A Wireless Sensor Node Architecture for Exploring Distributed Sensor Network Applications

DRAFT - Work in Progress

Dimitrios Lymberopoulos
Embedded Networks and Applications Lab
Yale University
51 Prospect St, #000
New Haven, CT, 06520, USA
(203) 432-0042
Email: dimitrios.lymberopoulos@yale.edu

Andreas Savvides
Embedded Networks and Applications Lab
Yale University
51 Prospect St, #212
New Haven, CT, 06520, USA
(203) 432-1275
Email: andreas.savvides@yale.edu

Abstract— **This document describes a new versatile general-purpose wireless sensor node architecture designed around the OKI ARM ML67Q500x microprocessor and the IEEE 802.15.4 compliant CC2420 radio from Chipcon. This platform designed for experimentation, educational projects and preliminary deployment in industrial environments. The *X*YZ node architecture is described in the context of a 3-D testbed recently installed at ENALAB to facilitate wireless sensor experiments at scale. In the next few months, this testbed will host approximately 100 wireless devices, and it will form a flexible environment for instrumenting and testing complex sensor network scenarios prior to their deployment. The primary set of applications to be implemented on the testbed includes, node localization in 3-D space, sensor calibration and a set of controlled mobility applications such as boundary estimation.**

I. INTRODUCTION

The proliferation of wireless sensor networks and the recent efforts for the standardization of sensor communication under the umbrella of the IEEE 802.15.4 and Zigbee standards are beginning to facilitate a rapidly expansion application space. In addition to the numerous industrial control and home automation applications, this technology carries the potential to significantly impact the course of information technologies in industrial and scientific applications. According to a recent National Research Council report [1], “could well dwarf previous milestones in information technology”.

Our research focus in this domain is the investigation of problems where multipoint sensor measurements are fused to make complex decision on the detection and characterization of distributed phenomena. The main application currently being investigated under this umbrella is the detection of boundaries with distributed wireless sensors. Such detection of delineation between regions in large physical spaces is a highly desirable capability across many domains. Scientists, regulating authorities and public safety officials are interested in studying the propagation of gases and fluids in physical environments. Example applications include the tracking and monitoring of poisonous gases, oil and chemical spills in terrestrial and marine environments, algae blooms and fire spreading. While the study of such distributed phenomena can sometimes be studied using the remote sensing capabilities of satellites or radars, the observation of these phenomena becomes more challenging in obstructed environments such as urban settings, dense forests and indoor environments. Remote sensing is also practically infeasible with some types of chemical sensing where sensors are required to have physical contact with the chemicals being sensed. Despite the fact that boundary estimation has been studied in other domains, its application over large areas imposes a new set of challenges that remain to be addressed. An overview of the challenges associated with boundary estimation given in [2] suggests that this problem in also inherently coupled with several other challenging problems such as that of node localization, sensor calibration, time synchronization and clustering.

The aggregate of these problems calls for the consideration of both data fusion algorithms for the interpretation of sensor information, and highly integrated platforms that offer reliable functionality at low cost and minimal power consumption. Our design of the *X*YZ node based on commercial off-the-shelf components strives to attain a middle-ground between a small form factor, low power device and the current needs for experimental evaluation of sensor network systems. Despite the availability of a few wireless sensor nodes in the community we have decided to develop our own to facilitate specific experiments that we cannot conduct with existing nodes. These reasons include the need to have sensor peripherals directly connected to the microcontroller, small form factor for wearable applications, 32-bit computation resources for running optimization and sensor fusion algorithms, low cost and long term ultra low power sleep modes. Our implementation is based around the OKI ARM ML67Q500x microcontroller, that provides ample computational resources for data fusion and the IEEE 802.15.4 and Zigbee compliant CC 2420 radio from Chipcon. With this platform we are in the process of building a state-of-the-art 3D testbed that will allow us to develop and experiment with scalable sensor network applications. An overview of this testbed is described in the next section.

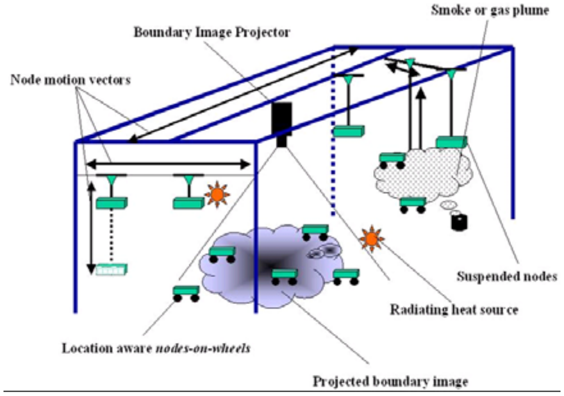

Fig. 1. Visualization of the ENALAB 3-D testbed.

II. 3D TESTBED OVERVIEW

The 3D testbed aims to recreate a wide variety of sensor network scenarios. This allows the flexibility to develop and test certain sensor network services and applications prior to deployment. The testbed has been designed with real-world deployment in mind, so applications developed in the context of the testbed can be easily moved to the actual deployment environment. Figure 1 provides an illustration of the 3D testbed. It is a rectangular structure measuring 4(W) x 6(L) x 3(H) meters constructed inside ENALAB at Yale University. This structure can support a large number of sensing entities build on top of the *X*YZ wireless sensor node. A group of suspended nodes will be able to move in three dimensions on mesh of strings, that will allow the nodes to move in 3-D space by transitioning from one string to another. A group of nodes-on-wheels will be able to roam around the testbed floor. The testbed always has provisions for generating sensor stimuli using an overhead projector and a set of point heat and light sources. The majority of the sensor nodes are derived from the *X*YZ node that we describe next.

III. THE *X*YZ NODE ARCHITECTURE

A. *Design Philosophy*

Our research efforts in wireless sensor networks [2], [3], [4] have identified the need for an open platform that would provide a rich set of interfaces and customized hardware peripherals while providing low power operation and ease of programmability. At the same time, an additional design consideration is the use of a standardized communication interface so that the same platform can be directly deployed in industrial environments.

An alternative research avenue to this approach would be to use commercially available PDAs. Although PDAs offer enough processing power and memory resources, they do not provide enough flexibility for experimentation. For instance, if a PDA is used, one cannot easily design custom peripherals, and there is no control over form-factor requirements. Couple with cost and power requirements, these factors could hinder deployment at scale. Our focus in designing *X*YZ is to provide experimentation flexibility but at the same time make use of existing standards and off-the-shelf components, so as to allow researchers concentrate their efforts on the development and deployment of new applications. More specifically, in the context of our 3-D testbed we are interested in enable platform mobility and simultaneous node localization in 3D space. Another design goal of this architecture is to allow the development of a set of custom peripherals for sensor networks that will be provided in the form of IP core libraries.

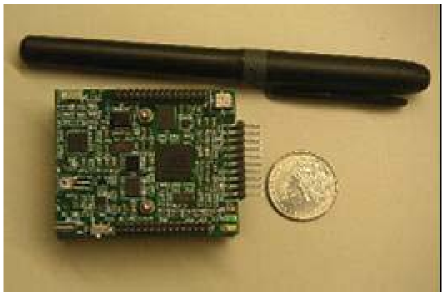

Fig. 2. The *XYZ* sensor node.

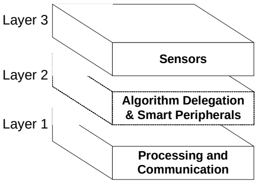

Fig. 3. Sensor node hardware stack

While some industrial and research efforts are directed towards providing processing and wireless communication on a single chip, we plan to focus our efforts in the development of different algorithms and protocols in hardware. Our vision is that in the future, these optimized signal processing protocols will be implemented and organized in libraries of IP cores. Users will be able to utilize these protocols to produce application specific ultra low cost sensor network applications. We envision that the *X*YZ architecture will serve as the enabler for these applications. We plan to use the *X*YZ node as a generic wireless communication interface that *delegates* more complex tasks to an external FPGA that will act as the development platform for IP core implementation as shown in Figure 3. By reusing some of the existing features and protocols running on the main processor, this will enable the development of application specific hardware libraries that can later be used in ASIC implementations.

The first generation prototype of *XYZ* is shown in Figure 2. It is 2” x 2” in size and it is powered by 3 AA rechargeable batteries attached to the bottom of the board. The next subsection provides the details of this implementation.

B. *The XYZ Hardware Components*

Figure 4 depicts the architecture of the *XYZ* sensor node. The OKI ML67Q500x (x=1,2,3) microcontroller [5] was chosen as the processing unit of our sensor node. It is based on a 32-bit ARM7TDMI core capable of operating at a maximum frequency of 58MHz, providing reasonable processing power to explore a wide variety of distributed signal processing applications. Moreover, when 32-bit computations are not required, the CPU can operate in 16-bit THUMB mode that achieves higher code density and thus reducing the memory requirements and power consumption.

The OKI processor provides sufficient memory resources for a wide range of experiments. The on-chip memory includes 32KB of RAM, and up to 512KB of flash and a small 4KB boot ROM. Our design also uses an external 256K x 16 single-chip SRAM to facilitate the use of an on-board monitor debugger as well as to provide additional resources for some signal processing applications such as acoustic localization that require high quantity sampling.

In addition to the memory flexibility, we found the OKI ML67Q family processors to be an attractive choice due to its rich set of peripherals. An embedded DMA controller, a PWM generator, and a rich set of interfaces including four 10-bit ADC inputs, serial ports (RS232, SPI, I^2C, SIO) and 42 multiplexed general purpose I/O pins, are provided by the CPU. Most of the multiplexed GPIO pins are available on two 30-pin headers along with the voltage provided directly by the batteries as well as the voltage provided by the on-board voltage regulator [10]. These connectors facilitate the process of interfacing and powering custom circuits and/or additional boards to the *XYZ* sensor node as it is shown in Figure 3.

The communication subsystem of the *XYZ* sensor node is based on the Chipcon CC2420 radio [6], which is connected to the processor through an SPI interface. The CC2420 is a 2.4 GHz IEEE 802.15.4 compliant single-chip RF transceiver. It is Zigbee compliant and has an effective data rate of 250Kbps, providing high communication bandwidth. The on-board processor can completely turn off the radio or simply put it in sleep mode through the SPI interface, while the CC2420 can wake up the processor when an RF message has been successfully received. In addition, the radio chip provides a variety of 8 different output power levels that can be used for transmission. The power consumption of the radio during transmission heavily depends on the output power level used. Thus, our protocols and algorithms running on *XYZ* can take advantage of these output power levels to reduce the energy consumption of the communication subsystem. Also, the CC2420 features hardware MAC security operations based on AES encryption that uses 128 bit keys. These operations include counter mode (CTR) encryption /decryption, CBC-MAC authentication and CCM encryption and decryption.

The *XYZ* sensor node provides the ability to the user to implement application specific sophisticated power management schemes. The processor itself provides a wide variety of power management features, such as dynamic frequency scaling and a deep sleep mode. The maximum CPU frequency (CCLK) of the XYZ Node is 58Mhz. The operating frequency of the sensor node can be dynamically varied from the user application resulting to significant energy savings. Through the use of internal clock gears, the CPU can change its frequency to CCLK/2, CCLK/4, CCLK/8, CCLK/16 and CCLK/32. Furthermore, the microcontroller peripherals can be individually powered down by the user application to conserve power. Consequently, user applications can dynamically configure the CPU as their requirements change over the time, reducing in that way the energy consumption of the processor chip.

TABLE I

CONSUMPTION ESTIMATES OF *XYZ*’S PRIMARY POWER MODES

Processor	Radio	XYZ’s current(mA)
ON	ON	60
ON	OFF	40
OFF	ON	20
OFF	OFF	0.01

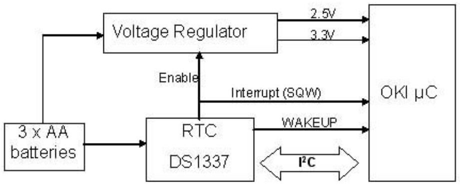

Fig. 5. Supervisor circuit for deep sleep modes

The power consumption of the CPU chip can be furthermore reduced by putting the processor in STANDBY mode. In STANDBY mode, all the peripherals of the chip are disabled and only the necessary circuitry for waking up the processor (external interrupts circuitry) is functional. This ”deep” sleep mode reduces the energy consumption of the processor by orders of magnitude (current consumption of the CPU core in STANDBY mode is approximately 2uA).

When the processor has to remain idle for long periods of time, the real time clock [11] (the supervisor module in the power subsystem shown in Figure 4) can be used for minimizing the overall energy consumption of the sensor node. The real time clock can disable the on-board voltage regulator and thus completely shut-down all the components on the board. After a period of time, that the processor can determine before shutting down, the real time clock can re-enable the voltage regulator and make the sensor node operational again. The supervisor chip can also put the processor in STANDBY mode and wake the CPU up after a pre-determined (from the processor) amount of time. During this time the sensors and/or the radio of the node can be fully functional and they can wake up the CPU if necessary. The circuit for the supervisor circuitry is shown in Figure 5.

In addition to the power control mechanisms, the *XYZ* sensor node also includes a power tracking interface that facilitates the power characterization of the sensor node. Beyond the overall current that *XYZ* consumes, the current consumed by the radio, the core of the processor chip and the

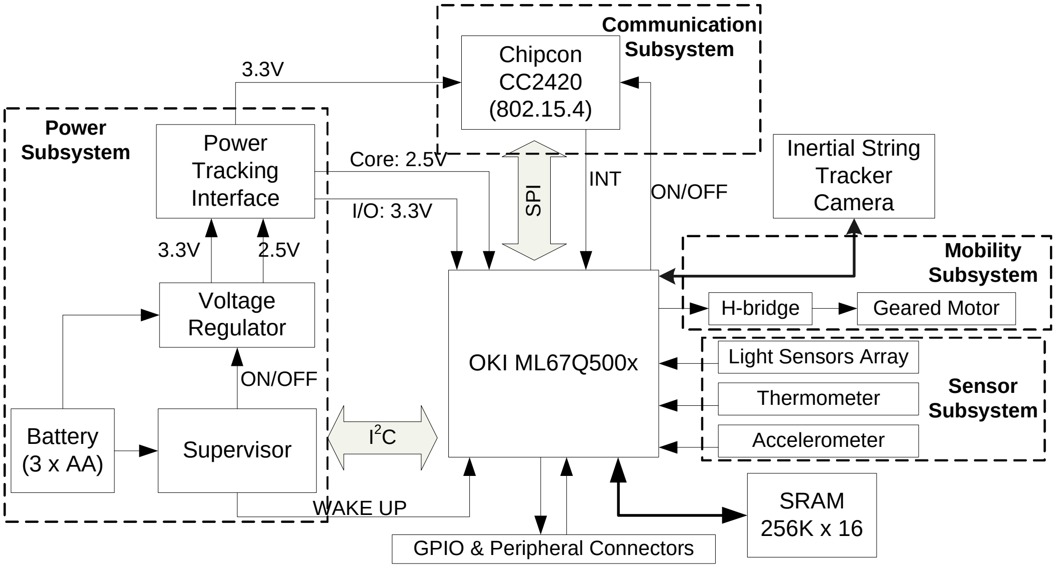

Fig. 4. *T*he XYZ node architecture

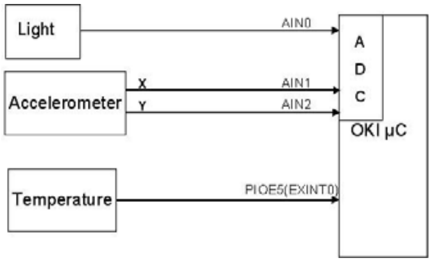

Fig. 6. XYZs on-board sensing subsystem

peripherals of the processor chip can be very easily measured. This feature will allow the fine-grained characterization of the node's power consumption during deployment.

To facilitate initial experiments and for instructional purposes, the XYZ also includes three different sensors: a light sensor array [8], a digital thermometer [7] and a MEMS accelerometer [9]. The interface of these sensors to the CPU are shown in Figure 6. Additional boards with application specific sensors such as ultrasound sensors or various chemical spill sensors will be created and interfaced to the sensor node (refer to Figure 3).

C. Mobility Support Mechanism

A miniature geared pager motor and a custom H-bridge form the mobility subsystem. An early prototype, shown in Figure 7, was implemented using a Mica2 mote [12]. Two output pins of the processor control the motor direction and braking. The motor is controlled via a custom H-bridge which converts the 3.3V logic of the CPU to a 6V DC supply necessary for supporting speed and motor power requirements. A LED/phototransistor pair focused on a 4 segment black and white pattern that is pasted directly onto the wheel which acts as an optical encoder, is used as odometer. A transition from one segment to another, detected by the sensor, occurs when the sensor node is moving. The transition is fed directly to a counter on the mote, and by accumulating the counter values, a position value is determined. In that way, our prototype can follow light gradient (using data from the light sensors) and at the same time it can measure the distance traveled while following this gradient. Using sensing data from different types of sensors we can easily have our prototype following the gradient of a chemical spill or a toxic blob.

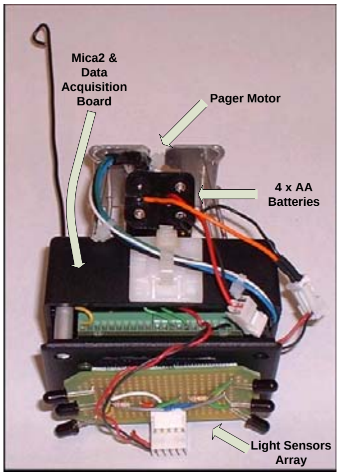

Fig. 7. Early motion enabled prototype.

Implementing this prototype, we found out that for the purposes of our research a more powerful processor with 32-bit computation capability for sensor processing and fusion and the ability to sustain an on-board tracking camera is required.

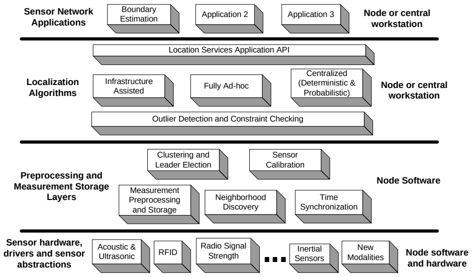

Fig. 8. Layered view of the software running on a sensor node.

IV. SOFTWARE INFRASTRUCTURE

The available software infrastructure at ENALAB follows the layered approach shown in Figure 8. The lower layer makes up the sensor interfaces and drivers. This layer interfaces to a measurement and preprocessing layer which commands the sensor drivers and the initial processing, filtering and organization of the sensor data. The third layer examines the collected data and handles the constrained formulation. A set of services running parallel to this layer perform neighborhood discovery and grouping of the nodes (clustering). The higher layers host specific algorithm implementations. Our implementation is done in the C programming language using the GNU tools.

Our implementation will focus on the use of the SOS embedded operating system, an open source operating system written in C at UCLA/NESL for distributed sensor network applications. The XYZ port of the operating system will be released shortly.

V. CONCLUSION

We presented a new wireless sensor node architecture that facilitates the experimentation of distributed sensor network applications along with our 3-D testbed framework installed inside ENALAB at Yale. Our infrastructure, both hardware and software, provides a versatile development environment that enables us to experiment with networked devices at scale. A detailed power mode characterization and the exploration of an initial set of applications will follow.

ACKNOWLEDGMENT

The authors would like to thank OKI Semiconductor for supporting this project. We also acknowledge the contribution of Cogent Computer in providing help with the design and fabrication of the node. The prototype for the motion enabling mechanism was done by Jimmy Zhang and Stephen Tully as part of a class project in EENG 449b at Yale.

REFERENCES

[1] D. Estrin et. Al, “Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers”, National Research Council Report, 2001

[2] A. Savvides, J. Fang, D. Lymberopoulos, ”Using Mobile Sensing Nodes for Dynamic Boundary Estimation”, To appear at WAMES workshop in conjuction with MobiSys 2004 on June 6, 2004.

[3] A. Savvides, H. Park and M. B. Srivastava, ”The n-Hop Multilateration Primitive for Node Localization Problems”, In Proceedings of Mobile Networks and Applications, vol. 8, pp. 443-451, 2003.

[4] Smart Kindegarten project, http://nesl.ee.ucla.edu/projects/smartkg/default.htm

[5] OKI ML67Q5003 datasheet, http://www2.okisemi.com/us/docs/Intro-5400.html

[6] Chipcon CC2420 datasheet, http://www.chipcon.com

[7] TMP05 complete temperature monitoring system, Analog Devices, $http : //www.analog.com$

[8] TSL251R light sensor, Texas Advanced Optoelectronic Solutions (TAOS), $http : //www.taosinc.com$

[9] ADXL202E accelerometer, Analog Devices, $http : //www.analog.com$

[10] TPS70158 voltage regulator, Texas Instruments, $http : //www.ti.com$

[11] Maxim DS1337 Real Time Clock data sheet, $http : //www.maxim - ic.com$

[12] Berkeley Mica2 motes, http://www.tinyos.net/scoop/special/hardware/