XBee-Fio Sensor System Deployment

Colin Heinzmann Robbie Johnson Li Li Joe Ross Greg Schrock

***Abstract*—This short paper presents a sample deployment of a simple, one-hop wireless sensor network. Data was collected over a 24 hour period in Breakiron 164. Nodes used a sense-and-react architecture to collect data and wirelessly transmit it to a base station. The data is then analyzed and evaluated to verify that collection and transmission functioned successfully.**

I. Introduction

Wireless sensor modules provide numerous avenues for analyzing and reacting to an environment when used in conjunction with hardware peripherals. By pairing an Arduino Fio micro controller with a XBee radio communication module, data collected by hardware peripherals can be wirelessly transmitted to a base station for analysis or used in a standalone matter; with potential communication between other deployed units.

Multiple hardware modules were used and experimented with to realize the potential of the wireless sensor package. The hardware modules/peripherals used via the Arduion Fio include (along with their description):

- Door Switch - The door switch is essentially a reed switch that is in the open state. Upon bringing the supplied magnet within proximity of the switch the switch will close. Connecting the door switch to a digital pin will allow for monitoring of the switch state.
- Temperature and Humidity Sensor - A temperature and humidity sensor can collecting data as the sensor name implies. The protocol for communication with this module is response oriented in that data will be collected when the module is ping'ed for collection; rather than constant flow of readings.
- Solar Panel - A solar panel has one main purpose which is to capture sunlight and create current. Charging a battery is a likely option when using a solar panel, but when connected to an analog pin the amount of energy from sunlight being captured can be quantified. The latter is what is intended for this project.
- Thermistor - A thermistor is used for collecting temperature readings in a space encapsulating the thermistor external. The data can be collected through a analog pin and can be converted into a corresponding temperature reading on a micro controller. The relation of analog signal strength to temperature is given from the product info page.
- Ultrasonic Distance Sensor - The ultrasonic distance sensor emits high frequency sound waves and measure the time taken for the sound to propagate, reflect off a surface, and return to the listening end of the module. The start and receive times get be collected on a micro controller and used to determine the distance from the module to the target object. This sensor will use a digital pin for input with a high value indicating that a sound wave has been received.

The goal of this project is to use a sense and react architecture to transmit sensor data from a wireless module to a base station. This project aims to replicate modern use of wireless sensor networks. All of the aforementioned sensor peripherals will be used and the collected data will be aggregated over the course of an entire day; or a continuous 24 hours. At the conclusion of the collection session the collected data can be used to identify if any faults existed in the setup and methods for resolving these issues.

The system design can be found in SECTION 1234 along with the setup and implementation. Breakdown of individual sensor data can be found in SECTION 1234 as well as their accompanying graphs. System analysis, conclusions, and responses can be found in SECTION 1234.

II. System

A. Design

One Arduino Fio collects data from different sensors and sends data to base station. As data was collected by the sensors, the information was sent to the base station using single-hop 802.15.4 communication. The base station then took the information received across the network and wrote the data to output files, with each sensor having its own file. We separate code that controls sensors into different libraries so it's easier to manage and integrate. The sensors sample continually and send data back to the base station every 3 seconds. The door switch is connected to an interrupt pin, and it sends data back whenever someone opens the door. Base station stores data from each sensor into a different file so it's easier to manage. Base station uses Python. Every time new data arrives, it appends the data to the file.

B. Implementation

During the deployment, the Sparkfun Fio micro-controller was hooked up to a Honeywell temperature and humidity sensor, a thermistor, a distance sensor, a solar sensor, and a door switch. The base station was implemented using a Raspberry Pi. The sensors continually sampled the environment, collecting data. Every three seconds the sensors would send a packet of the most recent data to the base station, where it the data was written to a file. When packets become available to the base station, the station parses the contents of the packets to find out which sensor sent the data. Each packet sent over the network contains a header section that uniquely identifies which sensor collected the data. The base station uses this header to determine which file to write the data to. The base station keeps a dictionary mapping the different headers received to their corresponding files.

C. Deployment

The sensor network was deployed for twenty four hours in Breakiron 164. Sensor nodes were placed on the door. This location was selected because it is a high-activity area, where chances of collecting interesting data were higher. Sensor nodes were powered by a 2200 mAh, 3.7 Volt, Lithium Ion battery.

III. Analysis

A. Challenges

The deployment of the sensor network presented a set of challenges. The first was that existing libraries for Xbee-arduino communication are limited in availability and questionable in quality. The main library that we used, xbee-arduino, had a number of open bugs that had to be worked around. Another challenge was the quality of equipment. This was especially apparent in working with the motion sensor. We were unable to get reliable data from the motion sensor over any period of time. Because of this and a lack of interrupt pins, we opted to leave the motion sensor out of our final deployment. We also had hardware problems with some of our Xbee radios, which led to inexplicable bugs where packet reception acknowledgement messages were not sent, and the radio would stop transmitting after 30-34 packets. An added complication was the fact that our distance sensor required 5V to run as apposed to the 3.3V produced by the provided battery. After some deliberation and calculation, the decision was made to place this sensor on a separate board and deploy it at a different location.

Within the deployment, there were several complications. The first came when some unknown individual shut down the computer which was running our base station. The lost data is associated with a number of lengthy straight lines in the collected data and the time period from 10:58 AM to 2:14 PM on Sunday. Additionally, we did need to change the battery on our node as we were running all sensors (excluding the distance sensor) using a single board. This put a considerable power strain on the battery and caused it to die more quickly than it would have had there been a single sensor on each board.

B. Collected Data

One sensor which gave considerable trouble was the distance sensor. The difficulty was that it required a higher voltage than what is provided by our 3.3 V battery. Because of this complication, we were unable to deploy the motion sensor on the node with the other sensors. Rather, it collected data as to the individual using a computer in the lab while it was connected to the computer.

The data collected from this sensor can be seen in Figure 1. This sensor was not able to be deployed for the same amount of time as the other sensors because there was only one Sparkfun Fio board available at the time of deployment. The actual deployment time was 2:55 PM to 12:15 AM, a nine hour period.

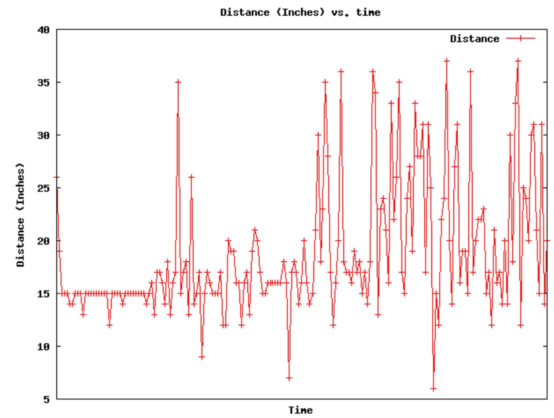

Fig. 1. Data From the Distance Sensor

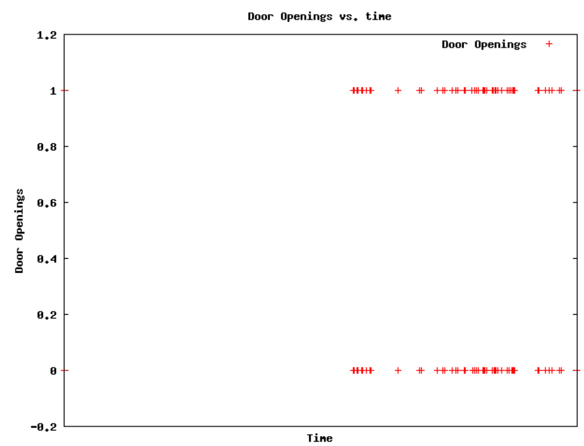

Fig. 2. Data From the Door Switch

In the time that the sensor was deployed, we see a range of activity. Initially, beginning at 2:55 PM Sunday, there was no one at the computer. Then there is a short burst of activity followed by a short period of rest. In the remaining period, exact movements are difficult to distinguish; however, there is considerable activity which continues until the end of the detection time frame at 12:15 AM Monday morning.

The Door Switch was one sensor which allowed an immediate understanding as to the use of the room we were monitoring. The data collected by the door can be seen in Figure 2. In this plot, we can see a very distinct difference between the inactive night period and the heavy traffic period of the day. The initial data points on the left are those that were produced when the sensor was initially deployed and are the only ones before 2:15 PM Sunday afternoon. This is a reasonable result as it is unlikely that students would use the lab during those hours. It must also be considered that we were detecting the use of only one door when there are two entrances to the lab. Thus, some may have entered the lab without being detected.

The data collected from the Honeywell Humidity Sensor was shown in Figure 3. Surprisingly, the only data points

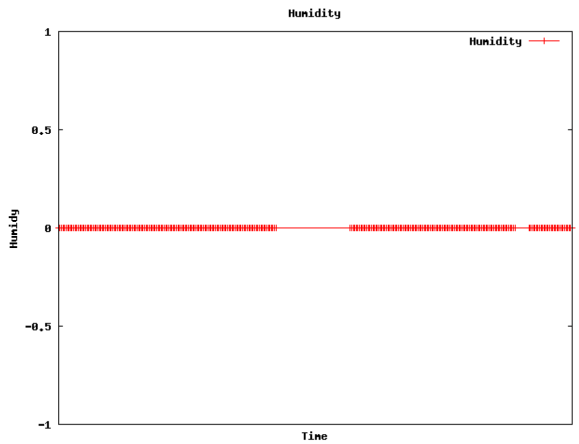

Fig. 3. Data Collected From the Honeywell Humidity Sensor

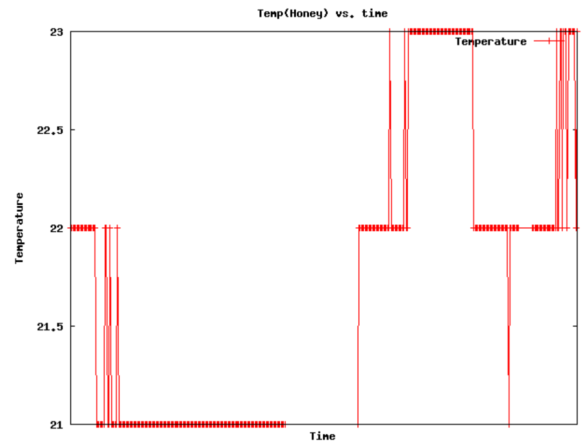

Fig. 5. Data Collected From the Honeywell Temperature Sensor

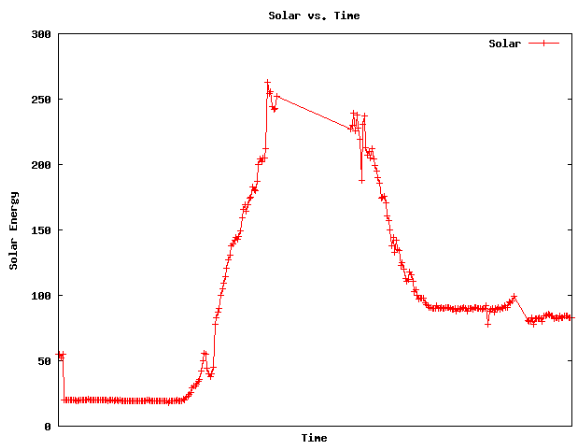

Fig. 4. Data Collected From the Solar Sensor

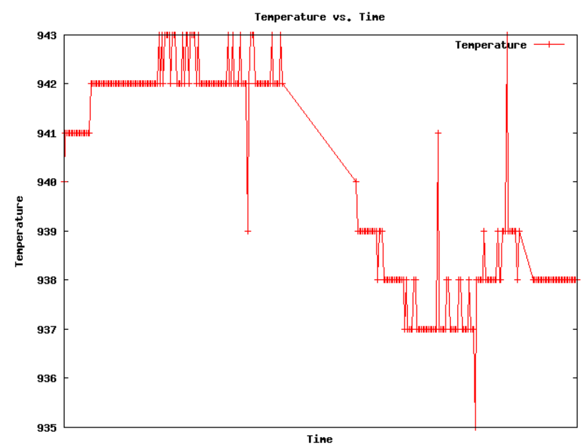

Fig. 6. Data Collected From the Thermistor

the Honeywell Humidity Sensor has collected are zeros. This might be due to the fact that the data was not collected from the sensor correctly, or that the sensor was simply not collecting data. We did not investigate this problem further because of the time limit of this assignment.

The data obtained from the solar sensor matches really well with what had happened in the room. As shown in Figure 4, the system was deployed at around 12:30 AM, Sunday, Feb 22, and there was no light in the room during the night, so the solar sensor readings were very low. Starting from around 8:10 AM, the sensor readings started to rise because of incoming sunlight. At 10:39 AM, the readings reached the peak, and started to fall at 3:18 PM. During the night, there was a help session and people were constantly working in the room, so the solar sensor was detecting room lights.

The data collected from the Honeywell Temperature Sensor was graphed in Figure 5. As shown in the figure, the temperature change over the 24-hour period was not very large. This was due to the fact that the system was deployed indoor without direct exposure to heat sources or windows. At around 2:21 AM, Sunday, temperature started to drop from 22 Celsius to 21 Celsius, and at around 2:16 PM temperature started to rise. All these facts correspond with our common senses, so we concluded that the Honeywell Temperature Sensor was operating correctly.

The thermistor is a means of more fine-grained temperature data than that of the Honeywell sensor. While the exact temperatures are not produced, the relative resistance is measured (Figure 6). Note that a decrease in the plotted value is associated with a rise in temperature. We Find that there are several distinct points in time when temperature is altered. The first change is a sudden decrease which takes place at 2:21 AM Sunday morning. This time has no apparent significance although we speculate that it corresponds closely to the time when our last team members left the room. The considerable rise in temperature, however, begins at 2:16. This is a very short two minutes after individuals began to enter the room. This indicates either a very coincidental result or some intelligent sensing taking place in this area.

IV. Further Studies

As shown in Figure 3, the base station was collecting null values from the Honeywell Humidity Sensor. Further studies include investing the cause of the wrong readings. In addition, we would like to deploy the motion sensor to detect motions.

Furthermore, the fio sensor node used in the project was not energy-efficient enough. It ran out of battery twice during the 24-hour period and this was not acceptable in real-world applications. We would like to do more research on sensor motes that consume less energy and thus can be deployed for a longer period of time.

V. Conclusion

In this project, we designed, developed and deployed a basic wireless sensor node and collected different types of data at the entrance of Breakiron 164 over the period of 24 hours. Even though the system was turned off for a short time and we changed battery for the sensor twice, the data that the base station reports (except for the data from the humidity sensor) corresponds to various events happened in Breakiron 164. We would like to see the system be used in real-world environmental monitoring applications.