Pervasive Sensing and Computing for Natural Disaster Mitigation

Daniel H. Quintela
University of South Florida
Pervasive Sensing and Computing for Natural Disaster Mitigation

By

Daniel H. Quintela

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Wilfrido A. Moreno, Ph.D.
James T. Leffew, Ph.D.
Miguel Labrador, Ph.D.

Date of Approval:
April 6, 2005

Keywords: Wireless Sensor Network, Remote Sensing, Disaster Management, Adaptable Architecture, Motes

©Copyright 2005, Daniel H. Quintela
ACKNOWLEDGEMENTS

I would like to express my appreciation for the opportunity given to me Dr. Moreno, my major professor, in realizing this thesis. I would like to thank the committee members, Dr. Leffew and Dr. Labrador, for their guidance and support.

I am indebted to many student colleagues for their encouragement and cooperation throughout the course of this thesis. I am especially grateful to Mauricio Castillo, Oscar Gonzalez, Jaime Dimate, Karim Souccar, Yohan Prevot, and Nhat Nguyen.

Words would not describe how grateful I am for all the help and support given by my family and friends throughout my graduate studies.

Finally, I would like to express all my gratitude towards the Electrical Engineering faculty in helping me succeed academically.
TABLE OF CONTENTS

LIST OF TABLES iii

LIST OF FIGURES iv

ABSTRACT vi

CHAPTER 1 INTRODUCTION 1
  1.1 Problem Statement 3
  1.2 Research Scope 4
  1.3 Thesis Organization 5

CHAPTER 2 STATE-OF-THE-ART SURVEY 6
  2.1 Space Technology for Remote Sensing 6
  2.2 Telemetry-based Solutions for Remote Sensing 7
  2.3 Wireless Sensor Networks for Remote Sensing 8

CHAPTER 3 WIRELESS SENSOR NETWORK OVERVIEW 10
  3.1 The WSN Concept 10
  3.2 Hardware 11
    3.2.1 Mote Platforms 12
    3.2.2 Data Acquisition Boards 14
    3.2.3 Sensors 15
    3.2.4 Programming Board 16
    3.2.5 Network Gateway 16
  3.3 TinyOS 18
    3.3.1 Concept 19
    3.3.2 Component 20
    3.3.3 Concurrency Model 21
    3.3.4 Scheduling 22
    3.3.5 Programming Language: nesC 22
  3.4 MintRoute Algorithm 23

CHAPTER 4 OVERALL SYSTEM CONCEPT 25
  4.1 Proactive Approach to Disaster Management 27
  4.2 Remote Sensing Network Architecture 28
    4.2.1 Identifying Sensing Parameters 29
    4.2.2 Identifying Locations to Deploy Sensors 31
    4.2.3 Identifying the Available Infrastructure 32
    4.2.4 Cell Design 33
  4.3 Central Processing Office 35
  4.4 A WSN Applied to Natural Disasters 36

CHAPTER 5 SENSOR NODES 38
  5.1 Data Acquisition Requirements for Flash-Flood Monitoring 39
  5.2 Node Types 39
  5.3 Signal Conditioning Techniques 41
5.3.1 Pulse Generating Sensors  
5.3.2 Resistive Transducers  
5.3.3 Data Processing for COTS Sensor boards  
5.4 Sensor Drivers  
5.5 Test Software  
5.6 Cluster Configuration  
5.6.1 Meteorological Mote  
5.6.2 Hydrological Mote  
5.6.3 Seismic Mote  
5.6.4 Repeater Mote

CHAPTER 6 POWERING TECHNIQUES FOR MOTES AND STARGATE  
6.1 Battery Power for Motes  
6.2 Powering the Stargate Board

CHAPTER 7 NETWORK STATISTICS AND COMMUNICATION QUALITY

CHAPTER 8 NETWORK GATEWAY: STARGATE CONFIGURATION  
8.1 Local Database  
8.1.1 Alarm Generation  
8.1.2 Fuzzy Logic  
8.2 Connecting to the GPRS Network  
8.2.1 Establishing a PPP Connection  
8.3 Establishing WLAN Connectivity  
8.4 Internet Connectivity

CHAPTER 9 CONCLUSIONS AND FUTURE WORK

REFERENCES

APPENDICES  
Appendix A PPP Scripts  
A.1 ppp-on script  
A.2 ppp-on-dialer script  
A.3 ppp-off script  
Appendix B Intersema MS5534 Algorithm  
Appendix C Taos TSL2550 Algorithm  
Appendix D MDA300CA DAQ board Schematic  
Appendix E Sensor Driver  
E.1 SoilDriver.nc  
E.2 SoilDriverM.nc  
E.3 TestDriver.nc  
E.4 TestDriverM.nc
LIST OF TABLES

Table 1: MICA2 and MICA2DOT Platform Specifications, [1] 13
Table 2: Absolute Maximum Ratings for the MDA300CA DAQ board, [2] 15
Table 3: Sensing Parameters and COTS Equipments 41
Table 4: Data Packet Structure for the MTS420 Sensorboards 53
Table 5: Transducers Coupled with the Meteorological Mote 58
Table 6: Transducers Coupled with the Hydrological Mote 59
Table 7: Transducers Coupled with the Seismic Mote 60
Table 8: Stargate Computer Current Draw in Different Modes 68
Table 9: Multipliers Based on Winter Time Ambient Temperature 71
Table 10: Communication Quality and Network Statistics for the USF Experiment 75
Table 11: Communication Quality and Network Statistics for the Maracay Experiment 76
Table 12: Communication Quality and Network Statistics for the USF Experiment 77
Table 13: Sample Database Table Running on the Server and Client Computers 80
Table 14: Continuation of Sample Database Table Running on the Server and Client Computers 81
Table 15: Alarm Levels Used to Trigger Transmission of Database Files 83
Table 16: Fuzzy Logic Algorithm as a Proof of Concept 85
Table 17: Rules for the Fuzzy Logic Alarm Algorithm 85
Table 18: Continuation of the Rules for the Fuzzy Logic Alarm Algorithm 85
LIST OF FIGURES

Figure 1: Mica2 Mote and Platform’s Block Diagram [1] 12
Figure 2: MICA2DOT Mote and Platform’s Block Diagram [1] 13
Figure 3: MDA300CA Data Acquisition Board for the MICA2 Motes [2] 14
Figure 4: MDA500CA Data Acquisition Board for the MICA2DOT Motes [2] 14
Figure 5: MIB510 Programming Board [1] 16
Figure 6: Stargate Board [3] 18
Figure 7: TinyOS Component-based Structure 19
Figure 8: TinyOS Task Scheduler [4] 20
Figure 9: Component Block Diagram 21
Figure 10: MintRoute Multihopping Routing Algorithm Used in the Motes [4] 24
Figure 11: Proactive vs Reactive Approaches in Disaster Monitoring 28
Figure 12: Chain of Events Culminating in a Flash-flood Disaster 30
Figure 13: System Concept Block Diagram 33
Figure 14: System Data Structure Concept 34
Figure 15: Basic Concept Architecture 35
Figure 16: “Davis” Rain Collector II 42
Figure 17: “Davis” Anemometer 43
Figure 18: “Swoffer” Water Flow Transducer 44
Figure 19: Pulse Signal Conditioning for Different Types of Transducers 44
Figure 20: Schematic for Resistive Signal Conditioning 45
Figure 21: “InterMountain” Float & Pulley Water Level Transducer 45
Figure 22: “Davis” Soil Moisture Sensor 46
Figure 23: Pressure and Temperature Readings from an Intersema MS5534 Sensor 48
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Signal Conditioning Circuitry for the Resistive Soil Moisture Sensor</td>
<td>50</td>
</tr>
<tr>
<td>25</td>
<td>Test Sensor Application Wiring</td>
<td>50</td>
</tr>
<tr>
<td>26</td>
<td>Soil Driver Configuration Wiring</td>
<td>51</td>
</tr>
<tr>
<td>27</td>
<td>Testing Configuration for Sensor Devices Prior to Deployment</td>
<td>52</td>
</tr>
<tr>
<td>28</td>
<td>Front Panel of the LabView Application Used to Test the MTS420CA Sensor board</td>
<td>54</td>
</tr>
<tr>
<td>29</td>
<td>Block Diagram with the Implementation of the Sensor Algorithms</td>
<td>55</td>
</tr>
<tr>
<td>30</td>
<td>Implementation of the Sensor Algorithms</td>
<td>56</td>
</tr>
<tr>
<td>31</td>
<td>Cluster Architecture for Flash-Flood Monitoring</td>
<td>57</td>
</tr>
<tr>
<td>32</td>
<td>Block Diagram of the Meteorological Motes</td>
<td>58</td>
</tr>
<tr>
<td>33</td>
<td>Block Diagram of the Hydrological Mote</td>
<td>59</td>
</tr>
<tr>
<td>34</td>
<td>Block Diagram of the Seismic Mote</td>
<td>60</td>
</tr>
<tr>
<td>35</td>
<td>Powering Scheme Using Two AA Alkaline Batteries</td>
<td>62</td>
</tr>
<tr>
<td>36</td>
<td>Discharge Characteristics for 1.5V Panasonic Industrial Alkaline Batteries</td>
<td>63</td>
</tr>
<tr>
<td>37</td>
<td>System Characteristics Investigated for the Two Powering Schemes</td>
<td>64</td>
</tr>
<tr>
<td>38</td>
<td>Battery Lifetime vs. Battery Capacity for the Two Powering Schemes</td>
<td>65</td>
</tr>
<tr>
<td>39</td>
<td>Powering Scheme Using Three AA NiMH Batteries</td>
<td>66</td>
</tr>
<tr>
<td>40</td>
<td>Discharge Characteristics for 1.2V Energizer Rechargeable NiMH Batteries</td>
<td>67</td>
</tr>
<tr>
<td>41</td>
<td>Architecture to Provide Solar Energy for the Stargate Computer</td>
<td>69</td>
</tr>
<tr>
<td>42</td>
<td>Testing Network Statistics for the Sensor Network at USF</td>
<td>74</td>
</tr>
<tr>
<td>43</td>
<td>Deployment Site in Maracay,Venezuela</td>
<td>75</td>
</tr>
<tr>
<td>44</td>
<td>TinyOS Message Structure with Dynamic Payload Length</td>
<td>79</td>
</tr>
<tr>
<td>45</td>
<td>Fuzzy Inference System for the Alarm Algorithm</td>
<td>85</td>
</tr>
<tr>
<td>46</td>
<td>Intersema MS5534 Algorithm</td>
<td>98</td>
</tr>
<tr>
<td>47</td>
<td>Taos TSL2550 Algorithm</td>
<td>100</td>
</tr>
<tr>
<td>48</td>
<td>MDA300CA DAQ Board Schematic</td>
<td>101</td>
</tr>
</tbody>
</table>
PERVASIVE SENSING AND COMPUTING FOR NATURAL DISASTER MITIGATION

Daniel H. Quintela

ABSTRACT

This research proposed the use of state-of-the-art wireless communications and networked embedded systems technologies to provide environmental sensing for the early detection of natural disasters. The data is acquired, processed and transmitted, from the location where the disaster originates, to potentially threatened conurbations in order to promptly notify the population. The acquired data is transformed from its raw form into information that can be utilized by local authorities to rapidly assess emergency situations and then to apply disaster management procedures. Alternatively, the system can generate alerting signals without human intervention. Furthermore, recorded historical data can be made available for scientists to build models, to understand and to forecast the behavior of disasterous events. An additional, important, contribution of this research was the analysis and application of Wireless Sensor Network technology for disaster monitoring and alerting.
CHAPTER 1

INTRODUCTION

Technology has evolved in quantum leaps in recent years. However, technology is still not widely used for mitigating natural disasters that have devastating consequences in regions of the world where the lack of preparedness and basic infrastructure are characteristic. Such disasters are unpredictable by nature and will continue to be a threat to mankind in the years to come. However, it has been observed that where technology is available, strategies to lessen the impact of a disaster are employed in advance to mitigate the loss of lives and property. In developed countries, where technology is applied, the integration of state-of-the-art wireless communications and information technologies has become the foundation in the development of solutions to monitor, to alert and to mitigate such unpredictable events. Historical databases and behavioral models of the environment, which are extracted with ubiquitous sensing from the site, provide accurate and reliable data to authorities. The timely access to relevant information on hazardous environmental conditions provide time for the community to apply preparedness procedures that are capable of alleviating damage and reducing the number of casualties derived from the event. The contrast in disaster preparedness from developed to underdeveloped countries that are still subject to the lack of information and infrastructure is evident. In the underdeveloped countries disaster management is reduced to response and recovery efforts from the governments after the event has
occurred. Little can be done by the time civil defense and other assisting agencies arrive at ground zero.

Disaster management procedures can be compared over time in those regions of the world where technology is currently present for monitoring and alerting. The Galveston Hurricane of September 1900, regarded as the greatest natural disaster to ever strike the United States, caused at least 8,000 deaths in the hours following the landfall of the hurricane in Texas, [5]. Even though warnings were issued, at the time, they were not taken seriously and many chose to stay at home and not seek shelter. In 2004, from August to September, four hurricanes struck the Florida coast causing billions of dollars in property damage. However, these disasters accounted for only approximately 152 deaths, [6]. This relatively small number of casualties can be attributed to an effective and accurate monitoring system that alerted authorities and the population in advance.

What is observed in underdeveloped countries nowadays is similar to what was seen in the United States at the beginning of last century. The technology available to these countries is insufficient to provide their populations with reliable systems for disaster monitoring and management. The lack of infrastructure and technology necessary for collaboration in the mitigation and management of disasters put at risk lives and properties that could be preserved. The most recent natural disaster, which was one of the most devastating of all time, occurred at the end of 2004 when a tsunami hit South Asia causing approximately 221,100 deaths and several billions of dollars in property damage, [7]. The lack of communication channels from other disaster monitoring sites to authorities in the South Asia region, to warn that a catastrophe was imminent, contributed to the several thousands of lives that were lost.
In recent years, disasters caused by mankind have been as devastating as the ones caused by nature. The common ground between the two is their unpredictable nature and their consequences. Man-made disasters are even harder to predict since the parameters involved in the monitoring of such events are more subjective. In the 9/11 terrorist attacks of 2001 suspicious evidence that an attack was eminent were underestimated by United States governmental agencies. In the case of man-made disasters technology can be applied for the creation of databases for suspects, the development of biochemical sensors and for the development of an interoperable communication technology among the emergency workers

1.1 Problem Statement

Money and effort are normally invested to mitigate the effects of natural disasters after they have occurred. However, in order to lessen the effects of such events it is necessary to anticipate their occurrence. A monitoring system that provides authorities accurate and reliable information prior to a natural disaster provides the community time to apply preparedness procedures, which will save lives and minimize property loss. The majority of the current commercially available monitoring and alerting systems for disasters use telemetry solutions that are expensive, difficult to install and are configured on centralized schemes that often compromise the reliability of the system.
1.2 Research Scope

This research proposed the use of state-of-the-art wireless communications and networked embedded systems technologies to provide environmental sensing for the early detection of natural disasters. The scope of this research was restricted to identifying the best solution for environmental monitoring, establishing the requirements and overall system concept for the solution, adapting technology for natural disaster monitoring and discussions of the test results obtained from one proven deployment.

The central workstation where data is processed and analyzed is discussed conceptually. However, it was not implemented. Throughout the thesis flash-floods will be used as the main example of the natural disaster being monitored. Other types of natural disasters would utilize the same system architecture. The only difference with respect to disasters that are different from the flash-flood solution presented here would be the adaptation of different sensors. Based on the fact that for disaster monitoring the position of the sensing points is predetermined and strategically placed in locations that extract the most relevant data, the resulting network topologies are fixed. Such fixed topologies yield simpler routing algorithms.

The environmental sensors, for this research, collected environmental data related to temperature, barometric pressure, precipitation, humidity, ambient luminosity, two-axis accelerations, water level, water flow and sound readings. These were relevant parameters for the sample application.
1.3 Thesis Organization

This thesis consists of nine chapters. Chapter 2 presents a State-of-the-Art survey that was conducted to identify the technologies being currently used for remote sensing related to natural disaster monitoring solutions. Chapter 3 discusses Wireless Sensor Networks, which is an emerging technology that is well suited for remote sensing. Basic concepts that include discussions of hardware and software components are presented. Chapter 4 presents the system concept and identifies the requirements imposed by the problem statement for this research. Chapter 5 describes the hardware architecture used for the sensing nodes and how they were configured according to application-specific requirements to collect, process and transmit information from remote locations to the network gateway. Chapter 6 discusses the powering techniques used for the sensing nodes and the network gateway. Chapter 7 presents results from field tests regarding network statistics and communication quality. Chapter 8 describes the implementation of the network gateway and Chapter 9 includes the conclusions and recommendations for future work in this area.
CHAPTER 2

STATE-OF-THE-ART SURVEY

There are several environmental monitoring systems currently used for disaster management, [8], [9], [10]. Traditionally, space technology and telemetry systems have been used in the remote sensing of the environment at risk. However, the emergence of Wireless Sensor Networks, (WSNs), in recent years has prompted researchers to investigate the possibility of implementing WSNs for disaster monitoring and management, [11], [12]. This section of the thesis describes the advantages and disadvantages of sensing the environment using space technology and telemetry-based systems. Afterwards, two examples of projects implementing WSNs for disaster monitoring and management are presented.

2.1 Space Technology for Remote Sensing

Satellite remote sensing is the most sophisticated technology used for environmental monitoring in the prediction of natural disasters. The satellites carry onboard sensors that are capable of providing information on every natural feature that prevails on the surface of the Earth. Depending on the type of disaster being monitored, different onboard sensors are employed. For example, thermal sensors capture fire
hazards, infrared sensors are more suitable for floods and microwave sensors can record soil moisture, [13]. The two main types of satellites used to observe the Earth are the Geostationary and the Polar-Orbiting satellites. The Geostationary satellites are primarily used for meteorological observation whereas the Polar-Orbiting satellites are particularly important in the monitoring of natural disasters. The data extracted from the satellites are transmitted back to ground stations where the information is processed by computers designed for complex signal processing. The data extracted from the satellites is applied to precisely detect, map, measure and analyze the environment. The accuracy, the extended coverage and the spatial continuity obtained from satellite readings are among the main advantages available from this technology for remote sensing, [14].

Furthermore, satellite remote sensing provides real-time assessment of the event, which is helpful in identifying evacuation routes to safe zones away from the disaster.

Unfortunately, not all countries can rely on space technology for remote sensing. In fact, most developing countries have limitations in terms of hardware, software and human resources, [14]. The satellite solution requires powerful high-end computers for signal processing, software such as a Geographical Information System, (GIS), to implement data analysis, statistics based behavioral models and most importantly qualified professionals to operate the system. The cost to set up and operate a solution of this magnitude and complexity is also an issue for developing countries.

2.2 Telemetry-based Solutions for Remote Sensing

Many of the remote sensing solutions used for disaster management are based on telemetry systems, [10]. Remote sensing solutions make use of remote terminal units that
are coupled with sensors to collect data and in a point-to-point strategy transmit the data to a central terminal unit. Each remote terminal unit needs to be self-powered. The remote terminals can be powered by a solar panel or by using an Uninterruptible Power Supply, (UPS). The medium used for communication consists of elements such as cable, radio, telephone and satellite. Telemetry-based solutions utilize UHF, VHF and cellular networks for communication. However, a centralized scheme compromises the reliability of the solution since each section or even the entire sensed field might be isolated in the event of terminal unit failure or malfunction. In addition, these invasive architectures are difficult to deploy and to operate. The installation process is time consuming and once established the infrastructure is permanent and not easily extendable.

2.3 Wireless Sensor Networks for Remote Sensing

The use of Wireless Sensor Networks for natural disaster monitoring is still a novel approach in the attempt to minimize the loss of lives and property incurred as the result of a disastrous event. Initially fueled by the evident commonalities with environmental monitoring, disaster monitoring applications are rapidly evolving as the technology is trying to adapt to support the new and imposed requirements. To date only a limited number of major projects have implemented WSNs for natural disaster monitoring and response. For example, the FireBug project focuses on extracting environmental data from remote locations to alert first responders and the population to the risk of wildfires, whereas the CodeBlue project exploits the use of WSN technology to obtain vital signs of patients in disaster response. The FireBug project was implemented by the University of California, Berkeley and sponsored by the NSF.
Information Technology Research Division. This effort was one of the initial attempts to make use of a WSN for such an application. The project aimed at the development of a platform to detect initiation and to monitor the spread of wildfires in rapidly changing environments, [12]. Each sensing node is equipped with a GPS module and an environmental sensor board. The collected data is routed back to a central station and made available to first responders and the general public. 

The CodeBlue project, developed by Harvard University, explores applications of wireless sensor network technology to a range of medical applications that include pre-hospital and in-hospital emergency care, disaster response and stroke patient rehabilitation, [11]. The patients or victims wear motes equipped with a wireless pulse oximeter and a two-lead EKG to collect heart rate, oxygen saturation and EKG data. The collected data is routed back to remote stations such as PDAs, laptops or ambulances to be stored in the patient’s profile. Additional features encompassed in the motes include alarms to notify first responders in the event of any vital sign that reaches life-threatening levels. The results from both experiments showed that WSNs lend themselves well for natural disaster monitoring. However, these experiments also exposed issues that still need to be addressed in order to improve overall performance and reliability. This research identified aspects that need to be resolved for the appropriate implementation of a WSN for natural disaster monitoring, proposed solutions and suggests areas for future work.
3.1 The WSN Concept

Wireless Sensor Networks are “low-power, multihopping systems that combine multiple wireless nodes into an extendable network environment with non-Line-Of-Sight coverage and a self-healing data path” that provide ubiquitous sensing of any environment in the monitoring of natural disasters, [15]. WSN nodes communicate only with neighboring nodes, which reduces the need for high transmission power and eliminates the need for expensive transmitters and repeaters such as those used in traditional telemetry systems. Every node, in a WSN, can act as a data acquisition device, a data router or a data aggregator. As will be discussed later, the clustering architecture chosen to be implemented for this solution maximized the redundancy and consequently, the reliability of the entire monitoring system.

The independence from third-party providers and the absence of infrastructure requirements such as those required in cellular based telemetry systems allow a WSN to be deployed quickly. Furthermore, in scenarios where threats may come from unexpected locations, having a dynamic and adaptable solution enables first-responders to act according to critical situations. For example, these features facilitate the placement
of additional nodes to provide a more comprehensive reading of the event as it happens and to replace “dead” nodes. In the event of network congestion, node failure or simply obstacles blocking line-of-sight communications, the meshed interconnection of wireless sensor nodes generate alternative paths for data routing from the source, where the phenomena occurred, to the destination, which is a network gateway. Network gateways in a WSN allow interaction with external systems that possess more storage and computational capacity to create historical databases and for purposes of modeling and forecasting.

### 3.2 Hardware

This section describes the hardware specifications of the equipments used to configure the proposed solution. All components were purchased off-the-shelf as general-purpose sensor network equipment for later adaptation in order to fit the objective of the research. The discussion on the functionalities of each component and how their integration forms a Wireless Sensor Network to monitor natural disasters is discussed in Chapters 4 and 5. The network was comprised of sensor nodes, or “motes”, data acquisition boards, sensors and a network gateway. In addition, a programming board was required to download the nesC code into the motes. For further details of these devices refer to reference [4].
3.2.1 Mote Platforms

The MICA2, (MPR410,) and the MICA2DOT, (MPR510), were two types of mote platforms used. These platforms are interoperable and the major difference between them is the physical size. Figures 1 and 2 respectively illustrate the MICA2 and MICA2DOT platforms along with their block diagram. Table 1 describes the specifications of both platforms.

Figure 1: Mica2 Mote and Platform’s Block Diagram, [1]
Figure 2: MICA2DOT Mote and Platform’s Block Diagram, [1]

Table 1: MICA2 and MICA2DOT Platform Specifications, [1]

<table>
<thead>
<tr>
<th></th>
<th>MICA2</th>
<th>MICA2DOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Range</td>
<td>433.1 to 434.8 MHz</td>
<td>433.1 to 434.8 MHz</td>
</tr>
<tr>
<td>Processor</td>
<td>Atmel ATmega 128L</td>
<td>Atmel ATmega 128L</td>
</tr>
<tr>
<td>Radio Transceiver</td>
<td>Chipcon CC1000</td>
<td>Chipcon CC1000</td>
</tr>
<tr>
<td>Program Memory</td>
<td>128 kbytes</td>
<td>128 kbytes</td>
</tr>
<tr>
<td>External Flash Memory</td>
<td>Atmel (512 kbytes)</td>
<td>Atmel (512 kbytes)</td>
</tr>
<tr>
<td>Programming Interface</td>
<td>(2) UARTs</td>
<td>UART</td>
</tr>
<tr>
<td></td>
<td>Serial Port Interface</td>
<td>Serial Port Interface</td>
</tr>
<tr>
<td></td>
<td>I2C bus</td>
<td>I2C bus</td>
</tr>
<tr>
<td>Expansion Connector</td>
<td>51-pin connector</td>
<td>19-pin connector</td>
</tr>
<tr>
<td></td>
<td>(8) 10-bit analog I/O</td>
<td>(6) 10-bit analog I/O</td>
</tr>
<tr>
<td></td>
<td>(21) digital I/O</td>
<td>(6) digital I/O</td>
</tr>
<tr>
<td>User Interface</td>
<td>(3) LEDs</td>
<td>LED</td>
</tr>
<tr>
<td>Power Supply</td>
<td>(2) AA batteries</td>
<td>3V Lithium Coin Cell</td>
</tr>
<tr>
<td></td>
<td>1850 mAh</td>
<td>560 mAh</td>
</tr>
</tbody>
</table>
3.2.2 Data Acquisition Boards

There were two types of data acquisition boards used in the solution, one for each platform. The DAQ board used for the MICA2 motes was the MDA300CA, which is pictured in Figure 3. The MDA500CA board was used for the MICA2DOT motes and is pictured in Figure 4. Analog sensors can be attached to different channels of the MDA300CA boards based on the expected precision and dynamic range. Digital sensors can be attached to the digital or counter channels. A mote samples analog, digital or counter channels and can actuate via digital outputs or relays. The combination of the MICA2 mote and a MDA300CA can be used as a low-power wireless data acquisition device or process control machine. Table 2 details the absolute maximum ratings for various electrical parameters. For the MDA500CA boards, all of the major I/O signals of the MICA2DOT mote are routed to plated-thru holes on the MDA500 circuit board, [2].

Figure 3: MDA300CA Data Acquisition Board for the MICA2 Motes, [2]

Figure 4: MDA500CA Data Acquisition Board for the MICA2DOT Motes, [2]
Table 2: Absolute Maximum Ratings for the MDA300CA DAQ board, [2]

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+VDD to GND</td>
<td>-0.3V to +5.5V</td>
</tr>
<tr>
<td>Digital Lines</td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td>-0.5V to VDD+0.5V</td>
</tr>
<tr>
<td>Continuous output low current</td>
<td>50mA</td>
</tr>
<tr>
<td>Continuous output high current</td>
<td>-4mA</td>
</tr>
<tr>
<td>Analog Lines</td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td>-0.2V to Vcc+0.5V</td>
</tr>
<tr>
<td>Counter Line</td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td>0V to 5.5V</td>
</tr>
<tr>
<td>Relays</td>
<td></td>
</tr>
<tr>
<td>Maximum Contact Voltage</td>
<td>100V</td>
</tr>
<tr>
<td>Maximum Contact Current</td>
<td>150mA</td>
</tr>
</tbody>
</table>

3.2.3 Sensors

Several sensors are used to monitor the environment at risk. The motes are equipped with the following sensors:

- Crossbow MTS420CA and MTS510CA Environmental Sensor boards,
- InterMountain Water level sensor,
- Swoffer Water flow sensor,
- Davis Rain Collector II Precipitation sensor,
- Davis Anemometer Wind direction sensor,
- Davis Anemometer Wind speed sensor,
- Davis Watermark Soil moisture sensor.
3.2.4 Programming Board

The MIB510 serial interface board is a multi-purpose board that is used to program the MICA2 and MICA2DOT motes. Code is downloaded to the ISP through an RS-232 serial port where the ISP programs the code into the mote. The ISP and Mote share the same serial port. The ISP runs at a fixed baud rate of 115.2 kbaud. The ISP continually monitors incoming serial packets for a special multi-byte pattern. Once the pattern is detected it disables the Mote’s serial RX and TX and takes control of the serial port, [1]. The MIB510 Programming board is pictured in Figure 5.

![MIB510 Programming Board](image)

Figure 5: MIB510 Programming Board, [1]

3.2.5 Network Gateway

The Stargate board is the "sink" of the Wireless Sensor Network. It possesses enhanced communications and signal processing capabilities. The features of the Stargate board are:

- 32-bit, 400 MHz Intel PXA-255 XScale RISC processor,
- SA1111 StrongARM Companion Chip for Multiple I/O Access,
• 32 MB of Intel StrataFlash,
• 64 MB of SDRAM,
• 1 Type II CompactFlash+ Slot,
• 1 PCMCIA Slot,
• Small Form Factor with dimensions of 3.5" x 2.5",
• Reset Button,
• Real Time Clock,
• Lithium Ion Battery option,
• MICA2 Mote capability with GPIO/SSP and Other Signals via 51pin Expansion connector,
• I2C connector via an Installable Header,
• A 51-pin Daughter Card Interface,
• Wired Ethernet via a 10/100 Base-T Ethernet port,
• Host USB port,
• JTAG Port,
• External A/C power supply adapter,
• RS-232 Serial Port via a DB-9 Connector.

The Stargate board is pictured in Figure 6.
3.3 TinyOS

The unique characteristics of Wireless Sensor Networks required the development of a unique operating system to comply with the hardware requirements imposed by wireless embedded sensor networks. The Tiny Microthreading Operating System, (TinyOS), is a low-power, component-based, and event-driven operating system designed to support intense concurrent operation. TinyOS was written in nesC, which will be discussed in section 3.3.5. The structure of TinyOS is illustrated in Figure 7.
3.3.1 Concept

TinyOS operation forces the motes remain asleep while waiting for an event to happen. Whenever an external event is captured by a transceiver or the sensors an interrupt is generated and the lower-level components signal events to the higher-level components. The event handlers then post tasks that run to completion unless preempted by another event. Tasks run asynchronously from events, which provide a threaded system behavior. Tasks are placed in a queue inside a First In First Out, (FIFO), task scheduler. Figure 8 illustrates the storage of tasks. After all tasks have been executed and the queue is emptied, TinyOS shuts down the processor while maintaining the peripherals operational.
Perhaps the main issue regarding a WSN concerns power consumption. With regard to power consumption, TinyOS enforces a power management strategy within the task scheduler to power the processor only when events are detected.

**Figure 8: TinyOS Task Scheduler, [4]**

### 3.3.2 Component

The structure of the operating system is based on components that are comprised of a fixed-size frame, tasks, event handlers and command handlers. Figure 9 presents the block diagram for a component. Components, which are the building blocks of TinyOS, possess bi-direction interfaces as “communication ports” where interface providers implement commands and interface users implement events. The bi-directional interfaces alleviate the density of data flow, which simplifies the system’s structure.

Commands are generated as non-blocking requests from higher-level components to lower-level components to request parameters, where return status is expected, and post tasks for later execution. Events are generated from lower-level components to
higher-level components to signal asynchronous preempt tasks, to call commands and to post tasks among other duties. Both commands and events, explicitly declared to favor modularity, are simply ‘C’-like function calls implemented inside the program module. The component frame handles the internal state and memory. The frame, which is statically allocated before compilation, reduces the memory requirement and prevents the overhead associated with dynamic allocation, [16].

**Figure 9: Component Block Diagram**

---

### 3.3.3 Concurrency Model

In TinyOS the concurrency model is comprised of tasks and hardware event handlers. Tasks are functions that, once scheduled, run to completion. Tasks are atomic with respect to other tasks and may be preempted only by events. Tasks can call lower level commands, signal higher level events and schedule other tasks within a component. The run-to-completion semantics of tasks make it possible to allocate a single stack and assign it to the currently executing task. This capability is essential in memory constrained systems, [16]. Hardware event handlers are executed in response to a hardware interrupt. They run to completion but may preempt the execution of a task or
other hardware event handler. When preemption occurs, the interrupt routine handler
saves the status at the start and restores it when it ends. The context switch between the
activated hardware event handler and another hardware event handler or a task is
performed automatically without the need of any special context management.

3.3.4 Scheduling

TinyOS executes only one program consisting of selected system components and
custom components required for a single application. A complete system configuration
consists of a tiny scheduler and a graph of the components. A complete system
configuration runs in a single address space and contains two execution environments.
Interrupt handlers running at high priority comprise one execution environment. Tasks
that are scheduled in a FIFO order at low priority comprise a second execution
environment. Tasks are stored in a FIFO that holds a maximum of 8 tasks.

3.3.5 Programming Language: nesC

The programming language nesC was specifically designed to handle the
restrictions inherited by low-power networked embedded systems such as Wireless
Sensor Networks. Derived from the ’C’ programming language, this dialect was
developed to support TinyOS-powered motes with the same syntax and structure as the
ones possessed by the operating system. Given their limited resources, nesC addresses
several fundamental issues in mote operation. Equipped with an executable code space
of only 128 Kbytes of reprogrammable flash memory and severe power constraints, the
MICA2 and MICA2DOT motes need to remain asleep for the majority of the time, wake
up, execute the process quickly and go back to sleep. Limitations in computational power require the language to have a flexible and reusable architecture to ease the job of wiring components during the assembly process of an application. Components are the building blocks of TinyOS and nesC where specific functions are performed and the application-specific codes are implemented. Components are explicitly wired together with bi-directional interfaces to form a configuration or an application code. A pre-compiler for nesC, converts wiring of high-level modules into code where the nesC output is a ‘C’ program file that is compiled and linked using gnu and gcc tools for a specific mote, [17]. Configurations and modules comprise the two types of components available. Configurations are composed of one or more components wired together whereas modules contain the actual nesC code to be implemented.

3.4 MintRoute Algorithm

The power constraint of Wireless Sensor Networks requires the use of multihopping routing schemes to minimize long range transmissions from remote nodes to the base station. In this fashion, data packets are hopped from node to node in short range transmissions, which conserves power and extends network lifetime. The MintRoute algorithm routes the data by selecting the path with the best link quality and the least transmission “cost”. A dynamic routing table specifies the least power consuming path from any node to the base station. The table is managed by the base station and is changed as the network topology changes or when better routing paths are discovered. Based on this multihopping concept motes can communicate around obstacles, which are non-Line-of-Sight and even exploit the environment by
communicating through multipath reflections. Figure 10 illustrates the MintRoute algorithm.

Figure 10: MintRoute Multihopping Routing Algorithm Used in the Motes, [4]
CHAPTER 4

OVERALL SYSTEM CONCEPT

The main objective of designing a natural disaster monitoring solution is to gather environmental information. Based on the collected data information is passed autonomously, to the authorities and the population, to alert them of the level of risk. In order to accomplish this goal it is necessary to first collect data and then transmit it to a centralized station and consequently to broadcast alerts. Compared to applications that focus only on collecting environmental data a disaster monitoring system has more stringent requirements since the information delivered is of vital importance. Disaster information is considered to exist in “soft-real-time”. The system has to remain operational at all times even when individual components fail. Therefore, the system requires the characteristics of distributed systems to avoid bottlenecks and to avoid constraining the reliability of the entire system upon individual components. The locations where the data-gathering must take place often lack electrical and communication infrastructure, which makes conventional monitoring systems inappropriate. The remote sensing solution must adapt itself to the environment and rely solely on its own resources, which must be independent of third-party providers and work under unattended operational conditions. The possibility of extending the network coverage by integrating complementary networks to the solution would only enhance the system but the overall system’s functionality would remain intact even if the
complementary networks failed. As a consequence, one design priority is to maximize the autonomy and reliability of the remote sensing solution. Sensors must be easily deployable in order to be strategically placed in the locations prone to the events that are to be monitored. Another enhancement to the system is to extend the network during or after the event has occurred to obtain a more comprehensive reading of the sensed field and to compensate for malfunctioning sensors. Such an enhancement requires an extendable and flexible solution. The core design of the system architecture must be adaptable to all types of natural disasters. Different types of sensors must be used according to the phenomena being observed. It is of fundamental importance to develop a system that is sensitive and able to effectively recognize hazardous conditions but at the same time the system must be “intelligent” enough not to overreact and trigger false alarms. A fundamental tradeoff for natural disaster monitoring systems is sensitivity verses false alarms. Regarding user interface related requirements, local authorities and first-response personnel have expressed the desire to be able to interact with the alerting system from the urban area through media such as cell phones and internet access or to obtain information from the sensed field through portable equipments such as Personal Data Assistants, (PDAs), and laptop computers. The ability to assign a spatial location to specific events directly related to the occurrence of a disaster is essential. For example, spatial orientation is necessary in order to observe physical magnitudes on a map of the monitored region. According to the challenges and requirements identified in this research, the system has to fulfill the following tasks:

- Gather relevant data from the threatened region where the disasters originate,
• Transmit the data from the "sensed field" to the urban area,
• Extract and graphically display relevant information to assist authorities to make decisions,
• Register the collected data and store it for later use,
• Generate alerting signals,
• Allow the users to interact with the system via mobile portable devices.

4.1 Proactive Approach to Disaster Management

As illustrated in Figure 11, communication and information technologies provide a proactive approach to disaster management. Early detection of the event provides the time necessary for the population and authorities to apply preparedness procedures in the hours anticipating the disaster. The response and recovery from the event can be planned out by local authorities prior to the disaster and carried out quickly and effectively in the hours following the event. When technology is not available the disaster is unforeseen by the population and authorities. Response and recovery is more difficult because there has been no time for authorities to prepare themselves for the disaster. In addition, the panic caused by the unexpected event leads to an even greater number of injuries and deaths.
4.2 Remote Sensing Network Architecture

Before designing the remote sensing network architecture for a natural disaster monitoring system it is fundamental that the region and the phenomena that affect it are well understood. Even though two different sites may be monitoring the same natural disaster, each environment is unique and may present different challenges in designing the architecture. A site survey is necessary to expose issues regarding communications, network density, cell architecture, connectivity to complementary networks and to better envision the solution. Regardless of the phenomenon being monitored, the first steps in
designing the system is identifying the parameters that need to be monitored and the locations where the sensors need to be deployed. Based on the location where the network should be deployed, the available infrastructure at the site must be determined. It is important to determine possible complementary networks that will be used along with the WSN in order to structure data dissemination from the network source. With such information available, a cluster cell architecture can be designed to ensure that nodes are able to communicate among themselves and that communication quality such as link quality, packet loss, prediction, BER, RSSI and network statistics such as battery voltage, duty cycle, average level, level changes, parent changes, received package, sent package and success rate within the network are satisfactory. It may be necessary to increase network density if any of the communication and network parameters are unsatisfactory.

4.2.1 Identifying Sensing Parameters

For each type of disaster monitored, application-specific sensors are required. Once the sensing parameters have been established the flexibility of the system concept allows the designer to keep the core architecture of the solution intact by adapting and integrating only the required sensors to the network. Identifying the sensing parameters requires structuring the chain of events that will eventually trigger the disaster. The sensors are selected according to the requirements determined by the local authorities. The solution is designed to comply with the necessity of each “customer”. Flash-floods, which are sudden discharges of large amounts of water, present the chain of events depicted in Figure 12.
Figure 12 demonstrates that the parameters that trigger flash-floods are the ones concentrated in the prediction and formation stages. Rainfall is the first predictor in flash-flood monitoring. The meteorological predictors that will form the basis of the threat recognition consist of precipitation, wind direction, wind speed, temperature, humidity, luminosity and barometric pressure sensor readings. Combinations among these parameters indicate the likelihood of rainfall. In luminosity, it is possible to detect the intensity of the solar rays. Under a torrid sun, it is very unlikely to rain. Slow winds, high barometric pressure and low humidity are other indicators that the climate is steady and that rain is not imminent. The precipitation sensor provides the decisive reading in determining if rain is present or not. Changes in any one of these parameters might trigger a chain reaction leading to rainfall. Identifying other factors that may contribute to flash-floods is the next step. Depending on the geographical characteristics of the region other phenomenon such as landslides may contribute to flash-floods. Under high precipitation levels mountainous terrains that surround the rivers become unstable. The
heavy rain softens the ground, which triggers landslides that cause a dam-building effect on the rivers. Early detection of unstable terrain can be observed using soil moisture sensors to sense how deep the water has penetrated the soil and indicate the humidity level of the soil. The use of seismic sensors is appropriate to detect land movement in the locations prone to collapse. Eventually, these naturally built barriers cannot sustain the high potential of energy of the accumulated water and finally break; discharging large amounts of water downstream. Water level and water flow sensors at different locations along a river provide differential reading that are used to predict the disaster. For example, if a sensor location along a river, which is known to be highly affected by landslides, senses abnormal increases in water level readings and at another location downstream the level readings are detected to be at or under normal levels, the differential can be used to indicate that a dam has been formed at the location where the increase in level is detected and that a flash-flood disaster is imminent if that barrier is not broken. Water level and water flow readings also assist in identifying when the safety thresholds of the environment are crossed.

### 4.2.2 Identifying Locations to Deploy Sensors

Understanding the environment at risk optimizes the response of the network by extracting the most relevant data at the most appropriate location. These locations are usually determined by professionals of the environmental field in conjunction with Civil Defense authorities. Such locations are characterized by their tendency to trigger events that lead to a disaster. By extracting the data from the field at these strategic locations, the parameters being investigated are likely to break safeguards at earlier stages of the
monitoring process, which provides more time for authorities to alert the population and to apply preparedness procedures.

4.2.3 Identifying the Available Infrastructure

The remote sensing solution for disaster monitoring must be adaptable not only to the environment but also to the available infrastructure. Even though the network can be extended to reach higher networks the basic remote sensing architecture should be independent of any complementary network and its architecture must be sufficient and efficient in providing the reliable data expected from a monitoring and alerting solution. It is important to understand that the network itself is independent of any third-party provider. However, the network sink still needs to communicate to a higher network or a back-bone infrastructure in order to route the data sensed from the field to a central workstation for data processing. The expandability discussed here relates to the integration of the network gateway. In order to provide the population and Civil Defense authorities other means of receiving alerts or accessing data directly from the WSN network to view current readings and the overall status of the environment at risk the network gateway must be able to communicate with other networks such as cellular networks, 802.11 WLANs and the internet. These add-on features enhance the solution by insuring that data dissemination reaches the maximum number of people in the least amount of time when an event occurs.
4.2.4 Cell Design

As presented in Figure 13, the Monitoring Subsystem, often located in the inhabited or rural areas, performs data acquisition of all relevant variables and incorporates internal communication links that allow the transmission of information from the spatially scattered locations to an interfacing port.

![Figure 13: System Concept Block Diagram](image)

The Communication Subsystem manages the transmission of the collected information from the Monitoring Subsystem to the urban area and assumes the role of an interfacing bridge. Physically, the reception point can be a “Local Office”. For example, the local office could be the nearest Civil Defense office or Fire Department where the data can be collected and analyzed. The data collection and analysis must be handled by a system with sufficient computational resources and storage capacity.
Based on the processed data an Alerting Subsystem is responsible for generating alerting messages that can be broadcast by different means. Figure 14 depicts the data acquisition and data dissemination of the proposed solution. The basic cell architecture for natural disaster monitoring is comprised of a minimum set of application-specific sensors and a network gateway connected to a back-bone structure. This minimum set of sensors contains all sensors required to collect data identified as relevant to the prediction and the formation related to the disaster. The network gateway must serve as a local data storage unit in addition to its primary purpose of transmitting the sensed data from the field to a central processing station. The solution can be scaled depending on the required coverage area. For example, adjacent cells for flash-flood monitoring can be formed along the river and the surrounding regions as illustrated in Figure 15.
4.3 Central Processing Office

The concept of a “Local Office” is any location where signal processing and data analysis takes place. Civil Defense offices and Fire Departments, if not chosen to be the Local Office, must interface with the system in real-time in order to broadcast early alerts and respond to events quickly. The office must have connectivity to the remote network and, if possible, all complementary networks supported by the network gateway. This interface must be robust to avoid isolating the remote sensing network. The data collected from the remote network is transmitted to a central processing station where analysis is performed. These stations must be equipped with software capable of implementing environmental models based on the historical database created by the system. Furthermore, the software must be able to generate alarms whenever thresholds are violated. Alarm generation is initiated by the software or by human intervention.
whenever hazardous conditions are detected. Alarms generated by the computer are based on safety thresholds established by the local authorities and first responders. The Local Office personnel can manually generate alarms if conditions are hazardous but the condition has not been detected due to node failure or vandalism for example. Depending on the infrastructure available at the Local Office, data dissemination may take different routes to reach the general public. Mass media is the most effective method to disseminate information since it reaches the largest number of people in the least amount of time. For people that are on the move another possibility includes the telecommunication media. Broadcasting of alert message would describe the current status of the event and suggest an action to be taken such as evacuate, seek shelter or return home.

4.4 A WSN Applied to Natural Disasters

Based on the imposed requirements, this research proposed the use of a WSN to provide the remote sensing of any environment for natural disaster monitoring. Some commonalities found in a WSN for natural disaster monitoring and general WSN applications are:

- The sensing nodes must work in uninhabited environments. Therefore, nodes have to remain functional for long periods of time without human intervention. The unattended nature of the network requires the solution to be energy efficient in order to prolong the lifetime of the network. Energy optimization at all levels is regarded as the primary goal in WSN design.
• The solution needs to be flexible and extendable to accommodate network growth and topological changes. The number of nodes may increase at any time to improve redundancy or simply to expand the monitored region. Changes in topology are common in a WSN since they are subject to weather and nature-originated inclemencies. The network routing protocol needs to include node discovery strategies and a self-forming capability in order to enable new nodes to join the network.

• The network discussed can be regarded as a Distributed System since sensors have to act cooperatively to provide the collected data to bridge-nodes, to identify internal failures and to adapt to changes in topology. At the same time sensor nodes have to collectively and dynamically use mechanisms that maximize the lifetime of the network such as those proposed in, [18].

• Exploitation of the processing power of the nodes cannot be overlooked. Localized processing algorithms transmit only useful data, which reduces data rate requirements, network congestion and transmission power consumption.

In order to adapt the generic concepts of Wireless Sensor Networks to the specific application presented in this thesis, the development of a monitoring and alerting system was designed as is explained in the following chapters.
CHAPTER 5

SENSOR NODES

The MICA2 and MICA2DOT “motes” were used as sensor nodes for the Wireless Sensor Network. These motes were configured according to application-specific requirements to collect, process and transmit information from remote locations to the network gateway. As data acquisition devices the motes are equipped with transducers to sense the environment at predetermined intervals or in an event-driven fashion.

However, a major challenge in the design of sensing nodes is the selection of suitable application-specific sensors. Since low-power operation is one of the main requirements for a WSN, transducers must also operate with low power. In addition, the operating voltage is restricted to the range from 2.5 to 5.0V since all parts of the system are powered by the same types of batteries. In order to compensate for the additional load that the transducers represent to the mote the sensor devices are only activated when a measurement reading is scheduled. Passive sensors with high impedances are preferred to reduce the current draw when operating in the “ON” mode. The following sections describe the hardware architecture for all types of motes used in flash-flood monitoring and the integration of custom transducers to the system.
5.1 Data Acquisition Requirements for Flash-Flood Monitoring

As described in the system concept, the disaster must be well understood for the definition of parameters to be monitored. According to the analysis presented in section 4.2.1, the following magnitudes need to be collected from the sensing field in order to monitor flash-floods:

- Precipitation,
- Wind Speed,
- Wind Direction,
- Soil Moisture,
- Water Level,
- Water Flow,
- 2-Axis Seismic Accelerometer,
- Temperature,
- Pressure,
- Humidity,
- Luminosity.

5.2 Node Types

The transducers were grouped together in three categories to form environmental nodes that comprise the flash-flood monitoring system. These categories are defined as:

- **Meteorological nodes**: Meteorological nodes are positioned in the surrounding fields of the river that causes the flood. These nodes are
responsible for monitoring luminosity, temperature, humidity, barometric pressure, wind direction, wind speed and precipitation. These nodes have the function of measuring meteorological conditions that are characteristics before flash-floods.

- **Hydrological nodes**: Hydrological nodes are located in the shore along the river. These nodes monitor water level and water flow. These magnitudes are critical during the formation of flash-floods.

- **Seismic nodes**: Seismic nodes are strategically placed in hazardous locations in the neighboring mountains. These nodes collect soil moisture and 2-axis accelerometer magnitudes that indicate seismic movement. Since the occlusion of the river has been identified as the main triggering effect for flash-floods it is mandatory that these magnitudes be monitored.

Motes serve as low-power wireless data acquisition devices. This is possible by coupling transducers to the data acquisition boards and interfacing them to the motes through proper signal conditioning circuits. In addition to these environmental nodes, repeater nodes are placed in the cell architecture to enhance communication performance and overall system robustness. Table 3 describes the magnitudes that need to be monitored and the commercial-off-the-shelf, (COTS), sensor boards that may be used. The use of COTS sensor boards is very practical since they incorporate all the sensors and the signal conditioning modules that are required to measure specific variables.
Table 3: Sensing Parameters and COTS Equipments

<table>
<thead>
<tr>
<th>Magnitude</th>
<th>MTS420CA Sensorboard</th>
<th>MTS510CA Sensorboard</th>
<th>Custom sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precipitation</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Wind Speed</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Wind Direction</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Soil Moisture</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Water Level</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Water Flow</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Sound</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2-axis Accelerometer</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Temperature</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Pressure</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Humidity</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Luminosity</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

As Table 3 indicates, there are many magnitudes that can be measured using sensor boards. However, there are some that require customized signal conditioning circuitry.

5.3 Signal Conditioning Techniques

The custom transducers used for flash-flood monitoring were classified according to their signal conditioning method as:

- Those that generate switching pulses. Such was the case for the precipitation, water speed and wind speed transducers,
- Those whose transduction principle is based on resistive changes according to the measured variable. Such was the case for the water level, soil moisture, and wind direction transducers.
5.3.1 Pulse Generating Sensors

The pulse generating transducers are responsible for sensing precipitation, water speed and wind speed.

Precipitation

The “Davis” Rain Collector II, pictured in Figure 16, was used to sense precipitation in the field. The number of bucket tips corresponds to pulses that are measured and counted by the input digital channel of a DAQ board. The rainfall calibration number, \( \text{CAL} \), was used to determine the amount of water each bucket tip represented. For the 0.01 inch rain collector calibration, CAL was equal to 100. This means that the bucket tips and records rainfall for every 0.01" of rain. The calculated rainfall is obtained as

\[
\text{Calibrated Rainfall} = \text{Number of Pulses} \cdot \frac{1}{\text{CAL}}
\]  
(1)

Figure 16: “Davis” Rain Collector II
Anemometer

The “Davis” cup-type anemometer, which is pictured in Figure 17, measures wind speed based on the revolutions per minute of the transducer’s arms. Each revolution is detected by magnetic switches and counted as pulses by the digital channel of the data acquisition board. The value is processed and then converted to wind gust and average wind speed readings. Unit transformations were implemented in the application software to output engineering units. These transformations were formulated as

\[
\text{Wind Gust} = \text{counter} \cdot (1.81) + 0.5
\]

\[
\text{Avg Wind Speed} = \text{counter} \cdot \frac{3.62}{3} + 0.5
\]

Figure 17: “Davis” Anemometer

Water Flow

The “Swoffer” Fiber-Optics Water flow transducer, which is pictured in Figure 18, measures water velocity based on the number of turns of the propeller rotor. Each turn generates four pulses that are used to determine velocity. The calibration number represents the number of counts a specific rotor produces as it travels through 10 feet and 10 meters of still water. In order to obtain accurate measurements, the transducer must be calibrated in accordance with the formula
The schematic for the pulse signal conditioning is illustrated in Figure 19.

Figure 19: Pulse Signal Conditioning for Different Types of Transducers

5.3.2 Resistive Transducers

The transducers that used resistive signal conditioning included water level, soil moisture and wind direction. The general schematic is presented in Figure 20.
Water Level

The “InterMountain” float & pulley water level transducer, presented diagrammatically in Figure 21, operates by moving with the water level. As the water level raises or lowers the float moves up or down and turns a pulley.

Figure 21: “InterMountain” Float & Pulley Water Level Transducer

The pulley shaft is coupled to a precision 5 turn potentiometer that can register a change of 5 feet. The data acquisition board provides the transducer with 2500 mV of excitation and detects voltage changes from 0 to 2.5 VDC according to the change in the potentiometer resistance, which has a range from 0 to 5kohms. In other words, the measured voltage between the reference, which was 0 ohm, and the potentiometer leg that changed according to the water level was the voltage reading correspondent to the change in water level. The analog signals from the water level transducer were measured
by the single-ended channels of the data acquisition board that were labeled as A0 to A6. The ADC reading was converted to a voltage reading by the equation

\[
\text{Voltage} = 2.5 \cdot \frac{\text{ADC Reading}}{4096}
\]  

(5)

**Soil Moisture**

The electrical resistance type Davis soil moisture sensor, which is pictured in Figure 22, converts electrical resistance from the sensor to a calibrated reading of soil water content measured in soil water potential, which is given in bars. The principle of operation is that the resistance of electrodes embedded in a porous block is proportional to its water content. Therefore, the wetter a block, the lower the resistance measured across two embedded electrodes. This implies that the soil water potential is directly influenced by the soil temperature.

![Figure 22: “Davis” Soil Moisture Sensor](image)

Resistance and temperature maintain a linear relationship when soil water content ranges from 0 to 2 bars. The resistance measurement was normalized to degrees C by

\[
R_{21} = \frac{R_s}{[1 - (0.018 \cdot T_d)]}
\]  

(6)
In order to calculate the output resistance, a voltage divider circuit had to be implemented and interfaced with the data acquisition board. Soil Water potential, (SWP), was then calculated by

\[
SWP = (0.07407 \cdot R_{21}) - 0.03704
\]  

(7)

An excitation voltage of 2500 mV was applied and the analog signals from the transducer were read by the single-ended analog channel of the data acquisition board.

**Wind Direction**

The Davis anemometer wind direction sensor measures a rotational potentiometer and converts the value to an offset from north. A 2500 mV excitation voltage was applied by the data acquisition board and the ADC readings were converted to a voltage reading by

\[
\text{Wind Direction} = ADC \cdot \frac{356.0}{4096} + 0.5
\]  

(8)
5.3.3 Data Processing for COTS Sensor boards

Whereas hardware and drivers are available for sensor boards their function has to be analyzed for the development of applications that receive the data. In this section, the MTS420CA sensor board is used as the main example for the data processing that had to take place in order to interpret the sensor data correctly.

The MTS420CA sensor board is capable of sensing environmental parameters of temperature, barometric pressure, humidity, luminosity and 2-axis acceleration. The temperature reading from the Sensirion module was disregarded. The temperature and barometric pressure readings from the Intersema MS5534 module provided uncompensated 16-bit digital values for both parameters, which were designated by D1 and D2 respectively and compensation, which was represented by (Word1...4) was performed by an external microcontroller as illustrated in Figure 23.

![Figure 23: Pressure and Temperature Readings from Intersema MS5534 Sensor](image)

Every module was individually factory calibrated at two temperatures and two pressures. As a result, the 6 coefficients necessary to compensate for process variations and temperature variations were calculated and stored in the 64-Bit PROM of each module. These 64-Bits were partitioned into four words of 16-Bits each, read by the
microcontroller software and used in the program that converted D1 and D2 into compensated pressure and temperature values.

For the ambient light sensor, the algorithm used is described in the Taos TSL2550 datasheet, [ch\$\do5(t)aos]. The TSL2550 contains two ADC registers, which are designated as channel 0 and channel 1. Each ADC register contains two component fields that are used to determine the logarithmic ADC count values, which are designated as CHORD bits and STEP bits. The CHORD bits correspond to the most significant portion of the ADC value and specifies a segment of the piece-wise linear approximation. The STEP bits correspond to the least significant portion of the ADC count value and specify a linear value within a segment. All CHORD and STEP bits equal to zero indicates that the light level is below the detection limit of the sensor. All CHORD and STEP bits equal to one indicate that an overflow condition exists. Each of the two ADC value registers contains seven data bits and a valid bit. Table 4 explains the meaning of these bits

5.4 Sensor Drivers

Sensor drivers provide the software interface between transducers that require custom signal conditioning circuitry and the data acquisition boards. For example, this section implements the driver interface for the soil moisture transducer using the MDA300CA data acquisition board. Figure 24 presents the circuit that provides signal conditioning to the soil moisture transducer. As illustrated by the DAQ board schematic presented in Appendix D.1, the transducer has two I/O lines. The line designated INT1
corresponds to a power line for the DAQ board and the line designated ADC1 corresponds to the ADC input channel. Both lines were controlled by the software. The resistor “R” varied according to the specifications of the soil moisture transducer.

![Signal Conditioning Circuitry for the Resistive Soil Moisture Sensor](image)

**Figure 24: Signal Conditioning Circuitry for the Resistive Soil Moisture Sensor**

An application called “TestSensor”, which is depicted in Figure 25, was developed to test the driver. The diagram, which was generated using the make mica2.docs command and modified slightly for better clarity, shows the components that were written to test the sensor driver. All red components are part of the TinyOS operating system whereas the blues components were implemented for this particular application.

![TestSensor Application Wiring](image)

**Figure 25: TestSensor Application Wiring**
For simplicity, the application only scheduled a timer, which read the channel every time the timer expired and blinked the LEDs as it was executing. The application consisted of the configuration wires together with the “TestSensorM” module, the “SoilDriver” driver, and two TinyOS components termed “TimerC” and “LedsC”. The module controlled the initialization, start and stop of the components, the start of the clock and the activation of an event to measure the sensor.

The “SoilDriver” driver, written in nesC, mapped the I/O lines of the soil moisture transducer into TinyOS software commands and controlled the sensor using TinyOS components that composed the driver. The “SoilDriver” configuration wired two components termed “SoilDriverM” and “ADCC”. The “SoilDriverM” module contained the code written for the implementation of the driver. The “ADCC” component controlled the A/D converter. The SoilDriver component could be decomposed into more internal components. The SoilDriver configuration wiring is illustrated in Figure 26.

![Figure 26: SoilDriver Configuration Wiring](Image)

**5.5 Test Software**

A LabView application was developed in order to verify the functionality of the MTS420CA sensor boards. It only reads the Intersema MS5534 sensor readings for
humidity and pressure, the Taos light sensor and the battery voltage information contained in the data packet. This approach certified that the sensors were working properly prior to deployment in the remote field. Similar applications could be developed for the other sensors attached to any mote since the concept does not change. The concept is illustrated in Figure 27. The MIB510 programming board must be connected to the computer’s serial port for data transfer. A MICA2 mote serving as the base station was attached to the MIB510 board and a remote MICA2 mote was coupled with the MTS420 sensor board that sent data packets wirelessly to the base station. The application read the raw data packets coming in to the serial port, parsed the data stream and decoded the sensor readings in hexadecimal format into engineering units.

Figure 27: Testing Configuration for Sensor Devices Prior to Deployment

The IOInstrumentation.vi was an assistant to the application that communicated with the serial port without the need for a driver. After communication was established the I/O assistant read and parsed the incoming packets. The data was manually parsed into tokens and assigned variables according to the packet structure. A typical data packet read from the serial port and parsed by the I/O assistant is depicted by:

```
7e000201d860101007d019803bf1aacc1eabe3aacad090686545a300000002e50100.
```
Table 4 describes the data packet format.

Table 4: Data Packet Structure for the MTS420 Sensorboards

<table>
<thead>
<tr>
<th>Packet Byte Name</th>
<th>Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOS Header = 7e00007d1d</td>
<td></td>
</tr>
<tr>
<td>UART Address</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>7e00</td>
</tr>
<tr>
<td>Group ID</td>
<td>00</td>
</tr>
<tr>
<td>Data Payload = 86010100</td>
<td></td>
</tr>
<tr>
<td>Sensorboard ID</td>
<td>86</td>
</tr>
<tr>
<td>Packet ID</td>
<td>01</td>
</tr>
<tr>
<td>Node ID</td>
<td>01</td>
</tr>
<tr>
<td>Reserved</td>
<td>00</td>
</tr>
<tr>
<td>Sensorboard Reading = 7d019803bf1aacc1eabe3aacad090686545a30000000002e50100</td>
<td></td>
</tr>
<tr>
<td>Battery Voltage</td>
<td>017d</td>
</tr>
<tr>
<td>Sensirion Humidity</td>
<td>0398</td>
</tr>
<tr>
<td>Sensirion Temperature</td>
<td>0dca</td>
</tr>
<tr>
<td>Compensation Word#1</td>
<td>0labf</td>
</tr>
<tr>
<td>Compensation Word#2</td>
<td>0c9ac</td>
</tr>
<tr>
<td>Compensation Word#3</td>
<td>0abe</td>
</tr>
<tr>
<td>Compensation Word#4</td>
<td>0ae3</td>
</tr>
<tr>
<td>Intersema Humidity</td>
<td>0d0ca</td>
</tr>
<tr>
<td>Intersema Temperature</td>
<td>0ae3</td>
</tr>
<tr>
<td>TAOSCH0</td>
<td>06890</td>
</tr>
<tr>
<td>TAOSCH1</td>
<td>4565</td>
</tr>
<tr>
<td>AccelX axis</td>
<td>00000</td>
</tr>
<tr>
<td>AccelY axis</td>
<td>02000</td>
</tr>
</tbody>
</table>

The front panel of the application captured current and past readings from the remote nodes for analysis. The environmental monitoring parameters extracted from the MTS420CA sensor board were temperature, pressure, and luminosity. Since the environmental readings do not change much over short periods of time, a histogram was available to better evaluate the changes over longer periods. In addition, the lifetime of the network was proportional to the battery lifetime. Therefore, voltage readings were also included in the data packets sent from the remote nodes to the base station. It is important to continuously monitor the battery status of the nodes to ensure that they remain within their operating voltage range. The front panel is pictured in Figure 28.
Figure 28: Front Panel of the LabView Application

Used to Test the MTS420CA Sensor board

The block diagram of the application is presented in Figure 29 and the implementation of the application is presented in Figure 30. Figures 29 and 30 present implementations of the algorithms, which were described by the sensor manufacturer and presented in Appendix B.1 and Appendix C.1.
Figure 29: Block Diagram of the Implementation of the Sensor Algorithms
5.6 Cluster Configuration

Based on the requirements imposed by the solution concept, the cell architecture for flash-flood disasters evolved as illustrated in Figure 31. Civil Defense authorities in the city of Maracay, Venezuela expressed an interest in placing at least one transducer for each type of magnitude identified in the section titled “Data Acquisition Requirements..."
for Flash-Flood Monitoring”. Therefore, the basic cell architecture for flash-flood monitoring consisted of:

1. Stargate network gateway,
2. Meteorological Motes,
3. Hydrological Mote,
4. Seismic Mote,
5. Repeater Motes.

5.6.1 Meteorological Mote

Two MICA2 motes, illustrated in Figure 32, were used to implement the meteorological nodes. One mote had the MTS420CA sensor board attached to the 51-pin connector interface to collect light, temperature, humidity and barometric pressure.
readings. The other mote had the MDA300CA Data Acquisition board attached to the
connector interface and coupled to custom sensors that collected precipitation, wind
direction, and wind speed readings. The DAQ board was able to sustain the channel
requirements for all the transducers. Therefore, only one DAQ board was necessary.
The precipitation and wind speed transducers used one digital channel each and the wind
direction transducers used one single-ended analog channel. Table 5 describes the list of
COTS transducers used to compose this type of mote.

![Image of Block Diagram of the Meteorological Motes]

**Figure 32: Block Diagram of the Meteorological Motes**

**Table 5: Transducers Coupled with the Meteorological Mote**

<table>
<thead>
<tr>
<th>Transducer</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind Direction</td>
<td>Davis Std. Anemometer</td>
</tr>
<tr>
<td>Wind Speed</td>
<td>Davis Std. Anemometer</td>
</tr>
<tr>
<td>Precipitation</td>
<td>Davis Rain Collector II</td>
</tr>
<tr>
<td>Luminosity</td>
<td>Taos TSL2550</td>
</tr>
<tr>
<td>Temperature</td>
<td>Intersema MS5534</td>
</tr>
<tr>
<td>Barometric Pressure</td>
<td>Intersema MS5534</td>
</tr>
<tr>
<td>Humidity</td>
<td>Sensirion SHT11</td>
</tr>
</tbody>
</table>
5.6.2 Hydrological Mote

The MICA2 mote coupled with the MDA300 Data Acquisition board and the custom water level and water flow sensors were used to implement this node. The hydrological mote is depicted in Figure 33. The water level transducer used one single-ended analog channel and the water flow transducer used one digital channel. Table 6 describes the transducers used for this mote.

![Figure 33: Block Diagram of the Hydrological Mote](image)

Table 6: Transducers Coupled with the Hydrological Mote

<table>
<thead>
<tr>
<th>Transducer</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water Level</td>
<td>IEI Float &amp; Pulley and Pulley sensor</td>
</tr>
<tr>
<td>Water Speed</td>
<td>Swoffer</td>
</tr>
</tbody>
</table>

5.6.3 Seismic Mote

Figure 34 illustrates the two MICA2DOTs that were used to implement these nodes. One mote was coupled to the MDA500 Data Acquisition board along with soil moisture whereas the other mote was coupled with the MTS510CA sensor board. Table 7 describes the transducers used for this mote.
5.6.4 Repeater Mote

Along with the nodes mentioned above, MICA2 motes were used to implement the repeater nodes. They were introduced to the network as redundant nodes and used primarily to ensure that all nodes within the cell could form a mesh network. The repeater nodes were important in the communication and power-saving schemes. They present themselves as alternate datapaths in the mesh topology, expand the routing table and ensure non-Line-of-Sight connectivity. Furthermore, the increased number of nodes alleviates communication overload on certain nodes of the topology and extends the battery life of each individual node, which extends the lifetime of the network.
CHAPTER 6

POWERING TECHNIQUES FOR MOTES AND STARGATE

Natural disaster monitoring systems usually operate in unattended regions where power grids are not available and access to the location is difficult. Therefore, motes must be self-powered and should include energy awareness as a part of their operation. Power saving strategies used for such a monitoring system includes reducing the duty cycle, using long-lasting batteries and coupling the motes with solar panels as an external power resource.

6.1 Battery Power for Motes

The MICA2 motes were originally designed to be powered by two AA, 1.5V, alkaline type A91 batteries. These resources provide the motes with 3V and enough power to operate the CC1000 transceiver, the Atmel processor and the sensor devices. The motes have guaranteed operation down to 2.7V, [1]. Below this threshold the functionality of the mote is compromised. Most sensors and I/O devices do not operate below 2.5V. The CC1000 radio transceiver does not operate under 2.1V and the microprocessor operates at most around 2.2 to 2.3V. One of the most important issues in natural disaster monitoring involves the fundamental tradeoff of sensitivity verses false
alarms. When this safety threshold has been reached the accuracy of the sensed data is compromised, which puts at risk the reliability of the system. The operating voltage range provided by two Alkaline AA batteries, from 2.7 to 3V, was concluded to be very narrow.

Experiments involving the use of three AA, 1.2V and Rechargeable Nickel-Metal Hydride batteries were performed to verify how long the mote lifetime could be extended. This scheme increased the operating voltage range from 2.5 to 3.6V.

An experiment was conducted that involved two AA, 1.5V, Panasonic Industrial Alkaline batteries to power the motes. Then an experiment was conducted that involved three AA, 1.2V, and Rechargeable Nickel-Metal Hydride batteries as the power source. For both schemes, two models with different duty cycles of 1% and 0.5% were investigated. The initial configuration of the experiment, as presented in Figure 1, used the two Alkaline AA batteries with a power supply load ranging from 10 to 15mA.

![Figure 35: Powering Scheme Using Two AA Alkaline Batteries](image)

Considering a constant discharge rate from both batteries in conjunction with the fact that at 2.7V the functionality of the mote is compromised, each cell was only functional to the system until it discharged to 1.35V. At full operation, which was at a 100% duty cycle, and a power supply load of 10mA the MICA2 mote lasted for approximately 170 hours.
This configuration provided 1700 mAh of battery capacity for the mote. This battery discharge characteristic is presented in Figure 36. By reducing the duty cycle of the mote to less than 1%, the expected lifetime for the same mote, using the same pair of batteries was dramatically increased. For model 1, operating at a, the mote lasted for 10.45 months. For model 2, operating at a 0.5% duty cycle, the mote almost doubled the 1% duty cycle lifetime and functioned for 19.08 months.

![Typical Discharge Characteristics With Constant Current At 68 F (20 C)](image)

Figure 36: Discharge Characteristics for 1.5V Panasonic Industrial Alkaline Batteries

Figure 37 describes the system specifications for the battery lifetime verses duty cycle modeling for both experiments. Since the logger was not used for this application, the radio was by far the most power consuming module on the mote. From the total current of 0.2169 mAh used in one hour the radio draws approximately half of the current, which was 0.0920 mAh.
Figure 37: System Characteristics Investigated for the Two Powering Schemes

Figure 38 illustrates the battery lifetime for both models as a function of battery capacity.
The second experiment, which is illustrated in Figure 39, used three AA, 1.2V, and Rechargeable Nickel-Metal Hydride batteries. Based on a 500mA or 0.2C discharge rate, the typical average capacity for the cell approximated 2500mAh. At full operation with a
power supply load of 10mA the MICA2 mote lasted for 250 hours. At a 1% duty cycle, the mote lasted for 15.79 months.

![Figure 39: Powering Scheme Using Three AA NiMH Batteries](image)

Finally, at a 0.5% duty cycle, the battery lifetime of the mote was extended to 29.54 months. This represents more than 10 months of extended operation at a 0.5% duty cycle and approximately 5 months at a 1% duty cycle when compared to the first experiment. The second configuration increased the total voltage across the MICA2 mote to 3.6V. Making the same assumption that the discharge rate is constant for all three batteries and not violating the functional threshold of the mote of 2.7V each cell now remained useful to the system until it discharged to 0.9V. By lowering the minimum voltage requirement from 1.35 to 0.9V, the mote was able to drain each cell for longer periods of time, which increased battery capacity. This configuration would allow the battery lifetime to be extended for nearly 2 ½ years if operating at a 0.5% duty cycle. The characteristics of the cell for the second experiment are presented in Figure 40.
From these experiments, it was concluded that in order to achieve multi-year performance from the Wireless Sensor Network motes should sleep the majority of the time. Reducing the operating duty cycle of the motes to a minimum is required to obtain a long lifetime from any type of batteries. The three battery configuration was proven to be the most adequate for remote sensing since the motes remain unattended for long periods of time and maximizing battery lifetime reduces operating costs and increases the network autonomy. It is important to emphasize that for this type of application, since the environment does not change rapidly, even with a low duty cycle, events are not missed.

Figure 40: Discharge Characteristics for 1.2V Energizer Rechargeable NiMH Batteries
6.2 Powering the Stargate Board

The Stargate computer provides the communication link between the remote sensing network and the higher network responsible for data dissemination. The importance of this component in the design concept makes the selection of a robust powering scheme fundamental for the proper operation of the monitoring system. In the selection process, there were two major concerns:

- The Stargate board operates in ACTIVE mode at all times
- The Stargate board must operate unattended for long periods of time.

Table 8: Stargate Computer Current Draw in Different Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Current Draw (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>180</td>
</tr>
<tr>
<td>Sleep</td>
<td>60</td>
</tr>
<tr>
<td>Active</td>
<td>330</td>
</tr>
<tr>
<td>Idle/PCMCIA</td>
<td>360</td>
</tr>
<tr>
<td>Active/PCMCIA</td>
<td>530</td>
</tr>
</tbody>
</table>

Based on these requirements solar panels were chosen to power the gateway. Figure 41 illustrates the components involved in powering the Stargate using solar energy.
Figure 41: Architecture to Provide Solar Energy for the Stargate Computer

The solar module was connected to a charge controller in order to regulate the voltage and the current to feed the battery at the right state-of-charge, (SOC). Since the controller only operates with 12V loads a DC/DC converter was used to interface with the Stargate computer. Linear converters were not used because of their low efficiency. Instead, a switching DC/DC converter with 85% efficiency was implemented in the design. The powering subsystem of the Stargate computer included a linear regulator that down converted incoming voltages to the different component’s operating voltage. The regulator could handle input voltages from 5V and above. However, since the linear regulator was not equipped with a heat sink and considering that the current could be as high a 0.5A it was not advisable to input voltages higher that 6V for long periods of time. Such a situation would represent a power of 0.5W, ((6V-5V)*0.5A), which would have to be dissipated by the linear regulator. Operating at 5V and assuming the worst case conditions with the Stargate in ACTIVE mode at all times and the cellular card ON, the peak power consumption was
Peak Power = \((530mA) \times (5V) = 2.65\text{Watts}\) \hspace{1cm} (9)

Based on the 85% efficiency of the Stargate the input power requirement for the DC/DC converter was

\[
Pin = \frac{2.65}{0.85} = 3.12\text{Watts}
\] \hspace{1cm} (10)

For continuous operation, throughout one week, the Stargate operated for 168 Hrs/Wk, which produced a total watt-hour-per-week of

\[
\text{Total Wh/Wk DC} = 3.12 \times 168 = 524.16\text{Wh/Wk}
\] \hspace{1cm} (11)

The total amp-hour-per-week based on the 12V DC/DC converter supply was

\[
\text{Total Ah/Wk} = \frac{524.16}{12} = 43.68\text{Ah/Wk}
\] \hspace{1cm} (12)

In one day, the total amp-hour was

\[
\text{Total Ah/day} = \frac{43.68}{7} = 6.24\text{Ah/day}
\] \hspace{1cm} (13)

Authorities in Maracay suggested that flash-flood formation takes an average of 3 days. Considering the presence of cloudy weather during this period, the system was designed to remain autonomous for the same number of days. Thus, the system needed to store

\[
\text{Total Ah} = 6.24 \times 3 = 18.72\text{Ah}
\] \hspace{1cm} (14)
Dividing the amount of amp-hour of storage by the battery discharge rate yields the theoretical battery bank size. To diminish the aging affect of deep discharge, a 30% depth of discharge was chosen. Therefore, the battery capacity becomes

$$\text{Battery Capacity} = \frac{18.72}{0.3} = 62.40 \text{Ah}$$  \hspace{2cm} (15)

Another factor that needed to be considered when sizing the battery was the winter time ambient temperature. During cold weather the battery bank experiences charging limitations. Table 9 presents the required multipliers based on temperature. These values need to be multiplied by the battery capacity to ensure that the batteries will be able to overcome cold weather effects. For example, in Maracay, the multiplier was 1.1 based on ambient temperature during the winter. Therefore, the total battery capacity required was

$$\text{Battery Capacity} = 62.40 \times 1.1 = 69.26 \text{Ah}$$  \hspace{2cm} (16)

<table>
<thead>
<tr>
<th>Temperature (F)</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>1.00</td>
</tr>
<tr>
<td>70</td>
<td>1.04</td>
</tr>
<tr>
<td>60</td>
<td>1.11</td>
</tr>
<tr>
<td>50</td>
<td>1.19</td>
</tr>
<tr>
<td>40</td>
<td>1.30</td>
</tr>
<tr>
<td>30</td>
<td>1.40</td>
</tr>
<tr>
<td>20</td>
<td>1.59</td>
</tr>
</tbody>
</table>

Table 9: Multipliers Based on Winter Time Ambient Temperature

The “Universal Battery” battery model UB12750 has a 75 Ah capacity, which matches the requirements for this application. Therefore, only one battery would be required.
Based on these specifications, the number of solar modules required to power the Stargate can be determined.
Wireless Sensor Network field tests were performed to verify network behavior in different environments. Two different sites were investigated. A network with nine motes was deployed and communication quality and network statistics parameters were recorded. The motes associated with the nine mote deployment were operated at a 100% duty cycle, which meant they were “always on”. Motes #1 through #7 used the 3-battery configuration whereas motes #8 and #9 used the 2-battery configuration. Prior to deployment, experiments were performed at the USF campus. The network was established outside the Engineering II building in an open field with direct Line-of-Sight, (LOS), communication among all the motes. Motes were placed one meter above the ground with an average distance of 15 meters between motes. It is important to place the motes at least one meter off the ground because of the antenna height. Otherwise, the communication range drops to approximately one tenth of the maximum range, which was 1000 ft. Figure 42 illustrates the test network topology, which represented a relatively flat environment. Results in such a flat environment yielded reliable fidelity between transmitted and received packets. Table 10 data verifies that the motes remained within the desired voltage operating range of 2.7V to 3.9V. Table 10 presents and relates the most relevant parameters extracted from the USF site. The flat field allowed link
qualities that approached 100%, which means that nearly all transmitted packets were received. As a consequence, BER results were negligible.

Figure 42: Testing Network Statistics for the Sensor Network at USF
Table 10: Communication Quality and Network Statistics for the USF Experiment

<table>
<thead>
<tr>
<th>Node #</th>
<th>Packets Received</th>
<th>Packets Sent</th>
<th>Link Quality</th>
<th>BER</th>
<th>Battery Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1224</td>
<td>1224</td>
<td>1</td>
<td>0</td>
<td>3.42828</td>
</tr>
<tr>
<td>1</td>
<td>1219</td>
<td>1234</td>
<td>0.93589</td>
<td>0.06410</td>
<td>3.65847</td>
</tr>
<tr>
<td>2</td>
<td>1130</td>
<td>1213</td>
<td>0.93157</td>
<td>0.06842</td>
<td>3.57463</td>
</tr>
<tr>
<td>3</td>
<td>1070</td>
<td>1175</td>
<td>0.91063</td>
<td>0.08936</td>
<td>3.70652</td>
</tr>
<tr>
<td>4</td>
<td>1211</td>
<td>1241</td>
<td>0.97582</td>
<td>0.02417</td>
<td>3.78328</td>
</tr>
<tr>
<td>5</td>
<td>1239</td>
<td>1256</td>
<td>0.98646</td>
<td>0.01353</td>
<td>3.76102</td>
</tr>
<tr>
<td>6</td>
<td>1230</td>
<td>1240</td>
<td>0.99193</td>
<td>0.00806</td>
<td>3.82254</td>
</tr>
<tr>
<td>7</td>
<td>1059</td>
<td>1160</td>
<td>0.91293</td>
<td>0.08706</td>
<td>3.84675</td>
</tr>
<tr>
<td>8</td>
<td>1216</td>
<td>1242</td>
<td>0.97906</td>
<td>0.02093</td>
<td>2.82038</td>
</tr>
<tr>
<td>9</td>
<td>1205</td>
<td>1225</td>
<td>0.98367</td>
<td>0.01632</td>
<td>2.99908</td>
</tr>
</tbody>
</table>

The same experiment was performed in the mountains surrounding Maracay, Venezuela, which is a region constantly under the threat of flash-floods. As pictured in Figure 43, this environment presented several issues that needed to be considered before deployment.

Figure 43: Deployment Site in Maracay, Venezuela
The network topology for this type of environment has to rely on multihopping routing schemes since Line-of-Sight communication was not possible. The dense vegetation and the rocky terrain scatter and sometimes block communication between nodes. In addition, the traffic on the road increased the frequency selectivity of the wireless channel. However, in some situations these adverse characteristics actually help the communication. The multipath reflections created by the mountains and the vegetation create additional datapaths for the network. These reflections are beneficial in reaching hidden motes. Each mote was placed approximately 1.5 meters above the ground. The results from communication quality and network statistics, which are presented in Table 11, validated the use of a Wireless Sensor Network in such an environment.

Table 11: Communication Quality and Network Statistics for the Maracay Experiment

<table>
<thead>
<tr>
<th>Node #</th>
<th>Packets Received</th>
<th>Packets Sent</th>
<th>Link Quality</th>
<th>BER</th>
<th>Battery Volrage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1039</td>
<td>1039</td>
<td>1</td>
<td>0</td>
<td>3.75646</td>
</tr>
<tr>
<td>1</td>
<td>998</td>
<td>1045</td>
<td>0.95502</td>
<td>0.04497</td>
<td>3.58694</td>
</tr>
<tr>
<td>2</td>
<td>1078</td>
<td>1198</td>
<td>0.89983</td>
<td>0.10016</td>
<td>3.05847</td>
</tr>
<tr>
<td>3</td>
<td>970</td>
<td>1208</td>
<td>0.80298</td>
<td>0.19701</td>
<td>3.57465</td>
</tr>
<tr>
<td>4</td>
<td>1011</td>
<td>1156</td>
<td>0.87456</td>
<td>0.12543</td>
<td>3.24847</td>
</tr>
<tr>
<td>5</td>
<td>937</td>
<td>1129</td>
<td>0.82993</td>
<td>0.17006</td>
<td>3.36465</td>
</tr>
<tr>
<td>6</td>
<td>969</td>
<td>1287</td>
<td>0.75291</td>
<td>0.24708</td>
<td>3.57466</td>
</tr>
<tr>
<td>7</td>
<td>1028</td>
<td>1145</td>
<td>0.89781</td>
<td>0.10218</td>
<td>3.51872</td>
</tr>
<tr>
<td>8</td>
<td>1045</td>
<td>1206</td>
<td>0.86650</td>
<td>0.13349</td>
<td>2.73634</td>
</tr>
<tr>
<td>9</td>
<td>972</td>
<td>1284</td>
<td>0.75700</td>
<td>0.24299</td>
<td>2.94347</td>
</tr>
</tbody>
</table>

Even in harsh environmental conditions such as the one found at the site in Venezuela, the network performed with satisfactory quality. The average link quality was 86.365% and the average BER was 0.13634%. To improve the results repeater motes could be added to the network. Such additions would create additional datapaths,
reduce communications range and increase the network lifetime since the network load would be distributed among a greater number of nodes.

A final experiment investigated the effects on link quality of increasing network density. Three repeater nodes, using the 3-battery configuration, were added to the network. This experiment was performed at USF. Table 12 presents the results. When compared to Table 10, the additional datapaths are shown to have prevented packet loss and improved link quality.

Table 12: Communication Quality and Network Statistics for the USF Experiment

<table>
<thead>
<tr>
<th>Node #</th>
<th>Packets Received</th>
<th>Packets Sent</th>
<th>Link Quality</th>
<th>BER</th>
<th>Battery Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1297</td>
<td>1297</td>
<td>1</td>
<td>0</td>
<td>3.73434</td>
</tr>
<tr>
<td>1</td>
<td>1203</td>
<td>1255</td>
<td>0.95856</td>
<td>0.04143</td>
<td>3.53232</td>
</tr>
<tr>
<td>2</td>
<td>1187</td>
<td>1233</td>
<td>0.96269</td>
<td>0.03731</td>
<td>3.82343</td>
</tr>
<tr>
<td>3</td>
<td>1163</td>
<td>1179</td>
<td>0.98643</td>
<td>0.01357</td>
<td>3.84374</td>
</tr>
<tr>
<td>4</td>
<td>1226</td>
<td>1249</td>
<td>0.98159</td>
<td>0.01841</td>
<td>3.64343</td>
</tr>
<tr>
<td>5</td>
<td>1173</td>
<td>1216</td>
<td>0.96464</td>
<td>0.03536</td>
<td>3.78434</td>
</tr>
<tr>
<td>6</td>
<td>1181</td>
<td>1202</td>
<td>0.98253</td>
<td>0.01747</td>
<td>3.43724</td>
</tr>
<tr>
<td>7</td>
<td>1125</td>
<td>1133</td>
<td>0.99294</td>
<td>0.00706</td>
<td>3.65475</td>
</tr>
<tr>
<td>8</td>
<td>1217</td>
<td>1235</td>
<td>0.98543</td>
<td>0.01457</td>
<td>3.10765</td>
</tr>
<tr>
<td>9</td>
<td>1098</td>
<td>1128</td>
<td>0.97340</td>
<td>0.02659</td>
<td>3.02543</td>
</tr>
<tr>
<td>10</td>
<td>1127</td>
<td>1198</td>
<td>0.94073</td>
<td>0.05926</td>
<td>3.43627</td>
</tr>
<tr>
<td>11</td>
<td>1192</td>
<td>1216</td>
<td>0.98026</td>
<td>0.01974</td>
<td>3.45361</td>
</tr>
<tr>
<td>12</td>
<td>1157</td>
<td>1185</td>
<td>0.97637</td>
<td>0.02363</td>
<td>3.53232</td>
</tr>
</tbody>
</table>
CHAPTER 8

NETWORK GATEWAY: STARGATE CONFIGURATION

The Stargate board is responsible for performing the functionalities described in Chapter 4 of a WSN network gateway for natural disaster monitoring. The Stargate participates in all phases of the monitoring and alerting system:

- Network Sink: It collects and locally stores the data sensed from the network,
- Communication Link: It transmits the database files from the remote location to a central workstation for further processing,
- Gateway for higher networks: It is connected to backbone infrastructures such as the internet, 802.11 WLAN, and the cellular network. Such connectivity extends the range of communication from the network to the Local Office,
- Alarm Generation: It generates alarms to Civil Defense authorities if there is communication breakage from the remote network to the Local Office.

8.1 Local Database

The data sensed from the remote field was collected by the base station and processed by the Stargate computer. This application reads the SPI port of the base
station, (MICA2 mote), decodes the data packets based on mote-specific algorithms and writes the parameters into a relational database server that is stored in the network gateway. The server then communicates with the Local Office database client through query requests or by triggers. Queries may be generated by the client database at any time. Since the server is continuously monitoring client requests local authorities have the flexibility to download database files at their convenience. The other possibility is through the generation of triggers based on alarm levels. As the alarm level increases, the transmission interval decreases. The alarm levels and the fuzzy logic algorithm that determine transmission intervals will be discussed in the following section. Regardless of how the transmission is initiated, the server database generates a comma-separated-value, (csv), file and via the cellular network sends the information to the client database at the Local Office. At the client end database files are stored in relational databases and later used in applications such as LabView and Geographical Information Systems, (GIS).

The TinyOS message structure, which is illustrated in Figure 44, dynamically allocates memory space in the data packet payload according to the type of mote transmitting the message and the number of transducers coupled to the device. Thus, each type of mote has its unique packet length where the payload includes only the parameters sensed by that mote. The database server checks the mote ID on the packet to identify the algorithm to be used in the parsing and decoding of the packet.

![TinyOS Message Structure with Dynamic Payload Length](image)

Figure 44: TinyOS Message Structure with Dynamic Payload Length
The uniform structure in which data is stored allows the creation of dynamic tables that adjust themselves depending on the number of magnitudes such as temperature, humidity, water flow and pressure that are being monitored. Database tables were arranged in a similar fashion to Tables 10 and 11 but with additional fields to store the transducers readings. Therefore, both the server and client have tables with the same format. Assuming that:

- NodeID 0: Base station + Stargate gateway,
- NodeID 1: Meteorological Mote #1,
- NodeID 2: Meteorological Mote #2,
- NodeID 3: Hydrological Mote,
- NodeID 4: Seismic Mote #1,
- NodeID 5: Seismic Mote #2,
- NodeID 6: Repeater Mote.

A sample database table for flash-flood monitoring is presented in Tables 13 and 14.

Table 13: Sample Database Table Running on the Server and Client Computers

<table>
<thead>
<tr>
<th>NodeID #</th>
<th>Time Stamp</th>
<th>Batt Voltage</th>
<th>Temp</th>
<th>Pressure</th>
<th>Hum</th>
<th>Light</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3/23/05@12:30:00PM</td>
<td>3.43143</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>3/23/05@12:30:02PM</td>
<td>3.54254</td>
<td>22.95</td>
<td>1011.8</td>
<td>3.58694</td>
<td>356.53</td>
</tr>
<tr>
<td>2</td>
<td>3/23/05@12:30:04PM</td>
<td>3.13414</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>3/23/05@12:30:05PM</td>
<td>3.54542</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>3/23/05@12:30:08PM</td>
<td>3.04342</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>3/23/05@12:30:11PM</td>
<td>3.34322</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>3/23/05@12:30:16PM</td>
<td>3.54342</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Table 14: Continuation of the Sample Database Table

Running on Server and Client Computers

<table>
<thead>
<tr>
<th>Precip.</th>
<th>Wind Direct.</th>
<th>Wind Speed</th>
<th>Water Level</th>
<th>Water Flow</th>
<th>Seismic</th>
<th>Soil Moist</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0.05</td>
<td>SW 54.4</td>
<td>12.32</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>7.64</td>
<td>5.43</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>2.42</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>13.32</td>
<td>X</td>
</tr>
</tbody>
</table>

8.1.1 Alarm Generation

The life cycle of flash-flood monitoring is as follows:

- **No Signs of flash flood**: The WSN nodes indicate stable parameters throughout the sensed field. Characteristics of a stable condition include dry weather, normal water level and flow, high barometric pressure, with respect to other areas in the country, and static behavior of mountains,

- **Rain formation**: Pressure can be an important indicator in rain formation, since it tends to drop before rainfall. For instance, meteorological, hydrological and seismic nodes may be alerted by the pressure sensors of a possibility of rainfall in order to begin a network-wide flash-flood check. The check verifies that the hazardous conditions are met before triggering alarms,

- **Rain**: Precipitation is of vital importance for a flash-flood monitoring system. Landslides and river overflow are products of high levels of rainfall for long or even sometimes short but intense periods of rainfall,
- Landslides: They are monitored to prevent dam forming along rivers. Rainfall, seismic activity and unstable soil are a few examples of how landslides may be triggered. Such parameters need to be carefully monitored as well as the sudden disappearance of nodes,

- Dam forming: In most cases of flash-floods studied in the region, landslides were the major cause for dam forming. As the mountain collapses the earth rolls downhill and eventually reaches the rivers. Such a situation should trigger high-priority alarms to initiate actions by first responders. Under these circumstances the level of water raises rapidly in the upstream portions of the river and the flow is reduced drastically downstream,

- Flash-flood: This situation is very hard to monitor and it is not expected that most alerting messages will have taken place before it happens. Nodes being wiped out and extreme readings in some sensors are characteristic of this stage of the disaster.

In order to generate alarm messages the occurrence of a flash flood has to be identified and/or predicted. The life-cycle of a flash-flood can be associated with different alarm levels. Higher levels reflect a more critical situation implying the imminent occurrence of a flash-flood. An automated alarm system could use two types of mechanisms where the results could be compared in order to avoid false alarms. Such mechanisms could consist of

- A simple inference mechanism based on fuzzy logic or intelligent systems that are based on actual measurements or changes with respect to previous
measurements that tries to identify which phase of the flash flood exists. Such a mechanism was formerly termed an "expert" system,

- A more sophisticated mechanism that makes use of models to forecast the disaster or critical situations.

Table 15 describes the alarm levels used to trigger transmission of database files from the Stargate to the Local Office.

Table 15: Alarm Levels Used to Trigger Transmission of Database Files

<table>
<thead>
<tr>
<th>Alarming Level</th>
<th>Color</th>
<th>Description</th>
<th>Transmission Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Blue</td>
<td>Clear Conditions</td>
<td>30 minutes</td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
<td>Disaster Formation</td>
<td>15 minutes</td>
</tr>
<tr>
<td>3</td>
<td>Yellow</td>
<td>Potential Disaster</td>
<td>10 minutes</td>
</tr>
<tr>
<td>4</td>
<td>Orange</td>
<td>Imminent Disaster</td>
<td>5 minutes</td>
</tr>
<tr>
<td>5</td>
<td>Red</td>
<td>Disaster</td>
<td>2 minutes</td>
</tr>
</tbody>
</table>

8.1.2 Fuzzy Logic

The proposed solution employed a fuzzy inference system to generate the alarms. A fuzzy Inference system, (FIS), has following advantages:

- Possibility of using human knowledge,
- It is possible to constantly improve the system,
- It is simple to maintain, improve and update,
- Possesses the ability to handle uncertainty: Since the alerting levels are based on “conjectures”, the system has to handle subjective information,
• Possesses the capability of working with linguistic variables, which makes the user interface very simple.

The system could use a color code to indicate a measure of:

• The time left before the disaster happens,
• The probability that the disaster will strike.

The execution of such an inference engine does not mandate high computational resources or access to the historical register of data collected from the field. Therefore, it could be implemented in a computer with direct access to the sensor network or even in a distributed manner within the WSN.

The fuzzy logic algorithm needs to be developed with the assistance of Civil Defense authorities and experts on the environment at risk. This is because the algorithm takes into account all variables and, based on certain combinations, outputs alarm levels correspondent to the situation. As discussed in Chapter 4, it is of fundamental importance to develop a system that is sensitive and able to effectively recognize hazardous conditions. However, at the same time the system must be "intelligent" enough not to overreact and trigger false alarms. For example, if “Luminosity” is LOW, “Pressure” is LOW, “Precipitation” is HIGH, and “Water Level” is HIGH, then there is a high probability that a disaster is imminent. The algorithm would then output an alarm level that corresponds to this scenario.

As a proof of concept, a fuzzy logic algorithm with 3 alarm levels and 3 monitored magnitudes was implemented. Table 16 describes the 3 alarm levels that were used. The magnitudes monitored were water level, precipitation and soil moisture.
Table 16: Fuzzy Logic Algorithm as a Proof of Concept

<table>
<thead>
<tr>
<th>Alarming Level</th>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Green</td>
<td>Clear Conditions</td>
</tr>
<tr>
<td>2</td>
<td>Yellow</td>
<td>Potential Disaster</td>
</tr>
<tr>
<td>3</td>
<td>Red</td>
<td>Disaster</td>
</tr>
</tbody>
</table>

For this proof of concept fifteen (15) rules were generated. Figure 45 illustrates the mamdani inference system that was utilized.

![Fuzzy Inference System for the Alarm Algorithm](Image)

Figure 45: Fuzzy Inference System for the Alarm Algorithm

The rules for this FIS are what determined the sensitivity of the alarms. Tables 17 and 18 relate the 3 magnitudes and the possible combinations among them. For each combination an alarm is generated.

Table 17: Rules for the Fuzzy Logic Alarm Algorithm

<table>
<thead>
<tr>
<th>Water Level</th>
<th>low</th>
<th>low</th>
<th>low</th>
<th>low</th>
<th>med</th>
<th>med</th>
<th>med</th>
<th>med</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precipitation</td>
<td>low</td>
<td>med</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>med</td>
<td>med</td>
</tr>
<tr>
<td>Soil Moisture</td>
<td>low</td>
<td>low</td>
<td>med</td>
<td>high</td>
<td>low</td>
<td>med</td>
<td>low</td>
<td>med</td>
</tr>
<tr>
<td>ALARMING LEVEL</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 18: Continuation of the Rules for the Fuzzy Logic Alarm Algorithm

<table>
<thead>
<tr>
<th>Water Level</th>
<th>med</th>
<th>med</th>
<th>med</th>
<th>high</th>
<th>med</th>
<th>high</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precipitation</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>med</td>
<td>high</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>Soil Moisture</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>med</td>
<td>high</td>
<td>med</td>
<td>high</td>
</tr>
<tr>
<td>ALARMING LEVEL</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
8.2 Connecting to the GPRS Network

The Sierra Wireless Aircard 750 connected to the Stargate PCMCIA slot provided the gateway connectivity to a Global System for Mobile Communications, (GSM), cellular network. The General Packet Radio Service, (GPRS), network used for data communication was mounted on top of the GSM network. The Wireless Sensor Network used the GPRS network as the communication link between the remote monitoring subsystem and the data analysis and alerting subsystem. The GPRS network was also used as the system for broadcasting alerts to the population and local authorities. For example, if a safety threshold was violated, text messages would be sent to cellular phone numbers cataloged with the Civil Defense Authority. In the text messages the population would be instructed to evacuate or to apply preparedness procedures and local authorities would be instructed to take necessary measures. The Stargate computer and the Local Office workstations accessed the GPRS network by establishing a Point-to-Point Protocol, (PPP), connection. Such a network represents an ideal back-bone infrastructure for such applications since it can be used for data communication and data dissemination. In such a network database files are transmitted from the Stargate gateway, which is the server, to the Local Office, which is the client, via the GPRS network.

8.2.1 Establishing PPP Connection

The Aircard emulates a serial port over the PCMCIA connection, which enables a PPP connection to be initiated once the fuzzy logic model determines that it is time to transmit database files. The PPP is a mechanism for creating and running the Internet
and other network protocols over a serial connection, a telnet established link or a link established through the use of modems and telephone lines.

The Stargate gateway computer and the Local Office workstation were configured as both the PPP client and server because transmissions could be initiated by client requests or by database triggers. When the call was initiated by triggers in the database server the Stargate computer acted as the PPP client and the Local Office workstation acted as the PPP server. When authorities send queries to the network gateway the reciprocal was true.

In order to secure the transferring of the files, the PPP connection requires both peers to authenticate themselves using the Password Authentication Protocol, (PAP). Once the connection is established, each end requests the other to authenticate itself by sending a user name and a password. Similar scripts as the ones presented in Appendix A.1 were used to automatically initiate PPP connections from both the client and server ends.

8.3 Establishing WLAN Connectivity

The Stargate board, which was used as the network gateway, had a Compact-Flash slot in the motherboard that allowed the insertion of an 802.11 Wireless Compact Flash Card. Due to the short range of the technology, which is usually around 100 meters, WLANs cannot be used as the communication link between the network gateway and the Local Office. However, this permits Civil Defense authorities to establish 802.11 connectivity with the Stargate and download database files from the gateway to a laptop computer or a PDA that has a relational database installed.
8.4 Internet Connectivity

The internet can perform a two-fold functionality for the monitoring and alerting system. One of the features of the network gateway includes running an Apache HTTP server. The data collected from the sensed field and stored in the local database is transmitted to the Local Office primarily through the GPRS network. Another possibility is to publish the database files in the HTTP server and let the workstation at the Local Office download the file remotely. This redundancy is important in order to guarantee that the database files will reach the central workstations for data processing. The other functionality of the internet is in data dissemination. The data can be made available to the general public after it has been processed by the Local Office. This provides the population with almost real-time information on the monitoring of a disaster.
CHAPTER 9

CONCLUSIONS AND FUTURE WORK

Wireless Sensor Networks have proven themselves to be a reliable solution in providing remote sensing for natural disaster monitoring systems. The motes were adapted according to application-specific requirements to sense, collect and transmit relevant parameters. The integration of custom transducers to motes required the implementation of signal conditioning algorithms to accurately extract the data from the devices and apply it to the data acquisition board channels. The DAQ boards allowed different mote configurations such as meteorological, hydrological and seismic to be designed. The information collected from the network was stored in a local database and made available for transmission via the GPRS network to Local Office workstations for analysis. Since the cell architecture was comprised of different types of sensor nodes, unique data packets were generated according to the type of transducers that were coupled to the mote. The TinyOS message structure allowed adaptations where the payload length could be customized as desired.

Passive transducers were selected to minimize current draw. Experiments with three 1.2V batteries increased battery lifetime and consequently, network lifetime. At 1% duty cycle the three battery setup enabled the motes to operate for approximately 16
months. More experiments involving battery capacity and mote lifetime are recommended. The integration of a solar panel to power the motes is also an alternative to be investigated. It is vital to continue using energy-aware approaches since energy consumption continues to be the limiting factor of Wireless Sensor Network technology.

In the field tests performed at USF and in Maracay, the network was established in approximately 20 minutes. The lack of infrastructure requirements enabled easy and fast deployment at both sites. The flexibility to adapt to any environment makes WSN desirable for the monitoring of several natural disasters. The network proved to be robust under the conditions tested. For regions such as USF, with flat characteristics, communication quality and network statistics approximated 100% fidelity. However, in the mountainous environments of Maracay the results, on average, remained above 85%. Particularly in Maracay, several challenges involving network communications connectivity to higher-networks and accessibility were analyzed. The WSN used for remote sensing successfully established communication. However, future work should include investigations with respect to transmitting the collected data from the network gateway to the Civil Defense office.

A second experiment involved increasing network density. Improvements in the fidelity of results were detected as additional datapaths were created. Furthermore, transmission distances were reduced, which diminished transmission costs and prolonged battery lifetime in each mote.

Presently, Wireless Sensor Networks have generic features for a variety of applications. However, the requirements for disaster monitoring are unique and differ from the ones for industrial monitoring or environmental monitoring. In fact, monitoring
the same phenomena in different locations may require different cell architectures since each environment is unique. In the near future it will be critical to adapt technology to comply with all requirements imposed for natural disaster monitoring.

The use of WSNs to remotely sense the environment is a more reasonable solution in places that lack money and human resources. It is affordable to operate and maintain, adaptable to any environment, scalable and provides the population with accurate information regarding the threatened region, which saves lives and minimizes property damage. The adaptation and flexibility for each environment is what makes Wireless Sensor Networks attractive for use in remote sensing. The power of using a WSN in conjunction with a signal processing station lies in its ability to transform raw data, such as the data coming from the WSN, into useful information. The workstation, in the hands of decision makers and authorities, becomes a powerful instrument that helps in the rapid assessment of critical situations.

The monitoring requirements for these concept solutions are determined by local authorities. More specifically, the types of transducers to be used and the strategic locations where motes will be deployed to obtain optimum readings require understanding of the environment to be sensed. Increasing the understanding of the environment with the collected data from the solution can be used to enhance the system by improving the fuzzy logic algorithm that is responsible for the transmission interval of packets from the Stargate gateway to the Local Office. If data is not available prior to deployment, it is difficult to establish thresholds for the alerting of hazardous conditions.
REFERENCES


APPENDICES
Appendix A PPP Scripts

A.1 ppp-on script

/**Script to initiate a PPP connection. These are the parameters.
**/Change as needed.

TELEPHONE=974-4551 ACCOUNT=network PASSWORD=monitor
LOCAL_IP=0.0.0.0
REMOTE_IP=0.0.0.0 NETMASK=255.255.255.0

export TELEPHONE ACCOUNT PASSWORD

DIALER_SCRIPT=/etc/ppp/ppp-on-dialer

*Initiate the connection

exec /usr/sbin/pppd debug /dev/ttyS1 115200 \n   $LOCAL_IP:$REMOTE_IP \n   connect $DIALER_SCRIPT
Appendix A (continued)

A.2 ppp-on-dialer script

* It will perform the connection protocol for the desired connection.

```
/usr/sbin/chat -v
    TIMEOUT 3
    ABORT '\nBUSY\r'
    ABORT '\nNO ANSWER\r'
    ABORT '\nRINGING\r\nRINGING\r' ''
    \rAT
    'OK-+++\c-OK' ATH0
    TIMEOUT 30
    OK ATDT$TELEPHONE
    CONNECT ''
    login:--login: $ACCOUNT
    password: $PASSWORD
```
Appendix A (continued)

A.3 ppp-off script

* Determine the device to be terminated.
* if [ "$1" = "" ]; then
  DEVICE=ppp0
else
  DEVICE=$1
fi
* If the ppp0 pid file is present then the program is running. Stop it.
if [ -r /var/run/$DEVICE.pid ]; then
  kill -INT `cat /var/run/$DEVICE.pid`
* If the kill did not work then there is no process running for this.
* pid. It may also mean that the lock file will be left. You may wish
* to delete the lock file at the same time.
  if [ ! "$?" = "0" ]; then
    rm -f /var/run/$DEVICE.pid
    echo "ERROR: Removed stale pid file"
    exit 1
  fi
* Success. Pppd cleanup.
  echo "PPP link to $DEVICE terminated."
  exit 0
fi
* The ppp process is not running for ppp0
echo "ERROR: PPP link is not active on $DEVICE" exit 1
Appendix B Intersema MS5534 Algorithm

Figure 46: Intersema MS5534 Algorithm
Appendix B (continued)

Figure 46: Continued
Appendix C Taos TSL2550 Algorithm

<table>
<thead>
<tr>
<th>VALID</th>
<th>CHORD BITS</th>
<th>STEP BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B7</td>
<td>B6 B5 B4</td>
<td>B3 B2 B1 B0</td>
</tr>
<tr>
<td>VALID</td>
<td>C2 C1 C0</td>
<td>S3 S2 S1 S0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FIELD</th>
<th>BITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALID</td>
<td>7</td>
<td>ADC channel data is valid. One indicates that the ADC has written data into the channel data register, since ADCEN was asserted in the COMMAND register.</td>
</tr>
<tr>
<td>CHORD</td>
<td>6 to 4</td>
<td>CHORD number.</td>
</tr>
<tr>
<td>STEP</td>
<td>3 to 0</td>
<td>STEP number.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHORD BITS B5, B4</th>
<th>C. CHORD NUMBER</th>
<th>CHORD VALUE (Note A)</th>
<th>STEP VALUE (Note B)</th>
<th>STEP BITS B3, B2, B1, B0</th>
<th>S. STEP NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>16</td>
<td>2</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>2</td>
<td>49</td>
<td>4</td>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>011</td>
<td>3</td>
<td>115</td>
<td>8</td>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
<td>247</td>
<td>16</td>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
<td>511</td>
<td>22</td>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
<td>1039</td>
<td>64</td>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>7</td>
<td>2095</td>
<td>128</td>
<td>1111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

A. CHORD VALUE = INT (16.5 x (2^C - 1))

B. STEP VALUE = 2^S

The ADC count value is obtained by adding the CHORD VALUE and the product of the STEP NUMBER and STEP VALUE (which depends on CHORD NUMBER).

ADC Count Value = (CHORD Value) + (Step Value) x (Step Number)

The ADC count value can also be expressed as a formula:

ADC Count Value = (INT (16.5 x (2^C - 1)) + (S x (2^S))

where:

C is the CHORD NUMBER (0 to 7)
S is the STEP NUMBER (0 to 15)
as defined in Table 3.

Figure 47: Taos TSL2550 Algorithm
Appendix D MDA300CA DAQ Board Schematic

Figure 48: MDA300CA DAQ Board Schematic
Appendix E Sensor Driver

E.1 SoilDriver.nc

includes sensorboard;

configuration SoilDriver {
    provides interface ADC as SensorData;
    provides interface StdControl;
}

implementation {
    components SoilDriverM, ADCC;

    StdControl = SoilDriverM;
    SensorData = ADCC.ADC[TOS_ADC_SOIL_PORT];
    SoilDriverM.ADCControl -> ADCC;
}

}
E.2 SoilDriverM.nc

module SoilDriverM {
    provides interface StdControl;
    uses {
        interface ADCControl;
    }
}

implementation {

    command result_t StdControl.init() {
        TOSH_MAKE_SOIL_CTL_OUTPUT();
        TOSH_SET_SOIL_CTL_PIN();
        return call ADCControl.init();
    }

    command result_t StdControl.start() {
        TOSH_MAKE_SOIL_CTL_OUTPUT();
        TOSH_SET_SOIL_CTL_PIN();
        return SUCCESS;
    }

    command result_t StdControl.stop() {
        TOSH_CLR_SOIL_CTL_PIN();
        return SUCCESS;
    }
}
Appendix E (continued)

E.3 TestDriver.nc

configuration TestSensor {
// this module does not provide any interface
} implementation {
    components Main, TestSensorM, LedsC, TimerC, SoilDriver as Sensor;

    Main.StdControl -> TestSensorM;
    Main.StdControl -> TimerC;

    // Wiring for New Sensor
    TestSensorM.SensorControl->Sensor;
    TestSensorM.SensorData -> Sensor;

    TestSensorM.Leds -> LedsC;
    TestSensorM.Timer -> TimerC.Timer[unique("Timer")];
}

}
module TestSensorM {
  provides {
    interface StdControl;
  }
  uses {
    interface StdControl as SensorControl;
    interface ADC as SensorData;
    interface Timer;
    interface Leds;
  }
}

implementation {
  /******************************************************************************/
  /* Initialize the component. */
  /******************************************************************************/
  command result_t StdControl.init() {
    call Leds.init();
    call SensorControl.init();
    return SUCCESS;
  }

  /******************************************************************************/
  /* Start the component. Start the clock. */
  /******************************************************************************/
  command result_t StdControl.start() {
    call Leds.redOn();
    call SensorControl.start();
    call Timer.start(TIMER_REPEAT, 1000);
    return SUCCESS;
  }

  /******************************************************************************/
  /* Stop the component. */
  /******************************************************************************/
  command result_t StdControl.stop() {
    call SensorControl.stop();
  }
}
return SUCCESS;
}

/******************************************************************************
* Measure Sensor
******************************************************************************
event result_t Timer.fired() {
    call SensorData.getData();
    call Leds.redToggle();
    return SUCCESS;
}

/******************************************************************************
* Sensor ADC data ready
* Issue a command to sample the Soil ADC data.
******************************************************************************
async event result_t SensorData.dataReady(uint16_t data) {
    call Leds.greenToggle();
    return SUCCESS;
}