

Prototype Wireless Sensor Network for Ecological Study: REU Summary Report

T.D.C. Little, A. Agarwal, J. Tang

Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877
tdcl@bu.edu

MCL Technical Report No. TR-12-31-2005

Abstract—This document describes activities and results of work performed during the period 7/1/2005—12-31-2005 under REU Supplement to NSF Grant No. CNS-0435353 with PIs Little, Alanyali, Phillips, Kunz, and Saligrama.

The objective of this research is the evaluation of in-network belief propagation algorithms and their efficacy as triggers in a deployed sensor network. This technique is a foundation for our proposed system comprised of event triggering and attribute-based routing for the purpose of achieving sensor net longevity. PIs Phillips and Little engaged two undergraduates Johnathan Tang and Meghan Ryan in the application and use of a wireless sensor network for data collection to study impulse behavior in photosynthesis in an ecological research area (tree hydrology). The intent of this work has been to involve both specialists (engineers) and non-specialists (domain scientists) in the use and deployment of the wireless sensors and to contribute to the education of young men and women via a summer research opportunity. This report summarizes the REU activity and results from the summer effort.

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0435353. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1. Background

One of our research activities is to use several ecological application domains to determine requirements for the development and deployment of a wireless sensor network testbed. With such a testbed, we will evaluate techniques for in-network belief propagation and attribute-based routing. In the following, we describe an application of measurement of sunlight for estimation of photosynthesis.

Photosynthesis is the fundamental process that powers the biosphere, but our ability to estimate photosynthesis in ecosystems is currently limited by our lack of understanding of the dynamic nature of photosynthesis in the space-time varying environments in natural ecosystems. Mechanistic models of photosynthesis - applied in contexts as broad as global change biology or crop production - can now be applied to entire ecosystems at time steps approaching a few minutes, but recent increases in computing power that allow this level of time resolution have created the new challenge of needing to characterize high frequency space-time variation in sunlight, temperature, and moisture which characterize almost all ecosystems. Because photosynthesis is strongly non-linearly related to sunlight, temperature, and humidity, researchers cannot simply average environmental variation to predict time-averaged ecosystem photosynthesis. For example, photosynthesis models operating on short time steps may incur large errors if they fail to resolve the impacts of even the passing of a cloud. Thus, we need new tools in which we can capture space-time variation in the key drivers of photosynthesis.

One solution to capturing space-time variation in sunlight is to deploy a high-density network of light sensors that continually resolve space-time light dynamics. However, such a solution is computationally and power costly, because ecosystem environments typically show periods of relative stable environmental conditions punctuated by short periods of rapid change. In other words, there is no need to sample clear, sunny days at high frequency, but there is a need to resolve the short, sudden events such as the drop in sunlight accompanying cloud movement or rain events.

Our goal is to use intelligent sensor networks composed of approximately 15-20 nodes in a grid to efficiently record stable and transient variation in sunlight, temperature, and humidity, in a meadow ecosystem.

The remainder of this report describes the project focus, the activities sponsored under the research experience, results, discoveries, and a summary of future work.

2. The REU Experience

Two students were recruited for the project: Johnathan Tang, a junior in the Department of Electrical and Computer Engineering, and Meghan Ryan, from the Geography Department. They will be tasked to focus on programming the sensor network, and data analysis, respectively.

Activities and Achievements

The students were required to overcome many obstacles leading to a field trial. In particular, we learned how difficult it is to quickly adapt to mote programming under TinyOS. Specific activities and achievements of the students include the following:

1. Packaged (weatherized) the motes (18) for use in the field trials (meadow/light sensing and bat/bridge scenarios)
2. Packaged (weatherized) the Stargates (4) for use as data loggers in field trials
3. Designed and implemented solar and battery configuration for the Stargates
4. Programmed the Stargates to act as data loggers for a set of motes collecting environmental data
5. Created a scientist friendly Stargate 'plug-and play' behavior for use of compact flash (CF) card replacement
6. Implemented CRC and data conversion routines to convert raw logged data to engineering units for temperature, light, barometric pressure, acceleration, moisture (in c and using Matlab)
7. Consulted with PI Kunz on requirements for collecting environmental data in the bridge/bat habitat
8. Deployed the motes and Stargate in Texas (with the help of Kunz) and collected several data sets for the bat/bridge scenario
9. Conducted analysis of data sets from Texas
10. Consulted with Kunz on requirements for collecting data in bat/barn habitat
11. Deployed the motes and Stargate in Paxton Massachusetts and collected data sets
12. Conducted analysis of data sets from Paxton
13. Consulted with Prof. Phillips on requirements for collecting data measuring solar occlusion in meadow scenario
14. Designed packaging specific to solar radiation collection using the Mica2 motes
15. Programmed the motes for data collection specific to solar radiation measurement
16. Deployed the motes and Stargate at Sargent Camp (16) and collected data set
17. Conducted analysis of data set from Sargent Camp

Results

The project immersed the undergraduates in a rich setting of multidisciplinary research involving the PIs, graduate students, laboratory and fieldwork. Students were expected to address design challenges in packaging, software development, and field experimentation. As such, this exposure would appear to match the goals of the REU program. An additional measure of success is demonstrated by the successful follow-on work by Johnathan Tang in the development of his capstone Senior Design Project in a related topic area (completed in May 2006).

The project yielded specific packaging design (see appendix for photos), software for data logging on the Stargate platform, on the motes, in Matlab, and data sets.

Conclusions and Observations by Johnny Tang

The areas of this project that I have worked on this summer span across an extremely wide range. Having worked on both the software and hardware aspects of the project has

given me a seemingly intimate knowledge of two of Crossbow's motes and TinyOS but in reality I have only scratched the surface. My first task was to create a conversion program that converted the data that had been transmitted to the Stargate into values that we can read. Then the data had to be graphed which was a project itself. At first Excel was used but this method lacked automation and required a lot of tweaking from the user. Next Matlab was used and it seemed to be a lot more user-friendly. I was successful in these conversion and graphing programs for both the Mica2 and Mica2dot.

Next I was tasked to modify the code of the motes. I was successfully able to modify the code so that the sampling and transmission rates changed. I was also able to change the program of the Mica2dot to transmit temperature readings from the onboard thermistor. I was unsuccessful in putting the motes to sleep and but was not able apply the light sensor into the extended mode for wider dynamic range.

We did field trials at Paxton MA where a barn is maintained for studying bats. We also collected light readings in the meadow scenario at Sargent Camp in NH. Data were also collected from Texas; however, the configuration of the motes in the presence of the metal bridge caused the data to be very sparse and less interesting as a set.

The Paxton Trial

The first observation I made from Paxton is that there are a lot of these mutant mosquitoes that can bite through shirts and pants, so if you go out there, bring a gallon of mosquito repellent! For the most part our contribution to the experiment was successful. Data were sent to the Stargate during the time we had power. There were a few defective motes but after the Sargent Camp test, this is not a surprise. The biggest problem was that a storm knocked out the power and so we had to run the Stargate off of a generator, which only lasted 14 hours. However, had that not been the case, we have no reason to believe that the motes would stop transmitting. Having tested battery life in the lab, we observed that the batteries would last about 7-8 days even without power management in effect.

Another problem that was discovered was that the Mica2dots' light sensor saturates too quickly. It reached its peak value of 1023 ADC counts in indirect sunlight in the lab. The Mica2Dot light sensor was deemed to simplistic for good light level readings. The configuration of the barn at Paxton was also a challenge. Smaller mote enclosures are needed to fit into small crevices in the roof near where the bats roost.

The Sargent Camp Trial

The Sargent Camp trial was successful in displaying the wide variety of problems that we did not foresee from our prototyping in the lab. First, we discovered that there seems to be a connection with sampling rate, number of motes, and range from the Stargate. At a high sampling rate (of about every 3 seconds) we were confined to a 40-foot grid of 16 motes with the Stargate at its center. The listed range of 400ft was unattainable for our application. Perhaps with one mote transmitting at a slow sampling rate, 400 ft could be achieved. We attempted a range test where we left the motes in the grid formation and moved one mote further and further away. The mote successfully transmitted at 90ft but

we were forced to conclude when it began to rain. Many other frustrating issues arose in this trial including faulty antennas, boards becoming unattached during activation, and batteries running low.

We discovered the limitations of the Mica2 as a sensor for direct sunlight. Although the sensor had been used in the lab under indirect lighting, the direct sunlight created total saturation. We determined several ways to remedy this situation including the use of the sensor's extended mode (shorter integration time on the counts) or the use of a calibrated attenuator – a wratten filter to reduce the intensity of direct sunlight. Unfortunately, under different conditions, for example, overcast, the sensor might have operated perfectly. In essence, we need some kind of automatic gain control for the very wide range of possible light values (from less than 0.01 lux to more than 100,000 lux). Also it is important to note that the Taos SL2550 distinguishes between what type of light is incident upon the sensor. It can tell the difference between sunlight, fluorescent light, and incandescent light. The range for each type of light differs. This is for future consideration.

3. Future Work

Power management is a key factor to the success of long-term deployment of these motes. Developing a scheme to successfully reduce power and ensure longevity is a task that needs more attention.

Improve dynamic range of light sensor. The light sensors provided by the Crossbow MTS400 were satisfactory for measuring light intensity in the context of bat habitats; however, they were found to easily saturate in direct sunlight. Solutions include the use of light attenuating materials or identification of a low-cost light sensor with a broader dynamic range. Also, we did not complete the reprogramming of the Taos SL2550 light sensor extended (5x) mode, which would shift the operation point by a factor of five.

Improve transmission range via robust antenna design. In spite of ideal conditions including 2m elevation and unobstructed line of sight between motes, we experienced very poor range during single hop data logging. This severely limited the utility of the experiments (original grid of 360' x 360' was reduced to 30' x 30' before all motes were detectable by the Stargate logger).

4. Conclusion

This was an engaging experience for all parties. The project was tractable from the standpoint of the students – there was sufficient time to wrangle with TinyOS and the motes for the purpose of enabling data logging and visualization. The fieldwork was extremely enlightening as to challenges with real-world application of the technology. We were disappointed to have not collected more complete data sets from the various trials; however, the lessons learned will make future trials much more successful.

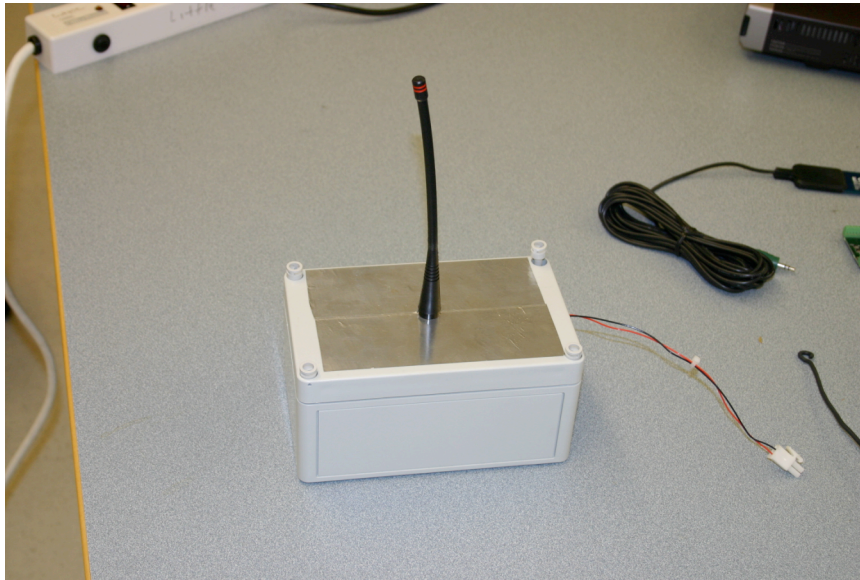
5. References

1. <https://mail.millennium.berkeley.edu/pipermail/tinyos-help/2004-January/002718.html>. This helped me a little with putting the motes to sleep.

2. <http://cvs.sourceforge.net/viewcvs.py/tinyos/tinyos-1.x/>. Contains a lot of test files that exhibit certain abilities. Check under contrib/ucb/apps for files not included in the newer TinyOS downloads (i.e. programs you will not find on this machine).
3. <http://www.tinyos.net/>. Check out the mailing lists if you have any questions. Very useful. Tutorials section is a good introduction to TinyOS and NesC.
4. <http://www.xbow.com/>. Contains all of the user manuals and data sheets for the hardware. FAQ was pretty useful as well.
5. http://www.taosinc.com/faq_category.asp?cateid=4. Introduction to a possible solution of the light sensor issues.
6. http://www.zanware.com/DataSheetAndDrawings/U2_LightSensor_2550-e58.pdf. A more in depth look at a possible solution to the light sensor issues.
7. <http://www.utexas.edu/math/Matlab/Manual/faq.html>. Useful matlab topics.

A. Appendix

A.1. Illustrations



Stargate Weather Packaging



Mica2 Mote and Stargate Weather Packaging



Base Station Setup, Paxton, Mass



Mica2 Motes in Barn, Paxton, Mass



Data Collection in Bat Habitat, Paxton Mass



Base Station Setup, Sargent Camp



Sensor Array, Sargent Camp

A.2. Organization of Work by Johnny Tang

There are many versions of my programs on iceman and colossus. The most up-to-date versions appear in three folders: *Paxton_July19-24*, *Sargeant_July27*, and *dataconversionprograms*. The programs in the Paxton and Sargent folders are identical, however; the programs in *dataconversionprograms* were the ones I was trying different things with so they are not nicely commented. In any case, to get data from the Stargate to graphs in matlab follow these steps:

1. Remove flash card from Stargate and attach a PCI adapter to it.

2. Place the PCI adapter into the computer and a window will pop up. The file you are looking for is in the *motedata* folder and is named *motedata.txt*. I suggest you rename it to something more meaningful.
3. Copy the file to *dataconversionprograms*.
4. Open either a root terminal or a cygwin window, access *dataconversionprograms* and type: **gcc final4.c -lm**
5. This will compile the program in case it hasn't been already. I'm not sure why I need the '-lm' especially since I have already included math.h, but it is needed to avoid an error.
6. When it is finished compiling do one of two things: If you are on a windows machine type: **./a.exe** , if you are on a Linux machine type: **./a.out** .
7. You will then be prompted for the name of the raw data file. Enter 'motedata.txt' or whatever you named the file (without the quotes) and press enter.
8. It may take a few seconds to finish running. When it is finished a few files will be generated.
 - 000.txt is a copy of the raw data file with the UART error fixed
 - 001.txt is the mica2dot data still in the raw form
 - 002.txt is the mica2 data still in the raw form
 - 003.txt is the converted mica2 data with any errors that exist
 - 004.txt is the converted mica2dot data with any errors
 - mica2.csv is the converted mica2 data with the errors removed
 - mica2dot.csv is the converted mica2dot data with the errors removed
9. If all goes well then mica2.csv and mica2dot.csv are the only important files. I will explain some possible solutions for problems that may arise in the **Common Problems** section.
10. Once these files have been generated, decide what data you wish to graph first: mica2 or mica2dot. Since the mica2dot is less interesting, we'll pretend we're looking at the mica2 data first.
11. Open Matlab and the program graphprog3.m (graphprogmica2dot2.m for mica2dot). Modify the date in the code as instructed by the comments. Then click on *Debug* → *Run* or hit F5. It'll take a while to run depending on the size of the data file. A 4.4MB raw text file takes about 5-6 minutes to run through Matlab. When the program finishes a blank Figure 1 will pop up. Maximize this window and look near the top right for a tool labeled *Graphs*. Click on *Graphs* and you will see a few options drop down. Click on whatever graphs you want to see. It is possible to view all of the graphs at once by simply clicking on all of the options one at a time, however; this is not recommended because it usually causes the computer to freeze. I remind you to maximize Figure 1 because you don't want to accidentally close it and have to run the 6-minute program again.
12. This completes the instructions on how to get the data from the Stargate to graphs.

Common Problems

I am not going to pretend that my programs are error-proof nor will I suggest that they are the most efficient means to the end we desire. The following are a list of problems I have encountered and some possible solutions to them.

The UART Problem (the 126's)

If you run the programs and you notice that mica2.csv and mica2dot.csv are empty this may be the problem. Check 003.txt and 004.txt, if either one of them contain 'Error,Error,Error*' at the beginning of each line, the following should help.

Ideally the transmission of data from a node to the Stargate should begin and conclude with the same value. In our case this value was 126. However, we noticed that while early transmissions achieved this form, the broadcasts quickly deformed. The concluding 126 was getting wrapped around to the beginning of the next transmission, causing the data to begin with '126,126,...'. The latest version of my program should add a 126 to those early transmissions that contain only one 126 to the beginning of the line and leave those lines with two 126's alone so that all lines begin with '126,126,...'. (Note I am using 126 as an example, the program should work regardless of what the value is) A problem I noticed that occurs sometimes is that the program adds a 126 to lines that already begin with '126,126,...' so that the line then begins with '126,126,126,...' and since my code depends on counting commas, everything is thrown off and every line will contain an error causing the mica2.csv and mica2dot.csv to be empty. If this is the case then you have two options, both of which require you to modify the sep2.c program.

The first option is to try and debug the sep2.c code. Look around for where a variable named 'uartprob' is compared to a variable named 'same'. As of now I am reading in each character of the problematic UART value (126 in my case), and comparing each character to the character in the next comma space. Use whatever tricks you can muster to remedy the problem although I would avoid *strcmp* because during the comparisons they would always turn out to be false. For some reason when comparing the two variables as strings one of them picks up erroneous characters.

The second option is to comment the entire code in *int sep2()* of sep2.c and simply rename your raw text file 000.txt. The rest of the program will see the lines with only one 126 in the beginning as an error but fortunately, compared to the rest of the file, the number of data points you will lose is minimal.

The Random Error Problem

Another problem I encountered, which is again signaled by an empty or near empty mica2.csv or mica2dot.csv file, is the random error problem. In the m2conv.c and m2dotconv.c programs in a big *switch* statement you will see where all of the conversions are called and values are printed. This is also where seemingly unusual results will be called errors. However, through the various test we have performed, I have realized that what I thought were unusual results may not be the result of an odd environment but of mote error. For example, in the Sargent Camp test the accelerometer sensors read constant values of 41 G's. This is highly unlikely because they were stationary for 99% of the time. Perhaps the heat caused the inaccuracy in the readings, but in either case, my program would call every line an error because I figured 41G's would be the result of a transmission error. Anyway, if you notice that one property (like temperature or x-acceleration...) is constantly being called an error, just go to the *switch* statement and

either comment out the *if...else* that governs what is called an error or widen the range of acceptable values.