A Sensor Network for Soil Moisture Analysis and Ephemeral Stream Detection

Michael A. Murphy
Department of Electrical Engineering and Computer Science
Syracuse University
mike.murphy@ieee.org

Christopher J. Post*
Department of Forestry and Natural Resources
Clemson University
261 Lehotsky Hall
Clemson, SC 29634
cpost@clemson.edu

September 12, 2005

Abstract

Sensor networks based on the *de facto* standard Berkeley TinyOS platform are changing the way environmental information is collected in the field. One such network has been designed, deployed, and tested in order to determine where ephemeral streams (small, temporary channels of runoff) form during precipitation events. This small test network was designed around a generic nondeterministic finite state machine component, which was built to be re-used in later environmental sensor network applications. In this paper, the design and testing of the ephemeral stream detection network are discussed, along with changes that will improve this application in the future.

1 Overview of the Soil Moisture Network

The primary motivation for this experiment was the detection of ephemeral streams, or channels of runoff that contain water less than 30% of the time [North Carolina Division of Forest Resources, 2004]. Detection of these streams is important for land use analysis and planning, because pollutants that enter ephemeral streams are likely to wind up in major bodies of water. With sensors that can gage the total amount of moisture in the soil, it is possible to identify these streams by finding locations with high moisture content during storm events.

*Corresponding Author

In order to compare soil moisture levels over a geospatial area, an inexpensive means of recording moisture data was needed. In addition, a power-efficient method to take samples and produce these records was required, because grid power was not available at the site. To meet these requirements, Berkeley mote hardware was selected for its low power consumption, its ability to maintain a log of data, and its radio communication abilities for sharing data between nodes [Polastre et al., 2004].

For this experiment, the Crossbow Corporation's Mica2dot mote was chosen. This mote features a low-power micro-controller, a 512 kilobyte flash storage chip, six available 10-bit analog-to-digital (ADC) converter lines, and a 916 MHz radio [Polastre et al., 2004]. Each mote was enclosed in a sealed housing, with its batteries and sensors connected externally. Software was written in the nesC language for the TinyOS operating system, and the software application was installed on each mote.

The sensors used in this experiment were Delmhorst model GB-1 gypsum block sensors, typically used with the Delmhorst KS-D1 moisture meter (Delmhorst Instrument Company, Towaco, NJ). Comprised of a probe surrounded by a gypsum buffer, these sensors were designed to reach moisture equilibrium with the surrounding soil. Once equilibrium was reached, the total resistance to current flow was proportional to the total amount of moisture in the soil, and so the mote could measure the voltage drop through the sensor in order to gage relative soil moisture [Delmhorst Instrument Company, 2003].

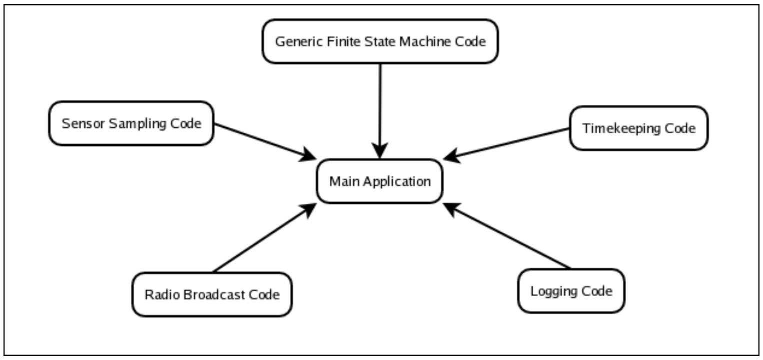
2 The Software Design

2.1 Nondeterministic Finite State Machine Model

Underlying the software used in the experiment was the premise that "interesting," and hence loggable, events only occurred when there was a change in the sensor readings. Whenever the sensors remained at the same level, it was assumed that no event was occurring, and so no logging took place. This assumption was necessary due to the high current draw of writing to the flash chip [Mainwaring et al., 2002]. As a result of acting only upon changes in state, the principal model of software development was based on the concept of a finite state machine. This model has been used in other sensor network designs in the past [Kim and Hong, 2003]. Figure 1 shows how the application is designed to take advantage of a general finite state machine implementation.

Because the sensors themselves are imprecise, it was necessary to account for some drift in the outputs. This drift could occur even when the soil moisture did not actually change, so it was necessary for the mote to make a determination as to whether or not any particular change constituted an actual deviation in soil moisture. This determination was made by first encoding the sensor readings on a 0-15 scale. If the average of these encoded readings exceeded the average of the previous set of readings by some

Figure 1: Component Design of the Soil Moisture Application



number of standard deviations,[1] the transition function dictated a state transition and a logging of the event (figure 2). However, when the specified threshold was not exceeded, the function did not produce a transition. As a result of this calculation, the finite state machine model was necessarily nondeterministic, since the sensor inputs alone did not dictate transitions between states.

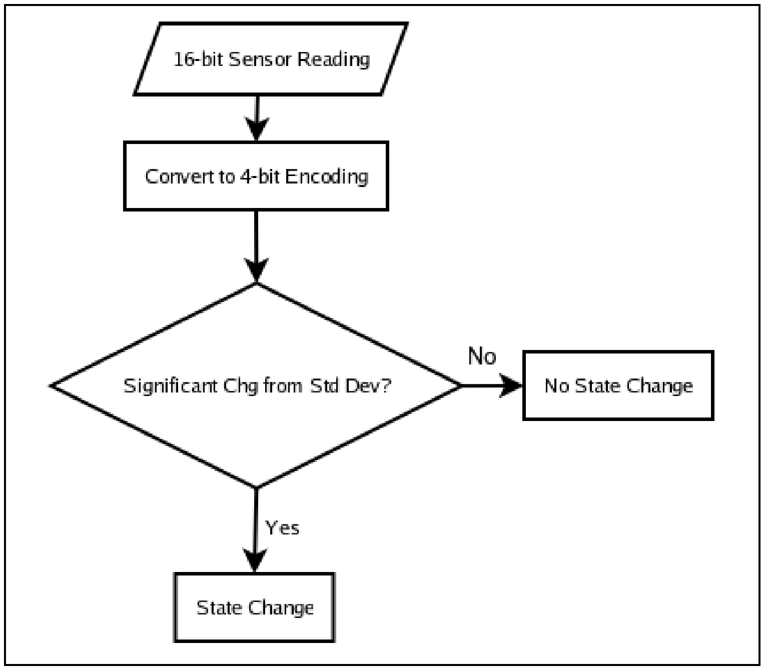
2.2 nesC and TinyOS Programming

Due to the relatively steep learning curve for nesC and TinyOS, programming for the experimental network required several months. During this time, it was necessary to create software components that could read the sensors, perform the ADC conversions, log the data, broadcast messages to other nodes, and keep track of the current time and date.

In order to save power, it was necessary to take sensor readings typically at fifteen minute intervals. However, to achieve better temporal resolution, the software was designed to increase sensing frequency to five minute intervals if several nodes were detecting a rapid increase in moisture (which would be indicative of a rainfall event of sufficient intensity to cause runoff). In order to accomplish this collective, dynamic change in sensing interval, broadcast messages were employed so that each node could announce when it was getting wet. If such messages were received by any given node while moisture was increasing, then that node would increase its reading frequency until the broadcasts from other nodes ceased. In this way, the network would collectively increase or decrease its sampling frequency as soil moisture increased or did not increase.

[1] Calculated from the readings of the six sensors at the previous sampling, with $n-1$ degrees of freedom.

Figure 2: Generic Finite State Machine Decision Process



3 The Experimental Method

Prior to deployment of the network, it was necessary to seal the motes and their external batteries against water intrusion. Because the pins on the Mica2dot motes bent easily, it was also necessary to use a more robust connector for programming, testing, and deployment. As the mote has 18 pins, a 25-pin sub-D connector (DB-25) was chosen to connect the mote to external features such as the programming and power/sensing interfaces. The mote itself was enclosed in an inexpensive waterproof container obtained from a local vendor, with the DB-25 connector mounted on the exterior. Before deploying the network, silica gel desiccant was placed in each mote container, and the container was sealed with silicone.

Power to both the mote and sensors was provided by an external battery pack containing two AA sized lithium batteries. To protect the batteries from the weather, they were enclosed in a semi-sealed container similar to those used for the motes. A DB-25 cable was used to attach a mote, while a 9-pin sub-D (DB-9) connector attached the sensors directly to the battery box. Each sensor was wired to a dedicated pin on the mote's ADC converter and tied to a common ground. At the other end of the sensor wires, each gypsum block was buried under a shallow layer of soil. All connections were given a coating of silicone sealant to seal out water ingress, and the mote boxes were elevated above ground level to avoid their being placed in flowing water situations.

Figure 3: Mote and Battery Box Setup

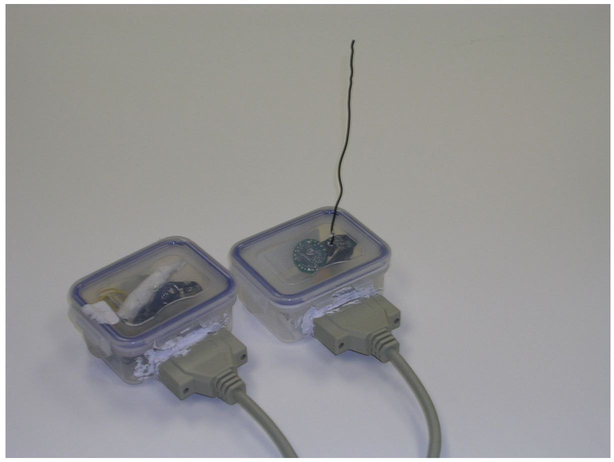


Figure 3 shows an example of the mote and battery box setup.

After being prepared, the combined sensor, battery, and mote units were deployed in a section of the Clemson Experimental Forest [Sorrells, 1984]. Fifteen days after the field deployment, the sensors, motes, and batteries were retrieved to the laboratory. The contents of the logs on each mote were downloaded to a workstation and analyzed. A multimeter was used to measure the final voltages of the battery packs in each battery box.

4 Results of the Deployment

During the deployment period, which began on 17 February 2005 and ended on 4 March 2005, the surface rainfall observation station in Anderson, SC (KAND) observed measurable precipitation on 20-21 February, 24 February, and 27-28 February [National Weather Service Forecast Office Greenville-Spartanburg, 2005]. This station was the closest site to the deployment area for which data were readily available. Also, because the weather pattern for these days involved widespread regional precipitation, it was assumed that the network observed events on these dates. The results of the deployment are summarized in Table 1.

Each battery box contained some water upon retrieval. If the water level was minimal and generally indicated by droplets on the underside of the cover, it was assumed that condensation had occurred. For the three boxes in which measurable standing water was observed in the bottoms, it was determined that rainwater had breached the seals around the connectors. Fortunately, the seals on the mote boxes fared much better, as the insides of each mote housing were completely dry.

Table 1: Post-Deployment Data

Mote	BattBox	LogDate	NumEntries	Voltage	Condition
1	6	3/2/2005	10	3.186V	Breached
2	2	(invalid)	0	1.880V	Breached
3	1	(invalid)	0	0.000V	Condensation
5	3	3/2/2005	20	3.283V	Condensation
6	5	3/1/2005	1574	3.164V	Breached

Key: **Mote:** unique mote number assigned to each mote box. **BattBox:** unique battery box number. **LogDate:** date of the last log entry. **NumEntries:** total number of log entries. **Voltage:** final voltage after deployment. **Condition:** conditions inside the battery box.

Two motes failed to log on the deployment (motes 2 and 3). The failure of mote 3 was obvious and due to human error. One of the batteries was installed backwards, and the mote never received power. Due to the low final voltage of battery box 2, it is likely that the battery box flooded during the first rainfall event, creating a short circuit. It is believed that a hardware wiring failure in a sensor harness is to blame for the behavior of mote 6, with its substantial number of log entries. Further analysis of the data revealed that the majority of entries were zero entries, with the exception of a few nonzero entries believed to be caused by short circuits during one of the rainfall events. Because the standard deviation of zero data is itself zero, the mote logged events at each sensor reading. However, the time stamps of these readings did show that the logging interval did change between five and fifteen minutes during the deployment, confirming operation of the radio communications.

Of the motes that logged properly, none recorded events on each date recorded at KAND. Motes 1 and 5 both recorded events on 20 February, while mote 6 continued to read all zero. Mote 1 recorded some drying on 22 February, as did mote 5 on 23 February. Mote 5 recorded a trend toward more moisture beginning on 25 February, with data running through 1 March. At about the same time in late February and early March, mote 6 appeared to short-circuit due to the moisture intrusion, as nonzero data was recorded. Some issues with event reading may have been due to problems with the sensors: several of the gypsum blocks partially disintegrated over the fifteen-day deployment, well ahead of expectations.

5 Conclusions and Changes for the Next Deployment

Overall, the results of the experimental deployment were both encouraging and discouraging. Many features of the software worked properly, while many aspects of the hardware failed miserably. Water intrusion was a major culprit in the hardware failures. Future deployments will use a single sealed container for both the mote and its battery, minimizing the risk of water intrusion and associated short circuits.

As a result of the observed disintegrations, it was determined that better soil moisture sensors should be used for future deployments. The chosen replacement sensors for the next test will be the Decagon Devices "ECHO" sensor probes, which are accurate to within 3% of soil moisture when left uncalibrated [Decagon Devices, Inc., 2002]. Due to the cost of these sensors, it will be necessary to scale back the number of sensors per mote from six to three.

Attendant with the reduction in the number of sensors, it will be necessary to adjust the transition function to perform a statistical analysis on a time series of data, instead of on the deviations in individual measurements. In addition, some adjustments will be needed to the threshold at which logging occurs. On the initial deployment, 1.0 standard deviations in the encoded data were needed to trigger a transition. However, that number is relatively high and likely caused the Type I error in failing to detect several events.

Once the issues with the existing soil moisture network have been fixed, it will be redeployed for another test deployment. A future operational version of this network will provide valuable information on soil moisture and ephemeral streams in the deployment area. Moreover, the core software in this application, along with its nondeterministic finite state machine design, will be reused in future environmental applications. The components that implement this software model will form the building blocks of other sensor networks that monitor changes in environmental parameters, speeding the development and deployment of future sensor network solutions. Future networks based on this design will monitor a number of application areas, including stream turbidity and forest fires.

References

Decagon Devices, Inc. *ECH2O Dielectric Aquameter.* Pullman, WA, 2002.

Delmhorst Instrument Company. *Model KS-D1 Operating Instructions*, 2003. URL `http://www.delmhorst.com/pdf/ks_d1.pdf`.

Tae-Hyung Kim and Seongsoo Hong. Sensor network management protocol for state-driven execution environment. In *International Conference on Ubiquitous Computing*, pages 197–199, Seoul, Korea, October 2003.

Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002.

National Weather Service Forecast Office Greenville-Spartanburg. Past weather: Anderson archived data. Internet, Greer, SC, 2005. URL `http://www.erh.noaa.gov/gsp/climate/cli_files/anda.htm`.

North Carolina Division of Forest Resources. Types of water resources related to forestry operations. Internet, April 2004. URL `http://www.dfr.state.nc.us/water_quality/wq_typeswater.htm`.

Joseph Polastre, Robert Szewczyk, Cory Sharp, and David Culler. The mote revolution: Low power wireless sensor network devices. In *Proceedings of Hot Chips 16: A Symposium on High Performance Chips*, August 2004.

Robert T. Sorrells. *The Clemson Experimental Forest: Its First Fifty Years.* Clemson University, Clemson, SC, 1984.