The Report Committee for Matthew John Tilleman
Certifies that this is the approved version of the following report:

Empirical Measurements on a Wireless Sensor Network

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Adnan Aziz

Mark McDermott

Empirical Measurements on a Wireless Sensor Network

by

Matthew John Tilleman, B.S.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2010

Abstract

Empirical Measurements on a Wireless Sensor Network

Matthew John Tilleman, M.S.E.

The University of Texas at Austin, 2010

Supervisor: Adnan Aziz

My project was to develop a hardware and software platform consisting of client nodes and a base station interconnected wirelessly. The nodes collect physical data for their local environment - I implemented a temperature measurement and a battery level reading. These measurements were placed in a packet which was then relayed via other nodes to the base station. The base station is attached to a USB dongle to a computer which collects the data and stores it into a log file for later analysis.

In designing such a network, my goal was to learn about routing protocols, take key concepts learned in classes, such as different modulation schemes and the study of wireless degradation in various environments due to reflections and interference, and explore an implementation of a commercial wireless system. Such a system could be modified to fit a multitude of applications such as environmental data collection for

farmers, low power networks for data communication for disaster recovery teams, or sensor networks or implemented in a house to collect data over long period and analyze variances in different regions and implement automated control through a feedback loop.

To implement my code, I used TI's EZ430-RF2500. This development kit contains the TI MSP430F2274, a 16MHz, 16 bit RISC processor which in active mode only pulls 270μA. The MSP430F2274 is coupled with a TI CC2500 which is a 2.4GHz RF transceiver used to communicate with the other devices. The EZ430-RF2500 connects to the computer via a USB dongle with proprietary firmware loaded which allows for programming and serial communication with the computer.

I built a network using three devices; one connected to a laptop acting as the access point and two remote devices powered by two AAA batteries acting as the end devices or clients. I performed a study of packet success rates in different environments, specifically inside a residential home, outside in a residential neighborhood and in a rural area. In close ranges (distances less than 50') there were no noticeable differences in performance between the three environments. I could not exceed 50' inside the residential environment due to the size of the tested house. Beyond 50' in the two outside environments, the results surprisingly did not differ greatly; successful transmissions were accomplished at distances only 10' further in Town Lake; that is that successful transmissions were capable up to 95' at Town Lake and 85' in my uban neighborhood. As a representative finding, in the urban environment, the clients were successfully transmitting at an 80% success rate at 80' pulling 84.48mW (26.4mA at 3.2V) while transmitting with 2-FSK.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 MOTIVATION

There are many applications for low power sensor networks. Such a system could be modified to fit a multitude of applications such as for farmers collecting environmental data, for disaster recovery teams which need low power networks for data communication and sensor networks implemented in a house to collect data over long periods which is then analysed for automated control through a feedback loop to control various electronics in the house.

My project was to develop a hardware and software platform consisting of client nodes and a base station interconnected wirelessly. The nodes collect physical data for their local environment - I implemented a temperature measurement and a battery level reading. These measurements were placed in a packet which was then relayed via other nodes to the base station. The base station is attached to a USB dongle to a computer which collects the data and stores it into a log file for later analysis.

In designing such a network, my goal was to learn about routing protocols, take key concepts learned in classes, such as different modulation schemes and the study of wireless degradation in various environments due to reflections and interference, and explore an implementation of a commercial wireless system. Such a system could be modified to fit a multitude of applications such as environmental data collection for farmers, low power networks for data communication for disaster recovery teams, or sensor networks or implemented in a house to collect data over long period and analyze variances in different regions and implement automated control through a feedback loop.

1.2 USER STORIES

We describe applications of our work by user stories.

- "Chad is a new home owner. He would like to be able to collect data around his house to analyze fluctuations in temperature versus light in different rooms at various points throughout the day."

- "Bob is the owner of a farm with a small staff. He would like a low power device network of inexpensive sensors to collect data on his farm to track temperature,

humidity, pH levels and other parameters around his farm to know when action needs to be taken with minimal human interaction."

- "James works for the National Guard doing disaster recovery missions. James would like an inexpensive device network which, while configurable through code, collects data at a low interval, going to sleep between intervals to save power to allow them to be place and forget but collect data about the environment such as water levels, bacteria levels and low resolution cameras."

- "Lindsey is a doctor who analyzes bacteria and other organisms. Lindsey would like a low power network of sensors which are easy to setup and can run for a long duration to collect data on her organisms."

1.3 RELATION TO PRIOR WORK

The original idea for this project was developed in a brain storming session with Dr. Adnan Aziz as an extension of a project he advised in 2009 from Andy Hocker called EtherLux [6]. EtherLux was based on broadcast – a server sent updates wirelessly at a low transmission rate to update an organic LCD displaying a status message. It was intended that this device could be implemented in a warehouse or large store but the ability for these low power devices to transmit at the needed distances became unreasonable. This project extends on EtherLux by allowing the clients to transmit aa well as receive and adds sensors to allow messages or other feedback mechanism.

"Simulation and Measurement of Bit Error Rates for a 2FSK-System in Indoor Environments" by Zwick, Demmerle, and Wiesbeck [8], investigates a new method to predict the bit error rate with a 2-FSK modulation network. They used a 2.45GHz wireless transmission frequency which correlates to my 2.4GHz signal. Their measurements were taken at 7.5m (24.6'). With a receive power of -40dBm they predict a BER of 10^{-6} but with a receive power of -45dBm, the BER rate increases to 0.01. The rate at which the BER increases with decreases in power supports the need for extending the network via routing for low power networks to preserve the original data when clients are needed a great distances from the base station.

In "An Improved Low Power Wireless Sensor Network Based on ZigBee for Agricultural Applications.", Tsiakmakis, Mallios, Charalampidis, and Spasos [9] implement a low power, wireless sensor network. While they do not talk on any detail about the implementation in hardware, they do provide their results. They were able to

achieve 1-link distances of 100m (328‘) and 170m (557.7’) by using a single hop. They implemented their network with handshaking and using the TI CC2480 which is in the same line as the CC2500 I used but is a 2.4GHz ZigBee transceiver and the CC2500 is a 2.4GHz RF transceiver. The wireless router they implemented only forwards packets and do not measure any sensors.

Chapter 2: Platform Review

2.1 HARDWARE

The project was accomplished using TI's EZ430-RF2500 development kit, one additional EZ430-RF2500T board was purchased to prove a single hop situation. This development kit contains the TI MSP430F2274 coupled with a TI CC2500 which is a 2.4GHz RF transceiver used to communicate with the other devices. The following figure [5] shows on the left the USB device used to program the EZ430-RF2500T boards as well as provide the serial link communication for the Access Point to push data onto the laptop. The device on the right is a EZ430-RF2500T board.

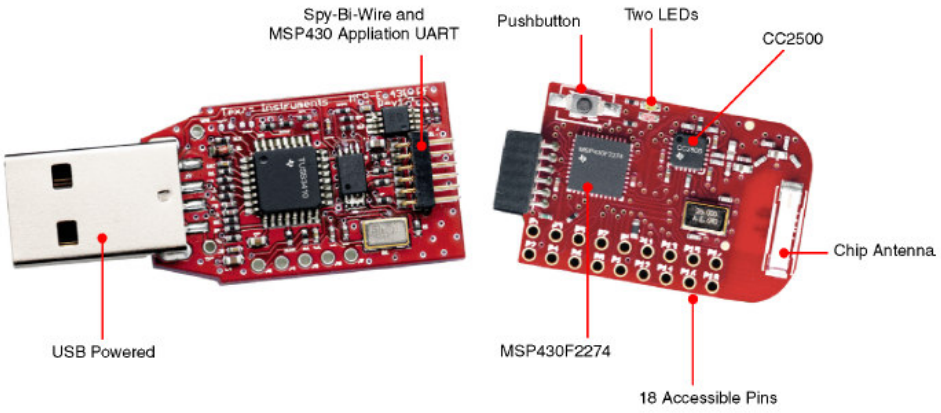


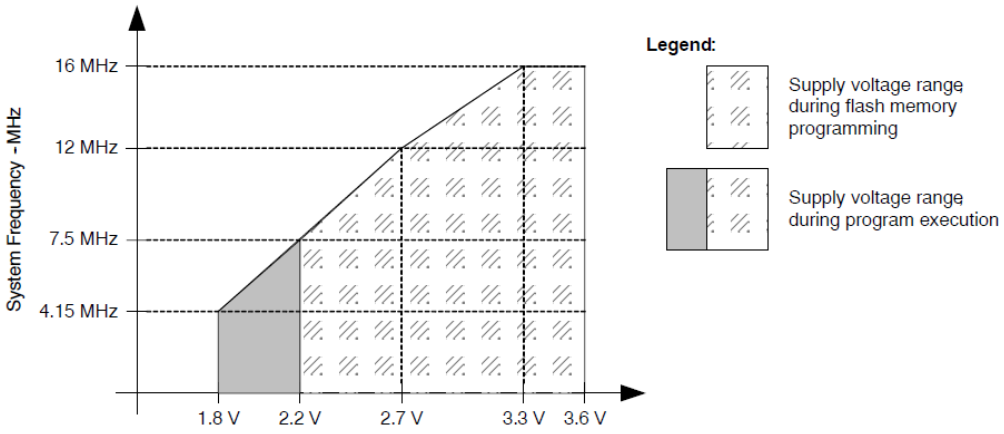
Figure 1: Photo of EZ430-RF2500 development board

The EZ430-RF2500 was ideal for my implementation because it had a wide operating range of voltages, down to 1.8VDD, allowing for longer life from the batteries. The boards are also paired TI's MSP430F2274 as the MCU and TI's CC2500 as the wireless transceiver. Both devices are high performance and low power offering a 16 MHz, full 16 bit processor and a 2.4GHz transceiver.

To maximize the distance the wireless nodes operate, a couple of features were designed by TI into the devices. First by starting with low power devices; the MCU chosen, the TI MSP430F2274, operates at 16MHz in active mode only pulls 270µA [4]. The full wireless device, while transmitting, was measured to be pulling 84mW at 3.2VDD (the power numbers will be expanded on in the results section). Also to lower power, a wireless transceiver with low output power levels was chosen but to achieve the distances requires, a routing protocol was implemented. Because of the limitations of the provided TI API, this was not easily available. Only three device types were available, an

access point which connects to many clients in a star configuration, a range extender which is a dumb device which only repeats all received messages in a flooding method but cannot do any work itself and an end device which makes a single connection to the access point. The range extended was investigated but not implemented in the final product.

The MSP430F2274 has a wide operating range of voltages, as shown below. Even though the core frequency scales with voltage, this did not affect the performance when tested from 2.5V to 3.5V. The below chart [4] shows the effective system clock as the supply voltage drops.



NOTE: Minimum processor frequency is defined by system clock. Flash program or erase operations require a minimum V_{CC} of 2.2 V.

Figure 2: MSP430F2274 Operating Area

The MSP430F2274 has a 10-Bit 200-ksps Analog-to-Digital (A/D) Converter with Internal Reference, Sample-and-hold, Autoscan, and Data Transfer Controller [4] which is multiplexed to multiple inputs allowing for multiple sources of analog capture from input pins, it's internal VDD as well as an internal thermal diode. This was important for this project so minimize external components and allow for many analog sensors for future growth. The EZ430-RF2500 also breaks out 18 pins to be used for analog capture or digital IO. The digital IO could be used to capture digital sensors, control other devices such as relays, motors, solenoids, displays, etc. to implement a feedback mechanism, making it not just a data collection device.

Another important design consideration was ease of implementation. The MSP430F2274 does not need any additional connections other than the power supply and as seen below, the CC2500 requires only a minimal number of defined components. The CC2500 datasheet specifies the recommended value manufacturer and series for each component as well as the schematic for a typical application of the CC2500 [1].

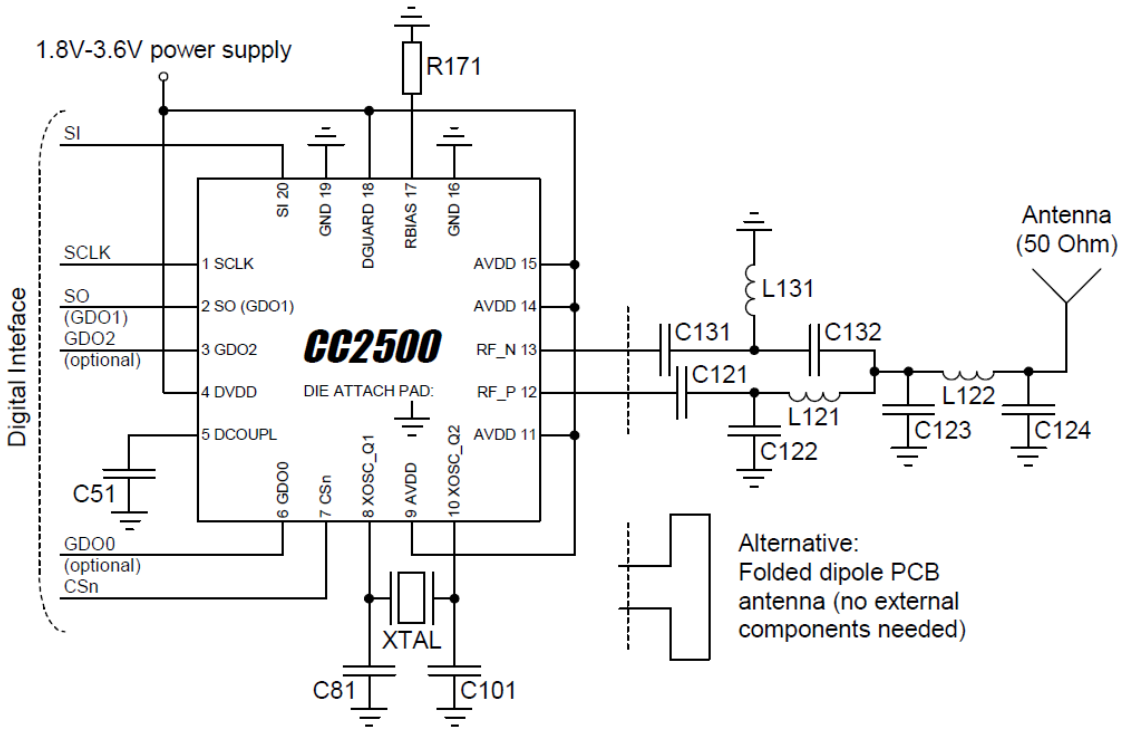


Figure 3: CC2500 Typical Application

Chapter 3: Implementation

3.1 SOFTWARE

I wrote three different pieces of software that ran concurrently. The first program controls the End Devices and was written in C using the IAR Embedded Workshop (http://focus.ti.com/docs/toolsw/folders/print/iar-kickstart.html). This program was written without any RTOS or similar running in the background, the code runs independently. Initially the device would go to sleep to save power and every two seconds the MCU would trigger an interrupt which would wake up the MCU and the transceiver, set a semaphore to take a measurement of the sensors and transmit it before going back to sleep. The CC2500 does not provide an interrupt pin to signal when new data is available so a polling scheme was implemented to query the CC2500's receive buffer for new data for implementing the routing scheme. For proof of concept, the code was modified to not put the MCU to sleep to allow to continuously pulling for received data. Any received data would increment a counter and forward the packet back to the Access Point. With further investigation of the depth of the CC2500's receive buffer, this could later be added back to extend the battery life if the project was to be extended. When the MCU was triggered for a self measurement, the MCU would measure its temperature sensor and battery level and packet this data up in an eight byte packet. The packet used consisted of four address bytes, one byte for the packet count, two bytes for the temperature in degrees Celsius and one byte for the battery level with a 100mV step resolution.

The second program controls the Access Point and was also written in C using the IAR Embedded Workshop. Similar to the End Device, the MCU triggered an interrupt every two seconds to capture a self measurement for reference to compare the End Devices against. Also in a pulling implementation, the Access Point would query the CC2500 for received packets. When a packet was received, it would push it over the USB link to the third application running on the laptop. The Access Point would also modify first byte of the address of the received packet to 0xAA to signify it was an echoed packet and to prevent a cyclical transmission of that packet and retransmit it the next device index. That is, if the received packet was from End Device n, the Access Point would send it to End Device n+1. If N+1 was greater than the number of connected devices, it would echo the packet to Device 1. As mentioned above, if the received packet's first address byte was 0xAA, it would only send it to the laptop for logging.

The third program ran on a laptop to log the data and was written in C++ using Microsoft Visual Studio 2010. This application pulled data received from the Access

Point, format it and push it out for logging. The program received the temperature in Celsius and converted to Fahrenheit and took the coded battery level (to allow a fixed point number to be sent as 8 bits) and converted it to volts. This program took the data and pushed it out to a CSV file so the data could be read with Microsoft Excel or a similar spreadsheet application as well as printing it to the standard IO for the user to debug with. The standard IO was allowed to be toggled via a debug variable.

To implement my code on the platform, I used TI's EZ430-RF2500. The EZ430-RF2500 connects to the computer via a USB dongle with proprietary firmware loaded which allows for programming and serial communication with the computer. I designed a packet format for transmitting the data, the packet consisted of eight bytes; four address bytes, one byte for the packet count, two bytes for the temperature in degrees Celsius and one byte for the battery level with a 100mV step resolution. Originally for packet routing, I was exploring the implementation of network flooding to keep the simplicity down. The provided wireless APIs provided by TI did not support multiple devices connected to multiple other devices; I was forced to implement packet echoing for proof of concept to avoid rewriting the existing TI libraries. The APIs supported a star network configuration in which one device (the access point) is connected to many end devices or clients. The existing API only allows one connection per end device. Because of this, when the access point had two or more connections, it would echo the received packet from End Device N to end device n+1. If N+1 was greater than the number of connected devices, it would echo the packet to Device 1. The end device which receives the echoed packet would then increment the counter and send it back to the access point for logging. This effectively modeled a 1-hop link and allowed for measurements of the packet success rate before and after the hop without any handshaking protocol implemented. The wireless transceiver, the TI CC2500, supported four different modulation schemes; OOK, 2-FSK, GFSK and MSK [1]. The final product was implemented using 2-FSK because initial testing showed no noticeable performance differences and 2-FSK was the power on default of the device, which made the power up sequence simpler.

The wireless transceiver, the TI CC2500, supported four different modulation schemes; OOK, 2-FSK, GFSK and MSK [1]. The final product was implemented using 2-FSK because initial testing at distances of roughly 10' showed no noticeable performance differences and 2-FSK was the power on default of the device, which made the power up sequence simpler. This was determined when initially setting up and testing the basic wireless performance using a very low bitrate (1 second per packet) application. Also, 2-FSK setup at 2.4kBaud used the smallest bandwidth at 91 kHz wide channels; MSK at 500kBaud occupied 489 kHz wide channels. The desired modulation scheme is

setup be setting bits 4-6 of the MDMCFG2, Modem Configuration register on the CC2500 [1]. Below is a topology map of the network I created.

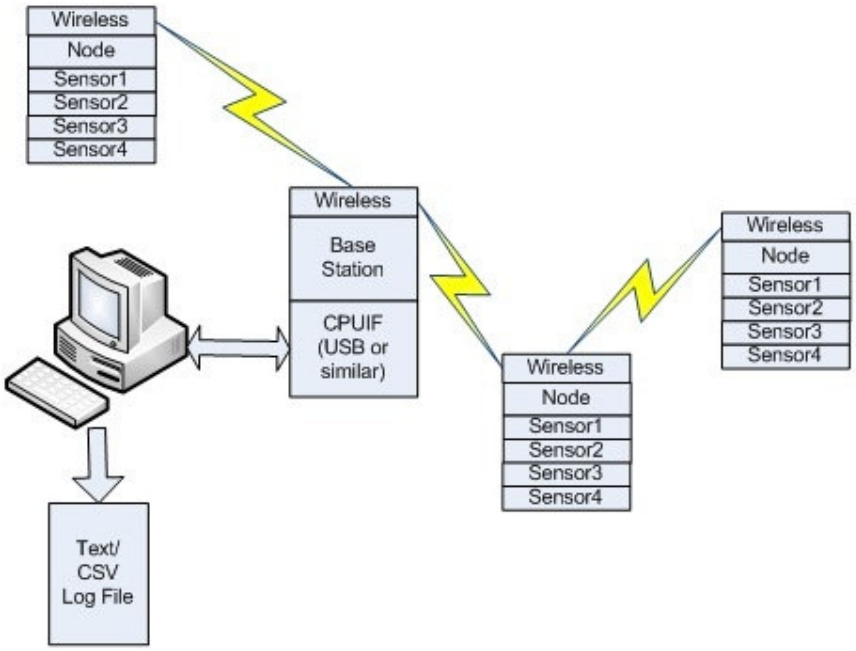


Figure 4: High level map of system

Chapter 4: Experiments

My goal was to determine the packet drop rate and true power consumption of my system in different environments using three devices on the network; one connected to a laptop acting as the access point and two remote devices powered by two AAA batteries acting as the end devices or clients.

4.1 DISTANCE

Three different environments where chosen to compare the network's performance; inside a residential home, outside in a residential neighborhood and at Town Lake, Austin to simulate a rural area.

In close ranges, distances less than 50', there were no noticeable differences between the three environments. I could not exceed 50' inside the residential environment because of the space limitations of the test house. Beyond 50' in the two outside environments, the results surprisingly did not differ greatly; successful transmissions were accomplished at distances only 10' further in Town Lake. A full data set for the rural environment was not captured so a comparison graph could not be made. While in an urban environment the device could successfully transmit up to 80'. In the rural environment, the device could transmit successfully at 90' via a single link but could not connect at 100'.

To measure distance, in a urban environment, I used a sample size of 500 packets. I started measuring off distances in 5 foot increments up to 50 feet. At 50 feet, I was still receiving good results so I started 10 foot increments up to 100 feet. At 90 feet, I could not get either device to connect so I backed it off to 85 for the final measurement. These results can be seen in Figure 5.

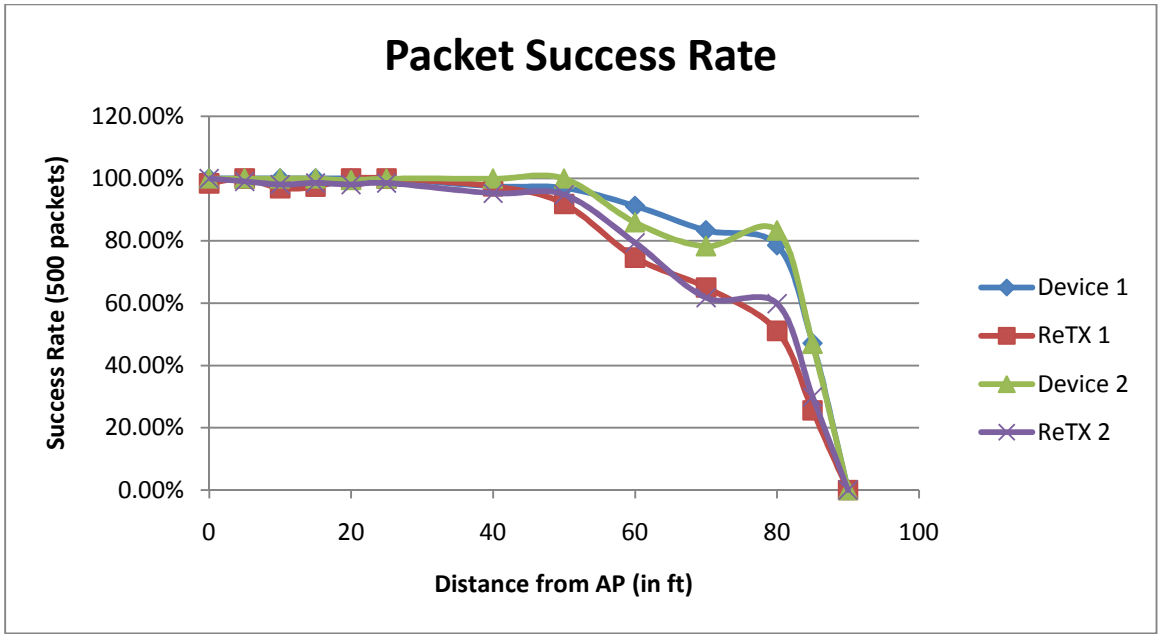


Figure 5: Packet Success Rate as a Function of Distance in an Outdoor Urban Environment

4.2 POWER

The device operates in two states. The first state is idle; the second is wireless transceiver active. The wireless devices operate from a minimum of 1.8 VDC and a maximum of 3.6 VDC provided by two (2) AAA batteries. When the batteries are fully charged (3.2 VDC measured) the device pulled 22.4mA when idle and 26.4mA when transmitting. After the batteries are drained to a measured charge of 2.83 VDC, the device pulled 21.9mA in idle and 24.8mA when transmitting.

From this, assuming the battery did not discharge slowly over time and went from full charge to no charge and the typical charge of a standard alkaline AAA battery of 2700mAh, these devices will run for roughly 120 hours when transmitting 1% of the time.

Chapter 5: Conclusions

5.1 Summary of System

This project took commercially available boards from TI and implemented a low power, wireless system that relayed messages using a simple routing protocol. The system was able to collect its battery level as well as the ambient temperature as a proof of concept for sensor data collection. Actual implementation of this system has proven to be practical; each board costs $20 plus the cost of any sensors to be attached, and the system was able to communicate effectively with links upto 80' in length in an urban environment for long (2 second) intervals.

5.2 Summary of Results

In both a rural and urban setting, the network proved to have adequate packet success rates (greater than 50%) up to 80' on a single link without any handshaking implemented. This coupled with the fact that without optimizations such as putting the device into a sleep mode when idle, the device was measured to be able to run for 120 hours on a single set of AAA batteries make it a viable solution for a practical implementation.

The network was not programmed for its maximum throughput (500kbps [5]) because of power concerns. The network was implemented such that each End Device captured its data and sent it wirelessly at a rate of 8 bytes of data every 2 seconds for each single link. With packet echoing, this rate is increased; with two End Devices you are now effectively at 8 bytes a second.

5.3 Summary of Software

The main effort was in writing and debugging the software. Three programs were written; one to control the End Devices, one to control the Access Point and one to log the data on a laptop. Initially debugging was difficult because only one device could be in debugging mode at a time due to only having one USB dongle. Many of the issues in the code arose from developing for a new platform that I had never worked with before. Through reading the source code of TI's libraries and the provided documentation, all available for download at http://focus.ti.com/docs/toolsw/folders/print/simpliciti.html, I was able to get a better understanding of the platform, the libraries and successfully work through each issue such as the End Devices ability to only have one active link and the End Device automatically coming up with the receive antenna turned off.

5.4 FUTURE DEVELOPMENT

5.4.1 Design

For future development, the TI SimpliciTI protocol could be modified to allow the End Device to forward the packets to truly extend the network. Based on the end application, the End Devices could be modified to control motors, relays, servos, solenoids, lights, displays, etc based on measured data, information sent back from the Access Point or other End Devices. Also, to save power and extend battery life, the MCU could be put to sleep and when it wakes up to do a self measurement, it could query the CC2500 queue for received packets. An external buffer (FIFO or similar) might be added to extend the time between wake ups if the data is not critical.

5.4.2 Documentation

If the product was marketed, a simple 'how to' document could be written to assist the final user of the device. Operation should be simple, the node, once turned on, should start to collect data at the current rate based on settings available to the user. The sensor interface could be open sourced in the future revisions so that a users group or Wiki could be created online to support this. For other assistance, a FAQ and contact information could be provided on a website.

5.4.3 Marketing

The targets for such a system would be farmers or other agriculture workers who wish to collect data on their crops. The system would also be targeted at disaster recovery organizations or governments who wish to place these devices in an environment to continuously collect data in an urban or rural area. The benefit of these 'drop and forget' devices is that they are simple to use, easily reconfigurable and do not require running long wires to collect data. Further development could be done to process and use the data in real time to actively control an environment. E.g. turning on and off fans, sprinklers or lamps/light. The initial customers would be peers to collect data in their home/yard to do a simple study of ease of use and data usefulness (type of data, sampling frequency, range, resolution vs data size, etc).

Bibliography

[1] "CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver." 28 June 2006. Web. Spring 2010. <http://focus.ti.com/lit/ds/swrs040c/swrs040c.pdf>.

[2] "CR2477 Datasheet." Web. Spring 2010. < http://www.houseofbatteries.com/pdf/CR2477>.

[3] "E92 Datasheet." Web. Spring 2010. <http://www.houseofbatteries.com/pdf/E92>.

[4] "MSP430x22x2, MSP430x22x4 MIXED SIGNAL MICROCONTROLLER DATASHEET." July 2006. Web. Spring 2010. <http://focus.ti.com/lit/ds/slas504d/slas504d.pdf>.

[5] "eZ430-RF2500 Development Tool: User's Guide." April 2009. Web. Spring 2010. <http://focus.ti.com/lit/ug/slau227e/slau227e.pdf>.

[6] Hocker, Andrew E. "EtherLux, A Low Power Wireless Display." Dec. 2009. Web. Fall 2010. <http://www.ece.utexas.edu/~adnan/andrew-hocker-ms-report-2009.pdf>.

[7] Molisch, Andreas F. *Wireless Communications*. Chichester: Wiley, 2006. Print.

[8] Zwick, Thomas, Frank Demmerle, and Werner Wiesbeck. "Simulation and Measurement of Bit Error Rates for a 2FSK-System in Indoor Environments." (1998). IEEE Vehicular Technology Society. Web. Spring 2010. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=686655&userType=inst>.

[9] Tsiakmakis, K., N. Mallios, N. Charalampidis, and M. Spasos. "An Improved Low Power Wireless Sensor Network Based on ZigBee for Agricultural Applications." *LATEST TRENDS on SYSTEMS* 2 (2010): 596-601. Print.

Vita

Matthew John Tilleman attended Texas A&M University as an Undergraduate and obtained a Bachelor of Science in Electronics Engineering Technology. While at Texas A&M, Matthew worked for the university as an undergraduate research assistant and a TA for two embedded programming classes. Matthew also accepted a summer internship with Freescale Semiconductors as a product test engineer. After graduation he obtained a job at Advanced Micro Devices in Austin, Texas working in product test engineering specializing in ATE systems and mobile devices. He has a passion for music and incorporating music and electronic design when possible.

Permanent address: 11716 Cherisse Dr, Austin, TX 78739

Email: M_Tilleman@hotmail.com

This report was typed by Matthew John Tilleman.