, and notification to the user are presented along with the user interface being tested.

The purpose of testing is to ensure that the resulting system meets the system requirements and there is transition of data flowing through each of the modules as well as in between one another.

6.1 Unit Testing

Unit testing provides a sort of living document. Unit tests help clients and other developers learn to use the module and gain a basic understanding of the system.

6.1.1 Testing Strategy

The testing strategy followed in the system for unit testing is indicated in the following points.

- The system is first broken into several units or modules during the design phase of the project.
- Review of the design specifications and source code for the units to be tested are carried out first.
- Unit test plan is created and a peer review is performed considering the common errors committed during development.
- A test "stub" is created to provide input to or receive output from the code module.
- The units are then compiled in the test environment for compilation errors and to check for any missing headers required for test plan execution.
- The tests are executed and information received from the tested software are compared with the expected values, as documented in the Unit test plan.
- When the unit is updated, the module is tested again and the results are recorded on the Unit Test Report Form.
- The final report form is archived when the unit is passes all tests.
- The Unit Test Report forms related to a given sub-system are compiled into a Summary Unit Test Report.

6.1.2 Test Cases

Unit test cases aim at removing the bugs and errors at the early stage of the development cycle. Each unit is responsible for a specific feature of the system and unit test cases aim at testing the behavior of the unit upon appropriate as well as inappropriate use.

6.1.2.1 Unit Test Case 1

C-Mote

Table 6.1 shows the test result of the most basic requirement of the project which is the working of the C-Mote. When the C-Mote is powered, the charge indicator LEDs should glow.

Table 6.1 Unit Test Case 1

Sl. No. of test case :	1
Name of test :	Powering of C-Mote
Item / Feature being tested :	C-Mote
Sample Input :	Power from a laptop
Expected output :	The charge indicator LEDs on the C-Mote glow
Actual output :	The charge indicator LEDs glow
Remarks :	Test succeeded

6.1.2.2 Unit Test Case 2

Flow Sensor

Table 6.2 shows the test result of the flow sensor which sends pulses as output when the water flows through it. A sample program is used to count the pulses received from the sensor.

Table 6.2 Unit Test Case 2

Sl. No. of test case :	2
Name of test :	Flow Sensor module test
Item / Feature being tested :	Working of flow sensor
Sample Input :	Sample program to count pulses
Expected output :	Count of pulses
Actual output :	Count of pulses
Remarks :	Test succeeded

6.1.2.3 Unit Test Case 3

Logic Level Converter

Logic Level Converter (LLC) is used to convert the 5V output from the flow sensor to 3.3V and transfer it to the C-Mote connected to it. Table 6.3 shows the result of testing the logic level converter.

Table 6.3 Unit Test Case 3

Sl. No. of test case :	3
Name of test :	Logic Level Converter test
Item / Feature being tested :	Voltage level at both ends of LLC
Sample Input :	LLC is bridged between flow sensor and C-Mote
Expected output :	5V at one end and 3.3V at the other end
Actual output :	5V at one end and 3.3V at the other end
Remarks :	Test succeeded

6.1.2.4 Unit Test Case 4

UbiSense Buzzer

The UbiSense Buzzer connected to the transmitter mote should be triggered when the mote receives any leakage detection from the flow sensor. Table 6.4 shows the result of testing the buzzer.

Table 6.4 Unit Test Case 4

Sl. No. of test case :	4
Name of test :	UbiSense Buzzer Test
Item / Feature being tested :	UbiSense Buzzer
Sample Input :	Signal from sensor
Expected output :	Triggering the buzzer and transmitting data
Actual output :	Buzzer triggered infinitely
Remarks :	Test failed

Correction: The buzzer as well as the interrupt service routine require an infinite loop. Hence the buzzer was triggered infinitely and data was not transmitted. The buzzer should be triggered for a specific time. Table 6.5 shows the updated test result.

Table 6.5 Unit Test Case 4 - corrected

Sl. No. of test case :	4
Name of test :	UbiSense Buzzer Test
Item / Feature being tested :	UbiSense Buzzer
Sample Input :	Signal from sensor
Expected output :	Triggering the buzzer along with data transmission
Actual output :	Buzzer triggered for specific time
Remarks :	Test succeeded.

6.1.2.5 Unit Test Case 5

Wireless Communication

This is the major unit of the system which is responsible for communicating data over the air from one mote to other. Table 6.6 shows the test result.

Table 6.6 Unit Test Case 5

Sl. No. of test case :	5
Name of test :	Wireless Communication test
Item / Feature being tested :	Communication between motes
Sample Input :	Sample data to transmit
Expected output :	LEDs should toggle at both motes
Actual output :	LEDs toggled
Remarks :	Test succeeded

6.1.2.6 Unit Test Case 6

User Notification

Once leakage alert is received by the coordinator, user should be notified through email and text message. Tables 6.7 and 6.8 shows test results for email and message alerts respectively.

Table 6.7 Unit Test Case 6.1

Sl. No. of test case :	6.1
Name of test :	Email test
Item / Feature being tested :	Notification through email
Sample Input :	Sample message
Expected output :	Email should be received
Actual output :	Email received
Remarks :	Test succeeded

Table 6.8 Unit Test Case 6.2

Sl. No. of test case :	6.2
Name of test :	Text Message test
Item / Feature being tested :	Notification through text message
Sample Input :	Sample message
Expected output :	Text Message should be received
Actual output :	Text Message received
Remarks :	Test succeeded

6.1.2.7 Unit Test Case 7

Database Upload

This test case is to test whether the data is updated to the database when the coordinator mote receives a leakage alert and the result is shown in Table 6.9.

Table 6.9 Unit Test Case 7

Sl. No. of test case :	7
Name of test :	Database upload
Item / Feature being tested :	Uploading data to database
Sample Input :	Writing sample data to serial port
Expected output :	Data should be uploaded to database
Actual output :	Data uploaded
Remarks :	Test succeeded

6.1.2.8 Unit Test Case 8

Website Dynamicity

This test case aims at testing the website whether the uploading of new data into the database updates the charts and tables and the result is shown in Table 6.10.

Table 6.10 Unit Test Case 8

Sl. No. of test case :	8
Name of test :	Website Dynamicity
Item / Feature being tested :	Website Statistics
Sample Input :	Data uploaded to database
Expected output :	Charts should include the new data upon refresh
Actual output :	Charts updated upon refresh
Remarks :	Test succeeded

6.2 Integration Testing

Integration testing is the next of stage or the expansion of unit testing. The units which are individually tested are combined into a single module and tested. Any error thus occurred could be an unnoticed bug in any of the units or is the result of the interaction between the units. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within the group of units being tested interact correctly.

6.2.1 Testing Strategy

The following points are symptomatic of the testing strategy for integration testing followed in the system.

- A draft of the plan is created initially in the execution phase. The major pieces of hardware and/or software that are to be integrated are known by this time from the high-level design work in the initiation phase. An early glance at the sequence of integration helps check amount of time that is put into the schedule for integration.
- The plan is reviewed with technical team members and others who support the integration testing.
- As the detailed designs are completed, the plans are refined accordingly.
- The interactions between the units that are integrated as defined in the initial stages are tested.
- The integrated component is retested with all the test cases defined for each of the participating units.

6.2.2 Test Cases

Tables 6.11 to 6.15 lists the Integration test Cases.

6.2.2.1 Integration Test Case 1

C-Mote, Flow Sensor and LLC

The Flow Sensor is connected to a 5V supply and C-Mote is connected to a 3.3V supply. The output of the flow sensor is connected to the high end of the LLC and the corresponding low end is connected to the pin P2.4 of the C-Mote. Table 6.11 shows the result of test performed on the integration of C-Mote, flow sensor and LLC.

Table 6.11 Integration Test Case 1

Sl. No. of test case :	1
Name of test :	Integration testing of C-Mote, Flow Sensor and LLC
Item / Feature being tested :	Whether the flow sensor signals are read by C-Mote
Sample Input :	Powering all the hardware and blowing the flow sensor
Expected output :	Count of sensor pulses should be printed
Actual output :	Count of sensor pulses printed
Remarks :	Test succeeded

6.2.2.2 Integration Test Case 2

C-Mote, Flow Sensor, LLC and UbiSense Buzzer

C-Mote is connected to the Flow Sensor thorough the LLC. The UbiSense buzzer is connected to the C-Mote. When flow sensor outputs pulses, the C-Mote counts them and triggers the buzzer. Table 6.12 shows the test result.

Table 6.12 Integration Test Case 2

Sl. No. of test case :	2
Name of test :	Integration testing of C-Mote, Flow Sensor, LLC and UbiSense Buzzer
Item / Feature being tested :	Working of the buzzer when the flow sensor sends a signal to the C-Mote
Sample Input :	Powering all the hardware and blowing the flow sensor
Expected output :	Count of the sensor pulses and buzzer should be triggered
Actual output :	Count of sensor pulses printed and buzzer triggered
Remarks :	Test succeeded

6.2.2.3 Integration Test Case 3

C-Mote, Flow Sensor, LLC, UbiSense Buzzer and Wireless Communication

C-Mote is powered with a 3.3V supply and Flow Sensor is powered with a 5V supply. They are connected to each other through the LLC. UbiSense buzzer is connected to the C-Mote. When the sensor sends pulses, C-Mote should trigger the buzzer and create a packet and transmit it to the coordinator mote wirelessly. The middle LED of the C-Mote is set to toggle after a successful data transmission. The coordinator mote, when receives the packet, toggles its middle LED too to provide a visual confirmation of the successful data transmission. Table 6.13 shows the result of the test case.

Table 6.13 Integration Test Case 3

Sl. No. of test case :	3
Name of test :	Integration testing of C-Mote, Flow Sensor, LLC, UbiSense Buzzer and Wireless Communication between the motes
Item / Feature being tested :	Wireless Communication between the motes along with triggering the buzzer when the flow sensor sends pulses
Sample Input :	Powering all the hardware and blowing the flow sensor
Expected output :	Value of count of sensor pulses should be transmitted to the coordinator mote and the LEDs should toggle
Actual output :	Value of count of sensor pulses received at the coordinator mote and LEDs on both the motes toggled
Remarks :	Test succeeded

6.2.2.4 Integration Test Case 4

User Notification, Database Upload and Website Dynamicity (Coordinator Mote)

When the coordinator mote receives some data, that data is written to the serial port of the laptop. The serial data is read at 4800 baud rate by a python script and the alert is sent to the users through a text message and an email. Also the timestamp and the location of the leakage are uploaded to the database. When the user accesses the website, the details of the new leakage are automatically included in the charts and tables. Since this is a continuously running process, if internet connection is not available the process should not be stopped. Result of the test case is shown in Table 6.14.

Table 6.14 Integration Test Case 4

Sl. No. of test case :	4
Name of test :	Integration test of user notification, database upload and website dynamicity
Item / Feature being tested :	Notification to the user through Email and Text message when the serial port reads some data. Also the uploading of data to database and website dynamicity
Sample Input :	Providing sample data at the serial port of the laptop
Expected output :	An email and a text message should be sent to the user, leakage details should be uploaded to database and the website should include this data. Process should continue if there is no internet
Actual output :	Email and text message is sent to the user, leakage details uploaded to database and website included this data. Error message printed and process continues if there is no internet
Remarks :	Test succeeded

6.2.2.5 Integration Test Case 5

C-Mote, Flow Sensor, LLC, Buzzer, Wireless Communication and Coordinator mote

C-Mote and the flow sensor powered with 3.3V and 5V respectively and connected together using LLC. Buzzer is connected to the C-Mote and when the flow sensor sends the pulses, data is sent to the coordinator mote from the C-Mote connected to it. The coordinator mote parses the packet to obtain the data and then writes that data to the serial port of the laptop. The serial data is read at 4800 baud rate by a python script and the alert is sent to the users through a text message and an email alert. Also the timestamp and the location of the leakage are uploaded to the database. When the user accesses the website, the details of the new leakage are automatically included in the charts and tables. Result of the test case is shown in Table 6.15.

Table 6.15 Integration Test Case 5

Sl. No. of test case :	5
Name of test :	Integration test of C-Mote, Flow Sensor, LLC, UbiSense Buzzer, Wireless Communication and Coordinator mote features
Item / Feature being tested :	Whether the transmission of flow sensor data is successful and user is notified about it along with uploading data to database
Sample Input :	Powering all the hardware and blowing the flow sensor
Expected output :	LEDs should toggle at both the motes confirming the successful transmission. Email and text message should be sent to the user and data should be uploaded to the database.
Actual output :	LEDs toggled at both the motes confirming the successful transmission. Email and text message were sent to the user and data was uploaded to the database.
Remarks :	Test succeeded

6.3 System Testing

System testing of a software or hardware is a black box testing technique performed on a complete, integrated system to measure the compliance of the system with the requirements specified. In System testing, the functionalities of the system are tested from an end-to-end perspective. All the integrated modules which have cleared integration testing are the input for the system

6.3.1 Testing Strategy

System testing must check whether the system or the application is performing as stated in the specifications and meets the user requirements. Hence the test cases and the process by which system testing is conducting must focus on these aspects. At this stage, all the components and modules should be integrated into a single module. Since all the components would have passed the integration testing before the beginning of system testing, a simple testing would be enough to ensure that no errors occur in the system. As in the integration testing, the above test steps should be performed again to verify if the system runs well and that no errors are generated because all the modules are combined into one system.

6.3.2 Test Cases

Table 6.16 list the system test case.

6.3.2.1 System Test Case 1

The aim of this test case is to test for the overall system of the water leakage detection. When there is leakage in the pipe, the user should receive an alert through email and text message along with that information being included in the website.

Table 6.16 System Test Case 1

Sl. No. of test case :	1
Name of test :	System testing
Item / Feature being tested :	Leakage detection system is tested for its complete working
Sample Input :	Powering all the hardware, connecting the pipes and creating a hole in the pipe causing leakage
Expected output :	The user should be notified through email and text message, buzzer should be triggered and website should be updated.
Actual output :	The user receives notifications through email and text message, buzzer is triggered and the website is updated.
Remarks :	Test succeeded

Chapter 7

RESULTS AND DISCUSSIONS OF 6LoWPAN BASED LEAK DETECTION

This chapter focuses on the results obtained and matched for correctness. Different sets of tests are run to verify if the system runs as it is expected to and gives the appropriate result. Then based on the result obtained, appropriate studies are done and the inference is made from the studies.

7.1 Results

This is a crucial phase where analysis is done, the results obtained are analysed for correctness and matched with the expected output for any anomaly.

7.1.1 The User Interface

The following are the results obtained for the user interface module of our system:-

- The UI is a website which shows statistics of the leakages occurred.
- The UI gives a good look and feel to the project.
- The statistics are displayed in the form of charts and tables.
- The website also displays details of the development of the project and the developers.

7.1.2 The Transmission and Receiver Modules

The results obtained by running the transmitter (tx) and receiver (rx) modules are as follows:

- The tx module is aptly able to read the count pulses and generate a trigger.
- The module is then able to create a packet and transmit the leakage data to the receiver module.
- The receiver module is able to receive the data packets sent by the transmitter module.
- It is then able to invoke a python script which reads data from the terminal and sends notification to the user via mail and text message.

7.2 Discussion

The proposed approach allows for users to be informed in case of any event of occurrence of water leakage in the pipes used in domestic household. The setup consists of a flow sensor attached between two junctions or on a continuous uninterrupted pipe length. The flow rate is measured and monitored continuously. When the pipe develops a crack, the water starts leaking through it at a specific rate of flow. This rate of flow is taken into account and generates a trigger indicating the occurrence of a leak. A packet is created and initialised with data by the transmitter mote which is then sent to the receiver mote. The receiver mote upon receiving this packet then invokes a script which makes use of the Twilio facility and mail service to notify the user via text message and email. The data is also pushed onto the server, where user can view all the statistics.

7.3 Inference

When the pipe develops a crack, the leakage of water through it is detected. The transmitter C-Mote then creates a packet and sends the data to the receiver C-Mote using the 802.15.4 header or 6LoWPAN standard. The receiver mote then invokes a python program which writes the data to the server, and sends a notification to the user via mail and also text message.

7.4 Snapshots

7.4.1 Sensor and Transmitter Setup

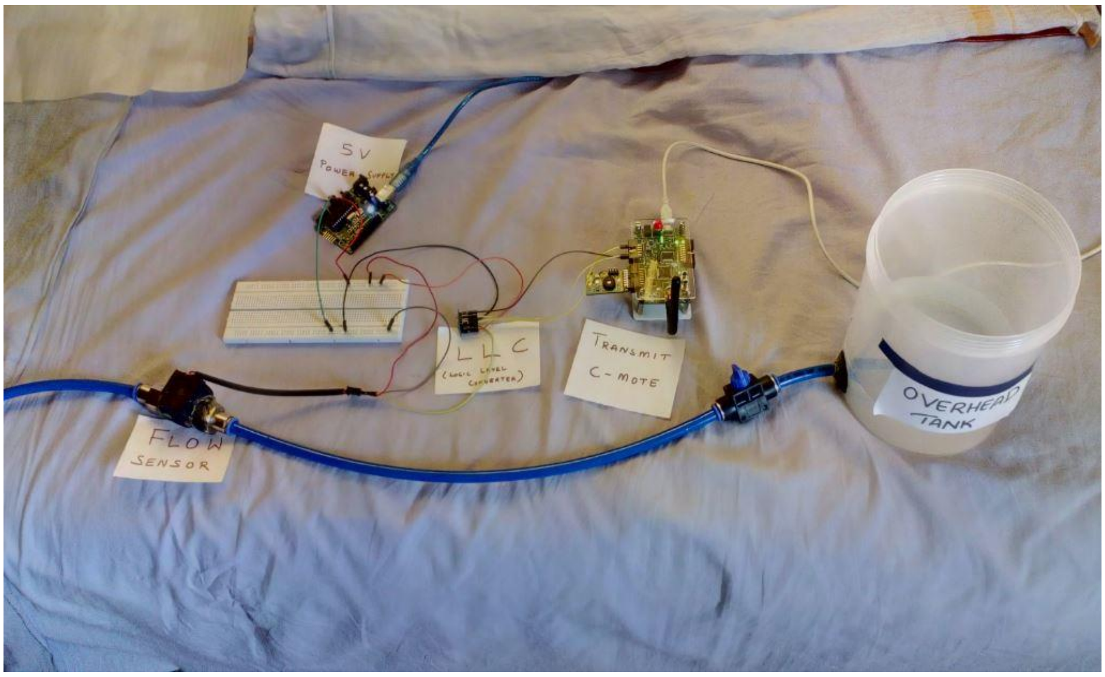


Figure 7.1 Sensor and Transmitter Setup

Figure 7.1 shows the sensor and transmitter setup for the project. The figure shows the overhead tank, valve, LLC, C-mote, Flow Sensor, pipes, breadboard and 5V source.

7.4.2 Receiver or Coordinator setup

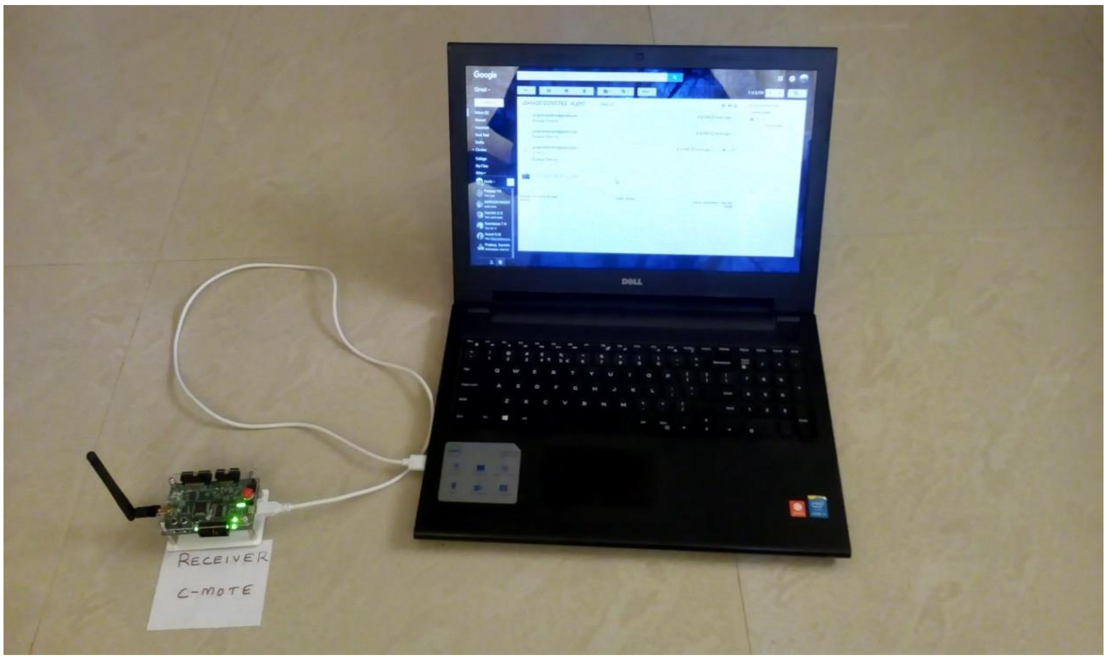
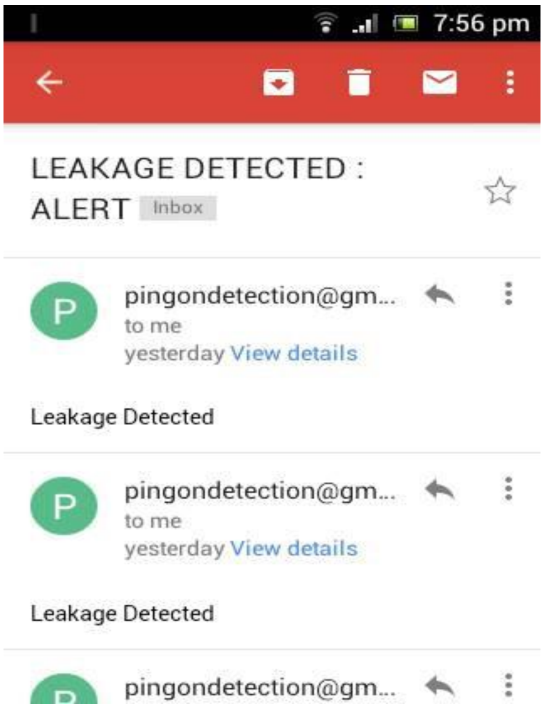


Figure 7.2 Receiver Setup

Figure 7.2 shows the receiver C-Mote setup. When this mote receives any data it writes that data to the serial port of the laptop.

7.4.3 Email notification


7:56 pm

LEAKAGE DETECTED : ALERT Inbox

pingondetection@gm...
to me
yesterday View details

Leakage Detected

pingondetection@gm...
to me
yesterday View details

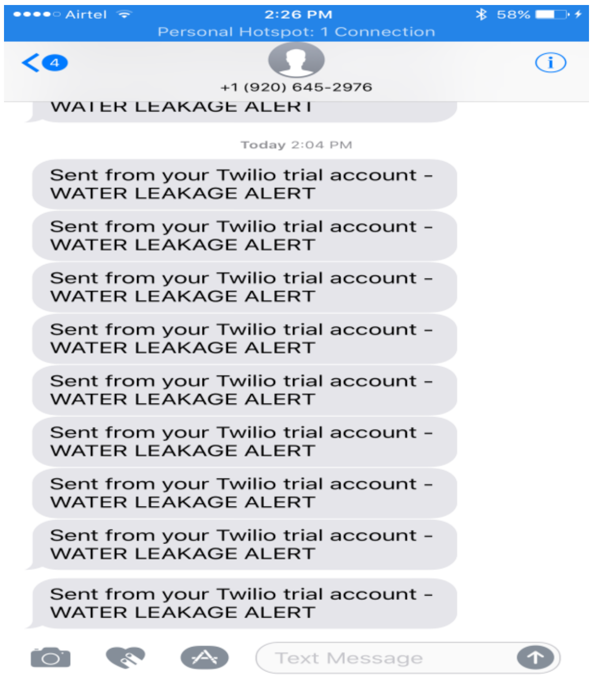
Leakage Detected

pingondetection@gm...

Figure 7.3 Email notification received by the user

Figure 7.3 shows a screenshot of a mail received by the user when the occurrence of a leak is detected.

7.4.4 SMS notification


●●●●○ Airtel 2:26 PM 58%
Personal Hotspot: 1 Connection
4
+1 (920) 645-2976

WATER LEAKAGE ALERT

Today 2:04 PM

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Sent from your Twilio trial account - WATER LEAKAGE ALERT

Text Message

Figure 7.4 SMS received by the user

Figure 7.4 shows a screenshot of the message notifications received by the user. These messages were sent to the user using the Twilio account.

7.4.5 Website homepage

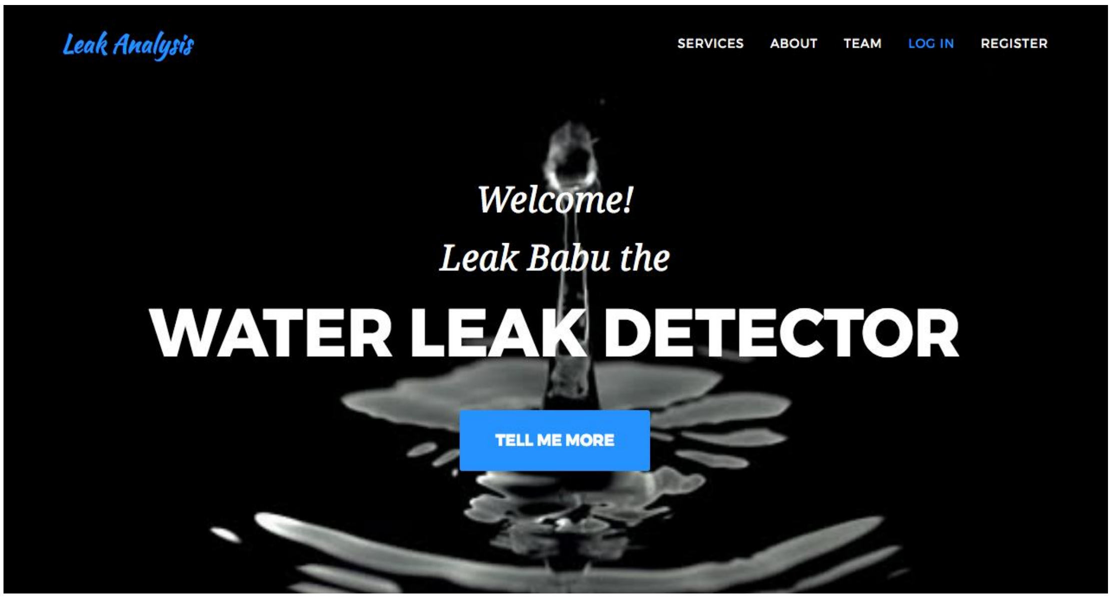


Figure 7.5 Homepage of the Website

Figure 7.5 shows the homepage of website Leakbabu, a GUI developed for the project wherein the user can view the leakage statistics.

7.4.6 Leakage Charts

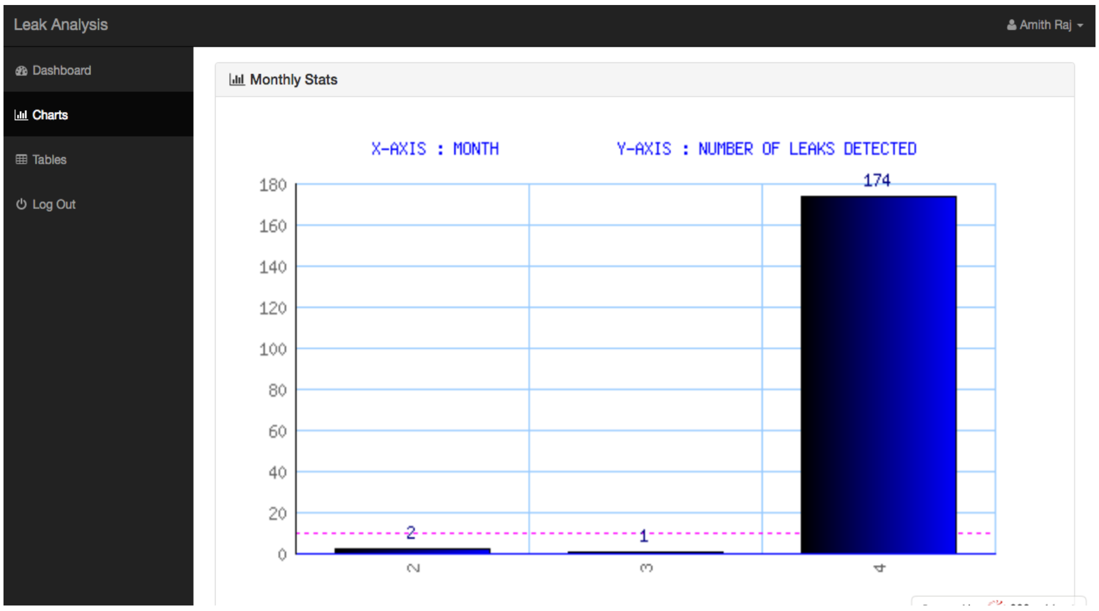


Figure 7.6 Leakage Statistics

Figure 7.6 shows the leakage statistics to the user in the form of charts. Here, the y-axis denotes the number of leakages whereas the x-axis denotes the month.

Chapter 8
CONCLUSION

8.1 Summary

This work presents the techniques that can be used to detect a water leak in a given piped setup. It provides methods to solve the existing issues of water leakages in pipes using low power, upcoming technology. Using flow sensors to detect the leakage, the data is obtained in one mote and transmitted to the coordinator mote wirelessly which is connected to internet. Hence only one mote should be connected to the internet while all other motes can communicate wirelessly without internet.

8.2 Limitations

The limitations of the product are not negative aspects to the work done. However, no product can fulfil all the needs of the user. With respect to this, the following might be the limitations of the setup.
Some important limitations as follows:

- This product works only for the range of the given flow sensor.
- The product has to be connected to the internet at all times.

8.3 Future Enhancements

- Scaling the application to encompass the entire household.
- Using programmable valve (solenoid valve) to shut the water supply when a leakage is detected.
- Having a rollback mechanism to shut down the system on all call from the user to the system.

REFERENCES

[1] Zach Shelby and Carsten, *"6LoWPAN: The Wireless Embedded Internet",* 1st edition, John Wiley & Sons, 2009.

[2] Jonas Olsson, "*6LoWPAN Demystified*", Texas Instruments, October 2014.

[3] IoT Group, "*C-Mote User Manual v1.1*", Centre for development of Advanced Computing (CDAC), Knowledge Park, Bengaluru, 2016.

[4] Anjana S, Sahana M N, Ankith S, K Natarajan, K R Shobha, "*An IoT based 6LoWPAN enabled Experiment for Water Management*", 2015 International Conference on Advanced Networks and Telecommuncations Systems, 2015.

[5] Zucheng Huang, Feng Yuan, *"Implementation of 6LoWPAN and its application in smart lighting"*, Institute of software application technology, Guangzhou & Chinese Academy of Sciences, Journal of Computer and Communications, 2015.

[6] Ardiansyah Musa Efendi, Seungkyo Oh and Deokjai Choi, *"6LoWPAN–based wireless home automation: from secure system development to building energy management"*, Smart Computing Review, vol. 3, no. 2, April 2013.

[7] Water Quality Facts. http://www.unwater.org/publications/publications- detail /en/ c/204326/

[8] R. Cardell-Oliver, "*Discovering water use activities for smart metering*," in Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on, April 2013, pp. 171-176.

[9] Lee. Young-Woo, Eun. Seongbae, Oh. Seung-Hyueb, "*Wireless digital water meter with low power consumption for automatic meter reading*", 2008 International Conference on Convergence and Hybrid Information Technology, ICHIT 2008, p 639-645, 2008.

[10] Baoding Zhang, Jialei Liu, "*A kind of design schema of wireless smart water meter reading system based on ZigBee technology*", in E-Product E- Service and E-Entertainment (ICEEE), 2010 International Conference on, Nov. 2010, pp 1-4

[11] LI Zhi-jun, SUI Xiao-hong, SHI Jian-ting and LIU Fu-gang, "*Design of Wireless Remote Meter Reading Based on Bluetooth and GPRS*", Applied Science and Technology, vol. 9, pp. 50-53, 2007.

[12] N. Sriskanthan, F. Tan, A. Karande, "*Bluetooth based home automation system*", *Microprocessors and Microsystems,* vol. 26, no.6, pp. 281-289, 2002.

[13] K. Gill, S.-H. Yang, F. Yao, X. Lu, "*A ZigBee-based home automation system*", IEEE Transactions on Consumer Electronics, vol.55 ,no.2, pp. 422–430, 2009.

[14] N. Kushalnagar, G. Montenegro, C. Schumacher, "*IPv6 over low-power wireless personal area networks (LoWPANs): overview, assumptions, problem statemen, and goals*" RFC 4919, 2007.

[15] M. Kovatsch, M. Weiss, D. Guinard, "Embedding internet technology for home automation," in *Proc. of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '10),* Sep. 2010.

[16] B.M. Dorge, T. Scheffler, "Using IPv6 and 6LoWPAN for home automation networks," in *Proc. of the 1st IEEE International Conference on Consumer Electronics (ICCE '11)*, pp. 44–47, Sep. 2011.

[17] B O'Flynn, Rafael Martínez-Català, S. Harte, C. O'Mathuna, John Cleary, C. Slater, F.Regan, D. Diamond, Heather Murphy, "*SmartCoast- A Wireless Sensor Network for Water Quality Monitoring*" 32nd IEEE Conference on Local Computer Networks, pp.815 - 816, 2007.

[18] Eduardo Garcia Breijo, Luis Gil Sanchez, Javier Ibañez Civera, Alvaro Tormos Ferrando,Gema Prats Boluda, "*Thick-Film Multisensor for Determining Water QualityParameters*" Proceedings of the IEEE, 2002.

[19] JayaLakshmi M., Gomathi V, "*An enhanced underground pipeline water leakage monitoring and detection system using Wireless Sensor Networks*", in ‚Soft-Computing and Networks Security (ICSNS), 2015 International Conference, February 2015, pp1-6 .

[20] Dominguez, F., Touhafi, A., Tiete, J. and Steenhaut, K, "*Coexistence with WiFi for a Home Automation ZigBee Product*", IEEE 19th Symposium on Communications and Vehicular Technology in the Benelux (SCVT), 2012, 1-6.

[21] X. Ma and W. Luo, "*The Analysis of 6LowPAN Technology*," in IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, vol. 1, no. 3. IEEE Com Soc, Dec. 2008, pp. 963-966.

SOURCE CODE

Transmitter:

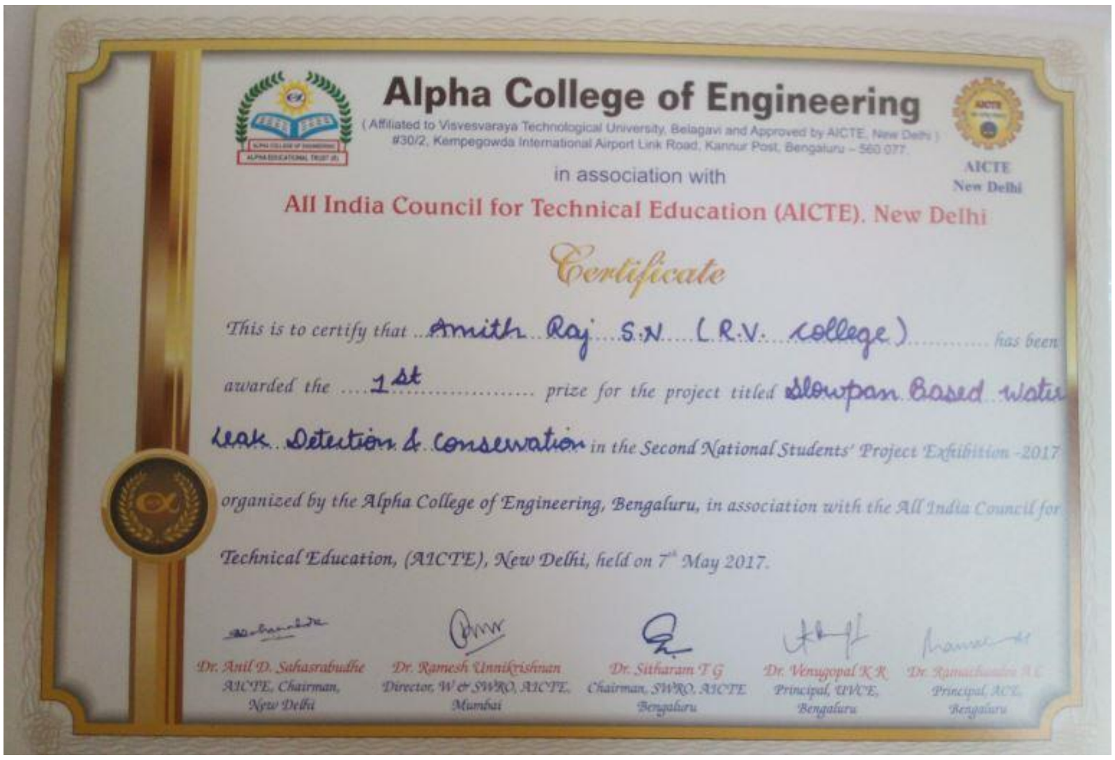
```
int main(void){

    WDTCTL = WDTPW + WDTHOLD;
    halBoardInit();
    halUartInit(HAL_UART_BAUDRATE_4800, 0);
    halLedSet(1);
    halLedSet(2);
    halLedSet(3);
    basicRfConfig.panId = PAN_ID;
    basicRfConfig.ackRequest = FALSE;
    basicRfConfig.channel = CHANNEL;
    basicRfConfig.myAddr = TX_ADDR;
    if(halRfInit()==FAILED) {
        HAL_ASSERT(FALSE);
    }
    halLedClear(3);

    halRfSetTxPower(TX_POWR);
     toggleMain();
    //appTransmitter();
    return 0;
}

static void appTransmitter(void){

     uint8 n,j;

    if(basicRfInit(&basicRfConfig)==FAILED) {
        HAL_ASSERT(FALSE);
     }
    basicRfReceiveOff();
    txPacket.seqNumber = count;
    for(j=0; j<25; j++)
     txPacket.padding[j] = 'a';
    //while(1)
     {

      //  wait(1);

        halLedToggle(2);

           //j = sensorInit();
        //sensorData(j, txPacket.padding);
           basicRfSendPacket(RX_ADDR, (uint8*)&txPacket,
sizeof(txPacket));
        txPacket.seqNumber++;
    }
}
```

```
void toggleMain(void) {

      WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer

      //P5DIR = BIT4;
      //P5OUT = BIT4;

      P2IE |= 0x10; // P2.4 interrupt enabled
      P2IES |= 0x10; // P2.4 Hi/lo edge
      P2IFG &= ~0x10; // P2.4 IFG cleared
      _BIS_SR(LPM0_bits + GIE);
      //printf("ROFL\t");

      while(1) {
            P2IE |= 0x10; // P2.4 interrupt enabled
            P2IES |= 0x10; // P2.4 Hi/lo edge
            P2IFG &= ~0x10; // P2.4 IFG cleared

            _BIS_SR(LPM0_bits); // Enter LPM4 w/interrupt
            wait(2);
            printf("Round Over ");
            //_BIC_SR_IRQ(LPM0_bits);
            //LPM0_EXIT;
            //if(count==0)   flag=0;
            //count = 0;

     }
}


// Port 2 interrupt service routine
#pragma vector=PORT2_VECTOR
__interrupt void Port_2(void) {

      count++; // Incrementing the counter when ever an
interrupt comes

      if(count%2 == 0)
      {
            //flag = 1;
            halLedClear(1);
            printf("%d\n",count);
            //if(!flag)
                  appTransmitter();
            //LPM0_EXIT;
            flag = 1;
            //_BIC_SR_ISQ(LPM0_bits);
      }
      P2IFG &= ~0x10; // P1.4 IFG cleared
}
```

Receiver:

```
static basicRfCfg_t basicRfConfig;
static perTestPacket_t rxPacket;
extern uint8 header[6];
uint8_t *payload;

int putchar(int c) {

    while(!(UC1IFG & UCA1TXIFG));   // WAIT TILL THE BUFFER IS
EMPTY
     UCA1TXBUF = c;
    return c;
}

int main(void){
    WDTCTL = WDTPW + WDTHOLD;
    halBoardInit();
    halUartInit(HAL_UART_BAUDRATE_4800, 0);
    halLedSet(1);
    halLedSet(2);
    halLedSet(3);
    basicRfConfig.panId = PAN_ID;
    basicRfConfig.ackRequest = FALSE;
    basicRfConfig.channel = CHANNEL;
    basicRfConfig.myAddr = RX_ADDR;
    if(halRfInit()==FAILED) {
        HAL_ASSERT(FALSE);
    }
    halLedClear(1);
    halRfSetTxPower(TX_POWR);
    appReceiver();
    return 0;
}

static void wait(uint8 sec){

     uint8 n;
     TACCTL0 &= ~(1<<0);
     for(n=0;n < sec;n++){
           CCR0 = 32768;
           TACTL = TASSEL_1 + MC_1;   // SMCLK, upmode
           while(!(TACCTL0 & (1<<0)));
           TACCTL0 &= ~(1<<0);
      }
}


static void appReceiver(void){
     uint8_t i;
    uint32_t data;
    int16 rssi;
    // Initialize BasicRF
    basicRfConfig.myAddr = RX_ADDR;
    if(basicRfInit(&basicRfConfig)==FAILED) {
        HAL_ASSERT(FALSE);
    }
     basicRfReceiveOn();

// Main loop
     while (TRUE) {
        while(!basicRfPacketIsReady());
        pc_based_gateway_rf_isr();
           halLedClear(2);
           if(basicRfReceive((uint8*)&rxPacket,
MAX_PAYLOAD_LENGTH, &rssi)>0) {
            halLedToggle(3);
            payload = (uint8_t *)(&rxPacket);
           long count = 0, lphr=0;
           count = rxPacket.seqNumber;
           int i=0;
           for(i=1; i<60; i++)
                count += count;
           printf("%d - ",rxPacket.seqNumber);
           printf("%ld - ",count);
           lphr = (count*60) / 7.5;
           printf("%ld", lphr);
           printf(" L/hr\n");
            /*for(i=0;i<2;i++){
                while(!(UC1IFG & UCA1TXIFG));
                UCA1TXBUF = header[i];
            }
            for(i=4;i<6;i++){
                while(!(UC1IFG & UCA1TXIFG));
                UCA1TXBUF = header[i];
            }
            for(i=0;i<29; i++){
                while(!(UC1IFG & UCA1TXIFG));
                UCA1TXBUF = payload[i];
            }*/
           /*
}
```

WINNING CERTIFICATES



Alpha College of Engineering
(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi)
#30/2, Kempegowda International Airport Link Road, Kannur Post, Bengaluru – 560 077.

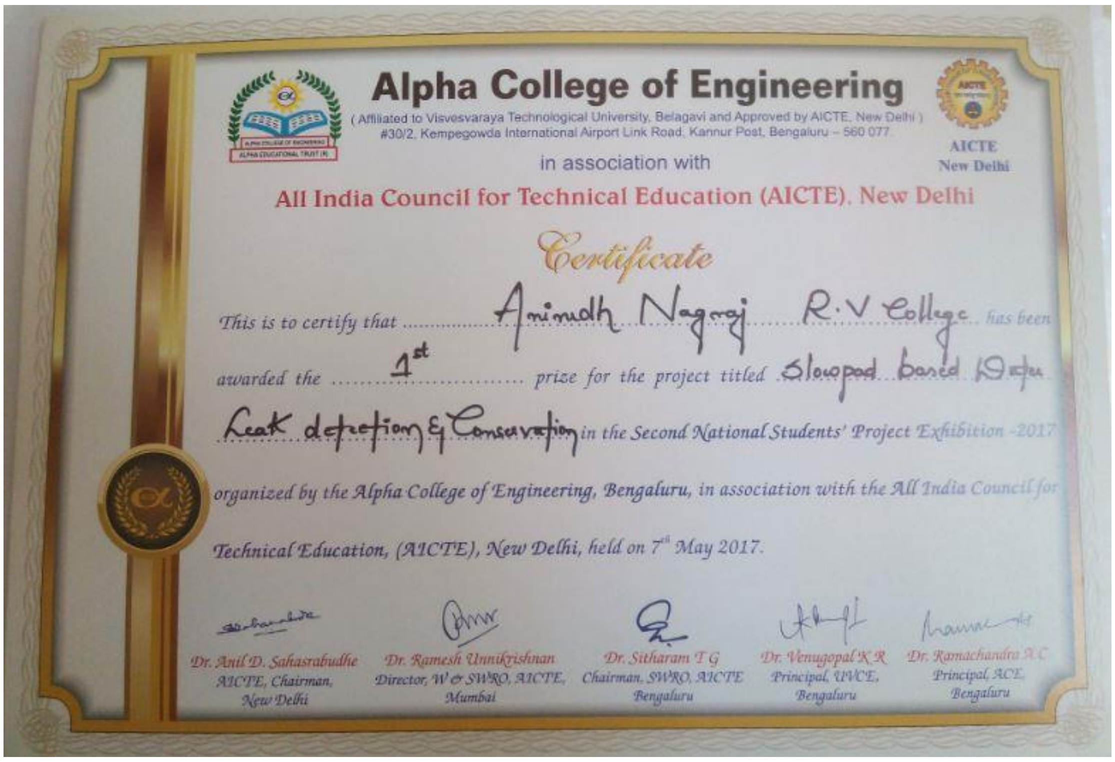
AICTE
New Delhi

in association with
All India Council for Technical Education (AICTE), New Delhi

Certificate

This is to certify that Amith Raj S.N (R.V. college) *has been awarded the* 1st *prize for the project titled* 6lowpan Based water leak Detection & conservation *in the Second National Students' Project Exhibition -2017 organized by the Alpha College of Engineering, Bengaluru, in association with the All India Council for Technical Education, (AICTE), New Delhi, held on 7th May 2017.*

Dr. Anil D. Sahasrabudhe	Dr. Ramesh Unnikrishnan	Dr. Sitharam T G	Dr. Venugopal K R	Dr. Ramachandra A C
AICTE, Chairman, New Delhi	Director, W & SWRO, AICTE, Mumbai	Chairman, SWRO, AICTE Bengaluru	Principal, UVCE, Bengaluru	Principal, ACE, Bengaluru



Alpha College of Engineering
(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi)
#30/2, Kempegowda International Airport Link Road, Kannur Post, Bengaluru – 560 077.

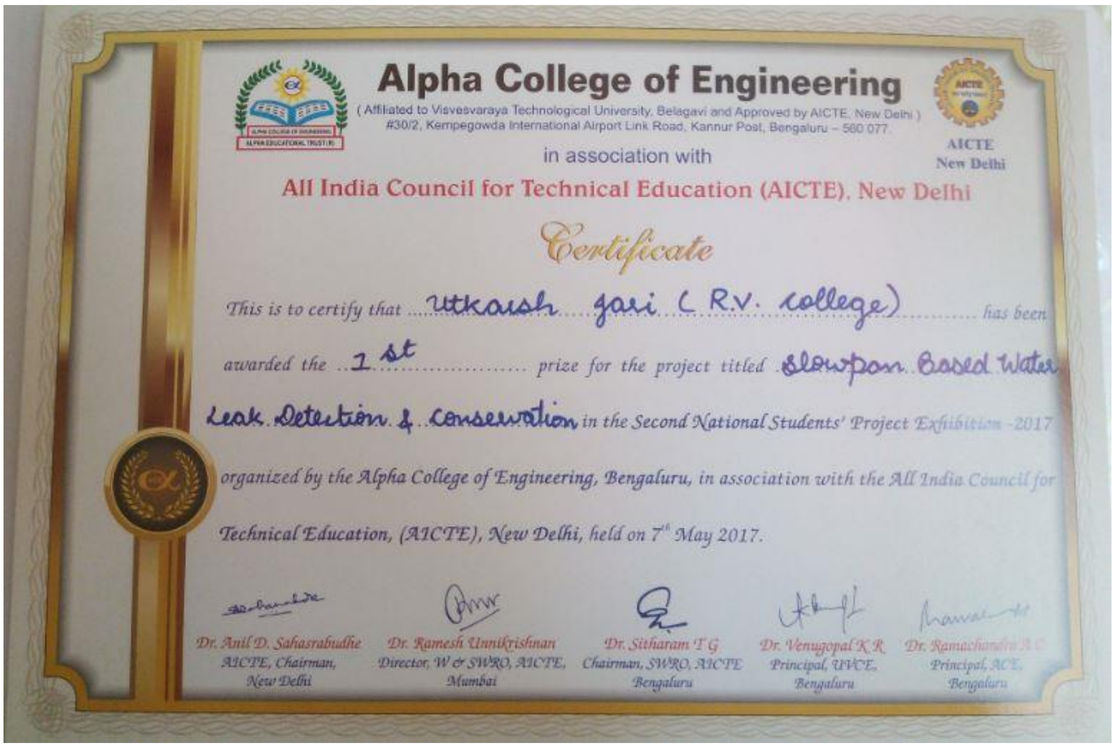
AICTE
New Delhi

in association with
All India Council for Technical Education (AICTE), New Delhi

Certificate

This is to certify that Anirudh Nagaraj R.V College *has been awarded the* 1st *prize for the project titled* 6lowpan based Water Leak detection & Conservation *in the Second National Students' Project Exhibition -2017 organized by the Alpha College of Engineering, Bengaluru, in association with the All India Council for Technical Education, (AICTE), New Delhi, held on 7th May 2017.*

Dr. Anil D. Sahasrabudhe	Dr. Ramesh Unnikrishnan	Dr. Sitharam T G	Dr. Venugopal K R	Dr. Ramachandra A C
AICTE, Chairman, New Delhi	Director, W & SWRO, AICTE, Mumbai	Chairman, SWRO, AICTE Bengaluru	Principal, UVCE, Bengaluru	Principal, ACE, Bengaluru



Alpha College of Engineering

(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi)
#30/2, Kempegowda International Airport Link Road, Kannur Post, Bengaluru – 560 077

AICTE
New Delhi

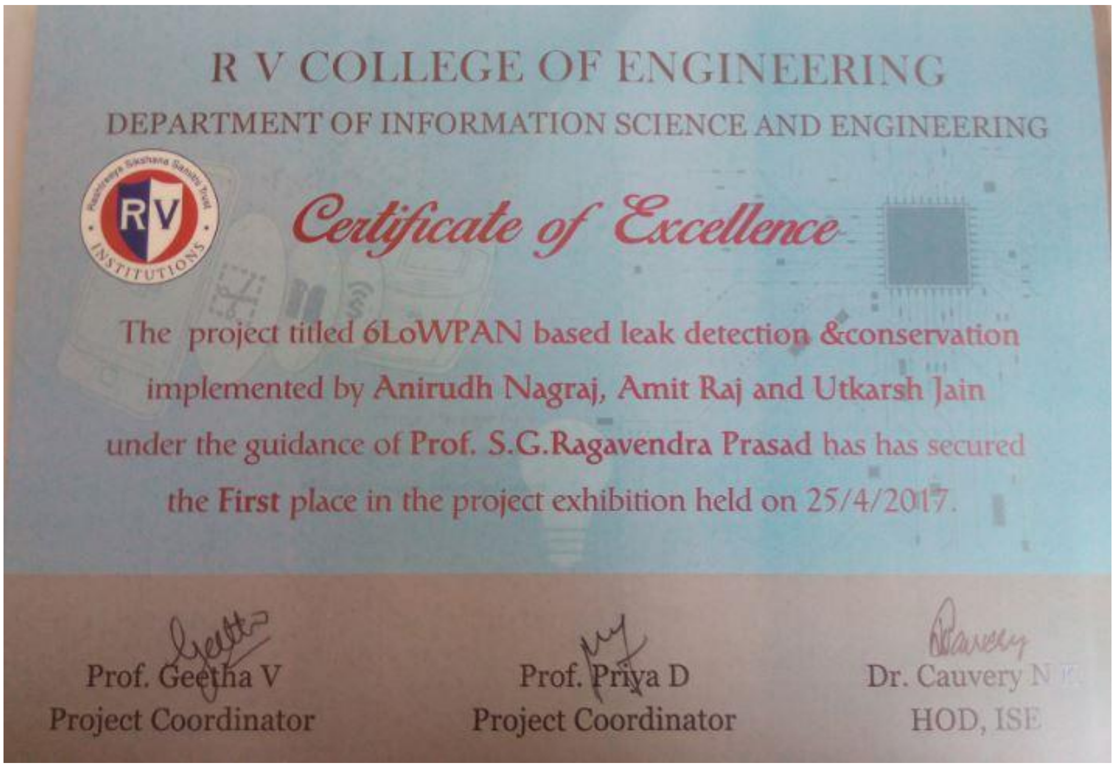
in association with

All India Council for Technical Education (AICTE), New Delhi

Certificate

This is to certify that utkarsh jain (R.V. college) *has been awarded the* 1 st *prize for the project titled* 6lowpan Based Water Leak Detection & Conservation *in the Second National Students' Project Exhibition -2017 organized by the Alpha College of Engineering, Bengaluru, in association with the All India Council for Technical Education, (AICTE), New Delhi, held on 7th May 2017.*

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Dr. Anil D. Sahasrabudhe	Dr. Ramesh Unnikrishnan	Dr. Sitharam T G	Dr. Venugopal K R	Dr. Ramachandra A C
AICTE, Chairman, New Delhi	Director, W & SWRO, AICTE, Mumbai	Chairman, SWRO, AICTE Bengaluru	Principal, UVCE, Bengaluru	Principal, ACE, Bengaluru



R V COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

Certificate of Excellence

The project titled 6LoWPAN based leak detection &conservation implemented by Anirudh Nagraj, Amit Raj and Utkarsh Jain under the guidance of Prof. S.G.Ragavendra Prasad has has secured the First place in the project exhibition held on 25/4/2017.

[illegible]	[illegible]	[illegible]
Prof. Geetha V	Prof. Priya D	Dr. Cauvery N K
Project Coordinator	Project Coordinator	HOD, ISE