DESIGN AND IMPLEMENTATION OF A HARDWARE PLATFORM FOR WIRELESS SENSOR NETWORKS

By

NC Amber Fareed

PC Sehrish Akram

NC Shaleem David

NC Shah Nawaz Anjum

Submitted to the Faculty of Electrical Engineering Department

National University of Science and Technology, Rawalpindi

in partial fulfillment for the requirements of a B.E Degree in

Telecom Engineering

JUNE 2010

ABSTRACT

Wireless Sensor Networks (WSNs) consist of a network of wireless nodes that have the capability to sense a parameter of interest like temperature, humidity, vibration etc that is often relayed to a base station through the network formed amongst these nodes. The devices used are typically characterized by low cost, low power and are rugged in operation and are commonly referred to as motes in the WSN domain. These motes are not readily and easily available in Pakistan. They have to be imported and not all educational institutions in the country have administrative procedures in place that enable them to procure these devices easily. It is therefore important that such devices are easily and readily available to the educational and research communities.

Our objective was to design and implement a mote that integrates programming, computation, communication, and sensing onto a single system and provides an easy user interface for operating and deploying it. This has been carried out by designing and implementing the hardware and integrating it with the operating system. The designed mote has been compared with the reference design (TelosB). The result of the project is a hardware platform that can be deployed in innumerable applications in the field of wireless sensor networks.

Dedication

This project is dedicated to our parents

Acknowledgments

All praises to Almighty Allah who has given us knowledge and strength to complete the objectives of this project. Vital support has been provided by our project supervisor Maj. Dr. Adnan Rashdi. He has been a continuous source of ideas and encouragement. We are grateful to lecturer Safwat Irteza for his concerned efforts and assistance. We are also grateful to Sir Farooq bhatti for foreseeing the upcoming complications and providing us with his valuable guidance.

TABLE OF CONTENTS

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

1.1 Wireless Sensor Networks ..……………………………………… 1

1.2 Mote ……………………...……………………………………… 2

1.2.1 Hardware …………………………………………………… 3

1.2.2 Operating System ………………………………………… 4

1.3 Motivation ………………...……………………………………… 5

1.4 Thesis Organization ……....……………………………………… 6

Chapter 2 Literature Review ……………………………..…………… 7

2.1 Related Work …………………………………………………… 7

Chapter 3 Design Considerations ……………………………………... 10

3.1 Hardware Selection ...…………………………………………… 10

3.1.1 Microcontroller …………………………………………… 11

3.1.2 Radio ……………………………………………………… 11

3.1.3 Flash Memory ………………………………………….....11

3.1.4 Temperature Sensor …..………………………………… 12

3.2 Antenna Selection ...……………………………………………… 12

3.2.1 Internal Antenna ……..…………………………………….. 12

3.2.1.1 Inverted F Antenna ……..………………………………….. 13

3.3 PCB Specifications ..………………..………………………………….. 14

3.4 Matching Network …………..…………………..…………………… 15

3.4.1 Comparison of Different Balun Solutions ……...………… 16

3.4.1.1 Transmission Line Implementation ……….…………… 16

3.4.1.2 Lumped Element Implementation ……………….……….. 16

Chapter 4 Design Implementation ………….…………………………… 18

4.1 PCB Design ...………………………………………….……….. 18

4.1.1 Software Selection …………………………….……… 18

4.1.2 Footprints Designing ………………………….……… 19

4.1.3 Schematics Drawing ………………………….……… 19

4.1.5 Export of schematic to PCB ……………….………… 20

4.1.6 Components Arrangement ………………………… 21

4.1.7 Routing ……………………………………………… 21

4.1.8 Output File Generation …………………………... 22

4.2 Antenna Design ……………………...……………..…………….. 23

4.2.1 Software Selection ………………..…………………... 23

4.2.2 Antenna Simulation ………………..…………………….. 23

4.2.3 Design Optimization ………………..……………………. 23

4.3 Matching Network Performance Analysis ……………….……… 26

Chapter 5 Performance Analysis …………………......……………………. 29

5.1 TinyOS ……………………...………………..…………………….. 29

5.1.1 Hardware – Software Integration ……..…………………… 30

5.2 Hardware Testing ……………..……………..…………………. 30

5.3 Measurements ………………………………..………………..…... 31

5.3.1 Radio Range Measurement ……….………………….… 31

5.3.2 Power Consumption Measurement ……………….……… 32

5.3.3 Cost Analysis ……………………………...……….…… 33

Chapter 6 Conclusion and Future Work ...………………..……………... 34

6.1 Conclusion ……………..…………………………….………………… 34

6.2 Future Scope ..…………………………………..………………... 35

Appendices

A Schematic Diagrams ……………………………….……………… 38

B PCB Layout ………..…………………………….………………… 41

C Process of Installing an Application on Mote ……….…………… 44

D nesC Code for Temperature Sensor …..……………..………..…. 46

Bibliography …………………………………………………………...……... 50

LIST OF TABLES

Table no.	Title	Page no.
2.1	Evolution of Motes	8
2.2	Comparison of Popular Motes	9
4.1	Summary of the Properties of IFA	24
5.1	Power Consumption	32
5.2	Cost Breakdown	33

LIST OF FIGURES

Figure no.	Title	Page no.
1.1	A Wireless Sensor Network Architecture	2
1.2	A Commercially Available Mote	3
1.3	Architecture of a Mote	4
3.1	Inverted F Antenna (IFA)	13
3.2	PCB Layer Stack	14
3.3	Balun Circuit Integration	15
3.4	Schematic and Layout of Transmission Line Implementation	16
3.5	Schematics and Layout of the Lumped Element Implementation	17
4.1	Reflection at the Input (S11)	24
4.2	Gain in dB	25
4.3	Radiation Pattern	25
4.4	ADS Model of the Balun Circuit	26
4.5	S13 and S23 (Transmission Coefficients)	27
4.6	S11 and S22 (Reflection Coefficients of Differential Ports)	28
4.7	S33 (Reflection Coefficient of Single Ended Antenna)	28

Chapter 1

INTRODUCTION

This chapter introduces the concept of wireless sensor networks and their application areas. It also introduces with the term ‘mote’ that is the wireless node, basic requirements of a mote and how are they achieved. It describes the motivation for taking up this project and the thesis organization.

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSN) consist of a network of wireless nodes (motes) that have the capability to sense a parameter of interest (like temperature, humidity, vibration etc). The sensed parameter is often relayed to a base station through the network formed amongst these nodes. They are commonly referred to as “motes” in the WSN domain. [1]

A number of motes when deployed in some area form a network. These motes are not only able to communicate with each other but can also exchange information and transfer it to a centralized station where it can further be analyzed. [2]

The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including telemedicine, detection of a major disaster such as earthquake, volcanic eruption, landslide and fire, area and

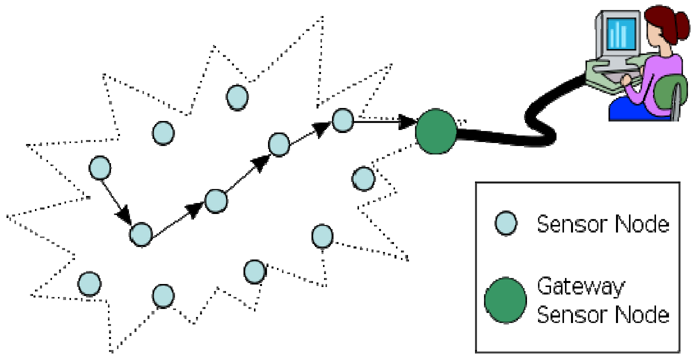


Figure 1.1 A Wireless Sensor Network Architecture

environmental monitoring, Habitat monitoring, Industrial and structural monitoring, Traffic flow monitoring and wildlife tracking.

A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station). [2]

In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year. [2]

1.2 Mote

The term mote refers to a tiny particle and these embedded devices for use in a WSN are essentially tiny in size and hence the name. A mote, is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. Some of the requirements of a mote are as under:

- The ability to have some amount of on-board processing.
- They should be able to communicate over the air and also have some basic networking capability.

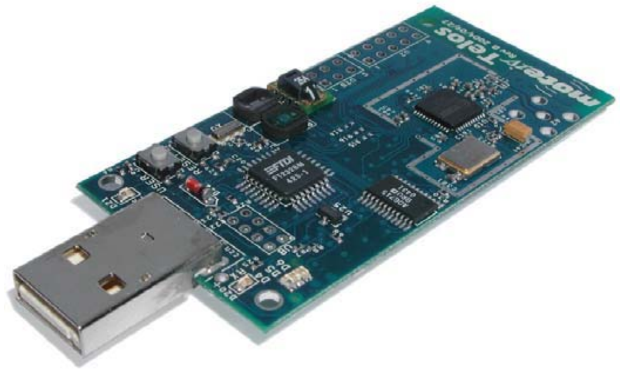


Figure 1.2 A Commercially Available Mote

- They should have sensors embedded in them so as to interact with the environment in which they are deployed and communicate the sensed parameter to a central location. [1, 3]

1.2.1 Hardware

To be able to fulfill the basic requirements, a mote should have the following components:

- Microcontroller
- Radio
- An embedded sensor or the ability to connect to sensors/sensor boards.
- On board memory: RAM, Flash.
- Power source

Figure 1.3 shows the basic architecture of a mote. [3]

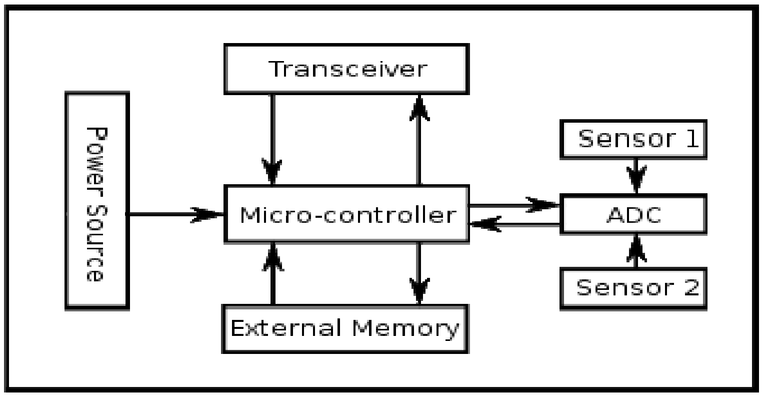


Figure 1.3 Architecture of a Mote

A mote might vary in size, the cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the size of the sensor network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. [1, 2]

1.2.2 Operating System

An embedded hardware platform is only useful if it can be programmed to run a desired application. To achieve this, the platform needs to be integrated with an embedded operating system. Operating systems for wireless sensor network nodes are typically less complex than general-purpose operating systems both because of the special requirements of sensor network applications and because of the resource constraints in sensor network hardware platforms. Wireless sensor network hardware is not different from traditional embedded systems and it is therefore possible to use embedded operating systems such as eCos or uC/OS for sensor networks. TinyOS is perhaps the first operating system specifically designed for wireless sensor networks. TinyOS is based on an event-driven programming model instead of multithreading. TinyOS programs are composed into event handlers and tasks. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS calls the appropriate event handler to handle the event. Both the TinyOS system and programs written for TinyOS are written in a special programming language called nesC which is an extension to the C programming language. Examples of other Operating Systems are Contiki, MANTIS, BTnut, SOS and Nano-RK. [1, 2]

1.3 Motivation

There are a number of commercial off-the shelf (COTS) motes available in the market today but none of them are readily and easily available in Pakistan. They have to be imported and not all educational institutions in the country have administrative procedures in place that enable them to procure these devices easily. It is therefore important that such devices are easily and readily available to the educational and research communities. Shipment charges of these motes are high enough to deter widespread use in educational and research institutes. Another motivation behind this project is the use of motes in innumerable applications. Some of which are

- Battlefield surveillance
- Telemedicine
- Detection of a major disaster such as earthquake, volcanic eruption, landslide and fire
- Area and environmental monitoring
- Habitat monitoring

- Industrial and structural monitoring
- Traffic flow monitoring
- wildlife tracking

We therefore, believe that easy availability of these motes coupled with low cost would facilitate growth of WSN applications.

We therefore define our problem statement as:

- To design and implement a hardware platform for use in WSNs
- Build a simple working prototype that can be improved upon.

1.4 Thesis Organization

Organization of the thesis is as follows. Chapter 2 discusses the evolution of various motes and comparison of commercially available motes. Chapter 3 discusses the design considerations. In this chapter issues relating to hardware selection, PCB design and antenna selection have been discussed and various parameters are defined. Chapter 4 discusses implementation of the PCB design, antenna and matching circuit simulation. Chapter 5 discusses the hardware-software integration, testing methodology of mote and the the range and power consumption tests. The work is summarized and the future scope is discussed in Chapter 6. The appendices at the end give detailed schematics of the mote, PCB layout, procedure for integration of mote with TinyOS and nesC Code for Temperature Sensor.

Chapter 2

LITERATURE REVIEW

2.1 Related Work

Although wireless sensor nodes have existed for decades for applications as diverse as earthquake measurements to warfare, the modern development of small sensor nodes dates back to 1998 in the Smartdust project and the NASA Sensor Webs Project. The Smart Dust project explored the possibility of using motes to form a massive distributed network. The main goal of the project was to explore micro fabrication technology's limitations and study the feasibility of such a platform. Because of the mote's discrete size, substantial functionality, connectivity, and anticipated low cost, Smart Dust was envisaged to facilitate innovative methods of interacting with the environment, and provide more information from physical regions less intrusively. The intended applications for smart dust were:

• Deploy sensor networks rapidly by unmanned aerial vehicles or artillery in battlefield scenarios and track enemy movements.

• Tracking the movements of birds, small animals, and insects and understand their behavioral patterns

• Monitor health of rotating machinery and understand the reasons for high cycle fatigue. [1, 3]

There have been a number of platforms which were developed on the lines of smart dust motes. The initial devices incorporated small microcontrollers (8 bit, 4 KB flash) and a simple radio (4 Kbps data rate) and their life time was up to a maximum of 2 years. Table 2.1 shows the evolution of mote and depicts the relative capabilities of the mote as it evolved. [1]

Table 2.1 Evolution of Motes

Mote Type	**WeC**	**Rene**	**Rene**	**DOT**	**Mica**	**Mica 2**	**Mica**	**Telos**
Processor	AT90LS8535		ATMega163		ATMega128			MSP430
Radio Chip	TR1000					CC1000		CC2420
Flash	32 KB				512 KB			128 KB
Sensors	No on-board sensors			Yes	No on-board sensors			Yes
Frequency	916.5 MHz					868 MHz		2.4 GHz
Tx Power	0 dBm							
Max Data	10 Kbps				40	38.4 Kbps		250
Power	36 mW					42 mW		35 mW
Sleep Power	45				75			6
Wakeup	1000		36		180			6
Interface	IEEE 1284 and RS232							USB

As can be seen from the table, the main focus in the design of the motes is low power of operation. Almost all the microcontrollers used in various designs operate at low voltages of 2.7 volts to 3.6 volts. Another important factor for the choice of microcontroller is the current drawn during its sleep and wake up time. Low sleep current and low wake up times minimize the duty cycle of a mote and therefore extends the life of a mote and consequent network life time. Telos design uses MSP430 microcontroller which has lowest sleep and wake times as compared to the other microcontrollers operating at low voltages. On-chip RAM also plays a vital role in the processing capability of the mote and MSP430 offers the maximum on-chip RAM in comparison. Another salient factor in the mote’s evolution is their usage of radio chips for communication. With the introduction of

802.15.4 standard for wireless sensor networks, the motes too started incorporating chips that use these standards for interoperability of platforms. [1]

A comparison of state-of-the art popular motes is shown in Table 2.2.

Table 2.2 Comparison of Popular Motes

	MicaZ	**TelosB**	**Tiny Node**
Processor	AT Mega 128	MSP430	MSP430
Radio Chip	CC2420	CC2420	XE1205
Flash	512 KB	8 MB	4 MB
Sensors	No on-board	Humidity	Light
Frequency	2.4 GHz	2.4 GHz	868-870 MHz
Tx Power	0 dBm	0 dBm	0 to +12 dBm
Max Data Rate	250 Kbps	250 Kbps	152.3 Kbps
Max Range	Upto 70m	Upto 150m	200m @ 76.8
Power	28 mA	21 mA @ 0	25 mA @ 0
Interface	USB/ Serial	USB	Serial
Operating	TinyOS	TinyOS	TinyOS
Cost	125	130 (witout	180

As seen from the table, TelosB offers the best functionalities compared to the other motes.

Chapter 3

DESIGN CONSIDERATIONS

This chapter discusses various design considerations regarding PCB, antenna and matching network. Available possibilities along with their respective pros and cons have been compared. This chapter illustrates the selected choices of PCB parameters, hardware, antenna and matching network and the reasons why they have been selected.

3.1 Hardware Selection

Hardware components have been selected by keeping TelosB as the reference design. All the components are Restriction of Hazardous Substances (RoHS) complaint and lead free. Moreover all the components are Surface Mount Devices (SMD).

The main components are

- Microcontroller (MSP430F1611)
- Radio (CC2420)
- Flash memory (M25P80)
- Temperature Sensor (TMP36FSZ)

3.1.1 Microcontroller

Texas Instruments MSP430 F1611 microcontroller features 10kB of RAM, 48kB of flash, and 128B of information storage. This 16-bit RISC processor features extremely low active and sleep current consumption that permits mote to run for years on a single pair of AA batteries. The MSP430 has an internal digitally controlled oscillator (DCO) that may operate up to 8MHz. In addition to the DCO, the MSP430 has 8 external ADC ports and 8 internal ADC ports. The F1611 also includes a 2-port 12-bit DAC module, Supply Voltage Supervisor, and 3-port DMA controller. [4]

3.1.2 Radio

The CC2420 is an IEEE 802.15.4 compliant radio providing the PHY and some MAC functions. With sensitivity exceeding the IEEE 802.15.4 specification and low power operation, the CC2420 provides reliable wireless communication. The CC2420 is highly configurable for many applications with the default radio settings providing IEEE 802.15.4 compliance. The CC2420 is controlled by the TI MSP430 microcontroller through the SPI port and a series of digital I/O lines and interrupts. The CC2420 has programmable output power. The CC2420 provides a digital received signal strength indicator (RSSI) that may be read any time. [4]

3.1.3 Flash Memory

ST M25P80 40MHz serial code flash is used for external data and code storage. The flash holds 1024kB of data and is decomposed into 16 segments, each 64kB in size. The flash shares SPI communication lines with the CC2420 transceiver. [4]

3.1.4 Temperature Sensor

The TMP36 is a low voltage, precision centigrade analog temperature sensor. It provides a voltage output that is linearly proportional to the Celsius (Centigrade) temperature. It does not require any external calibration to provide typical accuracies of

±1°C at +25°C and ±2°C over the –40°C to +125°C temperature range. The low output impedance of the TMP36 and its linear output and precise calibration simplify interfacing to temperature control circuitry and A/D converters. [5]

3.2 Antenna Selection

There are two options for the antenna; an internal antenna designed onto the PCB and an external SMA connector for connecting to external antennas. If an application requires an external antenna or a different directional pattern than the internal antenna, an SMA connector may be installed and an antenna may be connected directly to motes' SMA female connector.

3.2.1 Internal Antenna

There are two recommendations for CC2420:

- $\lambda/2$ Dipole
- $\lambda/4$ Monopole

The length of the $\lambda/2$-dipole antenna is given by:

$$L = 14250 / f \qquad (1)$$

So an antenna for 2450 MHz should be 5.8 cm. Each arm is therefore 2.9 cm.

The length of the $\lambda/4$-monopole antenna is given by:

$$L = 7125 / f \qquad (2)$$

An antenna for 2450 MHz should be 2.9 cm.

In order to minimize overall size of the board, dimensions of the antenna had to be minimized. To achieve this, Inverted F Antenna (IFA) was chosen to be the on-board antenna.

3.2.1.1 Inverted F Antenna

It's a monopole antenna whose radiating arm is folded back to make it parallel with the ground plane and thus the length of the radiating arm decreases.

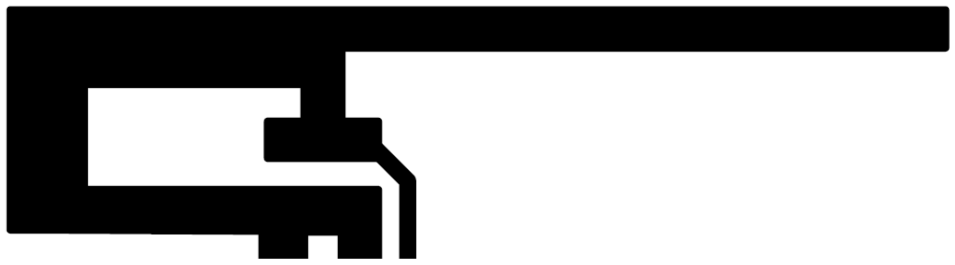


Figure 3.1 Inverted F Antenna (IFA)

It is one of the most commonly used antennas at 2.4 GHz and offers the following:

- Good omnidirectional radiation pattern
- No additional cost
- Ease of manufacture
- Compact size
- High efficiency
- Acceptable bandwidth

3.3 PCB Specifications

4 layered PCB is recommended for use with the radio chip (CC2420). Total board thickness is 62 mils (1.6 mm) and the dimensions of the board are 1.3” x 2.6”. 1 oz copper pour is used.

Other PCB specifications are as follows

- 4 Layer, Double sided, FR4 (Flame Retardant woven glass reinforced epoxy resin.)
- Layer Spacing:
  - 1-2: 8 mils
  - 2-3: 46 mils
  - 3-4: 8 mils

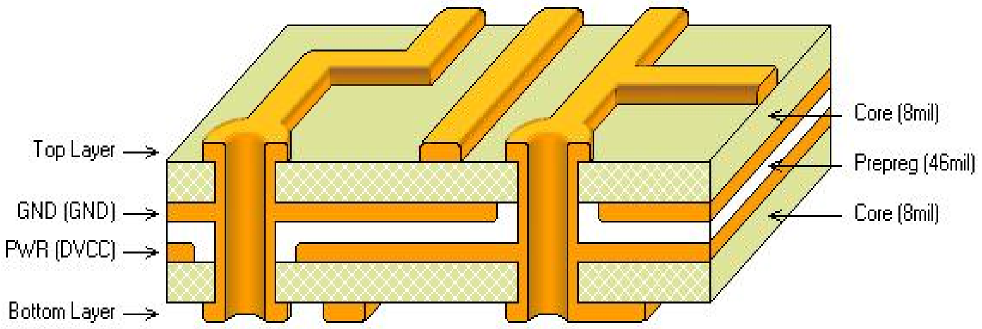


Figure 3.2 PCB Layer Stack

Among the four layers, two are the signal layers i.e. the top and the bottom layers. Whereas the two internal layers are the power plane and the ground plane.

3.4 Matching Network

The RF transceiver CC2420 has a differential output. The inverted F antenna that has been used as an internal antenna is a monopole i.e. a single ended antenna. Hence, there is a need to convert the differential output of the transceiver to a single ended signal during the transmission and vice versa during the reception. For this purpose, a balanced to unbalanced transformation has to be done using a balun circuit. The balun circuit is based on the manufacturer's recommendation which also includes the DC biasing of the RF power amplifier integrated in the CC2420 transceiver. The balun circuit and its integration are illustrated in Figure 3.3.

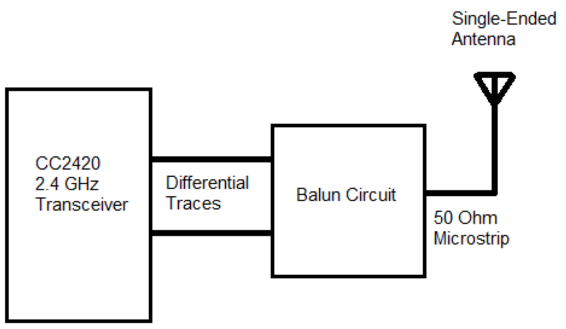


Figure 3.3 Balun Circuit Integration

3.4.1 Comparison of Different Balun Solutions

A multitude of possible balun implementations exist for CC2420. One of them uses a 180° transmission line and 4 discrete components. Another solution is a discrete lattice balun implementation that uses 7 components. [7] The differences and benefits that each of these solutions offers are mentioned below.

3.4.1.1 Transmission Line Implementation

The transmission line implementation employs only four discrete components. However it takes significant amount of space on the PCB and its performance is sensitive to changes of line width of the 180° transmission line, PCB thickness and variation in the PCB material. [7]

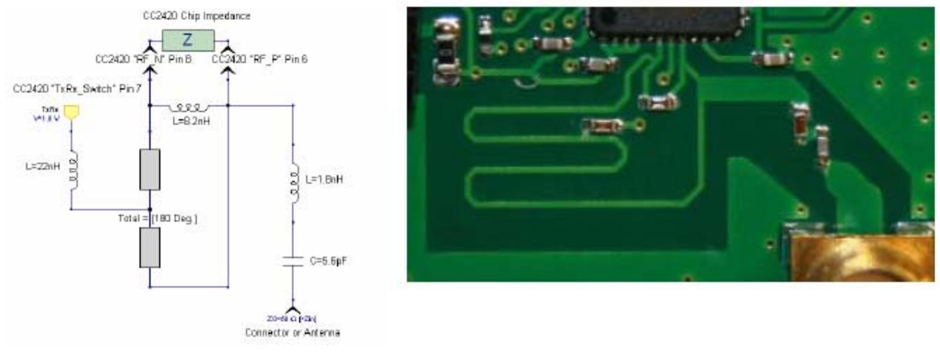


Figure 3.4 Schematic and Layout of Transmission Line Implementation

3.4.1.2 Lumped Element Implementation

The lumped element implementation is smaller in size than the transmission line version, but uses a total of 7 discrete components. This implementation is less susceptible to line width and PCB variations, but it is sensitive to discrete component tolerances. [7]

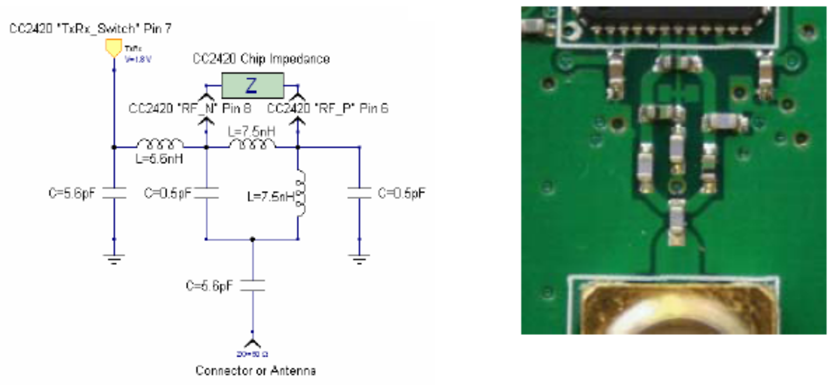


Figure 3.5: Schematics and Layout of the Lumped Element Implementation

To keep the size of the mote as small as possible, the lumped element design was chosen.

Chapter 4

DESIGN IMPLEMENTATION

After the hardware components and on-board antenna had been selected and the parameters of PCB defined, now was the time to actually implement the design. The implementation process consisted of PCB design, antenna simulation and optimization, and matching network simulation and optimization. This chapter details the entire implementation process along with the softwares used for each task.

4.1 PCB Design

The PCB designing is a time consuming process and consists of a number of steps whose detail is given in the following subsections.

4.1.1 Software Selection

The software used for PCB designing is Altium’s Protel DXP 2004. Protel DXP is an Electronic Design Automation (EDA) software. Schematics, PCB as well as footprints of components are designed and integrated via Protel.

Protel DXP is a software integration platform that provides the GUI for the various editors and viewers. It is a design compiler; file management, version control interface and scripting engine. It opens views and prints schematic documents. It has all the schematic document and object editing capabilities. It generates netlists. It performs pre-layout signal integrity analysis, includes full analysis engine and uses defaults for PCB parameters. It also performs post-layout signal integrity analysis with support for PCB routing analysis. It opens views and prints PCB documents. It can place / edit objects on mechanical layers, define design rules and layer stack, transfer design from schematic and position components. It can place / edit objects on electrical layers, create footprints, place from library, interactive routing, interactive / auto placement, modify netlist and generate Gerber, NC drill, ODB++ files. It also has the capability of autorouting PCB files.

4.1.2 Footprints Designing

Footprints for most of the components had to be designed as the footprints made available by the manufacturers were not compatible with Protel. Schematic symbols and land patterns of the components, not available in libraries of Protel, were designed either by component wizard or by modifying properties of the existing footprints and then integrating them to make integrated libraries. These integrated libraries were then installed to make use of these footprints.

4.1.3 Schematics Drawing

Schematics are drawn using multisheet design. Multisheet design allows design to be divided into modular elements. Primary reason for using multisheet design is the project size. Some projects are simply too large or complicated to fit on a single sheet. But even small designs can benefit from a multi-sheet approach. The designs may include various modular elements and dividing these into individual documents would allow several engineers to work on the project at the same time. Multi designs also allow using small format printing. When multi-sheet project is compiled, a structural framework of sheets is created. This is a tree structure, beginning with a top sheet and branching down to include all other sheets at one level or another. The sheet-to-sheet relationships are all defined by the sheet symbols that are placed in the design.

Design of mote was divided into three modules, namely

- Microcontroller side
- Radio side
- USB side

Net identifiers provide logical connections between sheets as well as within a single sheet. After drawing of all the three schematics, debugging was performed to make export of schematics to PCB possible.

4.1.5 Export of schematic to PCB

Once the project was compiled and all the errors in the schematic fixed, PCB Update command was used to generate ECOs that would transfer the schematic information to the target PCB. All the changes were validated. If the changes could not be validated, messages panel was checked for errors. Changes were executed. When completed, the target PCB opened with components positioned ready for placing on the board.

4.1.6 Components Arrangement

Once the schematics were exported to PCB, the next task was the arrangement of components. All the components were dispersed and not in proper order. Although wiring connections are intact but this was not enough to judge where to place them. Placing the right component at the right location, consulting the circuit again and again and doing this one by one, was a tedious task.

Ultimately, the component placement has the most significant impact on routing performance. Protel DXP includes a number of tools, such as dynamically optimized connection lines, that allows to fine tune the component placement. The optimal component placement is when the connection lines are as short and least ‘tangled’ as possible.

Components were arranged on the PCB so as to minimize the overall board size and such that routing is easier, keeping in view their net identifiers.

4.1.7 Routing

Routing is performed using ‘Autorouting’ feature of Protel DXP. It uses advanced topological mapping to first define the routing path, then calls on a variety of proven routing algorithms to convert path to a high quality route. It follows PCB electrical and routing rule definitions.

The DXP rules system is hierarchical. To start with a default rule for all objects, then add additional rules to selectively target other objects which have different requirements. The most important rules are the Width and Clearance rules. These routing technology settings define how tightly the routing can be ‘packed’. Selecting these is a balancing process, the wider the tracks and bigger the clearance, the easier it is to fabricate the board; versus the narrower the tracks and clearances, the easier it is to route the board. After consulting with the manufacturer and keeping in view their limitations of minimum track width, minimum via size and minimum clearance, 5 mils was chosen as the minimum track width and 12 mils as via diameter and 5 mils of clearance between tracks and vias. However Width of the track connecting the output of Radio to the onboard antenna was chosen to be 15 mils in order to provide 50 ohm of impedance at the input of the antenna. The width of the track is calculated from the equation:

$$w = \frac{7.463h}{\exp\left(\frac{Zo\sqrt{0.475\varepsilon_r + 0.67}}{60}\right)} - 1.25t\ cm \qquad (3)$$

Also ground plane was removed below the antenna and the matching circuit. A large number of grounded vias were placed around the path of radiation to minimize RF radiation due to the leakage of radiated power to other parts of the board.

After completion of autorouting, unnecessary vias were removed and the routing was improved, wherever it was possible. Some sections of the board were routed interactively.

4.1.8 Output File Generation

Once the design was completed, Gerber files were generated and sent to the manufacturer for PCB fabrication.

4.2 Antenna Design

4.2.1 Software Selection

Ansoft's HFSS v11 was used for simulation of the Inverted F Antenna. HFSS™ is the industry-standard simulation tool for 3D full-wave electromagnetic field simulation. HFSS provides E- and H-fields, currents, S-parameters and near and far radiated field results. It is an automated solution process where users are only required to specify geometry, material properties and the desired output. HFSS will then automatically generate an appropriate, efficient and accurate mesh for solving the problem using the proven finite element method.

4.2.2 Antenna Simulation

Antenna was designed using a reference document. [8] This document describes a PCB antenna design that can be used with all 2.4 GHz transceivers and transmitters from Texas Instruments. Antenna was simulated in HFSS and various parameters such as Reflection (S11), Gain and efficiency were measured for proper operation of the antenna. Once the antenna provided acceptable results, next step was its optimization.

4.2.3 Design Optimization

Optimization is achieved by variation of different antenna parameters such as

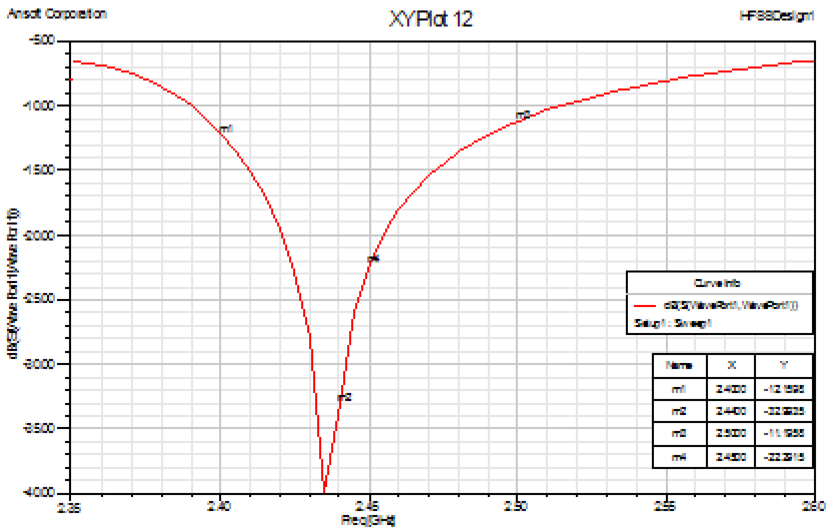
- Length and width of the feed arm
- Length and width of the radiating arm
- Separation between feed arm and shorting arm

Results after optimization are tabulated in table.

Table 4.1 Summary of the Properties of IFA

Antenna Dimensions	27.5*8.61 mm
Reflection (S11)	<-11 dB (8%)
Minimum Reflection	-40 dB (0.01%) @ 2.435 GHz
Gain	2.1444 dB
Bandwidth	125 MHz (5.1%)
Radiation efficiency	77.78%

The primary goal during optimization process was to decrease value of S11 parameter i.e. Reflection at the input. Its graph is shown in Figure 4.1.



Name	X	Y
m1	2.4000	-12.1588
m2	2.4400	-32.9239
m3	2.5000	-11.1958
m4	2.4500	-22.2815

Figure 4.1 Reflection at the Input (S11)

The secondary goal was to improve the gain and efficiency as much as possible. This is shown in Figure 4.2.

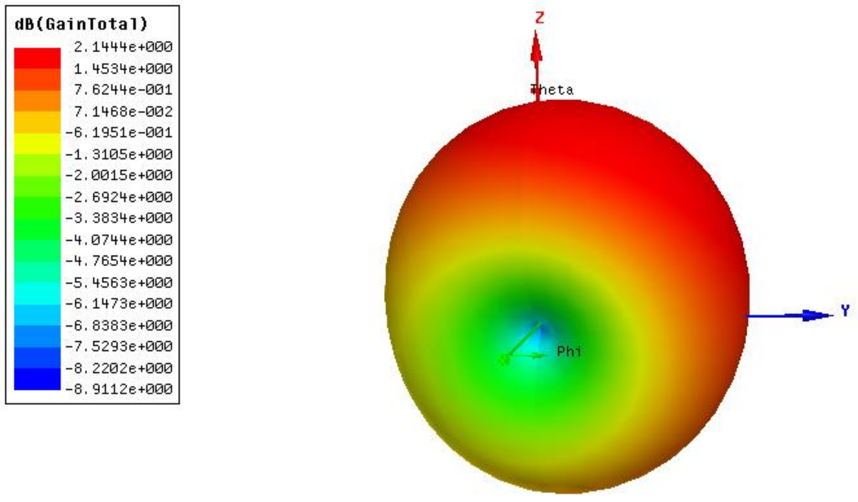


Figure 4.2 Gain in dB

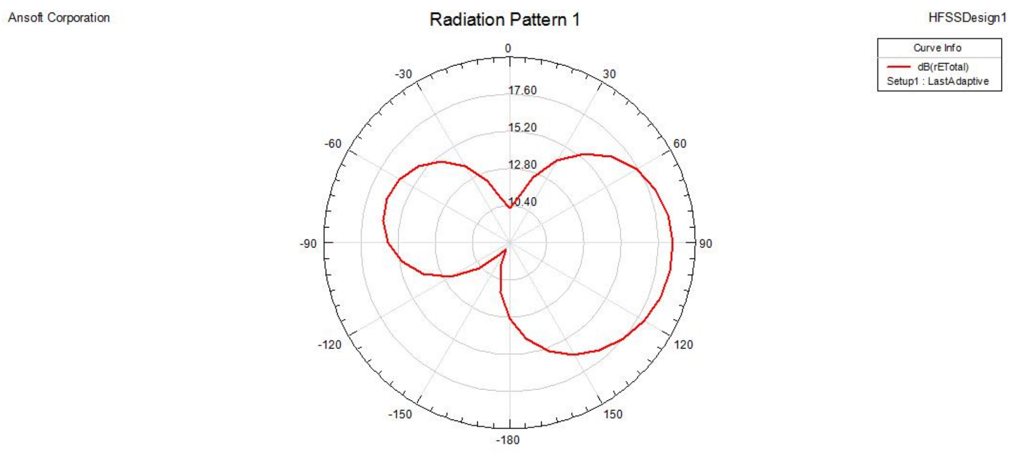


Figure 4.3 Radiation Pattern

4.3 Matching Network Performance Analysis

To keep the size of the mote as small as possible, lumped element design was preferred for matching network. Before the actual implementation of the balun design on the PCB, it was deemed appropriate to simulate it in some RF circuit simulator. For this purpose, Agilent’s Advanced Design System was used due to its high accuracy and wide industrial usage. The results for the values of discrete components, as recommended by the manufacturer, were not desirable in terms of reflection and transmission coefficients. This degradation in performance was mainly due to the difference in the PCB thickness and layer stack as compared to those of the manufacturer’s implementation. Because of the PCB manufacturer’s constraint, 62 mils PCB thickness was used in our design as compared to 1 mm (~40 mils) thickness used in the CC2420 reference design. Also, in the reference design of CC2420, layer 1 for routing, layer 3 for power and layer 4 for ground

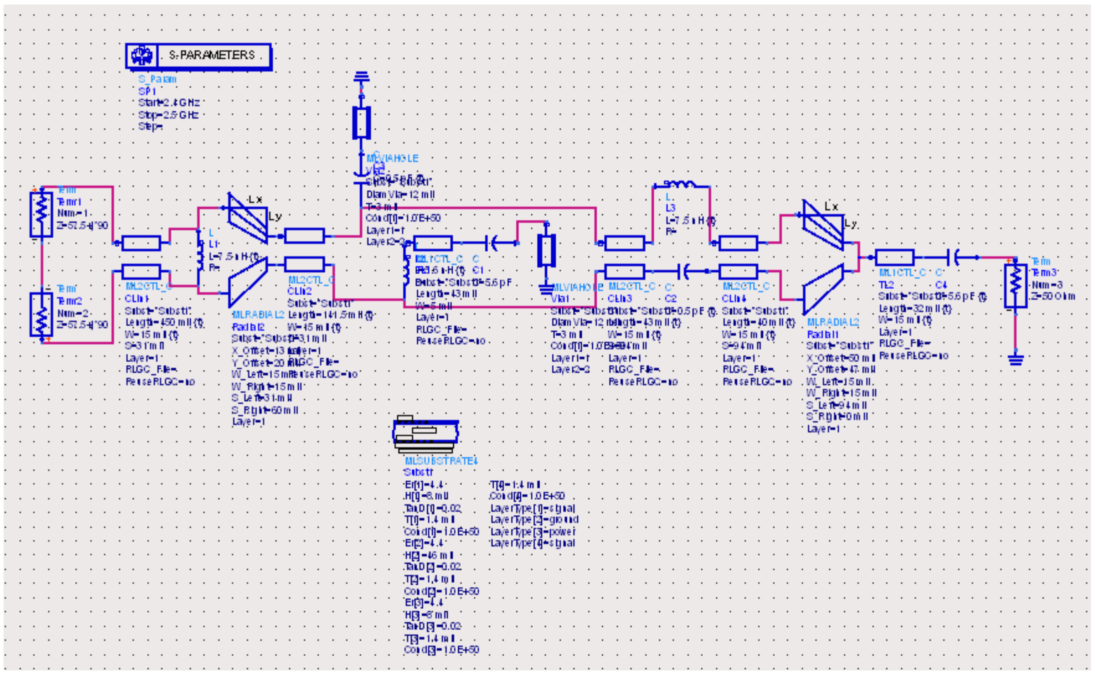
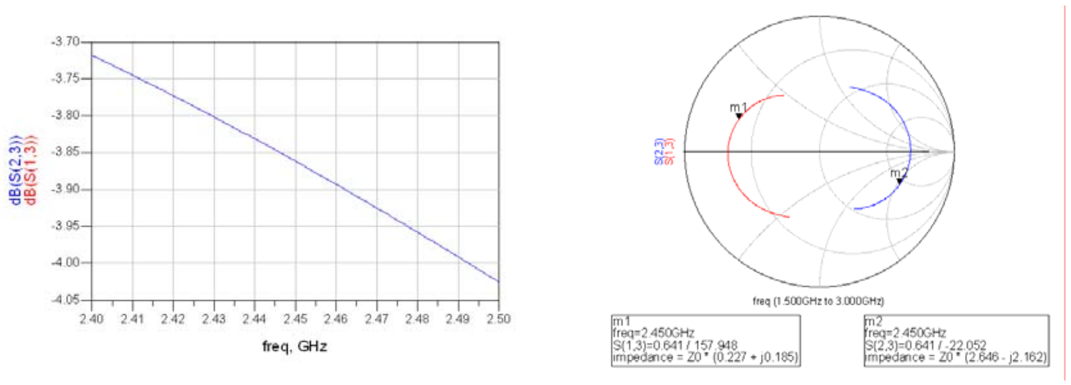


Figure 4.4 ADS Model of the Balun Circuit

and some routing have been used. Layer 2 has not been used. While the design that we have followed, uses all four layers. Therefore changes in the values of discrete components or traces' dimensions were necessary to achieve the desired results. The later was not possible as the PCB had already been sent for

fabrication at that point in time. So the only option was to change the component values. The balun implementation and the results achieved are shown in the following figures. In these figures, port 1 and 2 are the differential ports of the transceiver with differential impedance of 115+j180 ohm and port 3 is the single ended 50 ohm antenna port. The effects of the PCB and traces have also been incorporated in this design. With only a few changes, a transmission coefficient of -3.862 dB was achieved. As can be seen in the figure, the curves of S13 and S23 match exactly (i.e. equal loss in both arms of the balun) and there is a 180 degrees phase shift between them. The values of reflection coefficients achieved are -9.561 dB for S11 and S22 and -8.746 dB for S33. Keeping in mind the



```
m1
freq=2.450GHz
S(1,3)=0.641 / 157.948
impedance = Z0 * (0.227 + j0.185)
```

```
m2
freq=2.450GHz
S(2,3)=0.641 / -22.052
impedance = Z0 * (2.646 - j2.162)
```

Figure 4.5 S13 and S23 (Transmission Coefficients)

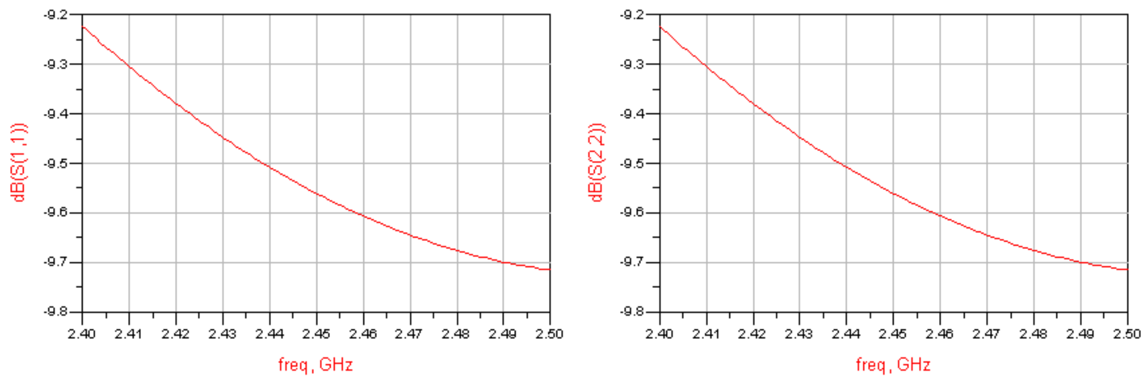


Figure 4.6 S11 and S22 (Reflection Coefficients of Differential Ports)

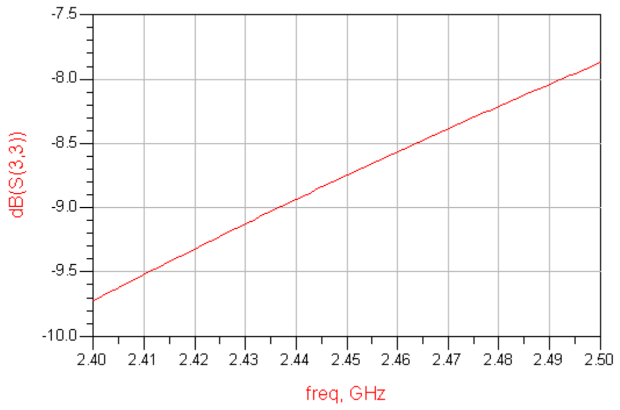


Figure 4.7 S33 (Reflection Coefficient of Single Ended Antenna)

constraint that we could only change the component values and not the trace dimensions, the results achieved are satisfactory.

Chapter 5

PERFORMANCE ANALYSIS

Once the hardware node is ready, now is the time to test it. There are a number of components and parameters that need to be tested but that can only be done with the help of software integration. This chapter describes the details of the integration of software with the hardware, testing of different components for their proper functioning and subsequently the process of taking measurements to compare the performance of our mote with that of the reference design.

5.1 TinyOS

TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS applications are written in nesC which is a dialect of the C programming language. The supplementary tools are mainly in the form of Java and shell script front-ends. Associated libraries and tools such as the nesC compiler etc are mostly written in C.

TinyOS programs are built out of components, some of which present hardware abstractions. Components are connected to each other using interfaces. TinyOS provides interfaces and components for common abstractions such as packet communication, sensing and storage etc.

TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools, all of which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model enables fine-grained power management yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces.[9]

5.1.1 Hardware – Software Integration

Though the provision of a USB interface increases the size of a mote but the advantage of such provision becomes evident when installing an application on the mote to make it functional. There exist a number of methods to install an application on the mote but the method we used was straight forward and is described in appendix C.

5.2 Hardware Testing

A number of components provided on the mote need to be tested to ensure the proper functioning of the mote. These include microcontroller’s timer and ADC, radio transceiver, LEDs, flash memory and on-board sensor. Different simple applications are already available to test these components and are integrated in the TinyOS library. However, the temperature sensor provided on our mote is different from the one available on the reference mote. Therefore, the driver for this temperature sensor was implemented in nesC.

Unfortunately, at the time of writing of this report, the mote still is in the final stage of development, so the results of the testing phase and any debugging, if needed, can not be included in this report. However, we will be able to present them during the final presentations.

5.3 Measurements

The final phase of the development of the mote is the comparison of its performance with that of the reference design. The most important parameters against which to compare the two motes include size, range, power consumption and cost. Dimensions of our mote are same as those of the reference mote. The detail of the remaining parameters is given in the following paragraphs.

5.3.1 Radio Range Measurement

The range measurement comprises of two components, the outdoor range and the indoor range. The reference mote offers 70 to 100 meters and 20 to 30 meters ranges in outdoor and indoor environments respectively. [10] Once our mote is ready, we will perform the range measurements in both environments. The method of this measurement is easy. One mote connected with the computer will act as receiver and another mote will be used for transmission. The RSSI value of the packets received from the transmitting mote will be noted and the distance between the two motes will be increased gradually. At the threshold value of RSSI, the distance between the two motes will be considered as the range of the mote. The results obtained should be comparable to those mentioned above.

5.3.2 Power Consumption Measurement

Power consumed by the mote gives an estimate of its life time. The more power a mote consumes, the earlier its battery will drain and it will become useless if

Table 5.1 Power Consumption

Parameter	Minimum Value	Nominal Value	Maximum Value	Unit
Supply Voltage	2.1		3.6	V
Current Consumption: MCU on, Radio RX		21.8	23	mA
Current Consumption: MCU on, Radio TX		19.5	21	mA
Current Consumption: MCU on, Radio off		1.8	2.4	mA
Current Consumption: MCU idle, Radio off		54.5	1200	uA
Current Consumption: MCU standby		5.1	21	uA

deployed in an inaccessible environment or the battery will need to be replaced otherwise. Thus, minimizing the power consumption leads to a better design and should always be opted. The current consumption values of reference mote are given in Table

5.1. [10] Our mote will undergo the same measurement process and the results achieved should be comparable to those of the reference mote. An ammeter will be connected in series with the battery and current drawn will be noted down for different states of the MCU and radio.

5.3.3 Cost Analysis

One of the primary aims of our work was the development of a low cost mote. In this section, a comparison of costs has been carried out. The motes are not available in Pakistan and their procurement, including the shipping and handling charges, can cost up to Rs. 25,000. However, we were able to develop the same mote in Rs. 10,000 yielding a 60% reduction in cost. The cost can further reduce if the motes are manufactured in large quantities.

Table 5.2 Cost Breakdown

Description	Cost
Components	Rs. 8700
PCB Manufacturing	Rs. 1300
Total	Rs. 10,000

Apart from the huge cost, the lead time for the procurement of motes is three to four weeks and also is a major problem for the colleges, universities and research institutes. Therefore, we argue in favor of the development of such motes in Pakistan which will be available easily and at very low price.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The idea of using small, low cost wireless devices for sensing has created a paradigm shift in contact-less, non-intrusive, distributed sensing. Initial experimental deployments of such networks using some of the commercially available sensor nodes are helping people understand the benefits that such networks can offer. The continued growth and evolution of such an experimental field into a mature technology can only occur if people in research and educational institutes have cheap, easy and ready access to suitable hardware platforms. However, state of the art sensor nodes available in the market cost a few thousand rupees and have to be imported. [1]

The high cost coupled with all the administrative hassles required to import these devices, makes access to them difficult for most educational and research institutes in Pakistan.

In this thesis we have designed and implemented a hardware platform taking telosB as the reference design that offers all the capabilities provided by commercially available sensor nodes. This will be tested for communication ranges and power consumption. The device is powered by two AA size batteries. Our design is a research prototype and one would expect the cost of a research prototype to be higher than that of the commercially available motes. The total cost of the mote as per industry standards can be computed to be around Rs. 10,000. This cost is lower than the cost of a TelosB. Hence it becomes a viable option to fabricate these boards for the advantages offered by such a development.

It is expected that the easy availability of such a platform at a fraction of the cost of equivalent commercially available platforms will allow increased participation by research and educational communities in the evolving field of wireless sensor networks and develop applications.

In this work, we have successfully designed and will implement the core functionalities of a mote by proving the operation of microcontroller and radio circuitry.

6.2 Future Scope

This is the first hardware version of the mote and the experience gained has shown a path for design of better hardware/increasing the flexibility of the mote.

A few of the scenarios that may be considered for future design are as under.

• The design may further be improved through usage of new generation microcontrollers that integrate radio and microcontroller on a single chip. This will enable reduction of the board size as the circuitry for interfacing radio with microcontroller will no longer be needed. This will also free the SPI lines of microcontroller and facilitate integration of additional flash/other high speed peripherals.

• Explore the feasibility of using a 32 bit low power microcontroller such as those from Atmel AVR32 UC3 processor family. This would increase the processing capability of the motes and they can be designed for complex applications.

• Explore the possibility of porting a tiny version of Linux on to the on-board flash. Porting an OS such as Linux would increase the flexibility of the mote.

• A design that includes an RF power amplifier at the output stage of the radio chip seems to be an interesting possibility to further increase the range of the mote using the on-board antenna.

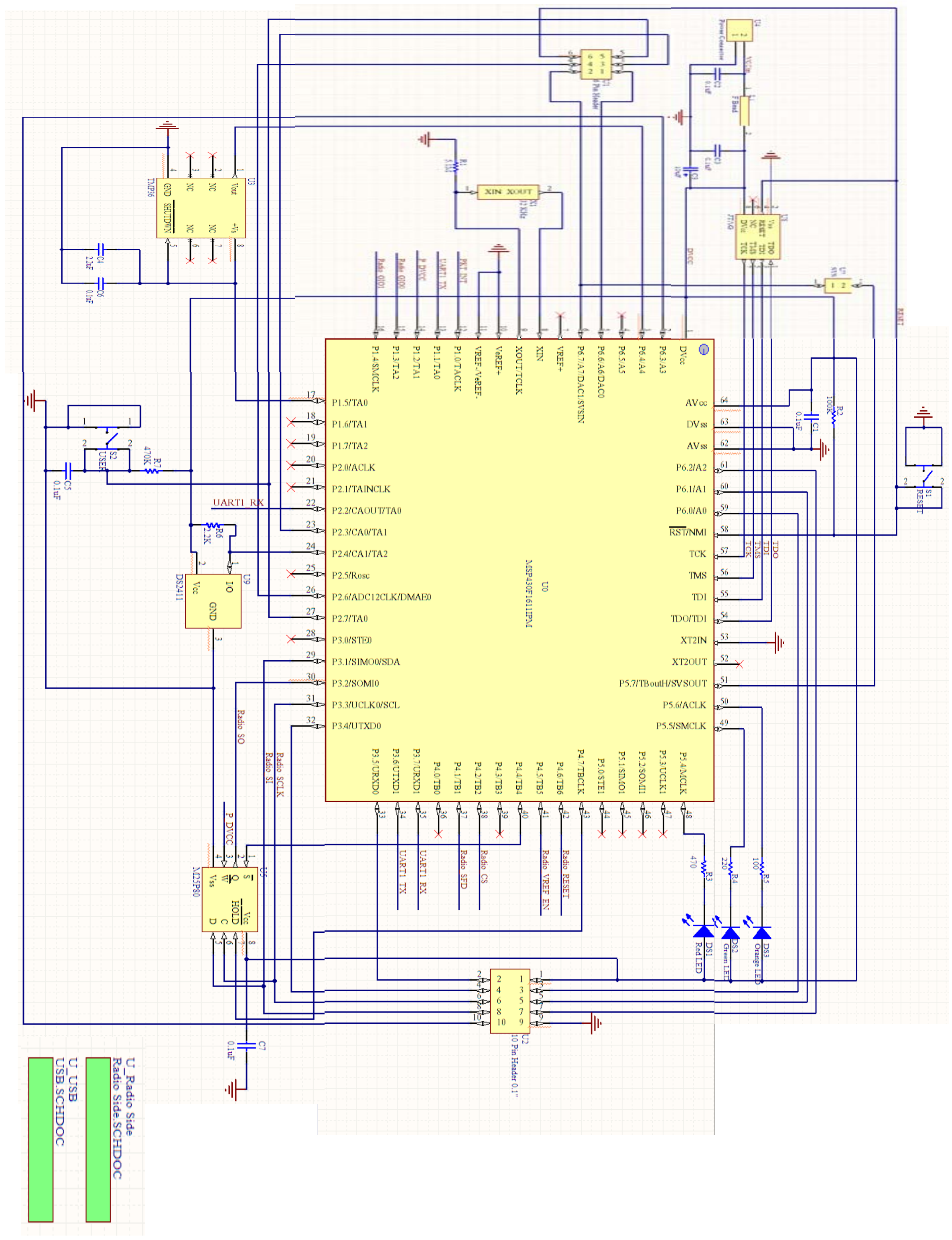
• Design of a directional antenna on the PCB can be considered that increases the range of the mote. The motes can then be used in point to point long distance links. [1]

Appendices

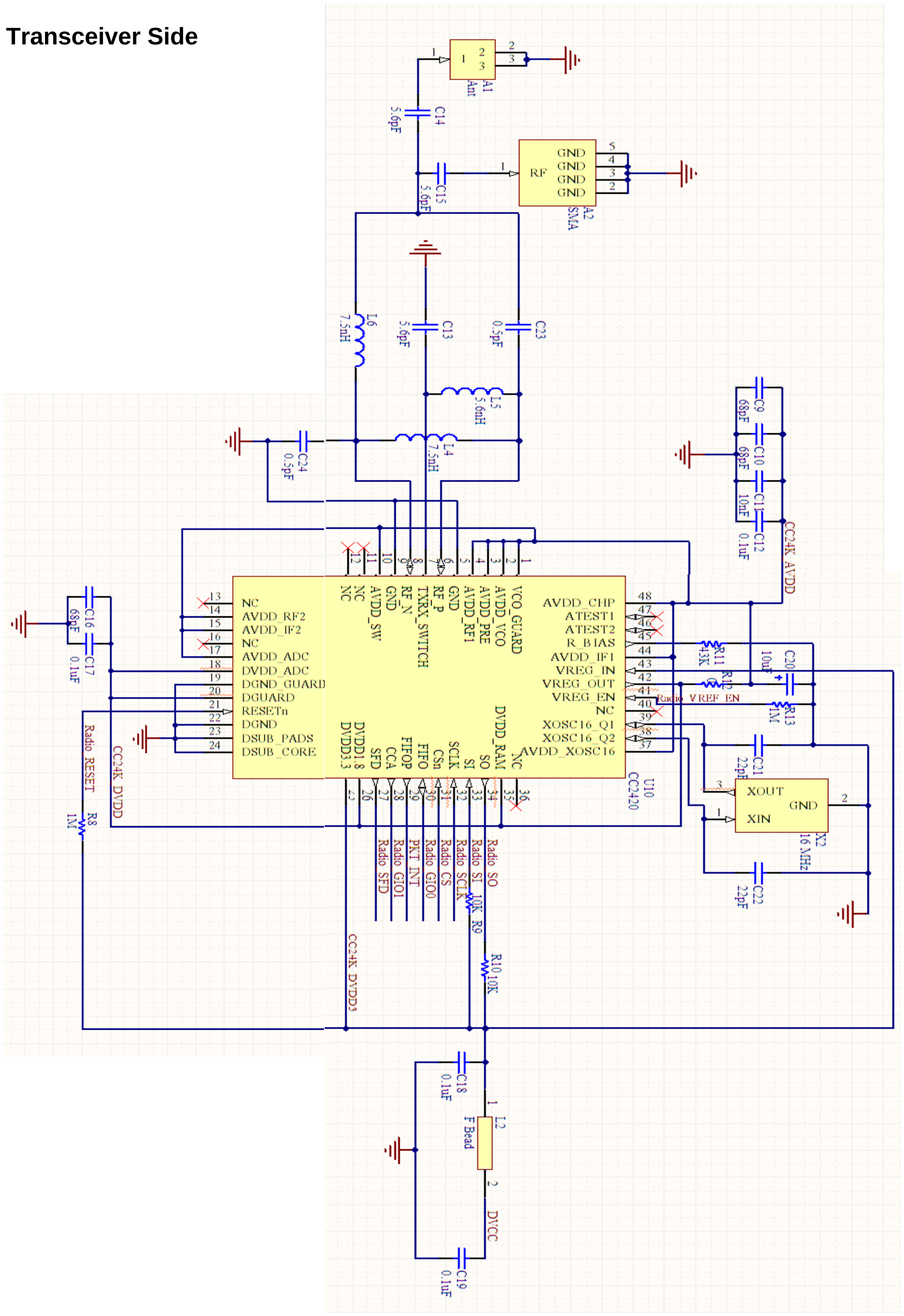
Appendix A

Schematic Diagrams

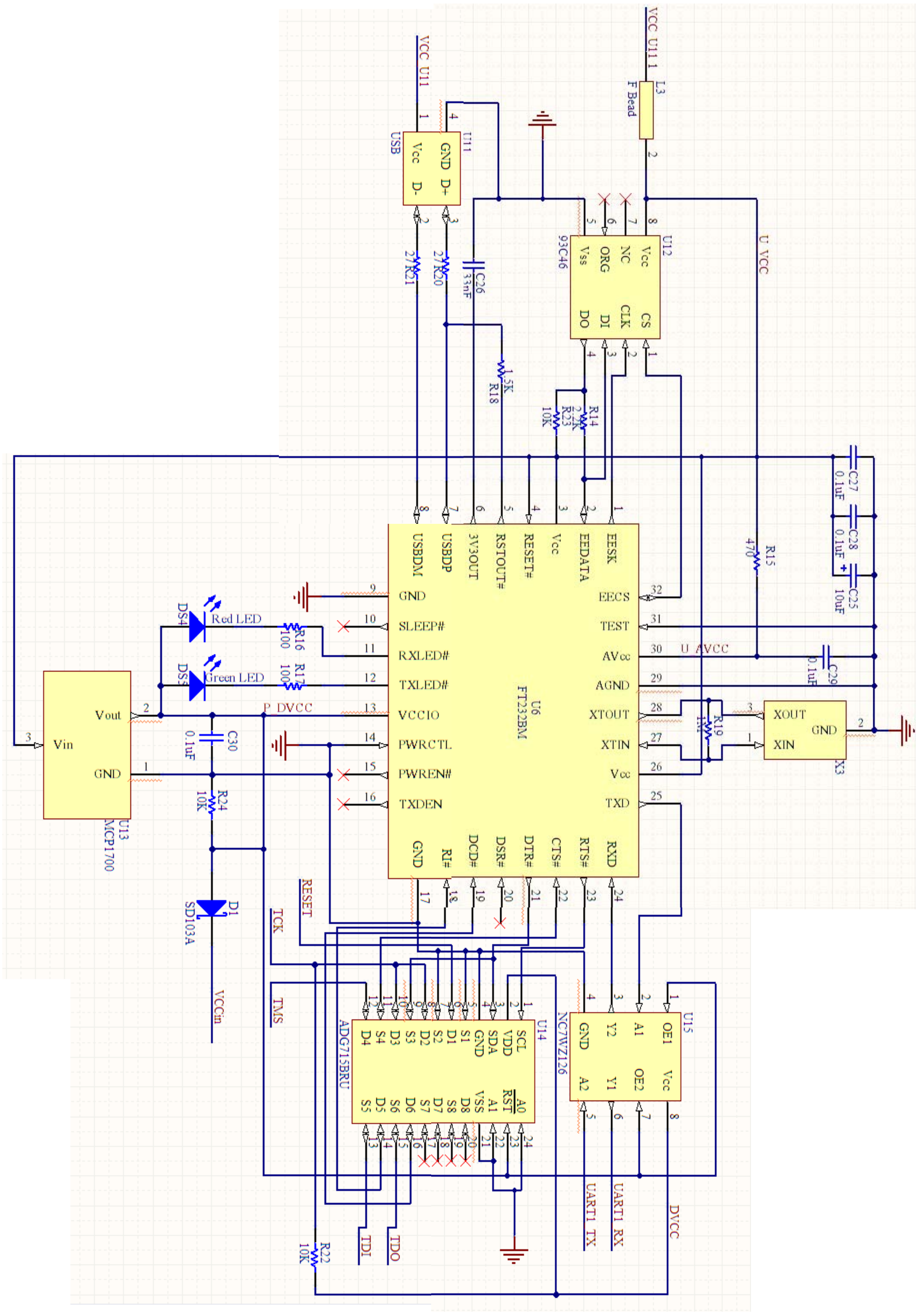
Microcontroller Side



Transceiver Side



USB Side

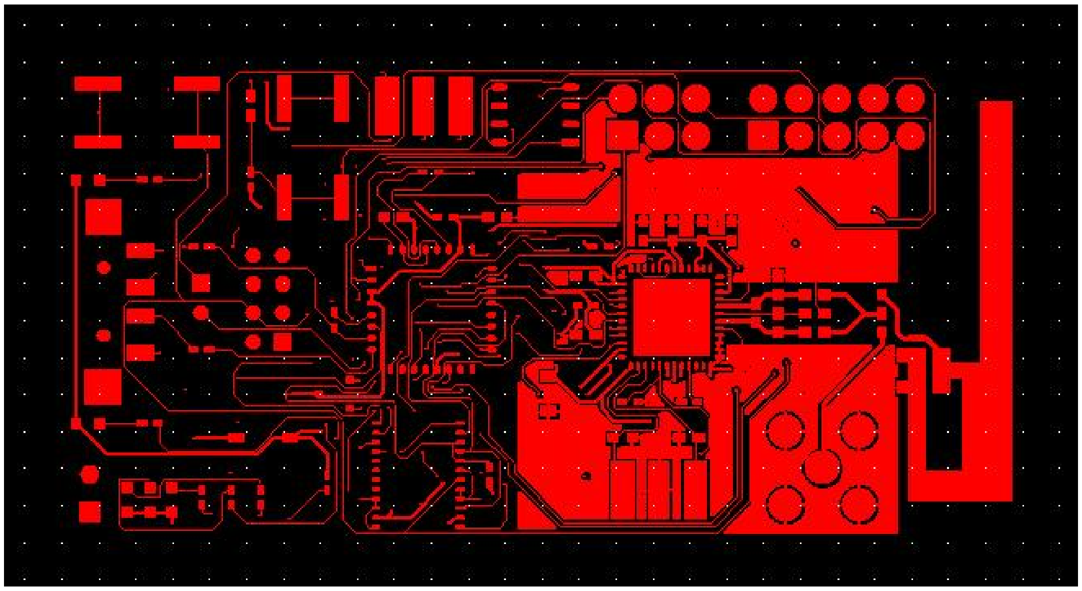


VCC U11
USB
U11
Vcc
GND
D+
D-
VCC U11 1
L3
F Bead
U12
93C46
Vcc
NC
ORG
Vss
CS
CLK
DI
DO
U VCC
C26 33nF
R20 27
R21 27
R18 1.5K
R14 2.2K
R23 10K
R15 470
C27 0.1uF
C28 0.1uF
C25 10uF
C29 0.1uF
U6
FT232BM
USBDM
USBDP
3V3OUT
RSTOUT#
RESET#
Vcc
EEDATA
EESK
GND
SLEEP#
RXLED#
TXLED#
VCCIO
PWRCTL
PWREN#
TXDEN
EECS
TEST
AVcc
AGND
XTOUT
XTIN
TXD
RXD
RTS#
CTS#
DTR#
DSR#
DCD#
RI#
U AVCC
P DVCC
R19 1M
X3
XOUT
XIN
GND
DS4 Red LED
DS3 Green LED
R16 100
R17 100
C30 0.1uF
R24 10K
U13
MCP1700
Vin
Vout
D1
SD103A
VCCin
RESET
TCK
TMS
TDI
TDO
R22 10K
U14
ADG715BRU
SCL
VDD
SDA
GND
S1
D1
S2
D2
S3
D3
S4
D4
A0
RST
A1
VSS
S8
D8
S7
D7
S6
D6
S5
D5
U15
NC7WZ126
OE1
A1
Y2
OE2
Y1
A2
UART1 TX
UART1 RX
DVCC

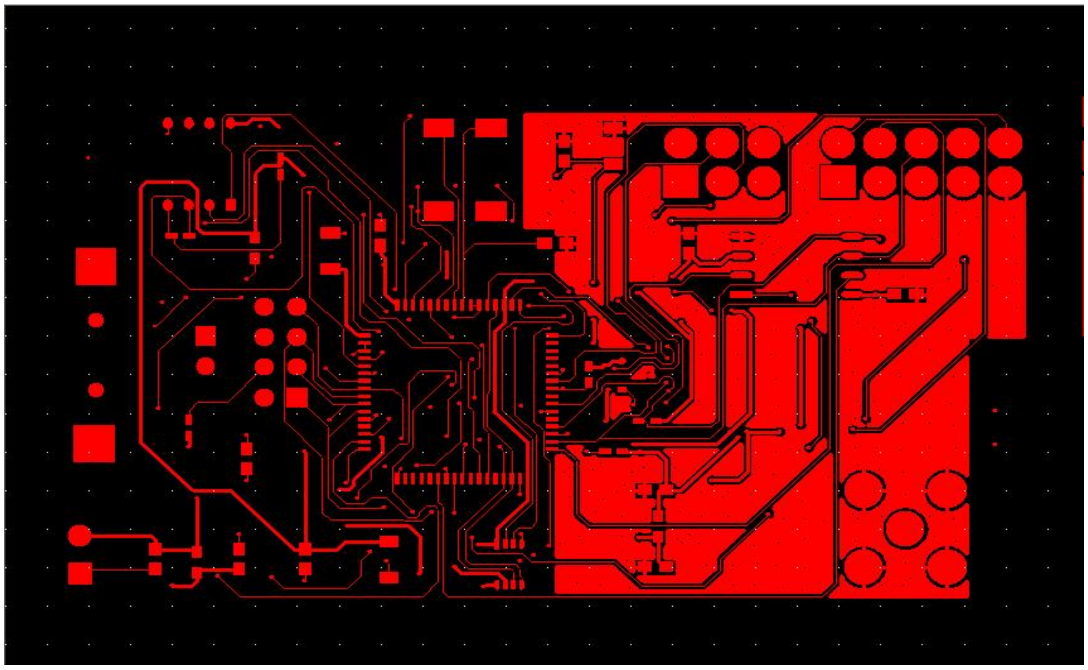
Appendix B

PCB Layout

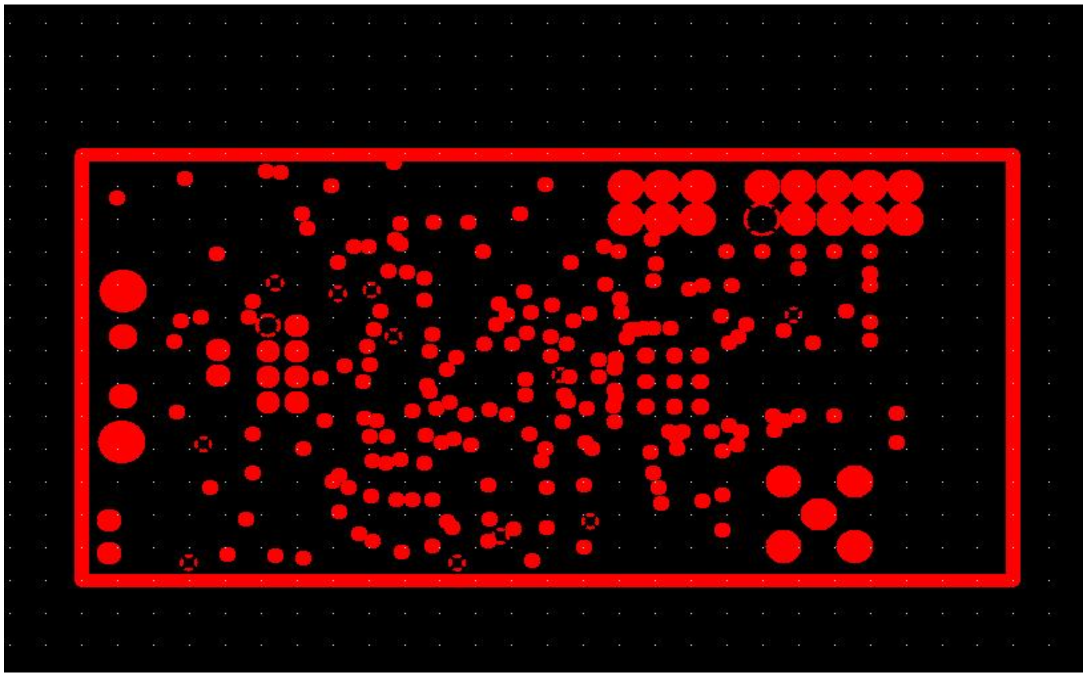
Top Layer



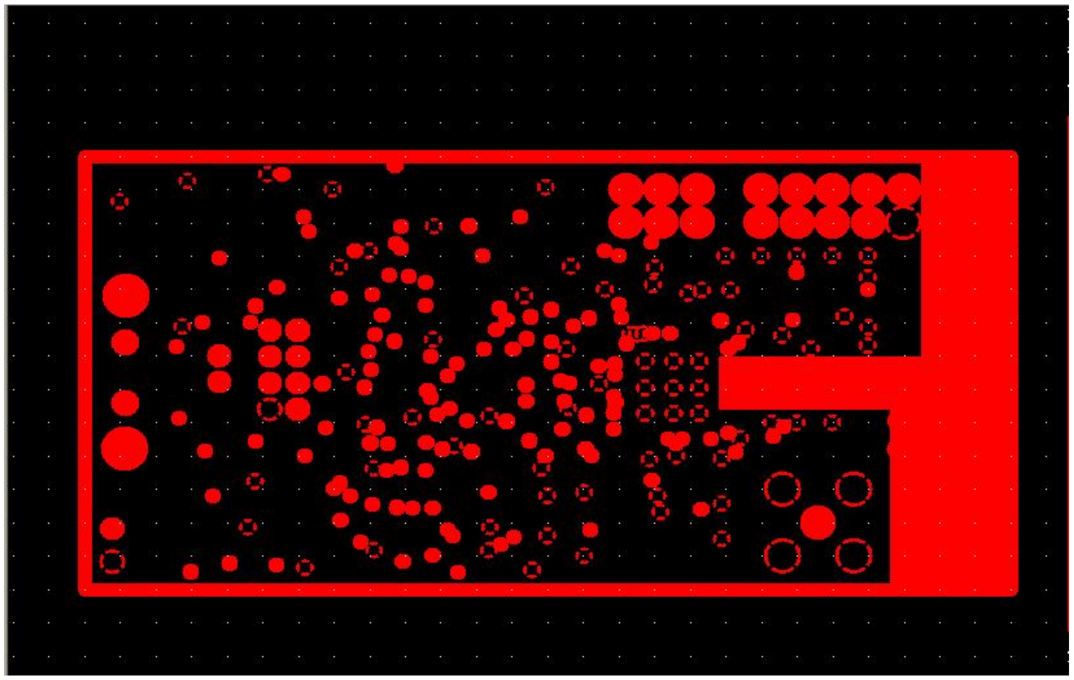
Bottom Layer



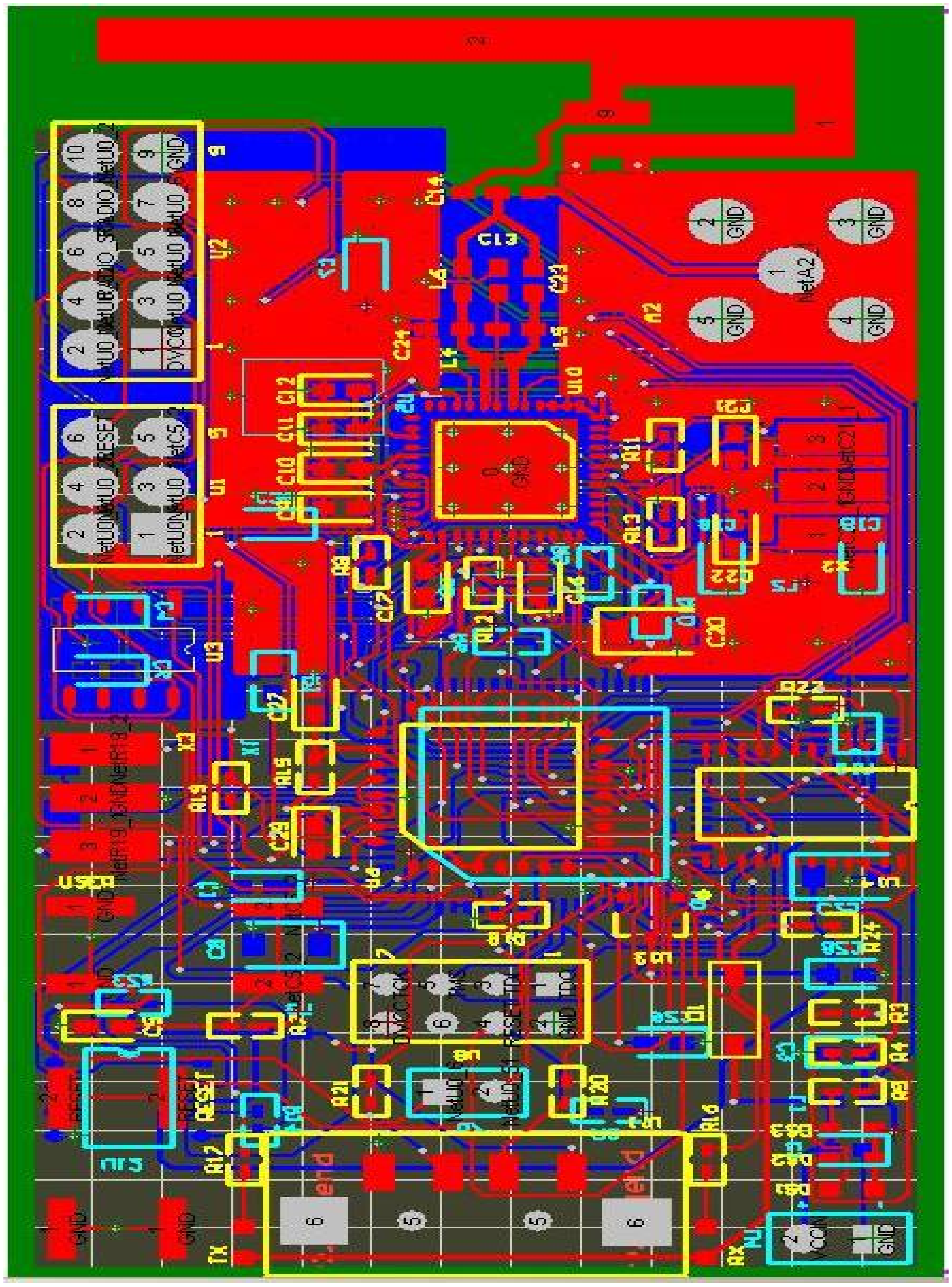
Power Plane



Ground Plane



Complete PCB Layout



Appendix C

Process of Installing an Application on the Mote

Follow these steps to get a TinyOS application installed on the mote. This procedure is same as given in [7].

- Install VMware Player on a computer. This software provides a virtual machine environment where multiple operating systems can be used without actually installing them. This is necessary to use the TinyOS.
- You must have XubunTOS on your computer. This is the actual operating system and provides a number of applications along with the component drivers libraries and compiler necessary to compile an application. The version we used was XubunTOS 2.1 which is the latest version available to date.
- You must have FTDI USB drivers installed on your machine because the USB controller used on mote is manufactured by FTDI. These drivers can be downloaded from the FTDI website.
- Now, simply plug in the mote in a USB port of the computer. Once the VMware Player is running, it will detect the mote.
- Open a Cygwin shell and type **motelist.** This command will find the COM port the mote is on.
- Change your directory (inside the *Cygwin* window) to the directory of the application which you want to install on the mote.
- Compile **make telosb**

- In the *Cygwin* window type: **make telosb reinstall,<*node_ID*> bsl,<*N*>** where **<*node_ID*>** sets the node address (optional) and **<*N*>** is an integer one less than the COM port number assigned to the mote. For example if the mote is assigned COM port 11, then the value of **<*N*>** would be 10.

Appendix D

nesC Code for Temperature Sensor

Since the temperature sensor used on our mote is different from the one in the reference design, therefore we have developed driver for it. The nesC code for the configuration and implementation is provided here.

TMP36C.nc

```
/**
 * TMP36C.nc is the configuration file for the temperature sensor.
**/
generic configuration TMP36C() {
  provides interface DeviceMetadata;
  provides interface Read<uint16_t>;
  provides interface ReadStream<uint16_t>;
}
implementation {
  components new AdcReadClientC();
  Read = AdcReadClientC;

  components new AdcReadStreamClientC();
  ReadStream = AdcReadStreamClientC;

  components TMP36P;
  DeviceMetadata = TMP36P;
  AdcReadClientC.AdcConfigure -> TMP36P;
  AdcReadStreamClientC.AdcConfigure -> TMP36P;
}
```

TMP36P.nc

```
/**
 * TMP36P.nc is the implementation file for the temperature sensor.
**/

#include "Msp430Adc12.h"

module TMP36P {
  provides interface DeviceMetadata;
  provides interface AdcConfigure<const msp430adc12_channel_config_t*>;
}
implementation {

  msp430adc12_channel_config_t config = {
    inch: INPUT_CHANNEL_A4,
    sref: REFERENCE_VREFplus_AVss,
    ref2_5v: REFVOLT_LEVEL_1_5,
    adc12ssel: SHT_SOURCE_ACLK,
    adc12div: SHT_CLOCK_DIV_1,
    sht: SAMPLE_HOLD_4_CYCLES,
    sampcon_ssel: SAMPCON_SOURCE_SMCLK,
    sampcon_id: SAMPCON_CLOCK_DIV_1
  };

  command uint8_t DeviceMetadata.getSignificantBits() { return 12; }
  async command const msp430adc12_channel_config_t* AdcConfigure.getConfiguration() {
    return &config;
  }
}
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] KMote - Design and Implementation of a Low Cost, Low Power Hardware Platform for Wireless Sensor Networks by Naveen Madabhushi

[2] http://en.wikipedia.org/wiki/Wireless_sensor_network

[3] http://en.wikipedia.org/wiki/Sensor_node

[4] http://uan.no-ip.com:8080/~ahamlett/uploaded/telosMote.pdf (TelosB datasheet)

[5] Datasheet TMP36FSZ

[6] Datasheet CC2420

[7] Anaren 0404 (BD2425N50200A00) balun optimized for Texas Instruments CC2420 Transceiver by Nithya R Subramanian and Niels Kirkeby, August 31st, 2007

[8] Texas Instruments' Design Note 0007 by Audun Andersen

[9] http://www.tinyos.net/

[10] Crossbow TelosB

[11] TPR2400/2420 Quick Start Guide, Rev. A, May 2005, Document 7430-0380-01