Title of Dissertation:

Design and Feasibility Study of WSN based Cyber-Physical System(CPS) in Flash Flood Forecasting

Name of Candidate: **Bipendra Basnyat**
Masters in Information Systems

Abstract Approved: **Yes**
Signature of Supervising Professor

nirmalya

Nirmalya Roy
Assistant Professor
Department of Information Systems
Date Approved: 01/30/2017

Abstract:

Title of Document: **Design and Feasibility Study of WSN based Cyber-Physical System(CPS) in Flash Flood Forecasting**

Bipendra Basnyat, P.E. Masters of Science,
Information Systems, 2017

Directed By: **Assistant Professor, Nirmalya Roy, Information Systems**

This thesis focuses on documenting the feasibility and challenges faced during design and deployment of the cyber-physical system(CPS) for early warning Flashflood Event. We have developed a mobile wireless sensor system that can monitor the rising level of water, generate the real-time alert and send out notifications to the concerned personnel. We will present our design, the learning experience in developing such system, its success, reliability and sustainability. We have also explored supplementing our environmental sensors with the Social Sensor by collecting, processing, and aggregating streams of social media data during and after the crisis. We will present our finding on the possibility of such multimodal data fusion and future research direction in being able to develop robust distributed data analytics platform for Real-Time Monitoring and Detection of Flash Flood in the Smart City

Design and Feasibility Study of WSN based Cyber-Physical System(CPS) in Flash Flood Forecasting

By

Bipendra Basnyat, P.E.

Thesis submitted to the Faculty of the Graduate School of the University of Maryland, Baltimore County, in fulfilment of the requirements for the

degree of Masters of Science in Information Systems 2017

Acknowledgements

This work would not have been completed without the guidance for Dr Nirmalya Roy. More than one time, "I have had it" moments with the device and would like to dump the entire setup. His calmness and unwavering belief on me to get this device functioning are why I could complete this thesis. As my mentor, I am grateful to have an opportunity to work with Dr Roy. I equally thank and pay my heartfelt due respect to my co-advisor Dr Gangopadhyay, for enthusiastically taking part in our research project and finding time to guide me despite his immensely busy schedule. Both of their openness in allowing me to explore various research ideas has taken me to avenues that would not have been possible otherwise.

Finally, I believe that this research is just a beginning of my stepping stone into the much wider and deeper world of Cyber-Physical Systems design and implementation. I will always be grateful to all my past present and future mentors. Also, the NSF/GCTC fund which made this research a reality.

Last but not the least, to my family far and near, I thank you for all the support always I receive from you.

Table of Contents

I. Chapter I Introduction

We are already living in the world of ubiquitous computing, smartphone, smart watch, smart home and now the smart city. The smart city movement is about integrating the sophisticated world of Information and Communication technology (ICT) into the daily chores of that city's citizen. Though there has been a various and evolving definition of the Smart City, one that is relevant to this research is "s an implicit or explicit ambition of a city to improve its economic, social and environmental standards" [1]. Thus, to provide a better quality of life for the citizen living there in, many innovative solutions are proposed on a regular basis and infrastructure upgrade and maintenance tops of the list. City's prosperity is defined by two types of infrastructure: physical (e.g. buildings, roads, transportation, and power plants) and digital (information technology (IT) and communications infrastructure). [2]. Therefore, it is eminent that the level of infrastructure must be top notch to have a better life for their citizens. This means making the use of physical infrastructure more efficient through artificial intelligence and data analytics by increasing public participation in local governance and decision-making process through collective innovativeness in the community. The smart city is one that can learn, adapt, and innovate, empowered to respond more effectively and promptly to changing circumstances by improving the intelligence of the city. The integration of physical world and such infrastructure is achieved by harnessing the power of the Internet of Things (IoT). IoT is the network of physical objects, devices and other items embedded with electronics, software, sensors, and network connectivity—that enables these objects to collect and exchange data. [3]

Once the city can provide adequate services under, technology and infrastructure, sustainability, governance and economy, next measure for its lasting success are its readiness to assist the citizen during the disaster situation. It is also an essential duty of a city is to facilitate the health, safety and security of its citizens. Cities may face various problems like increasing population, unprecedented weather manifestations, natural disasters, unemployment, unique geography, poverty, crime, and other social problems that pose a serious threat to the stable functioning of the city. [1]. With the ongoing smart city initiatives, government from around the world are using technological innovations to make a paradigm-shift to tackle the above challenges in urban environments. Thus, an increasing amount of data is collected and brought together at various levels to enable police officials to provide better security, doctors and health care professionals to enhance health care treatments, and inform governmental officials to solve social problems more effectively.

We have witnessed many recent calamities that prove that the disaster management is a critical component to consider when designing a smart sustainable city. Enabling interoperability between the first respondents and other corresponding civic agencies significantly hampers the rescue efforts [4]. To this end, the exploration of the use and potential of ICT in disaster management has come to light.

Water-induced disasters are one of such major security and disaster concern. There can be larger damaging effects such as the hurricane, the flood which devastates the city to its core or there can be minor nuisance problem such as puddle on the road, unsafe driving condition or smaller community level flooding event. Sometimes, water induced disaster could also be the result of dysfunctional manmade structures such as a dam, levy, embankments of water and sewer facilities.

This research specifically tackles the water induced disaster specially in Flashflood situation. The national weather service defines Flashflood as a flooding that begins within 6 hours, and often within 3 hours, of the heavy rainfall (or other cause). [5]. A major challenge with the flash flood is that it occurs so quickly that people are often caught off-guard. The situation soon becomes dangerous as water rises too fast for one to move to the higher safer ground. Flash Flooding is a common occurrence and can happen on tranquil streams and creeks, in the neighborhood or area.

Flash Flooding can also occur on city streets and highway underpasses [5]. Hence, it is imperative that we induce some ICT into the problem and help the city to mitigate the problem and save some life. Flash Flood and Storm Water Management Monitoring is one such infrastructure which still lacks the proper integration of ICT and connectivity to the real world. City officials must perform quality assurance and regular inspection of such facilities by visiting the site often during harsh weather condition.

Thus, with this Research and Development Deployment Project, we hope to deploy various sensors and IoT that can remotely monitor/sense the infrastructure and transmit data and deliver a functional cyber-physical system. Our system will enable us to gather the real-time data and empower us to act before the disaster. Further integration of these sensor data into existing Hydrologic and Hydraulics model can allow for real-time flash flood prediction, create community watchdog system and devise for the social media integration. We hence describe both our hardware and software architectures, the motivations for the design of our system and our cases. how we implemented it. We provide results gathered during deployments, about the network, the sensing stations, and, of course, the environment. We also share the experience we have acquired and go through the lessons we learned, detailing the problems we faced.

i. Study Objective

- Design a self-sustaining mobile Wireless Sensor Network (WSN) and integrate it with the real-world Application
- Assess and validate the design, construction and implementation of such Cyber-Physical systems under various condition.
- Get our feet wet (literally) into the world of Cyber-Physical System by assessing the effectiveness of Flashflood Monitoring and Early Warning Systems Design.

- Gather and transmit real-time weather-related data during and after storm events.
- Establish a framework of infrastructure to store the Information gathered by the Cyber-Physical System
- Develop a Single node proof of concept System Design that can be expanded in future for fully integrated use in City Officials Decision Support System.

ii. Significance of the Problem

The State of Maryland has witnessed the various scale of flood damages since the MDE started keeping records. The earliest record found for substantial damage in the state has been since 1933. [6]. The literature available on the internet suggests that most of the previous project undertaken to mitigate the flood hazard has been at the macro level. In other words, ample studies and data exist for larger stream networks and subsequent flood mitigation measure has been implemented around them. The ultimate outcome of such studies and projects is a 100-year flood prediction/mitigation plan. A 100-year Flood simply means that on average the flooding will happen once in 100 years. It is very unlikely that the public would be keen to take part in understanding cause and effect of and the environmental phenomenon that has such a low probability, obviously, a major setback in building the vibrant smart city. What people care is if there is a puddle on their community roadway, or has the smaller streams behind their backyard risen enough to cause the surge in use of their sump pumps.

Thus, with the proposed technology we hope to bring the information at a macro level to the public in a fashion that matters to them. When they understand the real impact, it might encourage them to take necessary measures that can ultimately help in the overarching goal of Clean Water Blueprint for the Chesapeake and its rivers and streams.

As stated above Most of our existing stormwater management facilities are designed and driven by the 100-year storm. Only within last decade, it has been realized that management and treatment of such infrequent and quantitative runoff do not solve the real problem of stream degradation and biological habitat thereby. MDE has already conceded that just managing peak flows for the 2 and 10-year storms do not provide sufficient stream channel erosion protection from the increased runoff caused by urbanization. [7]. The effectiveness of facilities during 100-year storms is even less prominent. A more frequent storm event such as 1-yr storm is being used to gauge the effectiveness of stormwater management facilities.

However, the problem lies in our dependency to regional, historical or even imperial data for our design parameters. We do not have enough rain gauges or other environmental sensors in place to effectively address the concern raised by a microclimatic factor. The existing Hydraulic and hydrologic (H&H) model ignores the microclimate effect often created near bodies of water which may cool the local atmosphere, or in heavily urban areas where brick, concrete, and asphalt absorb the sun's energy, heat up, and reradiate that heat to the ambient air. Among other, this phenomenon could directly impact on the amount of evaporation and ground infiltration. This can create a subtle failure to the effectiveness of the existing (H&H) model that are being used by city and county to design, build and monitor the system.

Thus, with the proposed technology we hope to capture real-time environmental features at the site such as rainfall amount, temperature, humidity and soil condition etc. This information when integrated with the (H&H) model can be used to validate the effectiveness of existing design paradigm.

Therefore, there is still a need to develop micro-climatic models that can better represent the ground condition. It will then pave a pathway to other urban weather forecasting models that can produce urban and building simulation ready weather data series adapted to the local microclimate within an acceptable accuracy [3].

iii. Motivation

While flooding and loss of human lives are not uncommon phenomena around the world, we had a very close encounter with such an unfortunate incident in nearby town, Ellicott City, Maryland. On Saturday, July 30th, 2016, torrential rainfall occurred around Ellicott City, Maryland, resulting in the severe flash flood. The event led to two fatalities, destruction to the historic city, and damages to hundreds of vehicles. People were enjoying their Sunday evening in the historic Ellicott City when an unprecedented storm swept through the area.

According to the meteorologists, such massive rainfall was a rare 1-in-1,000-year event. The National Weather reported that the storm dumped 6.5 inches of rain on Ellicott City in only about 3 hours, with 5.5 inches falling in just 90 minutes. The storm killed two people and damaged many buildings and vehicles parked on the road. Within minutes, the touristic street turned into a raging river, carrying away cars and other debris and forcing dramatic rescues of people trapped in flood. Though, this event not completely stoppable, series of similar devices as we have developed would have allowed time for better information dissemination. We see this event as our motivation and plan to implement our design solutions to see if we can propose an easy-to-deploy and-configure WSN that can help in collecting the data and server as an early warning system. We envision our system to be able to sense and extract the knowledge from the data and use this data to intelligently react to serve as the life-saving device.

iv. Outline of the Thesis

The thesis is organized into eight chapters. A brief overview of the structure is given below.

Chapter 2 is devoted to a review of related literature. The chapter documents similar research work that have been performed in the context. We look at the existing work and present our overview of what has been and how does our work contribute to the community.

Chapter 3 presents high level process flow of our system. It gives a broader picture of what we are trying to achieve with the design

In **Chapter 4**, detail discussion on the projects hardware design elements are discussed. We talk about the sensors and hardware that we have designed as a part of this Project.

In **Chapter 5**, we discuss the software components of this system. We document the software application and code that has been implemented to get the design functional. This section discusses the underlying physical database design and our approach to using them in further analysis.

In **Chapter 6**, we discuss various prototypes that we have built and present the research paradigm thereby. This chapter documents the evolution of our prototypes and our learning process from each iteration.

Chapter 7 presents our analysis on comparing our device with alternate device. We validate our design with the commercial product with similar functionality.

Chapter 8 presents data analysis and our preliminary findings on the data that we have collected so far. We use various statistical measures to understand our data and present their properties and use cases.

Chapter 9 documents the risk and challenges associated with such development. We document our own learning and risk in faced during the development of this research work.

Chapter 10 presents the conclusions of the research based on the various use cases presented in this thesis. Finally, we conclude by the possible directions for further research are suggested

Chapter II Related Work

Cyber-Physical systems design my definition possess two components the cyber or digital systems and physical or earthly elements. Unlike the cyber system which has matured over the time and is highly reliable now. The physical components, on the other hand, still possess many challenges. Furthermore, the physical components are ever evolving and its integration with the cyber system is not very reliable. Recent advances in microelectromechanical systems (MEMS) have opened up great opportunities for the implementation of smart environments [8]. Especially in the environmental science, several sensors are available to evaluate different environmental features such as temperature, humidity, soil moisture, air pollution monitors and, water quality gauge etc. Wireless networks are not a new approach to environmental monitoring and it is common to find systems where remote sensors in the field communicate to central points for data processing in a star network formation. [4]. Our research seeks to understand the risk and challenges associated with the deployment of environmental sensor network (ESN) and integrate it with our expertise in the areas of machine learning and pervasive computing. Interdisciplinary studies of ESN have come a long way since the first recorded such study a few decades ago. A very detailed analysis and learning from their own failure and finding a better working solution was well documented by Corke P and ET all [9].

We are aware of the challenges faced by previous researchers in this area and hope to build upon their failures and learning experiences. One of the learning experience that we gained from out literature review is the challenged that previous researchers had to go through wherein they had to build the system from scratch starting with a computer board. Thus, our approach was to use plug and play device and spend more time and resource is understanding the post data collection. Hence, we have opted to use a Commercially Off the self-plug and sense equipment along with the associated sensors. [10].

Our system is expected to provide meaningful and timely information about the potentially disastrous situation during the storm. Thus, we would need to build a clear understanding of the hydraulic and hydrologic condition and other microclimatic variables of the site. It has been reported that the hydrological study often poses a known risk of spatial heterogeneity. Most models attempt to capture the distributed nature of the landscape, but the availability of spatiotemporal data as input (initial and boundary conditions) has generally been lacking [11].

In fact, it is not an atypical situation in hydrology that engineers are bound to extrapolate data from a regional far distant sensor. Engineers are confronted with the problem of making predictions without spatial information. [4] Our system would, therefore, be driven towards capturing small watershed area where more frequent storm event such as 1 yr. storm has been found to create minor flooding. The other criteria would be to select either the roadways or other publicly accessible recreational areas. We believe that the proposed solution will enable us to capture data from small areas and provide better design parameters for the engineers. We acknowledge that this kind of research is impossible without a fine-tuned hydraulic and hydrologic model running in the background, the scope of this research will be limited toward extensive use of H&H models. We would rather aim to develop a minimum viable solution around integrating IoT and machine learning. The core functionality of this research is be to exploit basic level of H&H modeling techniques and focus more towards integrating the Cyber(WSN,ML,DSS,Social)-physical elements (storm water runoff) in this case to trigger our Wireless Sensor Network and create an automated Information flow system to feed and generate machine learning algorithm.

Various types of applications can be devised using the contemporary wireless sensor networks (WSNs). These applications share a common structure, where fields of sensors are tasked to take periodic readings, and report results and derived values to a central repository [12].As shown in conceptual deployment plan below, these platforms, or sensor nodes, can be fixed or mobile.

Coordinated communication and interaction among these nodes provide a local fusion of the dispersed data and results in a spatiotemporal understanding of the environment [4].

Chapter III Systems Design

As discussed earlier, the main objective of this research is to document our design process and walk through each element of the system. To, that effect the methodology section will present information about our system design and its working principle.

i. Distributed Units and Testbed Setup

The ultimate benefit of our system will be achieved when we have many interconnected sensor systems that are cable of collecting and transmitting the data in real time. Thus, we envision our final build out as a mesh of macro-instrument comprised of spatially-distributed sensor platforms. (***Figure 3.1***). The scenario or the location map can also be referred as a test-bed. The test-bed would be selected to demonstrate a complete sensing network, including robust wireless communications and self-sustained solar as a power source.

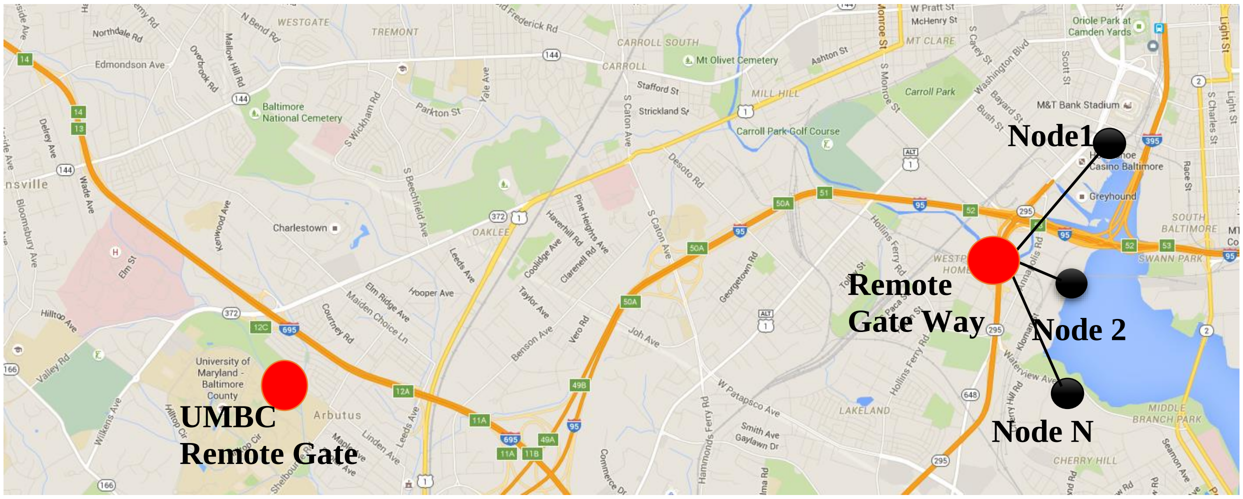


Figure 3.1: Ultimate Build Out Plan (Test Bed Setup) of the Device

Initially, these test-beds would be designed as low scale system limited by numbers of sensors yet be meaningful to be able to solve some real-world problems.

ii. Process Flow Diagram

The design of our system can be described by five functional modules. The function model or process flow diagram of our system is depicted in ***Figure 3.2*** below. The components included five major functional elements. shows how different kind of sensor reads the data from the natural environment (flooded city), captures, transfers them for information extraction. Once the information is processed and validated as an emergency, the emergency alert will be active or notifications sent to the concern parties.

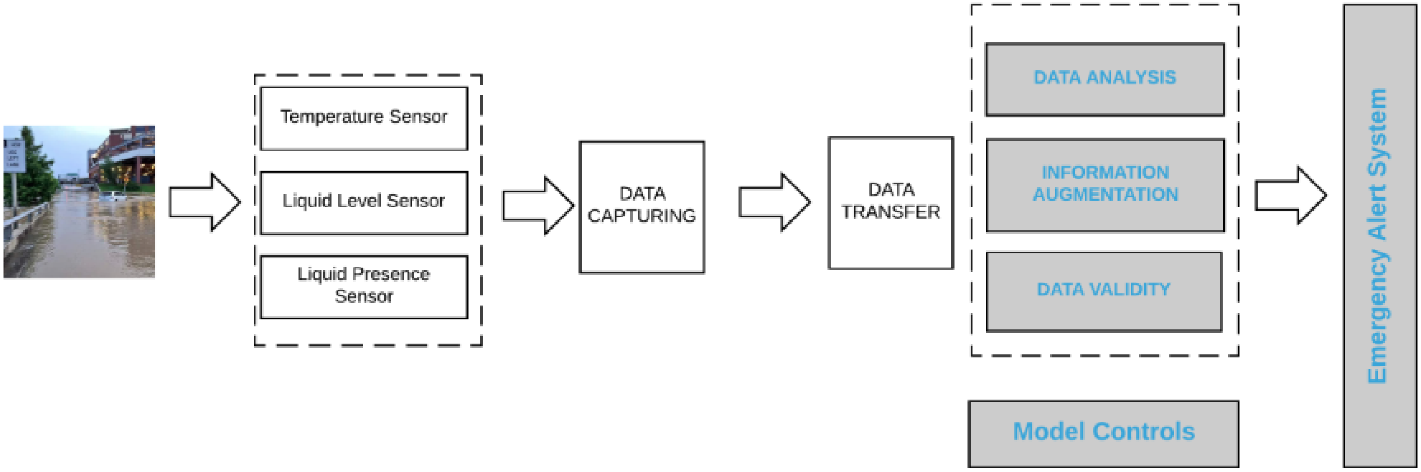


Figure 3.2: Process Flow Diagram

iii. System Components

To achieve the above modality, we have designed multiple hardware and software components as a part of our WSN based CPS system: a) the sensors b) hardware interface, c) networking system, d) Webservers. Overall end to end process flow and involved in our system in depicted in ***Figure 3.3.***

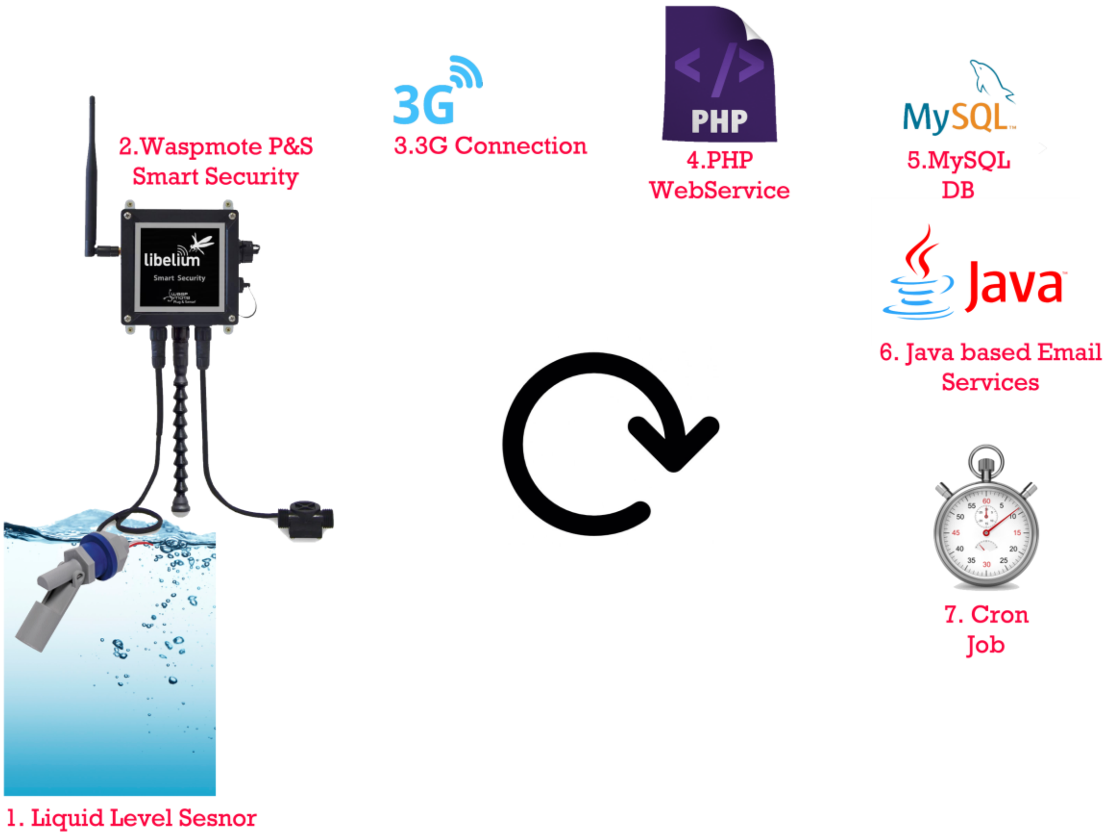


Figure 3.3: High-level System Components

At a high level, the system receives sensory pulses from the physical components such as temperature reading, the presence of liquid or liquid level threshold, it then triggers the hardware unit. The hardware device temporarily stores information and transmits it to the web via 3G data transfer protocol. We have designed a web service which listens to the data transferred by the hardware device and stores into the database. An automated emailing system was deployed in the web server.

Chapter IV Hardware Design

While the overall goal of this research was to design an emergency alert system during the disaster, every little component within the system had to be designed carefully. The system data flow diagram is depicted in ***Figure 4.1*** below. Each of the elements used in the research design project is discussed below.

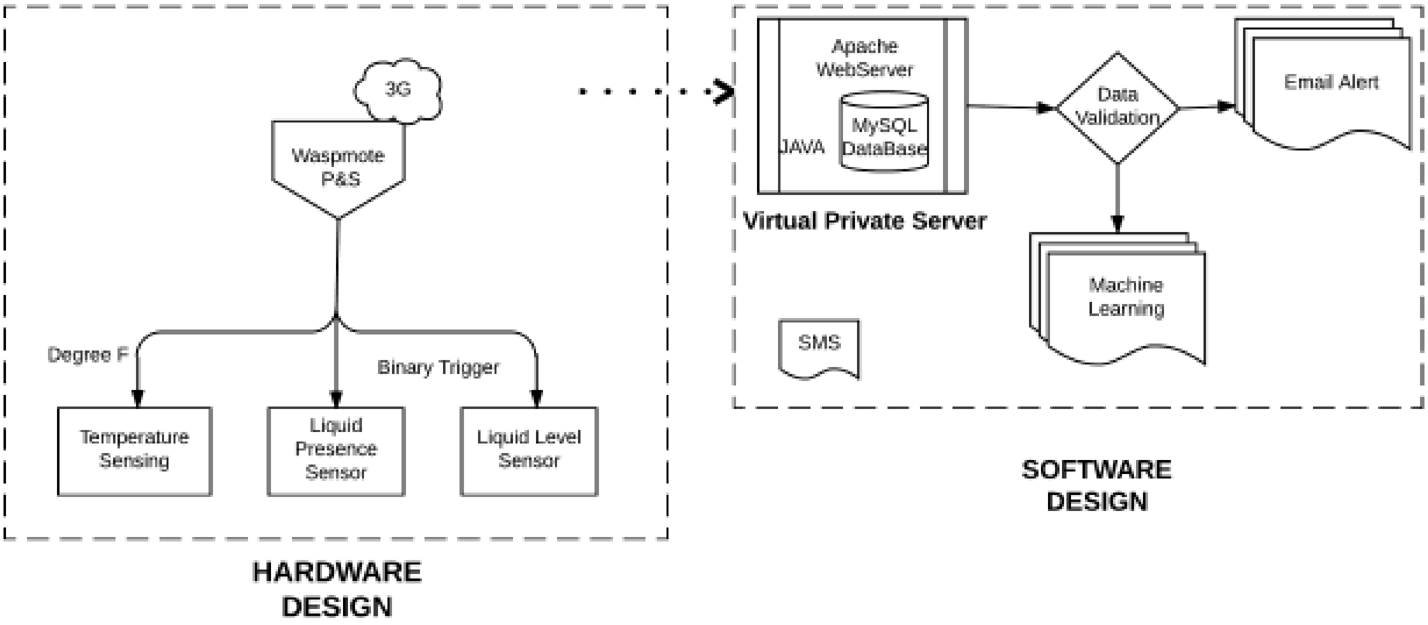
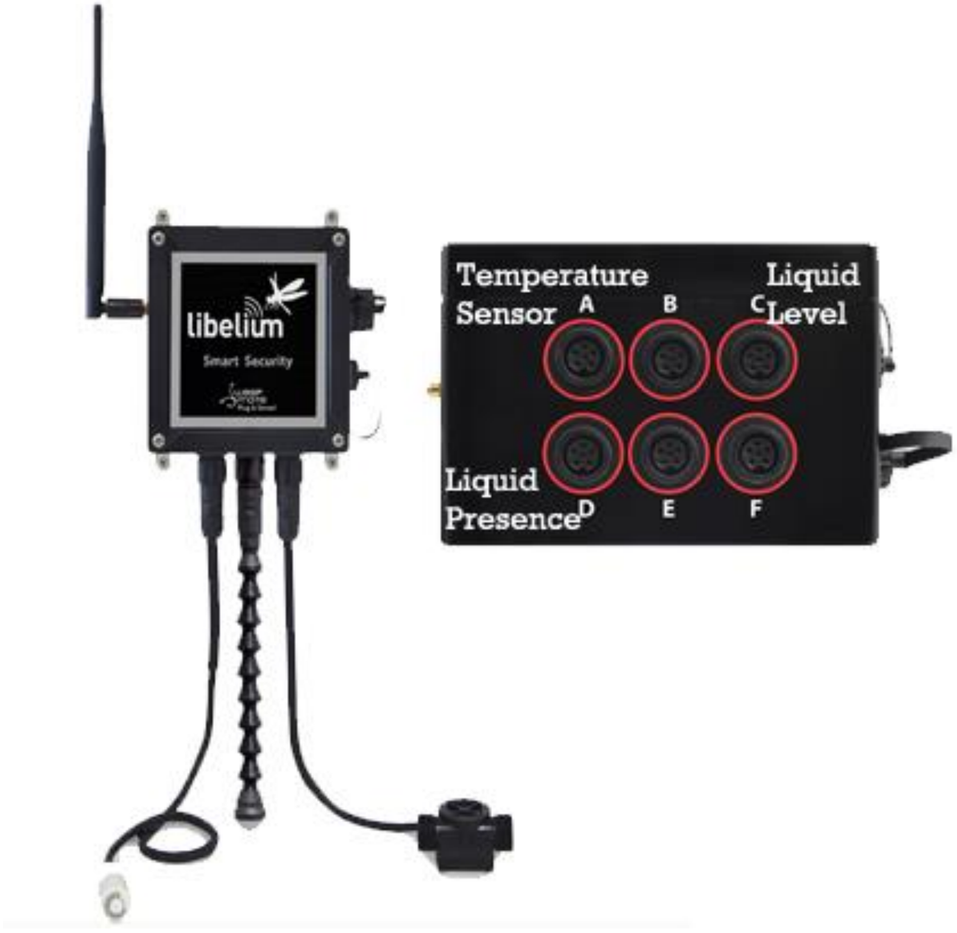


Figure 4.1: Hardware and Software System

i. Waspmote Plug and Sense(P&S)

The main component or the brain of the sensor system is called Waspmote Play and Plug System. Leading Spanish company called Libelium produces many similar systems. According to their website [10], the company produces ten models that can integrate more than 120 sensors. The sensors are external probes that can be easily replaced. The unit is enclosed in a waterproof enclosure with specific external sockets to connect the sensors. The device can be connected via SB cable to reprogram the node. The company makes many modules of Plug and Sense Unit such as Smart Cities, Smart Environment, Smart Security, Smart Agriculture, Smart Parking etc. All these units work for a domain with their corresponding sensors. The one that we used for our research was called ***Smart Security P&S Unit***. Their unit also has a built-in acceleration sensor which informs the mote of acceleration variations experienced on each one of the 3 axes (X, Y, Z). However, these units' capabilities were not used in our research. Waspmote Plug & Sense! has an internal SD (Secure Digital) card with up to 2GB storage. The battery in the unit is rechargeable and is charged through the solar panel. The solar power provides max power: 3 W, voltage: 5.8 V and the current: of 520 mA, measures 234 x 160 x 17 mm, weighs about 0.54 kg. [10]

Figure 4.1 Waspmote Security P&S System



ii. Sensor Probes

As depicted above in ***Figure 4.1,*** there is six sensors probe to each of the Waspmote's P&S unit. The sensors used for our project and their location are labelled in the diagram. Table 4.2 lists all

Sensor Socket	Sensor probes allowed for each sensor socket	
	Parameter	**Reference**
A, C, D or E	Temperature + Humidity + Pressure	9370-P
	Luminosity (Luxes accuracy)	9325-P
	Ultrasound (distance measurement)	9246-P
	Presence - PIR	9212-P
	Liquid Level (combustible, water)	9239-P, 9240-P
	Liquid Presence (Point, Line)	9243-P, 9295-P
	Hall Effect	9207-P
B	Liquid Flow (small, medium, large)	9296-P, 9297-P, 9298-P
F	Relay Input-Output	9270-P

Table 4.2: Waspmote Security P&S Sensor Socket Configuration [10]

a) Temperature and Humidity Sensor probe

The temperature sensor used in the research as provided by Waspmote is of the model SHT75 sensor by Sensirion. This sensor can measure both humidity and the temperature. The sensor output is read through two wires following a protocol like the I2C bus (Inter- Integrated Circuit Bus) implemented in the library of the board, returning the temperature value in Celsius degree (ºC) and the humidity value in relative humidity percentage (%RH). [10]. Since the US units are based in Fahrenheit degree (ºF), we explicitly used the mathematical function to convert the units and store temperature in Fahrenheit degree (ºF). According to Waspmote technical specification, the unit can measure temperature from 40(ºC) to 123(ºC) and its response time is 5 seconds.

Figure 4.2a Horizontal Liquid Level Sensor

b) Liquid Level Sensor

The liquid level sensor used for this research is a horizontal sensor(PTFA3415) whose operation is based on the status of a switch which can be opened and closed (depending on its placing in the container) as the level of liquid moves the float at its end) [10].

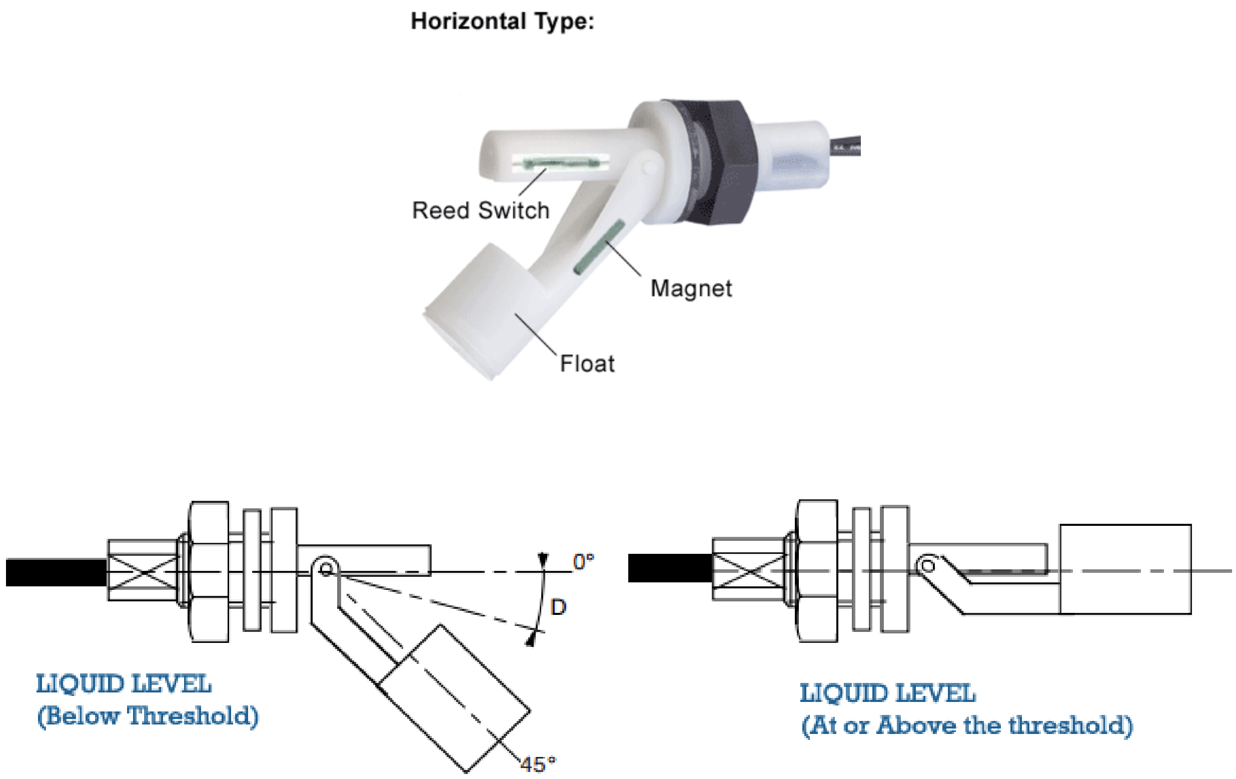


Figure 4.2b Horizontal Liquid Level Sensor

The working principle of this float switch device is simple and well-proven technologies for liquid level sensing. They comprise of a magnet contained within a float, as well as a magnetic reed switch contained within a fixed housing. The movement of the float, due to the changing liquid level, will cause the reed switch to operate (i.e. close or open) at a level. The magnet when in contact completes the circuit which sends the electric pulse to the Waspmote device that in turn fires the triggering effects.

c) Liquid Presence sensor probe (Point)

This is a Capacitance Sensors Which bases its operation on the variation in resistance between its two contacts in the presence of the liquid to commute a switch reed from open to closed, commuting to open again when the liquid disappear. This is also based on a similar principle of "circuit break and completion" however the metallic plates are capacitors and air creates the vacuum between these probes. Once liquid touches two probes, the vacuum no longer exists and the circuit is completed.

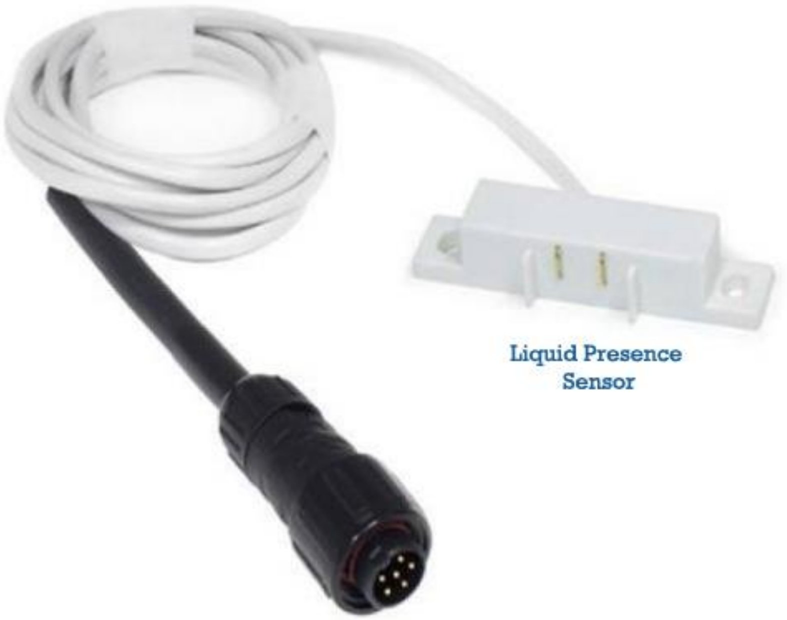


Figure 4.2c Liquid Presence Sensor

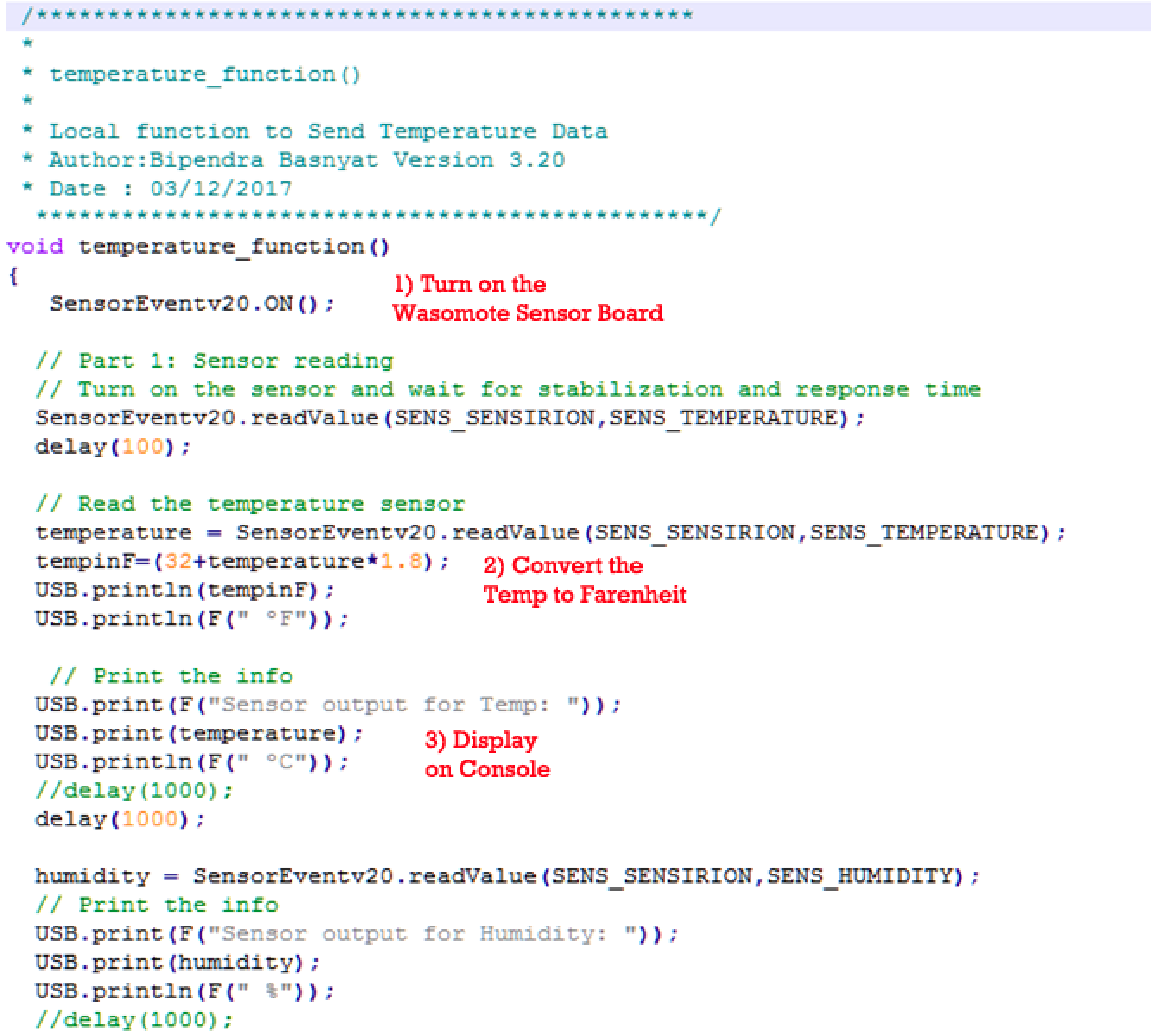
Chapter V Software Design

This research had a lot of software components involved with it. Once the Waspmote received the sensory data, its functionality would be complete. Though there are few third-party tools and mechanisms proposed by Waspmote to receive the data, these processes were expensive and unpractical for our purpose. To that end, we had to learn and design all the software components for this device. This was most of the challenging and time-consuming work involved throughout the design process and involved a lot of learning curve as well.

i. Waspmote IDE Language(Sketches)

Though the device calls itself a Plug and Sense, there is no easy way for us to get the data out. As a matter of fact, learning their language and Integrated Development Environment for Waspmote (Waspmote IDE) was the toughest part of this research. This IDE is used for writing the code and uploading it to Waspmote and Waspmote Plug & Sense. It also used to monitor serial output and for debugging. This IDE contains the Waspmote API (the API is the set of all libraries Waspmote needs for compiling programs) [10]. The piece of code written for a specific purpose are called sketches. These sketches are written in the text editor. Sketches are saved with the file extension ".pde". The message area gives feedback while saving and exporting and displays errors. The console displays text output by the IDE including complete error messages and other information. Only one setup code can be uploaded into the device and the latest code replaces all other code. The IDE validates the code before uploading for syntax or logical errors. Since the Waspmote board is an Arduino system, the basic code/language is like Arduino. Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. [13].

A sample code snippet is included in the figure Below. This code reads the temperature and displays the reading on the IDE connected via USB cable.



```
/*******************************************************
 *
 * temperature_function()
 *
 * Local function to Send Temperature Data
 * Author:Bipendra Basnyat Version 3.20
 * Date : 03/12/2017
  ******************************************************/
void temperature_function()
{
   SensorEventv20.ON();

  // Part 1: Sensor reading
  // Turn on the sensor and wait for stabilization and response time
  SensorEventv20.readValue(SENS_SENSIRION,SENS_TEMPERATURE);
  delay(100);

  // Read the temperature sensor
  temperature = SensorEventv20.readValue(SENS_SENSIRION,SENS_TEMPERATURE);
  tempinF=(32+temperature*1.8);
  USB.println(tempinF);
  USB.println(F(" °F"));

   // Print the info
  USB.print(F("Sensor output for Temp: "));
  USB.print(temperature);
  USB.println(F(" °C"));
  //delay(1000);
  delay(1000);

  humidity = SensorEventv20.readValue(SENS_SENSIRION,SENS_HUMIDITY);
  // Print the info
  USB.print(F("Sensor output for Humidity: "));
  USB.print(humidity);
  USB.println(F(" %"));
  //delay(1000);
```

1) Turn on the Wasomote Sensor Board

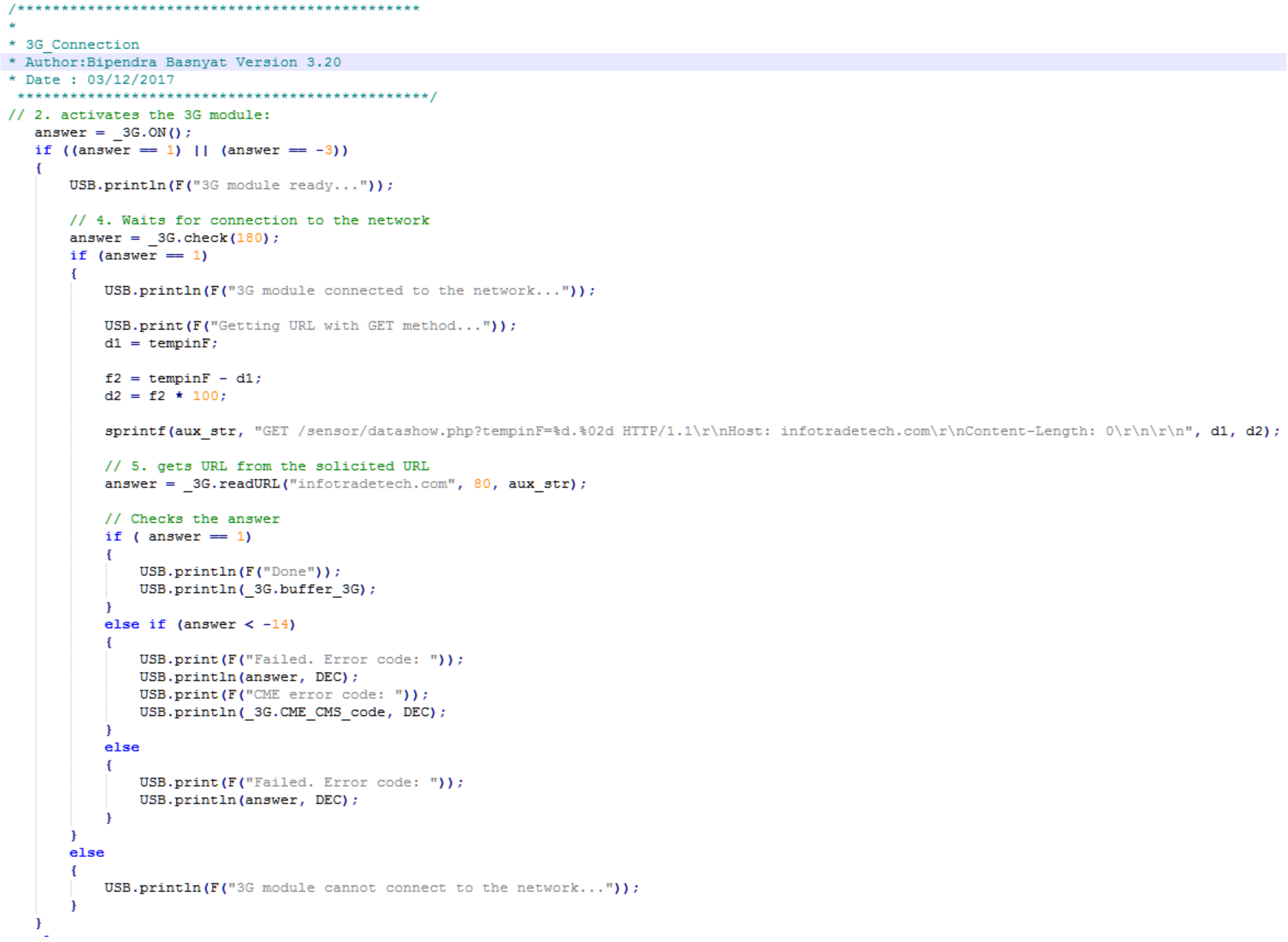
2) Convert the Temp to Farenheit

3) Display on Console

Figure 5.1 Sample Code in Waspmote IDE

ii. 3G Data Transfer from Waspmote IDE

This is the most critical and import part of our system design. We had to struggle a lot for this piece of code to work. It took us more than three months to figure out that the device given by Libelium was incompatible with US network providers such as AT&T, Verizon and T-Mobile. After rigorous follow up on emails and community forums, it was finally figured out that the device does not work in US 3G bandwidth and Libelium had to replace the device. Ultimately we are now able to make this device work with a subsidiary of T-Mobile called as TING.



```
/*********************************************
*
* 3G_Connection
* Author:Bipendra Basnyat Version 3.20
* Date : 03/12/2017
 *********************************************/
// 2. activates the 3G module:
   answer = _3G.ON();
   if ((answer == 1) || (answer == -3))
   {
       USB.println(F("3G module ready..."));

       // 4. Waits for connection to the network
       answer = _3G.check(180);
       if (answer == 1)
       {
           USB.println(F("3G module connected to the network..."));

           USB.print(F("Getting URL with GET method..."));
           d1 = tempinF;

           f2 = tempinF - d1;
           d2 = f2 * 100;

           sprintf(aux_str, "GET /sensor/datashow.php?tempinF=%d.%02d HTTP/1.1\r\nHost: infotradetech.com\r\nContent-Length: 0\r\n\r\n", d1, d2);

           // 5. gets URL from the solicited URL
           answer = _3G.readURL("infotradetech.com", 80, aux_str);

           // Checks the answer
           if ( answer == 1)
           {
               USB.println(F("Done"));
               USB.println(_3G.buffer_3G);
           }
           else if (answer < -14)
           {
               USB.print(F("Failed. Error code: "));
               USB.println(answer, DEC);
               USB.print(F("CME error code: "));
               USB.println(_3G.CME_CMS_code, DEC);
           }
           else
           {
               USB.print(F("Failed. Error code: "));
               USB.println(answer, DEC);
           }
       }
       else
       {
           USB.println(F("3G module cannot connect to the network..."));
       }
   }
```

Figure 5.2 Sample Code in Waspmote IDE for 3G connection and HTTP Call

iii. SMS from Waspmote IDE

The device can send text message via its inbuilt SMS function. We had to configure other sensor level functionality but calling SMS service as a straight forward process and all we needed was to give the desired cell phone number in the configuration file.

iv. Data Transfer

Once the Waspmote sends data via 3G and http protocol, the functionality of the device completes. It was upon us to devise an entire system to handle the data storage for future analysis. The dataflow diagram/pseudo code for this portion is depicted in ***Figure 5.3*** below. There are multiple PHP files written and saved on our server. Once the server receives notification from Waspmote, the Php file identifies the data bit and stores them in their appropriate table. More detailed discussion about the database design will be done later in this document.

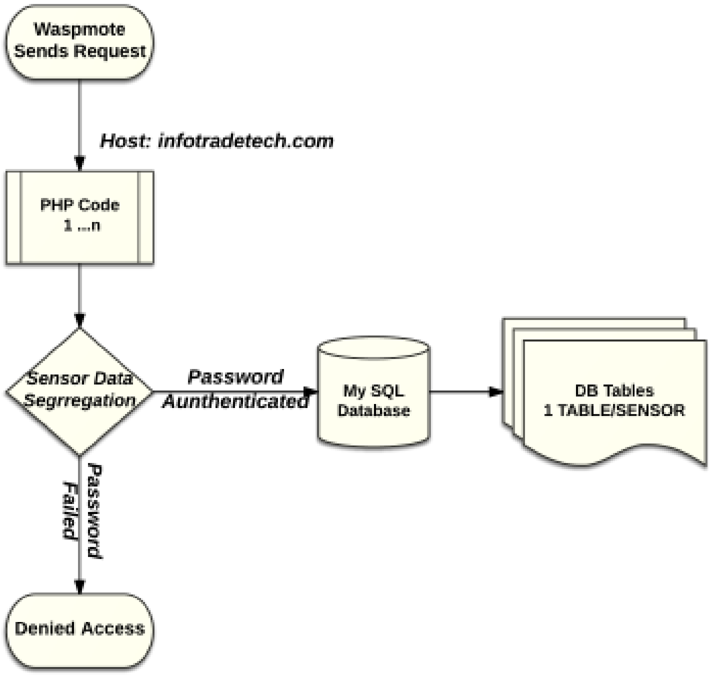


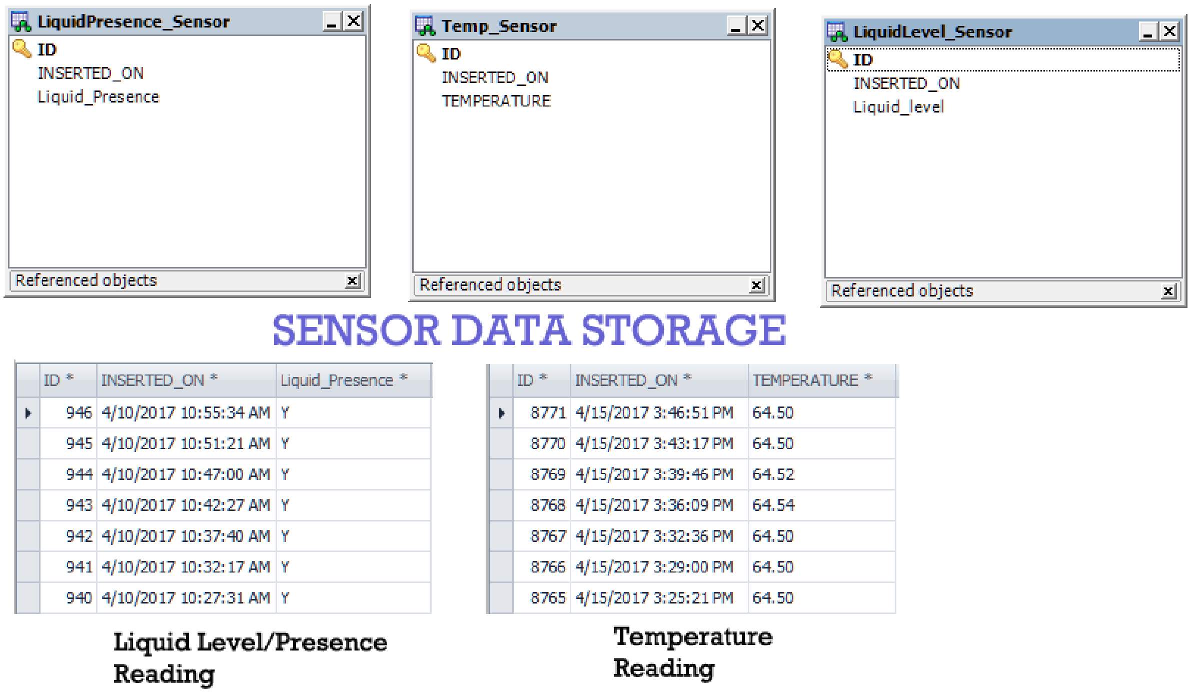
Figure 5.3 Waspmote & Server Connection

v. Data Storage

We have designed 5 tables to store the data for this project. We have explained their usages and types below.

a) Sensor Data Storage

The first three tables are the sensor related table out of which two tables LiquidPresence_Sensor and LiquidLevel_Sensor are triggered based. Since the sensor only sends a binary data for liquid level and liquid presence, we insert a timestamp with flag value as "Y”.



LiquidPresence_Sensor
ID
INSERTED_ON
Liquid_Presence
Referenced objects

Temp_Sensor
ID
INSERTED_ON
TEMPERATURE
Referenced objects

LiquidLevel_Sensor
ID
INSERTED_ON
Liquid_level
Referenced objects

ID *	INSERTED_ON *	Liquid_Presence *
946	4/10/2017 10:55:34 AM	Y
945	4/10/2017 10:51:21 AM	Y
944	4/10/2017 10:47:00 AM	Y
943	4/10/2017 10:42:27 AM	Y
942	4/10/2017 10:37:40 AM	Y
941	4/10/2017 10:32:17 AM	Y
940	4/10/2017 10:27:31 AM	Y

ID *	INSERTED_ON *	TEMPERATURE *
8771	4/15/2017 3:46:51 PM	64.50
8770	4/15/2017 3:43:17 PM	64.50
8769	4/15/2017 3:39:46 PM	64.52
8768	4/15/2017 3:36:09 PM	64.54
8767	4/15/2017 3:32:36 PM	64.50
8766	4/15/2017 3:29:00 PM	64.50
8765	4/15/2017 3:25:21 PM	64.50

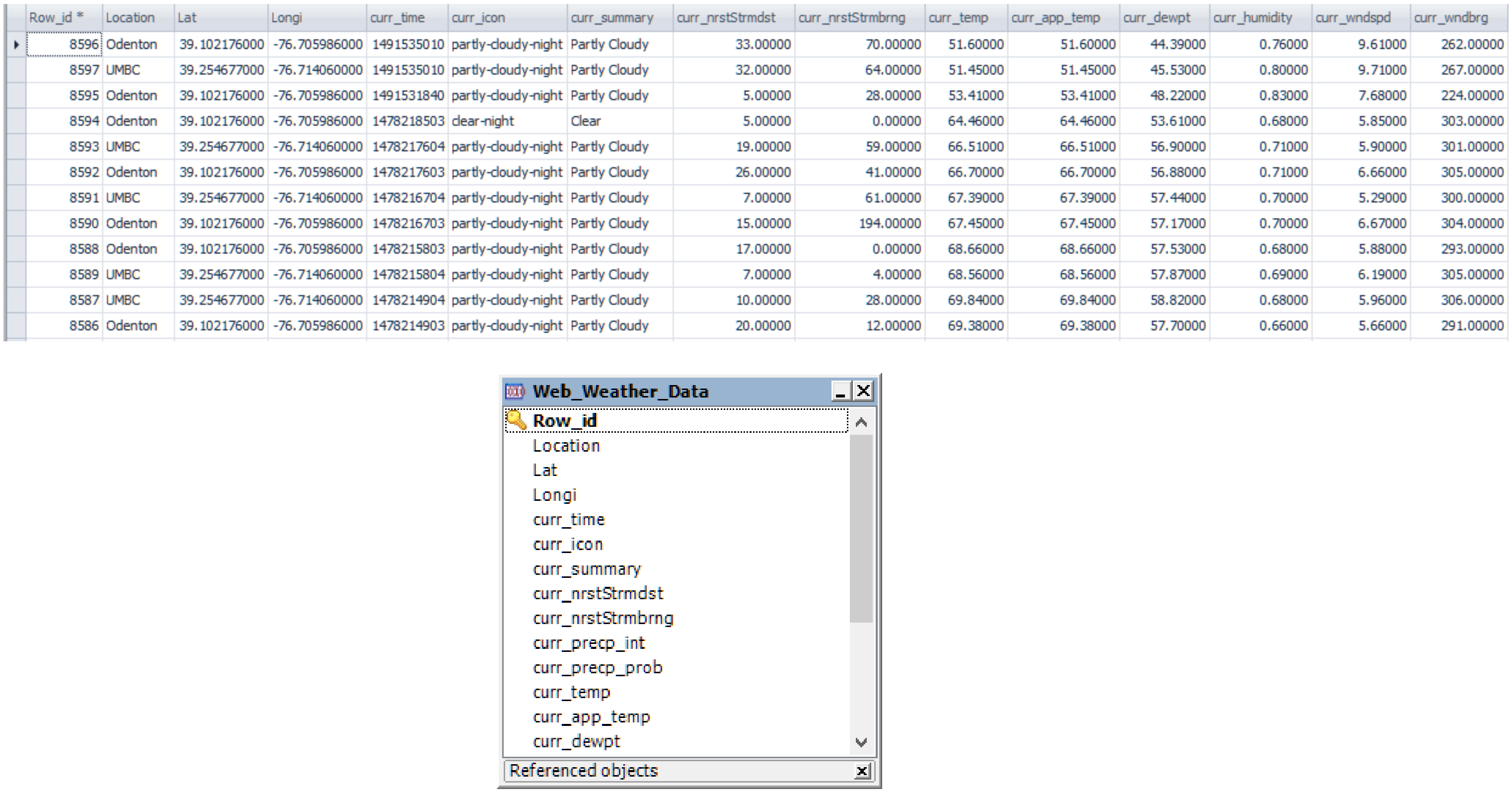
Figure 5.4a Sensor Data Storage

For the temperature storage, Waspmote transfer the actual temperature reading in degree Fahrenheit and we store the records. These times are tracked using a database timestamp function.

b) Web Data Storage

The second categorical table is created to store the data from Web service. We created a REST API to access the openWeatherMap portal and stored data in our database. To get access to weather API we needed an API key and connect to the given end, where the id could be any city we are interested. *HTTP://api.openweathermap.org/data/2.5/forecast?id=524901&APPID={APIKEY}*

The API is very flexible but had some limitation to the rate on how many requests we could make. We wrote a Php script to access their server and created a scheduler in cron to call the web service periodically and store the data in our database. The API delivers a lot of weather related information such as temperature, humidity, ozone, rainfall, rainfall intensity etc.



Row_id *	Location	Lat	Longi	curr_time	curr_icon	curr_summary	curr_nrstStrmdst	curr_nrstStrmbrng	curr_temp	curr_app_temp	curr_dewpt	curr_humidity	curr_wndspd	curr_wndbrg
8596	Odenton	39.102176000	-76.705986000	1491535010	partly-cloudy-night	Partly Cloudy	33.00000	70.00000	51.60000	51.60000	44.39000	0.76000	9.61000	262.00000
8597	UMBC	39.254677000	-76.714060000	1491535010	partly-cloudy-night	Partly Cloudy	32.00000	64.00000	51.45000	51.45000	45.53000	0.80000	9.71000	267.00000
8595	Odenton	39.102176000	-76.705986000	1491531840	partly-cloudy-night	Partly Cloudy	5.00000	28.00000	53.41000	53.41000	48.22000	0.83000	7.68000	224.00000
8594	Odenton	39.102176000	-76.705986000	1478218503	clear-night	Clear	5.00000	0.00000	64.46000	64.46000	53.61000	0.68000	5.85000	303.00000
8593	UMBC	39.254677000	-76.714060000	1478217604	partly-cloudy-night	Partly Cloudy	19.00000	59.00000	66.51000	66.51000	56.90000	0.71000	5.90000	301.00000
8592	Odenton	39.102176000	-76.705986000	1478217603	partly-cloudy-night	Partly Cloudy	26.00000	41.00000	66.70000	66.70000	56.88000	0.71000	6.66000	305.00000
8591	UMBC	39.254677000	-76.714060000	1478216704	partly-cloudy-night	Partly Cloudy	7.00000	61.00000	67.39000	67.39000	57.44000	0.70000	5.29000	300.00000
8590	Odenton	39.102176000	-76.705986000	1478216703	partly-cloudy-night	Partly Cloudy	15.00000	194.00000	67.45000	67.45000	57.17000	0.70000	6.67000	304.00000
8588	Odenton	39.102176000	-76.705986000	1478215803	partly-cloudy-night	Partly Cloudy	17.00000	0.00000	68.66000	68.66000	57.53000	0.68000	5.88000	293.00000
8589	UMBC	39.254677000	-76.714060000	1478215804	partly-cloudy-night	Partly Cloudy	7.00000	4.00000	68.56000	68.56000	57.87000	0.69000	6.19000	305.00000
8587	UMBC	39.254677000	-76.714060000	1478214904	partly-cloudy-night	Partly Cloudy	10.00000	28.00000	69.84000	69.84000	58.82000	0.68000	5.96000	306.00000
8586	Odenton	39.102176000	-76.705986000	1478214903	partly-cloudy-night	Partly Cloudy	20.00000	12.00000	69.38000	69.38000	57.70000	0.66000	5.66000	291.00000

Web_Weather_Data

Row_id
Location
Lat
Longi
curr_time
curr_icon
curr_summary
curr_nrstStrmdst
curr_nrstStrmbrng
curr_precp_int
curr_precp_prob
curr_temp
curr_app_temp
curr_dewpt

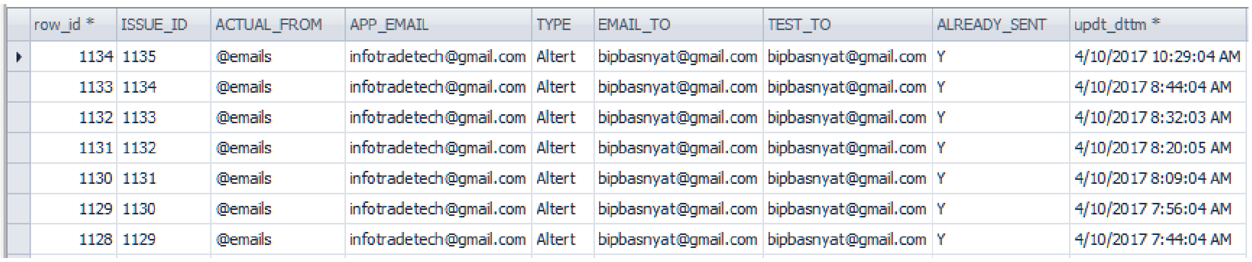
Referenced objects

Figure 5.4b Web Data Storage

vi. Email Alert System

We have also designed an email alert system. The system can send out emails to the persons that are prepopulated in our database. We envision to use this as our early warning system wherein the system will send out email messages alerting that the device has reached its pre-defined threshold level. We have created a table to store and track the email that goes out as an alert from the system. When Waspmote sends the binary notification that the liquid level has crossed its threshold, it inserts a row into the Liquid level sensors which have a trigger script to insert one more row into this email tracking table.

JAVA’s based email is sent out using google email client API. Our system check for the last row inserted into one of the alert table (liquid level senor) and sends out the email when a new row is inserted. Once the email is sent, we track it via already sent email flag. There is always a 1:1 relationship between alert email and the sensor reading. As depicted in the figure below, the sensor sends out the exact location of the WSN. Figures below show the sample data and flow of the email alert system.



	row_id *	ISSUE_ID	ACTUAL_FROM	APP_EMAIL	TYPE	EMAIL_TO	TEST_TO	ALREADY_SENT	updt_dttm *
▸	1134	1135	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 10:29:04 AM
	1133	1134	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 8:44:04 AM
	1132	1133	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 8:32:03 AM
	1131	1132	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 8:20:05 AM
	1130	1131	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 8:09:04 AM
	1129	1130	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 7:56:04 AM
	1128	1129	@emails	infotradetech@gmail.com	Altert	bipbasnyat@gmail.com	bipbasnyat@gmail.com	Y	4/10/2017 7:44:04 AM

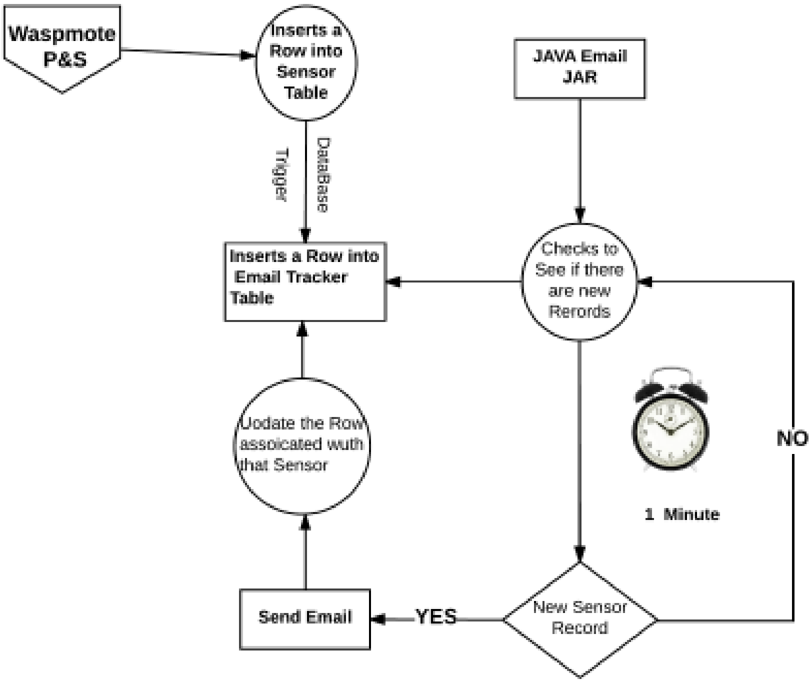
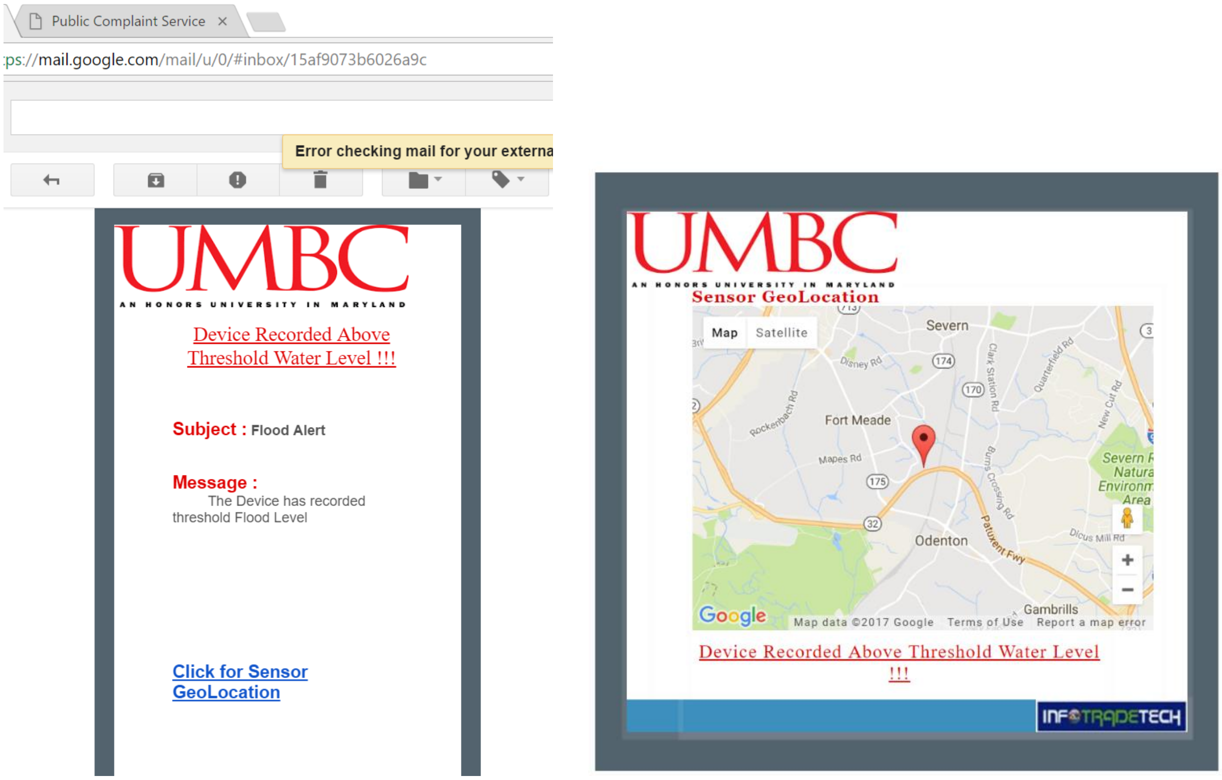

Waspmote P&S
Inserts a Row into Sensor Table
Trigger
DataBase
JAVA Email JAR
Inserts a Row into Email Tracker Table
Checks to See if there are new Rerords
Uodate the Row assoicated wuth that Sensor
NO
1 Minute
Send Email
YES
New Sensor Record

Figure 5.5 Email Alerts Flowchart


Public Complaint Service
ps://mail.google.com/mail/u/0/#inbox/15af9073b6026a9c
Error checking mail for your externa
UMBC
AN HONORS UNIVERSITY IN MARYLAND

Device Recorded Above Threshold Water Level !!!

Subject : Flood Alert

Message :
The Device has recorded threshold Flood Level

Click for Sensor GeoLocation

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Sensor GeoLocation

Map
Satellite
Severn
Fort Meade
Odenton
Gambrills
Google
Map data ©2017 Google
Terms of Use
Report a map error
INFOTRADETECH

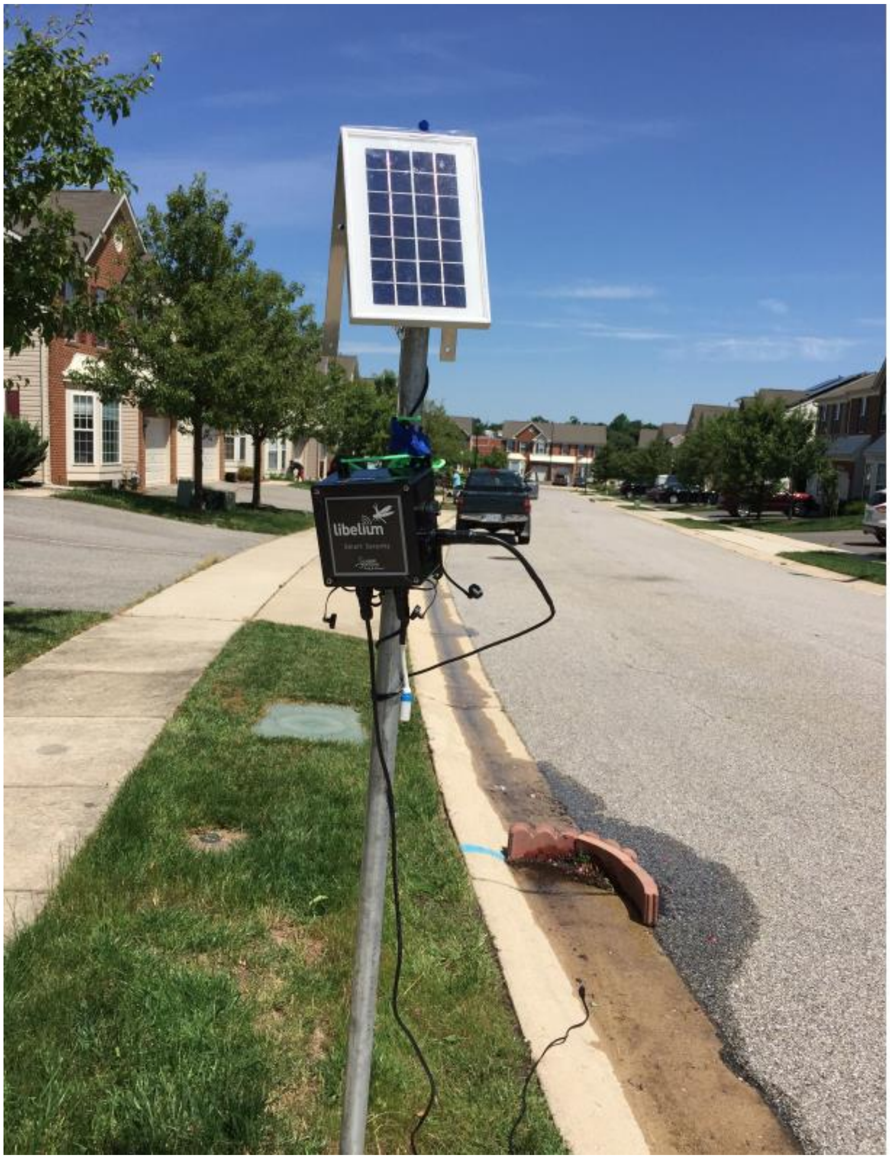
Device Recorded Above Threshold Water Level !!!

Figure 5.6 Email Alerts with Location Information

Chapter VI: Deployment in Heterogeneous Environments

i. Intermediator Prototypes

Over the period of last two semesters, the device has gone through various iterations and improvements. In this section, we document various deployments we have done and present our recall/learning. We had just received the device and were still trying to learn the code, understand how the system worked.



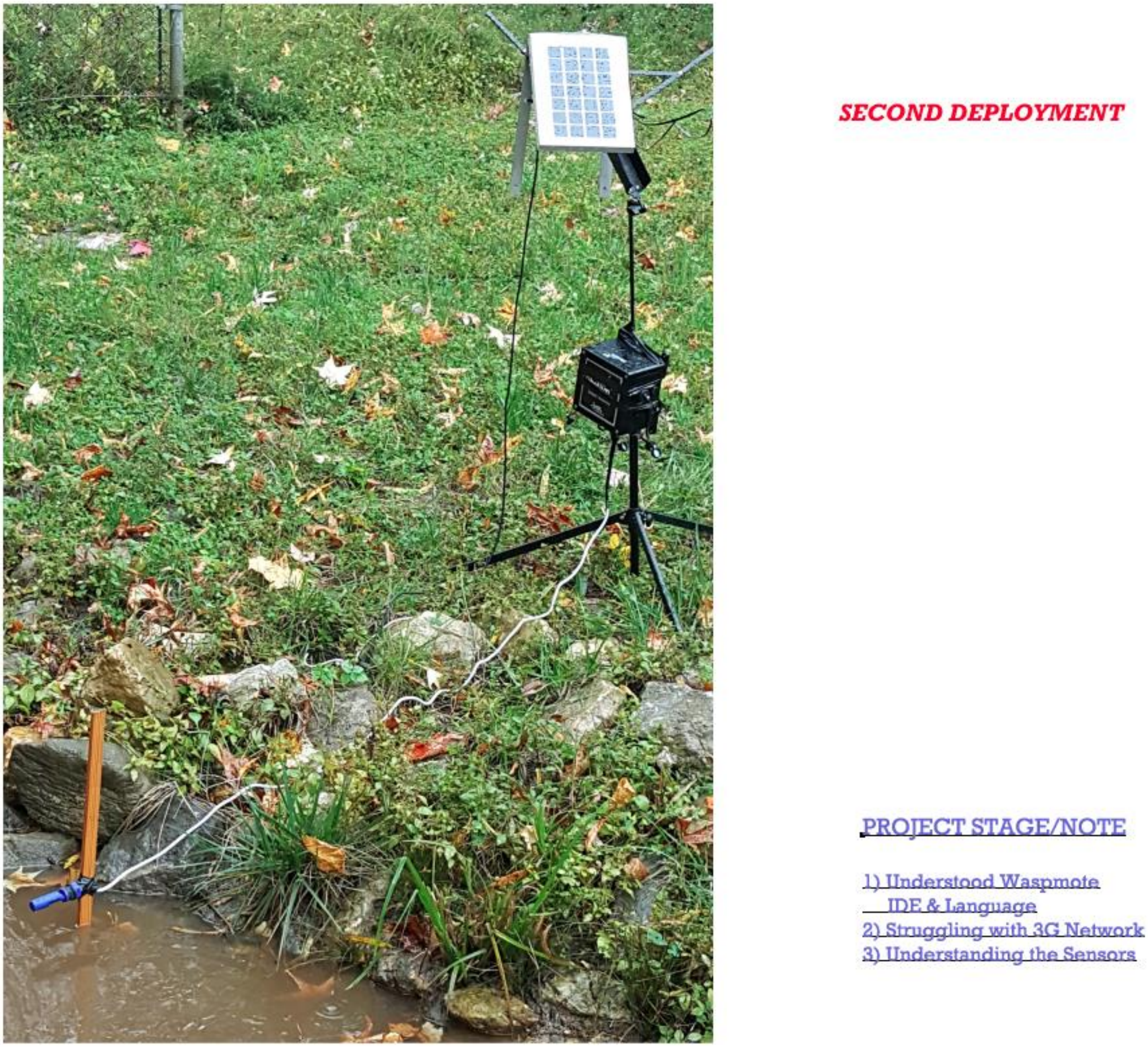
FIRST DEPLOYMENT

PROJECT STAGE/NOTE

1) Just Recieved the Device
2) Flood Simulation
3) Still Stuggling
with the Device
4) Understanding Waspmote
IDE & Language

Figure 6.1a First Deployment

This was our second deployment. We quickly learned that the sensor would receive lots of vibration and may be even swept away if we let it stand freely. We also realized that the structure of device needs to be in more stable condition.

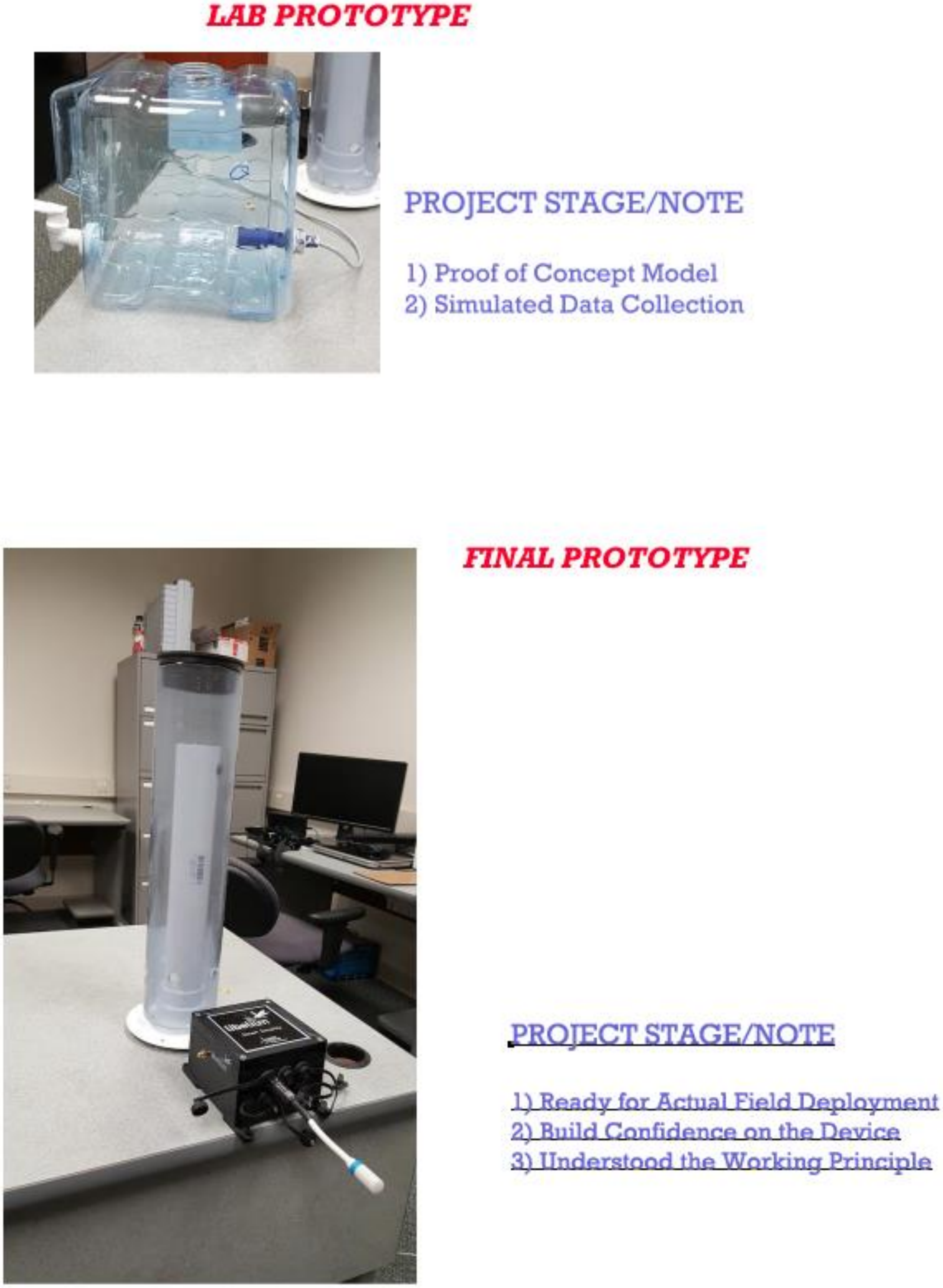


SECOND DEPLOYMENT

PROJECT STAGE/NOTE

1) Understood Waspmote
IDE & Language
2) Struggling with 3G Network
3) Understanding the Sensors

Figure 6.1b Second Deployment



LAB PROTOTYPE

PROJECT STAGE/NOTE

1) Proof of Concept Model
2) Simulated Data Collection

FINAL PROTOTYPE

PROJECT STAGE/NOTE

1) Ready for Actual Field Deployment
2) Build Confidence on the Device
3) Understood the Working Principle

Figure 6.1c Lab Deployment

ii. Final Prototype

Based on our previous learning experiences, we have finalized the prototype that can be deployed to the field. At the time of this thesis presentation we have had discussion with Howard County, Maryland officials and UMBC architect, and are in process to deploy this device on site. We are looking at two possible sites now.

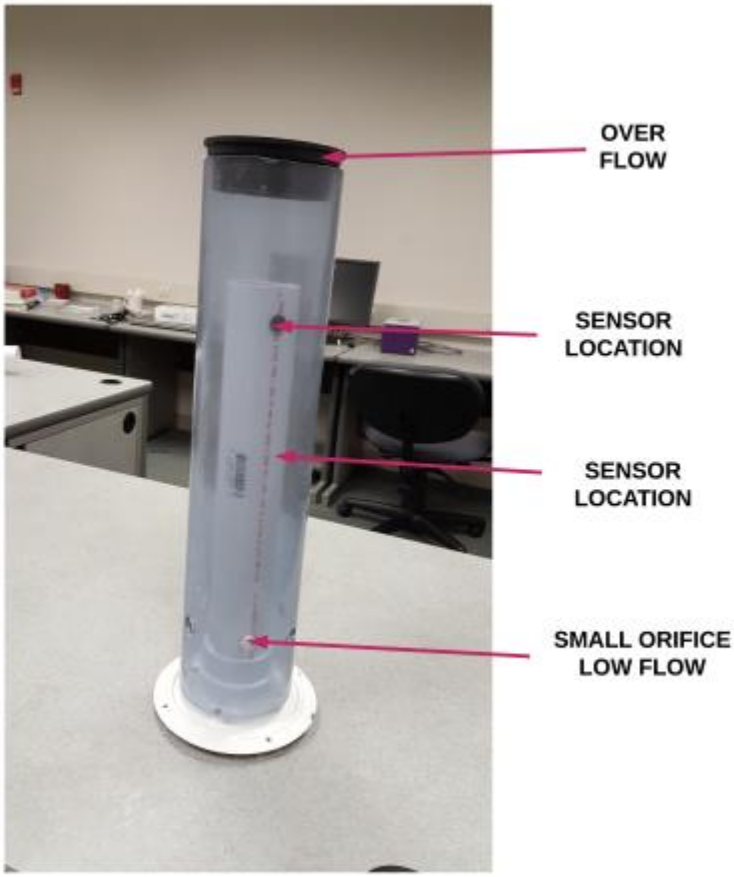
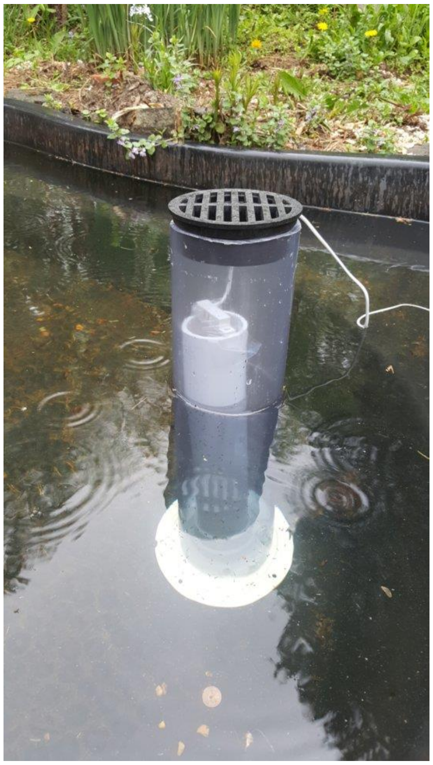
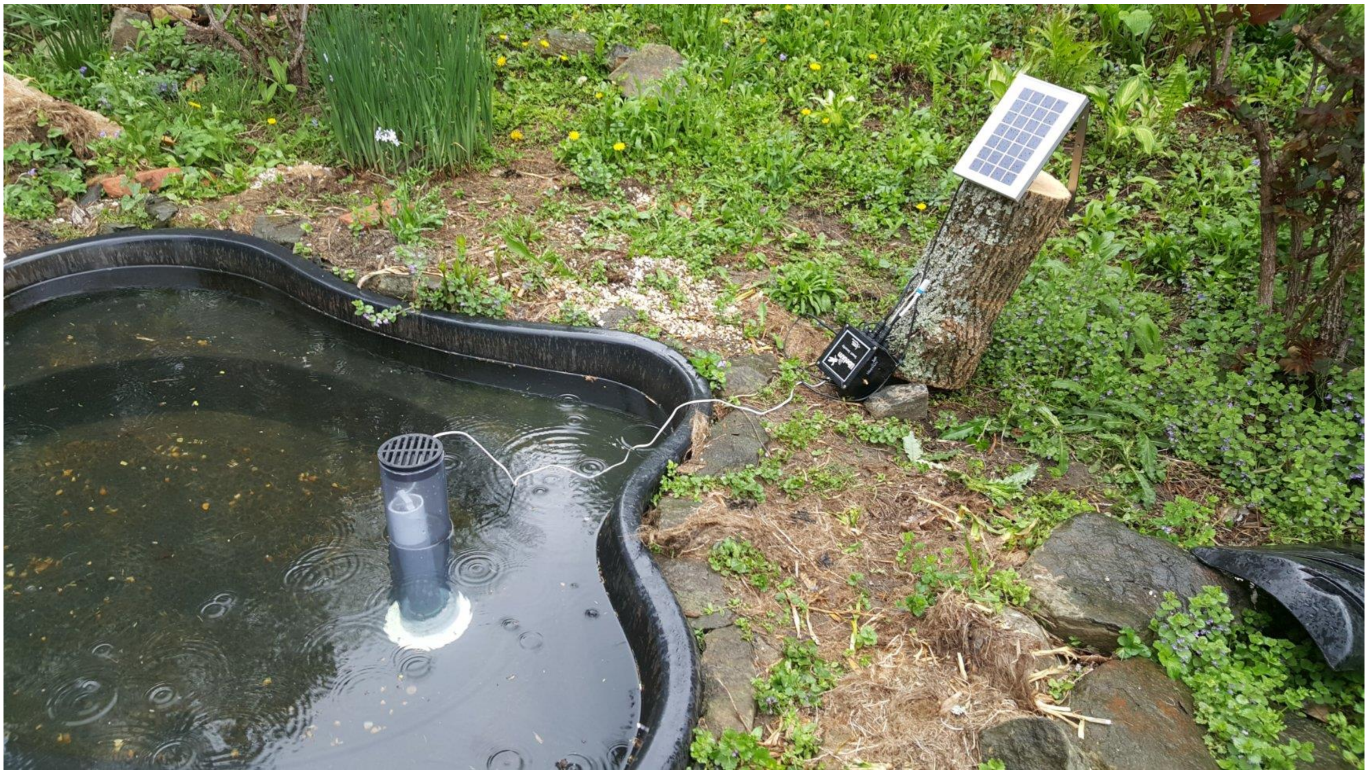


Figure 6.2 Site Ready Prototype

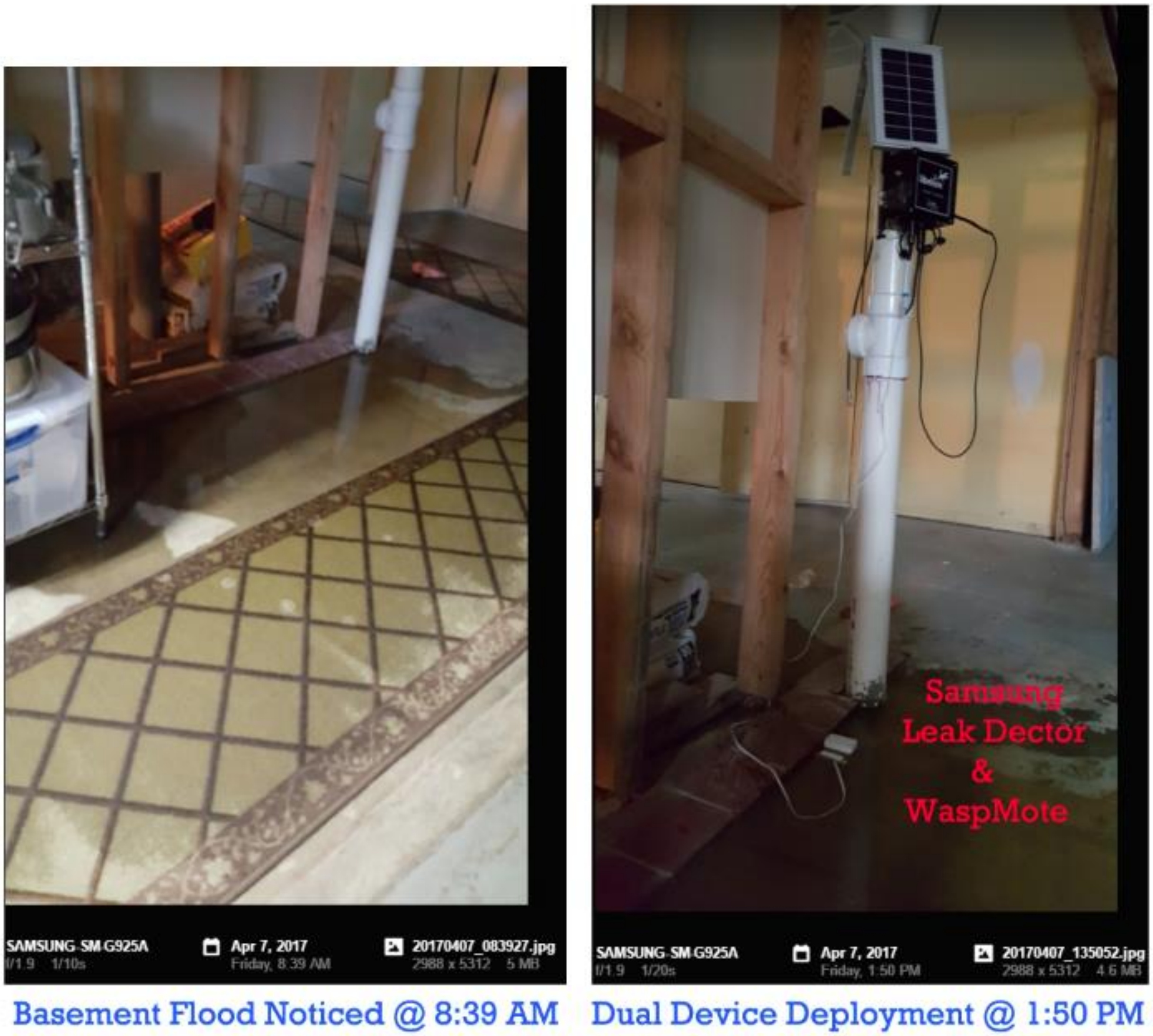
This device is a multi-level riser structure. Water enters the device through small holes in the bottom and when it reaches either of the sensors, it will trigger the sensors and rest of the process as discussed earlier will occur i.e., sending alerts etc. Once the water level rises above the structure it will leave the device through the top portion of the pipe.





Chapter VII: Device Validation & Comparisons

The major contribution for this research was to check on reliability of our system, to that effect we deployed this device in a real-world situation. During the recent storm of 04/06/2017, the basement of the researcher had flooded.

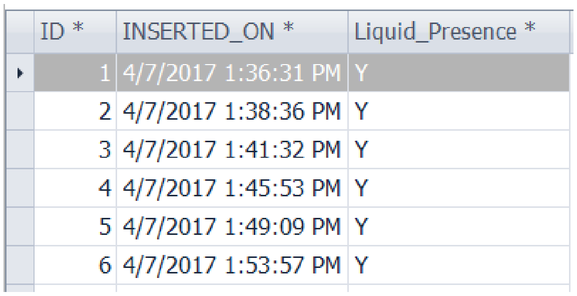


Basement Flood Noticed @ 8:39 AM

Dual Device Deployment @ 1:50 PM

Figure 7a Device Validation

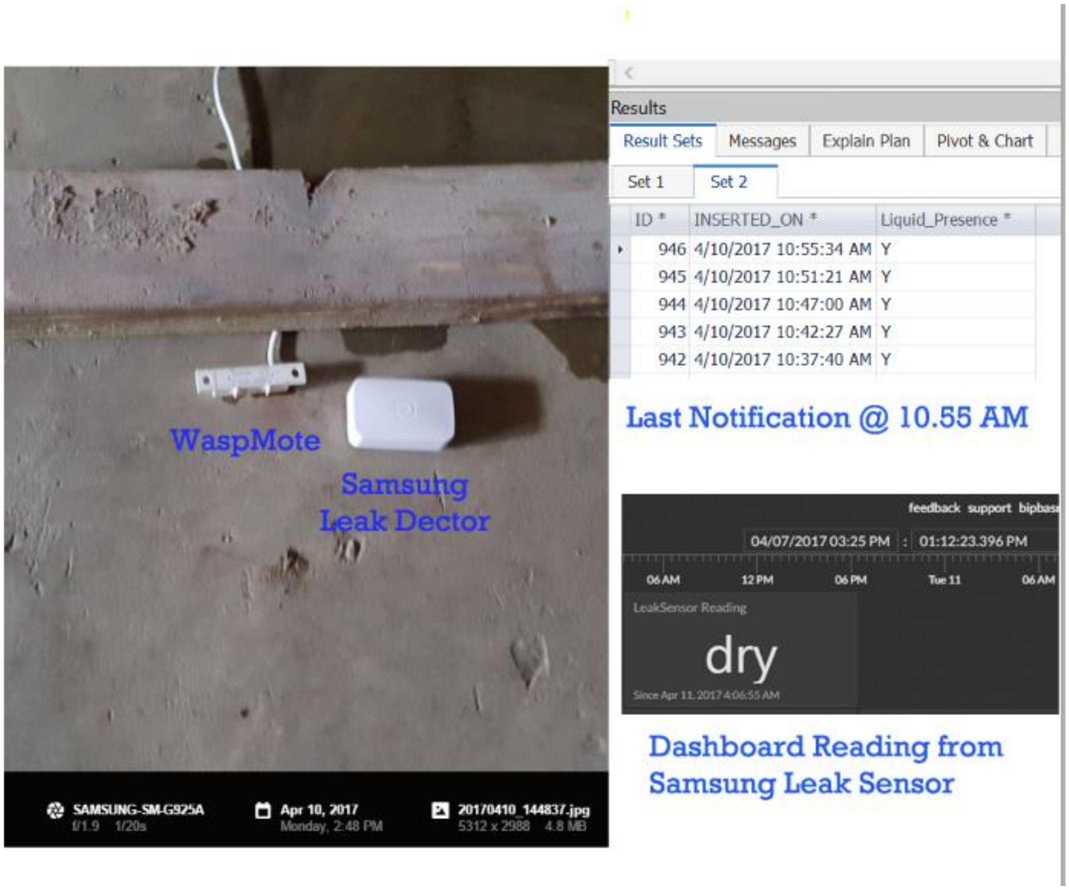
Thinking this as a perfect situation for the incident, the researcher decided to deploy another commercial product from Samsung called Samsung Leak Detector. The working principle of this device and the alternate device is the same, i.e. they work by capacitance circuit completion as described earlier. As exepected the device started trasnmitting the data through to our server. A screenshot with the time stamp ID# 5 is almost analogous to the picture.



ID *	INSERTED_ON *	Liquid_Presence *
1	4/7/2017 1:36:31 PM	Y
2	4/7/2017 1:38:36 PM	Y
3	4/7/2017 1:41:32 PM	Y
4	4/7/2017 1:45:53 PM	Y
5	4/7/2017 1:49:09 PM	Y
6	4/7/2017 1:53:57 PM	Y

Figure 7b Device Data Validation

Finally, the device stopped sending liquid presence sensor to the server at around 11:00 AM on 04/10/2017. All these time Samsung leak detector was in place and was validating the performance/reliability of our device.



ID *	INSERTED_ON *	Liquid_Presence *
946	4/10/2017 10:55:34 AM	Y
945	4/10/2017 10:51:21 AM	Y
944	4/10/2017 10:47:00 AM	Y
943	4/10/2017 10:42:27 AM	Y
942	4/10/2017 10:37:40 AM	Y

Last Notification @ 10.55 AM

Dashboard Reading from Samsung Leak Sensor

Figure 7c Device Validation

Based on the received data and performance, we could conlcude that the device perfomed sucesfully with the commericila device. A quick pro and con analysis is presented in the Table below about these two devices.

Parameter	**Waspmote**	**Samsung Leak Detector**	**Winner**	**Comments**
Function	Leak Detection	Leak Detection	N/A	Both device performed as expected.
Communication	3G	WiFi	Waspmote	Samsung relies on WiFi hub and hence has a limited range
Location	Any where	Indoor Only	Waspmote	Waspmote is designed for outdoor and more rugged environment
Cost	US$ 750 (Device + Sensor)	US$ 120 (Hub + Sensor)	Samsung	Waspmote is expensive compared to Samsung and other devices.
Ease of Use	Not User Friendly at all	Installation time less than 15 min	Samsung	Samsungs device connectivity and user interface are iniuative and easy to use
Third Party API Integration	Limited + Costly	Many + Open Source Code base	Samsung	Waspmote has a limited US market exposure where as Samsung was easy to connect to many smart home solutions including Goolge home.

Thus, it was a good exercise to see two devices hand in hand and make compariosion as explained here. The device gave us an additional confidence that it would work

Chapter VIII: Data Analysis & Research Findings

The main objective of this research is to validate the design and document our processes from end to end and hence we have not performed in-depth analysis of the collected data. Furthermore, the data collected so far has been from the prototype/simulation data hence its importance and relevancy is limited. Nevertheless, we present some of our data analysis and research finding based on the data we have collected so far.

We wanted to have a mechanism to control our emergency alert system and can predict the working condition of our device. To that effect, we captured the temperature reading from open weather API as described under Chapter 5 section 5b. We also captured the temperature reading from our sensor at the same time. In a period of around 48 hours we could match exactly 322 observations using timestamp as our epoch. In other words, there were 322 observations that were within microseconds. We thus created our full data set as given below on ***Table XX***. We have only included 20 rows for the brevity purpose. Full data set was included in our statistical analysis.

We derived one column with a differential temperature reading and created an arbitrary rule that we will **discard the data!** if the difference in temperature reading is more than **6 degree**. All the observations that had an absolute temperature great than 3 degrees were labeled as Bad data set. We then wanted to see if there is any correlation among other predictors and the temperature reading. Again, this is mostly done as an experiment and the result may not quantify the outcomes correctly. Our hypothesis on this data set is that since the device is reading temperature, other parameters such as cloud condition or humidity will have some effects on the data.

We used R statistical package to do our data analysis and perform prediction analysis to see if we can generate a model with acceptable confidence.

OBS#	SENSOR DATA		OPEN WEATHER (WEB API DATA)					DERIVED	ASSUMED CLAS
ROWID	SENSOR_TIMESTAMP	SENSOR READING	WEB TIMESTAMP	SUMMARY	WEB TEMPERATURE	WEB_ HUMIDITY	WEB_ PRESSURE	DELTA	DATA_ VALIDITY
1	9/13/2016 15:00	95.91	9/13/2016 15:00	Clear	87.46	0.5	1020	8.45	Bad
2	9/13/2016 15:00	95.91	9/13/2016 15:00	Partly Cloudy	85.81	0.5	1021	10.1	Bad
3	9/13/2016 16:30	95.97	9/13/2016 16:30	Partly Cloudy	86.07	0.51	1019	9.9	Bad
4	9/13/2016 16:30	95.97	9/13/2016 16:30	Clear	86.28	0.5	1020	9.69	Bad
5	9/13/2016 18:00	94.15	9/13/2016 18:00	Clear	83.53	0.55	1018	10.62	Bad
6	9/13/2016 18:00	94.15	9/13/2016 18:00	Clear	84.42	0.53	1019	9.73	Bad
7	9/13/2016 19:45	77.17	9/13/2016 19:45	Clear	77.85	0.65	1019	0.68	Good
8	9/13/2016 19:45	77.17	9/13/2016 19:45	Clear	78.5	0.63	1019	1.33	Good
9	9/13/2016 20:00	76.54	9/13/2016 20:00	Clear	77.5	0.65	1019	0.96	Good
10	9/13/2016 20:00	76.54	9/13/2016 20:00	Clear	78.05	0.64	1019	1.51	Good
11	9/13/2016 21:30	73.72	9/13/2016 21:30	Clear	73.5	0.73	1019	0.22	Good
12	9/13/2016 21:30	73.72	9/13/2016 21:30	Clear	74.02	0.71	1019	0.3	Good
13	9/13/2016 23:45	71.49	9/13/2016 23:45	Clear	72.43	0.76	1019	0.94	Good
14	9/13/2016 23:45	71.49	9/13/2016 23:45	Clear	72.63	0.76	1020	1.14	Good
15	9/14/2016 0:00	71.09	9/14/2016 0:00	Clear	72.29	0.76	1019	1.2	Good

i. Scatter Plot Matrix

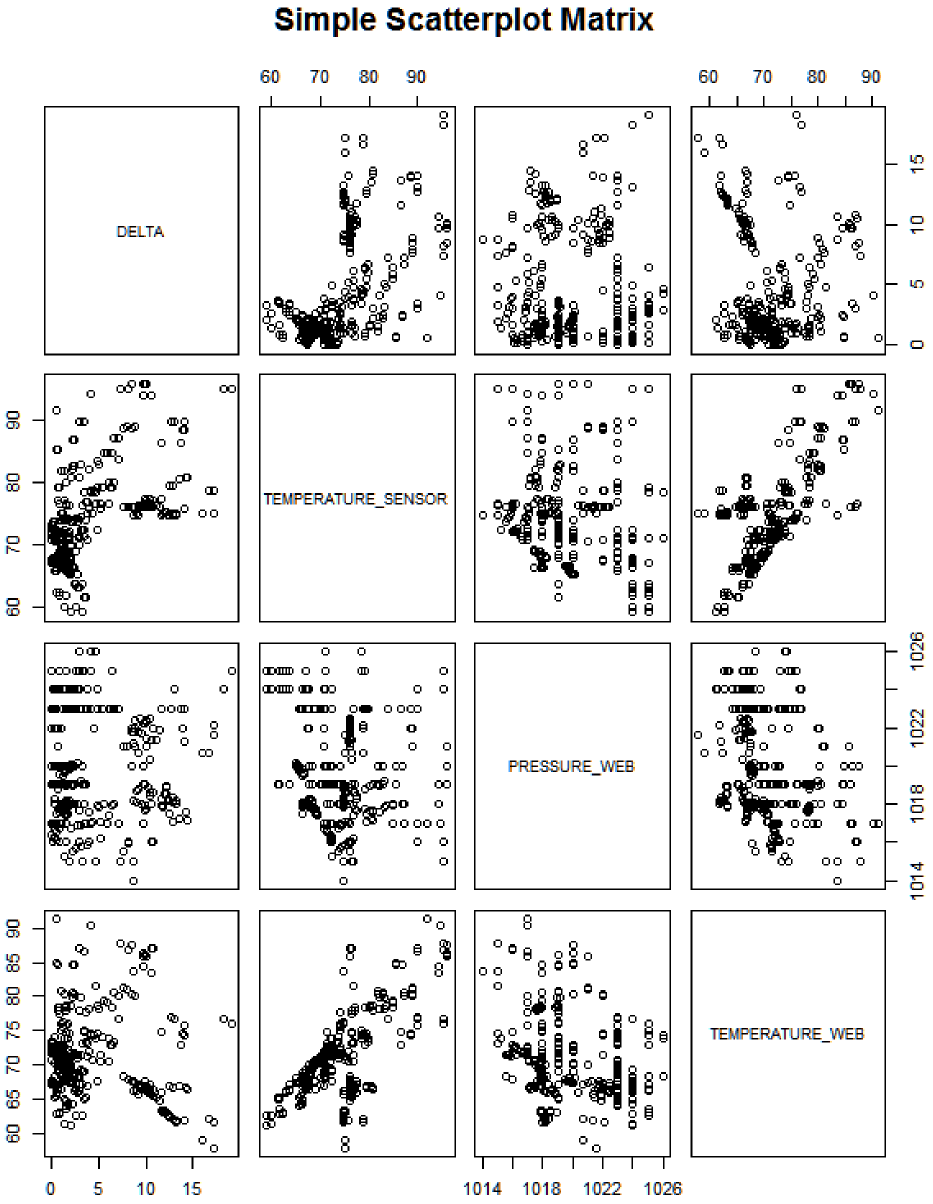


Figure 8a Scatterplot Matrix

We started our analysis by plotting the scatter matrix to see if there are any obvious correlation amongst the variables. As expected only obvious relationship is between the temperature_web and temperature_sensor variables. They seem to be linearly co-related and as one increases the other seemed to increases well. The other subtle relationship that’s observed here is the volume of delta decreases as the temperature increases. In other words, there are not many observations that fall into higher than delta value of 10.

ii. Effect of Weather Condition on Sensor Reading

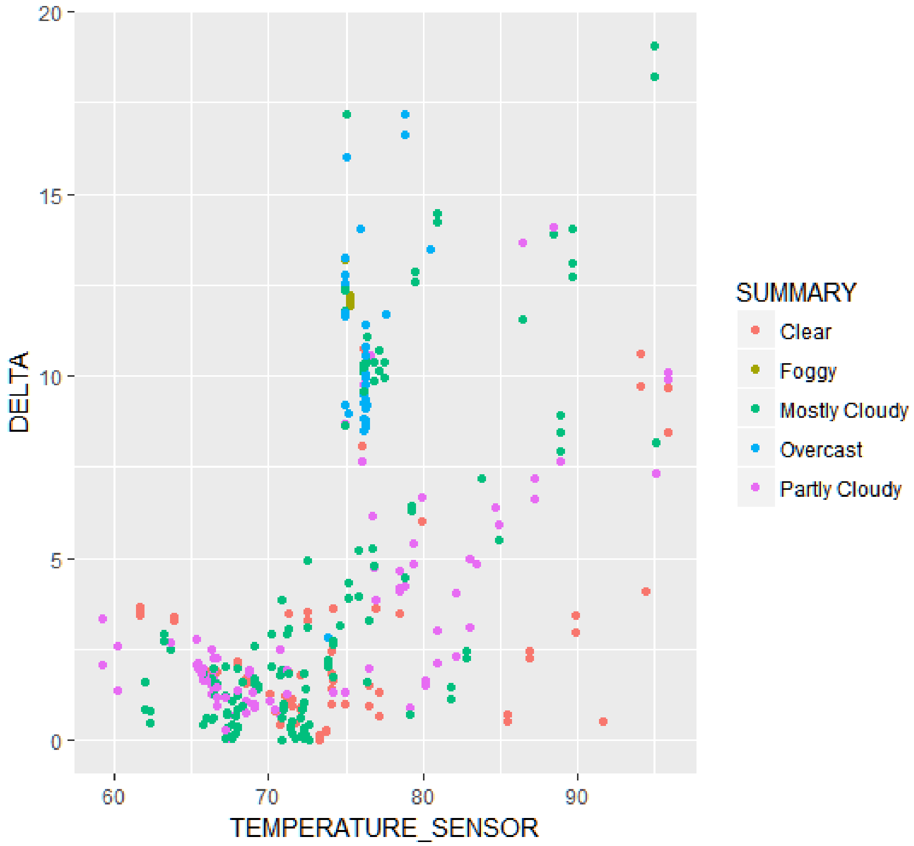


Figure 8b Sensor Vs Delta

We wanted to understand the effect of general weather on our temperature reading and hence and plotted the delta to see if there are pattern that can is associated with the weather condition. One observation that we see is our delta is mostly below 5. So, for the experimentation purpose we decided to set a threshold of **6.**

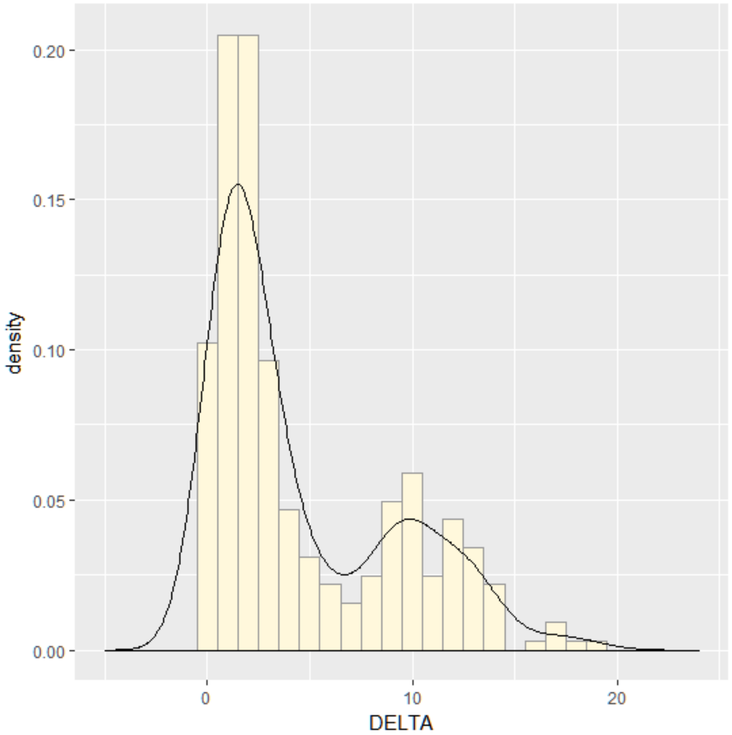


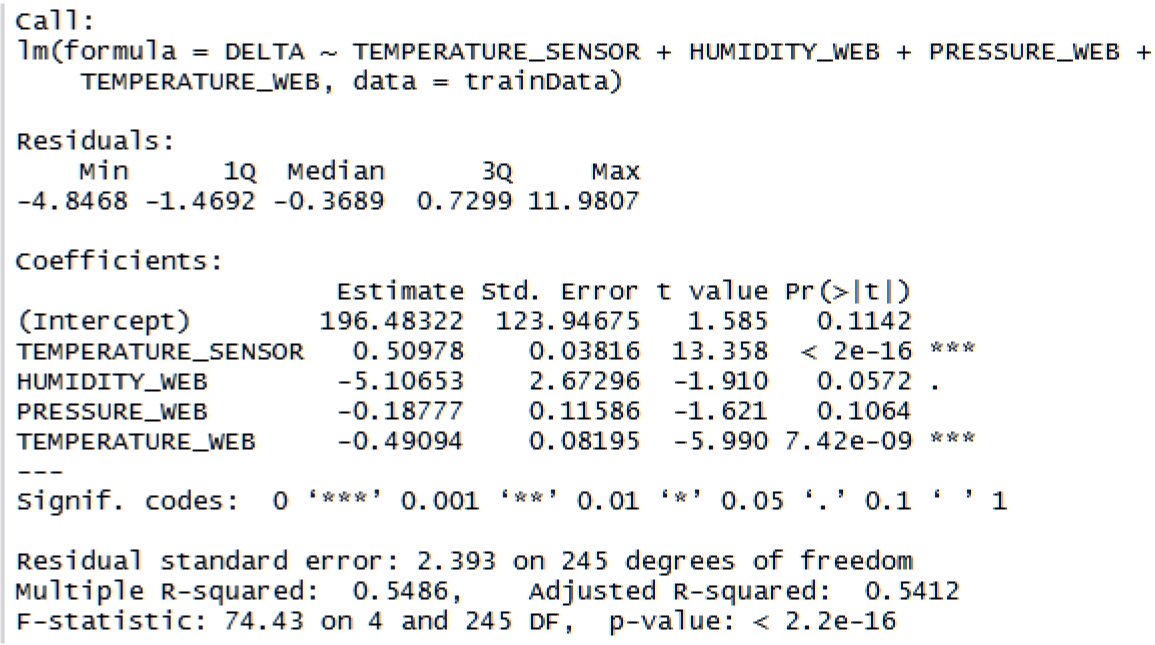
Figure 8b Sensor Vs Delta

We also plotted the histogram of delta. To ensure that the data separation (good and bad) lies within acceptable limits we plotted delta and its histogram. Again, majority of our data lies within the range and hence if delta higher than 6 as a bad reading is supported by our data distributions.

iii. Linear Regression and Prediction Model

We wanted to see if we can produce a linear regression model to predict the device reliability based on given parameters. In other word, try to understand under what condition will the delta be higher than 6 and hence we classify the recording as bad records. To do that we took our dataset and split them into to two sets (training and test). We took observation 1 through 300 as our training set and remaining as test set.

Passing all parameters to the model, we observe the output from **R** as below.



```
Call:
lm(formula = DELTA ~ TEMPERATURE_SENSOR + HUMIDITY_WEB + PRESSURE_WEB +
    TEMPERATURE_WEB, data = trainData)

Residuals:
    Min      1Q  Median      3Q     Max
-4.8468 -1.4692 -0.3689  0.7299 11.9807

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        196.48322  123.94675   1.585   0.1142
TEMPERATURE_SENSOR   0.50978    0.03816  13.358  < 2e-16 ***
HUMIDITY_WEB        -5.10653    2.67296  -1.910   0.0572 .
PRESSURE_WEB        -0.18777    0.11586  -1.621   0.1064
TEMPERATURE_WEB     -0.49094    0.08195  -5.990 7.42e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.393 on 245 degrees of freedom
Multiple R-squared:  0.5486,    Adjusted R-squared:  0.5412
F-statistic: 74.43 on 4 and 245 DF,  p-value: < 2.2e-16
```

Figure 8.3 Regression Model Output

It is noticeable from the **Delta** set that delta has a high significance with the two predictors TEMPERATURE_WEB and TEMPERATURE_SENSOR (p value is less than 0.005). This imperatively makes sense as DELA was computed using these two values.

However, the R-squared value and Adjusted R-Squared value are 0.5, which means the model performance is not that good. These numbers being close to 1 means model is 100% right and 0 is no correct prediction at all. Thus, linear model is not useful at all and can be discarded.

Chapter IX: Risk and Challenges

The main objective of this research was to document our learning and challenges faced with the technology. Here are few of the observations that we observed during our development.

Learning Curve

Reading at the web information from Libelium, it was not clear that there would be significant amount of software reading. We were under the assumption that we would receive the device, deploy it and start receiving our data. There was a lot of software coding and learning required before the device was ready to be used. This is a major challenge with our current device, as there would be a steep learning curve to future developer and a major hindrance to our goal of commercialization. The device is good for proof of concepts and prototype but we assume that researchers would want to spend more time in data analysis rather debugging the software/hardware code base.

Code Complexity and Rarity

The coding language required to interface the device is no a common software language such as JAVA, python or SQLs that anyone can easily pick up and start implementing them. Again, there would be challenges to find resources who can implement the device in mass scaling.

Module/Sensor Inferences

We noticed that when device has a code that is not hooked up with the sensor reading, we started to receive false notification. It means that we must go back and use the codes that are only pertinent to the sensors. There is no “use as required” mechanism to control the sensor probe reading.

Cost & Third-party Integration

Libelium has limited third party integration, specially since this is an European company, they seem to be more compatible with EU service providers. Limited US datahub integration but at a very high cost. The device itself if robust and rugged but costly. The current cost of device ($1000) per unit prohibits from using this in a multi-node setup.

US Network Challenges

We had a lot of issues in being able to transmit the data to our server. As of now we have one network called Ting that supports the device still not connected via Verizon, Sprint. Tried AT&T Machine to Machine (M2M) solutions that worked but again cost for data transfer form M2M is higher as compared to regular services.

Data Analytics Platform

The device does not provide any real-time data analytics platform and we have to do lot of manual coding to receive the data. A lot of IOT solutions in US Market now have built in data analytics which is a competitive loss for Libelium

Distance and Communication Lag

We had a hardware failure in our device (within the first 1 year) hence they agreed to replace the device for free. However, the device had to be mailed via FedEx and got stuck in US boarder and custom offices for week. By the time we sent and received our device back it took us almost 1 month in route. Thus, this is another major setback for us as Libelium does not have a dedicated US service center.

Chapter X: Conclusion and Future Works

This research was a great learning opportunity for me as a graduate student in UMBC IS department. I could bring in multidisciplinary knowledge from Civil Engineering background and mix it up with the Information Systems. The device and data collection mechanism are showing promises that we can use this for various CPS project and move toward creating our final Flashflood warning system. However, we are at a very tip of this problem and there are many research ideas that need to be further explored. Specially, the data analysis and machine learning algorithm are not generated at all.

We are in discussion with UMBC and Howard county where we can install the device and start collecting the real-time data. At that time we will look at building more robust machine learning algorithm based on our device and start validating the predictive model.

References

[1] "Smart City Profiles," 14 04 2017. [Online]. Available: http://www.pleecproject.eu. [Accessed 14 04 2017].

[2] "An overview of smart sustainable cities and the role of information and communication technologies," ITU-T Focus Group, 2014.

[3] "Wikipedia," Wikimedia Foundation, Inc., 13 4 2017. [Online]. Available: https://en.wikipedia.org/wiki/Internet_of_things. [Accessed 14 04 2017].

[4] K. A. D. S. P. J. W. J. S. C. B. R. W. J. M. M. J. M. D. F. I. P. F. F. R. a. V. R. B. Kevin A. Delin, "Environmental Studies with the Sensor Web: Principles and Practice," *Sensors,* vol. 5, pp. 103-117, 2005.

[5] "National Weather Service," US Department of Commerce, [Online]. Available: http://www.weather.gov/phi/FlashFloodingDefinition. [Accessed 14 04 2017].

[6] M. S. S. P. John M. Joyce, "An Assessment Of," Maryland Department of the Environment, Baltimore, 2005.

[7] W. E. P. Deborah J. Cappuccitti, "Final Deliverable U. S. Environmental Protection Agency Section 319(h) , Clean Water Act," MARYLAND DEPARTMENT OF THE ENVIRONMENT , Baltimore, 2000.

[8] H. L. L. W. O. L. Steven D. Glaser, "Sensor technology innovation for the advancement of structural health monitoring: a strategic program of US-China research for the next decade," *Smart Structures and Systems,* vol. 3, no. 2, pp. 221-224, 2007.

[9] T. W. J. H. Peter Corke, "Environmental Wireless Sensor Networks,Philip Valencia,Darren Moore," *Proceedings of the IEEE,* vol. 98, no. 11, pp. 1903-1916, 2010.

[10] "Libelium," [Online]. Available: http://www.libelium.com/libeliumworld/plug-sense/. [Accessed 20 March 2016].

[11] G. B. G. S. V. O. C. a. M. P. FRANC¸ OIS INGELREST, "SensorScope: Application-Specific Sensor Network for Environmental Monitoring," *EPFL—WSL/SLF.*

[12] A. M. P. A. C. Robert Szewczyk, "An Analysis of a Large Scale Habitat Monitoring Application," *SenSys'04,* 2004.

[13] Arduino, "Arduino," [Online]. Available: https://www.arduino.cc. [Accessed 15 04 2017].

[14] T. S. B. J. H. C. C. M. F. J. J. D. S. B. L. Bourikas, "Addressing the Challenge of Interpreting Microclimatic Weather Data Collected from Urban Sites," *Journal of Power and Energy Engineering,* vol. 1, pp. 7-15, 2013.

[15] A. T. S. S. C. G. R˘azvan Mus˘aloiu-E, "Life Under your Feet: A Wireless Soil Ecology Sensor Network," *Microsoft Research.*

[16] R. D. M. Jagannath Aryal, "Evaluation and Visualization of Hydrological Sensor Network: an Integrated Approach Using MODIS Images," *AGILE,* 2013.

[17] "A guide to selecting the right float switch for your application," Cynergy3, [Online]. Available: https://www.cynergy3.com/blog/how-select-correct-float-switch. [Accessed 15 04 2017].