Design of a
Wireless Sensor Node Platform

Michael May

This report is in partial fulfilment of the requirements for the degree of Bachelor of Computing and Mathematical Sciences with Honours (BCMS(Hons)) at The University of Waikato.

©2012 Michael May
Abstract

A wireless sensor network is a distributed network made up of low-power consuming, low cost devices called sensor nodes which combine sensing, computation and communication. Sensor networks have enabled many applications, including remote tracking and monitoring of environments in real-time. The sensor nodes in possession at The University of Waikato have been used in several theses and projects but the time has come to upgrade the node as newer technologies allow for a more powerful, feature packed sensor node.

A review of available sensor nodes discovered that suitable nodes could not be obtained in appropriate volumes. As a consequence the decision to design and manufacture a new node was taken.

This report describes the design and construction of a sensor node platform intended for use in sensor networks research. The system is made up of a hardware platform and a software stack for drivers and interfacing.
Acknowledgements

I wish to acknowledge the input and advice that Tony McGregor has had into the design of this platform. His ideas and availability as an audience to thoughts and problems has been invaluable. I would like to thank Dean Armstrong for his help, especially in the hardware design and manufacture stages, and for his expertise in radio communications.
# Contents

1 Introduction ............................................. 5  
1.1 Organisation of This Report .............................. 7  

2 Background .............................................. 9  
2.1 Sensor Node Characteristics ............................. 9  
  2.1.1 Microcontroller .................................... 10  
  2.1.2 Radio Transceiver .................................. 12  
  2.1.3 Sensors ............................................. 13  
  2.1.4 Power Source ....................................... 14  
2.2 ScatterWeb ............................................. 15  
2.3 Goals ................................................. 16  
2.4 Other Sensor Nodes .................................... 17  
2.5 New Sensor Board ...................................... 18  

3 Design Process .......................................... 20  
3.1 Component Selection .................................... 21  
  3.1.1 Microcontroller .................................... 21  
  3.1.2 Radio Transceiver .................................. 23  
  3.1.3 Sensors ............................................. 25  
  3.1.4 Power Source ....................................... 27  
3.2 Prototyping .............................................. 28  
3.3 Software ............................................... 29  
  3.3.1 Hardware Drivers ................................... 29  
  3.3.2 Device Interfacing .................................. 31  

4 Implementation .......................................... 33  
4.1 Circuit board design .................................... 33  
  4.1.1 Schematics ......................................... 34
4.1.2 Board layout .................................................. 34
4.1.3 Testing and Verification ................................. 35
4.2 Revision 2 ..................................................... 38
4.3 Performance ................................................. 41
   4.3.1 Battery ............................................... 41
   4.3.2 Communication Range ............................... 42

5 Conclusion ...................................................... 45
   5.1 Future Work .............................................. 46

A First Appendix ................................................. 51
   A.1 WOSPr1 .................................................... 51
   A.2 WOSPr2 .................................................... 60
# List of Figures

1.1 Design process .............................................. 7
2.1 WSN node hardware ........................................ 10
2.2 ScatterWeb node [23] ...................................... 16
3.1 Node hardware interfacing ................................. 29
3.2 Breadboard prototyping ................................... 30
4.1 Annotated WOSPr1 render ................................. 36
4.2 WOSPr2 (left) compared to WOSPr1 (right) ........... 40
A.1 WOSPr1 schematics sheet 1 ............................... 52
A.2 WOSPr1 schematics sheet 2 ............................... 53
A.3 WOSPr1 schematics sheet 3 ............................... 54
A.4 WOSPr1 schematics sheet 4 ............................... 55
A.5 WOSPr1 board top layer ................................ 56
A.6 WOSPr1 board second layer - Gnd ..................... 57
A.7 WOSPr1 board third layer - Vcc ......................... 58
A.8 WOSPr1 board bottom layer ............................ 59
A.9 WOSPr2 schematics sheet 1 ............................... 61
A.10 WOSPr2 schematics sheet 2 ............................ 62
A.11 WOSPr2 schematics sheet 3 ............................ 63
A.12 WOSPr2 schematics sheet 4 ............................ 64
A.13 WOSPr2 board top layer ................................. 65
A.14 WOSPr2 board second layer - Gnd .................... 66
A.15 WOSPr2 board third layer - Vcc ....................... 67
A.16 WOSPr2 board bottom layer ......................... 68
List of Tables

3.1 Microcontrollers investigated ........................................ 24
3.2 Radio transceivers investigated ....................................... 26

4.1 Power consumption .......................................................... 41
4.2 Node lifetimes ................................................................. 43
4.3 Communication Ranges ...................................................... 43
Chapter 1

Introduction

Wireless Sensor Networks (WSNs) have fast become an established field in research, industrial and consumer applications, to interact, monitor and relay physical and environmental details. A wireless sensor network is made up of ultra low-power consuming, low cost, distributed devices called sensor nodes that combine sensing, computation and communication. These nodes form part of a larger network, enabling a wealth of information that can be obtained in an instant across many spatially distributed devices covering an area. While a single node’s capabilities may be limited, a larger collection can offer new technological and information processing possibilities.

A sensor node consists of a microcontroller that performs organisation and administration. A radio transceiver is used to send and receive data packets whose contents can vary from sensor information, network topology updates, to over-the-air firmware updates. Integrated sensors provide the data that is transmitted and this can come from a range of sources such as a temperature sensor, microphone, a low powered camera or combinations of them. A power unit supplies energy for operation and the source of this energy can vary greatly from node architecture to application, but is commonly some form of battery. A larger factor in sensor networks is the drive to reduce the cost of each node. Cheaper nodes allow for denser deployment and more information acquisition. Therefore nodes are designed around cheap microcontrollers with limited resources such as a low clock speed of 20MHz and only a few kilobytes of memory.

Scenarios for use of these device range from monitoring environmental characteristics, real-time object tracking, or monitoring the health of buildings and structures. A typical application of a wireless sensor network is to monitor remote environments
via low frequency data collection. This could be a simple temperature measurement of a field or a more complex deployment in a manufacturing plant where gas leaks are monitored by hundreds of nodes spanning an entire factory.

Harsh resource constraints present a design challenge as complex networking protocols must be implemented while utilising less than a few kilobytes of RAM and a couple of tens of kilobytes of program storage. To counter each node’s low resources, the nodes rely on one another and together they scale to form a powerful mesh network. This network is capable of absorbing hardware and radio link failures through multipathing to provide reliable paths for information flow.

Significant research time and money is currently being put into wireless sensor networks as the field encompasses many areas of computer science, electrical engineering, materials and solid-state fabrication. Battery life is a big concern in sensor nodes. Extensive research has been undertaken to develop and improve ad-hoc routing, and efficient and distributed signal processing which helps minimise power consumption to improve a node’s lifetime from single battery charge.

Due to the inexpensive and remote nature of these devices, battery replacement or charging is a costly and sometimes unfeasible exercise. Therefore optimising a sensor node’s life is a requirement that is achieved through the use of ultra low-powered components, efficient programming and a doctrine that a node will spend most of its life in a low power sleep state. Operating requirements can change based on the application. For example, a temperature reading may only need to be taken every 30 seconds, but video or audio sampling may need to take place multiple times a second and this can be a strain on battery life. Therefore employment of data filtering on the node can make sure only the interesting pieces of data are transmitted, or data can be stockpiled to reduce sending incompletely filled packets and reduce the time spent in a transmitting state.

Significant development has been put into designing robust and effective sensor network routing protocols, and none more so than the work of the ZigBee Alliance [30] and the IEEE 802.15.3 standard [12]. This suite allows reliable ad-hoc ZigBee mesh networks to be formed with no central control while moving a lot of the protocol integration away from the firmware developer. ZigBee is specifically targeted at wireless sensor networks, home and building automation and control, and industrial control. Hardware based on the specification runs at the mid to upper end of the microwave band and generally the hardware has a low power output and low receive sensitivity.

There have recently been many successful open source hardware products with
thriving communities on the cutting edge of crowd sourced research. An open source sensor node platform would enable community involvement, development and refinement of more advanced sensor nodes, while increasing the collective pool of knowledge.

The objective of this project is to develop an extensible sensor node platform, the goals of which are provided in section 2.3. The process flow is outlined in figure 1.1, this shows the steps of design once the idea and requirements are specified. As we move through the steps there becomes less freedom and the directions become fewer. Component selection is a broad process as it requires choices from a multitude of products available and it dictates how the rest of the steps are undertaken. Prototyping and software development is more constrained and is about developing a prototype working sensor node and the related software. Once the prototype design has been made it is implemented in a schematic diagram and then as a printed circuit board. After this stage a completed circuit has been made and the last thing to do is testing and verification.

1.1 Organisation of This Report

This report is organised into five chapters. Chapter 2 presents the key constraints to which a wireless sensor node must adhere, along with a review of commercially available sensor nodes. The intention is to provide a background from which our
sensor node platform will be built. Chapter 3 discusses the component selection process, prototyping stages and testing software. This chapter covers the design process, up to having a complete working node that is built on a breadboard using development boards. In chapter 4, the physical implementation stages, performance comparisons, and a second generation sensor node is presented. The purpose is to take the design of the working prototype node and implement it in a custom printed circuit board to provide a node ready for use. Chapter 5 provides insight into the next stages of the platform and summarises the report.
Chapter 2

Background

A wireless sensor network node is a small wireless device that encompasses a microcontroller, a radio transceiver, an antenna, a power source, and one or more sensors. A lightweight operating system enables a node to function and provides features such as sensor polling, data manipulation, wireless communication, and remote access. The University of Waikato Computer Science department, currently has in its possession several sensor nodes from the Freie Universitt Berlin, Germany, called the Modular Sensor Board 430 as part of their ScatterWeb project [23]. These have been used in several projects and theses throughout the years, most recently Distributed Operating Systems on Wireless Sensor Networks by Waikato PhD candidate Paul Hunkin [11].

This chapter will cover some of the design choices required when designing a sensor node, review the existing sensor nodes along with various other commercial and research nodes available, and present an argument for the construction of a new sensor platform.

2.1 Sensor Node Characteristics

A microcontroller acts as the central controlling device, while the transceiver enables packets of data to be sent and transmitted. Sensors are used for data collection and in most cases provide purpose to a node. Figure 2.1 shows a functional diagram of the typical hardware that makes up a node. In such a small system, optimising every component based on its requirements is critical as it can save development time, board space, and money. This low cost aspect allows for many nodes to be deployed, providing denser coverage and multiple data paths to cope
In this section we consider each of the main components, outlining the ideal characteristics that make a sensor node.

### 2.1.1 Microcontroller

A microcontroller oversees operation of a sensor node and is one of the most important components. Almost every microcontroller family available is different; some may be designed for speed, others for power efficiency. It is a matter of choosing the correct microcontroller for the application and in a sensor node there are various aspects that need to be considered. The following will detail these points.

#### Power Consumption

A microcontroller’s architecture can be optimised for speed or power consumption. In a WSN, which must run off a battery for a great length of time, a microcontroller optimised for low power is important.

In order to minimise power consumption, a node should spend most of its life in a sleep mode, therefore a low current draw when asleep is essential [17]. This varies between microcontrollers, architectures and the various states of sleep but tends to range from less than $1\mu\text{A}$ to $100\mu\text{A}$. Active mode power consumption is a useful metric too, and becomes more important the longer a node spends in wake periods. This tends to range between $1\text{mA}$ to $10\text{mA}$ for small, low-powered microcontrollers,
but varies depending on attributes such as the built-in peripherals, clock speed, size of the memory and location of program execution.

**Memory**

Flash memory and RAM quantities should be chosen to suit the application as both can be expensive in terms of cost and size. In general, more RAM equals more wafer real estate and more power required to keep this volatile memory refreshed. This leads to higher costs through less manufacturing yield, and the larger wafer may require a larger package. However, newer microcontrollers are often made using newer semiconductor technology. A more recent microcontroller may incorporate several hundred kilobytes of flash memory and tens of kilobytes of RAM while continuing to use standardised packages and costing less. One particular new technology coming through is the feasible use of FRAM [27], a memory alternative that combines the speed of conventional RAM with the non-volatility of flash storage and eliminating the segregation between the two. This type of flash has lower power consumption when operating compared to standard flash and eliminates the power consumption of having to refresh RAM to preserve its contents, reducing power consumption in sleep states. FRAM was not used in this project as current processes cannot manufacture large enough quantities to meet the goals of section 2.3.

**Peripheral Support**

A microcontroller consists of pins that serve varying functions such as serial communications, timer outputs, comparisons, analogue and digital inputs and outputs; even USB connections. There is a desire to minimise the pin count as a smaller package can lead to lower costs and a smaller footprint. A problem is that microcontrollers tend to integrate more functions than the quantity of pins allow. This leads to pin multiplexing where each pin can have a variety of functions that are selectable via software. When selecting a microcontroller, knowing what types of pins are going to be used can help to ensure the microcontroller will serve all the required inputs and outputs without the need for an external I/O controller to provide the additional interfaces. For a sensor node many different types of pins are required: typically a radio is interfaced over a high speed SPI channel; sensors can require a variety of interfaces such as I²C, I²S, analogue-to-digital converters (ADCs), and interrupt enabled GPIO. Additional pins are required for clocking systems, programming, and outputs to interfaces such as LEDs.
\( I^2C \) \cite{18} is a half-duplex serial bus that can operate from 400KHz and each device on the bus has a unique 7 bit address assigned by its manufacturer. SPI is a higher speed full-duplex serial bus that can clock at over 10MHz and each device on the bus requires a separate select pin from the microcontroller to alert it when it is being talked to. The benefit of these protocols is that they allow simple, low pin count, communication between devices. \( I^2C \) has an advantage over SPI that it only requires two pins, clock and data, to do addressing and data transfers, whereas SPI requires three common pins, clock, data in and data out, along with an extra pin to each device connected to the bus as a device select. SPI has a much simpler driver as a lot of the functions are completed through its range of pins. In contrast, \( I^2C \) sends start, stop and addressing on the one data line and therefore takes more cycles to complete an operation.

### 2.1.2 Radio Transceiver

The transceiver allows two-way radio communication between nodes in order to distribute information, e.g., routing sensor data to a base station. The choice of transceiver has a big impact on power consumption as the transceiver, when it is idle, sending or receiving data, will generally consume more power than any other component.

A few important characteristics of a transceiver define how well it will perform in a sensor network application.

#### Frequency

Most sensor nodes make use of ISM regions in the 300 to 3000MHz Ultra High Frequency band, in particular the license free regions around 430MHz, 900MHz and 2.4GHz. In New Zealand this is referred to as the Short Range Devices bands \cite{21}. Areas within these bands are free to use so long as rules are followed, in particular staying within specified centre and deviation frequencies, and output power. Using ISM regions allows the use of commonly available radio transceivers, and the radios can provide long ranges with high bit rates.

#### Data Rate

In most cases wireless sensor network applications need very little data throughput as they tend to only transmit periodically and in low bit quantities \cite{10}. For example, they generally aren’t used in applications for streaming audio or video.
Low data rates help offer greater range but at the expense of longer periods spent in a high current consuming, transmitting state. Therefore it is beneficial to determine the data rate needed for the intended application when choosing a transceiver. Fortunately almost all ISM transceivers have a wide programmable speed from 500kbps down to near zero kbps. This project will make use of a variable data rate. As that needed in research may be different to the data rate of a practical application and a variable data rate will allow range and power saving customisation.

**Tx\Rx Power**

The maximum transmission power of a radio transceiver is an important factor as it determines the signal strength of a transmitted wave. The stronger the signal then the further it will travel. All spectrum allocations come with legal limits on power output when transmitting to limit exposure and biological effects on absorbing matter. Finding a transceiver that can closely match this limit is good because this platform will be used in multiple roles and having a transceiver that can be very powerful in one application while weak in another, all via software, will help extend its battery life and usefulness [3].

Although transmitter power mainly determines the transmission distance, other factors such as receiver sensitivity, antenna gain and efficiency, and the modulation scheme all have an impact on range. In section 4.3.2, range tests using various antennas and data rate are shown in order to demonstrate the effectiveness of using the correct antenna.

**Power Consumption**

The radio transceiver will most likely be the largest power consumer in the circuit as it tends to draw tens of milliamps in both transmit and receive states. This means choosing a transceiver with a small current consumption for a given output power, compared to other transceivers, will help extend the battery life available. Transmission power is also a good way of reducing the current consumption and lessening this is a sure way to extend a battery’s life.

**2.1.3 Sensors**

There is a vast range of sensors available that can be interfaced to with a microcontroller. When choosing the right sensors for a node consideration must be given to the way that it is interfaced; its characteristics compared to similar sensors, such
as accuracy; and its power consumption. If a sensor board is used in a variety of purposes then it makes sense to design modularity into it and allow customisation.

A recurring theme is minimising the power consumption of a component in order to maximise the battery life. Sensors should be chosen in such a way that power consumption is minimised. Minimal sleep and sensing currents, low turn on and off times, the time it takes for a sensor to be ready to accurately sense, and the time it takes to generate a sensor reading, can all affect power consumption in varying manners.

A sensor’s input and output interface can limit accuracy, features, and the number of pins used on the microcontroller. Reading accuracy can be diminished if, for example, an analogue sensor is outputting into a 10-bit ADC on a microcontroller but if an equivalent 12-bit digital sensor is available, then the digital sensor will provide more accurate readings. Digital sensors tend to use serial interfaces such as SPI and I²C, which can enable access to internal registers for enabling features or setting threshold values.

2.1.4 Power Source

Batteries provide a low cost, easily available and high capacity source of power, and have become synonymous with mobile devices. And this is no different in sensor networks. This area, with its ultra low power consumption and duty cycle, presents an opportunity for the application of advanced small-scale power generation, such as solar generation or parasitic energy harvesting. Analysis and application of these possibilities is out of the scope of this project, so conventional chemical energy is pursued due to its easy and constant high power.

A regulator [2] is a system that takes a varying input voltage and outputs a constant predetermined voltage. Regulators are useful due to most components having a small and specific working voltage range but power sources can vary in voltage.

A linear regulator is an inefficient way of step-down regulation as it draws a constant supply of current, part of which is converted into heat. The inefficiency is prevalent when the current draw of the rest of circuit is minimal, such as when the sensor node is asleep.

A switching-mode power supply operates as a very fast switch to regulate the output voltage, providing much improved efficiencies over a linear regulator. It also has the added benefit of being able to both step-down and step-up a voltage, providing a larger voltage window. Even though a switching-mode power supply is
highly efficient, almost up to 100% under ideal conditions, this is only obtainable at a rated current draw [14] [28]. This means that when fully operating a sensor node might draw 30mA and the regulator is 98% efficient, but as soon as the node goes to sleep and is drawing less than 100µA then the regulator might become only 20% efficient.

**Voltage Range**

Due to inefficiencies of voltage regulation and varying supply voltages all components need to have a wide voltage window that allows them to continue operating as the power source voltage drops over time and use. This means that the wider the voltage window, the longer a component can stay operational.

### 2.2 ScatterWeb

The ScatterWeb sensor boards [23], as shown in figure 2.2 have been used by the University of Waikato in several theses and projects based around physical world interaction, and network and operating systems research.

ScatterWeb is built around a Texas Instrument (TI) MSP430F1612 microcontroller with 5kB of RAM and 55kB of flash storage, and operates up to 8MHz. The radio transceiver is a TI CC1020 designed for 868MHz ISM operation, with an adjustable output power of up to +5dBm and a receiver sensitivity of up to -116dBm. The boards are powered by three AAA batteries in series providing up to 4.5V which is fed through a regulator to supply a stable 3V. This provides a long voltage drop window, but due to the chemistry and size, AAA batteries have a limited current capacity and only a small lifespan of the order of a hundred days at small sensing intervals.

Additional features include an integrated analogue 3-axis accelerometer, a microphone and passive infrared motion detector, as well as a SD card interface that can be used for data logging and temporary storage. These have now become outdated and newer technologies offer more resources and functionality at the same size and at lower costs.

Paul Hunkin’s distributed operating system, named Hydra, [11] was designed and tested using several of these nodes. Due to the small resources, limited capacities and only having a small number of nodes to distribute across, the performance when running the operating system was impacted. Additional nodes were expensive, $150, and would not solve the problem of small resources. The decision for acquiring new
nodes stemmed from the need for more capabilities and resources. A higher powered node would also mean fewer nodes were required to maintain or improve upon the performance of some typical applications. A cheaper node would mean more could be purchased for less as well.

For these reasons a new sensor node was called for and some constraints were defined.

2.3 Goals

Due to the shortcomings of the ScatterWeb nodes, the search for a new node that met several requirements began. These requirements are defined in order of importance as follows:

- RAM and flash memory greater than 8 kB and 64 kB respectively.
- Less than $100 per node in small volumes.
- Radio range of at least 200 metres at low data rates.
- Battery life exceeding the calculated life of the ScatterWeb nodes.
- A USB interface for easy programming and data collection.
A compact form factor.

As the quantity of RAM and storage was the biggest limiting factor of the ScatterWeb nodes an increase in these was essential. Ideally the node should have a long communication range but also be cheap, power efficient and operate within the legal limits of spectrum and power. Considering these requirements a goal of at least 200 metres at low data rates was desired. Due to the use of newer technologies, battery life should exceed that of the old node. USB interfacing would be nice to have as it reduces the part count necessary for interfacing with a computer and may allow programming without a specialised programmer device.

The node also had to provide allowance for multiple use cases and could not be designed solely for commercial use instead of research and development, and vice versa.

2.4 Other Sensor Nodes

A review of other sensor nodes was completed in order to determine if a suitable replacement existed that was targeted at research institutes while meeting the primary goals describe in section 2.3. Radio range was hard to judge as it was more of a practical characteristic, but link budgets and antennas should be taken into account. An external interface would also be required as this would allow the addition of extra components, mainly sensors, which was essential for both research and practical customisation. There are many readily available sensor nodes on the market but most are based around microcontrollers that do not meet our memory limits. The radio transceivers on new nodes seem to be targeted towards the ZigBee specification as this allows a simpler development cycle. An example is the Mica node series. Newer versions of this series are released every so often but none so far have increased the quantity of RAM and instead only modify the transceiver.

The following sections discuss some commercially available sensor nodes that meet the memory requirements. Each node has an argument presenting why it was not chosen and how it did not meet the other deciding goals.

Redwire Econotag

The Redwire Econotag [22] is a development board based on a Freescale MC13224V ARM7 microcontroller with a built-in IEEE 802.15.4 2.4GHz wireless transceiver but is not ZigBee compliant. It features 128kB of flash storage and 96kB of RAM. No
sensors are included on the board but 36 GPIO pins are brought to headers for expansion. This board is designed as a development board and not suited for practical deployment because there is little allowance for practical work.

**Virtenio Preon32**

The Virtenio Preon32 [29] sensor node is designed around an unnamed ARM Cortex-M3 microcontroller with 256kB of flash and 64kB of RAM. The radio is an unknown IEEE 802.15.4 2.4GHz transceiver that supports up to 250m wireless range when outdoors. It is designed as module to be soldered onto another PCB which provides the interfaces and power. This approach means that the cost to buy the module as well as designing and manufacturing a separate board would have been much more than the cost of producing a single board for a similar amount of work.

**Arago Systems WiSMote**

The Arago Systems WiSMote [1] is a development platform designed on a TI MSP430 microcontroller and has 16kB of RAM and 256kB of flash. The transceiver is a TI CC2520 2.4GHz radio that can communicate over 100m. The board is designed for research and development. As a consequence, it is less suitable for practical deployment. In particular the communication range is less than ideal.

**Libelium Waspmote**

The Libelium Waspmote [13] is a feature-packed sensor board designed around an Atmel ATmega1281 microcontroller with 8kB RAM and 128kB flash storage. It uses a modular approach to every peripheral and is designed to use XBee radio modules of various frequencies that support ZigBee. A GPRS socket, GPS socket and expansion socket provide the use of add-on circuit boards manufactured by Libelium. This board is unsuitable for our use due to its very high price, over $200, and the unnecessary modularity.

### 2.5 New Sensor Board

The search for a commercially available sensor node that was inexpensive, designed for both research and practical deployment, and met at least a few of the goals outlined in section 2.3 proved unsuccessful. It was decided to pursue an in-house developed board that was built to achieve all of the listed goals while serving as a
development platform and could handle field work. Another motivating factor was the reduced cost of designing to our specifications and having the ability to redesign the core board to suit an application. This was also attractive to the department as it would refresh our capacity to do a hardware project as one has not been done for some time.

As the new nodes were now going to be designed in-house, the sensors included on the board could be tailored to our needs. The nodes were to include a range of sensors so they could serve different purposes, provide flexibility, and allow for experimentation. The sensors desired to be included, in order of importance, were:

- A temperature sensor.
- A 3-axis accelerometer.
- Two pushbuttons and LEDs.
- A pin header to allow extra devices and boards to be connected.
- A passive infrared motion detector.

This array of sensors allows the nodes to serve in a variety of circumstances, both research and real-world applications, while also allowing expansion in the future. Including this range of sensors allows us to be confident that the board design will support most types of sensors.

A major motivator was to target the open source community by making the final hardware and software designs freely available in the hope that others can use and developed this sensor node platform. This should help increase to pool of knowledge on wireless sensor networks and generate well used variants that provide a simpler design process to generate a node for a specific project. If the community succeeds then there should be less maintenance and effort required by the university to keep the platform up to date.

There has been great success in open source hardware with examples such as the Arduino development board which enables low cost microcontroller development and easy to use interactions. Another example is the RepRap 3D printer community where people from all over the world collaborate to build and develop low cost 3D printers.
Chapter 3

Design Process

This chapter will cover the process of designing a node up to having a working prototype. It will outline the component selection process that was gone through in order to select the correct components for the project, as well as the prototyping stages of design. It will cover all noteworthy components and describe the various aspects that make them appropriate for a sensor node, and in some cases talk about other equivalent components that were considered. It will also look at how the board design was developed through a process of prototyping and verification through software.

As outlined in section 2.3 the goal is to design a wireless sensor node that can provide a suitable platform for work and research in the area of sensor networks going into the future. The primary design constraints were defined as follows:

- RAM and flash memory greater than 8 kB and 64 kB respectively.
- Less than $100 per node in small volumes.
- Radio range of at least 200 metres at low data rates.
- Battery life exceeding the calculated life of the ScatterWeb nodes.
- A USB interface for easy programming and data collection.
- A compact form factor.

This left the choice of microcontroller open ended and many ARM and manufacturer proprietary architectures were considered. A well recognised proprietary architecture was chosen as it had a proven background in the WSN area and is designed to be used in ultra low-powered devices.
Radio transceivers within license-free ISM bands were considered as they were easy to obtain and their characteristics meant they were well suited to this type of project. A 433MHz ISM transceiver was chosen as it is designed for low powered, moderate data rate, and high range applications.

As outlined in section 2.5 the nodes were to include a range of sensors so they could serve in various roles. The sensors desired to be included, in order of importance, were:

- A temperature sensor.
- A 3-axis accelerometer.
- Two pushbuttons and LEDs.
- A pin header to allow extra devices and boards to be connected.
- A passive infrared motion detector.

### 3.1 Component Selection

Each component was carefully selected to ensure it was the correct choice for the sensor node and there would be no compatibility issues. This section details the choice of each component, especially the microcontroller and transceiver, and discusses the various features, interfaces and costs of the components.

Due to the research orientated aspect of the nodes, it was beneficial to have components that could provide more functionality for a similar cost, in both money and power, to other parts. This meant some of the components were chosen because of additional features they provided over others. This was especially true for digital interfacing sensors compared to their analogue counterparts.

#### 3.1.1 Microcontroller

As noted in chapter 2 the choice of microcontroller is limited to low power, low voltage devices that met the minimum quantities of storage and RAM, and integrated USB.

Originally System-on-a-Chip devices, which incorporated the microcontroller and radio transceiver, were considered as they were low cost and had a small footprint. However that limited the availability to devices that were more suited for short range IEEE 802.15.4 and ZigBee applications where primarily the frequency used was 2.4...
GHz, and had low sensitivity and transmit power. This would have potentially meant the radio range goal would not be achieved as the radios had a small link budget and were based on a high frequency.

Pin count of the microcontroller had to be taken into account as the range of sensors and additional pin headers would require a moderate quantity of pins. The microcontroller should provide at least 64 pins, but depending on the quantity left routable to the external header, possibly more than this. Due to the nature of the device, it would be spending long periods in a low powered sleep state and waking up at regular intervals. This would mean there was a need for accurately timed intervals provided by a 32.768KHz Real Time Clock if the nodes were to operate with any sort of synchronisation.

A significant amount of time was spent evaluating low-powered microcontrollers and architectures. ARM based architectures generally exceeded all the criteria and almost every major manufacturer carried a product based on these architectures. As well as ARM cores, most manufacturers had their own proprietary architectures optimised for various applications. Anything based on an x86 design was not considered as they are not targeted towards the ultra low-powered embedded market. An FPGA based design was briefly looked at but the speed of such a design would be wasted and it would have increased the price.

Table 3.1 shows a compiled list of microcontrollers that are designed with power efficiency in mind, have USB slave capability and meet the required RAM and storage sizes. These were the lowest powered microcontrollers found at the time. The choice of microcontroller was mostly based on voltage and current consumption in a particular sleep state where the Real Time Clock was kept operating and provided a waking interrupt source. Pin functions were looked at to make sure the microcontrollers could support the other components as well as breaking out useful functions.

From analysis of data provided, based on current consumption, the STM32 consumes too much current in its sleep state, making it unsuitable as it is not as low-powered as the rest of the microcontrollers. The PIC24 and LPC1347 have a minimum voltage that is too high, meaning they cannot continue operating for as long as the others.

A tough choice was made between the Energy Micro EFM32L, Atmel ATxmega, and TI MSP430 based upon their energy consumption. The EFM32L was not in stock from any supplier at the time the choice was made so it was excluded. This was unfortunate as a Cortex-M3 based design would have provided greater performance.
In general the problem with ARM microcontrollers is their focus on performance, with only moderate energy-efficiency goals. Although this was not the case with the EFM32L as this had both great efficiency and high throughput. This left the Atmel ATxmega and TI MSP430. Both are proprietary architectures with similar characteristics but in the end the MSP430F5634 [26] was chosen. This was because it consumed a third less power in active mode, featured more RAM and flash memory. As Paul Hunkin’s distributed operating system, Hydra, was designed using the same architecture it may in the future ease the time and effort of porting it to work with the new node. The microcontroller costs up to $16 in quantities of one.

Although the ATxmega does operate to a lower voltage, the chosen radio transceiver only works down to 1.8V, so the node would not be able to transmit or receive data even though the microcontroller could still operate.

### 3.1.2 Radio Transceiver

The choice of radio was particularly important as the wrong one could have meant that range and lifetime goals might not be met. Microwave radios that work in the short range devices (SRD) bands were investigated. These frequency bands provide general, licensed-free use on the conditions that the centre frequency and bandwidth stays within the defined limits of the band, and a maximum output power level is not breached.

Table 3.2 shows the list of radios that were investigated and their various attributes. Those displayed are all low cost, have a wide voltage range and support frequencies within the SRD bands. As we can see, almost all the transceivers have a data rate of at least 250kbps, but the output power varies with only the RFM22B matching and exceeding the legal limit for its frequency band.

The 2.4GHz ZigBee compliant EM260 was excluded due to its high current consumption and low link budget; likewise with the TI CC2500. The low receiving sensitivity, high current consumption and low output power of the AT86RF212 and CC110L meant they were less suitable than the SX1211 and RFM22B. The RFM22B [9] was chosen thanks to its higher output power, lower frequency, lower voltage and the fact that it is available as a complete module.

Although the RFM22B does not have the smallest sleep current draw the difference between it and the lowest is small enough to be negligible at all but the longest sleep periods. The RFM22B features an external shutdown pin that can be used to bring the current draw to 0.015µA for those longer interval applications.

The RFM22B module comes as a complete separate circuit board that is sol-
<table>
<thead>
<tr>
<th>CPU</th>
<th>STMicro STM32L151</th>
<th>Microchip PIC24FJ128GB</th>
<th>Energy Micro EFM32LG332</th>
<th>Atmel ATxmega128C3</th>
<th>NXP LPC1347</th>
<th>Texas Instruments MSP430F5634</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus size (bits)</td>
<td>32</td>
<td>16</td>
<td>32</td>
<td>8 /16</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>Architecture</td>
<td>Cortex-M3</td>
<td>PIC</td>
<td>Cortex-M3</td>
<td>AVR</td>
<td>Cortex-M3</td>
<td>MSP430</td>
</tr>
<tr>
<td>Clock freq. (MHz)</td>
<td>32</td>
<td>32</td>
<td>48</td>
<td>32</td>
<td>72</td>
<td>20</td>
</tr>
<tr>
<td>Memory</td>
<td>64</td>
<td>128</td>
<td>64</td>
<td>128</td>
<td>64</td>
<td>192</td>
</tr>
<tr>
<td>Flash (kB)</td>
<td>10</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>RAM(kB)</td>
<td>128</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Power</td>
<td>1.8 - 3.6</td>
<td>2 - 3.6</td>
<td>1.85 - 3.8</td>
<td>1.6 - 3.6</td>
<td>2 - 3.6</td>
<td>1.8 - 3.6</td>
</tr>
<tr>
<td>Voltage(V)</td>
<td>0.3 @ 1MHz</td>
<td>0.3 @ 1MHz</td>
<td>0.24 @ 1 MHz</td>
<td>0.475 @ 1MHz</td>
<td>0.5 @ 1MHz</td>
<td>0.32 @ 1MHz</td>
</tr>
<tr>
<td>Active (mA @ MHz)</td>
<td>0.1</td>
<td>0.3 @ 1MHz</td>
<td>1</td>
<td>1.1</td>
<td>1.2</td>
<td>1.8</td>
</tr>
<tr>
<td>Sleep w RTC (µA)</td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Peripherals</td>
<td>SPI</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>I²C</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>16-channels 12-bit</td>
<td>2</td>
</tr>
<tr>
<td>ADCs</td>
<td>20-channels 12-bit</td>
<td>8-channels 12-bit</td>
<td>8-channels 12-bit</td>
<td>8-channels 12-bit</td>
<td>2-channels 12-bit</td>
<td>2</td>
</tr>
<tr>
<td>DACs</td>
<td>2-channels 12-bit</td>
<td>None</td>
<td>4-channels 12-bit</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>AES Encryption</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Timer</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3.1: Microcontrollers investigated
dered onto the main circuit board. This integrates a Silicon Laboratories SI4432 transceiver IC along with a 32KHz crystal and the necessary decoupling and RF matching circuitry. This reduced the chance of an error due to a faulty matching circuit or incorrect component placement occurred during design and production, and sped up development and prototyping. The cost of this module is around $14 at quantities of ten.

The radio also boasts features such as an on-chip temperature sensor, an 8-bit ADC and low battery detection. In practice the integrated temperature sensor is indicative at best. When compared to the TMP112 temperature sensor and an external thermocouple, it ranged from two to four degrees lower than that reported by the other sensors. A software adjustable power output is available on the radio for on-the-fly power adjustment. This can be useful in situations where the nodes are deployed at random spacings and a node can detect how close it is to its peers via the Received Signal Strength Indicator and adjusts its output accordingly.

**Antenna**

Choosing a less common frequency band, 433MHz, meant a smaller amount of options when it came to antenna choice, and due to lesser demand, the prices of these antennas were comparatively higher than that of the 2.4GHz range for instance. After having firsthand experience with two Monash University Smart Packet Radio sensor nodes, it was decided to use a Linx ANT-433-SP2 chip antenna [16]. This was in contrast to the ScatterWeb nodes which had pads placed for a SMA connector, but a single core length of wire soldered in place instead. The reasons for this choice of antenna were the low cost, highly integrated package that reduced size, and the perceived benefit of a purpose built planar antenna. The cost of this antenna was several factors cheaper, $2, than an dipole antenna and connector but more expensive than a random wire antenna.

**3.1.3 Sensors**

**Temperature Sensor**

There was a very limited range of available temperature sensors that operated within the required voltage window, while offering low power consumption. A Texas Instruments TMP112 10-bit digital I2C temperature sensor was settled on [25]. This offered ±1°C accuracy, 0.0625°C resolution, and two adjustable temperature alarms. It has the widest voltage range of all the devices, 1.4V to 3.6V, and a current
<table>
<thead>
<tr>
<th></th>
<th>Semtec SX1211</th>
<th>TI CC110L</th>
<th>Atmel AT86RF212</th>
<th>HopeRF RFM22B</th>
<th>Silicon Labs EM260</th>
<th>TI CC2500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency (MHz)</td>
<td>860 - 960</td>
<td>315, 433, 868, 915</td>
<td>700, 800, 900</td>
<td>433</td>
<td>2400 (ZigBee)</td>
<td>2400</td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>2.1 - 3.6</td>
<td>1.8 - 3.6</td>
<td>1.8 - 3.6</td>
<td>1.8 - 3.6</td>
<td>1.8 - 3.6</td>
<td>1.8 - 3.6</td>
</tr>
<tr>
<td>RX current (mA)</td>
<td>3</td>
<td>16.9</td>
<td>9.2</td>
<td>18.5</td>
<td>36</td>
<td>19.6</td>
</tr>
<tr>
<td>Sensitivity (dBm)</td>
<td>-113</td>
<td>-112</td>
<td>-110</td>
<td>-121</td>
<td>-99</td>
<td>-104</td>
</tr>
<tr>
<td>TX current (mA@dBm)</td>
<td>25@10, 16@1</td>
<td>34.2@12, 16.8@0</td>
<td>17@5, 13@0</td>
<td>30@13, 18@1</td>
<td>28@3, 24@0</td>
<td>21.5@1, 21.2@0</td>
</tr>
<tr>
<td>TX power (dBm)</td>
<td>12.5</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>2.5</td>
<td>1</td>
</tr>
<tr>
<td>Max data rate (kbps)</td>
<td>200</td>
<td>300</td>
<td>250</td>
<td>256</td>
<td>250</td>
<td>500</td>
</tr>
<tr>
<td>Lowest sleep mode (µA)</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.45</td>
<td>1</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 3.2: Radio transceivers investigated
consumption of 15\(\mu\)A when its \(I^2C\) interface is operated at 400KHz. The cost of this component was roughly $3.50 at quantities of one.

**Accelerometer**

A Freescale MMA8453Q 3-axis 10-bit digital accelerometer [7] was chosen as it has a wide voltage window and supports many additional functions such as tap and orientation detection. This sensor is also pin compatible with higher resolution 12-bit and 14-bit models, but these were not used as they added expense and this accuracy was not needed for development. This device can detect single and double taps greater than an adjustable threshold on each axis, can detect orientation changes in all six directions, and has a selectable scale of ±2g, ±4g and ±8g. This accelerometer operates down to 1.95V and consumes 24\(\mu\)A in normal use so it fits the energy requirements well, while being one of the cheapest available sensors at $2. It uses \(I^2C\) as its main communication interface in addition to two configurable interrupt pins for signalling the microcontroller that data is available.

**Motion detector**

The infrared motion detector chosen was a very low-power consuming Panasonic EKMB120111X [20] with a digital output. This is a very basic device which has a constant draw of 2\(\mu\)A and outputs a high voltage level when it detects motion within its 90\(^\circ\) field of view, out to a range of 5 meters. Its high minimum operating voltage of 2.3V and very high expense, roughly $30, make this part unattractive but is required as there was no alternative available.

### 3.1.4 Power Source

While experimenting with alternative energy sources would have been interesting, it falls outside the scope of this project. Conventional alkaline batteries were decided upon as they provide a high level of current capacity in a reasonably small package while being low cost. When compared to a AAA battery, alkaline AA batteries can have over twice the current capacity while not being significantly more expensive. Therefore two AA batteries were chosen as the primary power source.

Lithium coin cells were also looked at due to their low size and price but their current capacity was too low to meet the proposed lifetime goals and the nominal discharge rate was too small to supply a radio transceiver.
The idea to operate without a voltage regulator was brought up due to the high inefficiency of both linear and switching-mode regulators, and the high expense of switching-mode regulators. It was for these reasons that no regulator was used and all of the components were chosen to have a wide voltage window. In theory this should extend the node’s life as there will be no wasted power due to regulation and it lowers the overall board cost by minimising part count.

3.2 Prototyping

In order to minimise any issues that might come from an initial design of the circuit boards, two lots of all of the major components were sourced on what are known as breakout boards, as well as a specialized JTAG programming and debugging interface. These boards include the individual component, along with any necessary supporting componentry such as bypass capacitors or pull-up resistors, and route the device’s pins out to standard 0.1” pitch pin headers. This is to allow the components to be used in breadboards that are common in electrical prototyping and features rows and columns of holes with interconnecting wires. The microcontroller came on a special breakout board termed a development board which added features such as a JTAG programming interface, USB, and crystal clock sources.

With the parts in hand, software was developed that enabled the microcontroller to communicate with each device in turn. To begin, an Arduino was used to test each component as this provided a simple process of familiarisation and a substantial amount of example code for this platform was available as these components are commonly used. Once there was an understanding of how each component was required to work, they were wired in turn with the MSP430 microcontroller and code was developed from scratch that brought the device to life and enabled functionality. This involved writing drivers to allow the use of the \textit{I}^\textit{2}C and SPI serial lines, as well as functions to setup each of the devices internal registers and state. Figure 3.1 shows a block overview of how each component was interfaced to the microcontroller. The accelerometer and temperature sensor share an \textit{I}^\textit{2}C bus, along with three extra microcontroller pins that are used for interrupts. The radio interfaces over a SPI bus along with four GPIO pins for addressing, shutdown, and interrupts.

The motion detector and push buttons each interface over interrupt enabled pins on the microcontroller which are capable of waking the microcontroller from most sleep states.
3.3 Software

In order to debug and demonstrate that the hardware was working and capable of performing as required, software was developed to interface to the devices, along with any necessary drivers. A very basic operating system that collected and transmitted sensor data was written to tie the various interfaces together and demonstrate a complete working node. In addition to demonstrating basic functionality of each component, there was a need to know if it could operate in a way that was suitable for a sensor node. This meant that throughout the prototyping and development stages that the package would be tested as if it was a deployed node.

3.3.1 Hardware Drivers

The I^2^C and SPI drivers operate by polling for completion, and the microcontroller's onboard serial hardware is used to provide the signalling, rather than a slower software approach that deals with sending the bits itself. Using polling means there is an increased time spent busy waiting. Instead the microcontroller could go into a low powered mode and be woken by interrupts from the input and output buffers and others. Although energy profiling would have to be completed in-order
Figure 3.2: Breadboard prototyping
to determine whether this is worthwhile as the switching and stabilisation between operating states may be longer than clocking completion of the bus.

Developing the drivers led to the discovery that sharing the I\(^2\)C driver between the temperature sensor and accelerometer generated an issue where the accelerometer used an unusual method of interfacing. To read or write data, the desired register was sent on the bus first and a repeated start condition was sent followed by the data to be written or read. This meant another layer of abstraction to the device interfaces and led to another problem. An error in the hardware of the microcontroller was found to be a problem that occurred when a start bit was sent without the I\(^2\)C session ending under a second read operation; in other words a repeated start. This causes the I\(^2\)C communication to terminate with no data received. A workaround is to reinitialise the I\(^2\)C hardware between every accelerometer read and write command to ensure correct operation. This type of issue demonstrates the importance of prototyping within the design process.

UART drivers were written to allow TTL RS232 serial communication from the UART pins of the microcontroller, which was connected to a Future Devices FT232 UART-to-USB converter. This enabled simple serial communications with no flow control between the node and a computer. On receipt of a packet it could be outputted to serial for collection or debugging. This also utilised polling as it was deemed unnecessary for optimisation because the receiving node would be the only one utilising this and it should be connected to an external power source such as USB.

A basic driver for the 32KHz RTC was made. This was used to test that the RTC could wake the microcontroller from a low powered multisecodn sleep state. This worked by using one of the microcontroller’s onboard timer modules in a continuous up counting mode. A low powered mode, LMP3, was used which kept the 32KHz crystal oscillating but turned off the internal digital oscillators, frequency-locked loops and CPU, and drew a constant current of 1.8\(\mu\)A.

### 3.3.2 Device Interfacing

The drivers provide the low level hardware interfacing but in order to utilise the hardware higher level device interfaces were designed to build on the drivers and provide functions such as a temperature read or packet reception.

The temperature sensor interface provides access to the internal registers for retrieval of temperature values, high and low threshold values, and configuration registers.
The interface for the accelerometer allows it to be configured into the chosen measurement scale, sampling frequency, and the setup of the various extra functions. Handlers for the tap and orientation detection provide quick access to the internal interrupt registers to return values associated with the characteristics of the detection.

The radio transceiver device interface is an adapted form of an interface provided for use with an Arduino development board [19]. This enables sending and receiving data packets, packet header access, and configuration of various parameters such as centre frequency and data rate, amongst other things. The interface provides a front for sending and receiving unaddressed, 256 byte long data packets. Although not implemented, [19] provides Arduino code examples for designing a simple addressed and routed network as well as a scalable and recoverable mesh network, similar to what is defined in the ZigBee protocol [30].

The motion sensor did not require any special configuration as it is a very simple device that only used an interrupt vector of a pin. The two push button switches were the same.

A simple program was made that tied all the interfacing together and broadcast unaddressed, unreliable packets for testing and demonstration purposes. On the other end, one node had a receiving program which collected the packets and displayed them onto a terminal window via an FTDI FT232 UART-to-USB converter. The transmitting program was extended so that a variable sleep state was used to conserve power, based on the 32KHz clock driver. The use of separate drivers and interfaces should allow easy integration with a RTOS to provide real-time performance and a safe operating environment.
Chapter 4

Implementation

This chapter will present the steps that were undertaken to create the node hardware design based on the prototyping stage. In chapter 3 the process of selecting each major component was shown, how each of the components interfaced with one another, along with software verification that the interfacing worked as intended. The following will extend the prototyping work done (covered in section 3.2) and present a feature complete version on a manufactured circuit board, called the Waikato Open Sensor Platform (WOSP).

Performance comparisons of the new sensor node versus the older ScatterWeb node will highlight the improved power consumption and expected battery lifetimes. A practical comparison of communication ranges using various antennas and data rates will demonstrate some ranges possible. A second revision of the board will be shown that improves upon the size, features and uses components with better availability than the first board.

4.1 Circuit board design

Once the breadboard and software design was completed the prototype could be realised in a custom printed circuit board to deliver a compact package that integrates together the microcontroller, radio transceiver, antenna, sensors, and any necessary smaller components and external interfaces. This process was completed in two stages, circuit schematics that represented the circuit board at a highly descriptive level, and board layout, which using this description, places components and routes traces between them. All the board design was completed using Cadsoft EAGLE PCB.
4.1.1 Schematics

The circuit schematics represents a circuit in a simplified graphical manner that uses icons as components and lines between specific points on icons as wires or traces. It does not describe in detail physical aspects of a circuit, merely a diagram of what components are used and the interconnections. Most of the wiring completed in prototyping could be directly translated into schematic form. The remaining parts included the external interfacing, clocks, and decoupling, filtering and pull-up components.

Figures A.1 to A.4 show the schematic sheets created for the board. Figure A.1 contains the microcontroller, all immediate supporting circuitry and named wires linking to components on other sheets. The supporting componentry consists of decoupling and storage capacitors, a low pass filter and crystal circuitry. In figure A.2 and A.3 the sensors, LEDs and switches, the radio and antenna are shown. Figure A.4 shows the external interfaces made up of USB, the expansion port, JTAG header and power circuitry.

A lack of pull-up and pull-down resistors on some of the components is made up for by configurable internal resistors within the microcontroller, which reduces part count. There are no current limiting resistors on the two LEDs as their forward voltage was high enough to limit the current at supply voltage. This meant that the LEDs would only operate at above 3v, although it was thought that the LEDs would only be in use when powered from USB anyway, so it was not a concern.

4.1.2 Board layout

With the completion and validation of the schematics, the board layout was started. First, footprints of all the components needed to be created. This meant creating a virtual copy of the component based on its measurements obtained through the datasheets. For example, the microcontroller needed 100 pads placed in a square at the precise measurements described in its datasheet, along with a silkscreen to denote its position and the orientation of pin one. A silkscreen is an extra layer on a PCB with no electrical purpose, it is a non-conductive layer for labels and drawings.

Once the footprints were made, the schematics were converted into the beginnings of layout called a “rat’s nest”. This consists of components randomly placed on a virtual board with unrouted connections going between the components. The components were placed in positions, starting with the larger and more central components such as the microcontroller and radio, and the other components placed
around them in ideal orientations. This was performed iteratively to find locations that all the components could be easily connected while conserving space. Manual routing of each individual connection was used as this gave complete control and a more optimised solution compared to automatic routing, allowing tidier connections and a more compact board. The WOSP boards were fabricated by [4], with trace and via-hole measurements larger than the minimum design rules specified to ensure proper fabrication on the first run of boards.

WOSP was designed with four electrical layers consisting of a top signal layer; second ground layer; third voltage layer; and a bottom signal layer. This ensured a compact board. The added expense was negligible compared to a larger two layer board due to the need for complex routing of power and ground. These layers can be seen in figures A.5 to A.8, with the topmost layer figure A.5. Four boards were fabricated and assembled as this quantity was inexpensive enough to provide testing while allowing for any accidental destruction through power-up and testing. The size of WOSPr1 is shorter but slightly wider than a credit card. Several large components, the lack of components on the underside, and larger traces and via sizes have dictated the dimensions. The cost of this board does not meet the goal of costing less than $100. The total component cost came to about $80 per board at ordering quantities of one. Board manufacture for the four boards came to $50 a board and then assembly was another $80. These prices were all because of the low quantity prototyping run, and at a full production run it should be much cheaper.

Figure 4.1 shows an annotated render of what the final board looks like. The microcontroller sits in the middle with the radio close by, this allows short and therefore fast traces. The antenna is placed with no layer running underneath it and ground vias between its ground connections as per the datasheet [16]. The accelerometer and temperature sensors are close together as they share two of the same microcontroller pins. This first revision board is called WOSPr1.

4.1.3 Testing and Verification

Once the completed boards were received, all functionality was tested and several minor issues were found that did not affect function but would have to be fixed in a second revision. Aside from the following issues listed, everything worked as intended. The microcontroller could collect data from all the sensors, the radio could send and receive packets, and low-powered sleep operated correctly. The fact the board worked as intended the first time shows the effectiveness of the prototyping and development stages.
The temperature sensor was not thermally isolated from the rest of the board. It was positioned near the middle of the board and three layers of copper ran directly underneath it, providing it with conducted heat from the surrounding components, in particular the microcontroller and accelerometer. This skewed temperature readings. In worst cases, when the microcontroller was constantly running, temperature readings were of around 27° C when the ambient was approximately 20° C. In many practical situations this would not be a problem because most of the time the microcontroller would be in its low-powered sleep modes to minimise current draw and therefore wasted heat. A sleep of five seconds was observed to be long enough to dissipate the generated heat and provide accurate readings.

During assembly (by a professional assembly plant), on all boards, two pins of the accelerometer were shorted to the ground plane and therefore would not function. The pins served as interrupt lines to the microcontroller to alert it of changes to the environment or if a new reading was available. The cause of this is suspected to be a via through to ground placed directly underneath the device as well as the ground pins that flanked the shorted pins. A workaround is to poll the interrupt vectors over the I2C bus, although the effect of this is increased power consumption and time spent awake.

The design goal of minimising size lead to a decision to use a smaller 3mm x 4mm button rather than the commonly used 6mm x 10mm tactile pushbutton as it was
deemed too large. Unfortunately placement of these was not thoroughly considered and for two of the buttons the close proximity next to tall pin connectors meant they couldn’t be pressed without a pointed object.

RTC circuitry on all four boards display some time skewing when operating in low-powered mode. This is most likely due to the top ground fill that surrounds the clock traces, and the load capacitors are also connected to this fill. Noise on the ground may be being picked up and adding itself to the oscillations. This phenomenon is particularly troublesome with 32KHz clocks as they are driven by very low current and easily susceptible to noise. Unfortunately there is no fix available to this, short of redesigning that section of the board. Although one possibility would be to synchronise time between the nodes every time they are awake, assuming the wake window is long enough and length of time asleep hasn’t offset any boards so much to be outside of this window.

Testing revealed some problems with the chip antenna. It appeared to have high loss. In investigating this problem, eventually an old datasheet was discovered specifying -8.6dBi “gain” [15]. This information was omitted from the current datasheet. A calculation of the efficiency of the antenna is as follows. The equation for power in deciBel is shown in equation 4.1 using the gain factor as the comparison ratio.

\[-8.6 = 10 \log_{10} (G)\]  

Rearranging equation 4.1 to calculate the gain factor leads to equation 4.2 which equals 0.138. That is, no more than 13.8% of the power entering the antenna is radiated out from the antenna.

\[G = 10^{(-8.6/10)}\]  

As no antenna is isotropic, 13.8% is the best radiated power gain in one particular direction. In all other directions, both azimuth and elevation, the gain will be less than 13.8%. This means over 86% of the power sent to the antenna is lost. This was clearly not ideal and the boards were modified to output the antenna connection to an SMA connector for an external antenna. A dipole antenna tuned for an unknown frequency was available and, as an experiment this was tried. It provided vastly improved range, as outlined in section 4.3.2.

It was noticed after the first boards were made that the microcontroller and the rest in its pin compatible family line had gone out of stock at major distributors. This lead to a decision redesign the boards to use another microcontroller within the
same MSP430 family, as well as to address the known issues from the first board.

### 4.2 Revision 2

To improve upon the first board design, fix the problems discussed, reduce the board size and relieve some supply and cost issues, a second board, WOSPr2, was designed.

A TI MSP430F5529 microcontroller is used on WOSPr2 as this met the constraints defined initially while having huge stock quantities at the major distributors. It features 128kB flash memory and 8kB of RAM but an extra 2kB can be freed when USB is not used. Power consumption is slightly higher at 0.36mA at 1MHz (compared to 0.32mA in active) and 1.9\(\mu\)A in low-power mode (compared to 1.8\(\mu\)A of WOSPr1). The voltage window hasn’t changed and there is still the same quantity of I\(^2\)C and SPI channels. The pin count of this chip has been reduced to 80 pins which allows the overall size to be reduced. This aspect is important in the task of reducing the board size and allowed the board dimensions to be brought down to the same size as the battery container without having to place any major component on the underside of the board.

In addition to a change in microcontroller several components were removed or relocated to allow this reduction in board size. The motion detector was removed due to its expense, small stock availability and because we do not have any current applications that require it.

The JTAG programming header was modified as, although it directly mates with the programming adapter from TI, it occupies too much space and only half of the pins are used by the adapter. Instead, two rows of pin headers allow more of the microcontroller’s pins to be routed out and the JTAG pins shared half of one of these.

The chip antenna was not kept because of its poor performance and this was replaced by pads designed for an SMA connector. Like the ScatterWeb boards, this allows for a SMA antenna or a cheaper strand of wire instead.

More consideration was given to isolating the 32KHz crystal to improve the interval timing accuracy. A ring isolated from the top layer ground was formed around the crystal and load capacitors. This was tied through to the ground plane at two points and made part of the analogue-to-digital converter ground circuitry as recommended by TI [24].

The temperature sensor is now isolated from the other components by keeping all
four layers 5mm away from the sensor, and also keeping the sensor close to the edge of the board so that heat will dissipate before reaching the sensor. The accelerometer no longer has a ground via underneath it. This should avoid any of its pins being shorted to ground.

To allow more functionality, a microSD card socket was added to the underside of the board. This will allow features such as data logging in cases where a radio link goes down so a node can stockpile until it can flush the data. It can also be used in cases where too much sensor data is captured to store in the microcontroller’s internal flash and has to be temporarily stored elsewhere while it transmits the data. For example, we have users interested in using WSNs attached to camera modules for low rate, remote photography of environmental conditions. This could be used in these situations where a still camera image is taken and needs to be stored while being transmitted. Hydra also supports memory paging to an SD card if memory use is high.

The final board, as shown in figure A.13 to A.16. WOSPr2 is still four layers in the same order as before. All the main components are on the top layer but the design also uses the bottom layer for some smaller discrete components. Even though the battery container sits on the bottom of the board, the legs of the through hole components push the container away from the board, leaving enough space for these small components. The board has been designed to meet the smallest design constraints set by [4], meaning the traces, trace spacing and vias are as small as manufacturing allows, this has helped the board become smaller. From figure 4.2 we can see the size difference between the two boards. WOSPr2 is a little over half of the original WOSPr1 board and now the size of the battery case. This is due to eliminating the large JTAG header, motion sensor and chip antenna. as well as using a smaller microcontroller and component placement on both sides of the board.

The cost of this board does meet the goal of costing less than $100. The total component cost came to about $50 per board at ordering quantities of one. Board manufacture for the four boards came to $25 each board and assembly did not cost as they are being hand assembled. This board price is half the first board because it is half the size. The component cost is less because of the change in microcontroller and not including the motion sensor. Assembly costs may have been cheaper than the first boards as the board is smaller so the solder paste stencil would be reduced.

To save costs, this board is being assembled by hand so it has not been verified working yet. The timing of manufacture means, although we have the boards, other aspects of the project have taken precedent and they have not yet been assembled.
Figure 4.2: WOSPr2 (left) compared to WOSPr1 (right)
### 4.3 Performance

In addition to the goals inherently met in the design, like the memory and cost, the following goals also had to be met in order to verify the success of WOSP1: battery life exceeding that of the ScatterWeb nodes; and a communication range greater than 200 metres at low bit rates. To complete these, several theoretical and practical tests and calculations were undertaken. Battery life was calculated through power consumption values obtained from the datasheets, and communication range was tested from setting various data rates and measuring the maximum receive distance.

#### 4.3.1 Battery

Table 4.1 shows the active and sleep mode current consumptions of the WOSP1 and ScatterWeb nodes. These values are obtained from the datasheets of each component and particular operating conditions are specified.

We can see that, in both active and sleep modes, with all devices operating, the WOSP node has less current draw than the ScatterWeb node. This is particularly true in sleep mode with a 35 times reduction in current consumption. In active
mode, all components of the WOSP node, except for the radio, use less current than their counterparts on the ScatterWeb nodes. The radio uses more current if it is transmitting at a higher output power. The datasheet does not specify for the same output power as the CC1020 radio used on the ScatterWeb boards so we cannot compare the new radio directly to the ScatterWeb one.

In table 4.2 two different usage intervals and the maximum theoretical battery life of both nodes are presented. Both intervals use a period of 15ms spent active and the rest of the interval in a low-powered sleep. This active length was obtained through measuring the time it took WOSP to wake up, collect data, transmit the data at 57.6kbps and then return to sleep again, while running at 8MHz. With a one second sampling interval this wake period equates to a 1.5% on duty cycle, and at five second interval this is 0.3%.

One point worth noting is the battery capacities in amp-hours. The ScatterWeb nodes use AAA batteries which have a maximum capacity of 1200mAh for an alkaline type [6]. In contrast, the WOSP boards use AA batteries which are physically slightly larger but have a much higher capacity of up to 2600mAh [5]. In the calculations, both batteries are said to have a lower capacity than maximum as not all batteries with similar chemistry are the same. The calculations for the ScatterWeb boards are calculated with both the AA and AAA current capacities, which makes the comparison normalised. The average current is calculated by equation 4.3:

\[
A_{\text{average}} = \frac{(t_{\text{awake}} \times A_{\text{awake}}) + (t_{\text{sleep}} \times A_{\text{sleep}})}{t_{\text{awake}} + t_{\text{sleep}}} \tag{4.3}
\]

A lifetime is calculated by dividing the current capacity of the battery by the average consumption, as shown in equation 4.4.

\[
t_{\text{lifetime}} = \frac{Ah_{\text{capacity}}}{A_{\text{average}}} \tag{4.4}
\]

Table 4.2 shows that with the use of newer technology and a larger battery, the WOSP nodes have succeeded at their intended goal of a longer battery life.

### 4.3.2 Communication Range

Achieving the intended 200 metres communications range at low bit rates was harder to calculate and easier to measure. Due to the problems encountered with the chip antenna this goal was initially not met but a hardware modification to WOSPr1 allowed it to be achieved. The modification consisted of removing the chip
### Table 4.2: Node lifetimes

<table>
<thead>
<tr>
<th>Data Rate</th>
<th>Chip antenna</th>
<th>Dipole antenna</th>
<th>17cm single core wire</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4kbps</td>
<td>100 metres (LOS)</td>
<td>750 metres (non-LOS)</td>
<td>700 metres (non-LOS)</td>
</tr>
<tr>
<td>57.6kbps</td>
<td>60 metres (LOS)</td>
<td>500 metres (non-LOS)</td>
<td>400 metres (non-LOS)</td>
</tr>
<tr>
<td>120kbps</td>
<td>45 metres (LOS)</td>
<td>300 metres (non-LOS)</td>
<td>300 metres (non-LOS)</td>
</tr>
<tr>
<td>250kbps</td>
<td>20 metres (LOS)</td>
<td>200 metres (LOS)</td>
<td>200 metres (non-LOS)</td>
</tr>
</tbody>
</table>

### Table 4.3: Communication Ranges

antenna and soldering an SMA connector in place and using a dipole antenna. Also for testing, a quarter wavelength, 17cm long random wire antenna was used as this is very cheap. As testing shows, it performs better than the chip antenna.

Table 4.3 shows the measured ranges with varying bit rates and antenna configurations. These tests were conducted with the transmitter power output set to +14dBm using GFSK modulation, no Manchester encoding, a frequency deviation of 70KHz and no automatic frequency control. LOS means Line-of-Sight as some of the tests were short enough to be in a clear area but the longer ranges had to go through obstacles such as a small block of trees, houses and the side of a hill. The distances were calculated using Google Maps. They are conservative as they represent the shortest distance within the measurement error and are rounded down.

From table 4.3, we can see that the dipole antenna performs the best at all data rates. The gain of the dipole and wire antennas are unknown. The dipole
antenna came from an unknown source and is probably not a quarter wavelength for this frequency. Despite their non-ideal characteristics, both antenna perform better than the chip antenna. The second board revision has the pads for a SMA dipole or wire antenna.
Chapter 5

Conclusion

This report has presented the design and implementation of a wireless sensor node for use in research and in real-world deployment.

A replacement of the ScatterWeb sensor nodes was needed, and after an unsuccessful search for a commercial replacement, a new node was developed in-house. Many aspects came into designing a sensor node and these were outlined in section 2.1. The primary goals for this node, as defined in section 2.3, were:

- RAM and flash memory greater than 8 kB and 64 kB respectively.
- Less than $100 per node in small volumes.
- Radio range of at least 200 metres at low data rates.
- Battery life exceeding the calculated life of the ScatterWeb nodes.
- A USB interface for easy programming and data collection.
- A compact form factor.

A broad range of sensors and interfaces were desired to allow the boards to be generic by design and widen their potential usage. These were:

- A temperature sensor.
- A 3-axis accelerometer.
- Two pushbuttons and LEDs.
- A pin header to allow extra devices and boards to be connected.
• A passive infrared motion detector.

There were some small design faults but it is expected that the first node is capable of meeting the needs of most real applications. A second board that resolves these issues and enhances the first board has been designed and manufactured but is still to be assembled and tested.

Although the radio range was not achieved on the first node, that is remedied in the second sensor node. The motion detector was tested to work on the first board but dropped on the second because of cost and that there are no current applications that need it. The first board did not meet the goal of being less than $100 but with some alterations and a reduction in size the second board has. Aside from these points, all other goals were met in both the first and second sensor nodes.

The designed sensor nodes also allow tailoring to suit our needs in the future with expansion header and design files to allow a complete manufacture of a modified board. The effects of open sourcing this project should help the embedded sensor networks community and further develop the platform. A vision for an open platform that many can use will enable many tested variants and ease developed and enhancement of future generations of sensor nodes.

5.1 Future Work

There are numerous opportunities for future work and applications now that we have a powerful and cost effective sensor node platform. These are introduced in the following paragraphs.

The porting of Hydra on to this platform will allow both the nodes and Hydra to be properly tested in the environments they were designed for. Because of WOSP’s lower cost there is now the cost effective ability to deploy large numbers of nodes.

In the search for information to help write this report I came across a new low-powered microcontroller by Freescale [8] released a couple of months after the prototyping began. This microcontroller is based on the new ARM Cortex-M0+ architecture and boasts half the energy consumption of the MSP430 family. It has a 32-bit ARM core that can run up to 48MHz, and features 16kB RAM and 128kB flash storage. A redesign of the boards using this microcontroller may happen soon as this architecture looks very promising and has significantly more performance than the MSP430 family.

With the new sensor node designed, there is an opportunity to pursue two practical applications that are already in mind and work will start in November 2012 to
bring them to fruition.

The first application involves the integration of an ultrasonic microphone and amplifier circuitry for the use in monitoring New Zealand short and long tail bats in the Pureoara Forest area. Nodes will be set up in clusters and through the use of Hydra a collection could perform intensive Fourier analysis to identify individual bats and track them between nesting sites.

The second application integrates a small still camera module that will be used for taking periodic images for river water level monitoring in a study of the ecology of streams that periodically dry up. At the moment specialised digital cameras are hung over the river and periodically an image is taken and stored. Once a year the images are retrieved from the cameras and analysed to predict the river levels. Instead, sensor nodes could perform this and transmit the images to a base station that is connected to the cellular network, with images uploaded to a central site.

After a tidy up of the software stack and hardware design files, the project will be open sourced to the community in the hope that the community will benefit from it and vice versa.

WOSP fills an important role in wireless sensor networks and is expected to enable future research into sensor nodes, systems software and applications.
Bibliography


Appendix A

First Appendix

The following are design files used in the creation of the WOSP sensor board and show the schematic design, components used, and board layout of all four layers.

A.1 WOSPr1

This section covers the first board designed, WOSPr1.
Figure A.5: WOSP1 board top layer
Figure A.6: WOSPr1 board second layer - Gnd
Figure A.7: WOSP1 board third layer - Vcc
Figure A.8: WOSP1 board bottom layer
A.2 WOSPPr2

This section covers the second board designed, WOSPPr2.
Figure A.9: WOSPr2 schematics sheet 1
Figure A.10: WOSPr2 schematics sheet 2
Figure A.12: WOSPPr2 schematics sheet 4
Figure A.13: WOSPr2 board top layer
Figure A.14: WOSP:r2 board second layer - Gnd
Figure A.15: WOSPr2 board third layer - Vcc
Figure A.16: WOSP\textsubscript{r}2 board bottom layer