DESIGN AND APPLICATIONS OF ZIGBEE-ENABLED WIRELESS SENSOR SYSTEMS FOR A SPACE RELEVANT ENVIRONMENT

A Thesis in

Electrical Engineering

by

Erik D. Weir

© 2010 Erik D. Weir

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

December 2010
The thesis of Erik D. Weir was reviewed and approved* by the following:

Sven G. Bilén
Associate Professor of Engineering Design, Electrical Engineering, and Aerospace Engineering
Thesis Advisor

Julio V. Urbina
Assistant Professor of Electrical Engineering

W. Kenneth Jenkins
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School
ABSTRACT

A prototype wireless sensor system is developed to collect data from sensors that are of importance on space platforms, and also important in the testing of these systems in space-relevant environments. The elimination of wires and the associated failures such as chafing, sparking, ageing, and connector issues can increase reliability and design flexibility while reducing costs. These factors present an appealing case for the pursuit of wireless solutions for harsh environments and particularly for their use in space and on spacecraft.

Recent testing of this prototype system in the NASA Ames Research Center’s Aerodynamic Heating Facility during Test 289 in January 2010, demonstrated the feasibility of using wireless sensors in ground test programs and point to their use on flight vehicles. The test demonstrated data collection of a type-K thermocouple at ±2.5 °C resolution, with real-time data rates of 1 Hz, and high-speed data collection rates of 60 Hz and higher. The presence of the plasma, heat, and vacuum within the chamber did not inhibit data acquisition, although there was a period of radio blackout while there was plasma between the system and the receiver. A further test was attempted in the Panel Test Facility to show 100% communication uptime and multiple sensor support, and although that test was halted by critical battery failure, an upcoming “wedge” test in the Interaction Heating Facility, will showcase a fully functional system.

With continued refinement, the next version of the wireless sensor system has the potential to exhibit an increase in data frequency, a dramatic increase in battery life, and a marked reduction in size. This thesis demonstrates that wireless sensor systems have the capability to replace wired systems in space applications. Future testing will further increase the technology readiness level of the wireless sensor system, verifying that wireless systems have a place on future flight missions.
This thesis discusses the overall project, recent testing results, as well as future directions for research.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................... vii

LIST OF TABLES .............................................................................................................. x

ACKNOWLEDGMENTS ...................................................................................................... xi

Chapter 1 Introduction ....................................................................................................... 1

1.1. Motivation .................................................................................................................. 2
1.2. Contribution ............................................................................................................... 3
1.3. Overview of the Thesis ............................................................................................. 4

Chapter 2 Background ....................................................................................................... 6

2.1. Wireless Sensor Networks ....................................................................................... 6
2.2. Wireless Protocols and Topologies ............................................................................ 7
2.3. Thesis Motivation ....................................................................................................... 9
2.4. Current Wireless Sensor Technology ........................................................................ 11
2.5. Test Background ....................................................................................................... 12
2.6. Application Relevant Wireless Sensor Networks ...................................................... 13
2.7. Chapter Summary ..................................................................................................... 15

Chapter 3 System Description ........................................................................................ 16

3.1. System Specifications .............................................................................................. 16
  3.1.1. Functional Requirements ...................................................................................... 17
  3.1.2. Physical Constraints (AHF) ................................................................................ 19
3.2. WLSS Description .................................................................................................... 22
  3.2.1. Design Modularity ............................................................................................... 22
  3.2.2. Physical Breakdown ............................................................................................ 24
  3.2.3. Functional Breakdown ......................................................................................... 25
  3.2.4. Power Subsystem ............................................................................................... 26
  3.2.5. Signal-Conditioning Subsystem .......................................................................... 29
  3.2.6. Control Subsystem ............................................................................................ 32
3.3. Microcontroller Software ........................................................................................ 35
3.4. PC Control and Monitor Software ........................................................................... 38
3.5. Chapter Summary ..................................................................................................... 41

Chapter 4 Proof of Concept Tests and Analysis ............................................................... 42
LIST OF FIGURES

Figure 1-1: Wiring inside the wing of shuttle Columbia (left) and inside the AHF wiring patch panel................................................................. 2

Figure 2-1: Network topologies [from Lewis et al., 2004]. ............................................. 9

Figure 2-2: RF noise measurement of the Ames PTF in operation where the white marker denotes the bandwidth chosen and the observed spike at 1.8 GHz is attributed to smart phone communication. ...................................................... 13

Figure 3-1: Cutout view of arc jet model. ........................................................................ 20

Figure 3-2: Calorimeter example showing TPS materials. ............................................. 21

Figure 3-3: Example of complete test model showing the calorimeter in tan and the tile holder in white. ............................................................... 21

Figure 3-4: WLSS functional partitioning......................................................................... 22

Figure 3-5: WLSS functional diagram............................................................................ 23

Figure 3-6: WLSS data flow diagram. ............................................................................... 24

Figure 3-7: Three board system. In the right image: the leftmost board (bottom) is the signal conditioning board, the middle board holds the power subsystem, and the rightmost board (top) is the controller board...................................................... 25

Figure 3-8: Functional partitioning of the WLSS.................................................................................. 26

Figure 3-9: Single-cell 3.7-V LiPo battery [from Moore, 2008].................................. 26

Figure 3-10: Lithium polymer battery discharge curve [from Moore, 2008]. ............... 27

Figure 3-11: Power subsystem simplified schematic.......................................................... 28

Figure 3-12: Power subsystem circuit top/bottom view...................................................... 29

Figure 3-13: Signal-conditioning subsystem simplified schematic................................. 30

Figure 3-14: Signal-conditioning subsystem circuit top/bottom view. ........................... 32

Figure 3-15: Control subsystem simplified schematic...................................................... 33

Figure 3-16: Control subsystem circuit top/bottom view.................................................. 35
Figure 3-17: Microcontroller firmware flow diagram........................................................... 36
Figure 3-18: Microcontroller code for ADC data gathering........................................ 38
Figure 3-19: LabVIEW front panel.................................................................................. 39
Figure 3-20: Serial controller loop................................................................................ 40
Figure 3-21: Event handler loop..................................................................................... 40
Figure 3-22: Data parser loop......................................................................................... 41
Figure 4-1: WLSS tile-holder and TPS........................................................................... 44
Figure 4-2: Arc Jet diagram. [from Eddlemon, 2005]..................................................... 45
Figure 4-3: In-chamber view of the AHF arc jet............................................................. 46
Figure 4-4: Y-configuration of dual sensor systems....................................................... 47
Figure 4-5: WLSS sample in the arc jet.......................................................................... 48
Figure 4-6: WLSS sample after exposure in the arc jet.................................................. 48
Figure 4-7: Real-time data from run 001........................................................................ 49
Figure 4-8: CUIP Test 2 WLSS/WSS comparison.......................................................... 50
Figure 4-9: WLSS wedge adaptor.................................................................................. 54
Figure 5-1: Wired and wireless data at 60 Hz................................................................. 57
Figure 5-2: Wired and wireless real-time slopes............................................................... 58
Figure 5-3: Data offset.................................................................................................... 59
Figure 5-4: Rising temperature in CJC test................................................................. 59
Figure 5-5: Peak temperature in CJC test....................................................................... 60
Figure 5-6: Falling temperature in CJC test................................................................. 60
Figure 9-1: Power subsystem schematic....................................................................... 82
Figure 9-2: Power subsystem layout............................................................................. 83
Figure 9-3: Signal-conditioning subsystem schematic................................................. 84
Figure 9-4: Signal-conditioning sub circuit PCB......................................................... 85
Figure 9-5:  Control subsystem schematic (MCU).......................................................... 86
Figure 9-6:  Control subsystem schematic (I/O).......................................................... 87
Figure 9-7:  Control subsystem PCB................................................................. 88
LIST OF TABLES

Table 2-1: Decision matrix for different wireless technology [Swanson et al., 2007] ........ 8
Table 3-1: WLSS System Hardware/Firmware Requirements........................................... 18
Table 3-2: WLSS System Software Requirements......................................................... 19
Table 4-1: Data acquisition systems capabilities.......................................................... 43
Table 6-1: Future IRVE plan. ......................................................................................... 64
ACKNOWLEDGMENTS

This thesis could not have been completed without the help and support of a large and varied cast of individuals. My thanks goes out to everyone who has had a hand in supporting, encouraging, and occasionally slapping me back into action at various points during my Master’s degree and the completion of this thesis.

I would like to first thank my advisor, Dr. Sven Bilén. I think that no other advisor would tolerate, or empathize with, the frenetic pace that characterized certain pieces of this process. I am forever grateful for the knowledge that he has handed down about academia, entrepreneurship, and engineering that has shaped my life.

I thank my other thesis committee member, Dr. Julio Urbina, for the interest and excitement he showed in my project.

I thank Robert M. Capuro, Trey Morris, Corey Friedenberger, Aseem Signh, Allen Kummer, Kyle Holmes, and everyone else on the Penn State team who has worked toward helping this project succeed. Without the team, we would have had no chance of success.

I thank David Hash and Johnny Fu from NASA Ames for their continued support of this project and the process. I also wish to thank NASA’s Constellation Universities Institutes Project (CUIP) for the funding to develop this system.

I thank Penn State and its professors and staff for six and a half years of invaluable education and life-changing experiences and opportunities.

I thank my business partners at Buzby Networks for their patience and support while I sometimes struggled to manage my time.

I thank all of my friends and family. Without friendship and support, this would not have been close to possible.
I thank my family for their continued support and love that has helped me always to know that whatever happens, I will still be loved.

Finally, I thank my wonderful friend and companion, Becky. Your love, support, guidance, shining example, and occasional chastising motivated me to complete this project and still have a smile on my face.

Erik David Weir

The Pennsylvania State University

December 2010
Chapter 1

Introduction

Passengers on TWA Flight 800 departed John F. Kennedy Int’l Airport on July 17th, 1996 at 8:31 pm for Leonardo da Vinci Int’l Airport. Although the passengers surely knew that the Boeing 747 was not new, none of them could have been thinking that the plane’s aged wiring would play a part in what was to come. Shortly after takeoff, the plane’s fuel tank ignited, leading to an explosion that wrenched the plane from the sky and killed all passengers aboard [National, 2000]. The most widely recognized culprit for this terrible tragedy is a wire short that ignited fuel vapors within one of the fuel tanks. Further studies have shown that planes in service for more than a decade typically show signs of wire ageing that could lead to similar incidents [Furse and Haupt, 2001]. With the rapid improvement of wireless sensor technology, replacement of these dangerous ageing wire bundles with a new and safer wireless replacement seems inevitable. Sensor systems, in particular, can be cheaper in terms of cost and risk, allowing more data to be gathered and increasing the effectiveness and efficiency of any system. For these reasons, NASA engineers have identified the need for wireless sensors to enable more efficient data gathering on future space missions, as well as in testing applications [Constellation, 2008].

This document examines the motivations behind the design of a wireless sensor system, as well as the results of the testing of that system. Finally, conclusions are made about the system and its applications.
1.1. **Motivation**

Wireless solutions enable a reduction in the quantity and complexity of physical interconnects within a system. This offers a wide variety of advantages in several areas [TPS Project Specifications (Penn State), 2009]:

- **Reduced life cycle cost**: Wireless systems reduce life cycle costs through ease of upgrades, re-configuration, troubleshooting, and root-cause determination.
- **Improved reliability**: Wireless systems reduce typical wired system failure modes caused by failures of wires, insulation, and connectors.
- **Increased accessibility**: Structural barriers limit physical access to wired subsystems as shown in **Figure 1-1**.
- **Mass reduction**: Wireless systems offer the potential for significant mass savings by eliminating the need for cables, brackets, connectors, bulkheads, cable trays, structural attachments, and reinforcements.
- **Adaptability**: Wireless systems offer significant advantages over wired systems when upgrades or re-configurations are required.

![Figure 1-1](image.png)

**Figure 1-1**: Wiring inside the wing of shuttle *Columbia* (left) and inside the AHF wiring patch panel
Despite wireless’ list of compelling advantages, it is important to note that there are also disadvantages that need to be considered when replacing a wired sensor system with a WLSS. Some disadvantages are as follows:

- **Signal loss:** A break in communication during a vital period of data collection can have consequences if the information needs to be acted on in real-time.

- **Power lines:** Current systems require uninterrupted power during data gathering. The potential for power failure due to an old or bad battery causes most wireless solutions to be hard-powered, thereby tethering an otherwise wireless system.

- **Wireless speed:** Wireless systems are generally slower than wired systems due to the physical constraints of broadcasting over the air. Additionally, lower power applications typically sacrifice data rate to achieve their low power requirements.

- **Security:** No matter what security protocols are used, unknown parties can more easily intercept wireless data that is being broadcasted. Although encryption of wireless data is reaching levels equal to wired security protocols, tapping into a wired solution generally still requires physical contact whereas the wireless solution only requires an antenna.

The motivation behind this effort was to realize the advantages of a wireless system within the constraints of a system for use in a space-relevant environment, while at the same time, minimizing the disadvantages that also come with this paradigm shift.

### 1.2. Contribution

NASA engineers have placed a high priority on the development of wireless sensor networks in order to instrument Thermal Protection Systems (TPSs) as well as for use in other
subsystems and mission segments. An additional need for easily configurable wireless sensor systems has been identified within testing facilities that require rapid test-bed reconfiguration. Developing WLSS for testing facility applications has a lower developmental risk than deploying a WLSS on a space mission, yet can advance the Technology Readiness Level (TRL) of the technology. Previous testing by the University of Idaho ThermaSense team demonstrated wireless connectivity in a vacuum plasma test environment [Swanson et al., 2007]. The WLSS described herein was subjected to rigorous testing to verify that the system can support high-speed data collection, support multiple sensor types, and operate in the NASA Ames Aerodynamic Heating Facility (AHF) chamber in the presence of hypersonic plasma flow.

This thesis is focused on the system engineering aspects of the WLSS package. The author was responsible for development of the power system, the microcontroller firmware, and the data collection software, as well as the integration and testing of the system. The software for the firmware and the data collection were developed in CCS C-code and NI LabVIEW, respectively, and circuit design was completed using the Orcad suite. Integration and testing of the system were also accomplished to verify system compliance with the initial test goals.

1.3. Overview of the Thesis

This thesis is organized into six chapters. The first is an introduction to the body of work detailed herein.

Chapter 2 provides background material in order to understand and contextualize the project. It will discuss the motivations for the replacement of wired sensors with wireless counterparts, the previous work that has been drawn upon, the network protocols and topologies that exist to enable wireless sensor networks, the conceptualized solution, the benefits of that solution, and, finally, the potential issues that exist with the concept.
Chapter 3 provides a description of the system. The chapter describes the hardware, software, and MCU firmware of the system in great detail as well as the interaction between these three components. The functional breakdown and physical breakdown of the hardware explain the three-piece architecture as well as design decisions that were made in the production of the initial prototype. The software and firmware are described for the purpose of understanding the system interactions.

Chapter 4 describes the testing of the system at the NASA Ames Research Center. Goals for the test are covered, and the data from this test is used to verify the completion or partial completion of these goals.

Chapter 5 contains the analysis of the tests including explanations of the issues that were faced during the testing, theories about the causes for these issues, and the verification of these hypotheses through further testing. Conclusions will also be drawn about the data that were collected from the test.

Chapter 6 summarizes major conclusions derived from the testing. It looks back on the tests and enumerates the lessons learned as well as looks forward to future work in other aerospace applications, detailing a possible future WLSS design.

Finally, appendices are included at the end of the material. Appendix A contains selected microcontroller firmware. Appendix B contains selected LabVIEW code samples. Appendix C contains hardware circuit diagrams and PCB layouts.
Chapter 2

Background

This chapter provides the background to the work performed in this thesis project. We discuss the motivations for wired systems migrating to wireless systems, give a brief overview of network protocols and topologies, describe the previous work in the wireless sensor field, and identify and briefly evaluate the system solution.

2.1. Wireless Sensor Networks

Before we delve into the reasons why wireless sensor networks (WSNs) should be considered for replacement of wired sensor acquisition systems, it is important to understand what a wireless sensor network is. The short definition of a WSN as “a network of sensors that uses a wireless protocol to communicate” is not adequate to describe the full scope of the technology. While wireless electronic sensors have existed for decades, recent developments have brought what was once a very niche technology into the forefront of development on a number of platforms. The recent emergence of new wireless technologies and protocols has enabled the upsurge of research and development in the WSN field as well as an increase in the deployments of WSNs for myriad applications. Wireless personal area networks (WPANs) were originally intended to wirelessly interconnect devices around a person’s workspace. This application does not immediately apply itself to sensor networks that can be used to replace
additional wired systems; however, the technology that is needed to enable devices around a person’s body that can be mobile with the person is, by definition, a low-power and very robust network [Lewis, 2004], which are the same requirements for the use of WSNs in many applications.

2.2. Wireless Protocols and Topologies

To be considered for use in a WSN, a wireless protocol must be low power, have a small footprint (MCU code and PCB layout), and be robust. Depending on the application, varying requirements on power, size, and robustness are levied. The first major design question is whether active or passive sensors are required. An active sensor has an on-board battery or other power supply that is used to facilitate communications and data collection. Active devices facilitate multiple routing protocols, while passive sensors currently permit only point-to-point communication. Passive sensors, additionally, require close proximity to a reader (usually between 10 cm and 2 m), to allow actuation of the sensor and transmission of an identifier and any stored data. WSNs cannot be created because infrastructure cannot be built using devices that do not intercommunicate. An active network, however, can communicate data at any period in time, not just when activated by the reader. For the given application, active networks make more sense because of the need for real-time monitoring of sensor data.

A few wireless protocols were examined to determine which best fit the application of a WSN. This comparison is shown below in Table 2-1. Traffic streams for sensor data in a WSN will generally be low, somewhere on the order of 1–100 kb/s [Sohrabi et al., 2000]. Also noted was that the lifetime of the network must be prolonged, requiring energy conservation in the network protocol and topology [Sohrabi et al., 2000]. Therefore, goals in designing a WSN were
low power consumption, medium data rate, good range, and a variety of possible network
topologies. For these reasons, ZigBee was identified as a strong protocol for WSN applications.

Table 2-1: Decision matrix for different wireless technology [Swanson et al., 2007]

<table>
<thead>
<tr>
<th></th>
<th>ZigBee</th>
<th>802.11</th>
<th>Bluetooth</th>
<th>Wireless USB</th>
<th>IR Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Rate</td>
<td>250 kbits/s</td>
<td>54 Mbits/sec</td>
<td>1 Mbits/s</td>
<td>62.5 kbits/s</td>
<td>20–40 kbits/s</td>
</tr>
<tr>
<td></td>
<td>10–100 meters</td>
<td>50–100 meters</td>
<td>10 meters</td>
<td>10 meters</td>
<td>&lt;10 meters (line of sight)</td>
</tr>
<tr>
<td>Range</td>
<td>ad-hoc, peer-to-peer, star, or mesh</td>
<td>point-to-hub</td>
<td>ad-hoc, small networks</td>
<td>point-to-point</td>
<td>point-to-point</td>
</tr>
<tr>
<td>Network Topology</td>
<td>2.4 GHz</td>
<td>2.4 and 5 GHz</td>
<td>2.4 GHz</td>
<td>2.4 GHz</td>
<td>800–900 nm</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Complexity</td>
<td>very low</td>
<td>high</td>
<td>medium</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>very low</td>
<td>high</td>
<td>medium</td>
<td>low</td>
<td>low</td>
</tr>
</tbody>
</table>

Various topologies can be utilized in a ZigBee network. These topologies determine how
sensors and the coordinator communicate with each other. Typical topologies, such as star, ring,
bus, tree, fully connected (web), and mesh are shown in Figure 2-1 below. As seen in the figure,
these topologies dictate which wireless elements can communicate and, furthermore, indicate the
path of data flow. For a WSN, the star, mesh, or fully connected topology facilitate a plurality of
data paths [Lewis et al., 2004]. The most useful network topology, however, varies depending on
the needs of the application. Star networks, for instance, are very good for a small number of
wireless sensor nodes that are physically close to the network coordinator. The mesh topology,
on the other hand, is useful for huge deployments with many hundreds of sensors. This capability
has caused many companies to invest in wireless personal area networks (WPAN) for the purpose
of addressing colossal markets such as healthcare and industrial applications [Global Inventures,
2010]. It was determined that the WSNs should aim to support mesh networking in order to
facilitate better data pathing through a sensor array.
2.3. Thesis Motivation

Up to this point, we have seen that wireless sensor networks have a variety of capabilities that enable collection of sensor data. The issue then becomes identifying the needs where a short-term increase in cost and complexity is offset by the long-term benefits of a wireless system. There exists a significant need for robust WLSSs in harsh environments, particularly those that exist on aerospace platforms [Furse and Haupt, 2001; Furse et al., 2003; Bruning and Campbell, 1993]. Another area where this need is evident in aerospace systems is in the replacement of wires. Prevention of wire failure modes such as ageing, cracking, and fraying is a huge issue as, in one estimate, the U.S. Navy spends 1.8 million person-hours annually to troubleshoot and repair aircraft wiring [Furse and Haupt, 2001]. Due to the large cost, and in light of the downing of Swissair 111 in 1998 and of TWA 800 in 1996 largely attributed to faulty wiring, NASA, the U.S. military, and the White House have made the problem of wire ageing a national priority.
[Furse and Haupt, 2001]. This has predictably caused a significant research push in the field of detecting faulty wiring. It has also caused interest in research toward removing as much wiring as possible from these systems [Furse et al., 2003; Bruning and Campbell, 1993]. The Design for Safety Initiative WIRe study, completed by an appointed panel of NASA and industry personnel in 2000, suggested that new concepts for replacement of these wired systems with next generation wireless systems be researched [“Wiring Integrity Research”, 2000]. Additionally, the removal of wires in sensor and control systems will also significantly reduce the problems associated with detection of faults and failures.

This focus on wire removal causes us to look at sensor systems in detail. As previously stated, the purpose of any system of sensors is to gather data remotely from specific points of interest dispersed throughout a larger system. Often, however, it is difficult to make these measurements at the location(s) of interest if it is necessary to run physical wire(s) to those points of interest. For example, NASA engineers have identified the need for wireless sensors to ameliorate the problems associated with physically interconnected systems, specifically within spacecraft Thermal Protection Systems (TPS) utilized during re-entry [Swanson et al., 2007]. Wires that connect the sensors on the TPS to the spacecraft must be severed during separation, an operation that creates a possible failure mode. Hence, data collection on TPS can be significantly more difficult than elsewhere on a spacecraft and pits the desire for more data against the risk associated with wire severance malfunction.

NASA engineers have placed a high priority on the development of wireless sensor networks in order to instrument TPSs as well as for use in decreasing wire failure in aerospace systems [Constellation, 2008]. An additional need for easily (re-)configurable wireless sensor systems has been identified within testing facilities that require rapid test-bed reconfiguration such as the wedge test in the Panel Test Facility (PTF) at the NASA Ames Research Center.
Developing WLSS for testing facility applications has a lower developmental risk than deploying a WLSS on a space mission, yet can advance the technology’s TRL.

2.4. **Current Wireless Sensor Technology**

The current wireless sensor design space is extremely broad. The increasing capability of WSNs has enabled a huge number of applications for which wireless sensing may now be possible. WSNs can be defined as a “large-scale, *ad hoc*, multi-hop, un-partitioned network of largely homogeneous, tiny, resource-constrained, mostly immobile sensor nodes” [Römer and Mattern, 2004]. The design space for WSNs is comprised of the deployment type, mobility, cost, size, resources and energy use, heterogeneity, communication modality, network topology and coverage [Römer and Mattern, 2004]. For the particular WSN application of instrumenting the spacecraft TPS, the deployment will be a one-time deliberately chosen activity, the nodes may be mobile but will retain a fixed position with respect to each other, the cost must be low, size must be small, and the system must be extremely low power. The sensor nodes may differ in the type and number of attached sensors, but RF communication is the most viable communication solution. The network topology of these systems, as well as the coverage, is not as standardized or important.

Wireless sensor systems currently are used in a wide variety of applications, which can be divided into a few key categories. The first category is battery-powered wireless RF systems that are created for active sensor monitoring and can be exemplified by the systems detailed in Shan *et al.* [2005] and Mainwaring *et al.* [2002]. The papers show typical design processes for the creation of a wireless system with a direct design goal. The next systems detailed are those where wireless data transfer is achieved while devices are either powered remotely using energy harvesting techniques as in George *et al.* [2004] and Arms *et al.* [2009], or directly using
wirelessly beamed power as in Kocer et al. [2004]. As George et al. [2004] point out, however, these systems cannot be used if sensor power requirements or sensor-reporting rates are high.

2.5. Test Background

The NASA Ames Research Center plasma arc jets are ideal for testing systems that must operate in space environments. The chambers were built to accurately subject samples to vacuum and plasma environments that are experienced during atmospheric re-entry. A typical test run in a plasma chamber, which include the Aerodynamic Heating Facility (AHF), Interaction Heating Facility (IHF), and the Panel Testing Facility (PTF), among others, consists of exposing a sample to a pre-determined plasma flow for a pre-determined period of time. Sensors within the sample relay information from the test in real time to the operators of the test through a reconfigurable wire bundle. Due to the close proximity of atmospheric entry conditions as well as the opportunity to replace inefficient wire reconfigurations with the ease of wireless modularity, the plasma chambers are an ideal test bench for the WLSS.

One of the concerns about testing in an arc jet environment is the presence of broadband electromagnetic interference (EMI) due to the arcing process used to generate the plasma flow. To investigate the potential impact of this EMI, we measured the signal levels at frequencies of interest (particularly in the 2.4-GHz band). Figure 2-2 shows a spectrum analyzer capture while operating at the NASA Ames Research Center’s PTF, which has similar testing characteristics to the AHF. The figure shows that the 2.4-GHz band contains little noise during the operation of the facility.
With the lack of EMI in the 2.4-GHz band confirmed, it was determined that a ZigBee radio would likely be able to communicate during arc jet operation. Subsequently, an opening in the arc jet schedule was found to accommodate a test in the AHF. The AHF can provide up to 20 MW of energy and pressures of up to 12 kPa. In addition, the five sample insertion mechanisms (called sting arms) can be separately controlled to allow multiple samples to be tested sequentially before breaking vacuum. This capability allowed the WLSS to participate in the AHF 289 test series in January 2010. Similarly, a schedule opening in the PTF allowed a secondary test to be performed with little reconfiguration.

2.6. Application Relevant Wireless Sensor Networks

The following are examples of wireless systems that are in the same design space as the WLSS. The first of these systems is shown by the design of a wireless sensor node using the ZigBee protocol for the purpose of temperature monitoring. This system, designed by
Veerasingam et al. [2009], is created for the purpose of temperature monitoring using the LM35. This sensor system utilized a ZigBee module to perform continuous temperature monitoring and transmitting of data to a remote station. This implementation is similar to the WLSS we implemented, although the full package of the WLSS was tailored to the application of TPS monitoring and temperature monitoring in the arc jet.

Flammini et al. [2006] developed a thermal monitoring system that uses the ZigBee protocol to monitor J-type thermocouples with a good quality of service (QoS). They examined the use of channel diversity on backup frequencies to avoid collisions from multiple sensor nodes. The authors successfully monitored data from multiple sensor nodes, although each sensor node only monitors a single thermocouple.

Systems specifically designed for TPS monitoring have also been developed by other groups. One of these systems, created by Milos et al. [2001], is a passive system that requires activation through an RF reader to collect data. This system was designed to be built into the TPS of a reusable launch vehicle. The authors detail a sensor that has a temperature fuse that will indicate whether the sensor has reached a certain temperature during operation. The system is designed using an RFID reader that stimulates the sensor and receives a signal of one frequency if the sensor has been tripped and a second frequency if the sensor has not. This is useful in avoiding the time that is usually taken to visually inspect the TPS for signs of any damage.

The second system has an active transponder and is currently battery powered and, as such, is the most similar to the system we developed. Milos et al. [2003] have created a wireless system that sends historical thermocouple data when actuated by a reader. This allows for a determination of the TPS performance from a specific test or flight. However, this sensor does not achieve real-time monitoring of the TPS, which the WLSS seeks to do. The compact nature of the system (25 × 25 × 6 mm) can be viewed as a future design goal for the WLSS.
2.7.  Chapter Summary

Wireless sensor systems are growing in importance for many monitoring tasks. Even niche applications have shown huge improvements in ease of use and QoS by transferring to a wireless system. At the same time, there is a demonstrated need for wireless sensors to replace legacy wired systems due to problems with ageing. The niche of aerospace thermal protection systems will benefit from both the ability to replace aged wired systems, as well as improved service by utilizing new wireless technology.
Chapter 3

System Description

The goal of testing within a space-relevant environment is met by the development of a wireless sensor system (WLSS). This system was designed with the clear goal of completing the AHF test detailed above, and creating a system that could be used for further testing, and to further the body of knowledge in the wireless sensor field. This chapter describes the hardware, software, and microcontroller firmware of the system from a systems overview. Partitioning between the software and firmware will be described for the purpose of understanding the system interactions.

3.1. System Specifications

The overarching goal of the system is to successfully test the WLSS within the NASA Ames Aerodynamic Heating Facility (AHF) in order to better understand the WLSS performance in a space-relevant environment. Specifically the performance of an IEEE 802.15.4-based wireless sensor network must be tested in close proximity to re-entry conditions to enable its future use in TPSs for re-entry vehicles. Additional specifications are aimed at determining what performance parameters are necessary in a migration from a wired system to a wireless system.

A set of specifications was developed for the WLSS in order to best match the goals of the tests. The initial system specifications are mostly focused on the hardware and firmware (the
firmware refers to the software that is resident onboard the microcontroller). Software that is resident to a nearby personal computer (PC) and used for network control and data collection is developed primarily to demonstrate the system specifications that exist for the hardware and firmware. The primary reason for this is that the usability of the PC software is not the focus of the initial project. Therefore, the system specifications will be separated into a hardware/firmware section and separately, PC software. Note, however, that requirements found within the first subset will apply to the second subset, because the user interface from the PC software demonstrates system compliance of the hardware and firmware. An example of this is the requirement of real-time data acquisition. The requirement exists for the hardware/firmware, but the PC software must verify incoming data in a real-time fashion. Note also that these two sets of requirements do not include the physical requirements, which are examined in Section 3.1.2. The requirements are based on the following goals:

- Conduct WLSS functional evaluation in the AHF;
- Conduct WLSS performance evaluation in the AHF;
- Demonstrate WLSS ability to acquire low and high rate temperature data in the AHF;
- Demonstrate WLSS ability to acquire multi-channel temperature data in the AHF;
- Verify WLSS as a high performance data acquisition system; and
- Validate WLSS as a viable replacement for WSS within AHF and in a space-relevant environment.

### 3.1.1. Functional Requirements

Based on the above goals, **Table 3-1** and **Table 3-2** summarize the hardware/firmware and software requirements for the WLSS.
<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement</th>
<th>Requirement Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The WLSS shall demonstrate support for multiple sensors.</td>
<td>16 single-ended, or 8 double-ended sensor inputs from 0–3.3 V depending on hardware gain selection.</td>
</tr>
<tr>
<td>2</td>
<td>The WLSS shall demonstrate high-frequency data gathering.</td>
<td>&gt;10 Hz data gathering on each channel.</td>
</tr>
<tr>
<td>3</td>
<td>Demonstrate real-time data gathering.</td>
<td>Wireless data must be transmitted in real-time as the data is being taken.</td>
</tr>
<tr>
<td>4</td>
<td>Powered without wires.</td>
<td>System must be low power. A battery or wireless power must be able to fit within the size/power constraints of the system.</td>
</tr>
<tr>
<td>5</td>
<td>The WLSS Network shall be robust.</td>
<td>Wireless network must automatically reconnect if there is a break in communication.</td>
</tr>
<tr>
<td>6</td>
<td>WLSS shall accept thermocouple data from Type-K and Type-R thermocouples.</td>
<td>Incoming voltage signals from thermocouples are between 0 and 50 mV. A gain of up to 66 (yielding 3.3 V at maximum input voltage) is desired.</td>
</tr>
<tr>
<td>7</td>
<td>WLSS shall accurately report sensor data.</td>
<td>A real-time temperature sensor is needed on the circuit for software cold junction compensation.</td>
</tr>
</tbody>
</table>
### Table 3-2: WLSS System Software Requirements

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement</th>
<th>Requirement Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The WLSS shall display support for multiple sensors.</td>
<td>Show graph with multiple lines and a chart with multiple columns.</td>
</tr>
<tr>
<td>2</td>
<td>The WLSS shall display high-frequency data gathering.</td>
<td>Show graph with high frequency responses and log data.</td>
</tr>
<tr>
<td>3</td>
<td>The WLSS shall display real-time data gathering.</td>
<td>Show chart with real-time data, and real-time indicators.</td>
</tr>
<tr>
<td>4</td>
<td>The WLSS shall demonstrate circuit power.</td>
<td>Show communication between PC software and WLSS.</td>
</tr>
<tr>
<td>5</td>
<td>The WLSS shall display accurate data.</td>
<td>Calibrate sensor inputs and utilize CJC input data.</td>
</tr>
<tr>
<td>6</td>
<td>The WLSS shall control the network and the data acquisition of the system.</td>
<td>Enable network control with direct inputs to the network control dongle.</td>
</tr>
</tbody>
</table>

### 3.1.2. Physical Constraints (AHF)

Although the WLSS is a prototype of a system that will evolve into a system suitable for space missions, the form factor of the WLSS must be such that it can be adapted to multiple test systems and, in particular, the test system in the AHF, the PTF, and the wedge test in the IHF.

The WLSS is initially designed to fit into the arc jet enclosure comprised of a calorimeter, tile holder, and sting arm adaptor as shown in Figure 3-1. An example of the calorimeter can be seen in Figure 3-2. The WLSS is required to fit into the inside of the tile holder, which is the white section shown in Figure 3-3.
As seen in these three figures, the WLSS must be packaged into a cylinder. A three-piece design is utilized in order to add a degree of adaptability to the system, i.e. allowing for interchangeable pieces. Three-board construction also provides extra PCB space to allow for a more debug-friendly circuit design with larger components and fewer routing layers. In order to meet the envelope constraints of the holder, the WLSS is designed to be a stack of circular boards measuring 2.7 inches in diameter and with a total height of 0.88 inches.

**Figure 3-1:** Cutout view of arc jet model.
Trey Morris, an Aerospace Engineering student at The Pennsylvania State University designed and fabricated the tile holder during his internship at the NASA Ames Research Center.

**Figure 3-2:** Calorimeter example showing TPS materials.

**Figure 3-3:** Example of complete test model showing the calorimeter in tan and the tile holder in white.

Trey Morris, an Aerospace Engineering student at The Pennsylvania State University designed and fabricated the tile holder during his internship at the NASA Ames Research Center.
Ames arc jet personnel completed the PICA calorimeter and the final assembly at the NASA Ames Research Center.

3.2. WLSS Description

The WLSS is a modular design consisting of three integrated functional elements. The functional partitioning of the system is shown in Figure 3-4. As seen in this figure, analog thermocouple (T/C) sensor data enters the system at the signal conditioning functional block. The T/C signals are multiplexed, conditioned, filtered and converted to serial digital data for processing by the controller functional block. Finally the controller/ZigBee element transmits the digital data to the PC resident software via the Tx block. This multiplexed technique limits the dynamic range of the sensor input signals in order to utilize a single gain/filter element, but offers the advantage of minimizing power and volume requirements.

![Figure 3-4: WLSS functional partitioning.](image)

3.2.1. Design Modularity

The first design goal for the WLSS is modularity. This means that the system is broken into functional subsystems, which are physically separated onto different printed circuit boards.
System modularity is beneficial because physical separation allows easier debugging, interchangeable parts, and incremental upgrades. Debugging is improved because each board has a distinct set of I/Os that it provides to each other board. These I/Os can be checked before the board stack is assembled to limit adverse interactions between the subsystems. Interchangeable parts are beneficial for the same reasons stated above: if one subsystem breaks down, a copy of that board can be substituted to minimize downtime. The same logic is applied to incremental upgrading of the system. If separate subsystems have distinct I/Os, another subsystem that is compatible with those I/Os can be swapped in at any point. As wireless power beaming becomes more viable, the battery board can be easily replaced with a module that can receive wireless power. These three subsystems designed for the WLSS are detailed in Figure 3-5 below.

Figure 3-5: WLSS functional diagram.

As Figure 3-5 shows, the three hardware subsystems are the signal conditioning board, the controller board, and the power board. The PC base station is also shown connected wirelessly to the other subsystems. These subsystems make up the physical WLSS, which interacts wirelessly with the PC base station. In the signal conditioning subsystem, the data from the sensors pass through noise filtration and signal scaling before entering a 16-bit analog-to-
digital converter. The digital signals are read into the MCU and stored, and a subset of this data is transmitted over the ZigBee network to the USB receiver and to the PC base station. The data flow can be seen visually in Figure 3-6.

![Figure 3-6: WLSS data flow diagram.](image)

The modularity of the system is shown in the relatively simple interconnects between subsystems. Power, data control, and data are the only signals that are passed through the headers. This will be detailed further in Section 3.2.2.

3.2.2. Physical Breakdown

The system has a circular form factor with a stack of three boards measuring 2.7 inches (6.86 cm) in diameter and a height of 0.88 inches (2.23 cm), as shown in Figure 3-7. The
systems are connected through headers, which are the black connectors (seen on the front of Figure 3-7). Each board can be removed and replaced as it is a pin-compatible system.

![Figure 3-7: Three board system. In the right image: the leftmost board (bottom) is the signal conditioning board, the middle board holds the power subsystem, and the rightmost board (top) is the controller board.](image)

3.2.3. **Functional Breakdown**

The functional breakdown of the system is also straightforward. As has previously been described, the signal conditioning subsystem, the power subsystem, and the controller subsystem are broken down between three separate boards. The power system provides 3.3 V to the other two boards, charges the onboard battery, and has an optional constant current source (CCS) for future sensor powering. The data functionality of the system is divided between the signal conditioning subsystem and the controller subsystem. In Figure 3-6, as the data exits the ADC, it is leaving the signal conditioning system and entering the controller subsystem. This is also seen in Figure 3-5 if a line were to be drawn to bisect the conversion block it would neatly separate the two subsystems. Figure 3-8 shows the simplicity that this partitioning creates. The subsystems will be described in Sections 3.2.4–3.2.6 below in the following order: power, signal conditioning, and control.
3.2.4. Power Subsystem

The power subsystem is arguably the simplest of the systems from the standpoint of requirements and design. The purpose of the power system is to provide a steady 3.3V to the other subsystems, and potentially provide a constant current source to one of the sensors. Wireless power obviously requires that the system not be connected to any steady power source. Ideally, wireless power beaming will be used to fulfill this goal, but until that point, it was determined that a single-cell high capacity lithium polymer (LiPo) battery would be used. This battery can be seen in Figure 3-9: Single-cell 3.7-V LiPo battery [from Moore, 2008].
A single-cell LiPo battery has a standard charge voltage of ~3.7 V. Since the desired power supply voltage is 3.3 V, the efficiency of a linear regulator will be close to that of a switching regulator. LiPo batteries are characterized by a discharge curve as shown in Figure 3-10, where cells remain at a fairly constant voltage for most of their life, and drop off significantly after a certain level. For a low power circuit, it is not reasonable to utilize a boost converter because the period of operation for the converter will not overcome the efficiency loss during the rest of the period of operation.

![Li-po Discharge Curve (GC)](image)

**Figure 3-10:** Lithium polymer battery discharge curve [from Moore, 2008].

The high efficiency linear regulator TC1107-3.3VOA is used because any gain in efficiency from using a switching regulator is negatively realized in system noise and complication. Additionally, because the regulated voltage is only nominally 0.4 V under the regular battery voltage, only 16 mW (~40 mA current draw during typical operation) is lost through the regulator. The battery is connected so that J2 functions as a charger port when SW1 is toggled as seen in Figure 3-11. A Zener diode provides reverse polarity and overvoltage protection. In addition, an onboard optional constant current source (LM334/SO) can be used in
the event that a sensor requires power. The circuit diagram can be seen in Figure 9-1 in Appendix C.

![Circuit Diagram]

**Figure 3-11:** Power subsystem simplified schematic.

The layout of the power subsystem is shown in Figure 9-2 in Appendix C. The large white area in the center of the board is the battery, while most other components are placed on the opposite side of the board. There are no oscillating electronics onboard so no ground plane was needed. However, noise in the signal conditioning board may be somewhat helped with a ground plan on this central board. **Figure 3-12** shows the populated circuit boards in a top and bottom view.
3.2.5. Signal-Conditioning Subsystem

The purpose of the signal-conditioning subsystem is to convert the sensor inputs into a digital output that can be read by the microcontroller. Although the MCU controls the MUX and the ADC to dictate when the signal conditioning system will produce data from the sensors, the electronics for converting an analog sensor input into a 16-bit digital output is contained on this board. Figure 9-3 in Appendix C shows the full schematic for this subsystem.
The sensor inputs are selected using the MUX (ADG707BRU) control lines coming through the header from the MCU. This MUX was selected because it is low-voltage, single supply, and has a very fast switching rate (40-ns switching time). The output from the MUX is sent through the AD8236ARM instrumentation amplifier as a gain stage. The AD8236ARM was originally selected because of its extremely low power draw at 40 µA maximum supply current. However, this in-amp was replaced with the AD8223-ARZ because the AD8236ARM was found to have non-linear gain in the mV range prior to the first test. The data stream then passes through the LMP2231BMF op-amp, configured as a Sallen–Key low pass filter. The LMP2231 was chosen for its low power characteristics. This op-amp was utilized for a non-inverting gain stage during the first test due to issues with the in-amp gain stage. The data finally passes through a 16-bit ADC (AD7680BRM) that is controlled, again, by the microcontroller. Once again, this ADC was chosen due to its low power characteristics, as well as its low supply voltage operation. Optocouplers were included to avoid digital/analog crossover, and to attempt to

Figure 3-13: Signal-conditioning subsystem simplified schematic.
decrease noise. These optocouplers (ACSL-6410-00TE), however, were not utilized due to incorrect signal termination. Noise was visible during testing with thermocouples, and it is likely that the inclusion of the Sallen–Key LPF and the optocouplers in a future design will alleviate much of this problem.

The layout of the signal conditioning subsection is shown in Figure 9-4. During layout, an attempt was made to isolate analog signals and power from digital I/O. This is partially accomplished through the use of an extensive ground plane shown in grey, and the use of tantalum capacitors and isolation inductors connecting the power supplies. Jumpers J2–J4 and J7–J22 are utilized to adapt the system to single-ended or double-ended measurements. The thermocouples used for temperature measurement on channels TC_1–TC_4 produce double-ended measurements, requiring that jumpers J2–J4 and J7 are jumpered, while J15–J22 must be unpopulated. In addition, for initial testing, pin 18 on the MUX is directly jumpered to 3.3 V because the slew rate of the MUX is faster if just the inputs (A0–A2) are changed rather than utilizing this enable/disable pin. Figure 3-14 shows the populated circuit in both a top and bottom view.
3.2.6. Control Subsystem

The purpose of the control subsystem is to take data measurements through the signal-conditioning subsystem, and store and transmit this data as they come in. The system houses the MCU, the wireless module, the memory storage, and all associated peripheral components. The control system schematic is shown in Appendix C, Figure 9-5 and Error! Reference source not found.. A simplified view of the subsystem is shown below in Figure 3-15.

Figure 3-14: Signal-conditioning subsystem circuit top/bottom view.
The microcontroller (U3) is a PIC16F886-I/SS. This particular microcontroller was chosen due to pre-existing PIC sample code utilized on a previous system. It is also a relatively low powered system that we knew was able to connect and work with the Telegesis ZigBee module. The MCU reads in the data by ‘bit-banging’ the convert and clock line of the analog to digital converter (ADC) and then reading the data input line (DIN). This is due to the MCU not having enough serial ports to communicate with the ZigBee module, the external memory, and the ADC. The internal ADC is only 10-bit, and so could not be used in the collection of the thermocouple data. However, an 8-MHz on-chip clock was used to control the data flow. The chip has the added functionality of having power-saving sleep modes, but these were not used. The ETRX2 Telegesis ZigBee chip was chosen for its simple communication protocol called the AT Command Interface. This communication method does not give the type of flexibility that

Figure 3-15: Control subsystem simplified schematic.
can be expected with unmodified Ember, TI or Freescale chips; however, the time of
development is significantly decreased.

In addition to the MCU, the ZigBee transceiver, and the associated debugging and
programming chips, an important temperature sensor is placed near the MCU. The LM35DM
was replaced by the TMP36 due to voltage requirements of the LM35 and is utilized for cold-
junction compensation (CJC) of the thermocouple readings.

As this is primarily a digital circuit, appropriate measures were included in the circuit
layout. A ground plane was used to try to cut down on noise and reflections, while pad and trace
lengths were designed to be minimal. The layout can be seen in Figure 9-7. One flaw of this
layout is in the positioning of the Telegesis ZigBee chip. The Telegesis chip should be placed so
that the right side (large rectangle in the center of the image as viewed in Figure 9-7) is
overhanging the ground plane. Figure 3-16 shows the populated circuit board in a top and
bottom view.
3.3. **Microcontroller Software**

The MCU software was written using CCS C code in the Microchip MPLAB IDE suite. The software controls the collection, transmission, and storage of sensor data coming from the signal conditioning board. The microcode is attached in Appendix B. The MCU acts as a Busy Polling Machine, and constantly searches a series of flags that are controlled by the RDA and timer2 interrupts. This software architecture is shown in Figure 3-17.

![Control subsystem circuit top/bottom view](image)
All communication with the Telegesis ZigBee module is through the serial port with the use of the AT Commands. A full list of these commands can be found in “Telegesis AT Commands” [2010]. The most widely utilized, however, are the broadcast, unicast, and data channel commands, which send a message to the whole network, send a message to a specific wireless device, or send an unverified stream of data between two devices at the maximum

**Figure 3-17**: Microcontroller firmware flow diagram.
frequency (19200 baud) respectively. The syntax for these commands is provided in either the Telegesis literature or in the ZigBee communication portion of the code.

Data are read from the signal conditioning ADC when timer2 triggers the data flag. At this time, the data are always stored in onboard memory. However, at a subset of this frequency, the ucast flag is set and data are sent out in a unicast to the PC software. The PC software is able to utilize this data to create a real-time display and log of the data. After the test is completed, which occurs by either the user stopping the data transfer or when the maximum amount of data transfer is reached, the entirety of the data collected by the data flag can be streamed to the PC.

The chosen microcontroller does not have enough serial communication ports. This forces the communication of the ADC data line to be through a secondary data port. The secondary data port must be manually driven high and low to control the ADC. This is completed using the code shown in Figure 3-18. This code is formulated to walk through the 16-bit data while incrementing the serial clock and the conversion bit. The ADC_DIN variable references the data input pin on the microcontroller and is read at every serial clock. In this way, it was possible to reach a maximum data rate of ~32 kHz. Taking the single instruction time for an 8-MHz clock (i.e., 125 ns) and multiplying it by the number of instructions needed to complete a get_adc_data subroutine (247) gives the approximate maximum data rate. This rate is a flawed number because it does not take into account the storage of the data, communication with the network, or other subroutines. However, it approximates a pure maximum data rate given this type of software serial-port mimic and the speed of the clock.
The purpose of the PC software is to monitor the status of the WLSS, collect the data that the WLSS is transmitting, and send a series of commands to the WLSS to instruct the system to start collecting data, stop collecting data, and send the stored high frequency data. The PC software manages the ZigBee network through the use of a ZigBee coordinator USB dongle. The software was written using the National Instruments LabVIEW visual programming language, which allows for a nearly seamless integration of the graphical user interface (GUI) with the operational code. The front panel of the LabVIEW CUIP code is shown in Figure 3-19. As can be seen on the front panel, the software provides control for the ZigBee network and the WLSS, as well as displaying the data being transmitted from the WLSS in both a tabular and graphical

```c
int16 get_adc_data(chanConv) {
  int16 temp_val, tmp1;
  int c;
  temp_val = 0b0000000000000000;
  output_low(chanConv);
  // cycle through four clock cycles to get rid of the first 4 zeros
  for (c = 0; c <= 3; c++) {
    output_high(ADC_SCK);
    output_low(ADC_SCK);
  }
  for(c = 0; c <= 15; c++){
    output_high(ADC_SCK);
    if (input(ADC_DIN)) tmp1 = 0b0000000000000001;
    else tmp1 = 0b0000000000000000;
    temp_val = (temp_val|(tmp1<<(15-c)));
    //nine clock cycles before here
    output_low(ADC_SCK);
  }
  output_high(chanConv);
  return temp_val;
}
```

Figure 3-18: Microcontroller code for ADC data gathering.

3.4. PC Control and Monitor Software

The purpose of the PC software is to monitor the status of the WLSS, collect the data that the WLSS is transmitting, and send a series of commands to the WLSS to instruct the system to start collecting data, stop collecting data, and send the stored high frequency data. The PC software manages the ZigBee network through the use of a ZigBee coordinator USB dongle. The software was written using the National Instruments LabVIEW visual programming language, which allows for a nearly seamless integration of the graphical user interface (GUI) with the operational code. The front panel of the LabVIEW CUIP code is shown in Figure 3-19. As can be seen on the front panel, the software provides control for the ZigBee network and the WLSS, as well as displaying the data being transmitted from the WLSS in both a tabular and graphical...
format. The data are additionally stored in a series of CSV text files in its various forms of conversion.

![Image of LabVIEW front panel](image)

**Figure 3-19**: LabVIEW front panel.

The LabVIEW program consists of three while-loops that run in parallel. The three loops are the serial controller, the event handler and the data parser. Queues that act as FIFOs control the flow of serial data into and out of the serial controller, allowing the serial loop to operate freely without losing data during conversion and processing. The serial controller loop (**Figure 3-20**) reads and writes any serial data from or to the virtual serial port where the USB network coordinator is connected. It also searches for a TTL signal from an NI peripheral DAQ that is used as a system sync.
The event handler shown in Figure 3-21 interprets commands from the front panel and writes data to the outgoing network queue as needed. The events are driven by real-time clicks on the front-panel and act as interrupts within the event handler. However, the queue will buffer the outgoing messages so that the network does not become overwhelmed.

Figure 3-20: Serial controller loop.

Figure 3-21: Event handler loop.
The parser loop reads all of the incoming messages from the serial incoming queue. Every time that a message comes into the system from the WLSS, this loop will interpret the message and then update the front-panel to reflect the status of the device and network. This information includes the real-time data returned from the WLSS that is plotted and displayed on the real-time indicators. Where needed, the data parser will also convert data appropriately from sensed voltages to sensor readouts utilizing code shown in Appendix B. The data parser is shown in Figure 3-22.

Figure 3-22: Data parser loop.

3.5. Chapter Summary

The WLSS is comprised of three subsystems that are located on three separate PCBs connected by headers. The three-piece architecture provides an ability to modularly design upgrades and debug the system. The software and firmware were created to complement the WLSS hardware and enable the system to display its capabilities in high and low speed data gathering, real-time wireless transmission, and multiple sensor support.
Chapter 4

Proof of Concept Tests and Analysis

The WLSS, with capabilities similar to that of a wired sensor system, was tested in the AHF 289 test series in January 2010 and also in the PTF at the beginning of September 2010. This chapter details the tests performed on the WLSS in the AHF and at the PTF, as well as details of the upcoming wedge test in the IHF. The purposes of these tests are discussed, descriptions of the tests are given, and the results of the testing are shown.

4.1. AHF Test

4.1.1. Test Purpose

The purpose of the AHF test was to examine the WLSS and evaluate its performance compared to the wired data collection system already in place at the NASA Ames facility. AHF Test 289 sought to demonstrate high frequency data gathering as well as to establish wireless RF communication with the module for the entirety of the test. The WLSS additionally was required to demonstrate real-time communication. It is of note that the targeted WLSS capabilities do not reflect the maximum capabilities of the wired system that is currently in place. Table 4-1 provides the comparison between the WLSS targeted capabilities and the current wired system.
Although the initial WLSS prototype does not have equal capabilities to NASA’s wired system, the requirements more closely reflect a system to be used on a TPS.

### Table 4-1: Data acquisition systems capabilities.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Wireless System Capabilities</th>
<th>Wired System Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Type</td>
<td>Wireless ZigBee Network</td>
<td>Hard-wired</td>
</tr>
<tr>
<td>Real-Time Data Rate</td>
<td>2 Hz</td>
<td>10–100 Hz</td>
</tr>
<tr>
<td>High Frequency Data Rate</td>
<td>34 kHz (ideal)</td>
<td>10–100 Hz</td>
</tr>
<tr>
<td>Number of Channels</td>
<td>8 differential</td>
<td>32 channels</td>
</tr>
<tr>
<td></td>
<td>16 single-ended</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 Gardon gauge</td>
<td></td>
</tr>
<tr>
<td>Channel Resolution</td>
<td>16-bit</td>
<td>16-bit</td>
</tr>
<tr>
<td></td>
<td>Calorimeter: Gardon</td>
<td>Pressures: Barocel, Statham, ESP</td>
</tr>
<tr>
<td></td>
<td>Miscellaneous analog sensor with voltage output up to ±3.3 V</td>
<td>Calorimeters: slug and Gardon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Radiometer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pyrometer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Miscellaneous analog sensors with voltage output up to ±10 volts or current loop output in any milliamp range</td>
</tr>
<tr>
<td>Gain</td>
<td>Hardware-reconfigurable gain</td>
<td>Software-reconfigurable gain (1–1024)</td>
</tr>
<tr>
<td>Real-Time Display</td>
<td>Unlimited channels</td>
<td>8 real-time channels</td>
</tr>
</tbody>
</table>

#### 4.1.2. Test Overview

The AHF Test 289 was completed on January 27, 2010 in the NASA Ames AHF. Two separate runs were performed using the same WLSS contained in two separate tile-holder/calorimeter packages. This so called ‘slug’ has a four-inch diameter and can be seen
before the test in Figure 4-1. The slug is placed into the AHF heating chamber in the model support system as shown in Figure 4-1. The arc jet has the capability of testing with five different sting arms, which can be independently placed into the stream to enable testing multiple samples in a single run. These sting arms can be seen in Figure 4-1. The WLSS was mounted onto Sting 1 for both tests. This allows Sting 1 to be able to be placed into the stream for an arbitrary period and then taken out separately from any other sample. A second sample is included in run two because the WLSS was able to share tests with a TPS model from NASA Ames.

Figure 4-1: WLSS tile-holder and TPS.
Typically, the sting arm is used to route signal and power wires from the test model to the wired system. In the case of the WLSS, as the TRL increases and the WLSS replaces wired systems that are currently in place, the sting arm will no longer have to be wired. The wiring of the sting arm is one of the primary time sinks during reconfiguration of the AHF testing system, and the elimination of this procedure is one of the primary drivers for the replacement of the wired system with the easily-reconfigurable wireless system.

Figure 4-2: Arc jet diagram [from Eddlemon, 2005]
In the AHF Test 289, one of the primary goals was to verify the data accuracy of the WLSS. For this reason, a ‘Y’ configuration was used to attach the four thermocouples (three type-K and one type-R) to both the WLSS and the wired data acquisition system. This arrangement can be seen clearly in Figure 4-3. The wired connection to the facility has an input impedance of 10 k\(\Omega\) while the WLSS has an input impedance of \(~1\) M\(\Omega\). The Y configuration approximates two resistors in parallel to the voltage source that is the thermocouple. Because of this, the voltage sensed by each system is minimally impacted due to the difference in line resistance seen by the thermocouple. If these line resistances are 10 \(\Omega\) (an extremely high estimation) there is less than a 1.2% difference between the two setups. For this reason, it was expected that the Y configuration would not impact the readings of either of the sensors.

Figure 4-3: In-chamber view of the AHF arc jet.
4.1.3. Test Results

The sample was heated in the arc jet for 60 seconds at 170 W/cm$^2$ and data were taken at a 60 Hz data rate by the wired acquisition system. Figure 4-5 Error! Reference source not found. shows the WLSS in the arc jet stream. The sheath of plasma can be seen on the outside of the package, which we found blocked the wireless communication from the WLSS. Figure 4-6 Error! Reference source not found. shows the sample after exposure in the arc jet. The charred TPS was instrumented with thermocouples that were used to demonstrate the WLSS’s successful operation.
During the first run, real-time data were collected by the WLSS. This data is shown in Figure 4-5. However, this data collected by the WLSS were corrupted by a combination of user operator error and a systems oversight. The instrumentation amplifier could not be utilized for this test and therefore the thermocouples were set up as single-ended measurements. This meant that one end of each of the thermocouples was tied directly to circuit ground. Unfortunately, the PICA material that was used for the TPS is extremely conductive. The signals of all of the thermocouples that were embedded within the PICA material were shorted, leaving completely nonsensical data. The user error came about...
because it was unknown whether radio blackout would occur while the device was in the stream. This led to communication attempts during the period of radio blackout that were then relayed to the device when it had rejoined the network. This caused the system to wipe its stored data and restart data collection at a random period in time that is not recorded on the client side. Therefore, it is impossible to line up the test 1 data from the wireless client with that of the wired client and this data is not further analyzed.

![AHF Test 289 (test 001)](image)

**Figure 4-7**: Real-time data from run 001.

During test 2, data were collected from the AHF test at 60 Hz with the wired system, 10 Hz with the wireless system, and at 1 Hz with the wireless system in real time. Only one thermocouple was measured due to the hardware issues discussed above. These data are visible in the graph below (Figure 4-8 Error! Reference source not found.). There was a break in the real-time communication between 15 and 75 seconds, a period of 60 seconds, which reflects the period that the WLSS spent inside of the arc jet encased in a plasma sheath. This period of radio blackout is expected at this plasma density because of the envelope of plasma that forms around the sample. However, communication was restored after the sample was taken out of the arc jet.
The wireless real-time data were able to be collected during the test except for during this period of radio blackout. The device also was able to collect the wireless streamed data (high frequency data) during this period of radio blackout. However, the high-speed (stored) data were corrupted partway through its data stream transmission, so only data from the 100-second mark through the 230-second mark were able to be collected.

![Figure 4-8: CUIP Test 2 WLSS/WSS comparison.](image)

### 4.2. PTF Test

#### 4.2.1. Test Purpose

The PTF test, completed at NASA Ames, was designed to demonstrate the improved WLSS with demonstration of multichannel sensor data acquisition, as well as data acquisition with multiple sensor types during a single test. Additionally, the system was configured to
demonstrate a 100% link time between the WLSS and external instruments, illustrating system reliability. The different sensor types were used to evaluate the response of the WLSS to a variety of sensors and signal types.

4.2.2. Test Overview

The PTF WLSS utilized three thermocouples, one Gardon gauge, and one pressure transducer, for a total of five sensors. A 2.4-GHz antenna was mounted at some position along or below the sting arm to transmit data from the WLSS to the monitoring station throughout the entirety of the test. In addition, data was sent to the monitoring station via the Y configuration described in Figure 4-4Error! Reference source not found..

4.2.3. Test Results

Initial turn on and charging of the system were accomplished through the access of switches and a charge port located in the PTF’s patch panel. However, the patch panel ceramic overlay protection system complicated accessibility, thus extending the pre-test operating time as compared to the easy access of the AHF patch panel that was used in the AHF Test 289. Pre-test setup and initialization time of the PTF forced a truncation of the battery charge time two hours prior to test; therefore, full charging of the batteries did not occur.

During the initial two-hour period, the WLSS showed reliable connection and accurate data measurement of the ambient air through its connected thermocouple and on-board cold junction compensation circuit. However, twenty minutes before the start of the PTF test sequence, an anomalous rise in temperature was recorded before the WLSS lost contact with the laptop base station. The root cause of the WLSS anomaly could not be investigated until after the PTF test
sequence was completed. It is believed that the temperature anomaly and the loss of communication with the laptop base station were due to the response of the WLSS to the decay of the battery voltage as it discharged.

After completion of the first PTF test sequence, the dead battery condition was confirmed. Recharging of the WLSS battery commenced at this point for a span of 10–15 minutes. During the set-up time for the second PTF test sequence, successful connection to the embedded system was reestablished and continued for approximately 30 minutes. During this time network connectivity and collection of accurate data measurements were verified. However, as a result of a leak in a water coolant line, the start of the PTF test sequence was delayed and the battery power failed for a second time. As a result, sensor data could not be acquired during the second PTF test sequence and the goal to demonstrate 100% link uptime during exposure to the plasma layer was not accomplished.

4.3. IHF Wedge Test

4.3.1. Test Purpose

The ‘wedge’ test will be completed in the NASA Ames IHF. It is designed to showcase a fully functional WLSS that will demonstrate accurate low and high frequency data gathering, 100% link uptime, multiple sensor support, and system reconfigurability and reusability.

4.3.2. Test Overview

The WLSS electronics package is to be housed as closely as possible to the sensors and be non-intrusive with respect to concurrent experiments. It is also desirable to minimize
modifications to the existing facility in order to reduce the time and cost associated with testing. The IHF wedge, similar to the PTF, will utilize three thermocouples, one Gardon gauge, and one pressure transducer, for a total of five sensors. A 2.4-GHz antenna will be mounted at some position along or below the sting arm to transmit data from the WLSS to the monitoring station throughout the entirety of the test. In addition, data will be sent to the monitoring station via the Y configuration described in Figure 4-4.

The current interface design consists of a copper-101 box with 1/4-inch thick walls that is fastened to the side of the wedge’s sting arm “plug and play” adapter. It is not expected that the housing would be impacted by the full thermal loads of the plasma flow, but in order to minimize heating imparted to the electronics, the housing will be protected by LI-900 or similar TPS material. To provide additional thermal protection, as well as electrical isolation, the circuit boards are separated from the copper housing using ceramic standoffs. This is shown in Figure 4-9.
The thermocouples are to be provided by and fitted to the test setup by the WLSS team. At least one thermocouple will be located on the WLSS electronic boards and the remaining thermocouple(s) will be mounted to the test housing or wedge. Exact locations are not predetermined or critical for the test.

4.4. **Chapter Summary**

Two tests were completed at the NASA Ames Research Center, and a third test is pending. The first test in the AHF, test no. 289, was able to demonstrate real-time wireless data gathering as well as high-speed data gathering. The WLSS was not able to demonstrate multiple sensor support in the first test due to the sensor configuration in the test material, and the system experienced radio blackout during the period while the system was in the arc jet.

The second test in the PTF was able to show multiple sensor support before battery depletion and the system lost power. Further testing is necessary so that the system can provide valid data in real-time from all sensors without radio blackout.
The upcoming IHF wedge test will demonstrate a fully functional WLSS, showcasing accurate low and high frequency data gathering, 100% uptime, multiple sensor support, and system reconfigurability and reusability.
Chapter 5

Analysis of Test Results

This work developed a WLSS that was able to collect wireless thermocouple data with a battery-powered system during testing in an arc jet. The data that were collected are compared with data that were collected by a parallel wired system. Capabilities of the WLSS are examined compared to the desired characteristics and solutions are proposed.

5.1. Result Discrepancies

As can be seen in Figure 4-8 above, the wireless data do not directly map onto the wired data as would be expected from a successful deployment. The goals of creating the system were to demonstrate high-speed data gathering, real-time data acquisition, and support for multiple sensors. Additional goals were to show that the wireless system could function at the same level as a wired system. However, as shown in the figure, the two data sources do not align, and only one channel of data is usable.

5.1.1. Data Noise

Although it is difficult to see in Figure 4-8, the wireless data are not nearly as smooth as the wired. The streamed wireless data at 60 Hz are compared to the wired data at 60 Hz in Figure 5-1. This subset of data shows both an offset and noise inherent in the thermocouple data.
The noise that is shown in the data can either be attributed to noise in the signal-conditioning circuitry or noise in the wireless line. Noise inherent in the circuitry can be tolerated at this stage because it is easily fixed in another revision. Noise of this magnitude in the wireless protocol, however, cannot be tolerated because it brings into question the ability of the protocol to be used for this application. However, the same wireless data were retrieved twice, once while the arc jet was in operation, and a second time while after the arc jet ceased operation. This shows that the noise comes from the signal conditioning of the thermocouple and/or noise inherent in the WLSS hardware, not in the wireless protocol.

![Wireless and Wired at 60Hz](image)

**Figure 5-1:** Wired and wireless data at 60 Hz.

5.1.2. **Data Offset**

The more noticeable observation when looking at the data plot is the offset seen between the wireless and wired data points. The offset, however, does not markedly change the slope of
the wireless data as shown in Figure 5-2. However, as shown in Figure 5-3, the data offset is not constant over the entire period of data acquisition, and actually increases from 23 to 35 °C in a non-linear fashion. To explain this error, further tests are completed. It was hypothesized that this offset was due to the lack of cold-junction compensation of the thermocouples. Therefore, measurements were taken by the WLSS within a thermal chamber, as well as taken by a compensated thermocouple known to report true temperature in the chamber. A third thermocouple was finally placed at the cold junction of the thermocouple. Software post-processing is used to compensate the WLSS thermocouple reading with its cold junction temperature. The results are shown in Figure 5-4, Figure 5-5, and Figure 5-6. As can be seen in these figures, the thermocouples that are compensated with CJC map the WLSS data exactly to the wired data. This, along with verified modifications from the second WLSS taken to the wedge test, shows that CJC successfully reports thermocouple data.

![Wired and Wireless Real Time Slope](image)

**Figure 5-2:** Wired and wireless real-time slopes.
Figure 5-3: Data offset.

Figure 5-4: Rising temperature in CJC test.
Figure 5-5: Peak temperature in CJC test.

Figure 5-6: Falling temperature in CJC test.
5.1.3. Multiple Sensor Support

Multiple sensors could not be tested with the initial version of the WLSS due to the PICA conductivity interfering with the single-ended measurements of all but one thermocouple. However, this was fixed with the replacement of the in-amp in the first generation with a working model, enabling differential amplification of the sensor data. This was verified during the PTF testing before battery failure caused the system to shut down.

5.2. Conclusions from Results

The results demonstrate the capabilities of the WLSS. The wireless system can be made to perform at a level that is similar to a wired acquisition system. All of the problems in the initial WLSS concept can, and have been addressed in a new WLSS concept that will be tested in the upcoming wedge test.
Chapter 6
Conclusions and Future Work

6.1. Conclusions

The AHF and PTF tests provided an excellent baseline of WLSS performance toward validating the WLSS as a replacement for the legacy wired systems. The AHF test was able to demonstrate the viability of high-speed data communication, although the non-real-time nature of this high-speed communication may slow the adoption of such a system. In addition, the AHF test demonstrated the robust nature of the network in its ability to immediately reestablish communication after the predictable plasma-induced radio blackout occurred. The PTF test demonstrated multiple sensor support, despite the charging difficulty that stopped the test before its completion. The charging anomaly experienced during PTF tests exposed a characteristic that differentiates a WLSS for test bed applications from one that is used for flight applications.

The WLSS brings a level of adaptability to each application that would otherwise not exist. However, it is more important in testing applications for adaptability in sensor configurations than for adaptability in sensor placement. The opposite can be true for the WLSS configuration in flight applications. For these reasons, focus on low power is necessary for flight applications to enable battery or harvested power to be utilized and to give unrestricted placement capabilities. For the WLSS to be used in testing applications, power can usually be applied to the system. This allows the low power considerations to be significantly reduced in importance, and for data rate and number of sensors to have a higher importance.
In conclusion, the WLSS is an initial step toward wireless sensor network applications on a TPS. The successful network formation and data gathering, although leaving a clear path toward improvement, were still able to provide proof of concept toward a system that can continue to evolve and become a part of future aerospace missions.

6.2. Future Work

These considerations create two separate paths for development. Future work must include miniaturization of the system to a smaller envelope, while increasing accuracy and sensor flexibility. However, the next requirements will require parallel paths. A higher frequency of real-time data will be needed for testing applications. Although an eventual goal is to have similar instrumentation on both the flight TPS and the testing TPS, in order to gather enough data for the requirements of the testing, a network with a higher data rate will still be needed. Unfortunately, as data rate is increased, so does the power consumption. This is at direct odds with the WLSS for use in TPS materials for flight. This WLSS needs to have extreme power consideration, as it must document entire flights without running out of power. Additionally, the TPS system must be much smaller and will potentially have fewer sensors depending on the desired density.

A possible future platform that would allow a flight demonstration of the WLSS technology is NASA’s Inflatable Re-entry Vehicle Experiment (IRVE) [NASAfacts, 2009]. The requirements shown in Table 6-1 below reflect a logical next step for this system. The next step for the testing application is to switch out the power from the battery or wireless power, and instead use a line power from a 3.3 or 5 Vdc source. This will allow higher uptimes and better transmission power for this future system.
Wireless power will likely need to be implemented in the future if real-time data gathering is expected to take off. Work at Penn State is ongoing with the aim of wireless power beaming using nonradiative coupling of magnetic fields, with proof-of-concept shown [Singh, 2009]. The Penn State team has achieved encouraging results that point toward the possibility of battery replacement on the WLSS system. With the TPS application, the WLSS will not be able to have a replacement battery since it is expected to out-last the TPS materials. For this reason, the state of the art must be furthered by integrating the wireless power beaming along with wireless data transmission. This can create the next generation wireless sensor network.
Appendix A

WLSS Selected Microcontroller Firmware

///////////////////////////////////////////////////////////////////////
/*
This Program will program a PIC microcontroller to collect data from up to 8
thermocouples and then beam that data to a centralized computer running the
labVIEW interface over a ZigBee network. The microcontroller (MCU) will
control the ZigBee telegesis module using AT commands over the RS232 port of
the MCU and ZC (ZigBee Controller).

Erik Weir
erik.weir@psu.edu
*/
///////////////////////////////////////////////////////////////////////

// PRIMARY COMPILING SWITCHES

// 10Hz or 100Hz software, select one
//#define 10Hz    //defines an every tenth sample rate
//#define 100Hz   // defines a constant sample rate

// Define PIC type
#define 16F886

// END OF PRIMARY COMPILING SWITCHES

// delay after any Zigbee command
#define ZigbeeDelay 70
#define ADCDelay 1

#define packetsize512 64
#include "16F886.h" // 10/30/2007 change

#define Fuse NOWDT,NOPROTECT,NOLVP,INTRC_IO,NOBROWNOUT // Use WDT in case
bad powerup or surge, internal RC osc (next #use statement will set speed,
9/4/2007 to drop current ~100ua in sleep
#define delay(clock=8000000, restart_wdt)
#define rs232 (baud=19200, ERRORS, xmit=PIN_C6, rcv=PIN_C7)

//byte OSCCON = 0x90

// final adc channels
#define Adc_Temperature 1
#define ADCSpeed 16 // ADCSpeed = 4 > 4KHz, 11/7/07 measured 4000 samples/sec with setting of 4(period 125), CPU clk=8MHz; should be 4000
#define num_dat 30 // 11/10/2009 CHANGE to 3600 for 60Hz data acquisition rate
#define active_channels 5
#define num_minutes 10 // MAX @ 4 channels is 4 minutes.
#define data_freq 2 //careful.... this doesn't actually do anything except set the max number of data points
#define ymax 12 // frequency of data collection = 60/ymax Hz
#define zmax 120 // frequency of CJChan = 60/zmax Hz

// pin defs for pic 16F886
#define ADC_SCK PIN_A0
#define MUX_A0 PIN_A4
#define MUX_A1 PIN_A5
#define MUX_A3 PIN_A6
#define MUX_A2 PIN_A7
#define ADC_DIN PIN_B1
#define LED2 PIN_B2
#define ADC_CONV1 PIN_B5
#define TempPower PIN_B4 // power supply pin for temperature sensor chip
#define ADC_CONV2 PIN_C0
#define LED1 PIN_C1 // LED pin definitions 11/16/2009 - changed C2 to C1 to reflect schematic
#define EEPROM_SCL PIN_C3
#define EEPROM_SDA PIN_C4 // EEPROM pin definitions
#define INTS_PER_MINUTE 15 // ((32768/(2*65536))* 60 =15 for 1 minute

// status definitions
unsigned byte status = 0; // status byte
#define tick status,4 // tick flag for Acceleration acq.
#define Nobuff status,5 // double buffer flag for Acceleration acq.
#define ErrorDetect status,6 // Flag embedded in 3 byte of data for error detection
#define Powerup status,7 // Flag power up

//moved this include statement below the pin definitions
#include <24256_64_loop.c>
// comment - oddly, packet size of 64 seems to be the largest even though compiler reports space available (85-98% used)?? Removing debugger code did not seem to help.
static char packet0[packetsize512]= "01234";
#define LOCATE packet0 = 0x20

// input character for all serial reads
byte keybdbyte = 0;
char shift_char_array[12];
char bcast[6];
int16 d_buff[] = {0,0,0,0};

char START_array[] = {'S','T','A','R','T'};
char END_array[] = {'E','N','D'};
char DATA_array[] = {'D','A','T','A'};
char MIN_array[] = {'M','I','N'};
byte bcast_flag = 0;
byte end_flag = 0;
byte inputflag = 0;
byte joinflag = 0;
byte possible_bcast = 0;
byte start_flag = 0;
byte data_flag = 0;
byte reduced_data_flag = 0;       // for measuring CJChan at a lower
   frequency relative to data_flag rate
byte data_ready_flag = 0;
byte waitingforopen = 0;
byte openflag = 0;
byte ucast_flag = 0;

long long int data_collected;
int packet_index;
long int ptr;
long int packet_num;
int8 int_count = 0;                // Number of interrupts left before a minute
   has elapsed
int32 temp = 0;
int x =0;
int y = 0;
int z = 0;
long long int max_data;
long int ucast_count;

#include <TelegesisZigbee.c>

// Subroutine declarations
extern void set_mux_chan(int m_chan);
extern char IsCommand(char *bc, char *ta, int len);
extern void interpret_comm();
extern int16 get_adc_data(chanConv);
extern void clear_eeprom();
extern int16 measure(int channel, int average);
extern void mcu_setup();
extern void TempRead();

/////////////////////////////////////////////////////

/////////////////////////////////////////
// main program //
/////////////////////////////////////////

void main() {
    int16 t1,t2,t3;
    int k,j;
    int preschan = active_channels-1;
       //initialize cold junction channel number

    mcu_setup();
       // setup mcu
    enable_interruption(1); // setup INT_RDA;
    enable_interruption(GLOBAL);

    max_data = num_minutes*60*data_freq*active_channels;    //
        frequency*(60 seconds/min)*(number of minutes)*(number of active channels)
    printf("max data = %lu.",max_data);
       // max value is
// Init Zigbee on first power up and join network. Wait here until it joins a
// network, lengthening wait delay up to 5 seconds so that the process only happens
every 5 seconds
if(!bit_test(Powerup)) {
    ZigbeeInit();
    printf("join cmd sent\n\r");

    k=0;
    while(ZigbeeJPAN() == 0) {
        // wait to join PAN
        k++;
        //
        for(k = 0; k < sizeof(shift_char_array); k++)
            printf("%c",shift_char_array[k]);
// printf("\n\r");
        delay_ms(5000);
        printf("ZigBee join failed: %u trying again!\n\r", k);
    }

    delay_ms(ZigbeeDelay);
    // settling time

    // special LED to show that we've joined the zigbee network
    for(k=0;k<=3;k++) {
        output_high(LED1);
        // light LED
        delay_ms(250);
        output_low(LED1);
        // LED out
        delay_ms(50);
        output_high(LED1);
        // light LED
        delay_ms(750);
        output_low(LED1);
        // LED out
        delay_ms(50);
    }
    bit_set(Powerup);
    // no longer in powerup
    printf("AT+UCAST:0021ED00000355B6,CUIP_ONLINE\n\r");
}

// Main runtime loop
// looking for the interrupts and meanwhile doing some diagnostic casts
// interrupts : bcast_flag, end_flag, inputflag, joinflag, possible_bcast,
// start_flag
while(TRUE) {
    /*
    if(!bcast_flag && !end_flag && !inputflag && !joinflag) {
        output_high(LED1);
        // Diag indication - heart beat
        delay_ms(10);
        output_low(LED1);
        delay_ms(10);
    }
*/
// measure temperature
output_high(TempPower);
Temperature = measure(Adc_Temperature,45);
// use 45 as a means to average and scale to ~degrees *100
Temperature = Temperature/10;
// scale to temperature in ~degrees*10 (LM34 in F, LM35 in C)
output_low(TempPower);

// take away power

printf("at+bcast:01,SCV,Temp,#Readings= %05lu %05u
\r", Temperature, TempDataCount);
}

// broadcast, see what it is and set other flags accordingly
// ### Change this here to reflect new communication language
if (bcast_flag) {
  //
  for(k = 0; k < sizeof(bcast); k++) printf("%c",bcast[k]);
  //
  printf(\n"
\r
interpret_comm();
bcast_flag = 0;
bcast[0]=0;
//
printf("bcast_flag trig\n\r");
}

// stop data collection
if (end_flag) {
  //
  disable_interrupts(INT_TIMER2);
data_flag = 0;
  end_flag = 0;
  writep_ext_eeprom(ptr, packet0, packet_index);
  packet_index = 0;
data_ready_flag = 1;
//
printf("--ENDED-- times run: %u, data collected: %lu\n\r", int_count, data_collected);
  printf("AT+UCAST:0021ED00000355B6,DAQ DATA STOPPED\n\r");
}

// start data collection
if (start_flag) {
  int_count++;
data_collected = 0;
  start_flag = 0;
packet_num = 0;
packet_index = 0;
ucast_count = 0;
//
printf("start (int_count)= %D \n\r", int_count);
printf("AT+UCAST:0021ED00000355B6,DAQ STARTING DATA COLLECT\n\r");
set_timer2(0);
  enable_interrupts(INT_TIMER2);
}

// collect a data point, set end_flag if all data is collected
// ### Set the data rate here
if (data_flag) {
  if (data_collected <= max_data)
  {
    if (packet_index < sizeof(packet0))
    {
for(k = 0; k <= (active_channels-2); k++)
    //pressure sensor read seperately from those through mux
    {
        set_mux_chan(k);
        delay_us(300);
        t1 = get_adc_data(ADC_CONV1);
        //passing pin info into get_adc_data now
        d_buff[k] = t1;
        if (t1 < 257) t1 = 257;
        //printf("t1: %lu, data_collected: %lu, 
        packet_index: %u...\n\r",t1, data_collected + k, packet_index);
        packet0[packet_index + 2*k] = make8(t1,1);
        //MSByte of data
        packet0[packet_index + 1 + 2*k] = make8(t1,0);
        //LSByte of data
        //delay_us(100);
    }
    //Collect Pressure Transducer data
    t2 = get_adc_data(ADC_CONV2);
    d_buff[preschan-1] = t2;
    if (t2 < 257) t2 = 257;
    packet0[packet_index + 2*preschan] = make8(t2,1);
    packet0[packet_index + 2*preschan+1] = make8(t2,0);
    //Collect CJC data
    if(reduced_data_flag) t3 = measure(2,3);
    //channel 1 corresponds to AN1 on the ADC
    make8(t3,1);
    packet0[packet_index + 2*active_channels] =
    make8(t3,0);
    data_collected = data_collected + active_channels + 1;
    // +1 for the addition of the cold junction
    temperature channel
    packet_index = packet_index + (active_channels+1)*2;
    // the 2 multiplier is due to 16 bit or 2 bytes of data
}
for(j=0;j<sizeof(packet0);j++) printf("%u\n\r",packet0[j]);
if (packet_index + active_channels*2 > sizeof(packet0))
{
    ptr = packet_num*packet_index;
    writep_ext_eeprom(ptr, packet0, sizeof(packet0));
    packet_num++;
    packet_index = 0;
}
else end_flag = 1;
data_flag = 0;
}

// ### Unicast every half second maybe?
if (ucast_flag) {
    printf("AT+UCAST:0021ED00000355B6,DATCAST %lu %lu %lu %lu %lu\n\r", ucast_count,
d_buff[0], d_buff[1], d_buff[2], d_buff[3], d_buff[4], d_buff[5];
    ucast_count++;
    ucast_flag = 0;
}
//send flag handled in iscommand

//test code implemented here
//

//=====================================================================
}  // main while loop
}  // main program end

//=====================================================================
// INTERRUPT HANDLERS //
//=====================================================================
// note - where it is this necessary to wake from sleep (possibly, the power
measurement module), add jumper from RC7 to RB0, RB4, RB5, RB6, OR RB7
// RS232 input interrupt - will not wake from sleep
#define INT_RDA

void RDA_isr() {
    int n, m;

    // get character
    keybdbyte = getc();
    inputflag = 1;

    //looking for JPAN:<channel>,<PID>,<EPID>
    if((keybdbyte == 'P') && shift_char_array[sizeof(shift_char_array)-1] == 'J')
        joinflag = 1;

    //looking for ERROR:28
    else if((keybdbyte == '8') && shift_char_array[sizeof(shift_char_array)-1] == '2' &&
        shift_char_array[sizeof(shift_char_array)-2] == ':') joinflag = 1;

    //looking for BCAST:[<EUI64>,]<length>=<data>
    else if (keybdbyte == 'B') {
        possible_bcast = 1;
        m = 34;
    }

    else if (waitingforopen == 1) {
        if (keybdbyte == ':') openflag = 1;
    }

    if (possible_bcast) {
        if (bcast[0] != '=') {
            for(n = 0; n < (sizeof(bcast)-1); n++) {
                bcast[n] = bcast[n+1];
            }
            bcast[sizeof(bcast)-1] = keybdbyte;
            m--;
            if (m == 0) possible_bcast = 0;
        }
    }
else {
    bcast_flag = 1;
    possible_bcast = 0;
}
}

if (!possible_bcast) {
    for(n = 0; n < (sizeof(shift_char_array)-1); n++) {
        shift_char_array[n] = shift_char_array[n+1];
    }
    shift_char_array[sizeof(shift_char_array)-1] = keybdbyte;
}
} // end RDA interrupt

#INT_TIMER2
    // Paces ADC conversion
void clock2_isr() {
    // timer 2 frequency = 60Hz
    if (y <= ymax) y++;
    else
    {
        data_flag = 1;
        y = 0;
    }
    if (x <= 60) x++;
    else
    {
        ucast_flag = 1;
        x = 0;
    }
    if( z <= zmax) z++;
    else
    {
        reduced_data_flag = 1;
        z = 0;
    }
}

void mcu_setup(){
    // MCU SETUP!
    // setup IO, etc.
    int k;
    keybdbyte = 0;
    // OSCCON = 0x71;
    delay_ms(100);
setup_timer_2(T2_DIV_BY_16, 250, 8); // with 8MHz clock, timer 2 interrupts at 62.5 Hz

packet0[63]= 0xff; // EEPROM data packet trailer for error detection

set_tris_a(0b00000110); // setup i/o mode for reg A
set_tris_b(0b00000011); // setup i/o mode for register B, b0=in, b1=out, b2-5=out b6/b7 don't care
set_tris_c(0b01001000);

bit_clear(Powerup); // flag power-up (question: does this clear the fact that we're powering up? or does it show that we're powering up for the first time?)

setup_adc(ADC_CLOCK_INTERNAL);
setup_adc ports(sAN1);

init_ext_eeprom();
// alpha = ext_eeprom_ready();

// be assured that the convert bit for the adc is high
output_high(ADC_CONV1);
output_low(ADC_SCK);
output_high(ADC_CONV2);

// indicate power-up complete
for(k=0;k<=2;k++) {
    output_high(LED2);
    output_high(LED1);
    delay_ms(100);
    output_low(LED2);
    output_low(LED1);
    delay_ms(100);
}

printf("Setup complete.\n\r");
// diagnostic print startup, report start and firmware date

char IsCommand(char *bc, char *ta, int len) {
    // checks if the array is a passed command
    int a;
    char flag = 1;

    // test broadcast to see if it is the command
    for(a = 1; a < (len); a++) {
        if (bc[a] != ta[a-1]) {
            flag = 0;
            break;
        }
    }

    //printf("got here... T/F:\c\n\r", flag);
    return flag;
}
void interpret_comm() {
    int minute_change;
    // interpret the command
    if (IsCommand(bcast, START_array, sizeof(START_array))) {
        // printf("sizeof START_array: %u\n", sizeof(START_array));
        start_flag = 1;
        printf("start bcast received\n");
    } else if (IsCommand(bcast, END_array, sizeof(END_array))) {
        // set end flag to stop readings
        end_flag = 1;
        printf("end bcast received\n");
    } else if (IsCommand(bcast, DATA_array, sizeof(DATA_array))) {
        // send data
        printf("data bcast received\n");
        printf("AT+UCAST:0021ED00000355B6,Frequency = %u, # Points = %lu\n", ymax, data_collected);
        printf("AT+UCAST:0021ED00000355B6,DATA_COMING!\n");
        ZigbeeSend();
        // send data
        output_high(LED2);
        output_high(LED1);
        delay_ms(200);
        output_low(LED2);
        output_low(LED1);
        delay_ms(200);
        delay_ms(3000);
        // allow time to complete transmission
        printf("AT+UCAST:0021ED00000355B6,COMPLETE\n");
    } else if (IsCommand(bcast, MIN_array, sizeof(MIN_array))) {
        printf("minutes change bcast received\n");
        minute_change = bcast[5]*10 + bcast[6];
        if (minute_change < 17) max_data = minute_change*60*data_freq*active_channels;
        else printf("AT+UCAST:0021ED00000355B6,SAYTHEMAGICWORD\n");
    }
    else bcast_flag = 0;
}

// ### Allow the adc_conv bit to be changed
int16 get_adc_data(chanConv) {
    int16 temp_val, tmp1;
    int c;

    // printf("Getting Data\n");
    temp_val = 0b0000000000000000;

    // output_high(ADC_CONV2);
    // set MUX enable bit to push through mux channel
    // delay_us(500);
// set conversion bit low to start data collection
output_low(chanConv);

// cycle through four clock cycles to get rid of the first 4 zeros
for (c = 0; c <= 3; c++) {
    output_high(ADC_SCK);
    output_low(ADC_SCK);
}

//### look at breaking this out... speed up the process significantly
// go through 16 bits of data
for(c = 0; c <= 15; c++){
    output_high(ADC_SCK);
    if (input(ADC_DIN)) tmp1 = 0b0000000000000001;  // 
    else tmp1 = 0b0000000000000000;  // 
    temp_val = (temp_val|(tmp1<<(15-c)));  // instruction time
    decreases with more shifts
    // nine clock cycles before here
    output_low(ADC_SCK);
}

output_high(chanConv);
    // shut down convert bit, 20 bits should have been read: 4
    zeroes and 16 bits of data.
    // output_low(ADC_CONV2);  // disable MUX enable
    // delay_us(500);
    // printf("data = %lu\n", temp_val);
        return temp_val;
} // end get adc data

void clear_eeprom() {
    // zeroes out eeprom
    // to be implemented
}

int16 measure(int channel, int average){
    // MEASURE (DONE)
    // get stable ADC value with some averaging
    int ilocal;
    int16 value;
    set_adc_channel(channel);
        // ADC channel
    value = 0;
    for (ilocal = 1; ilocal <= average; ilocal++){
        // average
        value = value + read_adc();
    }
    return (value/average);
}

void set_mux_chan(int m_chan){
    // SET MUX CHAN
    // sets the mux channel (DONE)
int MUX_CH;
MUX_CH = m_chan;

switch(MUX_CH) {
    case 0:
        //
        printf("Data Channel 0\n\r");
        output_high(MUX_A2);
        output_high(MUX_A1);
        output_high(MUX_A0);
        break;
    case 1:
        //
        printf("Data Channel 1\n\r");
        output_high(MUX_A2);
        output_high(MUX_A1);
        output_low(MUX_A0);
        break;
    case 2:
        //
        printf("Data Channel 2\n\r");
        output_high(MUX_A2);
        output_low(MUX_A1);
        output_high(MUX_A0);
        break;
    case 3:
        //
        printf("Data Channel 3\n\r");
        output_high(MUX_A2);
        output_low(MUX_A1);
        output_high(MUX_A0);
        break;
    case 4:
        //
        printf("Data Channel 4\n\r");
        output_low(MUX_A2);
        output_high(MUX_A1);
        output_high(MUX_A0);
        break;
    case 5:
        //
        printf("Data Channel 5\n\r");
        output_low(MUX_A2);
        output_high(MUX_A1);
        output_high(MUX_A0);
        break;
    case 6:
        //
        printf("Data Channel 6\n\r");
        output_low(MUX_A2);
        output_low(MUX_A1);
        output_high(MUX_A0);
        break;
    case 7:
        //
        printf("Data Channel 7\n\r");
        output_low(MUX_A2);
        output_low(MUX_A1);
        output_low(MUX_A0);
        break;
    default:
        printf("Data Channel xx\n\r");
        break;
} // end switch statement
} // end set_mux_chan()
Appendix B

Selected LabVIEW Code

Data Parser
Data Converter

- **K-type TC**
  - Conversion formula: \( \text{raw data} \times 3.33/2^{16}/70.709 \)
  - Output: \( 7.112247899864E-7 \)

- **R-type TC**
  - Conversion formula: \( \text{raw data} \times 3.33/2^{16}/70.709 \)
  - Output: \( 7.1852174523706E-7 \)

- **Pressure Transducer**
  - Conversion formula: \( \text{raw data} \times 3.33/2^{16}/31114/488.53 \)
  - Output: \( 6.40139773119 \)

- **Gardon Gauge**
  - Conversion formula: \( \text{raw data} \times 3.33/2^{16}/70.709 \)
  - Output: \( 0.0007112247899864 \times 50.51 \)

- **CJC Voltage Transform**
  - Conversion formula: \( (\text{raw data} \times 3.33/2^{16} - 250) \times 1000 = v \text{ in mv} \)
  - Output: \( 0.00322235625 \)
Interpret Communication

Event Handler
Appendix C

WLSS Hardware Circuit Diagrams and PCB Layouts
Figure 9-1: Power subsystem schematic.
Figure 9-2: Power subsystem layout.
Figure 9-3: Signal-conditioning subsystem schematic.
Figure 9-4: Signal-conditioning sub circuit PCB.
Figure 9-5: Control subsystem schematic (MCU).
Figure 9-6: Control subsystem schematic (I/O).
Figure 9-7: Control subsystem PCB.
References


*All brand or product names mentioned in this thesis are trademarks or registered trademarks of their respective holders.*