Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Sensing and Feedback Stimulation via The Wireless ZigBee Protocol

Harding D. Chati* and Fathi M. Salem, *Fellow, IEEE*

Abstract—**The aim of this work is to implement a feedback sensing and control mechanism via wireless sensor nodes for medical sensing and actuation. To that end, we employ the Zigbee protocol in sensing and precisely commanding an analog signal to a probe. We employ a user application upon the layer available in the Zigbee stack to achieve our goal. Signal transmission delays, packet losses, and energy consumption are major challenges, we present strategies to minimize or solve these challenges. We also introduce strategies to enable sensor nodes to acquire and command (electrical current) signals from/to sensing/actuating platforms. Finally, we implement algorithms allowing the sensor node to compute and to regulate command signal on line.**

I. INTRODUCTION

Wireless sensor nodes have emerged recently as an attractive frame-work for low-power effective platforms for sensing and actuation. A typical wireless sensor node contains one or many sensors and/or actuators, additional microcontroller, and a wireless hardware to communicate with other sensor nodes. Thus, it becomes easy to collect data from an environment and to act on this environment. To allow the interconnectivity and communication between nodes and different components of a node, many protocols have been developed. However, these protocols present some restrictions. One of these is that they require many resources; and, therefore, the sensor node consumes too much energy and cannot be active for a long period of time. Another problem is the delay in the data acquisition and transmission, or the loss of data both due to the wireless communication and/or to the employed protocol. Thus it would be difficult to use these Wireless sensors in a real time environment where we generally deal with time critical measurements and/or stimulations. Recently, the ZigBee protocol has been promoted as a solution low power platform for sensing and control. The ZigBee protocol offers low complexity, reduced resource requirements and most importantly, a standard set of specifications.

The goal of this work is to develop a sensing and (on-line) feedback actuation for (heating) probes using the ZigBee protocol and its microchip stack as a wireless sensor node framework. The developed framework must function within the presence of delays and possible loss of packets/data. Additionally, considerations must be made to minimize the power/energy consumption at the node.

The remainder of this paper is structured as follows. We give an overview of the ZigBee Protocol employed. We also describe the experimental results in deploying an on-line feedback sensing and control for the wireless sensor node based upon the related protocol, which takes into account the specified problems and achieves an desired regulation solution.

II. BRIEF OVERVIEW OF THE ZIGBEE PROTOCOL

The ZigBee protocol is a wireless network protocol, related to the IEEE 802.15.4, and has been promoted as a simple, flexible, low speed and low latency transmission standard. It also offers low power consumption, reliability and security for control and monitoring for applications with low to moderate data rate. It is above all designed for relatively low-rate sensor and control wireless networks and can be used in many domains such as building automation, home security systems, industrial control networks, remote metering and PC peripherals. *In this work, we focus on its utility within the medical field and specifically in monitoring and maintaining temperature via a probe or other devices.* The advantage of the ZigBee protocol compared to other wireless protocols is that it is simple to use and it offers reduced resource requirements and a standard set of specifications. This provides a common platform and standard for developing wireless sensor applications in the medical domain. Here, we extend its utility to applications where on-line monitoring and command control are repeatedly applied to maintain signals (e.g. temperature) at a desired level.

In our application, we focus on the *Star topology* which consists of one coordinator node (C-node) and one or many end devices (E-nodes). All E-nodes communicate to the coordinator only. An E-node which desires to send data to another E-node, will first send its data to the coordinator which in turn forward the data to the intended receiver end device (E-node). Fig. 1 below sketches the non-Beacon communication mode where end devices (E-nodes) check for free channels. Once an end device acquires a free channel it either sends its data (or request for data to be sent) to (the) Coordinator node.

Manuscript received April 24, 2006. This work was supported in part by the MSU IRGP award, app# 03-42191 and by IC Tech Inc. in Okemos, Michigan 48864.

H. D. Chati was a visiting student scholar of the Department of Electrical and Computer Engineering of the University of Kaiserslautern, Germany, at Michigan State University, East Lansing, MI 48824 USA

F. M. Salem is a Professor at the Department of Electrical and Computer Engineering at Michigan State University, East Lansing, MI 48824 USA (Phone: 517-355-7695; fax: 517-353-1980; e-mail: salem@egr.msu.edu).

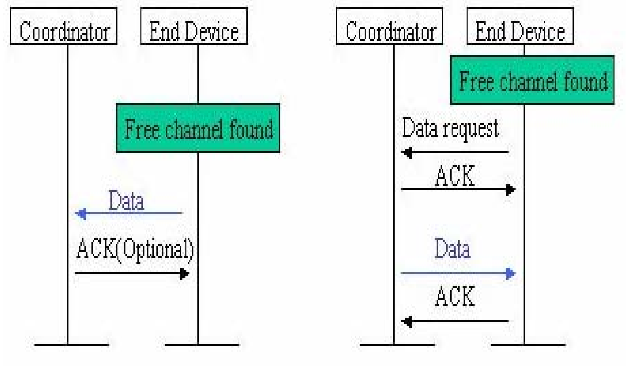


Fig. 1: Data transfer from/to a coordinator and to/from an end device in a non-Beacon network

III. DESCRIPTION OF THE OVERALL FORMUALTION

Within the framework of the Star Topology, we choose a coordinator leading one to several E-nodes, each performing sensing and/or on-line feedback actuation to control and maintain signal at desired levels. The challenge is in distributing the computational effort among the devices and allow for the presence of delays in signal transmission. Despite its advantages, the wireless sensor nodes framework also presents limitations. Examples include the delay in (loss of) the data transmission characteristic of wireless communication, and their excessive consumption of energy in certain protocols.

The challenge is to develop an on-line feedback control in a wireless environment with sensor nodes based on the ZigBee protocol and its microchip stack. The approach must be able to react to delay and loss of data due to the wireless communication. Additionally, the end sensor node must consume a minimum of energy to be able to work for a long period of time.

In our application, we desire to maintain a constant voltage value at the input of a probe in order to maintain temperature at a constant level. To this end, the actual temperature has to be measured continuously (or periodically) by a sensor on the E-device. These values will be regularly sent to the coordinator wirelessly. A user on a PC sets the reference value and passes it to the coordinator. The coordinator compares the received value from the end device with the reference value, computes the difference and wirelessly sends it to either the same, or another, end device within the network. The end device processes the error and produces the digital (binary) output values at 8 digital (output) pins. An external D/A converter converts the digital 8-bit value to an analog value that drives the probe.

The following implementations constitute the main work:

1) Implementing a user layer upon the ZigBee protocol satisfying our requirement and enable it to communicate with the sub layers of this protocol. The emphasis will be placed on designing strategies to minimize delay in data transmission and loss of data frame.

2) Implementing processing and control algorithms allowing the end device to receive sensor data from their pins, and to be able to set 8-bit digital value onto 8 output pins simultaniuosly, which will then be converted by a D/A converter into an analog actuation value.

3) Implementing algorithms to compute and to regulate the error between the received value from the sensor and the desired value provided by the user.

4) Allowing data to be displayed graphically onto PC to/from the device (s). So it will be easy to configure the devices (setting the ID on a device, setting a reference voltage value, join a new node, etc) and to see the actual state of the applications on the device.

5) Use an 8-bit D/A converter, or builds a custom one, and connects it to the 8 pins on the end device.

6) Use a sensor to capture the voltage values corresponding to temprature, which will be then sent to the end device and connect this sensor to the end device.

We will use the ZigBee protocol and its microchip stack to:

a) Allow comfortable and reliable communications between nodes and different components on a node

b) Take advantage of its set of specifications

c) Allow an optimal battery use on the device

d) Additionally, we will not need to re-implement the sub layers (Application Layer, Application Support sub layer, Network Layer, Medium Access Control layer and the physical layer) of the protocol and concentrate instead on the user layer and communication between the user layer and the sub layers. It is necessary to understand the sub layers and their source files and interfaces in the microchip stack.

IV. HARDWARE ARCHITECTURE

To satisfy our hardware requirement, we will expand the ZigBee node details [1-6] as depicted in Fig. 2. On the E-device, a sensor is mounted to collect temperature values represented as voltage signals. Additionally, we will mount an Actuator, which includes an A/D converter and the actuator itself, and connect this Actuator onto the E-device.

We now describe one architecture. It consists of one coordinator and one E-device. An actuator and a sensor are built on the E-device as shown in Fig. 3. There is a direct connection between the coordinator and the PC. This is achieved via a DB-9 male-to-female RS-232 cable that connects the two devices.

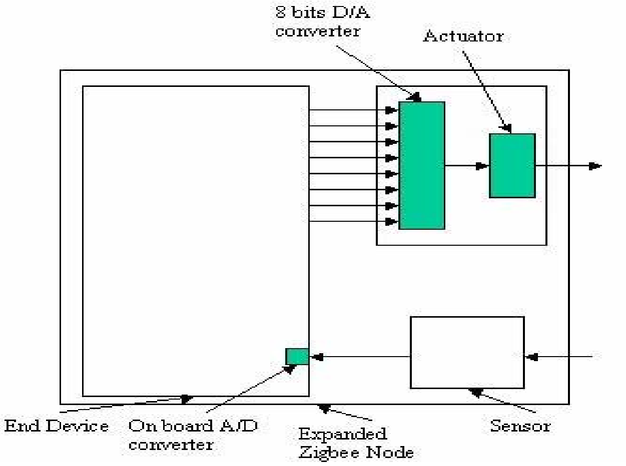


Fig. 2: Expanded ZigBee node

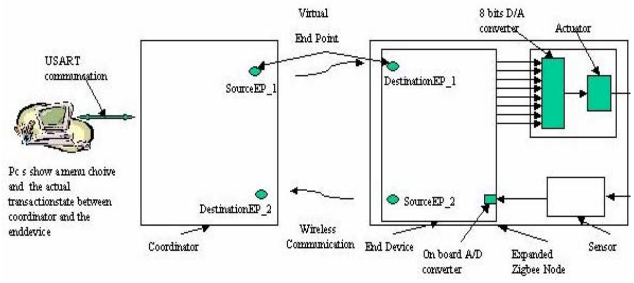


Fig. 3: Hardware architecture

To allow communication between applications, the ANSI C language provides the well-known header file concept. All API's (in particular those necessary for interfaces, communications, and services between layer and implemented in the source files) must be declared in the header file. Thus, applications of a layer that will need to interact with given applications of the sub layer of this layer, will only need to include the header files corresponding to these sub layers in its own files. Applications present in the microchip stack and the interaction between them are built on the same principle.

V. EXPERIMENTAL RESULTS AND CHALLENGES

The goals of the simulations are to:

(i) Test the accuracy and the effectiveness of the designed control.

(ii) Measure the delay time in one processing cycle (The time that the end device takes to sense a value from the sensor + the time used to send this value to the coordinator + the time taken by the coordinator to compute the error and sent this error back to the end device + the time taken by the end device to put the error on its pins and to transform this error to an analog value)

(iii) Measure the real range for an accurate and reliable wireless communication between the used nodes (coordinator node and sensor node)

To achieve these goals, we use the following closed loop system configuration:

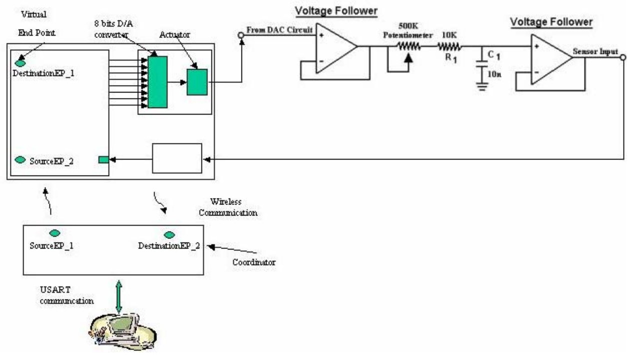


Fig. 4: Schematic diagram of the closed-loop circuit.

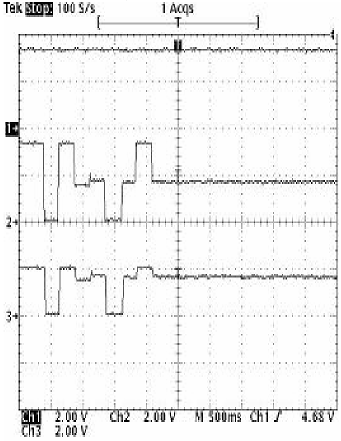
We regulate the error based upon the following PID control formula:

$$X = Const + (k*E) + k1* Div\_E + k3* Intg\_E$$

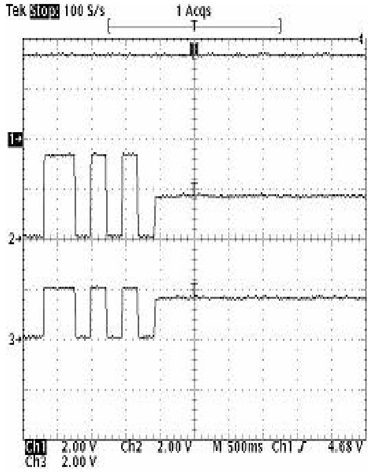
Where:

X: is the value to be send to the E-device; Const and k: are constant values; E: is the difference between the sensor value and the, say, X1, X2 or X3 reference value; Div_E an estimate of the digital derivative of E, and Intg_E is an estimate of the integral (sum) of E.

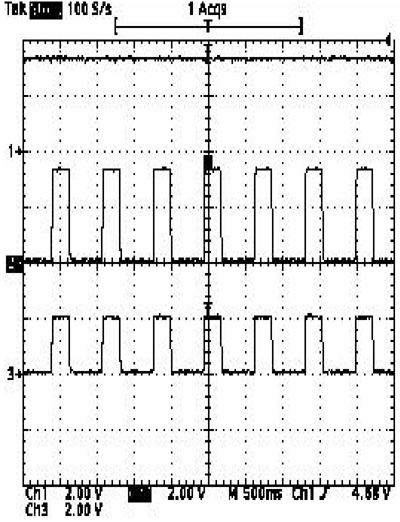
We use only one reference value (X1). This is to be sure that the feedback control functions as expected for one reference value. We conduct five tests with five different k values. We use in each test the same k values (k=1, k=2, k=3, k=5, or k=10) and tree different X1 reference values (X1=122, X1=128, or X1=133). Then, we observe on the oscilloscope the following plots for the closed loop feedback system: (Channel 1 is V_{DD}, or 3.3V; Channel 2 is the actuation signal heating the probe; Channel 3 is the Sensor measurement.)



Test_1. K=1, X1=122-- convergence



Test_2. K=2, X1=122-- convergence



Test_3. K=3, X1=128—oscillations due to signal delays

Fig. 5: Closed-Loop Graph of sensing and commanding a constant level temperature.

As can be seen from the plots, as the actuation signal from the DAC changes, the sensor output almost immediately responds with a change of its own. The delay time is virtually impossible to obtain just by viewing the graph alone. According to the LM324 datasheet, it can be seen that the rise time, t_r, and the delay time, t_d , are approximately 10 microseconds (us). For our purposes, it is safe to assume that these delays are no more than 100 us at any given time for this particular system. This delay does not include the sleep time on the E-node, which is about 256 ms.

The results of these tests show that the on-line feedback control through the wireless sensor network is realized and is effective. The actual distance range used for this accurate and reliable wireless communication between the coordinator and the E-device is 300 and 350 inches.

The following problem has been observed during the experiment. Sometimes, the firmware on the E-node goes into an unexpected state and stops the control. But when it comes back in the normal state, the control continue due to the ability of the E-device to rejoin the coordinator anytime that this coordinator is on and the E-device has performed its **join** procedure. This problem is probably due to a WDT (wachtdog timer) that expires too early before being cleared and causes a reset of the firmware on the E-node. Increasing the WDT value can solve this problem. However, in this case, the sleep time will rise. More investigation must be done to solve this problem with an acceptable and low WDT time.

VI. CONCLUSION AND FUTURE WORK

An on-line feedback actuation control with wireless sensor node network has been successfully implemented. The nodes are able to sense, to compute and regulate error occurring during the control action, and to actuate data (here voltage) on the probing device(s). As shown with the experimental plots, the control through wireless sensor node is effective and can be used as a replacement of the control with electric wires.

The experimental demonstration has shown that the nodes are able to produce an acceptable control of the voltage in closed loop configuration-- wirelessly. The problems noted during the experiment could be part of a future improvement of the network. Thus, further investigation is needed to resolve the problem of the WDT (wachtdog timer), which expires sometime too early before being reinitialized. Other configuration that utilizes multiple units can be investigated and compared with existing configuration. Moreover, distribution of efforts among coordinator and E-devices has to be exploited for a given architecture to achieve effective feedback actuation and control.

REFERENCES

[1] Microchip, *PICDEM Z Demonstration Kit User's Guide*
[2] Microchip, *Microchip Stack for the ZigBee Protocol*
[3] IEEE, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Network (LR-WPANS) " *IEEE 802.15.4 specification*
[4] Microchip, *PIC18F2525/2620/4620 Data Sheet*
[5] Y.C Tseng, *ZigBee/IEEE 802.15.4 Overview*
[6] Anis Koubaa, Mario Alves, Edouardo Tovar, "IEEE 802.15.4 for Wireless Sensor Network: A Technical Overview" at www.paperplaza.net/support/support.html.