System Profiling and Error Analysis of a PicoRadio Sensor Network

Daniel Petkovsek
University of Maryland, College Park

SUPERB 2001
University of California, Berkeley
Electrical Engineering and Computer Science

Graduate Mentor: Fred Burghardt
Faculty Mentor: Jan Rabaey

Abstract:

At the Berkeley Wireless Research Center, the PicoRadio group is conducting research into a wireless network of small, low-power nodes. The PicoRadio TestBed was constructed to provide a real world environment for evaluating PicoRadio algorithms. The purpose of my project is to perform analysis in order to characterize the hardware and software of the TestBed. This analysis will identify sources of error. In order to profile the entire system, it is necessary to divide the data path into sections. Each section is a checkpoint to pinpoint the error source. Many sources of error were discovered in the process. Solutions to each error were offered. Graduate students and faculty will be using the TestBed in the future, thus it is important for them to know which errors are inherent to the system. Any additional errors will be as a result of the algorithms under test.

I.) Introduction:

The PicoRadio project is developing a small, low-power wireless network of ad-hoc, multi-hop nodes. There is no central node in this configuration, as there may be thousands of nodes simultaneously transmitting data. The project in its simplest form is a system sending information between two or more nodes. Power dissipation in these nodes is so small (less than 50 uW) that it may be possible to run each node on energy scavenged from the environment, such as the energy from a foot-step. Some uses of this system are in smart home environments, sensor networks, toys, and communications.

The TestBed is made from off-the-shelf components. In order to evaluate algorithms before a single-chip PicoRadio is available, the PicoRadio TestBed was built. The strength of the TestBed is the ability to use this platform for experimentation of applications, networking, MAC, and locationing algorithms. The TestBed is also highly configurable to any application. In the TestBed, the wireless network is composed of a basestation and a corresponding sensor node.

A basic PicoNode consists of a power board, digital board, sensor board, and a radio. The digital board consists of an ARM microprocessor and a *Xilinx* Field Programmable Gate Array (FPGA). These elements process the information transmitted from the sensors. The sensor board has specific sensors for temperature, humidity, and light. Each sensor outputs a voltage, which is then converted into degrees Celsius, percent humidity, or lux value, respectively. The radio used now is a Proxim, at 800

Kbits/sec. An Ericsson Bluetooth radio will be implemented shortly, with much lower power dissipation.

II.) Tests:

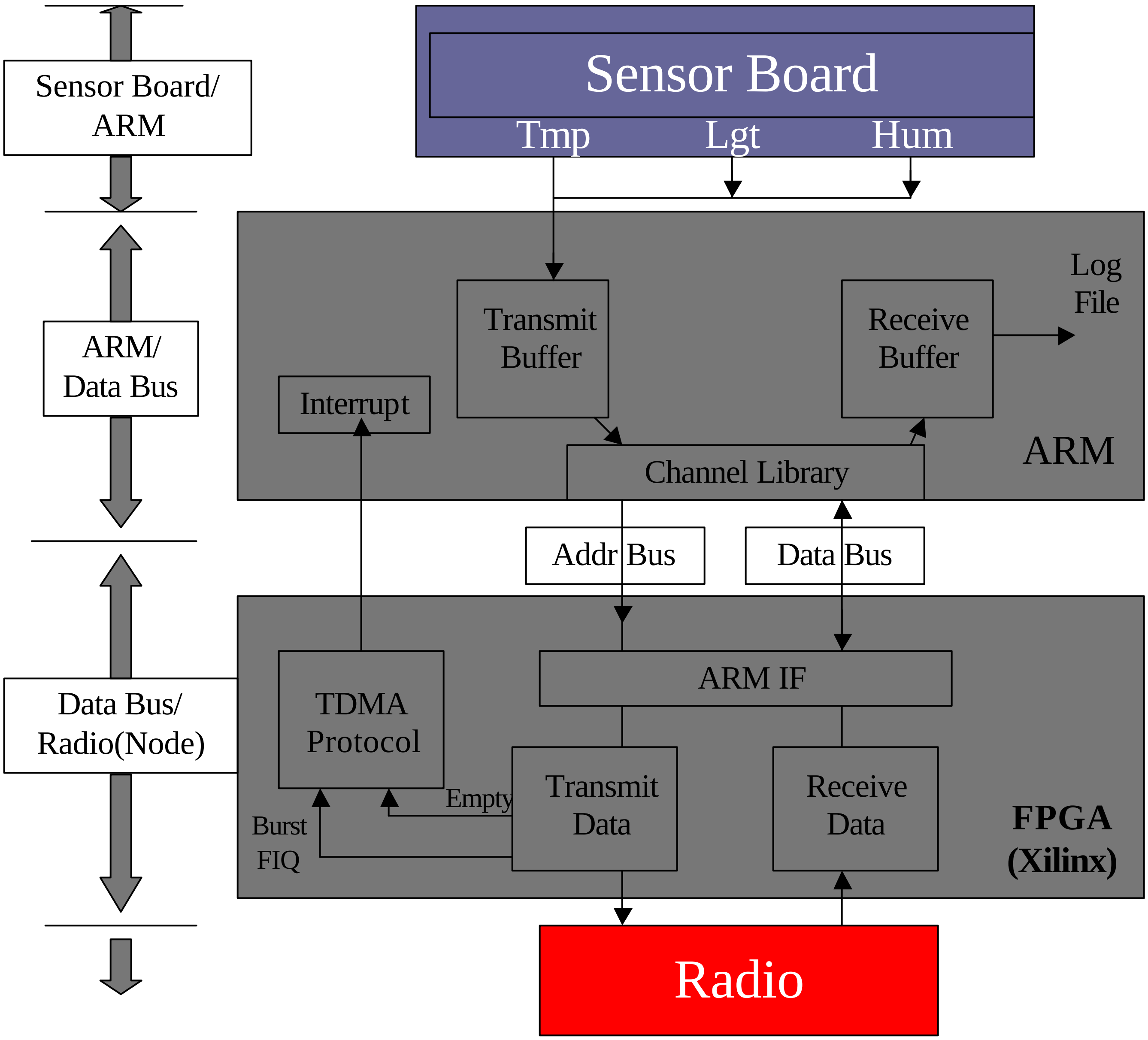
a.) Sensor Board Side

The first task is to check to see if what is being recorded at the sensors is what is being sent to the basestation. In order to accomplish this, you must break down the sensor node into sections.

The first section is between the sensors and the ARM processor. Tests were done to verify that the data taken by the sensors made sense at the ARM. Values were compared in the transmit buffers in the ARM after being logged at the sensor. These values were verified using a Digital Analysis System (DAS).

The second section is between the sensors and the ARM data bus. Tests were done to verify that the decimal values produced by the sensors were not modified or corrupted by the ARM (on the data bus). Again, using the DAS, you can accomplish this.

The third section is between the data bus and the radio. This section includes the FPGA. It was clear that the value on the data and address buses were the same as the values transmitted on the link of the radio. Thus, the analysis of the sensor node sending what is recorded by the actual sensors is consistent, and there are no errors with the processing of the information. Later tests will re-verify with larger data sample sets.



Each sensor on the board has a conversion associated with it. Temperature sensor:

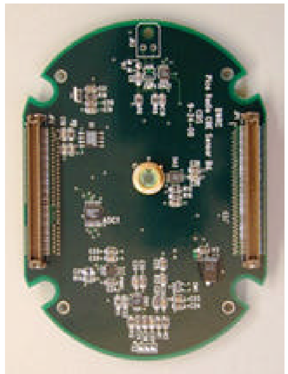
Temp© = (Vout-0.378) / 0.04242

Humidity sensor:

Hum(%) = ((Vout/Vsupply) – 0.16) / 0.0062

Light sensor:

Light(lux) = Vout / (Rfeedback * 1.1 nA/lux)

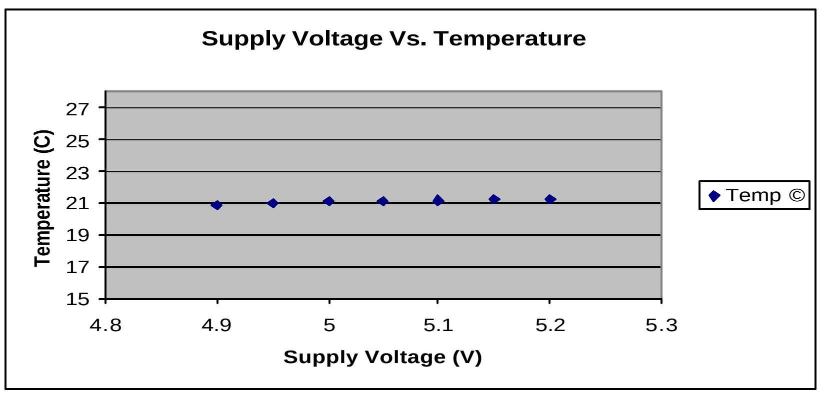


Sensor Board

In order to determine if the sensors would be affected by a change in supply voltage, I manually adjusted the power supply voltage from 4.9 to 5.2 Volts (nominal Vsupply is 5 V). It is possible for the supply voltage to be slightly different from one

Temperature

Voltage	Decimal	Temp ©
4.90	1267	20.93
4.95	1269	20.98
5.00	1271	21.03
5.05	1272	21.05
5.10	1275	21.13
5.15	1279	21.23
5.20	1281	21.27



node to the next, and it is necessary to see if this will affect the sensors. The light sensor was unaffected. However, there were slight changes in the temperature and humidity sensors. For the temperature sensor, the change was in the range of 15 decimal values (on a scale equivalent to 0→ 100 degrees Celsius relative to 0 → 4096 in decimal). However, when analyzing this data further, it is seen that for room conditions, this only amounts to 0.3 degrees Celsius. The same can be said for humidity: a change of tenths of a percent. For the purpose of the temperature and humidity sensors, when measurements are concerned with human comfort levels, this is an acceptable change of output with a change in supply voltage. In other applications this may be an issue.

While looking at the output of the light sensor on the oscilloscope, it was noticed that there was some oscillation in the output voltage. Looking closer, it was found that the fluorescent lighting system was actually being interpreted as an AC signal. This can easily cause problems since a one-bit error is significant in room ambient light levels, which are very low with respect to the light sensors full range. In order to check, a flashlight was shined into the light sensor. A DC signal was the result. Thus, ambient room conditions with fluorescent lighting (such as the lighting used in BWRC) may result in an oscillating signal and a possibility for error.

The robustness of each sensor was also a concern. Each sensor should be stable. In order to check this, a dynamic test for detecting the presence of bad values was installed. The program logged an initial GoodValue. Then samples came in at 5 samples per second. These samples were compared with the GoodValue. A tolerance was set (5%, 10%, 20%), and if the sample was outside of this tolerance, BadValue was incremented. It was found that the light sensor was giving bad values. However, these values weren't outside the tolerance of 20%. This would point to the fact that the room light conditions changed during the test duration (tests were run no longer than 10 minutes). An increase in sunlight would cause this change since a skylight was nearby the TestBed.

Sensor Board Conclusions:

The temperature sensor has been determined to be robust. Temperature is essentially ratiometric with a change in Vsupply. Both the humidity and light sensor have a tendency to be noisy, even if these noise values are not outside of a certain error tolerance. This fact must be known by future researchers. The noise becomes apparent when the sensors are tested for an extended period of time. A solution to this is an analog low pass filter or DSP filtering in the software. Finally, the light sensor will oscillate in room ambient conditions due to the fluorescent lighting scheme in houses and offices. The main problem with this oscillation is that at normal room light, the binary value of the sensor is very low, thus a small change could produce an error of up to 25% depending on the conditions. This can be fixed using a non-linear scale (log graph) on the light sensor.

b.) Basestation side:

The basestation receives data from the sensor node through the Proxim radio link. The sensor node sends this data at a rate of 5 samples/second in my testing phase. Both the sensor node and the basestation log their data to file. The sensor side logs as the data is measured, and the basestation side logs the data when it is received through the radio. It is important to determine that what is transmitted from the sensor node is indeed received at the basestation

The method used to compare this data was a batch compare. Both the basestation and sensor node were run simultaneously. After a set duration in seconds for the test, the programs would dump the data into files. Using a *diff* command, the text files were compared to each other, and the results examined for missing samples (a sample transmitted by the sensor node wasn't received by the basestation) and corrupted data (a sample transmitted by the sensor node was altered). These types of errors are inherent on a radio link. The tests were run for up to 10 minutes at distances up to 40 feet.

	Time (min)	Samples	**Missing Samples**	% Missing	**Data Errors**	% Error
	2	600	**3**	0.50	**0**	0.0
Baseline	5	1500	**8**	0.53	**0**	0.0
	10	3000	**31**	1.03	**0**	0.0
	2	600	**15**	2.5	**1**	0.17
20 FT	5	1500	**40**	2.7	**3**	0.20
	10	3000	**78**	2.6	**4**	0.13
	2	600	**32**	5.3	**8**	1.3
40 FT	5	1500	**83**	5.5	**15**	1.0
	10	3000	**173**	5.8	**32**	1.1

This data concludes that at a baseline distance where the radio are next to each other, less than 1% of the packets are missing and there are no data errors. The number of missing samples and data errors increase as the radios are moved further apart. At 20 feet, approximately 2.6% of the samples are missing, with a data error rate of less than 0.25%. At a distance of 40 feet, approximately 6% of the samples are missing, with a data error rate of 1%. These values are not unheard of as far as radio transmission error is concerned.

Basestation Conclusions:

One conclusion from analyzing the basestation side is that there may be inherent error in the form of missing samples or corrupted data through the radio link. An increase in distance with the radios will increase this inherent error. Methods or solutions to these problems come in the form of either correcting bad data, or making sure it is properly sent. Error-correction codes in the software can adjust for changes in the data. Redundancy, or sending more samples than needed, can also help: once the basestation receives a sample it throws away all copies. Also, if the basestation does not receive a sample, it can ask the node for re-transmission.

III.) Conclusions:

In conclusion, the entire TestBed system has been characterized for errors. Correctable errors in the system are now understood and suggestions have been given in order to correct these errors. Also, inherent errors with the system have been identified. Correctable and inherent errors ranged from bad sensor outputs to missing samples to corrupted data. With the further use of the TestBed for research, users will now know what errors are inherent to the system. Correctable errors will be fixed by TestBed designer extraordinaire Fred Burghardt.

IV.) Acknowledgements

I would like to thank Professor Jan Rabaey for giving me the chance to work in the Berkeley Wireless Research Center with the PicoRadio group this summer. A huge thank you goes to my mentor, Fred Burghardt, without whom this project would be fruitless. I cannot imagine a better mentor to have in the SUPERB program. Finally, I'd like to thank the entire SUPERB staff for making the summer so enjoyable. Not only was the research a great experience, but the social aspects far exceeded my expectations.

V.) References

[1] "The Berkeley Wireless Research Center Website"
http://bwrc.eecs.berkeley.edu

[2] J. Rabaey, M. Ammer, J. L. da Silva Jr. D. Patel, and S. Roundy. "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking." IEEE Computer. Jul. 2000, pp. 42-47

[3] Lab Datasheets from the Pico Radio website and past designers.