## Introduction

With over a decade of intensive research and development, wireless sensor network technology has been emerging as a viable solution to many innovative applications. Early works on sensor networks and cyber physical systems have been focused on the development of enabling technologies by addressing a myriad of technical challenges such as multihop routing, communication abstractions, middleware and operating systems (OS), and semantic abstractions and sharing of data. Most of the early testbed systems have been built using early stage sensor network research platforms such as CrossBow (now MEMSIC) motes and TinyOS software framework 1. The sensor network hardware platforms are basically lowpower embedded microcontroller systems with some onboard sensors and analog I/O ports to connect sensors. A suite of software components also need to be developed, including OS, sensor/hardware drivers, networking protocols, and applicationspecific sensing and processing algorithms.

There are a large number of prior efforts in building wireless sensor network systems in the literature. For example, a largescale soil moisture monitoring sensor network system is developed using CrossBow motes and it is integrated into a comprehensive environmental infrastructure system, Texas Environmental Observatory (TEO), as described by Yang et al. 1 MoteLab 2 was an experimental wireless sensor network deployed at Harvard University. It provided a

public testbed for development and testing of sensor network applications via an intuitive webbased interface and it was decommisioned in January 2014. INDRIYA 3 is a threedimensional wireless sensor network testbed developed at National University of Singapore, following the same design as MoteLab. Some other related works include the web services approach for the design of sensor networks 4, and a number of webbased applications and software architectures for wireless sensor networks 5,6.

More recently, we have witnessed a new wave of developments in opensource hardware/software, standardization, and commercialization of wireless sensor network technologies. The IEEE 802.15.4 standard specifies the physical and medium access control layers for low datarate wireless personal area networks 9. ZigBee is a lowcost, lowpower, wireless mesh networking standard built upon 802.15.410. The 802.15.4 RF transceivers and ZigBee protocol stacks are now available as commercialofftheshelf (COTS) modules for rapid prototyping of wireless sensing and actuation systems. For example, the Digi XBee series OEM modules implement the IEEE 802.15.4 radio and ZigBee networking protocol 11 and have become very popular in application system development.

Sensor network systems, like most embedded systems, needs to be tightly coupled to their applications. However, the aforementioned recent advances have helped to reduce the complexity of implementing wireless sensing and actuation systems and have made it fairly easy to implement a prototype system for proofofconcept and demonstration purposes. In this paper, we present a wireless sensor network system developed using opensource hardware platforms, Arduino and Raspberry Pi, and the ZigBee module, XBee S2B. Such a design has the advantages of low cost, easy to build, and easy to maintain, as compared to some earlier designs such as the TEO system 1. The major obstacles for sensor network technology to become a transformational force in engineering, scientific, and commercial application domains lie in its lack of reliability, flexibility, scalability, interoperability, and in its extreme difficulties in longterm deployment, operation, and maintenance especially by nonengineering application domain practitioners. The system presented in this paper represents a step forward towards addressing these challenging issues.

The rest of the paper is organized as follows. In Section 2, the overall system architecture is described. Then, in Section 3, the design of hardware and software components is presented in details. Some sample experimental deployment and measurement results are presented in Section 4 to demonstrate the usefulness of the design. Finally, the paper is concluded with a summary in Section 5.

## Overall System Architecture

Building a wireless sensor network system requires development and integration of many hardware and software components. Figure 1 shows the overall system architecture of an environmental monitoring wireless sensor network system that we have developed. The system includes an insitu base station and a number of distributed wireless sensor nodes. Each sensor node is a combination of sensors, microcontroller (uC), and a ZigBee radio transceiver,

# b d



a



c

Fig. 2. (a) Raspberry Pi Model B; (b) Arduino UNO R3; (c) XBee Pro S2B; (d) RHT03 Temperature and humidity sensor.

i.e., the XBee module. In addition, there is a user application program on each sensor node, which handles sampling data from sensors in a certain well defined manner and communication with the base station.

The XBee module on the base station is configured as coordinator and the XBee modules on the sensor nodes are configured as routers as discussed in Section 3. Then, the XBee modules work together to form a mesh network topology using ZigBee networking protocols. To support remote online configuration and management of sensor nodes, we also implemented a gateway application in the base station. It is used to control the behavior of the distributed sensor nodes and to query all or a selected set of sensor nodes to retrieve data. The base station also includes a relational database management system (RDBMS) MySQL for data storage and management. To access the sensor nodes and the data from the outside world, a web application is developed on the base station using the Apache HTTP web server.

In our system architecture, we have combined the gateway node of wireless sensor network, database server, and web server in one singleboard computer (SBC) hardware platform, which helps to reduce the cost and complexity of deployment. A web application is developed to provide users a convenient web interface to the system. Users can interact with the web application within the local area network or from any terminal on the Internet to access the sensor data or perform remote configuration and management of deployed sensor nodes. As compared to the largescale TEO environmental system 1 that we have developed earlier, the system design presented in this paper is well suited for smallscale environmental monitoring and data collection applications.

## Design of Hardware and Software Components

* 1. *Design of Sensor Node*

In this research, we developed networked sensor nodes using Arduino and Digi XBee moduels. Arduino is a widely used opensource singleboard microcontroller development platform with flexible, easytouse hardware and software components. Arduino Uno R3 is based on Atmel Atmega328 microcontroller and has a clock speed of 16 MHz. It has 6 analog inputs and 14 digital I/O pins, so it is possible to connect a number of sensors to a single Arduino board. Arduinocompatible custom sensor expansion board, known as shield, can be developed to directly plug into the standardized pinheaders of the Arduino UNO board.

For wireless communication and multihop mesh networking, we used the commercially available ZigBee module, XBee Pro S2B from Digi 11, which operates at 2.4 GHz ISM band. Indoor communication range of the XBee moudle is 90 m whereas outdoor range is nearly 2 miles. With low power consumption and data rates up to 250 kbps, ZigBee devices are particularly suitable for fast prototyping for wireless sensor network applications. It is possible to build a simple starstructured network or complex mesh network using these devices. To exploit some of the advanced features of ZigBee we need to develop application accordingly both at the gateway and sensor nodes. To keep the initial development simple we used just one type of sensor on the sensor nodes; the sensor that we used is a lowcost humidity and temperature sensor RHT03. The hardware components used in this design are shown in Fig. 2.

The XBee module encapsulates 802.15.4 RF transceivers and ZigBee protocol stacks, and it can be easily integrated into microcontroller or microprocessor systems such as Arduino and Raspberry Pi through UART serial communication interface. The XBee module can be configured into three types of devices: coordinator, router, and end device. Coordinator has the capability to control the entire network. Router can relay messages in a tree or mesh network

topologies. End device can only communicate with the coordinator or the router. There can be only one coordinator in a network; the number of router or end device is not limited.

The XBee module has support for transparent (AT) and application programming interface (API) modes. With the AT mode, the XBee module behaves as a serial line replacement. With the API mode, all data entering and leaving the module is contained in frames that define operations or events within the module. The API mode is required when the network needs to be formed into a multinode mesh or tree topology. In this work, we designed a simple framebased communication scheme in the API mode to implement two functions in mesh network: 1) collect data from sensor nodes to base station, and 2) send configuration commands from base station to sensor nodes.

In the XBee mesh network, the coordinator node can query sensor nodes using either multicast or unicast communication mode. The communication scheme based on unicast is illustrated in Fig. 3a, and the frame structure of one of the API frames that we employed is shown in Fig. 3b. With this scheme, a unitcast frame can be directly sent from base station to a specific sensor node using the 64bit unique device address of the XBee module on sensor node. More specifically, as shown in Fig. 3a, base station first sends a TX frame to a sensor node. Inside this TX frame the 64bit device address of the receiver is embedded. So, the XBee module connected to sensor node receives the RF Data containing the TX frame and sends ACK to base station; meanwhile, it also sends a RX frame to the sensor node. Sensor node application processes the RX frame to get the payload information, and then, based on that information, prepares another TX frame containing the sensor data and the 64bit address of the base station XBee module. The second TX frame is then transmitted back to base station.

* 1. *Design of Base Station*

For the base station, we used a low power creditcardsized singleboard computer Raspberry Pi Model B. The CPU on the board is an ARM processor with 700 MHz clock speed. CPU performance can be compared to a Pentium II 300 MHz processor and the GPU performance is similar to the original Xbox. It has a variety of interfacing peripherals, including USB port, HDMI port, 512MB RAM, SD Card storage and interestingly 8 GPIO port for expansion. Monitor, keyboard, and mouse can be connected to Raspberry Pi through HDMI and USB connectors and it can be used like a desktop computer. It supports a number of operating systems including a Debianbased Linux distro, Raspbian, which is used in our design. Raspberry Pi can be connected to a local area network through Ethernet cable or USB WiFi adapter, and then it can be accessed through SSH remote login.