

XBee Wireless Sensor Networks for Temperature Monitoring

Vongsagon Boonsawat, Jurarat Ekchamanonta, Kulwadee Bumrunghet, and Somsak Kittipiyakul

School of Information, Computer, and Communication Technology

Sirindhorn International Institute of Technology, Thammasat University, Pathum-Thani, Thailand 12000

vongsagon@gmail.com, {popo_jj, just_a_pear}@hotmail.com, somsak@siit.tu.ac.th

Abstract—This paper presents an embedded wireless sensor network (WSN) prototype system for temperature monitoring in a building. This network will be used for management of air conditioning systems at SIIT. The ultimate goal is to help saving the energy cost and reducing energy consumption. The system provides a web user interface for any user to access the current and past temperature readings in different rooms. The network consists of a data gateway or coordinator which wirelessly polls each WSN temperature-monitoring node located in each classroom. Each WSN node consists of a microcontroller on Arduino board and an Xbee wireless communication module based on the IEEE 802.15.4/Zigbee standards. The coordinator also has an Ethernet interface and runs a simple data web server. Hence, the coordinator allows data collection over Xbee and data access from web browsers.

Index Terms—Wireless sensor network, Zigbee, Xbee, Arduino, microcontroller, temperature monitoring

I. INTRODUCTION

Nowadays, the world is facing many challenges in reducing energy consumption and global warming. In the same time, there are many technologies that can be used to resolve these problems and moreover support better living. Wireless Sensor networks (WSNs) are the technology that could provides ubiquitous computing [1]. WSNs deploy many small sensor devices to detect environmental properties. WSNs are the combination of embedded system and wireless communication which allows data transmission among the sensor nodes over ad-hoc wireless networks as shown in Figure 1. The ad hoc networks requires no existing infrastructures unlike those in WLAN or cellular networks. The brain of each WSN node is the microcontroller which processes readings from its own sensors and, in some cases, readings from adjacent nodes as well since the sensors in different nodes located near each other may be highly correlated and hence the amount of sensor data needed to be transmitted from the two sensor nodes can be cooperatively reduced.

In this prototype system, we develop an embedded wireless sensor prototype system for temperature mon-

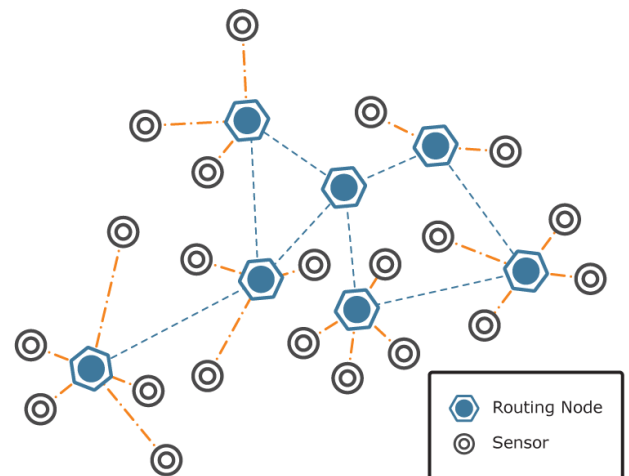


Figure 1. Ad Hoc Wireless Sensor Networks

itoring at the Sirindhorn International Institute of Technology (SIIT). The system is used to remotely monitor temperature in each classroom. It provides necessary data for the whole energy consumption management of the air conditioning systems.

We use the XBee modules based on the IEEE 802.15.4/Zigbee Wireless Personal Area Network (WPAN) standards to build a low-power, low-maintenance, and self-organizing WSN [7]. Small size, low power, low cost and long battery life are the reasons of using ZigBee. We use the Arduino board which comes with ATmega168 or 328 for easy interfacing with the ZigBee module and for easy programming (in C) of the microcontroller. The Arduino boards come with a library for interfacing with XBee module and for dealing with analog or digital inputs and outputs [3]. For the temperature monitoring sensor, we use a low-cost analog sensor to show the proof of concept.

The remaining of the paper is organized as follows. In Section II, we discuss some related works. Section III describes the methodology used in this project. Sections IV and V describe the testing environment and the results, respectively. We conclude the paper

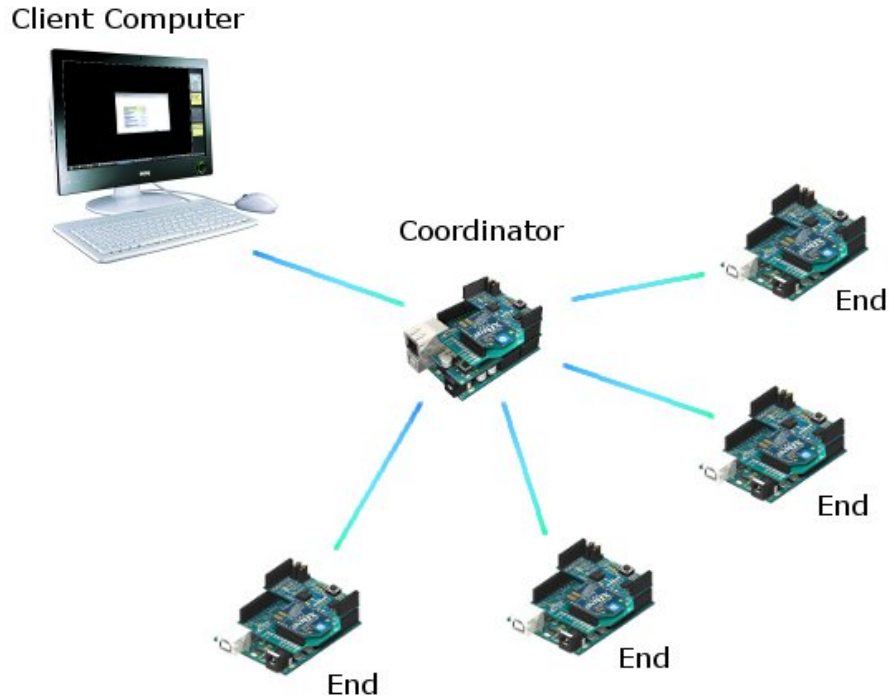


Figure 2. System Diagram

and discuss some future extensions of our current prototype in Section VI.

II. RELATED WORKS

There are several WSN hardware platforms available. In our prototype system, we chose our hardware platform to base on an Arduino board and an Xbee module. This was motivated by the SquidBee WSN platform developed by the Wireless Sensor Network Research Group [5]. SquidBee is an open-source WSN platform, where each SquidBee Mote takes environmental parameters with its three sensors (humidity, temperature and lightness) and sends them wirelessly through the SquidBee network using ZigBee protocol. The hardware of each mote consists of an Arduino board, an Xbee shield that supports an Xbee module, and sensors. Although our prototype hardware platform is not optimized for WSN applications in term of cost, energy consumption, size, nor real-time application support, the platform allows easy development of a WSN due to the simplicity of programming an Arduino microcontroller board.

Over the last decade, many WSN systems have been extensively developed and studied for numerous applications. An example of WSN systems is illustrated in an automation in construction [6]. The authors in [6] proposed a web-based building environmental monitoring system based on WSN. As in typical WSN

systems, the proposed system in [6] consists of three main steps: data acquisition, data collection and data retrieval. Likewise, our prototype system also contains these three main steps.

III. METHODOLOGY

A. Overview

The diagram of the system is shown in Figure 2 and consists of a client computer which provides a web user interface to the system, a coordinator as a data collector and web server, and several end devices which provide sensor readings over Xbee communication links.

B. Design and Development

The system is divided into three main parts as following:

1) *User Interface*: The User Interface communicates to the user via Ethernet and displays the result in a web browser. The User Interface is divided into two parts:

- Automatic display: In this part, it shows the data from every sensor via Web page in HTTP Response. Arduino duplicates itself as a Web-Server to generate HTTP Response in the format as shown in Figure 3.
- Display according to an instruction: The user instructs from the webpage in HTTP Request,

HTTP Request

```
GET /index.html HTTP/1.1
Host: 192.168.x.x
Connection:close
value.....
```

HTTP Response

```
HTTP/1.1 200 OK
Content-Type:text/html
Body.....
Table.....
Button.....
```

Figure 3. HTTP Request and HTTP Response Messages

whose format is shown in Figure 3. The HTTP request message is then sent to the Coordinator who is acting as the web server. The user can also request to read the temperature reading from a selected room (equivalently, a selected end device).

2) *Coordinator*: The Coordinator is the center of system. It collects sensor readings from the sensors back to the user. The function of the collector is divided into two parts: Web-Server and Xbee Interface to the WSN. These two functions as implemented on the Arduino board. The hardware of the coordinator consists of an Arduino USB Board, Xbee Shield, and Ethernet Shield, as shown in Figure 4. The Web-Server function uses the <Ethernet.h> library, while the Xbee Interface uses the <Xbee.h> library. The flowchart of the coordinator is showing in Figure 5.

3) *End Device*: Each End Device has one or more temperature sensor inputs. The device waits for data reading request (i.e., polling) from the Coordinator and then responds with the value from the sensor. The interfaces of End Device consists of two parts: Sensor Interface and Xbee Interface. These two parts work in Arduino USB Board, Xbee Shield and LM335A according to Figure 5. The Sensor Interface receives data from the temperature sensor (LM335A), the temperature signal in Volts is read via an Arduino Analog Input pin (as A/D converted into digital value in the range 0-1023, where 0 represents 0 V and 1023 represents 5 V. We can convert the temperature value in Volt into degree Celsius according to the equation (1). The Xbee Interface uses <Xbee.h> library which communicates between Xbee module and the Arduino.

C. System Specification

1) Network Specification:

a) *Star Topology*: Consists of a Coordinator as a center processing unit functions. It gathers sensor readings from all End Devices.

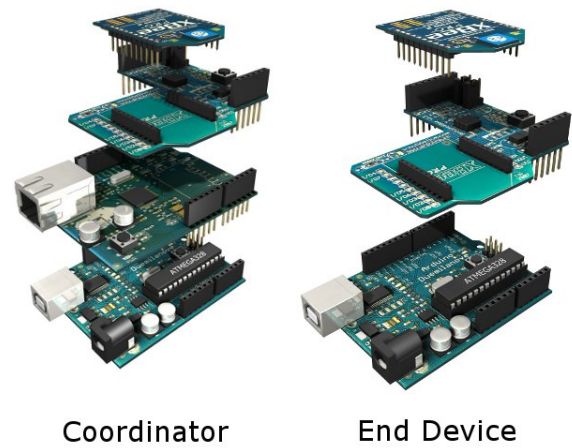


Figure 4. Hardware Platforms of Coordinator and End Device

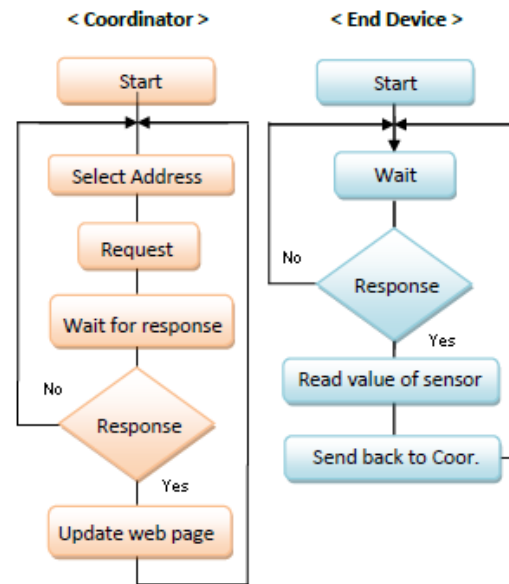


Figure 5. Flowchart of Coordinator and End Device

b) *Polling*: Polling is a method in which the Coordinator requests each End Device one by one to send sensor readings back to the Coordinator. The purpose of polling is to avoid interference from multiple nodes transmitting to the Coordinator simultaneously.

2) *Hardware Specification*: A WSN node consists of a microcontroller, some sensors, and a communication module

a) *Microcontroller*: We use Arduino USB Board which use the chip ATMEGA328 as a processor, controlled by the computer via USB port. Arduino composed of 5 pins Analog I/O and 13 pins Digital I/O which can interface with the other devices such as I2C, SPI, digital and analog.

b) *Sensor*: Temperature Sensor (LM335A) is a semiconductor [8] which measures temperature and displays the information in the voltage form. The

output from the temperature sensor is analog but is then sampled and quantized (A/D converted) by the Arduino. Hence, we can realize the temperature in degree Celsius by calibrating it by using the equation derived from an experiment if we use reference temperature is X degree Celsius and sampling value at reference temperature is Y as shown in equation (1).

$$\text{Temperature} = X + ((\text{analogRead} - Y) * 0.5)) \quad (1)$$

c) *Communication:* We use Xbee Pro 2.5 ZNet [7] in our system. The Xbee Pro module is compliant to the IEEE 802.15.4 standard. It is a series-2 version which supports mesh networking, fixes the address to 64 bits, and has 20 pins.

In summary, the system consists of two types of nodes: Coordinator and End Device, as shown in Figure 4. The Coordinator consists of Arduino, Xbee Pro, Xbee shield, and Ethernet Shield. On the other hand, End Device consists of Arduino, Xbee Pro, Xbee shield, and a temperature sensor.

3) Software Specification:

a) *Microcontroller:* We use Arduino IDE as a code editor and code burner.

b) *Web-Server:* We use a simple HTML/AJAX running on Coordinator.

c) *Database:* We use MySQL [11] to run on client computer. The client computer should has XAMPP Web-Server [10] for Query MySQL command with PHP language accordingly.

4) *System Limitation:* The distance between the Coordinator and End device should not exceed 140 meters from the specification. On the other hand, if there are any blockages, the distance should not be greater than 40 meters. For the sensor, it can measure the temperature to the accuracy of only one decimal point, i.e., the resolution is in 0.5 degree Celsius. MySQL and PHP code have to run on client computer in the same LAN network because Arduino cannot run MySQL by itself.

IV. TESTING

A. Sensor Testing

Every sensor is tested on the protoboard which connected to the Arduino analog input pin number 0-3 in as shown in Figure 6. We observe the sampling values in range 0-1023 to check the sensors. The result is shown in Table I.

B. Field-Test Temperature Measurement

Location: We deployed the Coordinator and End Device within The Activities Center Building, Thammasat University Rangsit Campus at 1st floor. Figure 7 shows the floor plan of the building.

We deployed four sensor nodes far away from each other at about 10-30 meters. We monitor the

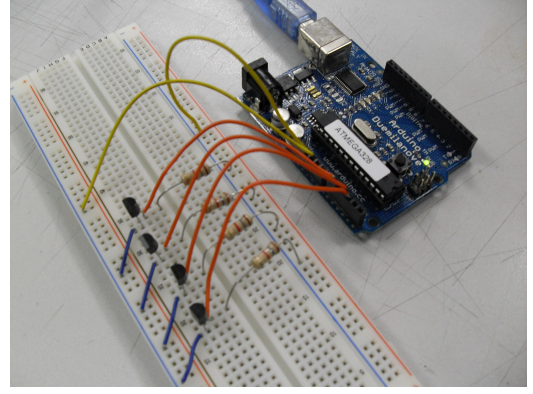


Figure 6. Accuracy and Stability of Temperature sensors Testing

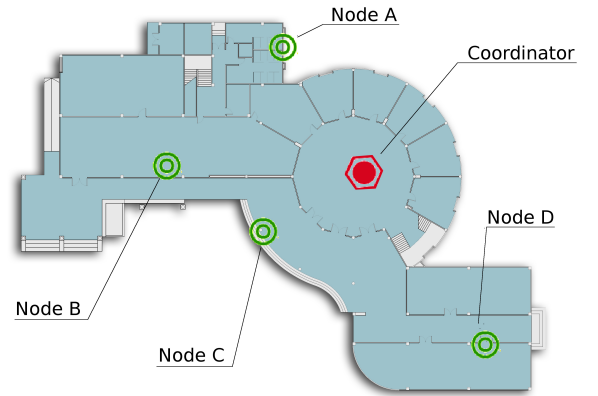


Figure 7. The Activities Center Building infrastructure and nodes' locations

temperature readings from different nodes via a web browser, communicate to Coordinator which is Web-Server and receive the data from each sensor and then monitor via Webpage.

a) *Automatic Polling Experiment:* The Coordinator automatically requests from each sensor in a round robin scheme and displays to the Webpage according to Figure 8. The temperature readings are shown in Table II.

b) *Demand-Based Polling Experiment:* The Coordinator requests from a sensor according to a fixed address and displays to the Webpage as shown in Figure 9.

V. RESULTS

A. Field-Test Temperature Measurement

We performed a field test for temperature monitoring between 8am to 4pm. The system collected the temperature readings every 1 hour and got the result as shown in Table II, got from Database as shown in Figure 8. For the Demand Polling, we got the result as the value at that time of that sensor as shown in Figure 9. From these results, we could say that the system is working in satisfaction in term of its functionality.

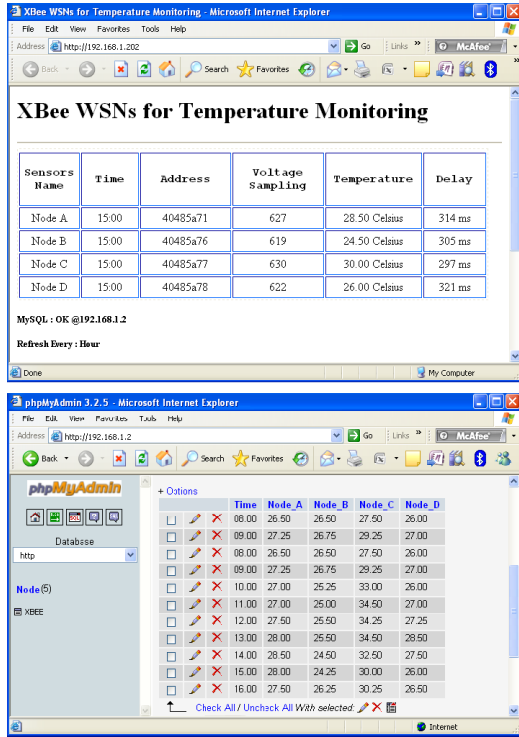


Figure 8. Auto Round Polling via Webpage

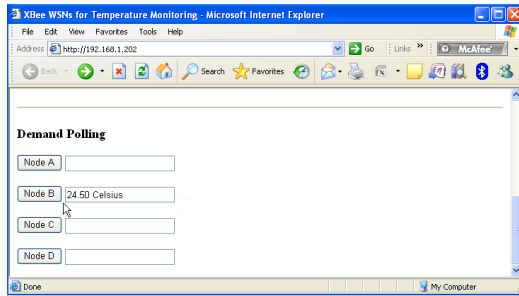


Figure 9. Demand Polling Experiment



Figure 10. End Device Location

Table I
TABLE OF ACCURACY AND STABILITY TEST

	pin(0)	pin(1)	pin(2)	pin(3)
1	619	620	618	619
2	619	620	619	618
3	619	619	618	618
4	619	620	620	619
5	619	620	618	619
6	619	620	619	618
7	619	619	618	618
8	619	619	618	618
9	619	620	619	619
10	619	620	618	618

Table II
TABLE OF FIELD-TEST

Time	Node A	Node B	Node C	Node D
08.00	26.50	26.50	27.50	26.00
09.00	27.25	26.75	29.25	27.00
10.00	27.00	25.25	33.00	26.00
11.00	27.00	25.00	34.50	27.00
12.00	27.50	25.50	34.25	27.25
13.00	28.00	25.50	34.50	28.50
14.00	28.50	24.50	32.50	27.50
15.00	28.00	24.25	30.00	26.00
16.00	27.50	26.25	30.25	26.50

B. Polling time

The time required for each polling to a node can be measured by using the function *millis()*; to realize the distance between Coordinator and End Device. This testing counts the 2-way transmission time. The delay times are shown in Table III.

VI. SUMMARY AND FUTURE WORK

In this paper, we built a prototype of an embedded wireless sensor network based on easy-to-use Arduino microcontroller board and Xbee module. We consider a temperature monitoring application to demonstrate the proof-of-concept of our system. The collected temperature data can be stored into the MySQL Database and retrieved later for analysis.

Extensions of our current work include an extension from a star network to a mesh network which will be useful for deploying sensor networks in large areas like in buildings with multiple rooms and multiple floors.

Table III
RESULT OF POLLING TIME DELAY

# Sensor	Distance	Delay 1	Delay 2	Delay 3	Avg.
Node A	8 m	324 ms	294 ms	312 ms	310 ms
Node B	15 m	311 ms	314 ms	291 ms	305.3 ms
Node C	17 m	309 ms	322 ms	310 ms	313.7 ms
Node D	20 m	301 ms	311 ms	291 ms	301 ms

This mesh networking capability is being explored by our colleague [12].

A cost reduction for each node can be achieved by removing the Arduino board. This takes advantage of the fact that each wireless sensor node can be equipped with an XBee module alone without a microcontroller. This can be done because the XBee module can automatically sample the sensor inputs and report back to the coordinator. It is called Zigbee I/O Sampling that is Xbee module can read sampling values on its pin by itself. So, if we can use only Xbee module, price per each node will decrease.

REFERENCES

- [1] Stefan Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*, Wiley, 2009
- [2] IEEE 802.15 WPAN Task Group 4 (TG4), <http://www.ieee802.org>
- [3] Arduino, <http://arduino.cc/>.
- [4] SparkFun Electronics, Temperature Sensor - LM335A, <http://www.sparkfun.com>
- [5] Wireless Sensor Network Research Group, <http://www.sensor-networks.org>
- [6] Won-Suk Jang, William M. Healy, Mirosław J. Skibniewski, Wireless sensor networks as a part of a web-based building environmental monitoring system, *Automation in Construction* Volume 17, Issue 6, August 2008, Pages 729-736
- [7] Digi International Inc., *XBee ZNet2.5/XBee-PRO ZNet2.5 OEM RF Modules*, Product Manual v1.x.4x - ZigBee Protocol For OEM RF Module Part Numbers: XB24-BxIT-00x, Digi International Inc. 11001 Bren Road East Minnetonka, MN 55343 912-3444 or 952 912-3444 <http://www.digi.com>
- [8] Sabri.C, *Mechatronic*, 1st-Edition, John Wiley & sons, Inc., 2007, ISBN 978-0-471-47987-1
- [9] James F. Kurose, Keith W. Ross, *Computer Networking: A Top-Down Approach: International Version*, 5/E, Pearson Higher Education
- [10] Kai Oswald Seidler, <http://www.apachefriends.org>
- [11] Timothy Boronczyk, Martin E. Psinas, *PHP and MySQL: Create Modify Reuse*, Wrox (May 5, 2008)
- [12] V. Mayalarp, N. Limpaswadpaisarn, T. Poombansao, and S. Kittipiyakul, "Wireless mesh networking with XBee," in *2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010)*, Pattaya, Chonburi, Thailand, 10-12 May 2010.