

# **Drainage Monitoring System Using IoT**

*Project report submitted*

*In partial fulfilment of the requirement for the degree of*

**BACHELOR OF ENGINEERING**

**in**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**By**

**Sriram Venkat Sai Bharath**

**191312146**

**DEPARTMENT OF ELECTRONICS AND  
COMMUNICATION ENGINEERING**

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA UNIVERSITY**

**MAY 2017**



## **BONAFIDE CERTIFICATE**

This project report entitled “**Drainage Monitoring System Using IoT**” by “**Sriram Venkat Sai Bharath** ” is approved for the degree of B.E (Electronics and Communication Engineering)

### **PROJECT COORDINATOR**

**Mrs. D. Hari priya M.E,( Ph.D. )**

**Associate professor**

**Department of ECE**

**Saveetha School of Engineering**

**Saveetha University**

**Chennai – 602 105**

### **HOD**

**Dr. P. C. Kishore Raja M.E, Ph.D**

**Professor and Head**

**Department of ECE**

**Saveetha School of Engineering**

**Saveetha University**

**Chennai – 602 105**

**UNIVERSITY EXAM DATE:**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **CERTIFICATION BY GUIDE**

The project entitled “**Drainage Monitoring System Using IoT**” submitted for the B.E degree is an original work of **Sriram Venkat Sai Bharath** under the guidance of **MS. B.Bhuvaneswari**.

**Signature of the guide**

**MS. B.Bhuvaneswari M.tech.,(Ph.D)**

**Assistant professor**

**Department of ECE**

**Saveetha School of Engineering**

**Saveetha University**

**Chennai – 602 105**

## **DECLARATION**

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

**Signature of the student:**

**Sriram Venkat Sai Bharath**

**191312146**

**Date:**

## ACKNOWLEDGEMENT

I am thankful to my Almighty for His blessings and also for giving me good knowledge and strength in enabling me to finish our project. At the outset, I wish to express my gratitude to our honourable Chancellor, **Dr.N.M. Veeraiyan** for providing me with the facilities for the completion of my project. I am obliged to our Vice Chancellor, **Prof.Dr. L. Jawahar Nesan** for his continual support.

With profound gratitude and respect, I express my sincere thanks to our director, **Ms. Ramya Deepak**, for her continual support.

I thank our Principal, **Dr. D. Dhanasekaran** for his encouragement in carrying out our project work.

It is my duty to acknowledge with gratitude for the generous help that I have received from our Head of the Department, **Dr. P.C. Kishore Raja**, who has been furnishing a wealth of information regarding our project and has been instrumental in the successful completion of our project.

I mention my special thanks to our Project Coordinator **Mrs. D. Hari Priya** for her guidance and support.

I am indebted to our project guide **B.Bhuvaneswari** for her motivation, guidance and valuable suggestions to carry out project and her consistent encouragement throughout the course of my project work.

I take the opportunity to acknowledge the efforts with gratitude of our staff members in Electronics and Communication Department for their sincere co-operation and generous help. Finally I thank my parents and friends for their moral support and care extended during difficult times.

## TABLE OF CONTENTS

BONAFIDE CERTIFICATE.....	2
CERTIFICATION BY GUIDE .....	3
DECLARATION.....	4
ACKNOWLEDGEMENT .....	5
ORGANIZATION OF REPORT.....	8
LIST OF FIGURES .....	9
ABSTRACT .....	10
1.INTRODUCTION .....	12
1.1 IOT .....	13
2.LITERATURE SURVEY.....	15
2.1 APPLICATION OF INTERNET OF THINGS IN URBAN WATERLOGGING PREVENTION MANAGEMENT SYSTEM.....	18
2.2 SMART DRAINAGE SYSTEM.....	19
3. PROJECT OVERVIEW .....	22
3.1 GENERAL.....	22
4. IMPLEMENTATION .....	23
4.1 REQUIREMENT ANALYSIS .....	23
4.2 SOFTWARE REQUIREMENTS .....	23
4.3 HARDWARE REQUIREMENTS .....	23
5. BLOCK DIAGRAM.....	26
5.1 BLOCK DIAGRAM DESCRIPTION.....	27
5.2 ULTRASONIC SENSOR .....	27
5.3 INTEL GALILEO .....	29
5.4 PROGRAMMING .....	32
5.5 BLOCK DIAGRAM.....	35
5.6 DATA BASE MANAGEMENT SYSTEM .....	35
5.7 AURDINO .....	37
5.8 SOFTWARE DEVELOPMENT .....	37

6. RESULT AND ANALYSIS .....	41
6.1. OUTPUT .....	41
7. CONCLUSION .....	44
8. REFERENCE .....	46

## **ORGANIZATION OF REPORT**

- Chapter I contains the introduction of our project. It also lists out the advantages of this device.
- Chapter II consists of Literature survey for the project.
- Chapter III contains the general overview of the project with module explanations.
- Chapter IV explains the requirements of the project.
- Chapter V has the circuit description and the components used.
- Chapter VI contains the results analysis and the comparison of the project.
- Chapter VII contains the conclusion and future work.



## LIST OF FIGURES

FIG.NO	CONTENT	PAGE.NO
1	General block diagram	26
2	Description of Ultrasonic Sensor	27
3	Measurement of Distance using Ultrasonic Sensors	28
4	Intel Galelio Gen2 Board	31
5	Block Diagram For Generating Alerts	35
6	Output	41

**ABSTRACT:**

The Internet of Things (IOT) consists of real life objects, communication devices attached to sensor networks in order to provide communication and automated actions between real world and information world. IOT came into existence because, without human interaction computers were able to access data from objects and devices, but it was aimed at, to overcome the limiting factors of human entered data, and to achieve cost, accuracy and generality factors. Sensor Network is a key enabler for IOT paradigm. This project represents the implementation and design function of an Underground Drainage Monitoring System (UDMS) for IOT applications. The vital considerations of this design are low cost, low maintenance, fast deployment, and high number of sensors, long life-time and high quality of service. The proposed model provides a system of monitoring the water level and the location of the blockage.

# **CHAPTER I**

## **INTRODUCTION**

## INTRODUCTION

A good and efficient drainage system is badly required for the developing countries like India. In creation of many smart cities the architecture of drainage system plays an important role. To maintain a good and proper drainage system it takes more human resource. Even though investing cores of rupees in drainage department when it rains the scenario will be the same.

Most of the cities adopted the underground drainage system and it is the duty of managing station (Municipal Corporation) to maintain cleanliness of the cities. If the drainage maintenance is not proper pure water gets contaminated with drainage water and possibility of spreading various diseases. if there are drainage blocks during rainy season, routine life of people will get affected. Drainage blocks are closely related with traffic irregularities, surroundings get polluted, etc. Suppose if there is a facility associated with the Municipal Corporation (managing station) through which officials will get an alert as soon as a block arises regarding the area which has the block and the exact place where it is blocked and it also informs if the manhole lid is open.

To overcome all these problems we need a remote monitoring the states of drainage inside drainage channels.

Smart drainage system can contribute to sustainable development and improve the places and spaces where we live, work and play by balancing the different opportunities and challenges that influence the urban design and development of communities.

The present existing drainage system has to integrate with technology to wipe out the problems we face. The smart drainage system has

1. A Predictive drainage clogging system: The intelligence of sensors and predictive system identifies the drain clogged spot and gives us the details for further actions to take.
2. Drainage clogging alert system: If there is any clogging in any area sensors will gives us the necessary details about the location.
3. Completely connected: The sensors are communicated through communication modules to share information.

Using our smart drainage system we can easily monitor, modify and rectify the problems in real time. No drainage system is effective without human interaction.

## **IOT**

The Internet of Things (IOT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A thing, in the Internet of Things, can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low -- or any other natural or man-made object that can be assigned an IP address and provided with the ability to transfer data over a network.

IoT has evolved from the convergence of wireless technologies, micro-electromechanical systems (MEMS), micro services and the internet. The convergence has helped tear down the silo walls between operational technology(OT) and information technology (IT), allowing unstructured machine-generated data to be analyzed for insights that will drive improvements.

The Internet of Things (**IOT**) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data.<sup>[1][2][3]</sup> In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies"<sup>[3]</sup> and for these purposes a "thing" is "an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks".<sup>[4]</sup> The IoT allows objects to be sensed or controlled remotely across existing network infrastructure,<sup>[5]</sup> creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power

plants, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of about 30 billion objects by 2020.

There are many technologies that enable IOT. Crucial to the field is the network used to communicate between devices of an IOT installation, a role that several wireless or wired technologies may fulfill:

### **Short-range wireless**

- Bluetooth low energy (BLE) – Specification providing a low power variant to classic Bluetooth with a comparable communication range.
- Light-Fidelity (Li-Fi) – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.
- Near-field communication (NFC) – Communication protocols enabling two electronic devices to communicate within a 4 cm range.
- QR codes and barcodes – Machine-readable optical tags that store information about the item to which they are attached.
- Radio-frequency identification (RFID) – Technology using electromagnetic fields to read data stored in tags embedded in other items.
- Thread – Network protocol based on the IEEE 802.15.4 standard, similar to ZigBee, providing IPv6 addressing.
- Transport Layer Security (network protocol)/TLS – Network security protocol.
- Wi-Fi – Widely used technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point.
- Wi-Fi Direct – Variant of the Wi-Fi standard for peer-to-peer communication, eliminating the need for an access point.
- Z-Wave – Communication protocol providing short-range, low-latency data transfer at rates and power consumption lower than Wi-Fi. Used primarily for home automation.
- ZigBee – Communication protocols for personal area networking based on the IEEE 802.15.4 standard, providing low power consumption, low data rate, low cost, and high throughput.

### **Medium-range wireless**

- HaLow – Variant of the Wi-Fi standard providing extended range for low-power communication at a lower data rate.
- LTE-Advanced – High-speed communication specification for mobile networks. Provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.

### **Long-range wireless**

- Low-power wide-area networking (LPWAN) – Wireless networks designed to allow long-range communication at a low data rate, reducing power and cost for transmission.
- Very small aperture terminal (VSAT) – Satellite communication technology using small dish antennas for narrowband and broadband data.

### **Wired**

- Ethernet – General purpose networking standard using twisted pair and fibre links in conjunction with hubs or switches.
- Multimedia over Coax Alliance (MoCA) – Specification enabling whole-home distribution of high definition video and content over existing coaxial cabling.
- Power-line communication (PLC) – Communication technology using electrical wiring to carry power and data. Specifications such as Home Plug utilize PLC for networking IOT devices.

## **HISTORY**

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine, commodity sensors, and embedded systems. This means that the traditional fields of embedded systems, wireless sensor networks, systems, automation (including home and building automation), and others all contribute to enabling the Internet of things (IOT).

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Mark Weiser's seminal 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IoT. In 1994 Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances

to entire factories". Between 1993 and 1996 several companies proposed solutions like Microsoft's at Work or Novell's NEST. However, only in 1999 did the field start gathering momentum. Bill Joy envisioned Device to Device (D2D) communication as part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

The concept of the Internet of things became popular in 1999, through the Auto-ID Centre at MIT and related market-analysis publications. Radio-frequency identification (RFID) was seen by Kevin Ashton (one of the founders of the original Auto-ID Centre) as a prerequisite for the Internet of things at that point. Ashton prefers the phrase "Internet for Things. If all objects and people in daily life were equipped with identifiers, computers could manage and inventory them. Besides using RFID, the tagging of things may be achieved through such technologies as near field communication, barcodes, QR codes and digital watermarking.

In its original interpretation, one of the first consequences of implementing the Internet of things by equipping all objects in the world with minuscule identifying devices or machine-readable identifiers would be to transform daily life. For instance, instant and ceaseless inventory control would become ubiquitous. A person's ability to interact with objects could be altered remotely based on immediate or present needs, in accordance with existing end-user agreements. For example, such technology could grant motion-picture publishers much more control over end-user private devices by remotely enforcing copyright restrictions and digital rights management, so the ability of a customer who bought a Blu-ray disc to watch the movie could become dependent on the copyright holder's decision, similar to Circuit City's failed DIVX.



# **CHAPTER II**

## **LITERATURE REVIEW**

## **2.1 Application of Internet of Things in Urban Waterlogging Prevention Management System**

This paper aims to realize the extensive application of Internet of things technology in urban waterlogging prevention management system, and has analysed the security requirement and security architecture of Internet of things technology, and discussed the demand of urban waterlogging prevention management system in combination with the key technology of Internet of things technology, to do the overall design and functional design well during designing of urban waterlogging prevention management system. The application result shows that the flood control and drainage function of Chongqing is gradually improved with smooth drainage facilities; the inspection and maintenance management is gradually standardized; operation monitoring and early warning management is fully strengthened.

Internet of Things technology has a strong nature of secrecy, paying attention to protect information's security, avoid bringing direct damage to the institute, and promote the integrity analysis of information from the Internet of Things during the process of information security expression, at the same time, guarantee the securing application of Internet with security. The basic organizational process of Internet of Things is mainly to collect information, thus to provide effective guarantee for information analysis. Design the model using sensor technology based on information collection, to achieve the safeguard application of information security, and also conduct the infrastructure management analysis. As for information transmission and security processing links, achieve the effective scale management of data network layer in combination with the effective safeguard function of data, and improve the security of firewall.

### **System Function Design:**

- 1) Drainage facilities management system.
- 2) Inspection and maintenance management system.
- 3) Operation monitoring early warning system.
- 4) Emergency command and dispatch system.

## **2.2 SMART DRAINAGE SYSTEM**

The drainage system is the action of draining waste water and sticky liquid components towards the rivers using particular patterns, drainage channels and streams. Drainage system basically refers to all the piping within the private and public premises which conveys sewage, rainwater and other liquid waste to a point of disposal.

The present existing drainage system has to integrate with technology to wipe out the problems us facing. The smart drainage system has

1. Predictive drainage clogging system: The intelligence of sensors and predictive system identifies the drain clogged spot and gives us the details for further actions to take.
2. Drainage clogging alert system: If there is any clogging in any area sensors will gives us the necessary details about the location.
3. Completely connected: The sensors are communicated through communication modules to share information.

The drainage channels are covered with manholes to operate and to clear the clogging present inside the channel. By placing the sensors inside of the manhole will detects and transfers the appropriate information about the water, sticky contents, to detect elevated flow levels of drainage and clogging. Using the communication modules it will communicates with the sensors places at nearby manholes.

A wireless sensor network consists of hundreds or thousands of sensors having the capability to communicate among them or to send the data. Sensors will just monitor the water levels. Based upon the values given by the sensors drainage water levels and location ID will send to the Gateway and that sends to cloud (server) or concerned authority.

## **Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications**

Wireless sensor networks (WSN) are well suited for long term environmental data acquisition for IoT representation. This paper presents the functional design and implementation of a complete WSN platform that can be used for a range of long-term environmental monitoring IoT applications. The application requirements for low cost, high number of sensors, fast deployment, long lifetime, low maintenance, and high quality of service are considered in the specification and design of the platform and of all its components. Low-effort platform reuse is also considered starting from the specifications and at all design levels for a wide array of related monitoring applications.

WSN environmental monitoring includes both indoor and outdoor applications. The later can fall in the city deployment category (e.g., for traffic, lighting or pollution monitoring) or the open nature category (e.g., chemical hazard, earthquake and flooding detection, volcano and habitat monitoring, weather forecasting, precision agriculture). This application can be part of all three WSN categories: event-driven (as we have seen), time-driven (e.g., if the sensor nodes periodically send the air temperature) and query driven (e.g., if the current temperature can be requested by the operator).

This means that the infrastructure that supports the operation of this application can be reused for a wide class of similar long-term environmental monitoring applications like:

- water level for lakes, streams, sewages;
- • soil humidity and other characteristics;
- inclination for static structures (e.g., bridges, dams);
- position changes for, e.g., landslides;
- lighting conditions either as part of a combined sensing or standalone, e.g., to detect intrusions in dark places;

# **CHAPTER III**

## **PROJECT OVERVIEW**

## **PROJECT OVERVIEW**

### **3.1 GENERAL**

The drainage monitoring system using internet of things is proposed and we use the both software and hardware.

The Theme of this project is to design a Drainage monitoring System to reduce the effect caused by water logging during floods.

To collect data from sensors and transmits the collected data to the processor, analyse the drainage monitoring system through cloud and sends an alert to the corresponding Municipal authorities.

# **CHAPTER IV**

## **IMPLEMENTATION**

This chapter describes about the requirement analysis in accordance with the resources used. It also describes the implementation of the project with the tool used.

#### **4.1 REQUIREMENT ANALYSIS**

It determines the requirements of a new system and analyse on product and resource requirement, which is required for the successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements define in brief about the software and hardware that are needed to achieve the required functionality.

#### **4.2 SOFTWARE REQUIREMENTS**

- Arduino

#### **4.3HARDWARE REQUIREMENTS**

- Galelio gen2
- Wifi module
- GSM shield

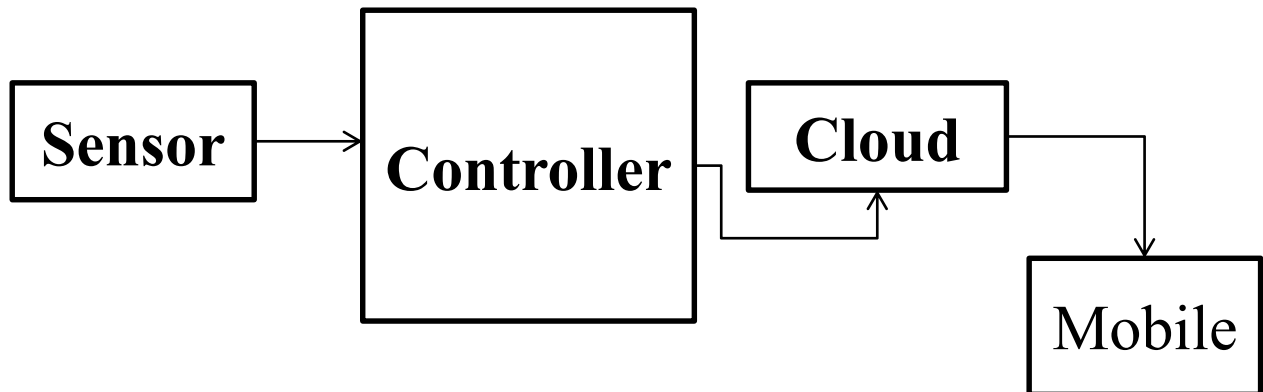


# **CHAPTER V**

## **BLOCK**

## **DIAGRAM**

## 5.1 BLOCK DIAGRAM DESCRIPTION



**BLOCK DIAGRAM**

### STEPS

- Required changes in the system –
  1. Detect the location
  2. Prior information of the blockage
- Design of SMART DRAINAGE SYSTEM.
- System governing the flow of sewage from pipes.
- To detect variations in the drainage flow.
- Collection of data.
- Obtain prior alerts regarding blocks and locate them using IOT

The above block diagram is explained with the different modules.

## MODULE 1:

### BLOCK DIAGRAM:

#### Sensor Module

#### Transmitter Module

#### Processing Module

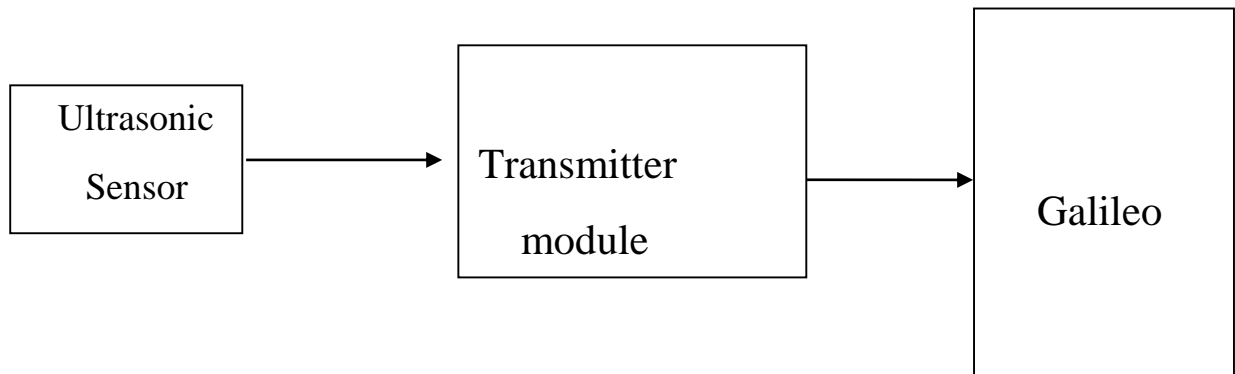


Fig 1: Block Diagram of Drainage Monitoring System

### 5.2 ULTRASONIC SENSOR

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic sensors measure the distance to the target by measuring the time between the emission and reception.

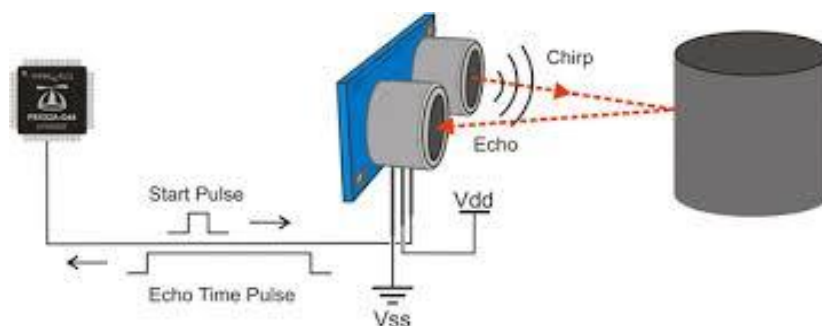


Fig 2: Description of Ultrasonic Sensor

Ultrasonic Sensors are self-contained solid-state devices designed for non-contact sensing of solid and liquid objects. For many applications, such as monitoring the level of water in a tank, ultrasonic technology lets a single device to do a job that would otherwise require multiple

sensors. These sensors are available in several sensing ranges and styles with either an analog or discrete output depending on the model.

An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

Ultrasonic distance measuring sensors provide information on an absolute position of a target or moving object. For glossy surfaces, transparent objects or in environments with a high degree of dust and humidity, ultrasonic technologies are often the only alternative to mechanical probing.

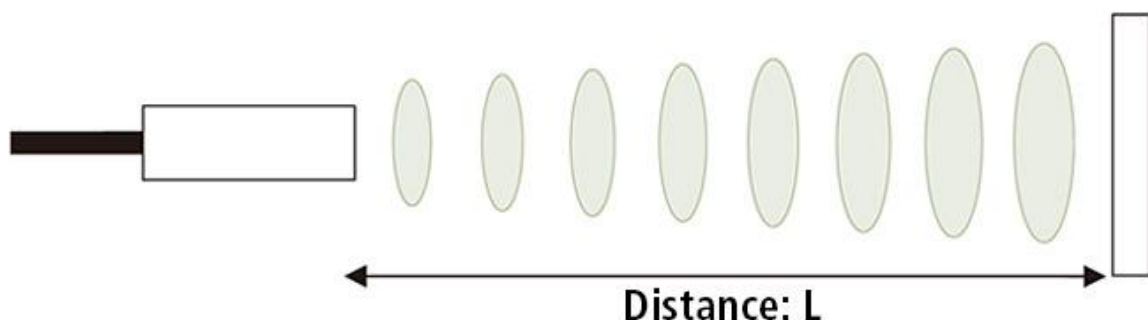


Fig 3: Measurement of Distance using Ultrasonic Sensors

The distance can be calculated with the following formula:

$$\text{Distance } L = \frac{1}{2} \times T \times C$$

Where L is the distance, T is the time between the emission and reception, and C is the sonic speed. (The value is multiplied by 1/2 because T is the time for go-and-return distance.)

## FEATURES

The following list shows typical characteristics enabled by the detection system.

**Transparent object detectable:**

Since ultrasonic waves can reflect off a glass or liquid surface and return to the sensor head, even transparent targets can be detected.

**Resistant to mist and dirt:**

Detection is not affected by accumulation of dust or dirt.

**Use of Ultrasonic sensor:**

1. Ideally suited to accurate, automatic distance measurement in normal and difficult environments
2. Particularly suitable for environments where optical sensors are unusable such as smoke, dust and similar.
3. Very accurate, stable and can be used over large ranges.

Ultrasonic sensors can measure the following parameters without contacting the medium to be measured:

Distance

Level

Diameter

Presence

Position

Ultrasonic sensors make accurate measurements in many difficult environments and unusual materials. Measurements are unaffected by:

Material

Surface

Light

Dust

**5.3 INTEL GALILEO**

**Intel Galileo** is the first in a line of Arduino-certified development boards based on Intel x86 architecture and is designed for the maker and education communities. Intel released two versions of Galileo, referred to as Gen 1 and Gen 2. These development boards are sometimes called "hacker boards".

Intel's Galileo Gen 2 Board is the first in a family of Arduino-certified development boards based on Intel® architecture and specifically designed for makers, students, educators, and DIY electronics enthusiasts. Based on the Intel Quark™ SoC X1000, a 32-bit Intel Pentium® processor-class system on a chip (SoC), the genuine Intel processor and native I/O capabilities of the Intel Galileo board (Gen 2) provide a full-featured offering for a wide range of applications. Arduino-Certified and designed to be hardware-, software-, and pin-compatible with a wide range of Arduino Uno R3 shields. Additionally it allows users to incorporate Linux firmware calls in their Arduino sketch programming.

Galileo is designed to support shields that operate at either 3.3V or 5V. The core operating voltage of Galileo is 3.3V. However, a jumper on the board enables voltage translation to 5V at the I/O pins. This provides support for 5V Uno shields and is the default behavior. By switching the jumper position, the voltage translation can be disabled to provide 3.3V operation at the I/O pins.

Of course, the Galileo board is also software compatible with the Arduino Software Development Environment (IDE), which makes usability and introduction a snap. In addition to Arduino hardware and software compatibility, the Galileo board has several PC industry standard I/O ports and features to expand native usage and capabilities beyond the Arduino shield ecosystem. A full sized mini-PCI Express slot, 100Mb Ethernet port, Micro-SD slot, RS-232 serial port, USB Host port, USB Client port, and 8MByte NOR flash come standard on the board.

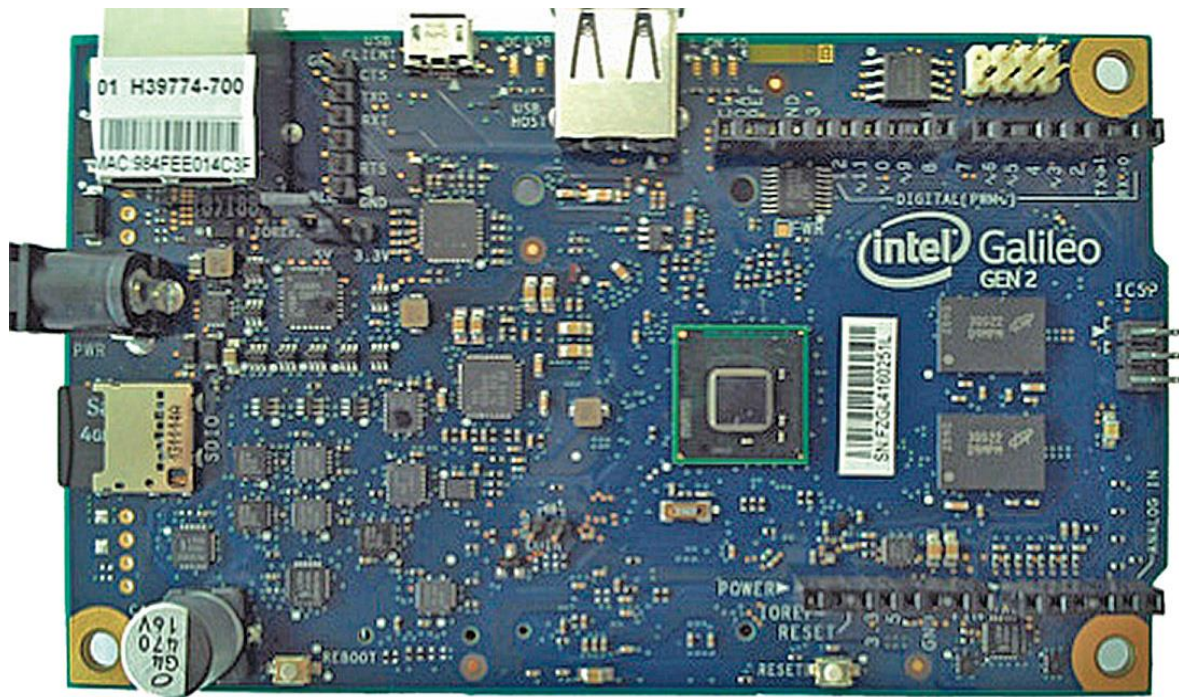


Fig 4: Intel Galelio Gen2 Board

### Detail of Intel Architecture Supported Features

The genuine Intel processor and surrounding native I/O capabilities of the Clanton SoC provides for a fully featured offering for both the maker community and students alike. It will also be useful to professional developers who are looking for a simple and cost effective development environment to the more complex Intel Atom processor and Intel Core processor-based designs.

400MHz 32-bit Intel Pentium instruction set architecture (ISA)-compatible processor o 16 KBytes on-die L1 cache

- o 512 KBytes of on-die embedded SRAM
- o Simple to program: Single thread, single core, constant speed
- o ACPI compatible CPU sleep states supported
- o An integrated Real Time Clock (RTC), with an optional 3V "coin cell" battery for operation between turn on cycles

10/100 Ethernet connector

Full PCI Express mini-card slot, with PCIe 2.0 compliant features

- Works with half mini-PCIe cards with optional converter plate
- Provides USB 2.0 Host Port at mini-PCIe connector
- USB 2.0 Host connector
- Support up to 128 USB end point devices

#### **USB Device connector, used for programming**

- Beyond just a programming port - a fully compliant USB 2.0 Device controller
- 10-pin Standard JTAG header for debugging
- Reboot button to reboot the processor
- Reset button to reset the sketch and any attached shields

#### **Storage options:**

- Default - 8 MByte Legacy SPI Flash main purpose is to store the firmware (or bootloader) and the latest sketch. Between 256KByte and 512KByte is dedicated for sketch storage. The download will happen automatically from the development PC, so no action is required unless there is an upgrade that is being added to the firmware.
- Default 512 KByte embedded SRAM, enabled by the firmware by default. No action required to use this feature.
- Default 256 MByte DRAM, enabled by the firmware by default.
- Optional micro SD card offers up to 32GByte of storage
- USB storage works with any USB 2.0 compatible drive
- 11 KByte EEPROM can be programmed via the EEPROM library.

### **5.4 PROGRAMMING**

Galileo can be programmed with the Arduino software (download). When you are ready to upload the sketch to the board, program Galileo through the USB Client port by selecting "Intel Galileo" as your board in the Arduino IDE. Connect Galileo's port labeled USB Client (the one closest to the Ethernet) to your computer. For details, see the reference, tutorials and Intel Galileo Getting Started Guide. Rather than requiring a physical press of the reset button before an upload, Galileo is designed to be reset by software running on a connected computer.

When the board boots up two scenarios are possible:

If a sketch is present in persistent storage, it is executed



If no sketch present, the board waits for upload commands from the IDE

If a sketch is executing, you can upload from the IDE without having to press the reset button on the board. The sketch is stopped; the IDE waits for the upload state, and then starts the newly uploaded sketch.

Pressing the reset button on the board restarts a sketch if it is executing and resets any attached shields.

**Here are some of the best features of this ground breaking collaboration between Intel and Arduino:**

### **Shield Compatibility**

The expansion header on the top of Galileo should look familiar since it's compatible with 5V and 3.3V Arduino shields designed for the Uno R3 (also known as the Arduino 1.0 pinout). This means that it has 14 digital I/O pins, 6 analog inputs, a serial port, and an ICSP header.

### **Familiar IDE**

The Intel-provided integrated development environment for the Galileo looks exactly like the Arduino IDE on the surface. Under the Boards menu, you'll see addition of the Galileo under "Arduino X86 Boards." The modified IDE also is capable of upgrading the firmware on the board.

### **ETHERNET LIBRARY COMPATABILITY**

Using the Ethernet port on the board is as simple as using Arduino's Ethernet library. I was able to get a HTTP connection to Google without even modifying the standard WebClient example.

### **Real Time Clock**

Most Linux boards rely on a connection to the Internet to get the current date and time. But with Galileo's on-board RTC (real time clock), you'll be able to track time even when the board is powered off. Just wire up a 3.0V coin cell battery to the board.

### **Works with PCI Express Mini Cards**

On the bottom of the board is an expansion slot for PCI Express Mini cards. This means you can connect WiFi, Bluetooth, GSM cards for connectivity, or even a solid state drive for more storage. When you connect a WiFi card, it will work with Arduino's Wifi library.

### **USB Host Port**

Galileo's dedicated USB On-The-Go port will let you use the the Arduino USB Host library to act as a keyboard or mouse for other computers.

### **MicroSD Support**

If you want to store data, a microSD card slot is accessible from your code by using the standard Arduino SD card library.

### **TWI/I2C, SPI Support**

Using the standard Arduino Wire library or SPI library, you can connect TWI/I2C or SPI components to the Galileo.

### **Serial Connectivity**

Not only is there the typical serial port for your sketches on pins 0 and 1 of the Arduino pinout, but there's also a separate serial port for connecting to the Linux command line from your computer. You'll connect to it through the audio jack interconnect next to the Ethernet port. This port is only used for serial.

### **Linux on Board**

A very light distribution of Linux is loaded onto the 8 MB of flash memory. If you want to use tools like ALSA (for sound), V4L2 (for video input), Python, SSH, node.js (for web projects), and openCV (for computer vision), you can boot Galileo from an SD card image that Intel provides.

## MODULE-2

### 5.5 BLOCK DIAGRAM

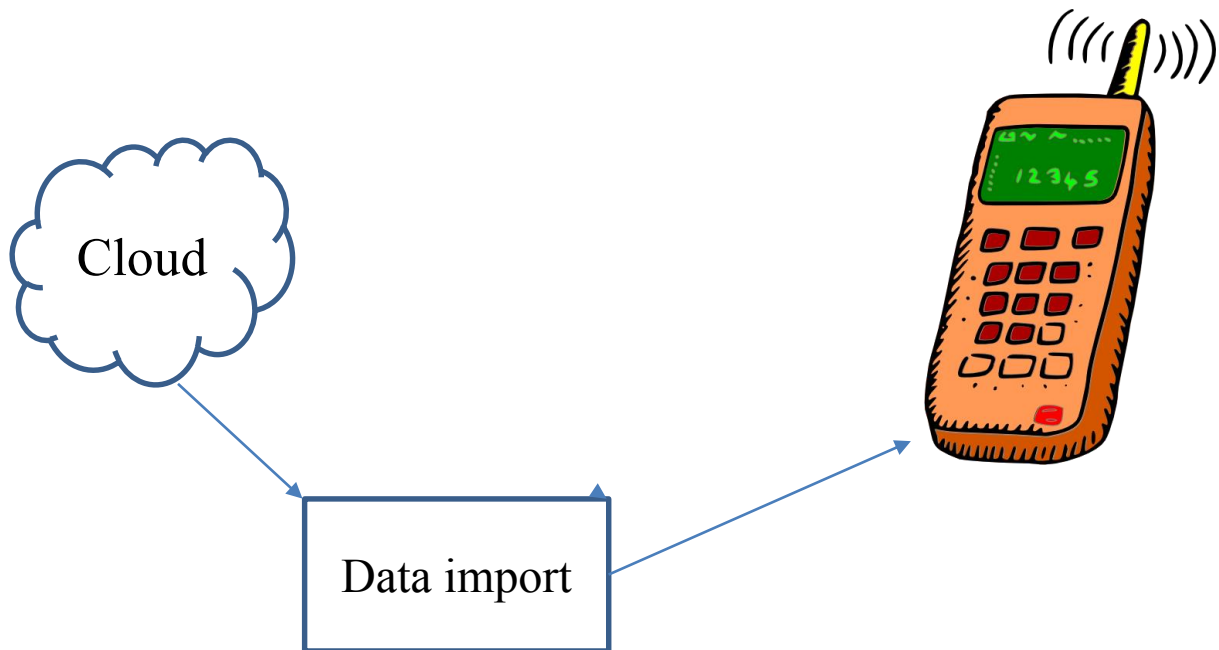


Fig 5: Block Diagram For Generating Alerts

### 5.6 DATA BASE MANAGEMENT SYSTEM

Contains general data obtained from various time instances with which the data received from the circuits will be compared and generate the alerts.

In the context of IoT, data management should act as a layer between the objects and devices generating the data and the applications accessing the data for analysis purposes and services. The devices themselves can be arranged into subsystems or subspaces with autonomous governance and internal hierarchical management. The functionality and data provided by these subsystems is to be made available to the IoT network, depending on the level of privacy desired by the subsystem owners.

Both offline and real-time data cycles need to be supported in an IOT-based data management system, to accommodate the various data and processing needs of potential IoT users.

*Querying:* In the context of IoT, a query can be issued either to request real-time data to be collected for temporal monitoring purposes or to retrieve a certain view of the data stored

within the system. The first case is typical when a (mostly localized) real-time request for data is needed. The second case represents more globalized views of data and in-depth analysis of trends and patterns.

*Production:* Data production involves sensing and transfer of data by the “Things” within the IoT framework and reporting this data to interested parties periodically (as in a subscribe/notify model), pushing it up the network to aggregation points and subsequently to database servers, or sending it as a response triggered by queries that request the data from sensors and smart objects. Data is usually time-stamped and possibly geo-stamped, and can be in the form of simple key-value pairs, or it may contain rich audio/image/video content, with varying degrees of complexity in-between.

*Collection:* The sensors and smart objects within the IoT may store the data for a certain time interval or report it to governing components. Data may be collected at concentration points or gateways within the network where it is further filtered and processed, and possibly fused into compact forms for efficient transmission. Wireless communication technologies such as Zigbee, Wi-Fi and cellular are used by objects to send data to collection points.

*Delivery:* As data is filtered, aggregated, and possibly processed either at the concentration points or at the autonomous virtual units within the IoT, the results of these processes may need to be sent further up the system, either as final responses, or for storage and in-depth analysis. Wired or wireless broadband communications may be used there to transfer data to permanent data stores.

*storage/Update—Archiving:* This phase handles the efficient storage and organization of data as well as the continuous update of data with new information as it becomes available. Archiving refers to the offline long-term storage of data that is not immediately needed for the system's ongoing operations

- Alerts will be generated when the blockage starts developing
- Time available to repair the blockage will be enough to prevent the drainage line from completely shutting down.
- This technology can be used for all fluid carrying systems. (e.g. Water pipes, gas pipelines)

## 5.7 AURDINO

The Arduino Integrated Development Environment - or 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 Genuino hardware to upload programs and communicate with them.

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures Single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

## 5.8 SOFTWARE DEVELOPMENT:

A program for Aurdino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages *Processing* and *Wiring*. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching,

and syntax highlighting, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a *sketch*. Sketches are saved on the development computer as text files with the file extension *.ino*. Arduino Software (IDE) pre-1.0 saved sketches with the extension *.pde*.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

## **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 righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

## **Tabs, Multiple Files, and Compilation**

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (*.c* extension), C++ files (*.cpp*), or header files (*.h*).

## Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. Theboards are described below. On the Mac, the serial port is probably something like `/dev/tty.usbmodem241` (for an Uno or Mega2560 or Leonardo) or `/dev/tty.usbserial-1B1` (for a Duemilanove or earlier USB board), or `/dev/tty.USA19QW1b1P1.1` (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be `/dev/ttyACMx` ,`/dev/ttyUSBx` or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

# **CHAPTER VI**

## **RESULT ANALYSIS**



## RESULT AND ANALYSIS

This chapter shows the final output scenario of the project. The working of the project is shown in this chapter.

### 6.1 OUTPUT



Fig 6:Output

- The result of the above drainage monitoring system shown are shown in fig
- Date and drainage level graphs are shown the drainage monitoring system using IOT.
- Water level before and after drain block are provided.
- Locates the area with drainage block.
- Clogs in the drainage system gets identified using these circuit. And the concerned authorities are notified through cloud and GSM.

# **CHAPTER VII**

## **CONCLUSION**

## **CONCLUSION:**

Many cities across the world are facing drainage system problems. Heavy Rain fall causes damage in roads and loss of valuable human hours affecting in one or other ways the country economy. There is a concern situation coming to scene frequently these days. Smart drainage system development and testing of emphasis (sensory drains) is currently carried out and expected to go on a trail in the real drains in coming next years.

The drainage monitoring system using Internet of Things was proposed in this project is to collect data from circuits and transmits the collected data to the processor and to analyse the drainage monitoring system through cloud and sends an alert to the corresponding Municipal authorities. This technique enables us to detect information of sewage flow through that node at different locations and send it to the central system which will generate alerts prior to complete blockage.

The Theme of this project is to design a Drainage monitoring System to reduce the effect caused by water logging during floods.

# **CHAPTER VIII**

## **REFERENCES**

## References

1. Lazarescu, M.T., "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on* , vol.3, no.1, pp.45-54, March 2013.
2. Manhole cover type sensor node apparatus US 20120206270A1 publication date 16 Aug 2012
3. Kelly, S.D.T.; Suryadevara, N.K.; Mukhopadhyay, S.C., "Towards the Implementation of IoT for Environmental Condition Monitoring in Homes," *Sensors Journal, IEEE*, vol.13, no.10, pp.3846-3853, Oct. 2013.
4. Romer, K.; Mattern, F., "The design space of wireless sensor networks," *Wireless Communications, IEEE* , vol.11, no.6, pp.54-61, Dec. 2004
5. J M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hmlinen, M. Hnnikinen, and T. D. Hmlinen, *Ultra-Low Energy Wireless Sensor Networks in Practice*. New York: Wiley, 2007.
6. K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Comm.*, vol. 11, no. 6, pp. 54–61, Dec. 2004.
7. "Manhole explosions causes and case studies", by W.Z.Black Georgia Institute of Technology, Atlanta GA June2013
8. Yan Xijun., Lu Limei., and XuLizhong., "The Application of Wireless Sensor Network In the Irrigation Area Automatic System", *International Conference on Networks Security, Wireless Communications and Trusted Computing*, Vol.1, pp.21-24, 2009
9. K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Comm.*, vol. 11, no. 6, pp. 54–61, Dec. 2004
10. P. Harrop and R. Das, "Wireless Sensor Networks 2010–2020," *IDTechEx Ltd*, Downing Park, Swaffham Bulbeck, Cambridge, CB25 0NW, United Kingdom, Report, 2010.
11. N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: Ultra-Low Power Data Gathering in Sensor Networks," in *Information Processing in Sensor Networks*, Apr. 2007, pp. 450–459.
12. I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 5:1–5:39, Feb. 2009.
13. C. Jardak, K. Rerkrai, A. Kovacevic, and J. Riihijarvi, "Design of largescale agricultural wireless sensor networks: email from the vineyard," *International Journal of Sensor Networks*, vol. 8, no. 2, pp. 77–88, 2010.

14. P. I. Fierens, "Number of wireless sensors needed to detect a wildfire," *International Journal of Wildland Fire*, no. 18(7), pp. 625–629, 2009
15. LiuJinHuaGaoGongLiang. Based on G S M of groundwater level and rainfall on-line monitoring system [J]. *Automation technology and application*, 2007, 6 (I) • 28-29
16. S. M. Khaled Reza., Shah Ahsanuzzaman Md. Tariq., and S.M. Mohsin Reza., "Microcontroller Based Automated Water Level Sensing and Controlling: Design and Implementation Issue", *Proceedings of the World Congress on Engineering and Computer Science*, Vol 1, pp.1-7, 2010
17. Wireless Real-time System for Monitoring the Storage of Urban Storm Drainage, YinHailongXuZuxin, DOI 10.1109/NCIS.2011.18711
18. Christian Reinisch., Wolfgang Kastner., Georg Neugschwandtner., and Wolfgang Granzer., "Wireless Technologies in Home and Building Automation", *5th IEEE international conference on Industrial Informatics*, Vol. 1, pp.93-98, 2007