5-11-2009

A Wireless Sensor Data Fusion Framework for Contaminant Detection

Joshua Kiepert
Boise State University

Sin Ming Loo
Boise State University

A Wireless Sensor Data Fusion Framework for Contaminant Detection

Joshua Kiepert, *Student Member, IEEE,* Sin Ming Loo, *Member, IEEE, Boise State University, 1910 University Drive, Boise, Idaho, 83725, joshkiepert@boisestate.edu, smloo@boisestate.edu*

Abstract—*In recent years, much research has been done on wireless sensor networks and sensor data fusion, however there has been limited work regarding implementation of real systems that are capable of providing a highly connected sensor network for data logging and data fusion applications. This paper describes the design and implementation of a wireless, portable, and reconfigurable sensor network framework. This sensor node design has proven to be effective for monitoring environmental conditions of aircraft cabins and is well suited to environmental monitoring and detection of contaminants in large areas when utilizing sensor data fusion features.*

I. INTRODUCTION

In the search for means to identify threats to security, new tools must be designed to meet the challenges of a diverse set of possible dangers. More than ever before, the probability of chemical and biological attack has become a real concern. This raises the need for a system that can detect agents before serious contamination occurs, and thus alert the authorities to the presence of a threat.

Sensor data fusion can provide the means to characterize threats in an automated way, but this requires a system that can provide real-time data from many types of sensors. Use of varied sensor types are necessary to enable the most rounded view possible of the environment in question. With the use of different sensors based on different (orthogonal sensing technologies) detection technology, the data collected can be transferred to a central location and provide enough information for data fusion processing.

Our work has focused on the design of a highly portable, reconfigurable, and wireless sensor network for collecting environmental data over large areas. Our focus is on the "back-end" interfacing, delivery, and storage such that different types of sensors can be interfaced to our sensor modules. With the use of wireless networking, the data may be delivered to a central server for further processing. This system utilizes mesh network architecture to allow low power radios to be effective even with low sensor dispersion density or in environments that have obstructions which prevent line-of-sight communications. This system is designed to allow a computer to be used to monitor all sensor activity as it is collected as well as allowing a computer to request information as needed.

By its very nature, sensor data fusion may require a large amount of processing to identify abnormal events, and a large amount of data to help reduce false positives. Our system is designed to store received data in a database which allows high powered computing systems to analyze the collected data as data becomes available. With the data available and accessible at one location, diffusion and pattern algorithms can process the data in near real-time which may aid the detection of contaminants.

This paper will discuss the design and implementation of a distributed wireless sensor system framework for detection of environmental contamination. The design has focused on a general purpose framework on which specific detection needs (such as deployment of different sensors) can be met without significant re-engineering. The database allows sensor data fusion algorithms to be designed to search the data for contamination patterns and potentially reduce false positives.

II. PREVIOUS WORK

Previous to this research much work was done to design a modular, flexible, and reconfigurable hardware platform for collecting sensor measurements. This took the form of a small battery powered device which could be configured with a wide range of environmental sensors. The focus was to design a stand-alone sensor module with the capability to reconfigure the set of sensors on a sensor module with minimal re-engineering. The sensor data storage medium was removable Secure Digital (SD) flash memory cards [1], [2]. The current design utilizes similar hardware, but the system has been redesigned for a more modular construction and includes communication hardware capable of mesh network formation.

Manuscript received March 26, 2009. This work has been funded by the FAA through Cooperative Agreement 04-C-ACER-BSU and 07-C-RITE-BSU. Although the FAA has sponsored this project, it neither endorses nor rejects the findings of this research. The presentation of this information is in the interest of invoking technical community comment on the results and conclusions of the research.

978-1-4244-4179-2/09/$25.00 ©2009 IEEE

III. Wireless Sensor Data Fusion Framework

While there are several commercial sensor systems available that provide sensor hardware and networking capabilities, e.g. [3], [4], and [5], there are few systems that provide a complete sensing solution for sensor data fusion applications. There has been development of interface frameworks for sensor networks, e.g., [6], but few with sensor data fusion applications in mind. This paper describes a wireless sensor data fusion framework containing the core elements in building wireless sensor data fusion system. The framework includes both hardware and software which make it possible to insert data fusion algorithms to process the sensor data. This framework has a highly flexible design such that different sensors can be integrated without re-design of the overall system. The general goal of this research is creating the framework (hardware and software) necessary such that once a sensor has been selected for a particular sensing application the system can be deployed quickly. Data sensor fusion algorithms can be inserted into the framework to analyze the data as desired. The framework is shown in Fig. 1. This diagram shows features that are classified into three categories: (i) sensor node, (ii) communication and interface, (iii) database, visualization, and fusion.

Sensor Node (level 1 to 4)

Sensor nodes with the contaminant detectors are the front-end of a wireless sensor fusion system. It is the element of the system that is passively or actively measuring the contaminant levels and reports the findings in a timely manner. As shown in Fig. 1, the sensor node has circuitry to interface to sensors as well as providing power sources and power regulation for sensors. Once data is collected from a sensor, it is processed, stored, and transmitted. The transmission will require connectivity to the server (or sink node) through a wireless link.

To be effective, the sensor modules were required to meet many design constraints. The primary objectives were to be small, low power, highly reconfigurable, highly connected, and visually inconspicuous. The sensor modules were required to not only send collected data to a remote location wirelessly, but also to enable the storage of collected data locally in case of network failure or applications that require limited network activity and thus send data only after long periods without wireless connectivity.

To meet the connectivity and re-configurability constraints, the hardware was required to have many input/output (I/O) ports and communications protocols support for connections with numerous sensors, and provide internal power supplies to support a variety of sensor types.

As with the hardware, the sensor module firmware was required to be highly reconfigurable in that it needed to have an architecture that minimizes re-engineering when adding or removing sensors from the sensor module. To achieve this objective, the firmware design was required to exhibit data coupling and a functionally cohesive architecture.

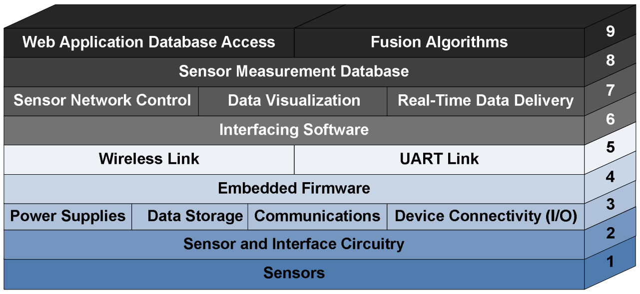


Fig. 1. The wireless sensor fusion framework consists of many design layers that work together to provide a full solution for remote sensing applications.

Communication and Interface (level 5 to 6)

The communication link and interface are the critical infrastructure that delivers the sensed data to the proper destination. For sensor data fusion (which sits in the backend to analyze data and make decisions) to work effectively, determining when the data arrived is critical. The delivery of data depends on reliable communication channels. To meet wireless connectivity objectives, the hardware was required to have a reliable, redundant, and robust network architecture formed between sensor modules (e.g. mesh network architecture). The server might not be reachable directly (one hop away), but the data can be delivered through intermediate nodes. More importantly, having multiple intermediate nodes will guarantee the data will be delivered to the server no matter what happens to any single node.

Database, Visualization, and Fusion (level 7 to 9)

This is the backend – where the heavy duty processing happens. The sensed data (usually in large quantities) is stored on a database server such that algorithms can be applied to “make sense” of the data. The raw data can be visualized, but with its intrinsic volume, it visualization may not be possible: that is why sensor data fusion algorithms can help reduce the data set allowing further attention to be placed on the reduced set. If effective analysis is to be done on the collected data, it must be organized such that relationships can easily be determined.

Both centralized and distributed database architectures have benefits. It is our belief that the backend should have flexibility. For example, new fusion algorithms can be written to process the data without fundamental changes to the system. The system should provide “hooks” such that a new algorithm can be used to analyze the data. The timely arrival of data to the database server is important; however, one will need to define the “real-time” expectation of delivery. The greater the responsiveness needed the greater hardware and design costs to implement the system.

Once the data has been analyzed, the system will need to deliver the results to someone in a timely manner. The result could be a decision (e.g. yes there is contaminant, or no, nothing is out there) or a series of plots and graphs for human analysis.

The remaining portion of this paper describes our prototype and implementation of this framework, starting with the hardware and software. It concludes with a description of applications and a discussion of sensor network simulations.

IV. HARDWARE DESIGN

General Purpose Sensor Modules

For this research, the sensor module hardware was further refined and the wireless communication capabilities were expanded to include mesh network architecture. Computer software was also developed to allow coordination of data collection and provide a facility to fuse data collected across the network of sensor modules. This software was also designed to store the collected data in a centralized database for post processing.

Wireless Capabilities

There are many cases where it is difficult or impractical to effectively determine the state of an environment from a single measuring unit. When the environment is large or its conditions vary greatly over space or time, it becomes necessary to use multiple measuring units to provide enough sensor density to gain a full perspective of the environment in question. It is in these cases where a network of measuring units becomes important. The network allows the measured data to be correlated with each measuring unit in the environment and moved to a centralized database for detailed analysis. With the prevalence of low power, inexpensive wireless communication devices, the creation of high density sensor networks is more easily achieved than it historically has been.

There are two main network architectures employed by the sensor system: star and mesh. The star architecture requires that all nodes connect directly to a single master node. This means that there is a fundamental limitation on how many members may be part of the network, as well as the maximum spacing between nodes. This also requires that there be an unobstructed "view" to the master node. The system will form this type of network provided that all remote nodes are within range of the master node; however, it may dynamically change to form mesh architecture if obstructions or distance begin to interfere with a remote node's communication with the master node.

Mesh, in contrast, has fewer constraints with regard to the layout of the remote measurement units. This architecture allows for multi-hop communication, thus the master node may be located anywhere among the remote units and need only be within range of any one of the remote units. Any messages addressed to the master will be relayed as required to get the message to the master. This architecture is also far more robust in constrained environments where line-of-sight communication to all nodes directly is not possible.

Modular and Reconfigurable Design

Utilizing the previous work done in [1] and [2], the wireless sensor units were redesigned with a more modular and reconfigurable design. This was accomplished by the design of a general purpose system board that provides a microcontroller, real-time clock, Secure Digital flash memory card, three (up to four) step-up/down power supplies, and many digital and analog I/O pins. Fig. 2 shows the main system board design.

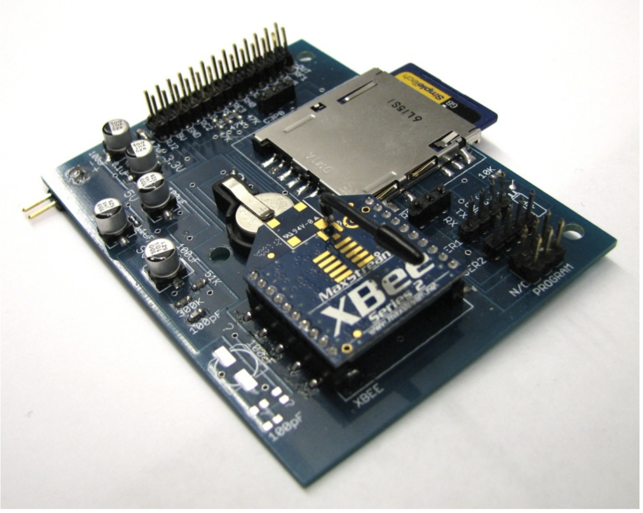


Fig. 2. The sensor module's main system board includes a microcontroller, Digi XBee Series 2 radio, Secure Digital card, real time clock, and four power supplies of various voltages.

Processor

The processor used in this system is a Microchip PIC18F8722 8-bit microcontroller. This platform offers a generous amount of program and data space for embedded applications with 128KB Flash, 4K SRAM and of 1K EEPROM. In addition it offers 70 I/O pins, sixteen of which can function as inputs to a 10-bit analog-to-digital converter (ADC). For communication with various devices the unit has several facilities including two RS232 universal asynchronous receiver/transmitters (UART), and two master synchronous serial ports (MSSP) that support 2/3/4 wire SPI and I^2C master/slave functionality. It also offers three capture/compare/pulse width modulation (PWM) modules and four hardware timers [7]. All of the features described above give the microcontroller significant versatility so that it may easily be adapted to a wide range of sensing applications.

This microcontroller provides not only a wide range of protocols and I/O options but reasonable computation power as well. On this system the microcontroller was set up to run at 8MHz, but the microcontroller has an internal phase-locked-loop (PLL) which allows it to use the 8MHz external crystal and internally run at four times the external crystal frequency. The microcontroller is rated to run up to 40MHz by use of a 10MHz external crystal and the internal PLL [7].

Power
There are many common voltages for sensors. Some of the most common are 5V and 3.3V; however there are others that may be necessary. The system board (motherboard) of the module provides four power supplies running at 3.3V, 5V, 9V and a custom supply that may be configured at build time. The system uses two types of DC-DC converters: a Maxim MAX642 and MAX710. The MAX642 is rated to output up to 18V at 450mA, whereas the MAX710 is rated to output up to 11.5V at 700mA. The system board has room for two MAX710s and two MAX642s. Both of these supplies have efficiencies of over 80% [8], [9]. In the current hardware configuration, the two MAX710 supplies are set up to output 3.3V and 5V. One of the MAX642 chips is configured to output 9V, whereas the fourth supply is not currently used.

Sensor Integration
To provide a means to easily change the sensor set configured on the system, a secondary board or "breakout board" was designed for sensor integration with the main system board. The breakout board currently provides an interface for six sensors: CO_2, temperature, relative humidity, barometric pressure, GPS, and sound intensity. The hardware has also been adapted to other form factors which allow further sensors to be connected externally to the enclosure. The current sensor set was chosen as a means to test the overall framework, as the suite provided several standard sensors for general applications.

Communications
The microcontroller supports many communication protocols; the primary protocol used for external communication is UART. This provides a standard protocol that interfaces with computers as well as other devices. The microcontroller used in the system offers two UARTs. One UART is used for controlling software system configuration via a computer while the second is used for wireless communication through use of a ZigBee modem. The second UART has also been used with a Bluetooth communication module to add link capabilities with PDAs to display sensor measurements.

Wireless communication is achieved with Digi XBee ZigBee modems that use the Industrial, Scientific & Medical (ISM) 2.4GHz band and support both the IEEE 802.15.4 standard and proprietary DigiMesh protocols [10]. These units provide a simple UART modem interface to the microcontroller, and help offload much of the communications processing by managing nearly all of the network formation and routing needed for wireless communication. The XBee modems automatically create ad-hoc star or mesh networks at power-up and dynamically route packets when a destination node becomes out of range for direct communication.

Data Storage and Transmission
Local data storage is accomplished with an SD card reader built into the main system board. This medium was chosen based on its availability, compatibility, and form factor. The small size of SD cards results in minimal space requirements on the system board. Additionally, media card readers and laptop computers commonly support SD media. The current system firmware supports SD cards up to 2GB, which would allow for approximately four years of data collection without removing the card (assuming measurements are taken every 30 seconds).

The data from sensor measurements is stored on the SD card in a human readable text format. This not only allows users to easily view the data collected but it results in simple programming to load data files into databases or generate plots. The same format is also used in wireless transmission of the sensor measurements. Each sensor measurement string is a collection of key-value pairs containing information such as the identification number of the sensor module that the measurement was collected on, the sensor identification number within the sensor module, the raw sensor reading, converted sensor reading, and a time stamp of when the measurement was taken. Additional strings stored in the data files identify the type of sensors, the sensor's measurement units, model number, and description. These strings provide a way to limit how much data is stored for each measurement and how much information is repeatedly transmitted or stored with each subsequent sensor measurement. Fig. 3 shows the data format used by the sensor modules.

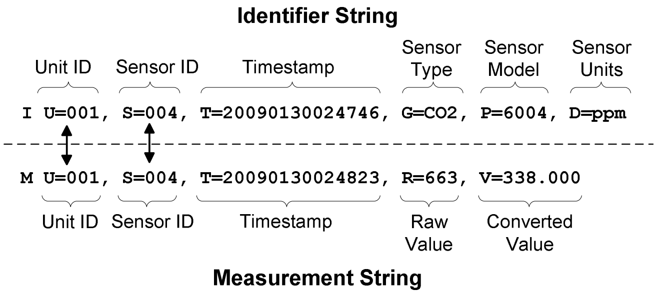


Fig. 3. Example identifier and measurement strings.

As is seen in Fig. 3, the identifier string contains information about the sensor on a particular sensor module, whereas the measurement string contains information about a measurement from a sensor on a sensor module.

The data transmitted has the same format as the data stored to the flash memory on the system. The plain text format requires more data to be transmitted than if formatted into binary packets, but it offers significant advantages in terms of versatility. The string structure of key/value pairs allows for only minor modification to the transmitter/receiver code to change the data fields transmitted or stored.

Time Management
Time keeping is an important part of any data logging device. It is particularly important for a system which must correlate

measurements among distinct, independent modules such as a multi-sensor data fusion system. While it is not strictly necessary to have perfect synchronization among the sensor modules, it is necessary to have the system self consistent in that all sensor modules agree on the ordering of the events recorded by the system. The sensor system described in this paper pushes most time management control to software.

As for the time-keeping hardware, the system utilizes a Maxim DS1339 real-time clock. This chip utilizes an external 32.768 KHz crystal oscillator and is controlled through an I^{2}C interface. It offers very low current operation (~450 µA) and accuracy which depends on the crystal used [11]. The typical crystal oscillators used have an accuracy of +/- 40ppm which means that the crystal has a potential error that could result in up to +/- 10 minutes per year depending on the temperature.

While the DS1339 maintains real time, an internal hardware timer is used to manage system events. The internal timer also utilizes an external 32.768 KHz crystal to maintain system time. This clock is synchronized with the DS1339 at startup and every 24 hours to insure accuracy.

Hardware Implementation

The sensor module hardware is packaged into a small portable form factor. Fig. 4 shows the external form of the sensor module and Fig. 5 shows a sensor module's internal layout when configured with a variety of sensors.

V. SOFTWARE DEVELOPMENT

As the sensor modules are typically battery powered with limited processing resources, it is important to limit the amount of computation that occurs locally. In addition it is important that the software be designed such that it is easily reconfigurable to allow for a wide range of sensors to be connected to the system. This was generally accomplished by a layered and modular design. At the lowest level the code for each sensor must provide a consistent application programming interface (API) that "hides" the low level hardware communication from the higher levels of the software. It is the low level hardware communication layer or driver that will need to be created to add new sensors to the system, whereas only a sensor configuration table will need to be modified to include the new sensors at the top level.

Interfacing Computer Software

As it was a goal to minimize the computation at each sensor module, computer software was required to both store and process the collected data. The processing responsibilities range from graphically representing the data to make it simple to view the real-time state of the monitored variables to applying algorithms to form inferences from the data received.

The primary computer-based application developed as part of this research was the BSU Sensor Monitor (BSUSM) application. This program was designed using the C# programming language as a general test-bed to provide an example of the types of interactions possible with the sensor system. This software offers a wide range of capabilities including, data plotting, data sinking, sensor module device control, and sensor data fusion.

Fig. 4. The sensor module's external form. The sensor module has overall dimensions of 15.3 x 9.2 x 5.4 cm.

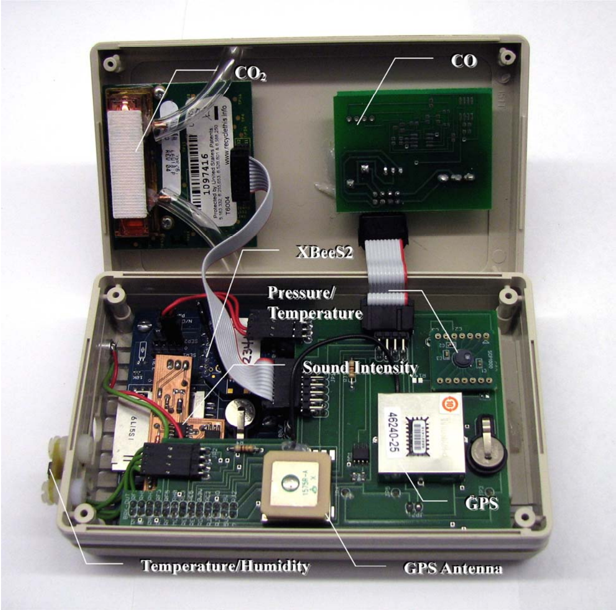
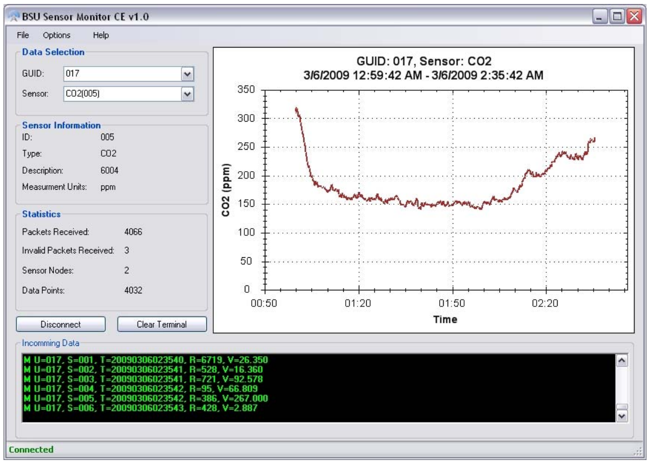


Fig. 5. The sensor module's internal layout. This configuration has carbon dioxide, carbon monoxide, temperature, humidity, barometric pressure, sound intensity, and GPS sensors.

The program's principal function is to allow a network of sensor modules to stream collected data to the application using a USB-XBee adapter connected to the computer. This allows the data to be plotted as it is received in real-time. Each of the active sensor modules has a unique identifier that the BSUSM can use to allow users to select the data of interest. Fig. 6 shows the BSUSM as it plots data received from sensor modules.



GUID: 017
Sensor: CO2(005)

ID: 005
Type: CO2
Description: 6004
Measurment Units: ppm

Packets Received: 4066
Invalid Packets Received: 3
Sensor Nodes: 2
Data Points: 4032

```
M U=017, S=001, T=20090306023540, R=6719, V=26.350
M U=017, S=002, T=20090306023541, R=528, V=16.360
M U=017, S=003, T=20090306023541, R=721, V=92.578
M U=017, S=004, T=20090306023542, R=95, V=66.809
M U=017, S=005, T=20090306023542, R=386, V=267.000
M U=017, S=006, T=20090306023543, R=428, V=2.887
```

Fig. 6. BSU Sensor Monitor software showing a plot of CO2 measurements from a single sensor module.

In addition to plotting data, the BSUSM was designed to store collected data in a structured query language (SQL) database and subsequently export in a comma separated values (CSV) format compatible with common spreadsheet software. Two versions of the software have been developed: one that relies on a SQL server service to be available and another that uses an internal database to manage the data that is presented in plots and data exportation. Utilizing a computer to provide a data sink for a network of sensor modules allows the data to be stored in a central location for real-time or post processing.

The BSUSM application was designed not only to have data "pushed" to it by the sensor modules, but also to directly request data from any sensor unit within the network. This offers many possibilities for the sensor network configuration as the computer may be used to request specific sensor measurements before the sensor module would otherwise have provided it, or remotely control the sensor module for some other purpose such as time synchronization.

Real-time sensor data fusion tools implemented in the BSUSM software allow various data fusion algorithms to be applied as data is received from each of the sensor modules connected. The software was designed in a layered architecture to make the addition of new fusion algorithms straightforward.

Fig. 7 shows the system's network architecture and the relationships between the various database locations. As is seen in the figure, an XBee-USB gateway radio can be connected to a computer and allow the sensor network data to be viewed in real-time, stored to a local database, or pushed to an internet accessible database.

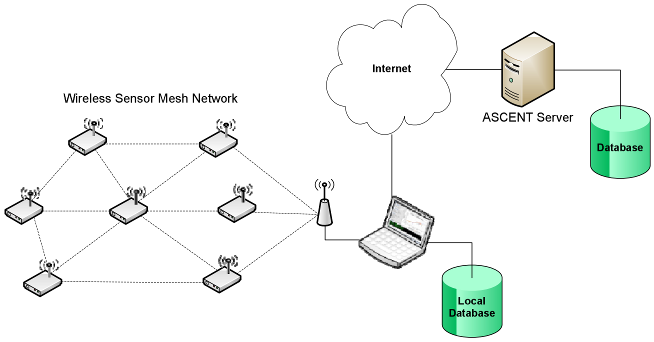


Fig. 6. The network architecture, hardware, and software design allows for flexible deployment options either though local or remote data collection and processing.

VI. APPLICATIONS

One of the primary applications that the hardware design has been tested with is the environmental monitoring of airliner cabins. Restrictions on radio frequency emissions in the airliner cabin environment have, thus far, prevented the use of the networking capabilities; however due to the local flash storage, sensor nodes have been used for single point measurements. There are, however, many potential applications for a complete sensor data fusion framework. Of particular interest for security applications is the deployment of large sensor networks to monitor environmental contamination. Such systems could be used to determine parameters such as threat-level to health, contaminant concentration, and point of origin of contaminants. Additionally, such systems may provide an early warning system that causes an alert if measured parameters exceed a defined level. The flexibility and portability of the hardware and firmware allow a wide range of contaminant sensors to be fitted to the system without re-engineering the sensor modules. The interfacing software and network connectivity allow large sensor networks to be managed both from a data processing standpoint and sensor module control.

One additional aspect of this research has been to develop a means to characterize how this sensor framework could be used to determine the point of origin for diffusive contaminants as in [12]. As experiments with diffusive contaminant sources over large sensor networks can be difficult to implement, particularly with repeatability, some work has been done to develop a sensor network simulator. This allows repeatable experiments to test the framework for identification of diffusive sources. Fig. 8 shows a visualization produced by the simulator as a wave front moves through a sensor network of 16 sensor nodes. Fig. 9 shows the output of the BSUSM while it monitors sensor nodes 0, 5, 10, and 15 during the simulation. As is seen in Fig. 9, there is a distinctive detection peak associated with each sensor node as an environmental change "moves" across the sensor network. The magnitude, spatial, and temporal relationships of the measurements would allow a data fusion algorithm to project the source and direction of the change. While this research is in early stages, it may provide a means to characterize the effectiveness of data fusion algorithms before they are deployed in the field.

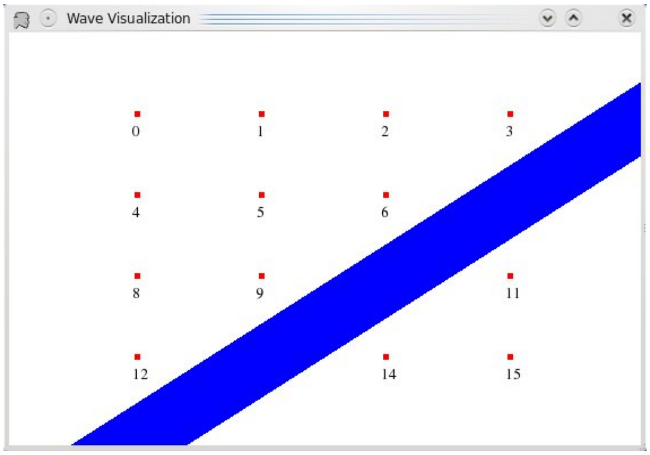


Fig. 8. The graphical visualization during the simulation of a source moving diagonally (to the bottom right) through a grid sensor network.

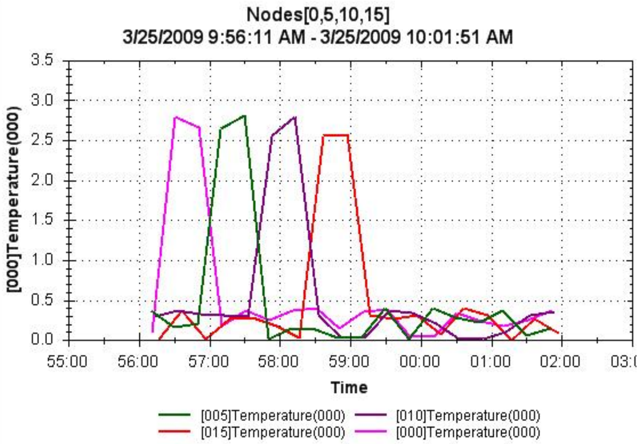


Fig. 7. The identification of diffusive contaminates can be determined from sensor modules arranged in a known locations. This plot shows the detection of a temperature spike moving across the sensor network over time.

VII. CONCLUSION

Many design parameters were considered during the design and implementation of a sensor data fusion framework. A modular and reconfigurable design was developed to reduce re-engineering and simplify sensor changes or application retargeting. Over the course of this research it has been found that wireless sensor networks may require a large amount of infrastructure to provide the best functionality and usability in real applications. The framework has been developed to provide a solution to many sensing applications such as those for security or scientific research.

REFERENCES

[1] M. Owen, "Portable Wireless Multipurpose Sensor System for Environmental Monitoring," M.S. thesis, Boise State University, Boise, ID, 2007.

[2] Sin Ming Loo, Mike Owen, and Josh Kiepert, “Modular, Portable, Reconfigurable, and Wireless Sensing System,” Journal of ASTM International, Vol. 5, No. 4, May 2008.

[3] Crossbow Technology, “Wireless Module Portfolio,” March 2009, https://www.xbow.com/Products/productdetails.aspx?sid=156.

[4] Dust Networks Inc., “SmartMesh XT 2.4 GHz”, March 2009, http://www.dustnetworks.com/products/SmartMesh_XT_2_4_GHz.

[5] Jason Hill, Mike Horton, Ralph Kling, and Lakshman Krishnamurthy. The platforms enabling wireless sensor networks. 2004. *Communications of the ACM* 47, no. 6: 41-46.

[6] Xianghui Cao, Jiming Chen, and Youxian Sun, “An interface designed for networked monitoring and control in wireless sensor networks,” Computer Standards & Interfaces, Volume 31, Issue 3, Industrial Networking Standards for Real-time Automation and Control, March 2009, Pages 579-585.

[7] Microchip Technology Inc., *Microchip PIC18F8722 Family Data Sheet*, 2008.

[8] Maxim Integrated Products, Inc. Technical Staff, *MAXIM Fixed Output 10W CMOS Step-Up Switching Regulators*, 1990.

[9] Maxim Integrated Products, Inc. Technical Staff, *MAXIM 3.3V/5V or Adjustable, Step-Up/Down DC-DC Converters*, 1997.

[10] Digi International Inc. Technical Staff, *XBee/XBee PRO DigiMesh 2.4 OEM RF Modules*, Digi International Inc., 2008.

[11] Maxim/Dallas Semiconductor Technical Staff, *MAXIM DS1339 I^2C Serial Real-Time Clock*, 2008.

[12] Tong Zhao, Nehorai, A., "Distributed Sequential Bayesian Estimation of a Diffusive Source in Wireless Sensor Networks," *Signal Processing, IEEE Transactions on*, vol.55, no.4, pp.1511-1524, April 2007.