AC 2007-1072: INTRODUCING ZIGBEE THEORY AND PRACTICE INTO INFORMATION AND COMPUTER TECHNOLOGY DISCIPLINES

Crystal Bateman, Brigham Young University

Crystal Bateman is an Undergraduate Student at BYU studying Information Technology. Her academic interests include ubiquitous technologies and usability. She is currently finishing an honors thesis on using mobile ZigBee motes in a home environment, and enjoying life with her husband and two daughters

Janell Armstrong, Brigham Young University

Janell Armstrong is a Graduate Student in Information Technology at BYU. Her interests are in ZigBee and public key infrastructure. She has three years experience as a Teacher's Assistant. Student memberships include IEEE, IEEE-CS, ACM, SWE, ASEE.

C. Richard Helps, Brigham Young University

Richard Helps is the Program Chair of the Information Technology program at BYU and has been a faculty member in the School of Technology since 1986. His primary scholarly interests are in embedded and real-time computing and in technology education. He also has interests in human-computer interfacing. He has been involved in ABET accreditation for about 8 years and is a Commissioner of CAC-ABET and a CAC accreditation team chair. He spent ten years in industry designing industrial automation systems and in telecommunications. Professional memberships include IEEE, IEEE-CS, ACM, SIGITE, ASEE.

Introducing ZigBee Theory and Practice into Information and Computer Technology Disciplines

Abstract

As pervasive computing turns from the desktop model to the ubiquitous computing ideal, the development challenges become more complex than simply connecting a peripheral to a PC. A pervasive computing system has potentially hundreds of interconnected devices within a small area. This is not only a departure from the typical computer-peripheral model it is also a departure from the typical client-server model.

ZigBee, based on IEEE 802.15.4, is an emerging standard within networked embedded systems. It has already been adopted by several major developers and the availability of devices and support systems is growing rapidly. This standard will become a foundation of future commonplace technologies.

Topics in wireless mesh networking should be presented to Information Technology and Computer Engineering Technology students to ensure they are well-grounded in this emerging area. This paper describes an instructional module. It includes background information on the technology, the key concepts students must understand regarding ZigBee networking, the selection of a development environment, and the design of a hands-on lab experience. We briefly discuss the necessity of teaching this technology.

Introduction

Pervasive computing, first expounded by Mark Weiser[1], embraces technologies that are so commonplace and ubiquitous that they become wholly unremarkable to the typical user. A typical ubiquitous system involves potentially hundreds of small, embedded devices collecting and exchanging environment data with each other. This type of system is typified by the "smart dust" project of Pister and Kahn[2, 3] which envisaged many millimeter-cubed-sized motes collecting data and communicating with each other. Ubiquitous system communications are more complex than the typical client-server model and this new computing model introduces new networking challenges such as devices power conservation, seamless network handling of device addition and removal, and conservative administration expectations.

Wireless sensor networking (WSN) is a key technology for ubiquitous systems. The future widespread availability of wireless sensor networking requires application designers and embedded engineers to be familiar with it and its emerging standards. One such emerging standard is the recent protocol based on IEEE 802.15.4, called ZigBee. This standard is important because it is specifically designed to address the issues of power, self-administration and minimal hardware requirements. It is also rapidly growing in popularity and may become the industry standard in the wireless sensor mesh networking field.

This paper suggests a method of integrating ZigBee into Information Technology and Computer Engineering Technology curricula. First, we briefly discuss key ZigBee networking concepts to introduce students to the protocol. Second, we discuss selection of ZigBee-compliant hardware.

Third, we present a sample lab structure designed to assist students in understanding ZigBee networking fundamentals. Adding content to an already-busy curriculum is always challenging however it is possible to make students aware of the concepts and possibilities of this emerging technology with a single lab experience, combined with some background instruction. This lab and background experience also addresses several basic networking concepts, and can thus be used to replace part of the existing curriculum.

Background

The IEEE 802.15.4[4] standard, which defines the physical (PHY) and media access control (MAC) layers of the network stack, is tailored specifically to low-bandwidth, limited battery devices communicating in peer-to-peer, ad-hoc networks. Building upon IEEE 802.15.4, the ZigBee Alliance created the ZigBee specification to define the network (NWK) and application (APL) layers, and additional security mechanisms[5]. Research assessing the quality of the new specification is prevalent[6,7], interest in creating specification-compliant embedded systems is increasing, and industry professionals are optimistic regarding consumer adoption[8] of wireless sensor networks.

Research into the performance of ZigBee-compliant devices is proliferating. For example, Tseng and Pan conducted a study on 'convergecast' using a ZigBee-compliant network[9]. Ran, et. al determined the most efficient way to implement ZigBee routing in order to conserve power and increase performance[10].

Interest in creating ZigBee-compliant applications is also increasing. ZigBee has been implemented in a wireless meter-reading system[11], in a camera-based human-detection network in which the protocol was made to transmit images[12] and for unmanned air vehicle landing using positioning and vision guidance[13].

Wireless Sensor Networking, a common type of ZigBee implementation, is expected to become a commonplace technology. An IEEE survey assessed the expectations of 500 IEEE Fellows regarding the likelihood of anticipated technologies. 129 of those surveyed responded on the subject of sensors and robotics. Of these 51.9% believed it likely that smart dust devices will "be widely deployed in sensor networks." Further, 29% believed this would likely occur within the next ten years and 40% believed this would likely occur between the next ten and twenty years[8].

In light of the anticipated adoption of ZigBee and the growing body of related research, it is important for Information Technology and Computer Engineering Technology students to be familiar with this technology so that they can be qualified to contribute to the development of these systems both in research and in industry. This paper focuses on introducing students to the networking layer of the ZigBee protocol, drawing upon the IEEE 802.15.4 specification[4] and almost entirely from the ZigBee specification[5].

ZigBee Concepts

The instructional foundation of Zigbee is the networking and application layers of the ZigBee Specification and how they interact to create various network structures. In order to understand this material students are expected to have a prior knowledge of basic computer architectures, basic networking, and some skills in network and program debugging.

ZigBee Nodes

Within a ZigBee mesh network nodes may be tasked with specific network functions. ZigBee defines three types of network node: Coordinators, Routers, and End Devices. A brief description of each follows:

The *ZigBee Coordinator* node is the genesis of a ZigBee network. There may be one, and only coordinator per network. The coordinator is responsible for identifying a suitable channel and selecting a PAN ID (Personal Area Network IDentification) for the network. The coordinator is also responsible for determining which global settings to apply to all network devices including network topology, network synchronization (called 'beaconing'), and network scale. Coordinators house routing and device discovery functions. (Note: A ZigBee Coordinator should not be confused with an IEEE 802.15.4 coordinator. Zigbee coordination is a different function from IEEE802.15.4 coordination. In this paper, Coordinator will refer strictly to the single ZigBee Coordinator which initiates the network.)

A *ZigBee Router* allows the network to be extended over a larger geographical area by allowing other devices, both routers and end devices, to connect to the network and send messages through it. It, like the ZigBee coordinator, contains routing and discovery tables and assigns other devices which it permits to join by assigning a 16-bit network addresses.

End devices are devices on the network which are incapable of routing or allowing other devices to join. The requirements for implementing an end device are much lower than those for a router or a coordinator, allowing them to use fewer resources and consume less power.

ZigBee Network Topologies

ZigBee defines three network topologies: star, tree, and mesh. These topologies are distinguished from one another by the way addresses are distributed, messages are communicated, routing is performed, and whether beaconing is supported. Each is discussed in more detail below.

A *tree* network is the default network topology for ZigBee. In a tree network, communication occurs exclusively within parent-child relationships; a child Router may have children devices of its own, but no child node may have more than one parent and End Devices may not have children. Device addresses are assigned using a distributed addressing algorithm, such that a computation can be performed on any network address to determine if that device is a descendant of the current device. This topology utilizes hierarchical routing in which messages are simply forwarded from parent to child or child to parent. The tree topology supports network synchronization beacons.

In a *star* network, End Devices communicate directly with, and only with, the Coordinator. End Devices may not communicate directly with each other. The Coordinator handles all address assignments. The Coordinator handles all message routing. There are no Routers within the star topology. This topology also supports network synchronization beacons.

Within the *mesh* network, devices are free to communicate in a peer-to-peer configuration. This topology includes a Coordinator node and one or more End Device nodes; there are no Router nodes within a ZigBee mesh network. The device association still requires that a node may have a single parent, but devices are free to communicate with any other device within range. Addresses are assigned to new devices on the network by a user-created application layer. Message routing is not necessarily in strict parent-child fashion, for all of the devices cooperate to route messages within this topology. ZigBee does not allow network synchronizations within this topology, so node duty cycles will not be synchronized and the probability of packet collision increases.

Sample Lab Design

ZigBee encompasses a wide array of networking and embedded systems concepts. We designed this lab based on some of the key functions of the ZigBee system. This makes it possible to add this experience into an existing class. We believe that if students understand basic Zigbee concepts they will be able to learn further extensions as they need them. A small class of senior and graduate students tested an earlier version of this lab design. Based on their experiences with wireless sensor networking and ZigBee, we have modified the lab to include several new experiences. We describe the lab learning objectives, hardware selection, experiment scenarios, and observations the students should make.

Learning Objectives

Upon completion of this lab, the following objectives will be accomplished:

1) Students will understand and demonstrate the connection capabilities of ZigBee mesh networking systems;

2) Students will be able to edit and compile source code for ZigBee nodes;

3) Students will learn to use the microcontroller development environment;

4) Students will demonstrate working code on ZigBee nodes.

Development Environment Selection

When selecting a suitable platform for an academic ZigBee lab several attributes were considered:

- ZigBee compatibility and stack availability
- Hardware and development system adequate to provide a meaningful experience for students.
- Cost of hardware and available education discounts also were considered, but will not be discussed within this paper.

The most distinguishing factor when selecting ZigBee hardware was if a ZigBee-compliant stack was written and available for the hardware. Many vendors offer ZigBee Alliance approved radio transceivers and systems, but relatively few at the time of hardware selection offered a publicly available stack. We felt it important to stay with a standard Zigbee solution so students could relate Zigbee theory to the lab experiences.

We considered Zigbee development systems from Crossbow, Freescale, Ember, and Microchip We selected the Freescale MC13213[14] NCB and SRB kit for the reasons discussed above. When this lab was developed, Freescale offered the Figure 8 Wireless ZigBee stack and software, but Freescale has since ceased to support the Figure 8. Freescale now offers their 802.15.4 BeeKit with ZigBee-compliant BeeStack[15].

We discovered additional concerns while working with the selected hardware. Often network and software debugging was difficult with the limited feedback capability of the motes – we had little confirmation if the mote was sending or receiving data. The inclusion of an LCD screen, push buttons, LEDs, and the ability to communicate over serial or USB between the mote and a desktop computer, together with software to drive them, greatly alleviated these potential difficulties. The motes in the development system were considerably larger than they would appear in a final product. They came in a plastic case and were powered by AA batteries. The cases made them easier to handle and relieved possible concerns of physical or static electricity damage.

Lab Scenarios

Three scenarios were designed to illustrate to students the essentials of ZigBee networking including joining the network, moving location within the network, and node deletion and route repair within the network. Each of these scenarios was developed for use with a tree topology in a beacon-enabled network. The tree topology was selected for its ability to conserve power by allowing devices to sleep between synchronization beacons and its allowance of End Devices to communicate with Routers in addition to the Coordinator Node.

Scenario 1 – Automatically Joining the Network

This scenario tests ZigBee automatic network joining. A device will enter the range of an End Device on the edge of the network and attempt to join the network; as End Devices cannot have children the attempt will fail. Next, the device will be placed within range of a ZigBee Router and again attempt to join the network. This time, the device will identify the router as a potential parent, successfully join the network, and begin receiving Coordinator-broadcast messages.

Scenario 2 - Moving locations within the network

This scenario demonstrates what happens in the event of a node moving to a location point out of range of its parent device, but remaining within range of other devices in the network. This lab assumes a tree topology in which communication only occurs through parent-child connections and only a Coordinator or a Router may be a parent. The node will attempt to send a message to its parent device. After being moved out of its parent's range the node will lose synchronization with the parent beacon. The nodes will then identify new potential parents, rejoin the network, and pass messages through its new parent. Students need to check that a Router is within range of the moved node, or the attempt to rejoin the network will fail.

Scenario 3 - Deletion of a node (route recovery)

By deleting a Router which is a parent to other nodes, this scenario will illustrate the ability of ZigBee to participate in route repair and continue network operation. For this scenario, a node will randomly be powered down and students will observe the effects on the network. The coordinator will issue a periodic broadcast during this time informing its children to toggle an LED on and off so that there will be visible evidence of whether a device is in communication with the network. This scenario should be repeated with both an End Device and a Router. Theoretically, the Coordinator may be turned off and the network will still function, so the scenario may be repeated with the Coordinator as well.

Student Lab Procedures

To accomplish the above scenarios, students will program the motes with appropriate stack and application code, position or move the programmed node according to the scenarios given, observe the results, and record their observations in a lab report. Students should record the networking structures and connections made during the course of each procedure to ensure they understand how the automatic connections are being established.

Node Programming

The instructor must perform certain procedures to reduce the time necessary for students to accomplish the lab scenarios and increase the success rate of inexperienced students. Prior to the lab, the ZigBee Coordinator and stationary Routers must be configured and programmed. These nodes should be placed such that there is minimal range overlap between alternating devices. This network preparation enables students the time necessary to design, program, and initiate the ZigBee network.

Each student will program an End Device equipped with an LCD screen. The screen may display network status messages as they are received such as where the node is at in the joining process or when an acknowledgement message is not received. The messages will aid the student in observing ZigBee networking processes as they occur in the provided scenarios. It is recommended that this code be provided to the students to avoid overwhelming them as they become familiar with ZigBee.

In order to understand basic ZigBee event processing and configuration, students will be provided with ZigBee stack code. First, each student must ensure the network starts as a tree network by editing the pertinent header file. Second, each student will compile the code for the proper device type and download the code to the target device. Third, each student will be asked to toggle an LED whenever a message is received containing a certain value or command string, such as "HELLO_WORLD".

Student Lab Observations

In the first scenario, students will place their motes at differing locations within the pre-existing network and observing the join process through a series of LCD messages. They should observe, in addition to verifying that ZigBee is able to handle automatic joining to the network, that in a tree network, another node cannot join through an end device, but must join through a router or the coordinator.

In the second scenario, since a tree topology is being used with beaconing enabled, it will attempt to rejoin the parent as soon as the child device loses synchronization with the beacon of its original parent. Upon failure to connect to the parent, the node will rejoin the network through a new parent. Students will observe this progress through the LED and LCD messages displayed by the child node.

In the third scenario, students will be able to make observations depending on which of the varying nodes were powered down.

- If an End Device is powered down, there should be almost no effect on the network.
- On the other hand, if a Router with children is powered down, route repair will in largely depend on whether there is another suitable Router within range.
- If no parent is found then that branch of the network will be cut off from the rest of the network.
- If there is a suitable parent, the devices will re-join the network and begin receiving Coordinator broadcast messages again.

These conditions will be observed by interpreting the LED indicating packet transmission status and LCD status messages.

The learning that students will experience will come from a combination of observations and implementing the various experiments. The networking capabilities of the Zigbee system will be obvious from the observations.

Conclusions

It is likely that ZigBee will increasingly play an important role in the future of computer technology. By breaking this complex protocol down and focusing on only the networking aspects in introducing ZigBee to students, it is hoped that wireless sensor network concepts will be more easily integrated into the classroom, and will lead the way for further exploration in a classroom setting. While this paper includes a lab design for networking, it would be advantageous to subsequently introduce concepts of ZigBee application design and development in future labs so that students can appreciate more fully the features that ZigBee has to offer.

Introducing ZigBee and other wireless sensor networking principles into the standard IT curriculum will furthermore help to expand the field of current research in this area and will stimulate improvement in the standards involved as concepts are put into practice. Adopting new technologies such as ZigBee will also spur the development of useful applications that may change future quality of life and the way people look at computing itself.

Bibliography

1. Weiser, M. (1991). The Computer for the 21st Century. *Scientific America*, September 1991. 94-104.
2. Kahn, J.M., Katz, R.H., Pister, K.S.J. (1999). Next century Challenges: Mobile Networking for "Smart Dust" Proceedings of the 5th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)
3. Pister K. S. J., Kahn J. M., and Boser B. E. (1999). Smart dust: Wireless networks of millimeter-scale sensor nodes. In 1999 UCB Electronics Research Laboratory Research Summary
4. *IEEE 802 Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks*, IEEE Computer Society, 2003.
5. *ZigBee Specification v1.0*, ZigBee Alliance, December 14th, 2004.
6. Tanenbaum, A. S., Gamage, C., & Crispo, B. (2006). Taking sensor networks from the lab to the jungle. Computer, 39(8), 98-100.
7. Kohvakka, M., Kuorilehto, M., Hännikäinen, M., & Hännikäinen, T. D. (2006). Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications. Paper presented at the Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks, Terromolinos, Spain. 48-57.
8. Gorbis, M., & Pescovitz, D. (2006). IEEE fellows survey: Bursting tech bubbles before they balloon. *IEEE Spectrum*, 43(9), 50-55.
9. Tseng, Y., & Pan, M. (2006). Quick convergecast in ZigBee/IEEE 802.15.4 tree-based wireless sensor networks. *MobiWac '06: Proceedings of the international workshop on Mobility management and wireless access*, 2006, 60-66.
10. Ran, P., Sun, M., Zou, Y. (2006). ZigBee routing selection strategy based on data services and energy-balanced ZigBee routing. *APSCC '06,* December 2006, 400-404.
11. Chen, B., Wu, M., Yao, S., & Binbin, N. (2006). ZigBee technology and its application on wireless meter-reading system. *Industrial Informatics, 2006 IEEE International Conference on*, August 2006, 1257-1260.
12. Schlessman, J., Shim, J., Kim, I., Baek, Y. C., & Wolf, W. (2006). Low power, low cost, wireless camera sensor nodes for human detection. *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, October 2006, 363-364.
13. Jiang, Y., Cao, J., & Du, Y. (2006). Unmanned air vehicle landing based on Zigbee and vision guidance. *WCICA 2006*, 2, 10310 - 10314.
14. Freescale Semiconductor, I. (2006). MC13211/212/213/214. Retrieved 03/07, 2007, from http://www.freescale.com/files/rf_if/doc/data_sheet/MC1321x.pdf
15. Freescale Semiconductor, I. (2007). BeeKit wireless connectivity toolkit. Retrieved 03/07, 2007, from http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=BEEKIT_WIRELESS_CONNECTIVITY_TOOLKIT&parentCode=1321xEVK&nodeId=01J4Fs25657103