Sensor Agnostic and Communication Independent SCADA

by

Gustavo Ariel Naigeboren

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Sensor Agnostic and Communication Independent SCADA

By

Gustavo Ariel Naigeboren

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

Master of Science

Abstract

SCADA solutions have typically been reserved for large organizations because of the high costs involved in customization, including the sensors, the communication channels and the specific implementation. In recent years with the evolution of a wide range of technologies there is an opportunity for a variety of both large and small organizations to utilize SCADA systems for monitoring and control. The development of new and affordable technologies in many areas such as computers, sensors and networks will help to achieve the goal of low cost and more universal SCADA systems. The objective of this thesis is to create prototype SCADA framework as independent as possible of the sensors or communication channels used, keeping in mind affordability for small or medium size organizations. Through the process of developing SCADA systems across various application domains, experience is gained and recommendations generated toward developing a universal SCADA framework that will be largely independent of sensors and underlying communication technologies.

Acknowledgements

I want to thank every person who helped me with ideas, suggestions and support to complete this research work. One of the most important things I learned through the last four years of study is that it is impossible to do a thesis like this one without the help of a lot of people.

I would like to start to thank Dr. Robert McLeod for his continued support, great ideas and the way he encouraged me to achieve my goals. Also, I would like to thank Dr. Marcia Friesen for her great spirit, knowledge and for helping me to be a better person.

The sandbag project could not be a reality without the unconditional help and support of Dr. James Blatz. He gave me the freedom to design the best possible solution to monitor sandbag dikes. I want also to thank Brian Oleson for letting me use his house to test the system. Moreover, I want to thank him not only for having the patience to be there every morning at 7.30 am but also for all the coffee and breakfasts he made for me.

The idea to create a system to track patients in the nursing facilities came up after a meeting I had with Sandra Delorme at Saul and Claribel Simkin Centre. She was the first one who introduced me to the world of RTLS. There are a lot of people that helped me to understand the need and the possible solutions. I will also like to thank several people from WRHA - Winnipeg Regional Health Authority - who trusted me and helped me to understand how to create a solution for their needs.

There is one special thank you that I want to make to my past employers: Infomagnetics Technologies and Conviron for supporting me during my studies in the Master's program.

Also, I want to thanks my friends for their unconditional help, encouragement and support.

Finally, I want to thanks my wife and kids for their patience and encouragement they have with me.

Table of Contents

Abstract ii
Acknowledgements iii
Table of Contents v
List of Tables x
Chapter 1: Introduction 1
1.1 Purpose 1
1.2 Scope 4
Chapter 2: SCADA Standard Practices 6
2.1 Definition 6
2.2 Introduction 7
2.3 System Functionality 8
2.4 System Configurations 9
2.5 System Features 11
2.6 Additional System Definitions 14
2.6.1 Communication Management [3] 14
2.6.2 Data acquisition 15
2.6.3 Data types 15
2.6.4 Supervisory Control Characteristics 16
2.7 Communication 16
2.7.1 Master Station Communication 16
2.7.2 Communication Channels 17
2.7.3 RTUs Communication 18
2.8 SCADA Protocols 18
2.9 Environment [3] 20
2.10 Reliability 22
2.11 Availability 22
2.12 System Security 23
2.12.1 Redundancy 24
2.12.2 Action Completed 24

2.12.3 Data Encryption or Intrusion Detection 24
2.13 Conclusions 25
Chapter 3: The Development of a SCADA for Monitoring Sandbag Dikes 27
3.1 Introduction 27
3.2 Sandbag SCADA Idea 28
3.3 Sandbag Monitor Version 2 29
3.3.1 Accelerometer 31
3.3.2 Microcontroller and Firmware 32
3.3.3 Radio Modems and Antennas 35
3.3.4 Enclosures 39
3.3.5 GPS 43
3.3.6 Batteries 43
3.3.7 Battery Charger 44
3.3.8 Data and Power Cables 44
3.3.9 Computer Software 45
3.3.9.1 Database 46
3.3.9.2 Communication Engine (CE) 46
3.3.9.3 GUI 48
3.4 Building the system 53
3.5 System Data Results Analysis 55
3.6 Recommendations for future versions 60
3.7 Conclusions 60
Chapter 4: The development of a SCADA system for an RTLS application 63
4.1 Introduction 63
4.2 RTLS 65
4.3 Principles of Passive RFID technology 66
4.4 RFID Passive Reader Selected 67
4.5 RTLS Architecture 70
4.5.1 RFID Passive Reader Installation 71
4.5.2 Communication Channels 71
4.5.3 Software Application 72

4.6 Passive RFID sensors challenges 73
4.7 Recommendations for Future versions 75
4.8 Conclusions 75
Chapter 5: Potential evolution: SCADA for an active RFID RTLS application 77
5.1 Introduction 77
5.2 Active RFID 77
5.3 Combining the sandbag monitoring system and the RTLS to create a new SCADA RTLS solution 78
5.4 Conclusions 79
Chapter 6: Conclusions and Future work 80
6.1 Project Conclusions 80
6.2 Future Work 84
Appendix A: Database Structure for Sandbag Dike Project 86
A1.1 Database Structure 86
Glossary 87
References 88

List of Figures

Figure 2.1: Typical SCADA system 8

Figure 2.2: Example of redundant and distribute or centralized SCADA 10

Figure 2.3: Proposed SCADA framework 25

Figure 3.1: Overview of the sensors installation 27

Figure 3.2 SCADA system design to monitor data from sandbag dikes wirelessly 30

Figure 3.3: Accelerometer Sensitivity at 25°C, Vs=3V [14] 32

Figure 3.4: Microprocessor flowchart 34

Figure 3.5: Convention utilized by the microcontroller to send data to the central station 34

Figure 3.6: Microcontroller with evaluation board and sensors mounted in the enclosure 35

Figure 3.7: System design overview 36

Figure 3.8: Enclosures: On the left enclosure for the microcontroller and sensors. On the Right enclosure for the RF circuits and antenna 39

Figure 3.9: Cable Gland: Two different connector sizes to install cables in the enclosures 40

Figure 3.10: Enclosure submerged in water to run water proof test with latex silicone 40

Figure 3.11: Enclosure sealed with electronics inside ready to be installed inside the sandbag .41

Figure 3.12: Enclosure built to protect the battery and to hold the external temperature sensor 42

Figure 3.13: Cable diagrams for the microcontroller and sensors and for the central computer system 44

Figure 3.14: Communication Engine Flowchart 48

Figure 3.15: RealTimeData module 49

Figure 3.16: RetrieveData module using the records parameter 50

Figure 3.17: RetrieveData module using dates as a parameter 50

Figure 3.18: DataPlot module Type 1: Accelerometer position in X, Y and Z 51

Figure 3.19: DataPlot module Type 2: Battery Voltage 52

Figure 3.20: DataPlot module Type 3: Internal and External Temperature 52

Figure 3.21: Accelerometer X motion 56

Figure 3.22: Accelerometer Y motion 56

Figure 3.23: Accelerometer Z motion 57

Figure 3.24: Additional support bar attached to the sensor to increase sensor sensibility 58

Figure 3.25: Battery discharge curves 59

Figure 4.1: RTLS system overview 63

Figure 4.2: RFID Reader and Tag operation 67

Figure 4.3: Passive RFID sensor used to create the RTLS 68

Figure 4.4: Reader mounted on a plastic pipe structure 69

Figure 4.5: Passive RFID card with 10 bytes of memory 70

Figure 4.6: RTLS Architecture 70

Figure 4.7: RTLS Interruption Mode 71

Figure 4.8: GUI interface for the RTLS 73

Figure 5.1: RTLS solutions combining SCADA projects from chapter 3 and 4 78

Figure 6.1: Proposed generic SCADA architecture for a sensor agnostic system independent of the communication channel 80

Figure 6.2: Polling and Interruption Modes 83

Figure A1.1 Sandbag database structure 86

List of Tables

Table 2.1: Typical availability values and annual downtimes 18

Table 4.1: Comparison between RTLS technologies 54

Chapter 1: Introduction

1.1 Purpose

SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to collect data often remotely, log, and process it and to take action if needed. As the name indicates, it is not a full control system, but rather focuses on the supervisory level [1]. Usually, a SCADA system is composed at least one computer system with a communication interface that connects the system to the remote unit to gather data or to control it. A simple example could be a weather station installed outside a building. A computer is connected to the external system to collect data. If the temperature drops below a threshold, the computer will turn the heating system on. This system described above can be easily replaced by a temperature control system. The point is that there is typically some measurement, remote control and actuation that is computer controlled.

There is an analogy between SCADA systems and philosophy. There are many definitions of philosophy. The one used by Aristotle is that the philosophy is all, existing knowledge. After a particular field gets specialization, it will create a new thread such as theology or math [2]. This idea also applies to SCADA systems. SCADA is the generic term used to describe systems with remote sensors and logging capabilities. For example, an Alarm System is a SCADA system but for an specific purpose. An alarm has sensors to collect data. If an intruder breaks into a building and the system is active, it will alert the police and trigger an alarm. In other words, the system collects the data and based on some state it will take action. This action can also be manual. When a system gets to certain level of specialization, it will use a name to indentify it for that specific purpose. Other examples could be smoke or fire detection systems or HVAC - Heating Ventilating Air Conditioning - control systems.

SCADA has been typically used only by large organizations or manufacturers to control complex systems and processes. Recently, there has been a significant need and opportunity to monitor and control smaller systems. Two such systems discussed here are a sandbag monitoring system and a RTLS - Real Time Location System - to locate people or equipment indoors.

High end SCADA systems are not a suitable fit for small applications. Over the last ten years the author has been developing SCADA solutions for different kinds of companies such as a utility company, an environment control company and for the Department of National Defence (Canada). The need to create a common architecture to offer SCADA solutions at a lower cost was evident, independent of the sensors or the communication channels used to connect the central station with the RTUs - Remote Terminal Unit - as well as the operator or user.

One of the difficulties noted was the lack of a standards track for developing SCADA applications. Although an official IEEE SCADA standard was evolving it was not widely used in practice as the majority of SCADA systems were built "one-offs". The high degree of specialization was a consequence of the application itself as well as the myriad of sensors across various SCADA systems.

The intention of the present research project is to gain understanding of the potential to create a SCADA framework that is sensor and communication protocol agnostic. Specifically, the goal of the investigation is to see if it is possible to create a reusable architecture to create a SCADA framework independent of the sensors and communication channels to the largest degree possible. All SCADA systems have some degree of customization by definition. The goal here is to minimize that level of customization on behalf of the application developer, thereby, facilitating reuse and redeployment of the basic SCADA framework.

SCADA has not evolved in the same way that internet applications have. SCADA does not have an open architecture like modern software applications. This project will present alternatives to make the software part of SCADA agnostic of the sensors and protocols used by the sensors. There is no reason why a software engineer (SCADA application developer) should know the details of SCADA that are not related to the field of interest. As an analogy, the internet application developer does not really need to know the details of the transport layer protocols to develop web applications. The work done suggests an alternative architecture that could evolve to become a simple solution for many SCADA applications with emphasis on ease of implementation and cost-effectiveness.

The data collected by SCADA often has a lot of uncertainty and is statistical in nature. As a consequence, additional analysis may be required. Although, it is not part of a SCADA framework to provide analysis results, the architecture suggested here will facilitate the integration of these types of modules.

In order to achieve the thesis goals, two different applications were prototyped although with the view that the thesis concerns the systems and not the specific applications. Chapter 3 is a SCADA solution to monitor sandbag dikes and Chapter 4 is a SCADA solution for an RTLS - Real Time Location System - application. Chapter 5 presents a potential combination of the SCADA solutions presented in chapters 3 and 4 as another SCADA solution reusing what was created previously. Chapter 6 contains the final conclusions and future work developing a more universal SCADA framework.

Most SCADA systems, historically and now, have been non-standardized, "one-off", customized systems for specific applications. This reality is partly due to the non-uniformity of applications,

the fact that many SCADA systems are connecting to legacy systems, and that the components are numerous and diverse. The SCADA systems that are prototyped in this work can shed light on what should be kept and integrated into SCADA standards or a standardized SCADA architecture.

This work derives its value from three components. First, SCADA systems are evolving and this work provides insights into a potential evolving standard for SCADA systems, including insights into which parts are best to standardize. Second, it demonstrates that SCADA systems can add value by, for example, connecting to statistical engines that add a measure of intelligence to the system. Third, the SCADA systems then become highly suitable to applications where data has uncertainty or stochastic behaviour associated with it. Many of the emerging applications for SCADA systems, such as environmental monitoring, have this characteristic.

This thesis sheds light on which SCADA technologies and modalities will survive, and is a step towards demonstrating what a universal SCADA framework would encompass.

1.2 Scope

This thesis is divided in five different sections:

Chapter One: This chapter presents an introduction to the SCADA technology. Also, it details the objectives and goals of the research work.

Chapter Two: This chapter presents the best practices to develop SCADA systems. This chapter has two main sources. The first is the IEEE SCADA standard. The second source is my personal experience with more than 10 years developing SCADA solutions.

Chapter Three: This chapter presents a real implementation of a SCADA system used to monitor sandbag dikes for potential structural damage or movement during a flood.

Chapter Four: This chapter presents a real implementation of SCADA system to create a RTLS to monitor people, equipment or inventory.

Chapter Five: This chapter presents a potential evolution/combination of the SCADA projects presented in chapters three and four.

Chapter Six: This chapter presents the final conclusions and future work.

Chapter 2: SCADA Standard Practices

2.1 Definition

SCADA is an acronym for Supervisory Control and Data Acquisition. The main objective of SCADA is to do remote monitoring and/or control of remote devices over a communication channel and present the data to the user in friendly manner. As the name indicates, it is not a full control system, but rather focuses on the supervisory level [2]. Another possible feature of SCADA is the capability to not only acquire data but also to record it for future analysis. Furthermore, SCADA systems can also log information about the status of the remote equipment [3].

A SCADA system is composed of at least one central computer system called a master station (MS) that sends or receives the requests to one or more remote stations to get data (acquisition) or to execute an action (control). The term to identify the user interface is usually called HMI (Human Machine Interface), but in this work it is referred to GUI (Graphic User Interface) to promote SCADA as a software application rather than part of the hardware solution. Another component of SCADA is one or more field data interface devices - usually RTUs (Remote Terminal Unit) or PLCs (Programmable Logic Controller)- to interface with the monitoring devices and control systems. The last element is the communication channel used to transfer data between the remote device and the master station [4].

Much of this chapter is an adaptation of the SCADA IEEE standard (used with permission from IEEE std. C37.1-1994 (Superseded) IEEE Standard Definition, Specification and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control. Copyright 1994, by IEEE. IEEE disclaims any responsibility or liability resulting from the placement and

use in the described maner), notes, articles and personal experience building SCADA solutions over the years.

2.2 Introduction

Actual SCADA systems are generally limited to large organizations. One of the main reasons is the excessive costs of SCADA solutions limiting its use to a small group of companies. The present chapter is based on the IEEE SCADA standard which is demonstrably complex, and as such has been simplified here updating the technologies used. The range of current SCADA systems is complex and disorganized. This work sheds light on how a standardized SCADA architecture could evolve.

There are many different implementations of SCADA from different vendors. There is a need to create a unique SCADA framework to reduce costs and to create a unique platform that is agnostic of the sensors used or the remote devices controlled. There is also a need to have SCADA frameworks that handle multiple protocols to communicate with the remote devices.

2.3 System Functionality

The SCADA system is composed of three basic elements as shown in Figure 2.1.

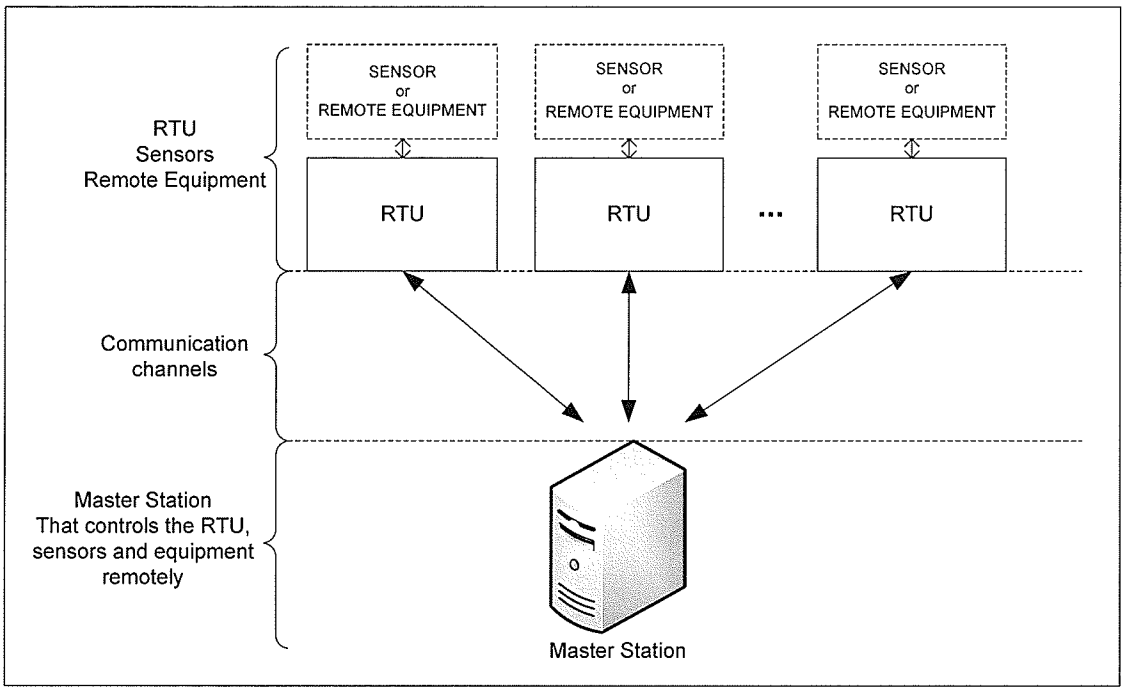


Figure 2.1: Typical SCADA system

There is no unique configuration for a SCADA system. However, a typical system will have at least one master station (MS) and at least one remote terminal unit (RTU). Usually, the protocol used requires that the MS initializes the communication with the RTU. The MS is the entity responsible for polling all the RTUs for updates of status and sensor information (if a polling protocol is used) [3].

There are two very distinctive modes to get data from the RTUs using SCADA. The first type is where the master station polls the RTUs to request data from the sensors. The second type is where the master station received that data from the sensors without requests (interrupt mode). In Chapter 3 an example of how to build an SCADA system to monitor a sandbag dike for potential

structural damage is prototyped using the polling technique. Chapter 4 presents another example of how to build an SCADA for RTLS is prototyped using the interrupt mode. It is also possible to utilize a combination of both systems. There is a big challenge to use interrupt mode using wireless technology because of collisions in the transmission process. This could delay an alarm to be sent to a master station. I had a challenge in a system deployed for a hydro company in Argentina - with over 200 sensors - where I had situations where the remote devices were experiencing significant delays getting access to the wireless channel while attempting send data back to the master station. The network used was a low speed connection, however this problem would be present on a high speed network if sufficient sensors or significant amounts of data were being collected.

The communication channel is the media that connects the remote equipment or RTU with the master station. There are two main groups: wired and wireless. The wired technologies are more commonly used in areas where the element to be monitored or controlled is close to the master station. This solution is also used in fields like telecommunications where all the nodes are connected with an existing network. The second alternative - wireless - has grown significantly in recent years. One example is shown in Chapter 3 where the sensors in the sandbag dike are connected using a wireless mesh network. Another possibility is to use the cell phone companies as a transport solution for the data.

2.4 System Configurations

The master station can be composed of one or more computer systems. The computer systems can be centralized or distributed over different locations (Figure 2.2). Also, the system can have multiple communication modules.

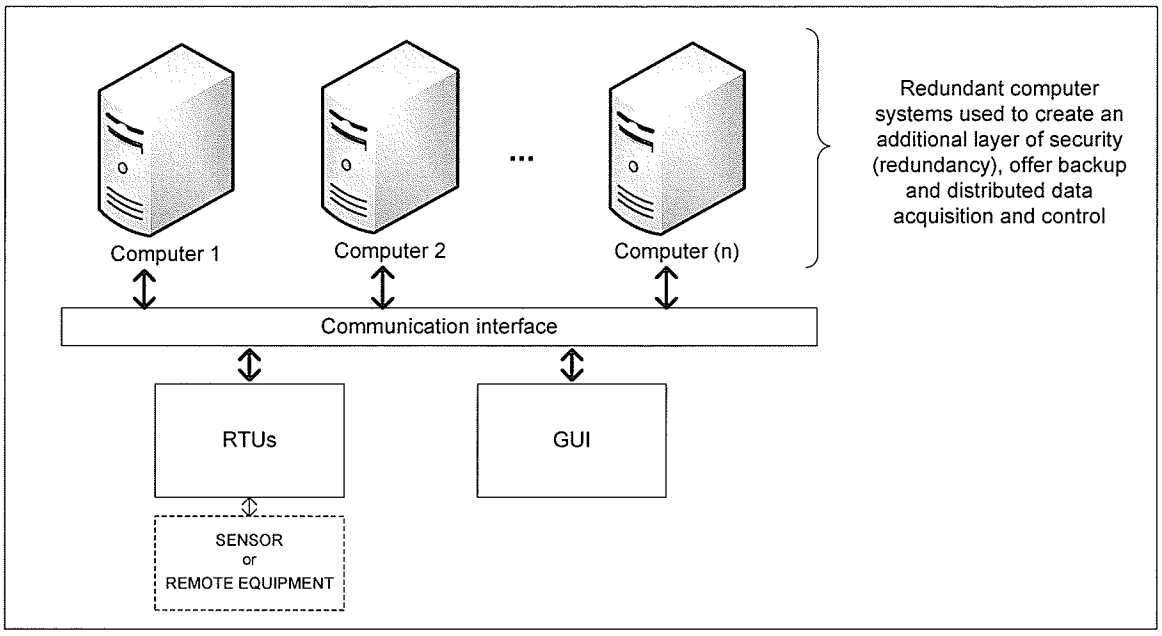


Figure 2.2: Example of redundant and distributed or centralized SCADA

There are many reasons to install a distributed computer system or multiple communication modules (different communication links between nodes). Some of the reasons are:

a) System redundancy: in case of failure there is another system as a backup.

b) Multiple plant building: a plant can have multiple buildings with one SCADA per building.

c) Different tasks: one computer system can be acquiring the data while other is recording the information into a database and retrieving it and presenting it to the end user.

d) Communication channel saturation: because of physical limitations on the communication channel there might be a need to have multiple master communication devices to contact the RTU. For example, if a system is using a wireless communication channel there is a limit of how many RTUs can be installed to guarantee them access to the MS in time to report data or alarms. If a channel is saturated, some RTUs might not find a timeframe to send data back to a MS.

2.5 System Features

The SCADA system can have multiple features or modules. SCADA systems differ from each other based on requirements. The following is a list with the most important components to consider in a SCADA system.

a) User Interface

The user interface or HMI (Human Machine Interface) or GUI (Graphic User Interface) is the interface between the system and the user. It has to be user-friendly and intuitive. The critical indicators have to be visible. Alarms and status indicators have to be clearly displayed on screen.

The goal of the user interface is obvious. However, there have been accidents caused by poorly designed systems or missing alarms. On February 14, 1982 the drill rig Ocean Ranger sunk into North Atlantic waters because operators could not understand the extent of a problem with the platform due to a poorly designed GUI. Also, the lack of electronic alarms failed to notify the crew of a potential problem. Another example is the accident at Three Mile Island on March 28th, 1979. The nuclear plant was only thirty minutes away from the meltdown. What saved the plant was an experienced operator who was called at 4.00am. The system was not capable of notifying the operator on duty of the problem [5].

b) Data Processing

The SCADA system's principal data processing task is to acquire data from a RTU. Then the system can optionally process the data before showing it to the user or storing it into a database.

c) Database Maintenance

The database is a key component of the SCADA systems capable of recording information for future analysis or reference. The database has to be fully functional and properly maintained to avoid critical failures. A typical problem of SCADA systems is database growth, filling up the storage device making the whole system crash or become inoperable.

d) Control Processing

The control processing is another optional feature for SCADA systems. For example, a SCADA can be logging data from a remote sensor and can take action in case of an alarm, major failure or just control a process. The 'control' side of the SCADA can be automatic or manual assisted by an operator.

e) System Backup

Usually, a SCADA system requires a very intensive configuration to function properly. Also, the historical data retrieved by the system is stored in a database. Special care is required at the time of the backups not only to restore a system in case of failure but also because some of the logged data may have to be kept following regulatory rules.

f) System Redundancy

SCADA systems are sometimes used in very critical situations. They can be as simple as taking temperature values from an outdoor sensor, or they can be responsible to manage the whole electrical distribution of a region. System redundancy is sometimes required. The systems can be located in different buildings in some cases. Also, there are cases where the RTUs are duplicated to have a backup in case of a failure of the main RTU device. Furthermore, there are cases where the communication channels are also duplicated in case of malfunction.

g) Self Diagnostics

The SCADA system is a compound of many subsystems. A self-test diagnosis is required to understand the functionally of each sub component. The system can be created in a way that the central system not only gathers valid data from sensor but also critical system information such as RTU internal temperature, external meteorological conditions or communication channel status.

h) Communication Interfaces

a. Communication with another computer system

Since SCADA systems can be distributed, it is critical to have a reliable system to interconnect the main computer systems. An example of this system could be a regional system that controls energy distribution. The networks can be LAN or WAN, and some combination of public and private.

Also, there are some SCADA configurations where many computers are needed. For example, you can have a system to communicate with the RTU, other one to record data into a database and a third one to present it to the user. The interconnection between the systems is usually done using a reliable LAN or WAN network.

b. Communication with a RTU device

The communication between the main computer system and the RTU can be done using LAN or WAN networks, serial connections, specific manufacturer's technologies, wireless, standard or proprietary.

i) Analog Inputs (usually connected to the RTUs)

Including but not limited to transducers and sensors.

j) Analog Outputs (usually connected to the RTUs)

Including but not limited to controllers, recorders and meters.

k) Digital Inputs (usually connected to the RTUs)

Including but not limited to pulse inputs, breakers, switches and relays.

l) Digital Outputs (usually connected to the RTUs)

Including but not limited to breakers, switches, generators, other devices.

2.6 Additional System Definitions

There are several elements that have been defined for a SCADA system, most of which are self explanatory.

2.6.1 Communication Management [3]

a) Message protocol

b) Number of communication channels

c) Channel considerations

d) Error detection techniques

e) Channel switching

f) Number of RTU per channel and/or channels per RTU

g) Number of retry attempts

h) Time out value(s) by message type

i) Communication error reporting, failure, criteria, and recovery

j) Channel quality monitoring (normal and backup)

k) Channel diagnosis/test provisions

l) Equipment interfaces

m) Report-by-exception of point scan

2.6.2 Data acquisition

The definition for each possible type has to be defined. All the possible ranges for data input and output, scale factors, rates, and accuracy also have to be defined [3].

Each RTU has to have a definition of its capacity. Typical elements are the number of inputs, number of outputs and data rate. The MS also has to have definitions to interface with the RTU properly such as data exchange rate with the RTU or communication media.

The RTUs can process data before sending it to the central system. If that is the case, exceptions have to be implemented to alert the MS of possible errors.

2.6.3 Data types

There are different kinds of data that can be retrieved from the RTU. The following list describes the most important types and the considerations required in each case.

a) Analog data: It is a very important to define where the analog data will be processed. The information about filters, amplifiers, single or multiple data reading processes has to be considered.

   The analog data sent to the controller can be one of the following:

   - Accumulated data
   - Computed data
   - Alarm data

b) Status Data: The information contained in a digital signal is based on the discrete states of the signal such as presence or absence of a voltage, current or contact.

2.6.4 Supervisory Control Characteristics

The control capability of an SCADA system is optional. It is possible to have a SCADA system with monitoring capabilities only. However, if control features are implemented, the following considerations have to be taken into account.

a) Clear definition of the control sequence
b) Feedback to confirm the request was completed
c) Duration of the output value and frequency of occurrence
d) Verify control action with previous established set points
e) System auto-pilot capabilities to avoid major failure or disasters
f) Mechanical and electrical interfaces of the RTU

2.7 Communication

The communication goal is to transfer the data acquired at the RTU into the master station and at the same time to execute actions in the RTU following the master station instructions. There are many components involved in the communication process. There are also different possible topologies.

The most common schema is where a computer acts like the master station. Based on this design, the master station manages the communication protocol. The MS has at least one communication channel that connects the whole system with the RTUs. Finally, the RTU has a communication module to handle the communication with the master station.

2.7.1 Master Station Communication

The master station usually has the communication module built in. However, it can be an external component. For example, if RTUs are installed over a LAN network over Ethernet, the

network adapter could be built-in inside the computer that handles the central system. On the other hand, if the system is using wireless serial RS-232 communication an additional communication interface may have to be installed outside the computer system.

A good practice is to create a physical isolation layer between the MS and the computer system to protect it from electrical shocks or lighting.

The communication interface can handle the communication protocol by itself or it can be done from the software controlling it from the master station.

Each system can have one or more master stations. The reason to have more than one master station is to avoid channel saturation - dividing the communications between multiple destinations - or to provide redundancy.

2.7.2 Communication Channels

Each master station has at least one communication channel with the RTUs. The communication channels can be wired or wireless.

The communication channels have some unique characteristics such as maximum distance and speed. The size of the protocol messages and the speed of the channel will determine the channel performance. The load of each channel has to be analyzed and documented. The worst case scenario analysis is recommended. The load factor per channel should be no more than 0.75. This is a standard procedure to allow retransmissions and data control overhead [3].

The channel data rate and the message size exchanged between the central system and RTUs will establish the maximum number of RTUs that each MS can handle.

The system installation can be very complex sometimes if wireless is used to access to RTUs. If that is the case, antenna towers may be required.

2.7.3 RTUs Communication

The RTUs have the communication modules built in to handle the requests from the central system and to respond to them. The communication aspect can be software, electrical and/or mechanical.

Each RTU has to be able to handle the protocol requests from the master station. Also, in some cases the RTUs have to be capable to control the communication channels.

In many remote environments, wireless communication on the RTUs is one of the main causes for power starvation. The communication can consume more power than the sensors used to acquire the required data or to execute an action.

2.8 SCADA Protocols

There are many possible protocols that can be used for SCADA. Two popular protocols are MODBUS and DNP3. There are also proprietary protocols and some industry specific protocols such as BACnet, "a data communication protocol for building automation and control networks" designed initially for the HVAC - Heating Ventilating Air Conditioning - industry. What makes BACnet special is that the rules relate specifically to the needs of building automation and control equipment—*i.e.*, they cover things like how to ask for the value of a temperature, define a fan operating schedule, or send a pump status alarm [6]. With the recent advance of internet and the reliability of networks, TCP/IP has come as a widely accepted alternative. Moreover, SNMP – a monitor and control protocol implemented over TCP/IP – has been proposed for SCADA [7]. MODBUS was introduced in 1979 by Telemecanique. The protocol has evolved but

kept its original rigid structure. MODBUS is an open standard widely used and accepted as an industry standard. MODBUS is a byte oriented protocol and independent of the network used. The MODBUS protocol has a MASTER/SLAVE topology where a QUERY is sent from the master device (MD) to the slave device (SD). The master sends a query message and the slave responds with a response message. The response can be the answer to the initial query request or an exception condition. The MODBUS message is rigid (non-flexible) with a unique ID that identifies each device in the network. MODBUS has improved from a simple RS-232 communication based protocol to a MODBUS over TCP/IP [8].

DNP3 is a protocol developed by GE Harris Energy Control Systems Canada Ltd. in the early 1990s. The protocol was attempting to address a need for communication between a master station and RTUs. The protocol became open in 1993 when the DNP user group was created. The protocol is regulated by the community using it. In order to implement the protocol that is DNP compliant, a membership to the group is required. The DNP3 was created having SCADA systems in mind. The DNP3 protocol was developed to help electrical companies standardize communications between equipment. From a network perspective, the DNP3 protocol has multiple layers: application, pseudo transport and physical. As such, is it not just an application layer protocol [9].

There are protocols such as SNMP - Simple Network Management Protocol - that can be used to control equipment remotely as well as SCADA-like systems. Although SNMP was not initially intended for SCADA system it is widely used for that purpose. SNMP is a protocol that works over TCP/IP to gather status data from remote systems such as servers and routers. It is also used to set values remotely to change configurations [7].

There is no simple answer to the question: What is the best protocol to use? Each protocol has advantages and disadvantages. The following list is intended to provide additional criteria at the time to decide which protocol to use:

a) Usually when a new SCADA system is required, there is something that has to be controlled. As a consequence, the protocol that might be adopted is the one used by the equipment to be controlled.
b) If the SCADA also contemplates the ongoing development of the RTU, the best practice is to understand the most used protocol for that particular field. For example, if the system is being developed to monitor electrical consumption in electrical plants a good option could be DNP3.
c) If the SCADA system will be acquiring data from remote sensors where the communication channels are narrow (e.g. 9600 bps) sometimes is a good option to create your own protocol. Existing protocols such as MODBUS or DNP have a lot of overhead. However, this could also narrow the opportunity to integrate a system with third party vendors. Different fields have different SCADA protocols. Additional research is necessary to understand the alternatives on a particular field.

There are many different protocols and not a simple answer to the best protocol to be used. It is very common to have SCADA systems with multiple protocol support. The contention here is to tend towards protocols based on the TCP/IP model wherever possible.

2.9 Environment [3]

There are many environmental conditions that could affect the normal functionality of a SCADA system. The most important considerations are:

a) Exposure to chemicals, acids or dust

Chemical or acids are pollutants that can affect the normal operation of the equipment. Also, dust is other problem that could affect the normal functionality of the system. To address these issues special enclosures, materials and filters are suggested.

b) Extreme temperatures or humidity

Special considerations have to be taken when the system is installed in environments where temperature or humidity could be extreme. A typical solution to this problem is use equipment, part or components that tolerate the extreme environmental circumstances. Sometimes a proper enclosure or a cooling or heating system could also help to address the problem.

c) Abnormal vibration or shock

The system has to be able to withstand some level of vibration. This requirement not only considers the place where the system will be installed but also how the system will be handled during transportation and installation.

d) Altitude of operating location

The IEEE standard required that the system has to be suitable to work in places with altitudes up to 2000m.

e) Abnormal electromagnetic interference (EMI) and Electromagnetic compatibility (EMC)

Manufacturers shall design and test their equipment to ensure that EMI limits are not exceeded, and users shall design and test locations (environments) to ensure that EMC limits are not exceeded [3].

f) Abnormal exposure to ultraviolet light (UV)

If the equipment will be installed outdoor special considerations have to be taken to avoid UV interference. Usually a cabinet with some NEMA - National Electrical Manufacturers Association - level certification will filter the UV.

g) Lightning and switching surge protection

There is more than one way to protect a system against lighting discharges and switches surges. In order to achieve the best possible protection, a combination of factors have be followed from the design stage up to proper installation.

The variety of factors a SCADA system has to contend with, is also one of the reasons that a universal SCADA framework has not evolved.

2.10 Reliability

The reliability is the probability that a unit or system will perform its intended function under specified conditions during a specified period of time. The manufacturers of SCADA solutions usually provide the system with the mean-time-between-failure or MTBF. This value provides an idea of the useful operating life of the equipment [3].

2.11 Availability

Availability is defined as the ratio of uptime to total time (uptime+downtime). i.e., uptime/(uptime+downtime) and it is normally expressed as a percentage of the total time. Downtime normally includes corrective and preventive maintenance. When system expansion activities compromise the user's ability to operate apparatus via the system this may also be included in downtime [3].

The table 2.1 shows typical availability percentages and real annual downtimes.

Availability	Annual Downtime
99.9999%	31.53 sec
99.999%	5.25 min
99.99%	52.56 min
99.9%	8.76 hours
99%	3.65 days
98%	7.3 days
97%	10.95 days

Table 2.1: Typical availability values and annual downtimes

2.12 System Security

SCADA systems are often responsible to control critical and strategic systems such as a regional electrical system. The system security is a very important element to consider. The following recommendations should be considered to enhance security with a SCADA framweork: 1) Isolate control systems from insecure networks by disconnection or adequate firewalls; 2) Adopt best practices for password control and protection, or install modern authentication mechanisms; and 3) Provide for individual accountability through protected action logs or the equivalent [10].

> *For several years, security risks have been reported in control systems, upon which many of the nation's critical infrastructures rely to monitor and control sensitive processes and physical functions. In addition to general cyber threats, which have been steadily increasing, several factors have contributed to the escalation of risks specific to control systems, including the (1) adoption of standardized technologies with known vulnerabilities, (2) connectivity of control systems to other networks, (3) constraints on the use of existing security technologies and practices, (4) insecure remote connections, and (5) widespread availability of technical information about control systems [11, pp 1].*

There are three areas covered in this section: redundancy, action completed and data encryption or intrusion detection.

2.12.1 Redundancy

The large implementations of SCADA are usually distributed systems. The master station is replicated to distribute the monitoring and control. At the same time, the distribution also implies that the system may be replicated. There is another system that can take over the operations without incident, in case of a failure in one node.

The redundancy is not only at the master station level. Sometimes, redundancy is needed in the communication channels or in the RTUs to reduce or eliminate risks in case of potential failure.

2.12.2 Action Completed

The action completed feature is a system that provides the master station with redundant information if a particular action was successfully executed. This kind of system also has a duplicate command feature. When the action taken by the SCADA is very critical, the RTU will expect to receive the command twice from the master station before executing it.

2.12.3 Data Encryption or Intrusion Detection

Modern SCADA systems use different kind of networks to send and receive data. If some of the networks are open, they can be intercepted potentially causing several damage.

It is critical to encrypt the data to insure data integrity if the SCADA system is using a wireless network. There are new SCADA systems that are using cell phone carriers as the communication solution provider. It is important to take all possible precautions to keep the communication channels isolated and secured. Some distributed SCADA systems are using VPN - Virtual Private Network - over internet to exchange data between them. If that is the case, additional security layers are needed not only to prevent intruders from access to the system but also to

assure that the data sent from one station will arrive with integrity to the other one. Issues of data security and privacy are not new, and solutions developed in the web application community can be directly port to the SCADA environment.

2.13 Conclusions

Although the standard for SCADA systems are complex, it should evolve in a different direction. The following describes how the applications prototyped will contribute to a standard SCADA framework. A modified SCADA framework is shown in Figure 2.3 based on a review of SCADA standards as well as expertise gained in developing SCADA combined with a background in modern telecommunication systems.

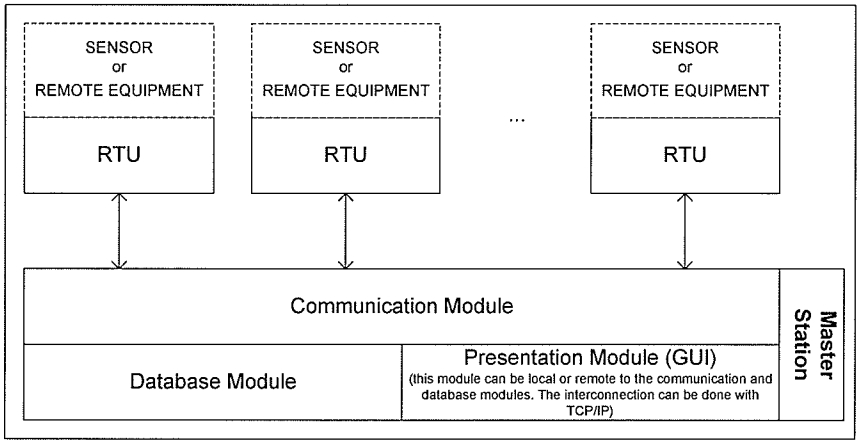


Figure 2.3: Proposed SCADA framework

This template SCADA framework is built upon work from other areas, and wherever possible addressing the main difficulties of SCADA such as deployment, legacy protocols and technologies.

Several elements are innovative in this new framework. This framework proposes an integration between the RTU and the sensor, thus saving costs. Usually, each RTU will have at least one microcontroller to handle the data acquisition, control, and communication exchange between

the remote device and the master station. The sensors have evolved to a level that they not only have better interfaces to interact with the microcontrollers, but they are also more affordable. Other changed is that the master station could be a Linux based station. The software part of the SCADA could be composed of multiple modules such as communication module, database module and presentation module (GUI). The data exchange between these modules can be done using TCP/IP or at high levels such as SOAP - Simple Object Access Protocol. Moreover, the modules can be installed on different systems and they can be local or remote to each other. There is a tendency to integrate SCADA solutions with existing corporate systems in order to make SCADA part of the information systems. SCADA will also evolve into web based format to allow systems to be controlled from anywhere in the organization [12].

The present chapter presented a framework that will be adopted in the next chapters to develop prototype SCADA.

Chapter 3: The Development of a SCADA for Monitoring Sandbag Dikes

3.1 Introduction

The SCADA system developed here was based on the most important principles described on the previous chapter. The goal was to create an SCADA system that is not only efficient and cost-effective but at the same time sensor agnostic so that it can be redeployed for subsequent applications. Moreover, the SCADA created for this project is portable. The next chapter describes how the SCADA system used to sense the sandbag dike movement can be reused in an RTLS (Real Time Location System) environment. This is possible because of the purposed SCADA framework.

SCADA is a suitable solution to monitor sandbag dikes for potential structural damage. The system senses the motion of the sandbags (due to water pressure and wave action against the dike) using an accelerometer. Dike movement is a sign of structural weakness and a potential precursor to dike failure. Figure 3.1 shows how the sensors are installed in the dike.

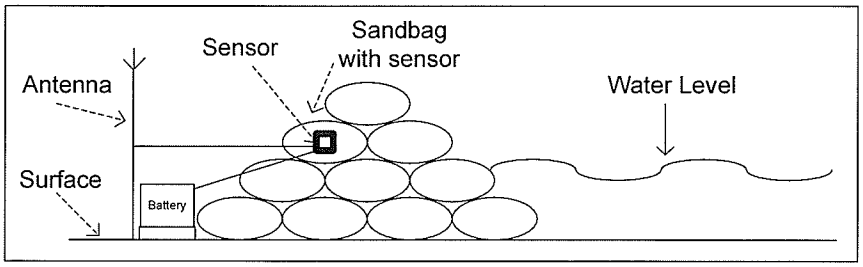


Figure 3.1: Overview of the sensors installation

The system has three main components: the sensors installed inside the sandbags, the communication system to send and receive data to/from a remote station and the SCADA software capable to collect the data and to present it to the user.

The sensors inside the sandbags are composed of different elements. The main element is a microcontroller responsible for capturing the data from the sensors and sending it back to the SCADA system. Each sensor to be installed inside the sandbag was built in a water proof enclosure. Inside the enclosure multiple elements were installed. One element is the accelerometer to detect motion of the dike itself. In addition, the system has an internal temperature and humidity sensors, an external temperature sensor and a battery meter. The communication module and the microprocessor are the RTU - Remote Terminal Unit - for this system.

The internal temperature and humidity sensors were installed to understand the conditions under which the electronics had to work. The external temperature was monitored as well to understand the discharge curve of the battery at relatively low temperatures. Finally, a battery meter was installed to measure the battery charge.

Another project objective was to create a wireless SCADA system to collect data from the sensors in the dike and to understand battery life under cold weather conditions. Moreover, the data collected is intended to analyze sandbag dikes' behaviour under water pressure and potential structural damage due to waves - an extreme condition.

3.2 Sandbag SCADA Idea

The sandbag dikes are built to defend land and property against rising water levels during seasonal flooding. Sandbag dikes are built by volunteers in times of emergency and sometimes are not properly constructed. The first version of the system was developed under the supervision of Dr. James Blatz – Professor of Civil Engineering at University of Manitoba - by Jay Ferchoff -

an undergraduate student - who was responsible for the implementation of the first version under a research project called "Remote Sandbag Dike Monitoring" dated August 22, 2008 [13].

The project was completed successfully and results were collected. Moreover, one of the most important conclusions was that with the right sensors inside the sandbags, a real time system to detect structural failures in sandbag dikes is possible. The first version was not installed in the field but was a laboratory prototype.

3.3 Sandbag Monitor Version 2

The second version of the sandbag dike project is based on the initial approach developed by Jay Ferchoff. The second version was designed using SCADA best practices and tested in a real flood environment. This version of the SCADA was completed in the middle of the 2009 Manitoba flood season where a sensor was installed in a sandbag dike along the Red River and data collected.

The first approach was to make a design of the entire system and then divide the system in individual subsystems (Figure 3.2).

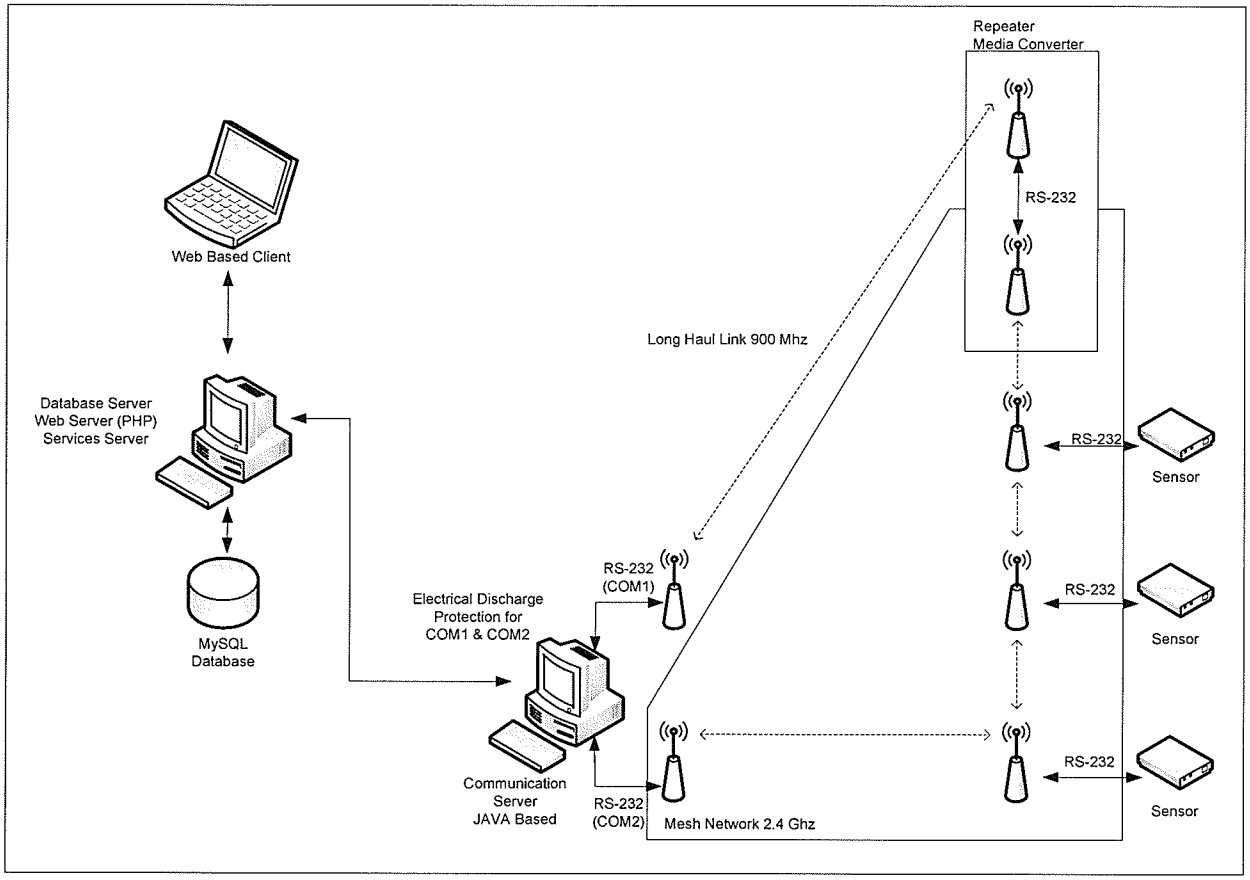


Figure 3.2 SCADA system design to monitor data from sandbag dikes wirelessly

The criteria used to create the subsystems was to address challenges in different areas (divide and conquer principle). Some of the challenges were financial while others were focused on technical aspects or level of expertise on certain topics.

The system has the following subsystems:

a) Inclinometer and/or accelerometer

b) Microcontrollers and firmware

c) Radio modem and antennas

d) Enclosures

e) GPS

f) Batteries

g) Battery charger

h) Cables

i) Computer software

3.3.1 Accelerometer

The first component to investigate further was the accelerometer. The original project used inclinometers to sense tilt. The idea with the accelerometer was to sense motion. Furthermore, the tilt or slope can also be calculated using individual axis acceleration.

The accelerometer is a device used to measure acceleration. The device used contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration [14].

The accelerometer used was the ADXL335 from Analog Devices. The selected device has 3-axis sensing and a sensitivity of ±3g. The sensor has analog outputs for X, Y and Z axis. Furthermore the accelerometer has a very low power consumption (350 μA typical) ideal for part of the remote system to sense the sandbag dike's possible movement.

Each analog output is represented by a voltage. The voltage needed to power up the accelerometer is the same as the microcontroller (3.3V). The nominal voltage for the output of the accelerometer is $V_{dd} / 2 = 1.65$ V.

The accelerometer has three outputs; one for each axis. If there is no movement in any direction the output value for each axis will be 1.65 V. However, if there is movement the value will increase or decrease based on the sense of the movement.

Consider the output for the X Axis. The output value will be 1.65V if the accelerometer is not moving. However, if there is some movement in the X direction the voltage will increase and if it is moved in the -X direction the voltage will decrease. The voltage changes radio metrically 300mV/g (typically) [14].

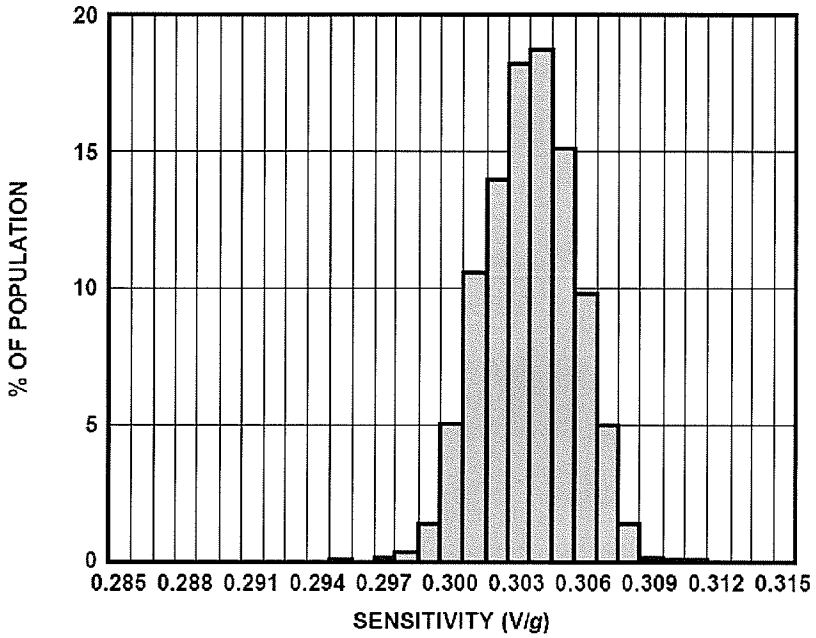


Figure 3.3: Accelerometer Sensitivity at 25°C, Vs=3V [14]

3.3.2 Microcontroller and Firmware

The microcontroller is the RTU core. The software built for the microcontroller is responsible to get the data from the sensors and send it back to the master station. The master station polls the RTUs for data from the sensors. The polling is done at regular configurable time intervals.

The microcontroller selected is the 4200 module from Rabbit Semiconductors. The reasons for the selection were price, familiarity, and module adaptability to the requirements. The microcontroller is 8 bits and works at 58.98 Mhz. Each RTU has a microcontroller with firmware to get the data from the individual sensors. In addition to the accelerometer, each RTU gets also information about the internal and external temperature, internal humidity and battery charge.

The software on the RTU collects data from the accelerometer until it gets a data request from the master station. After the system gets the petition, the software sends temperature, humidity and battery charge from sensors. Then, the accelerometer values are appended and the data is sent back to the master station. Finally, the loop starts again and the software collects new data from the accelerometer until a new request arrives.

The microcontroller software (Figure 3.4) gets readings of the accelerometers (50 at the time to filter noise at a sampling rate of 29 Mhz) and then checks if there is a request for data from the main station. The software gets additional values from other sensors and sends the data back to the master station, if there is a request. If there is no request, more readings are taken from the accelerometers. The accelerometer values sent to the master station are the maximum difference - worst case or maximum movement - between the first reading and the actual value for that particular cycle. The requests from the master station were sent every 15 seconds. In other words, the system was designed to detect a potential failure within 15 seconds that the initial movement is detected.

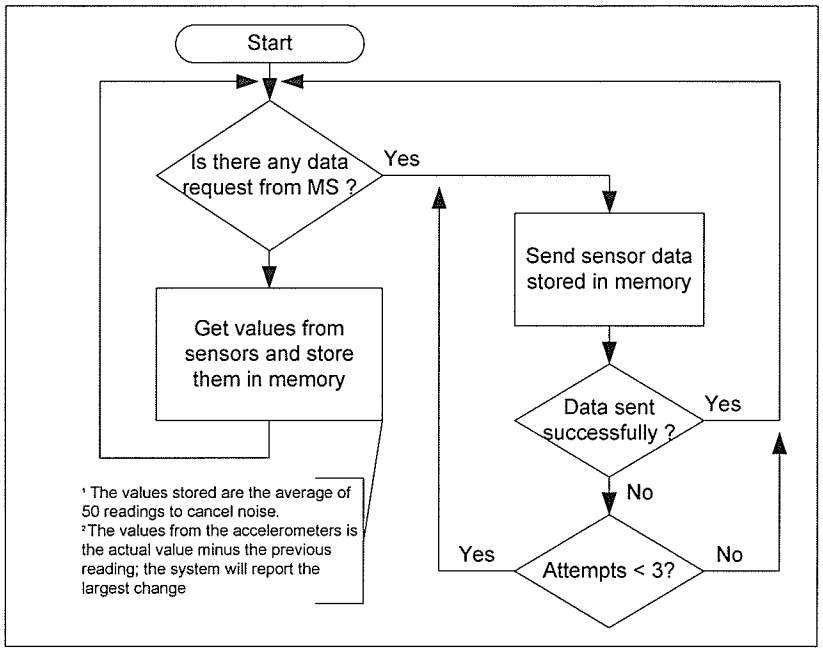


Figure 3.4: Microprocessor flowchart

The request from the master station arrives to the microcontroller with the symbol $ (0x20H). Then, the microcontroller responses using the convention described in Figure 3.5

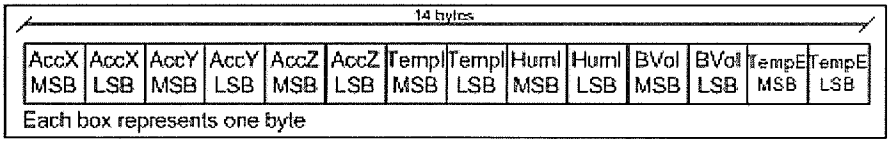


Figure 3.5: Convention utilized by the microcontroller in the RTU to send data to the master station

The microcontroller has seven sensors: accelerometer for axis X, accelerometer for axis Y, accelerometer for axis Z, internal temperature, internal humidity, external temperature and battery charge. All the sensors used are analog. The output of the sensors are connected to the microcontroller ADC converter (Figure 3.6).

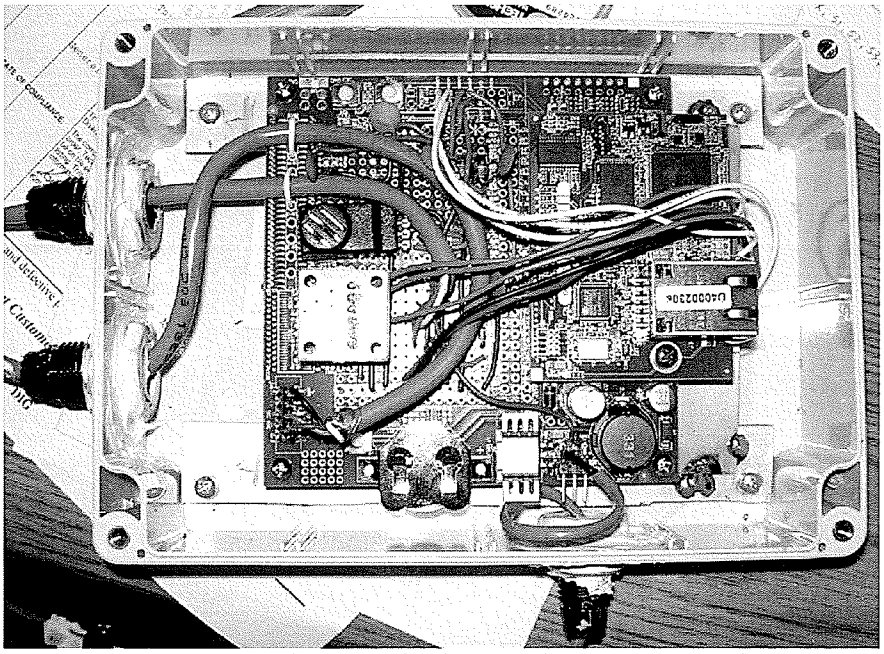


Figure 3.6: Microcontroller with evaluation board and sensors mounted in the enclosure

3.3.3 Radio Modems and Antennas

The communication channel between the MS and the RTU is very critical because it will determine the system range, data bandwidth and costs.

The system was designed using off-the-shelf RF solutions from Digi International. The system was designed keeping redundancy in mind (see Figure 3.7). However, for the first version only the mesh network option was implemented because of technical difficulties encountered with the long haul wireless link.

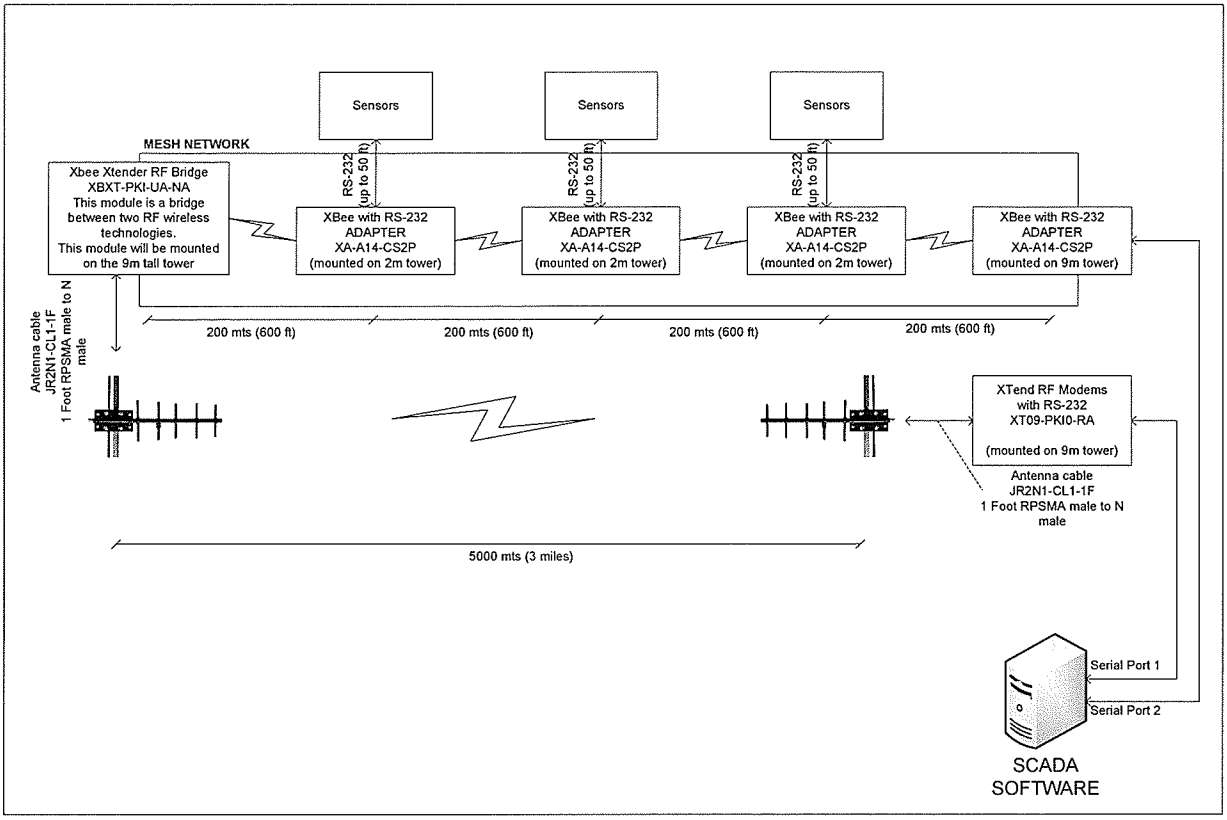


Figure 3.7: System design overview

The initial design considered two communication channels. The MS containing the SCADA software has two serial ports utilized for this purpose.

A device called XBee to Serial Port adapter is connected on one serial port. The wireless protocol used is a Digi International proprietary system using mesh topology at 2.4 Ghz. The protocol handles the communication process, error control and device identification. The mesh network adaptor is also installed in each sensor.

The master station polls data from the sensors using the wireless mesh network. The message received by the XBee node attached to the master station is similar to 'send message X to the device Y'. After the message is sent a confirmation is received from the transmitter. Each device has a unique 64 bits ID called serial number set at the factory. This is like a MAC - Media Access Control - address for Ethernet technologies.

The XBee module has two operation modes. The first mode is called transparent mode. This mode is like a RS-232 wireless bridge. The problem with this mode is the difficulty to address requests to different remote stations with different IDs. The second mode is called API - Application Programming Interface. In this mode, each request contains the destination address and other information necessary for the transmission. In the API mode each function has a 1 byte unique identifier - i.e. 0x10H, 0x90H, etc -. The master station sends the data using function 0x10 and receives transmission confirmation with function 0x8B. The sensor receives the request with function 0x90. Then it processes the request and sends the data back using function 0x10. After the transmission the microcontroller receives a 0x8B from the RF module [15].

The master station will attempt to send the request for data a number of times as previously configured in the system. If the transmission is not successful, it will report a communication problem. On the microcontroller side, the system will attempt to send the data three times. After that, it will drop the request.

Each communication module uses the same battery that the microcontroller and sensors are using.

The maximum distance between each mesh adaptor is 2.2 km. However, because of environmental conditions that distance can be significantly reduced. The objective was to install a sensor every 200 meters. Each transceiver had to be installed using an antenna at least 2 m tall to achieve the maximum distance of 2.2 km.

The mesh network has the feature that each sensor can talk to each other. Moreover, if one sensor between two fails the sensors on the extremes can talk to each other if still within the network maximum range.

The sensors together with the microcontroller are installed in an enclosure inside the sandbags. The RF adaptor is installed outside the sandbag. The communication between the microcontroller and the adapter is done using an RS-232.

The system was designed considering a distance of 200 meters between sensors. The manufacturer specification is that two nodes can see each other within a range of 2.2 km. If the dike fails, up to 10 sensors could be affected and the system would continue to work.

Additional redundancy was considered for use in the system in the event that a major catastrophe arises. A redundant communication channel was created between the MS and the end of the sensors in the dike as shown in Figure 3.5 using a second serial port on the MS.

The redundant channel uses the XTend module from Digi to create a point to point long haul network. The design specifications were intended to create a redundant channel able to send data up to 6 km from the master station. The XTend modules with the high gain antenna installed (8 dBi) uses a frequency of 900 Mhz and it can send data in a point to point configuration to a distance of 64 km with a proper antenna tower. This module uses a power of up to 1W.

Due to technical difficulties to convert from the XTend protocol to the mesh protocol, only the mesh network was used for the prototype. The manufacturer of the XTend modules at the moment that the system was being developed had some problems to integrate both networks. Moreover, a product called XTendGateway was removed from the market. As a consequence, the redundancy was not implemented because the integration between the mesh network and the XTend was not possible.

3.3.4 Enclosures

Three different enclosures were used to protect the parts from environmental conditions.

The first enclosure size is 6.73X4.76X2.17 (Figure 3.8) inches where the prototype board, the microcontroller and sensors are installed inside the sandbag.

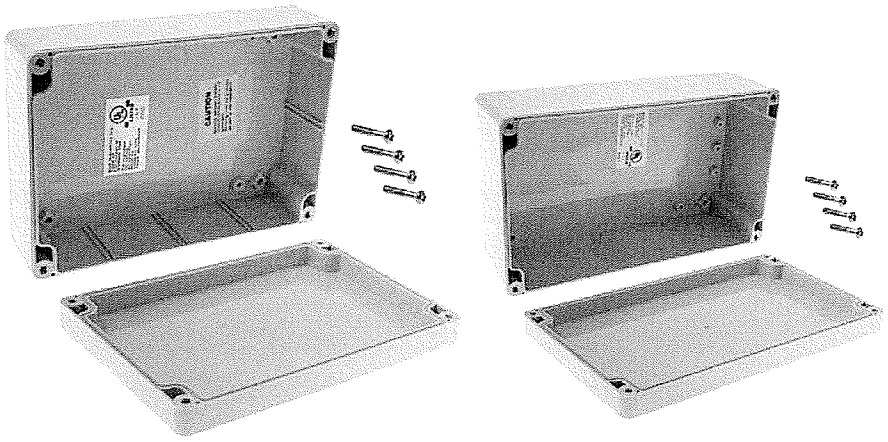


Figure 3.8: Enclosures: On the left enclosure for the microcontroller and sensors.
On the Right enclosure for the RF circuits and antenna

The second enclosure used was 7.87X4.72X2.95 inches. This enclosure is used to mount the RF component of the system outside the sandbag.

The enclosures had NEMAx4 compliant for UV and fire protection. The enclosures were also water proof. NEMA is an acronym for National Electrical Manufacturers Association. The type four are enclosures constructed for either indoor or outdoor use to provide a degree of protection to personnel against access to hazardous parts; to provide a degree of protection of the equipment inside the enclosure against ingress of solid foreign objects (falling dirt and windblown dust); to provide a degree of protection with respect to harmful effects on the equipment due to the ingress of water (rain, sleet, snow, splashing water, and hose directed water); and that will be undamaged by the external formation of ice on the enclosure [16].

Additionally, two special connectors were used for the cables (Figure 3.9).

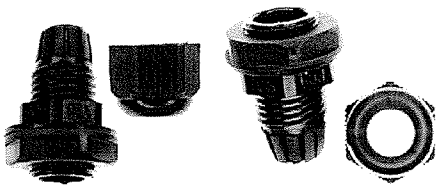


Figure 3.9: Cable Gland: Two different connector sizes to install cables in the enclosures

Tests ran with the enclosures showed that the system is water protected but it was not prepared to be submerged. As a consequence, additional steps were taken with the enclosure that was installed inside the sand bag.

The first waterproofing difficulty was a problem with cable glands. They were not able to stop the water from entering the enclosure. As a consequence, after the installation of the glands in the enclosure acrylic latex silicone was used. (Figure 3.10)

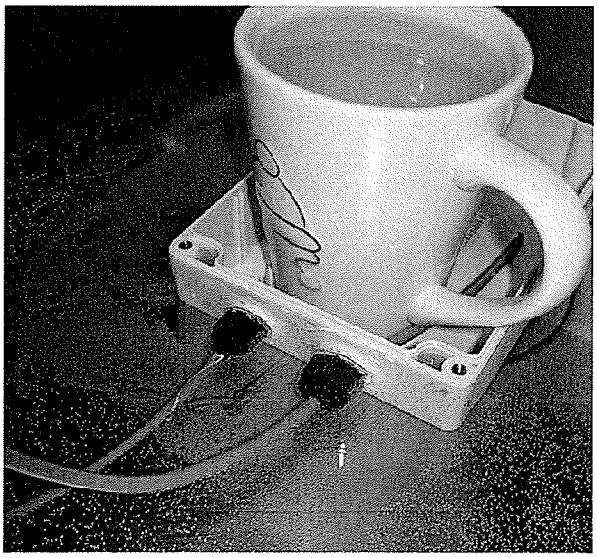


Figure 3.10: Enclosure submerged in water to run water proof test with latex silicone

Finally, to provide an extra layer of protection the enclosure installed inside the sand bag was wrapped with a sealed bag, of the type used routinely in environmental sampling of soils. (Figure 3.11).

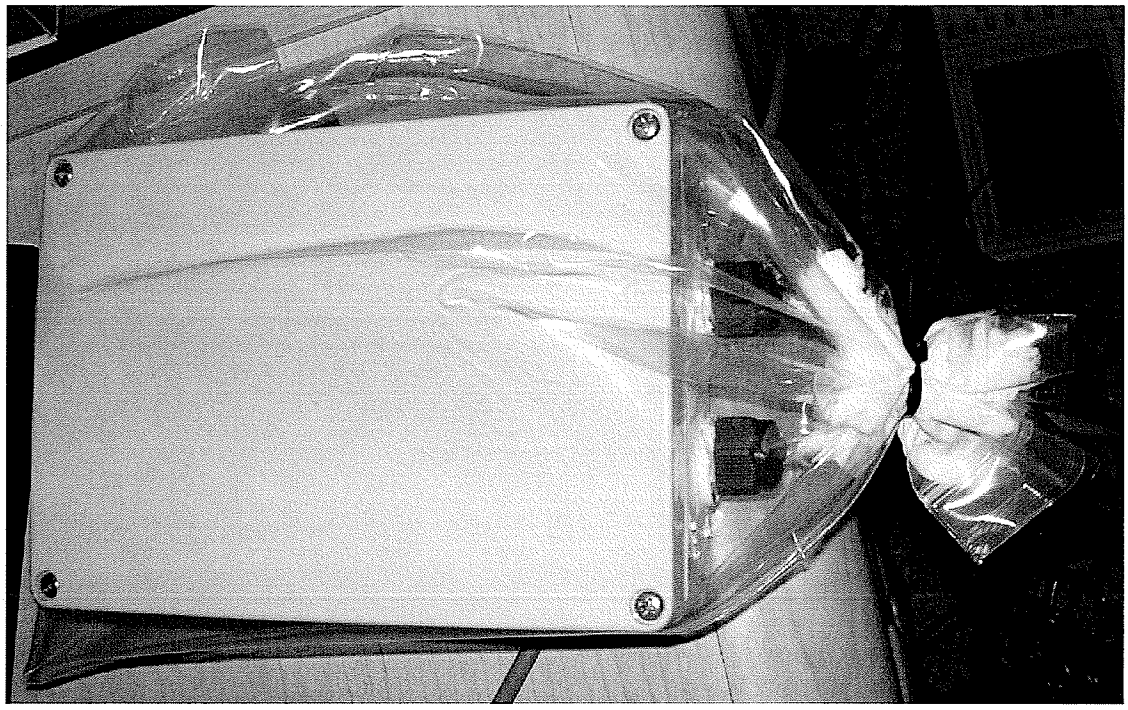


Figure 3.11: Enclosure sealed with electronics inside ready to be installed inside the sandbag

The last enclosure used was built out of wood. This enclosure was used to install the battery to protect it from the weather. The enclosure was also used to install a temperature sensor to measure external temperature. (Figure 3.12)

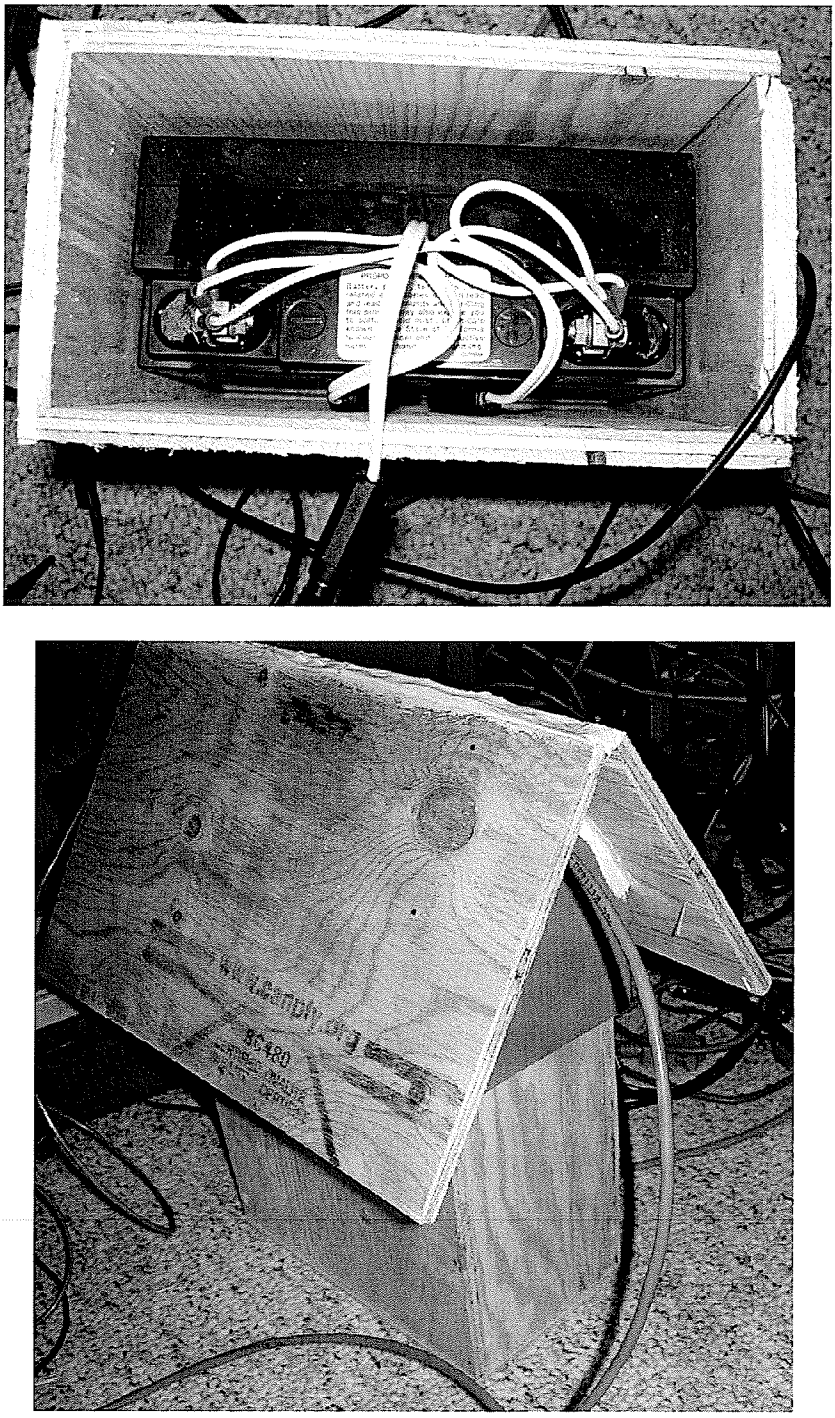


Figure 3.12: Enclosure built to protect the battery and to hold the external temperature sensor

3.3.5 GPS

The GPS is an optional component for the system. The GPS was not implemented in this version of the system. However, for future developments the GPS BR-355 from USGLOBALSAT is recommended. Some preliminary tests were conducted. The selected device is very cost-effective, water proof, resistant to low temperatures and has low power consumption.

The practical scenario is that the crew that will be building the dikes will have some bags with the sensors inside. It is very difficult to assign each bag to a specific location. In consequence, the GPS will help to locate the sensor.

3.3.6 Batteries

The battery used for this sensor is a Panasonic battery model LC-RD1217P. It is a 12V lead acid battery with a capacity of 17Ah.

The battery was selected based on the following reasons: 1) the battery is a maintenance-free lead acid battery, which could operate in any position; 2) this kind of batteries are typically used in mobile phone repeaters, cable distribution centers, internet hubs and utilities; 3) product quality; 4) power consumption of the electronic in the system; 5) low cost; 6) dependable service and minimal maintenance requirements; 7) not subject to memory; and 8) low self-discharge [17].

The system current consumption was calculated around 100mA @ 12 V. The first objective was to have the battery charge to last a week. The intention was to test if it is possible to run the system with batteries without a battery charger. The sections 3.4 and 3.5 present additional information about tests results and conclusions.

3.3.7 Battery Charger

The battery charger was something that was not included in the system because of the price in comparison with the whole system. The battery charger cost the same as the sensor with the RF and battery. As a consequence, because of the excessive price of individual sensors with the charger the system would not have potential for commercialization.

Two different battery chargers were considered: one using solar power and another using a small wind power generator.

The idea of replacing the battery with a similar one charged every week was possible and affordable since the system is seasonal. Sections 3.4 and 3.5 have more information about tests and conclusions.

3.3.8 Data and Power Cables

The system needed five different cables (Figure 3.13). Four of the cables are needed on the sensor side and one on the computer side.

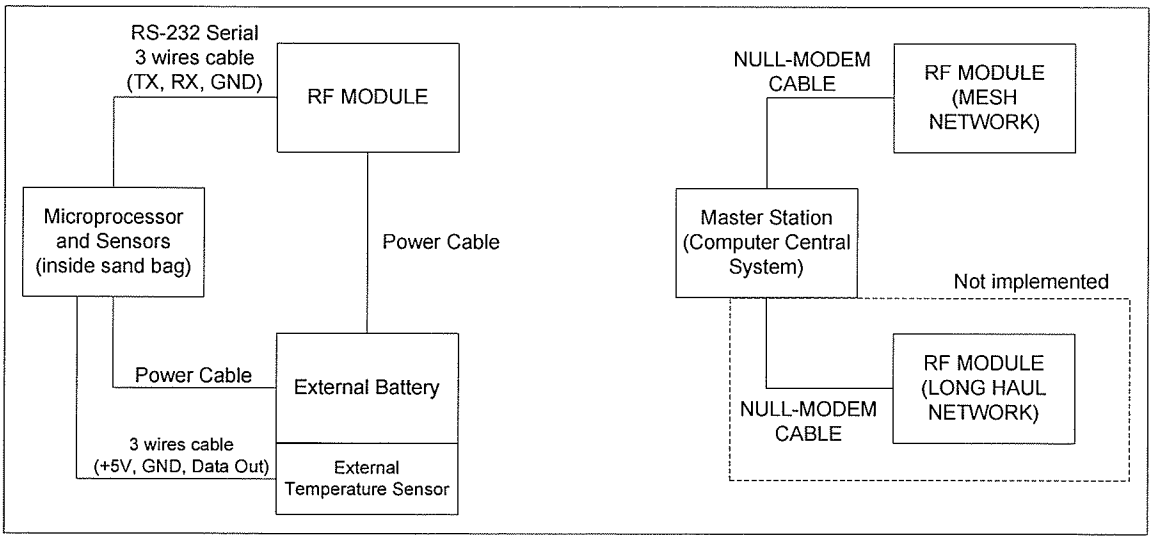


Figure 3.13: Cable diagrams for the microcontroller and sensors and for the central computer system

On the sensor side the following cables are used:

1. Power the microcontroller and sensors inside the sandbag with the external battery.

2. Power the RF module with the battery.

3. Three wire cable (TX,RX,GND) from the microcontroller to the RF module to send/receive data using RS-232.

4. Three wire cable (+5V, GND, DATA OUT) to the external temperature sensor located with the battery.

On the computer side only one null-modem cable was used to connect the computer with the node for the mesh network. In future implementations, another cable will be required to create the redundancy with the other long-haul network.

3.3.9 Computer Software

The computer software has three basic components: the communication module, the database and the presentation module (GUI).

The communication module was developed in Java. The communication module has the physical connection to the RF nodes responsible for the communication with the sensors.

The database component stores system configuration, historical data and last read values from sensors. Moreover, the database is used to link the data gathered by the communication module to the GUI.

The GUI is web based and presents the data from the database to the user. The data is presented in tables and graphs.

3.3.9.1 Database

The database selected for the project was MySql server. The database has four tables. The table descriptions can be seen in Appendix A. The table Configuration is used to create general system settings such as timeouts and alarms types[1]. The table Sensors contains all the information per individual sensor such as Sensor ID, last known valid data read from sensor, initial sensor settings, last record update, last read status and sensor correction factor. The table Scan is a table used by the communication engine to know the next sensor to poll. This table stores the sensor ID, priority and last read status. The last table is the HistoricalData. This table stores each reading from the sensor. In case of an error such as Communication Error or Timeout Error, the system will create an entry showing the problem. The HistoricalData table is used to create historical graphs to understand system behaviour.

3.3.9.2 Communication Engine (CE)

The Communication Engine (CE) is the responsible to get the data from the sensors into the database utilizing pre-established configuration. The CE was developed in Java. The CE application runs in the background 24/7/365.

The first thing that CE does is get configuration information from the configuration table such as the time to wait before requesting the next reading.

The main input for the CE application is the Scan table. The Scan table will tell CE the next sensor to capture data from. The system will retrieve the data from the SCAN using the LastRead and Priority columns. The default priority is 5. The system will retrieve data from sensor with priority 5 and with the most recent LastRead value under normal circumstances. If a user using

[1] The notifications can be by email or with sound. This option was not implemented but considered for future versions.

the GUI interface clicks on a particular sensor, the priority level will be changed to 1. In that case, the system will read one record with priority 5 and one with priority 1 from two different sensors. The data on the user screen will be updated more frequently using this technique than with the regular mode.

For example, if the system has 10 sensors and the reading time is 1 second per reading, to get one sensor data updated you will have to wait 10 seconds. Using the priority system, the CE will scan one regular node and one with higher priority (priority=1). In this way, the GUI reporting a particular sensor will be updated every 2 seconds maximum, compared with 10 seconds if the priority were not in place.

Finally, the data collected from the sensor is stored into the HistoricalData table. That information is also stored for future analysis in a case of an error.

The communication flowchart structure can be seen in the Figure 3.14

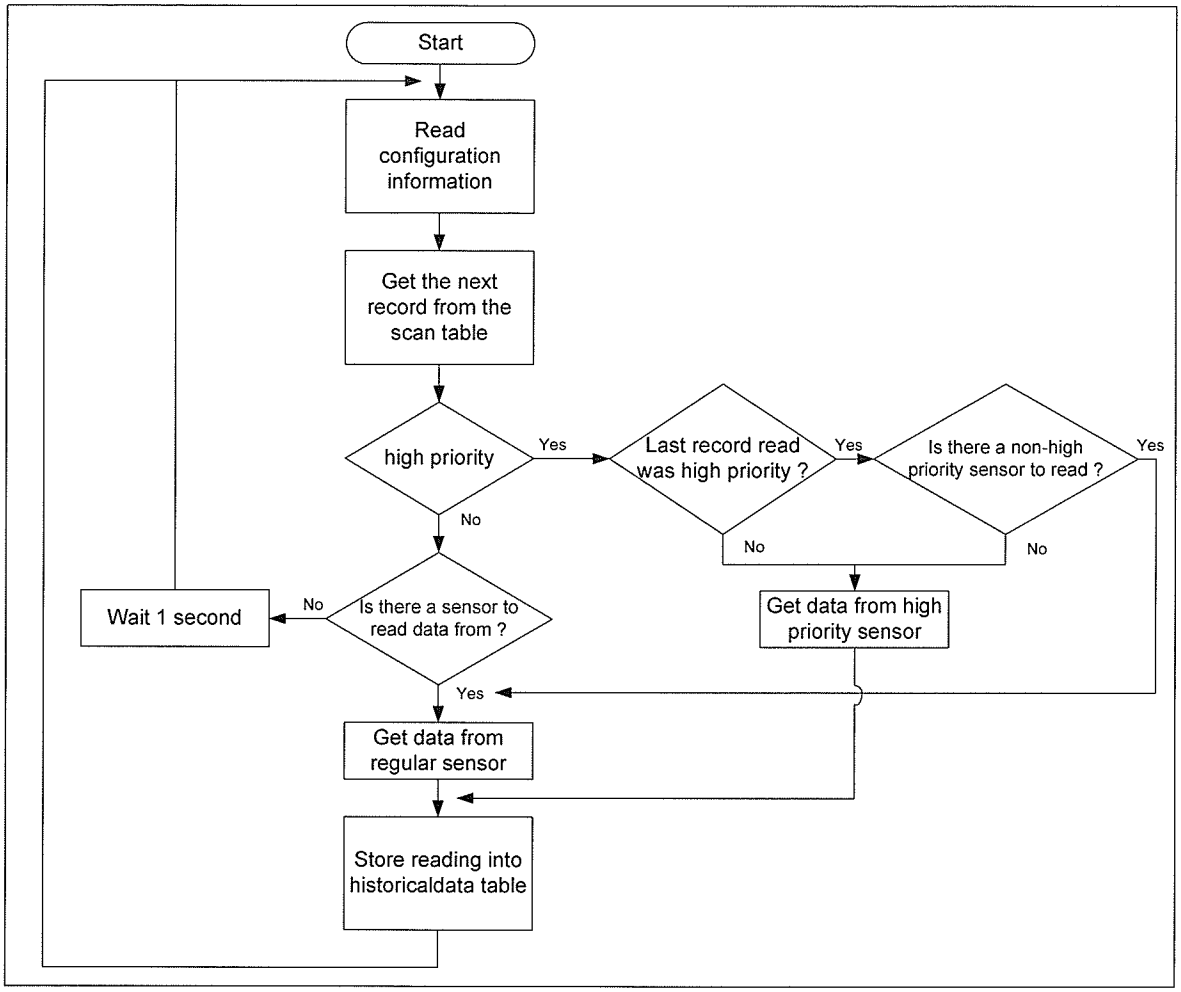
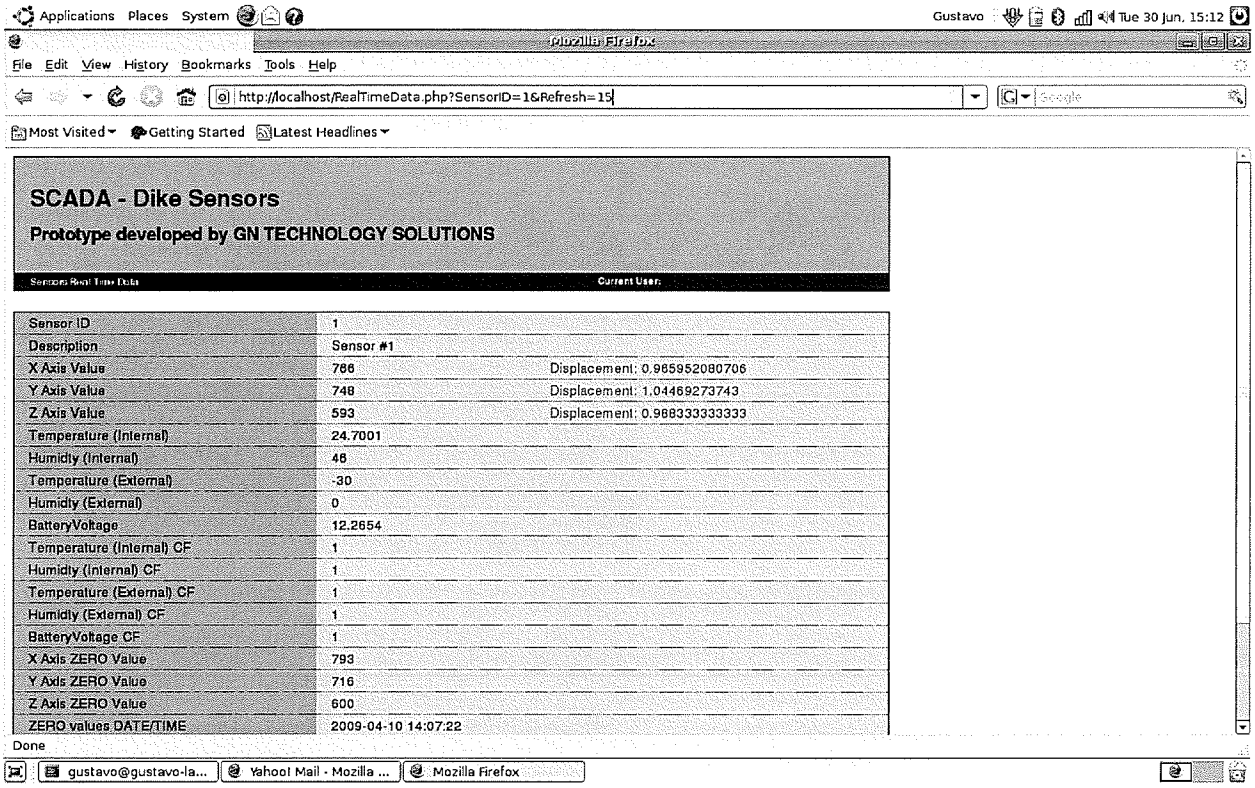


Figure 3.14: Communication Engine Flowchart

3.3.9.3 GUI

The GUI is built on web technology and it has multiple modules. The selected language was PHP because of its flexibility, low cost, high efficiency and for its interface with MySql. Each module has different functions. The first module name is RealTimeData (see Figure 3.15). The input parameters are the sensor ID and the refresh frequency. This module gets its data from the Sensors table. This web page will refresh every 'Refresh' seconds updating the last known information from the sensor. When the user accesses this page the priority for the sensor changes in the Scan table. In consequence, the user will get values more frequently for that sensor than

the other sensors. The user will have the impression that the system is working almost at real time.



Sensor ID	1	
Description	Sensor #1	
X Axis Value	766	Displacement: 0.965952080706
Y Axis Value	748	Displacement: 1.04469273743
Z Axis Value	593	Displacement: 0.988333333333
Temperature (Internal)	24.7001	
Humidity (Internal)	46	
Temperature (External)	-30	
Humidity (External)	0	
BatteryVoltage	12.2654	
Temperature (Internal) CF	1	
Humidity (Internal) CF	1	
Temperature (External) CF	1	
Humidity (External) CF	1	
BatteryVoltage CF	1	
X Axis ZERO Value	793	
Y Axis ZERO Value	716	
Z Axis ZERO Value	600	
ZERO values DATE/TIME	2009-04-10 14:07:22	

Figure 3.15: RealTimeData module

The next module is called RetrieveData. This module will retrieve the data from the database and present it to the user in a friendly manner. The columns are: date and time of the reading, accelerometer values for X,Y,Z, internal temperature and humidity, external temperature and humidity, correction factor (default: 1), status (ok, communication error, timeout, etc).

This module has two options to recover the data. The first option is to use the sensor ID and the number of records to retrieve (see Figure 3.16). The number of records is counted from the bottom. In other words, you will see the last X number of records recorded.

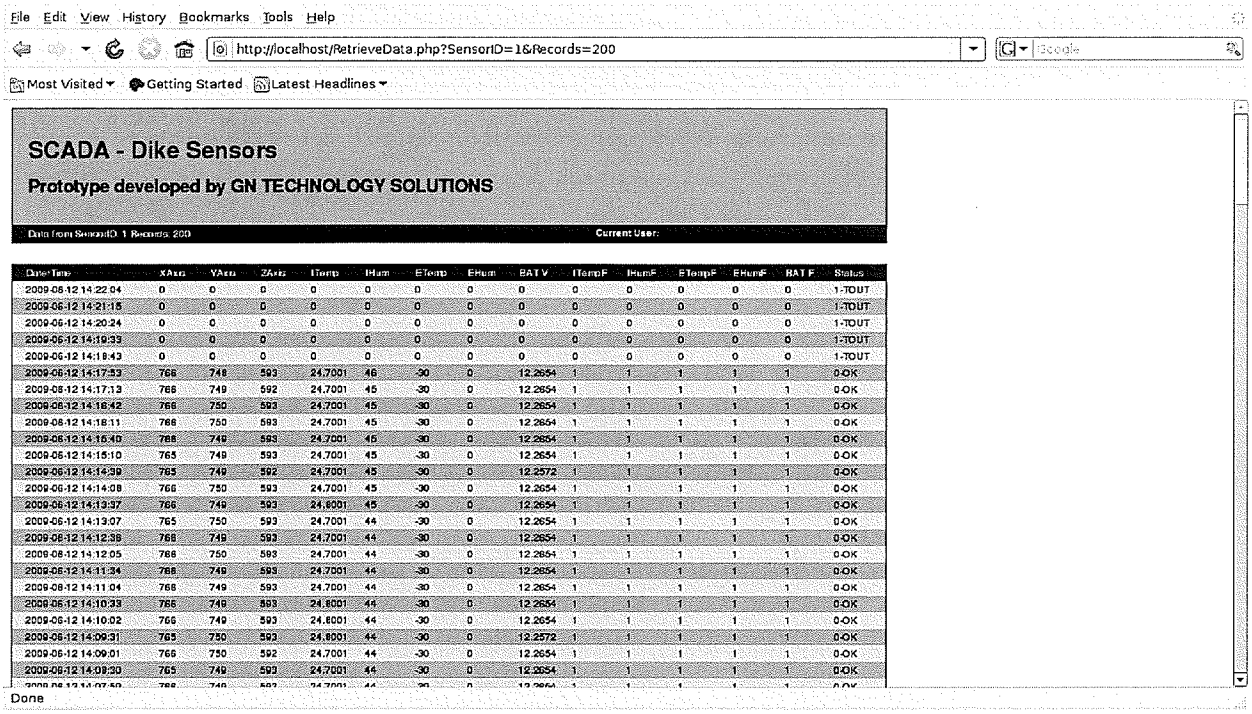


Figure 3.16: RetrieveData module using the records parameter

The second option is to retrieve the data using the sensor ID and a range of two days using YYYYMMDDHHMM format. (see Figure 3.17)

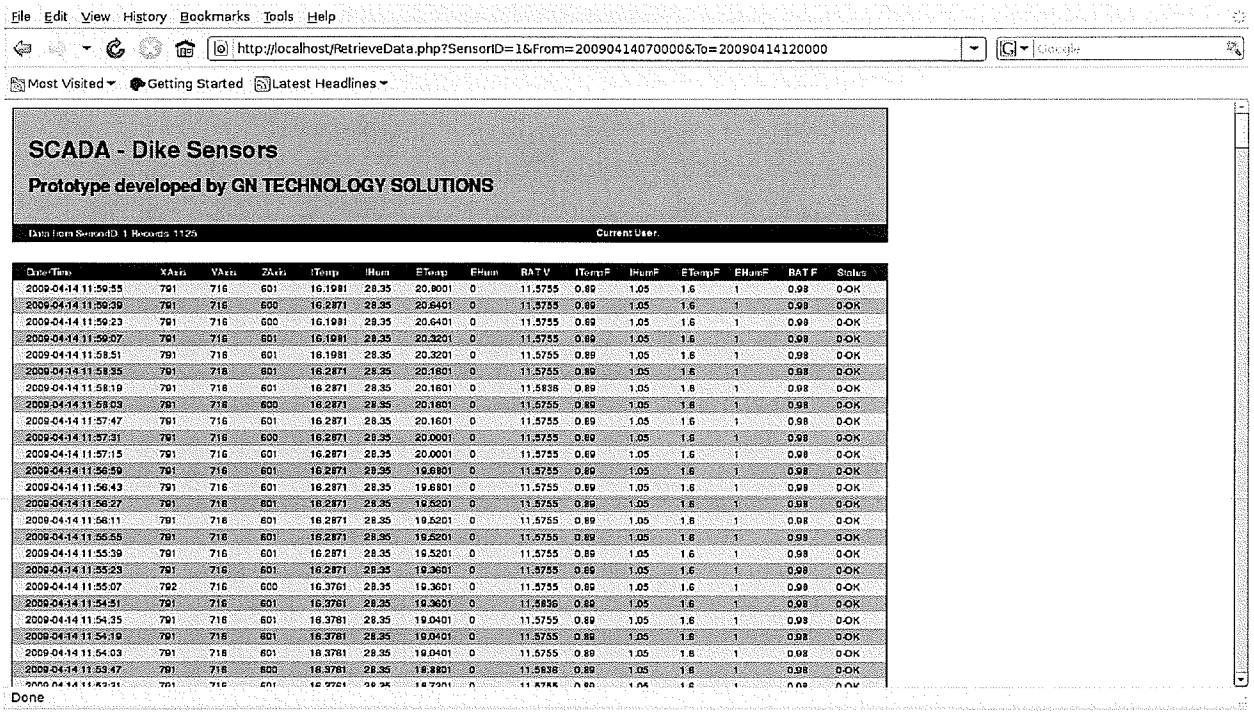


Figure 3.17: RetrieveData module using dates as a parameter

The last available module is a graph. There are three different options. The first parameter is the graph type. It could be 1, 2 or 3. The first graph - Type 1 - (see Figure 3.18) represents the data from the accelerometers X,Y and Z. This graph ideally would show the motion of the accelerometer. However, in this case the graph representing the last 50 readings did not show any movement because the sensor was static (no motion detected). The second graph - Type 2 - (see Figure 3.19) represents the data from the battery voltage. The last graph - Type 3 - (see Figure 3.20) represents the data from the internal and external temperature sensor. The X axis for the three graphs represents the number of sample. The system in the three cases is retrieving the last 50 readings.

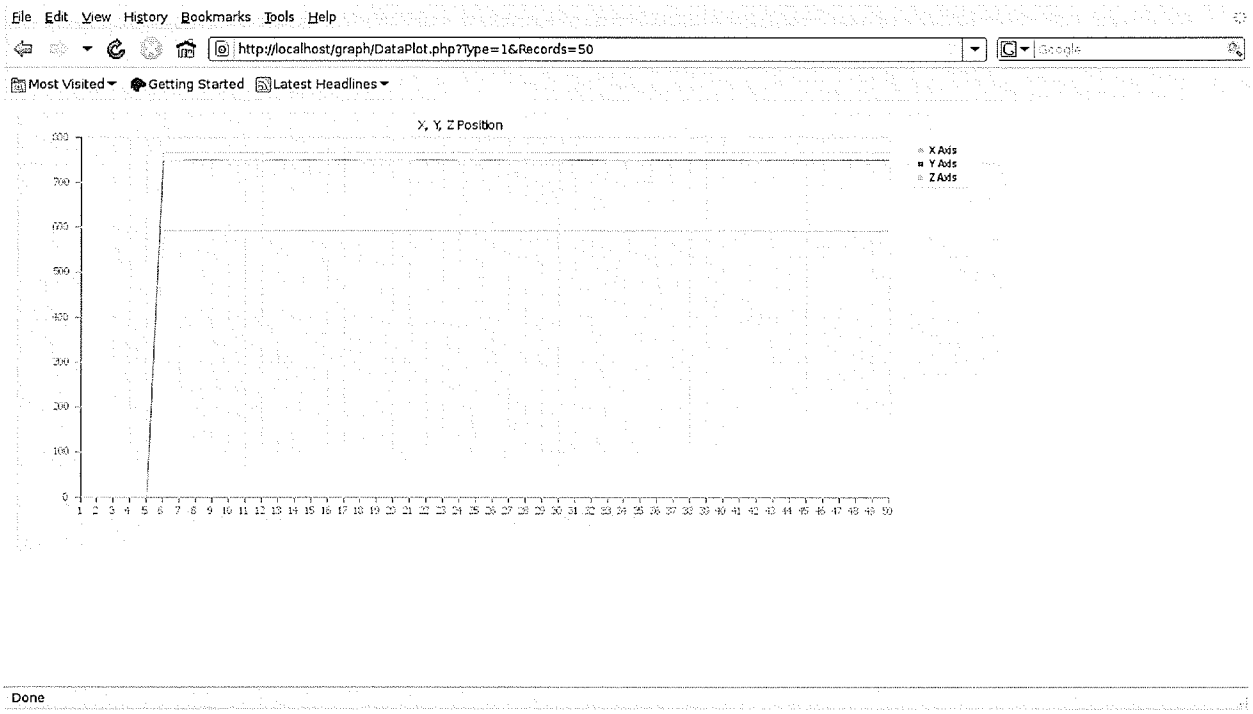


Figure 3.18: DataPlot module Type 1: Accelerometer position in X, Y and Z

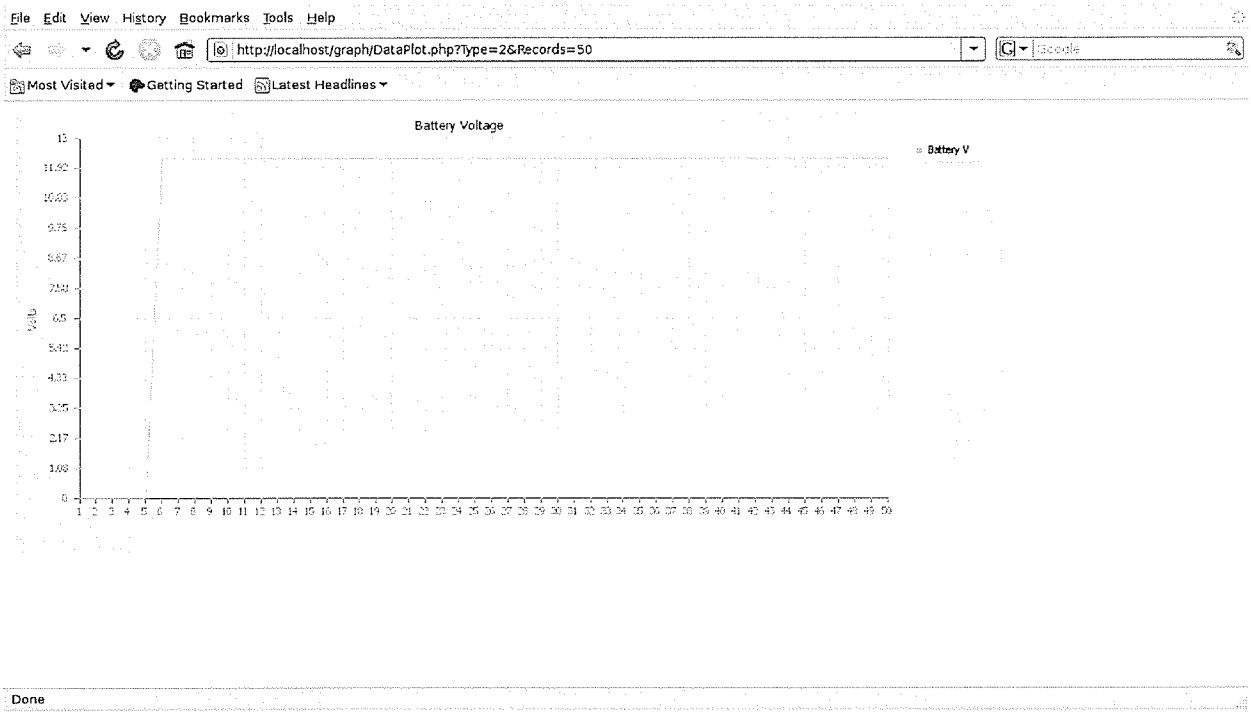


Figure 3.19: DataPlot module Type 2: Battery Voltage

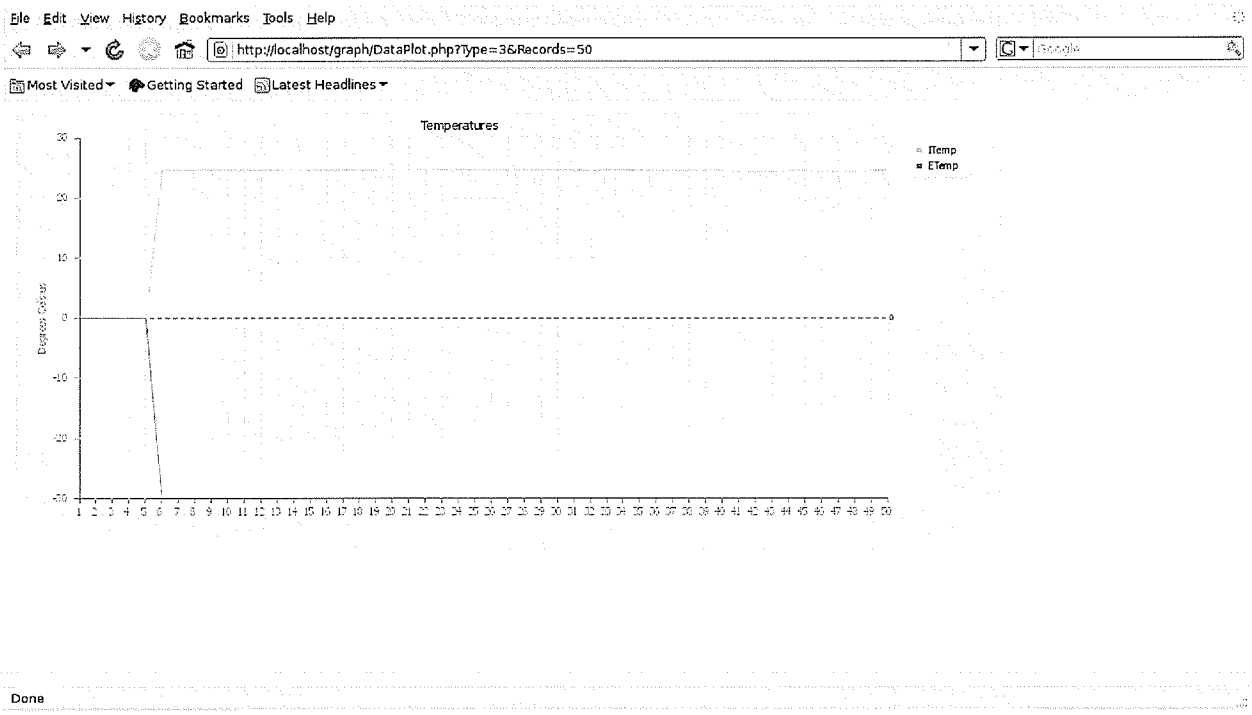


Figure 3.20: DataPlot module Type 3: Internal and External Temperature

3.4 Building the system

The objective was to build a wireless system capable of getting data from the sensors inside the sandbag dikes and to store the information for future analysis. The system had the main goal to store data from the sensors for analysis, although the following two questions had to be answered:

1) Is it possible to build a system using batteries only without an external charger? How will the battery work under low temperature exposure? (-5 Degrees Celsius)

2) Is the selected sensor sensitive enough to capture the sandbag motion? (the goal was to detect 1mm motion)

Also, one additional factor in the development of the system was trying to keep the price per individual sensor low to avoid excessive costs in large deployments.

The system was built starting with the sensors and the microcontroller. The microcontroller was bought with the evaluation board to make it easier to attach additional components. Then, the accelerometer was added and connected to the ADC - Analog to Digital Converter. An internal temperature sensor was also added and connected to the ADC to read the temperature and humidity conditions in the box installed inside the sandbag dike. Finally a voltage divider was installed to allow readings of the battery voltage.

One of the problems detected with the initial version of the system was the incompatibility between the accelerometer and the ADC. As a consequence, an operational amplifier was added to make the two elements compatible.

The enclosure has two external cables. One for the battery power and external temperature sensor and the second for data exchange with the RF module (RS-232 communication cable).

The first version of the application had the sensors and the microcontroller inside the enclosure with the external battery and external temperature sensor. The system build was started in February 2009. Because of the forecast of a big flood season, the system was initially modified to have it ready to be tested under real circumstances. The first version did not have the wireless communication implemented. The data cable was connected from the sensor the master station using serial RS-232 cable.

After the flood season ended, the system was removed and upgraded to make it wireless following the original idea. The only thing that changed was the network layer in the microcontroller on the software side, making the transition from wired to wireless only an incremental change.

The way that the data was presented was identical; that is, the user was not able to distinguish if the system was wired or wireless.

The actual system installed in the sandbag dike had one sensor only. The system was running from April 10th, 2009 to April 22nd, 2009. The system was able to run using two full battery charges. In other words, the system started with one battery fully charged and then another battery fully charged was installed on April.

The system was installed in a sandbag dike at a house located very close to University of Manitoba, along the Red River.

3.5 System Data Results Analysis

The results are divided in two categories. The first category is the information about the data sensor's behaviour. This is the information that is expected in a functional sandbag dike monitoring system. The second category is the data that helps understand the system performance and ways to improve it such as battery life and sensors sensitivity.

The results are presented under the Experiment 1 and Experiment 2 labels. The experiments are linked to the battery life. The time that the first battery was used is called Experiment 1 and the time that the second battery was used is called Experiment 2.

The accelerometer X oscillations (see Figure 3.21) shows almost no movement. Moreover, if comparing the experiment one with experiment two the results are similar. The same thing is noticed while verifying the data with accelerometer Y (see Figure 3.22) and accelerometer Z (see Figure 3.23).

The numbers shown on the Y axis on the Figures 3.21, 3.22 and 3.23 represents a digital voltage. In other words, the graphs represent the output of the accelerometers in raw values. One unit represents 1.5 mV. Based on the consideration that the output of the accelerometer can vary from 0 to 3.3 volts, a 4 unit difference in the worst case scenario represents 6mV. As a consequence the oscillation is not detectable. One can conclude that the accelerometer did not move or the movement was not detectable.

The data for the experiments 1 and 2 was taken at regular intervals of 15 seconds. The value shown is the maximum peak between the last value sent and the readings from the sensor.

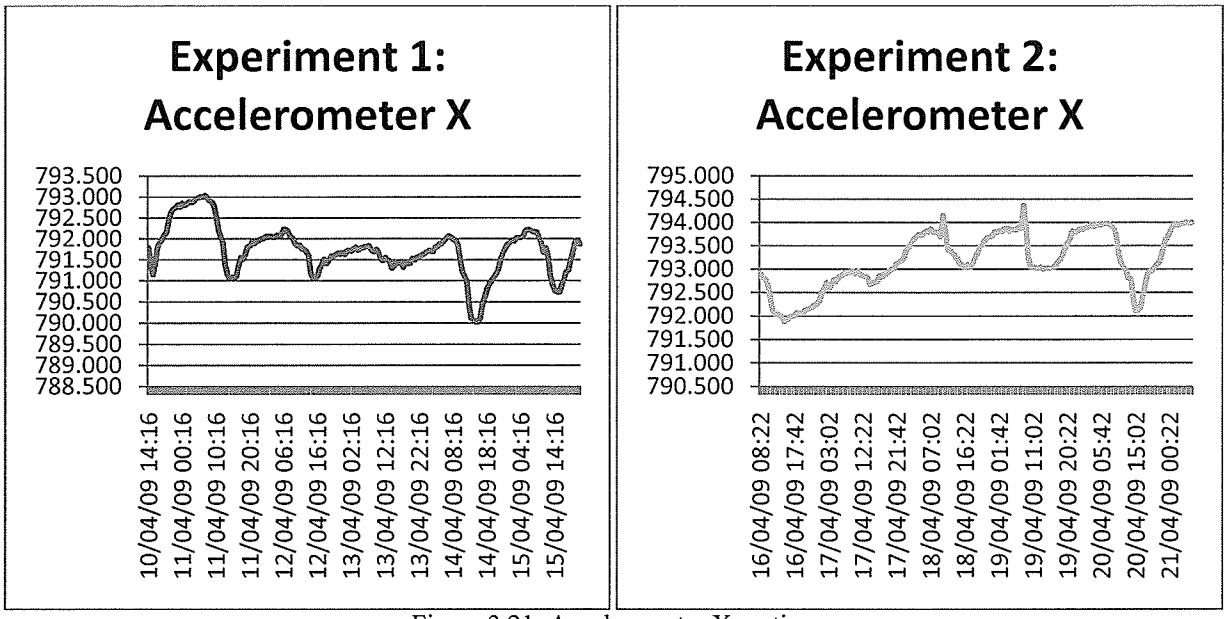


Figure 3.21: Accelerometer X motion

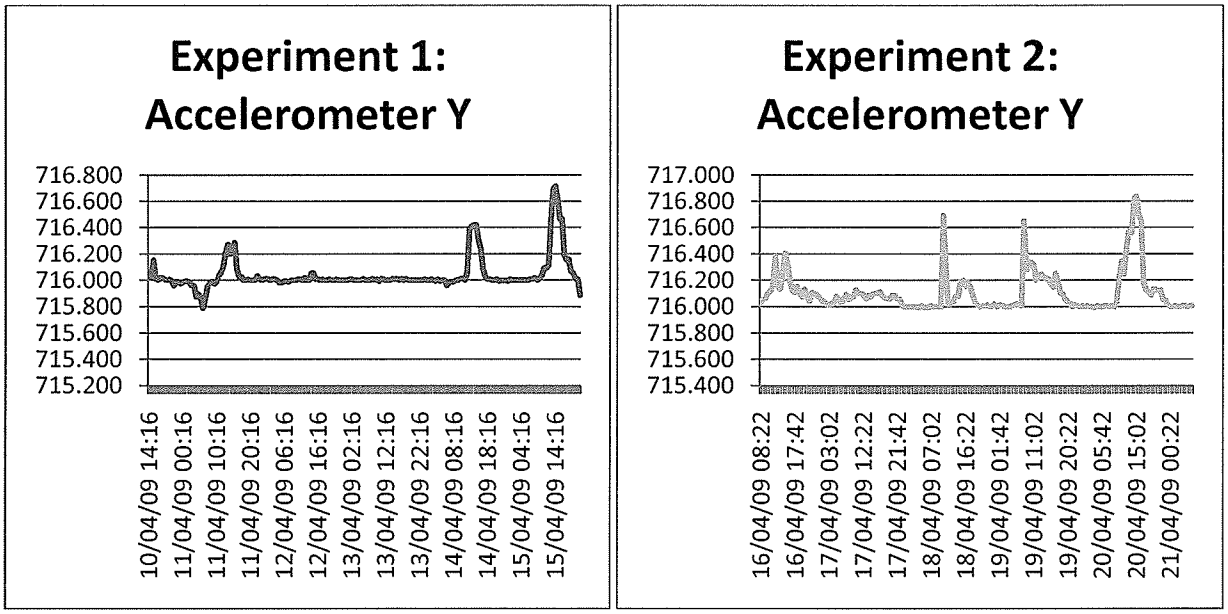


Figure 3.22: Accelerometer Y motion

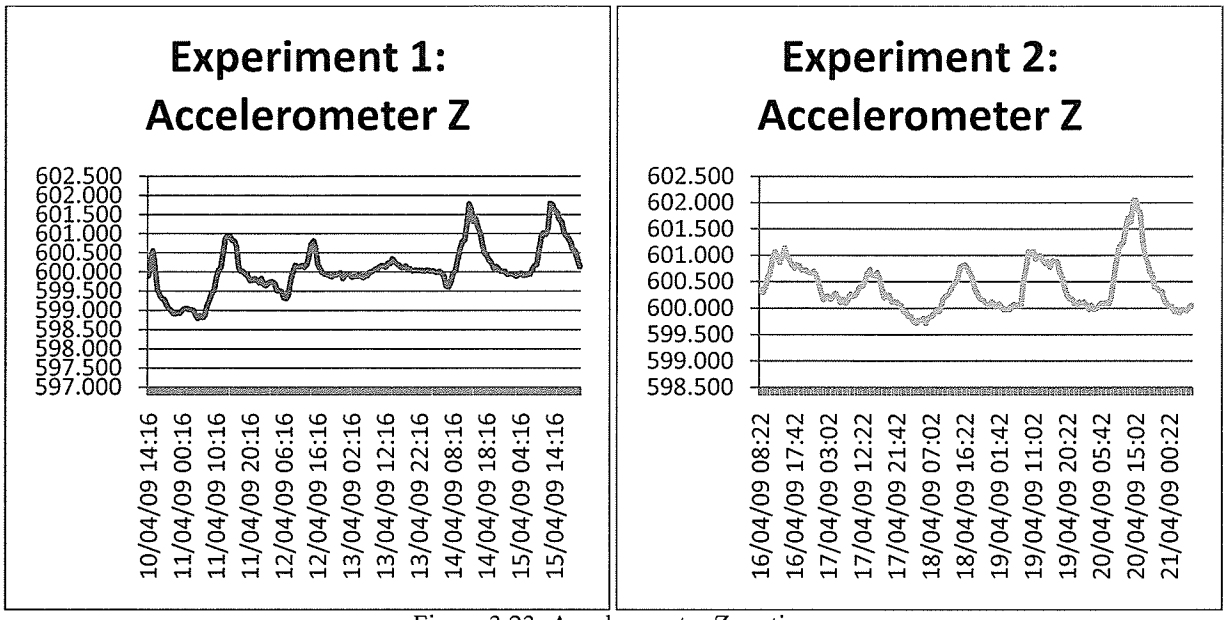


Figure 3.23: Accelerometer Z motion

The system was installed in a dike where the water level was never rose more than 12 inches relative to the base of the dike. The pressure of the water was not enough to show any movements. However, it will be useful to know if there is some water impacting the dike. As a consequence, further testing is required to upgrade the sensors to make them more sensitive to capture this information.

The sensor was installed inside one sand bag in the interior side of the dike (where the water was not present). An idea developed for future versions is to attach an additional support bar to the sensor and insert it in the middle of the sandbag dike to make it more sensitive to the whole dike movement instead of the last layer only (see Figure 3.24)

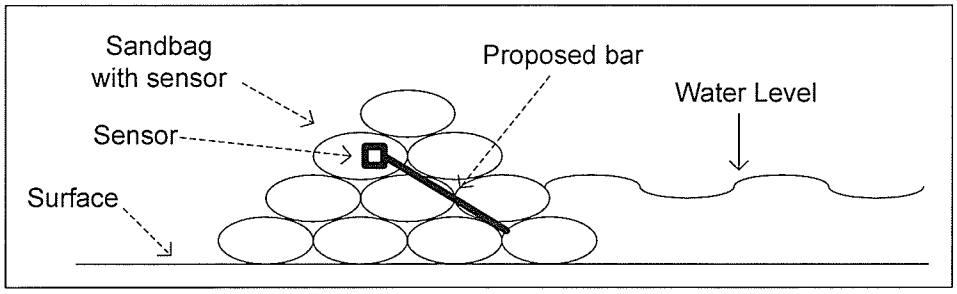


Figure 3.24: Additional support bar attached to the sensor to increase sensor sensibility

The second set of results focused on the battery life based and the temperatures and environmental conditions. Unfortunately, the temperature sensors were not calibrated properly, but the values shown in the graphs are reasonable relative estimations. Additional instrumentation was used to calibrate the system on the daily visits to the dike while capturing the data.

The power provided by the battery was used by different devices in the system. The microcontroller was consuming 250mA @ 3.3V, the Temperature and Humidity sensors where consuming 2.4 mA @ 3.3V, the accelerometer was consuming 350 uA @ 3.3V and the Evaluation board 100mA @ 3.3V. The total power consumption was estimated to be 1.16W (97mA @ 12V). The battery used is a 12V Panasonic with 17Ah. Using a regular discharge of 97mA @ 12V with 25 degree Celsius should make the battery last 175 hours or 7.3 days [18]. However, after the tests (see Figure 3.25) the battery was used for 110 hours or 4.58 days. This is 62% less than the manufacturer specifications. The main reason for the difference is the exterior low temperature.

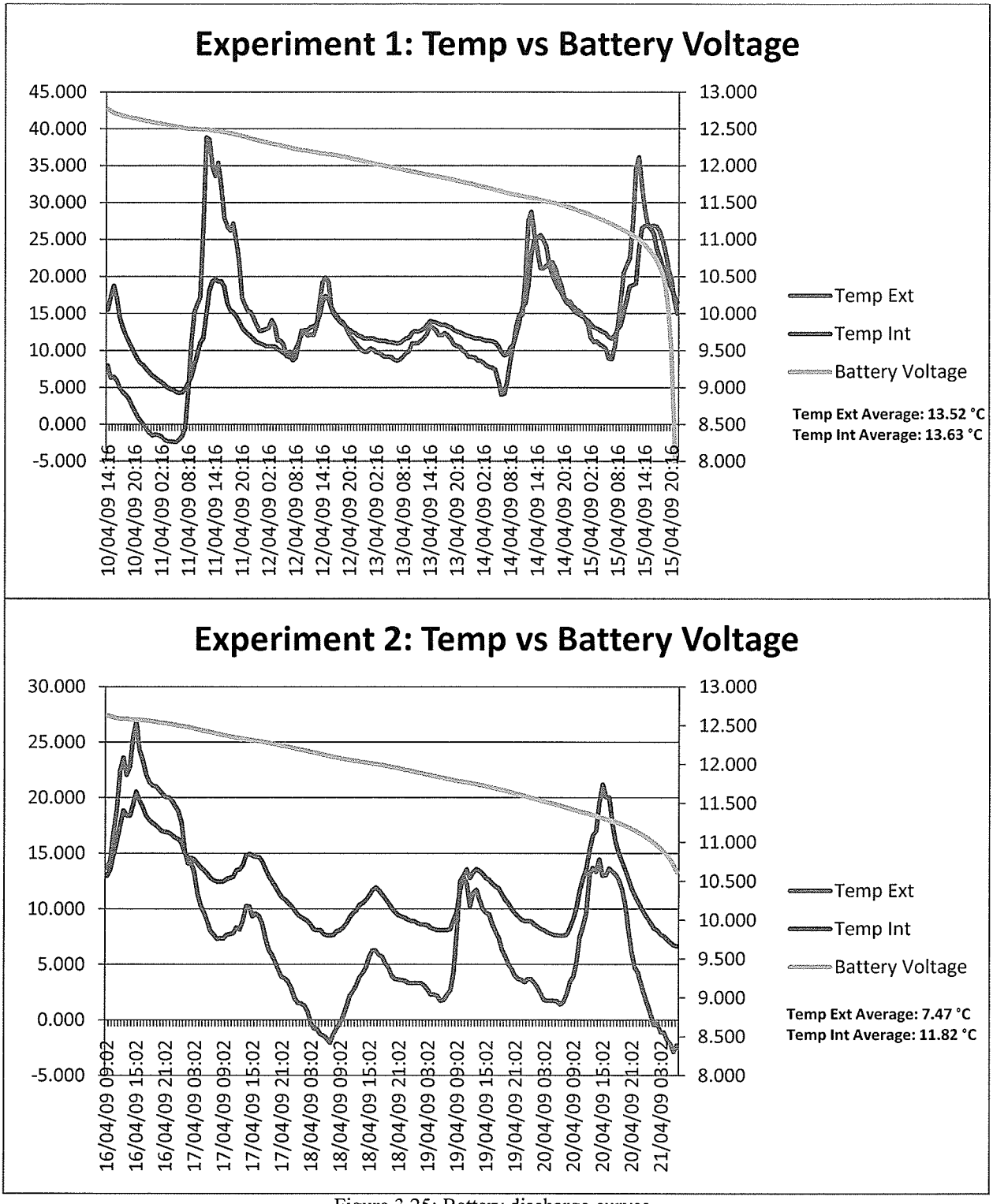


Figure 3.25: Battery discharge curves

3.6 Recommendations for future versions

The following is a list of the recommendations to follow in future versions of the sandbag dike SCADA system:

a) Complete the system redundancy using the long haul link between the MS and the repeater module at the end of the dike.

b) Test the system with different accelerometer/sensors to test sensitivities.

c) Additional research for affordable battery charger alternatives such as solar energy or wind energy.

d) Test the system attaching a support bar attached to the sensor inside the sandbag to test system sensitivity. Basically, a better sensor is required to monitor the stability of a sandbag dike.

e) More statistical approach and coordinating response analysis from a larger number of sensors.

3.7 Conclusions

The system was completed and tested under the 2009 flood season in Winnipeg. The system exceeded the original budget of $11,000, but the final deliverable is a product that is reasonable in price for that type of application.

One of the objectives of the project was to create a solution that was cost-efficient. The component that will impact on the final price of the whole system is the individual price per sensor installed inside the sandbag. The cost effectiveness is directly related to the individual cost per sensor. The final price per sensor with the battery and the communication channel (RF module) is below $ 300. The system proposed does not have a solar panel to charge the battery. This is a significant saving considering that a solar panel cost the same as the whole system. However, additional testing is required to understand how long the system will work if sending

the data at larger interval rates (less battery consumption). The proposed solution contemplates replacing the uncharged batteries with charged batteries every 15 days.

With respect to technical considerations, the system was installed and successfully tested. The data collected shows that the sensors were working properly but more sensitivity is required because the system was not capable to detect when the sensor moved small quantities (less than 2 mm). Also, the data collected shows that the battery charge decreases significantly with low temperatures. As anticipated, multiple batteries or a battery charger is required.

The GUI can be improved in at least two ways. The first thing is to add AJAX - Asynchronous JavaScript and XML - technology to refresh the sensor values on the screen without having to refresh the web page. AJAX separates the presentation from the content. AJAX was a term used by Jesse James Garrett in February 2005 in his essay: "AJAX: A New Approach to Web Applications". The beauty of AJAX is that combines existing technology: (X)HTML, JavaScript and XML. The first application to use AJAX was Google with its Google maps service [19]. The second thing is to create a system using SOA to create better modularity between the communication engine, database engine and the presentation layer. SOA stands for Services-Oriented Architecture. A services-oriented architecture consists of distributed services that communicate with each other. The communication medium is a web server; the messages that are exchanged between the services make use of the transport mechanism of the Web technologies. The most used underlying protocol is *HTTP*. In the communication between services, there are service requests and service responses. Service requests can be URLs using *HTTP Get* or XML documents using *HTTP Post*. Service responses are always XML documents [20]. The advantage of SOA is that reuse is heavily encouraged reducing the amount of customizing each newly deployed system would otherwise require.

The system has to be tested with a dike with higher level of water to understand the behaviour of the sensor under higher water levels and under wave action.

Finally, the present prototype showed an implementation of a SCADA framework with remote sensors that are independent of the media used to interconnect them. The system was initially tested with the dikes using RS-232 connectivity and then it was updated to wireless. Data was collected, and then processed for analysis. Useful information was obtained regarding battery life and sensors sensitivity.

Chapter 4: The Development of a SCADA system for an RTLS application

4.1 Introduction

The chapter describes the principles and techniques used to created an RTLS - Real Time Location System - (see Figure 4.1) based on SCADA technology. The prototype developed follows the standard concepts reviewed in Chapter 2. Moreover, the RTLS system presented has a similar architecture to the one presented for the sandbag dikes (common SCADA framework). This flexibility highlights the initial scope of the present work discussed in Chapter 1 regarding a system that is sensor agnostic, flexible and scalable.

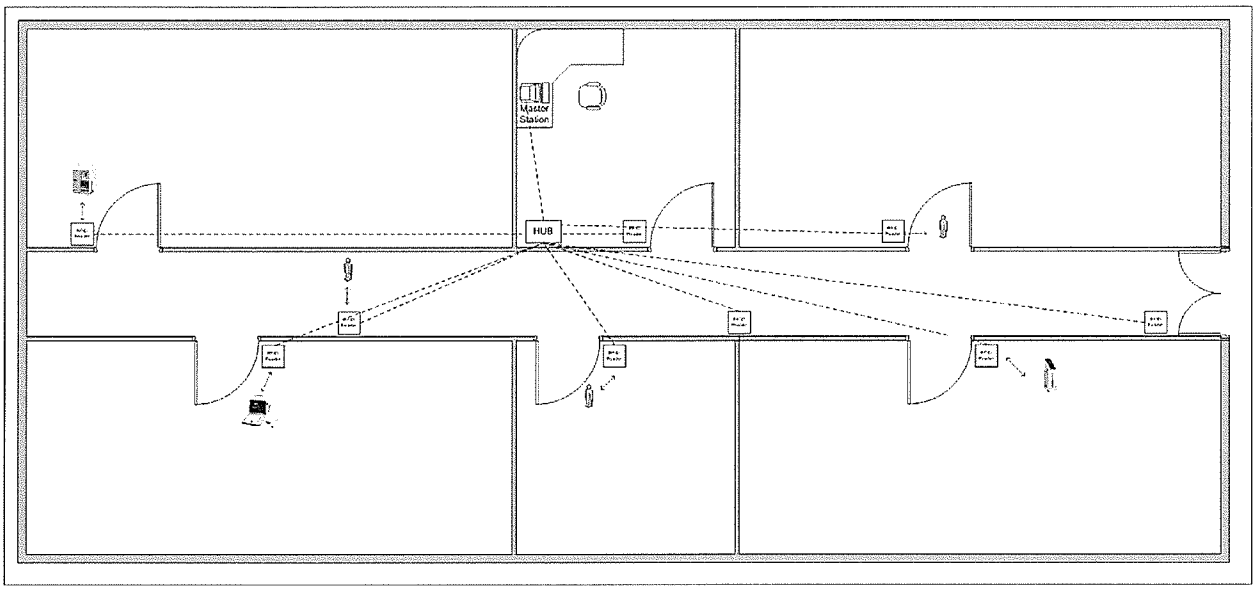


Figure 4.1: RTLS system overview

As previous stated, SCADA solutions have three elements: the remote sensors, the communication channel and a master station. As a consequence, SCADA is a good fit to solve the problem presented in this chapter. In this instance, the sensors installed in the building (remote

sensors) are collecting data about people and equipment location. Then, the data is transmitted (communication channel) to the master station in order to present it to the user.

The application considered here is an RTLS in a nursing home to track the location of patients and equipment. Because there are people with dementia, it is important to know they are protected and inside the building. Another important aspect of tracking is that some patients are not allowed to go into certain areas. It is also important to know where equipment is. Sometimes, it takes a person a long time to find a piece of equipment without any type of tracking in place.

The RTLS system proposed suggests that each patient or piece of equipment to be tracked with a tag that will be read from the sensors installed around the building to detect their proximity.

This project is a modification of the previous SCADA development described on Chapter 3. The objective of this chapter is also to illustrates that a common framework for SCADA that is sensor agnostic is possible. The same core code for the communication module was used, but a different GUI was created. One of the modifications to the communication module is that the RTLS system is expecting information from the sensors (interruption mode) instead of having to request it (polling mode). This application uses the RTU already built-in the RFID readers used to detect the tags that people are carrying.

This system uses passive RFID - Radio Frequency Identification - technology to create an RTLS for indoor tracking for the location of people, equipment or inventory. The information from the readers is sent to a master station and then presented to the user within a floor plan illustration.

4.2 RTLS

Real Time Location Systems or RTLS are systems capable of tracking people, equipment or inventory wherever they are. For example, you can track a container with goods from China to Canada in real time using GPS combined with a communication means. The RTLS SCADA was developed for indoor use only for this particular project. The basic idea is to create an indoor positioning or tracking system. A typical RTLS has five components: the tag (the element or person to be found), the location sensor (sensor that detects the tag), the location engine (software that communicates with the location sensor to determine tag location), the middleware (software that interacts with the tag, sensor and location engine to provide value to the application), and the application (end-user application that shows the location) [21].

RTLS requires multiple sensors installed depending on the sensors range and level of desired granularity. There are multiple possible implementations of RTLS for indoor use. Possible technologies considered were: infrared, ultrasound, passive RFID and active RFID [21]. The SCADA system prototyped was intended to be installed in nursing homes to track patients and/or equipment.

The first alternative considered was to utilize infrared technology. Three inconveniences were 1) the need of a battery on each tag, 2) the weight of the tags was excessive for some of the elderly with health problems and, 3) the tag has to be pointed at the sensor (like a remote control). The second technology was the ultrasound. It was challenging to find affordable ultrasound sensors to track people. There was also a problem to identify where to buy the parts and some issues with the ranges from the tags to the sensors. The last two technologies analyzed were active and passive RFID. Both technologies were seriously considered but a passive RFID was selected.

The reasons were: 1) passive readers were available for the prototype without incurring extra costs and 2) the affordability of passive tags. Table 4.1 has a summary of the different technologies presented.

Features	Infrared	Ultrasound	Passive RFID	Active RFID
Tag has to point to the receiver ?	Yes	No	No	No
Range	Short	Short-Medium	Short-Medium	Short-Long
Battery needed in tag	Yes	Yes	No	Yes
Battery life	Short	Long	N/A	Long
Many suppliers ?	Yes	No	Yes	Yes
Price	Affordable	N/A	Expensive readers	Expensive tags
Possibility to buy parts only	Yes	No	Yes	Yes
Heavy tags	Yes	Yes	No	No

Table 4.1: Comparison between RTLS technologies

Part of the thesis objective is to make prototyping SCADA as sensor agnostic as possible. As such an RTLS SCADA could have easily been developed for any tracking technology.

4.3 Principles of Passive RFID technology

The passive RFID technology was the one selected to create the RTLS. There are four different frequencies commonly available for passive RFID systems. The frequencies are 125 kHz, 13.56 Mhz, 868/915 Mhz and 2.45 Ghz based on ISO 15693, ISO 18000 and ISO 14223 standards [22]. The technology selected was the 125 kHz system as it is one of the most mature technologies and because the sensors and tags affordability and diversity.

The passive RFID technology has two basic components. The first component is the transponder or tag which is located on the object to be identified, and the reader which (depending upon technology) can be used to read or write information from/to the tag [22].

Each tag will contains information stored in a memory. The memory can be from few bytes up to 4 Kbytes. For this project, the cards used have 10 bytes as ID data. The data is stored in an EEPROM in the tag. Each tag has a unique identification. In other words, the bytes in each tag will be used to differentiate one tag from other. Moreover, each tag will be linked to person at the software level allowing identification of individuals.

The power supply to the data-carrying device and the data exchange between the data-carrying device and the reader are achieved using electromagnetic fields [22]. The passive system prototyped is similar to an air core transformer.

When the tag containing the data approaches the range of the reader, it will be energized with the electromagnetic field received from the reader. After that, the tag will "effectively" send its data back to the reader (see Figure 4.2).

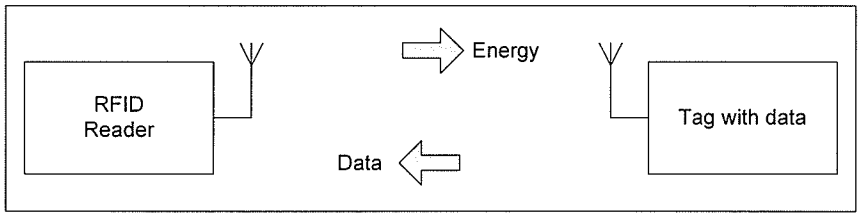


Figure 4.2: RFID Reader and Tag operation (schematics) [22]

This level of detail is not technically exhaustive but sufficient from the perspective of the SCADA developer.

4.4 RFID Passive Reader Selected

The passive reader used is the GP90A from Promag (see Figure 4.3). The reader has a range of 90 centimetres with serial port outputs to send the data received from the tags.

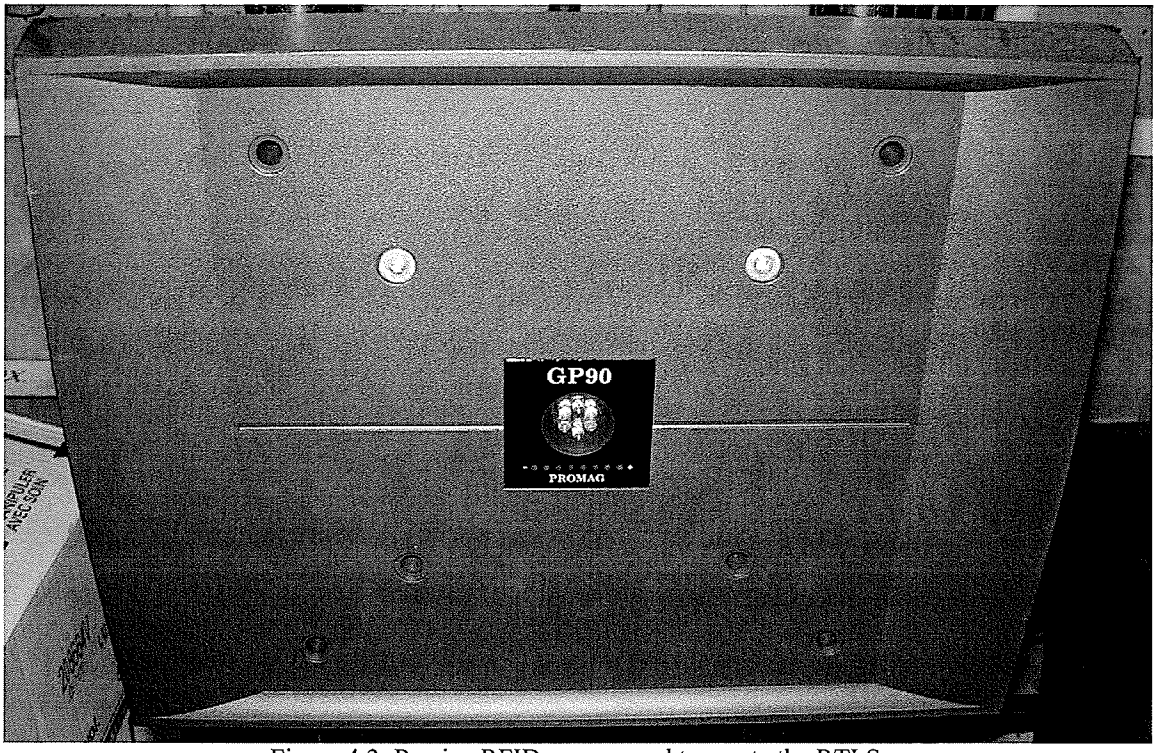


Figure 4.3: Passive RFID sensor used to create the RTLS

The reader was mounted on a plastic pipe structure (see Figure 4.4) to test the best height to install it. The objective was to make the reader capable of reading tags carried by personnel. The best results were obtained fixing it at 95 cm from the floor. Different people were asked to walk with different tags in different locations such as shirt pockets, jean pockets and in the hand. A tag in a hand would emulate a patient with wrist strip. As deployed in a nursing home, a more reasonable location for readers would be on a ceiling and doorways. The readers used here were for demonstration and prototyping purposes. Although not addressed here, SCADA allows for control and in certain instances could be used to control doors within certain institution. The SCADA systems approach developed here was not considered the actual application but again is an attempt to create a SCADA framework across various applications, similar in rationalization as being sensor agnostic.

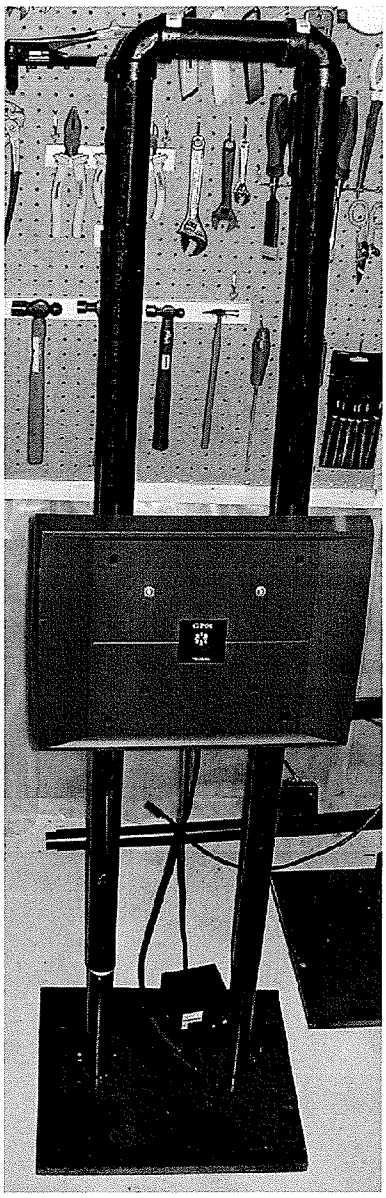


Figure 4.4: Reader mounted on a plastic pipe structure

Different tags were tested, but the only ones with a range that matched the manufacturer specifications were the clam shell type tags (see Figure 4.5). These tags have a significant coil able to capture the weak EM field at a maximum distance of about 1.5 m.

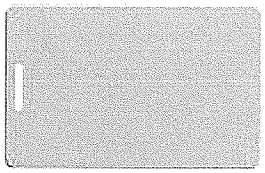


Figure 4.5: Passive RFID card with 10 bytes of memory

4.5 RTLS Architecture

The RTLS SCADA designed has three basic elements (see Figure 4.6). The first element is the RFID passive readers. The second element is the communication channel between the readers and the master station. The last and third element is the software capable of reading data from the sensors and presenting it to the user using a GUI interface.

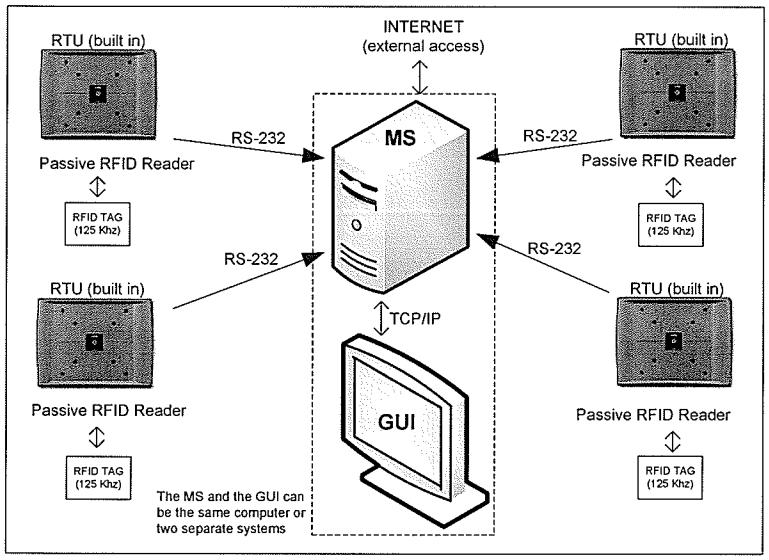


Figure 4.6: RTLS Architecture

This system works on an interruption mode. In other words, there are no requests from the master station to the remote sensor. Each time a sensor reads a tag, it will forward the data from the tag to the master station for processing (see Figure 4.7).

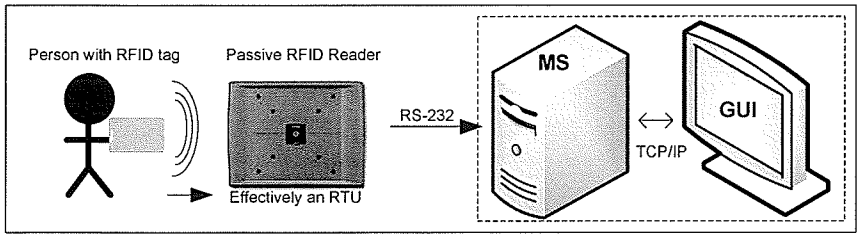


Figure 4.7: RTLS Interruption Mode

The principal requirement of this RTLS SCADA system is to estimate the last known time stamp location of a person in a building.

4.5.1 RFID Passive Reader Installation

The system was built with four RFID passive reader sensors. The sensors were installed in different locations in a room during a demo done in a nursing home located in Winnipeg, MB. Each sensor represents a Room numbered from 1 to 4. The sensors were installed few meters from one other, hanging from a plastic pipe structure (see Figure 4.4).

After the demonstration, the system was moved to a simulated environment where different people were asked to walk in an area covered by the four readers at different paces. The volunteers walked in front of the sensors for short period of times (less than 5 minutes) for an hour. The intentions were to understand if the system was reliable and at the same time the best location for the readers. Although coarse, this process aided in establishing the functionality of the system.

4.5.2 Communication Channels

The communication between the sensors and the master station was done using serial communication (RS-232). The master station used four USB to serial connectors emulating different COM ports.

4.5.3 Software Application

The software application was developed in JAVA reusing much of the code from the sandbag dike SCADA system. The application waits for data to arrive to the system from the serial ports. Each serial port as a unique ID. Each port is associated with a room number. After the data arrives at the port, the system associates the person in the respective room.

The system was tested with four tags. Each tag has a unique ID that represents a person. Each time the system reads information from a tag, the GUI (see Figure 4.8) will show the person in the correct location. Each tag is associated with a different color. In other words, each person is represented graphically using a different color (red, green, blue and purple). The four colors corresponding to particular individuals are for the prototype only. Future versions should have a dot to represent people or equipment. For example, different colors could represent different elements such as nurses, patients, visitors and medical equipment. When the user moves the mouse on an element, additional information will be displayed. Also, the system should contemplate an option that a particular element could be search. If found, the dot will flash indicating the last known location.

After the system gets the information from the sensor, the system adds the date and time and creates a representation of the person with a unique color showing the tag ID, date and time. In future versions, the tag ID can be linked to the person's name from a database. The database engine was not used for this application but it was implemented on the dike project. This proves the flexibility of the proposed SCADA framework.

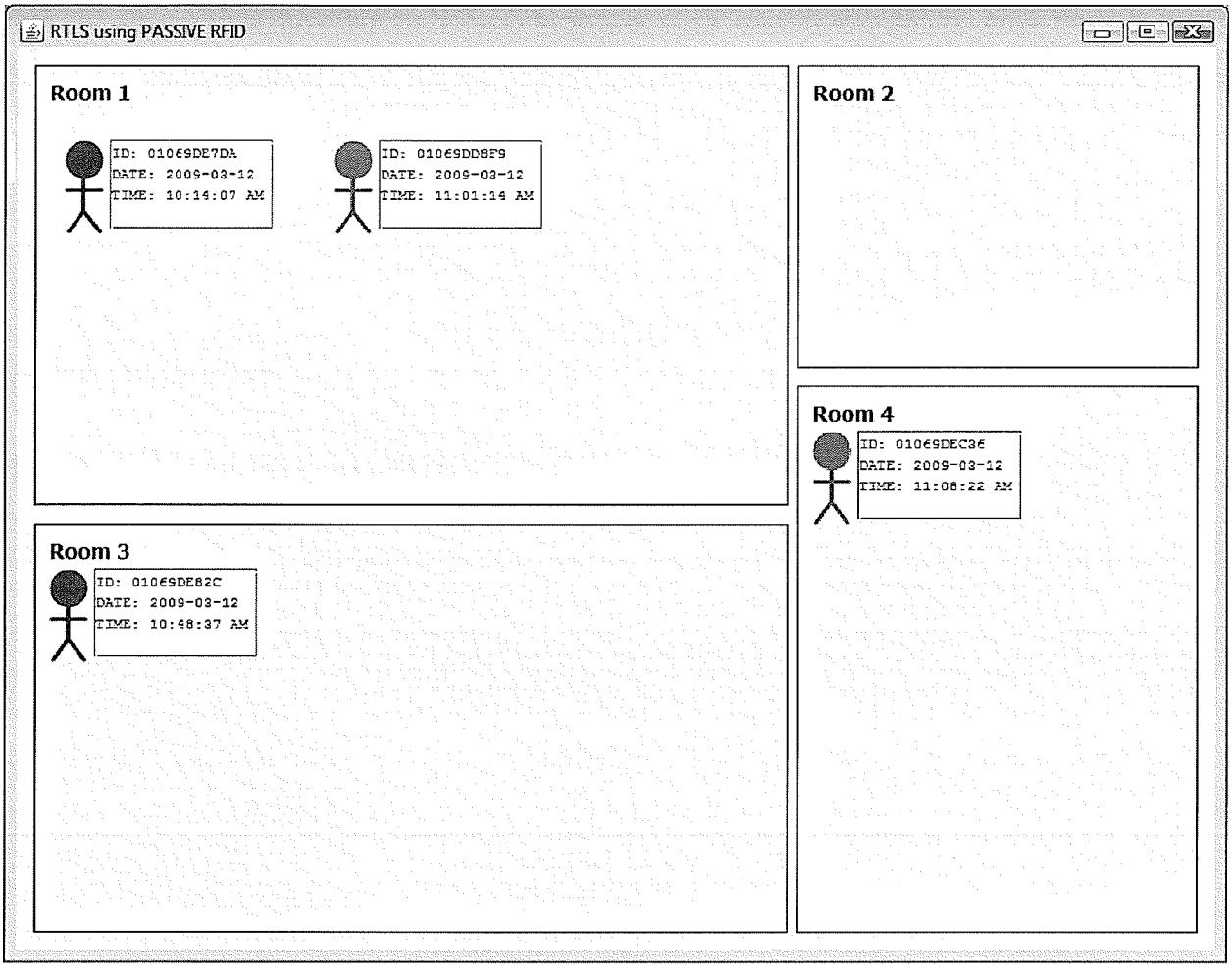


Figure 4.8: GUI interface for the RTLS

4.6 Passive RFID sensors challenges

The passive RFID readers are not capable of reading data 100% of the time. The signal may not be always captured by the sensor each time a person passes by the reader. As a consequence, the technology might not be reliable enough to be installed in areas where the system has to lock a door to stop a person to walk outside the building. This degree of uncertainty is problematic with most RFID systems. As such, considerable attention has to be employed to mitigate against missing data while tracking movement, particularly in high risk environments.

Another challenge with the passive RFID technology is the distance between readers. If the readers are too close one to another, none of them will read a tag or they will read sporadically. For example: the sensor maximum distance range is approximately 90 cm. However, some tests showed that the distance can increase up to 1.5 meters. If the system is installed in a corridor 2.5 meters wide and the readers are installed facing one another, the chances to get a reading from a tag may be reduced if the tag is in the middle. In other words, the system may fail to read the tags in overlapping ranges.

This chapter and the previous one are two very different applications of SCADA systems but with the same framework. There is a communication module responsible for communicating with the devices and a GUI interface in both cases. However, the sandbag dike monitoring system has a database engine in the background to store readings for processing and analysis. This difference is part of what the proposed framework is trying to prove: how to create a flexible SCADA system of varying capability that is at the same time sensor agnostic. The communication channel in the sandbag dike monitoring system was initially done with wires - serial communication - and then upgraded to wireless technology. The RTLS SCADA system used the same serial communication that was first implemented in the sandbag dikes. The last element that both systems share is the sensor component. The sandbag dike monitoring system has an accelerometer, temperature and humidity sensor connected to a microprocessor and then linked to the communication module. In the same way the RFID readers have a microprocessor that controls the readings from the tags, and then the data is sent using the communication channel.

4.7 Recommendations for Future versions

The following list contains recommendations for future versions of the RFID-RTLS system:

a) Test the system using active RTLS technology.

b) Add hardware on the readers to make it wireless (reducing installation costs).

c) Include database capabilities to store people's movement reusing the module used in the sandbag monitoring SCADA system.

d) Create set of rules for alarms. For example: if the system is installed in a nursing home, an alarm has to be triggered if a person is walking into an unauthorized area (application tailoring).

4.8 Conclusions

The objective of this chapter was to build a system capable of tracking people and to show the last known location on a GUI based on a common SCADA framework.

The system built was based on the project developed in the previous project with minimal changes, illustrating that the principle of a common SCADA framework is possible. SCADA is a technology that can tend to a common framework where this framework has to evolve from exposure to different situations and improved to handle more unknown situations.

The RTLS system was built following the standards and principles utilized by SCADA. This approach is to bring industry standards such as SCADA into the health sector.

The system was tested for few days with very good results. Different people walked in front of the sensors with the tags in different body locations and the readers were able to detect them. However, the system is not 100% reliable because of limitations in the passive RFID technology.

Although the SCADA sandbag monitoring system at the RTLS system are completely different applications domains, much of the middle layers of the SCADA are similar. A concentrated effort should be made to develop standard middleware in this regard similar to the advent of web based applications, most of which are developed using standardized protocols, emanating from the TCP/IP protocol suite.

The SCADA from this chapter as well as the last, deals with developing SCADA where there is considerable uncertainty in the data and its interpretation. Future SCADA frameworks will have to address this issue and allow for the seamless integration of statistical analysis tools.

This project showed how is possible to take the project presented on the previous chapter and after small modifications reuse it for a different application. It illustrated the reusability of the major components and demonstrated a step forward that is required to create a more universal framework for SCADA solutions.

Chapter 5: Potential evolution: SCADA framework for an active RFID RTLS application

5.1 Introduction

The present Chapter presents ideas to build a modern common-architecture SCADA system, based on experiences gained from the previous two prototype projects. Chapter 5 briefly describes an active tag RFID RTLS SCADA application. The system extends both the SCADA framework used for the sandbag dike monitoring system as well as the passive tag RFID RTLS system. The active tag (battery powered) RFID recommended operates in a frequency of 433 MHz [24] in the present case. This system offers considerable reliability compared to a passive tag system while supporting the wireless communication between the master station and the RTUs similar to the one used in the sandbag monitoring project. Although modifications are required, the versatility of the concept of a more universal SCADA framework is demonstrated.

5.2 Active RFID

The active RFID technology differs from the passive - explained in the previous chapter- in the fact that the tag has a battery built in. The primary advantages are their reading range - up to 100 meters - and reliability. The tags tend to be more reliable because they do not need a continuous radio signal to power their electronics [25]. The active RFID readers tend to be more affordable than the passive ones, but the tags are more expensive.

5.3 Combining the sandbag monitoring system and the RTLS to create a new SCADA RTLS framework

This section will briefly describes how the two previous SCADA projects presented in Chapters 3 and 4 can be combined to create a new application using the best of each one with minor changes, advancing the principle of a unique and flexible SCADA framework.

The Figure 5.1 presents a combination of a possible evolution combining aspects of the SCADA presented in previous chapters.

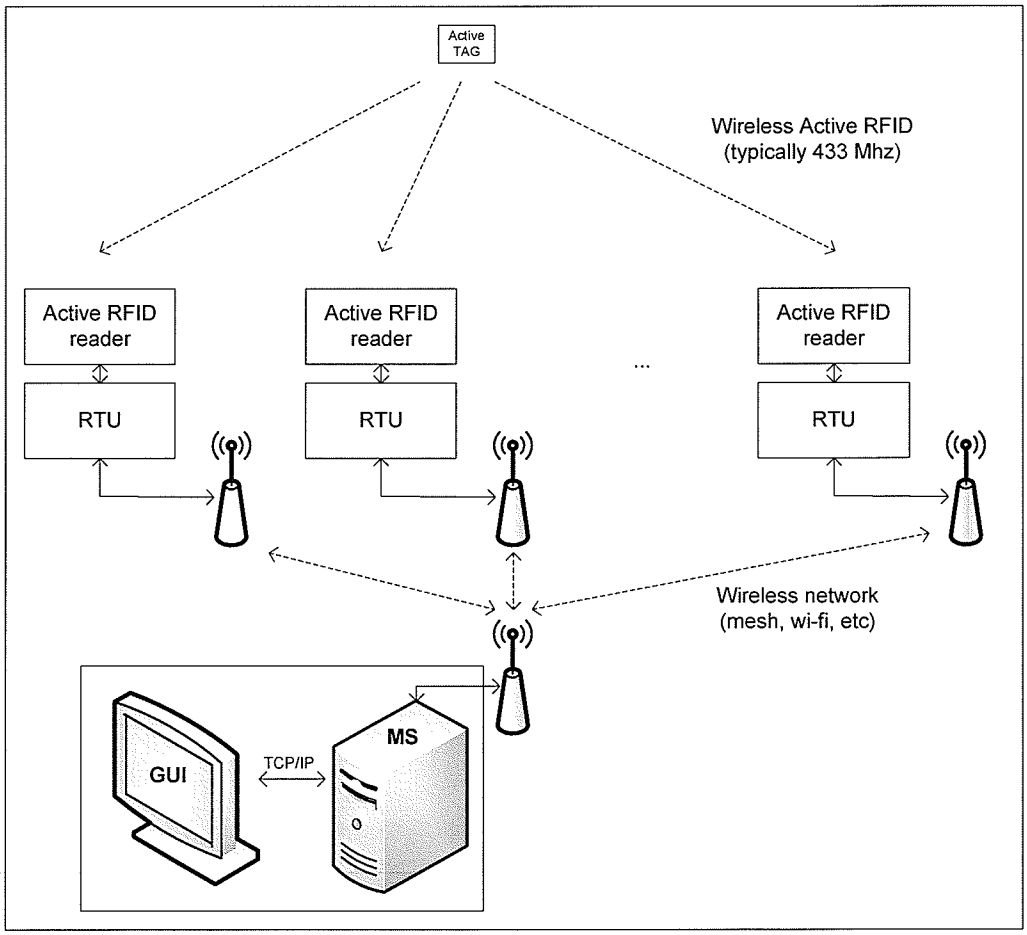


Figure 5.1: RTLS solutions combining SCADA projects from chapter 3 and 4

The idea is to create an RTLS system using the active tags and make it wireless. The RFID readers presented in Chapter 4 can be replaced with the active RFID readers using the same software on the MS. Another change is the replacement of the RS-232 communication channel with the wireless principles used in Chapter 3. The firmware developed for the sensors should be reused to interact with the active RFID readers and to add the wireless connectivity with the master station. Another alternative is to add WI-FI capabilities into the RTU to make the RTLS system interact with existing wireless infrastructure to reduce communication costs.

There are other modules that would be required to improve the accuracy of the active RFID RTLS. These include trajectory estimation using techniques based on conditional probability, e.g. Kalman filters [26]. Also location estimation can be improved through signal processing and radio triangulation, techniques not amenable to passive RFID. It is important that these types of modules can be incorporated into the SCADA framework.

5.4 Conclusions

This chapter presented a potential evolution of the RTLS SCADA presented in Chapter 4 combined with the wireless principles described in Chapter 3. In order to create the new SCADA, the modules that were created for the previous solutions can be reused with minimal changes, reinforcing the principles of the present research work described on Chapter 1. The reusability advances the notion that a common framework for SCADA solutions is possible. The framework includes two basic components: the software in the master station and the firmware in the microcontroller attached to the remote equipment that also acts as an RTU.

Chapter 6: Conclusions and Future work

6.1 Project Conclusions

The projects presented in Chapter 3: SCADA System for Monitoring Sandbag Dikes and Chapter 4: SCADA for RTLS applications as well as the extended RTLS system of Chapter 5, shared the same basic architecture (SCADA framework). However, the features are not the same.

Figure 6.1 shows the architecture designed to create SCADA systems independent of specific sensors and communication channels, adaptable across varied application domains. The design also presents an affordable solution using standards, exploiting reuse, and using off-the-shelf sub systems where possible.

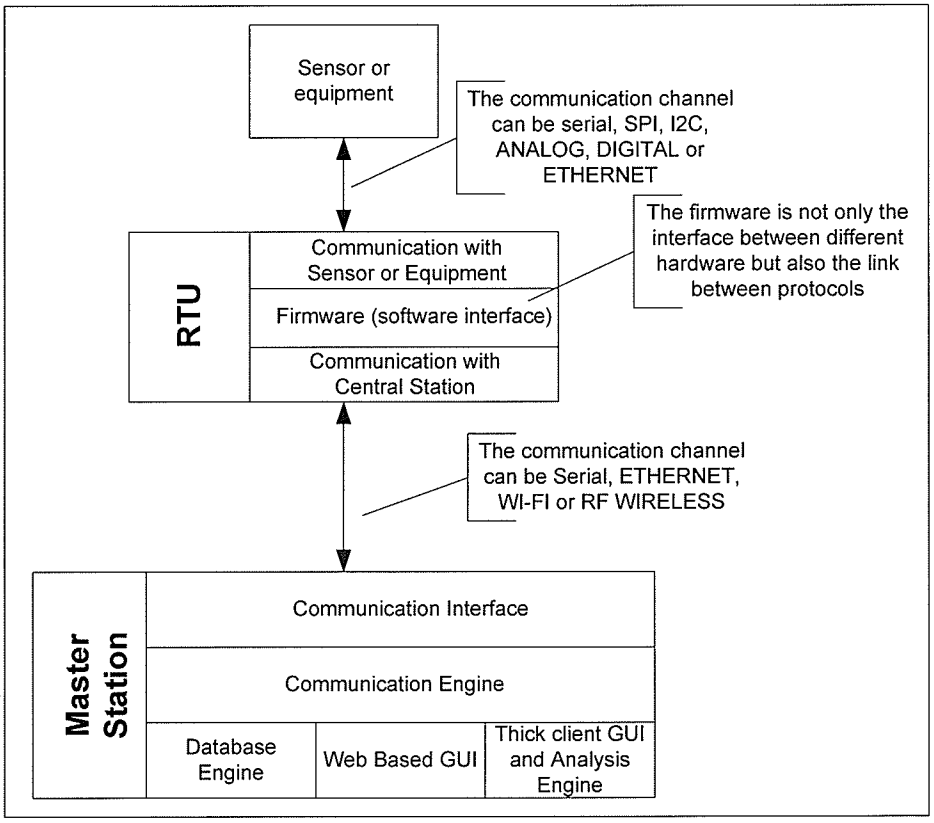


Figure 6.1: Proposed generic SCADA architecture for a sensor agnostic system independent of the communication channel

Within the basic architecture, the SCADA framework should have at least one master station and at least one RTU with its respective sensors and/or actuator.

The master station has many elements. One of the elements is the communication interface that can be installed inside of a computer or as an external device, such as the module for the mesh connected network used in the sandbag dike monitoring project presented in Chapter 3. The communication engine has to have different libraries - each one capable to handle different protocols and media - responsible for the interface between the requests from the presentation or database modules with the remote devices. The communication engine has to receive the requests and be able to translate the request using different communication media such as serial, Ethernet, Wi-Fi and so on. The details of specific media should not be the responsibility of the SCADA application developer in a more commercial version.

If the system is to store information, a database engine is required not only to keep the system's historical readings but also to be used as an interface between the GUI and the communication engine. This also facilitates remote access to the data as a web service with information made available to others over the public internet. Provisions for security in this case would be leveraged from Internet developments such as virtual private networks and IPSec [27].

The last element contained in the master station is the GUI interface. The GUI interface is the software that will help the user to control the system and get data, statistics and reports. There are two alternatives to GUI interfaces. An alternative is the one presented in Chapter 3: a web based interface. The second alternative is the one presented on Chapter 4: an installable application with a traditional software interface. The main disadvantage of the web interface is the limitations in the screen features that can be added. Another inconvenience is that the screen

typically has to be refreshed to update the data. On the other hand, the information presented in the installable application is updated faster on the screen. It also has all the functionality of any kind of regular installable software application. However, it has at least two deficiencies. The first limitation is remote access to the SCADA and secondly the application has to be installed on every system that will be using it.

The master station can be as simple as one computer will all the elements described above, or different computers for each element: communication engine, database engine and GUI.

The project presented on Chapter 3 has the Communication Engine, Database Engine and Web GUI interface on the same computer system. The communication interface is external to the computer. The project on Chapter 4 has the communication interface, communication engine and thick client on the same computer system. The potential evolution - a combination between the projects on Chapter 3 and 4 - presented in Chapter 5 illustrates a combination of features of both systems to create a new solution with minor changes in the SCADA framework. As it can be seen, the proposed system is not only flexible but also it can be used in very different kinds of applications.

The SCADA framework can have multiple RTUs which may also be different from one another. For example, the system can read the external temperature using WI-FI Ethernet and proprietary application protocol while at the same time be used to control a heating system using the BACnet protocol.

The RTUs are also flexible in structure. It is not mandatory to have all the different layers described on the Figure 6.1. In the case of the project for the sandbag dikes, each sensor inside

the sandbags had a microcontroller acting as an interface between the wireless communications and the sensors on board. The RTU was integrated with the passive RFID reader in the case of the RTLS project. The only layer used was the one to communicate with the master station. This also is a common internetworking methodology, layers are specified and used if needed. Similarly this is a common design paradigm from object orientation, it is more difficult to generalize than to exploit inheritance.

The focus to keep a flexible framework for SCADA not only concentrates on the software application developed in the master station but also on the firmware that is part of the remote device.

Another important thing to define is the data request process. The two alternatives are polling and interruption. The sandbag dike monitoring system used polling. The master station initiates the communication. It is a master-slave configuration. After the MS decides the next station to monitor, it will send the request and wait for an answer. If there is no answer within a short period of time, a communication error event will be triggered. The RTLS uses interrupts. After a person walks close to a sensor, the event will trigger a communication with the MS. The MS will be waiting for the data from the sensors. The MS has the option to reply acknowledging that the data was received (see Figure 6.2).

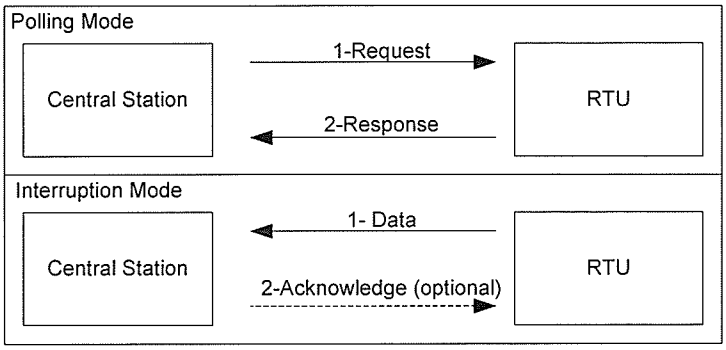


Figure 6.2: Polling and Interruption Modes

A combined mode is also possible. For example, a RTLS where the sensors start the communication can also have status updates initiated by the MS. In other words, the sensors will send data to the MS (interrupt mode) and the MS will request sensor status updates (polling mode).

In summary, it is possible to create a SCADA framework that it is not only sensor agnostic but also independent of the communication channels used. Moreover, the system can be built using affordable off-the shelf components.

However, reflecting on the work and comments above, it is possible that the extent of the SCADA application domain may serve as the major impediment to the development of a usable universal SCADA framework. The counterargument, auguring well for the initial conjecture, is that many of the most advanced and varied web applications could not have proceeded without universal standards with respect to frameworks (e.g. XML, SOA, TCP/IP, etc.)

6.2 Future Work

The following list contains recommendations for future work in evolving SCADA technology to improve integration and affordability:

a) Create an additional layer of protection and modularity for the Communication Engine, Database and GUI using SOA.

b) For the web based GUIs, add AJAX technology to avoid screen refresh difficulties.

c) Analyze the need and affordability of additional levels of data security and encryption.

d) Explore alternatives to connect additional communication channels to the RTUs such as USB or WI-FI.

e) Create a common framework for the microcontroller firmware for multiple sensors or equipment to connect.

Appendix A: Database Structure for Sandbag Dike Project

A1.1 Database Structure

The database has four tables to support the Sandbag Dike project. The table descriptions are shown in Figure A1.1.

Configuration

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Mode	tinyint(4)	NO		NULL	
Time	int(11)	NO		NULL	
AlarmXType	tinyint(4)	YES		NULL	
AlarmX	smallint(6)	YES		NULL	
AlarmYType	tinyint(4)	YES		NULL	
AlarmY	smallint(6)	YES		NULL	
AlarmZType	tinyint(4)	YES		NULL	
AlarmZ	smallint(6)	YES		NULL	

Sensors

Field	Type	Null	Key	Default	Extra
SensorID	smallint(6)	NO	PRI	NULL	
SensorSN	text	YES		NULL	
Description	varchar(40)	NO		NULL	
XAxis	smallint(6)	YES		NULL	
YAxis	smallint(6)	YES		NULL	
ZAxis	smallint(6)	YES		NULL	
ITemp	float	YES		NULL	
IHum	float	YES		NULL	
ETemp	float	YES		NULL	
EHum	float	YES		NULL	
BatteryVoltage	float	YES		NULL	
Active	tinyint(1)	YES		NULL	
Status	tinyint(4)	NO		NULL	
ITempErr	float	NO		NULL	
IHumErr	float	NO		NULL	
ETempErr	float	NO		NULL	
EHumErr	float	NO		NULL	
BatteryVoltageErr	float	NO		NULL	
LastUpdate	datetime	YES		NULL	
XAxisZero	smallint(6)	YES		NULL	
YAxisZero	smallint(6)	YES		NULL	
ZAxisZero	smallint(6)	YES		NULL	
ZeroDateTime	datetime	YES		NULL	
LastValidData	datetime	YES		NULL	

HistoricalData

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
SensorID	smallint(6)	NO		NULL	
SensorSN	text	YES		NULL	
TodayDateTime	datetime	NO		NULL	
XAxis	smallint(6)	YES		NULL	
YAxis	smallint(6)	YES		NULL	
ZAxis	smallint(6)	YES		NULL	
ITemp	float	YES		NULL	
IHum	float	YES		NULL	
ETemp	float	YES		NULL	
EHum	float	YES		NULL	
BatteryVoltage	float	YES		NULL	
ITempErr	float	NO		NULL	
IHumErr	float	NO		NULL	
ETempErr	float	NO		NULL	
EHumErr	float	NO		NULL	
BatteryVoltageErr	float	NO		NULL	
Status	tinyint(4)	NO		NULL	

Scan

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	0	
SensorID	smallint(6)	NO		NULL	
SensorSN	text	YES		NULL	
Wait	int(11)	NO		NULL	
Status	tinyint(4)	NO		NULL	
LastDateTime	datetime	NO		NULL	
Priority	tinyint(4)	NO		NULL	

Figure A1.1 Sandbag database structure

Glossary

AccX	Accelerometer that measures acceleration in the X axis
AccY	Accelerometer that measures acceleration in the Y axis
AccZ	Accelerometer that measures acceleration in the Z axis
ADC	Analog to Digital Converter
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CE	Communication Engine (CE). Java application to link the data to/from the sensors with the database.
GUI	Graphic User Interface
HMI	Human Machine Interface
LSB	Low Significant Bits
MAC	Media Access Control
MS	Master Station (Central Computer System)
MSB	Most Significant Bits
PLC	Programmable Logic Controller
RFID	Radiofrequency Identification
RTLS	Real Time Location System
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SD	Slave Device
SOA	System Oriented Architecture
SOAP	Simple Object Access Protocol
VPN	Virtual Private Network

References

[1] A. Daneels and W.Salter, " WHAT IS SCADA?", International Conference on Accelerator and Large Experimental Physics Control Systems, 1999, Trieste, Italy, pp. 339.

[2] M. García Morente, "Lecciones preliminares de filosofía", Editorial Losada, pp 15-24, 1938.

[3] The Institute of Electrical and Electronics Engineers, Inc. , "IEEE Standard Definition, Specification, and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control, IEEE Std C37.1-1994, pp 10-44, 1994.

[4] "Supervisory Control and Data Adquisition (SCADA) Systems", National Communications System, pp 4, October 2004.

[5] J. R. Chiles, "Inviting Disaster. Lessons from the edge of the technology", Harper Business, pp 37-115, 2002.

[6] H. M. Newman. (March 1997). "BACnet: Answers to Frequently Asked Questions". HPAC Heating/Piping/Air-conditioning [PDF], pp 47. Available: http://www.bacnet.org/Bibliography/HPAC-3-97.pdf

[7] V. Prakash and R. Casey, "SNMP - a new paradigm for SCADA", Semaphore, November 2008.

[8] Modicon Inc, Industrial Automation Systems, "Modicon Modbus Protocol Reference Guide", PI-MBUS-300 Rev. J, 1996.

[9] K. Curtis, DNP User Group, "A DNP3 Protocol Primer Revision A", DNP User Group, 2000, 2005.

[10] P. Oman, E. O. Schweitzer and D. Frincke, " CONCERNS ABOUT INTRUSIONS INTO REMOTELY ACCESSIBLE SUBSTATION CONTROLLERS AND SCADA SYSTEMS", SEL, pp 12, 2000.

[11] "Critical Infrastructure Protection Challenges in Securing Control Systems", General Accounting Office (GAO) Report, GAO-04-140T, pp. 1, October 1, 2003.

[12] D. Bailey and E. Wright, "Practical SCADA for Industry", Elsevier, pp. 235, 2003.

[13] J. Ferchoff, "Remote Sandbag Dike Monitoring", University of Manitoba, Undergraduate Summer Project, 2008.

[14] Analog Devices, "Small, Low Power, 3-Axis ±3 g Accelerometer", pp 10, 2009.

15] Digi International Inc., "X-BEE / X-BEE PRO DigiMesh 2.4 OEM RF Modules", 90000991-A, Dec-2008.

[16] National Electrical Manufacturers Association (NEMA), "NEMA Enclosure Types", NEMA, pp. 2, November 2005.

[17] I. Buchmann, "Batteries in a Portable World", Cadex Electronics, Second Edition, pp. 34-37, 2001.

[18] Panasonic, "Valve Regulated Lead Acid Batteries: Individual Data Sheet", LC-RS1217P, Panasonic, August 2005.

[19] A. T. Holdener III, "Ajax: The Definitive Guide", O'Reilly, pp. 3-21, January 2008.

[20] T. Place, "Final Report on the SOA Architecture Design", NEEO, pp. 2, April 2008.

[21] A. Bhatia, B. Mehta and R. Gupta (2007). Different Localization Techniques for Real Time Location Sensing using passive RFID, http://kict.iiu.edu.my/integ/docs/LocalizationTechniques.pdf

[22] K. Finkenzeller, "RFID Handbook", Wiley, pp 24, 2003.

[23] A. Malik, "RTLS for Dummies", Wiley, pp 9-10, 2009.

[24] M. Ward, R. van Kranenburg and G. Backhouse, "RFID: Frequency, standards, adoption and innovation", JISC Standards and Technology Watch, pp. 10, 2006

[25] S. Garfinkel and B. Rosenberf, "RFID: Applications, Security and Privacy", Addison-Wesley Professional, pp 17, July 2005.

[26] G. Welch and G. Bishop, "An Introduction to the Kalman Filter", University of North Carolina - Department of Computer Science -, TR-95-041, July 2006.

[27] M. Gupta, "Building a Virtual Private Network", Premier-Press, 2003.