Android controlled Zigbee motes for Wireless Sensor Networks

B. Sree Charan Teja Reddy, N. Sai Teja and G. V. V. Sharma
Electrical Engineering Department
Indian Institute of Technology, Hyderabad
Yeddumailaram, India 502205
Email: {ee12m1010, ee11b022, gadepall}@iith.ac.in

***Abstract*—This demo describes the implementation of a peer to peer message chatting application in the android platform over Wireless Sensor Networks (WSNs). Using the IIT-H Zigbee motes we show how to interface the android platform with the Zigbee motes and the different modes of communications involved.**

I. Introduction

WSNs are primarily used in applications where low bit rate, low cost, low complexity and more lifetime are required. WSN nodes usually consist of an 8 or 16 bit microcontroller and a short range transceiver in order to create a low cost network with maximum lifetime. Traditionally, WSNs have been used for sensing and reporting. Recently, however, there is a great demand to support a short range communication system in WSNs for use in educational institutes, residential areas and even in emergency situations. Voice over WSNs was implemented in [1]. However, this implementation was done on proprietary hardware. In this demo we use the android platform along with Zigbee protocol both of which are quite popular and open in nature.

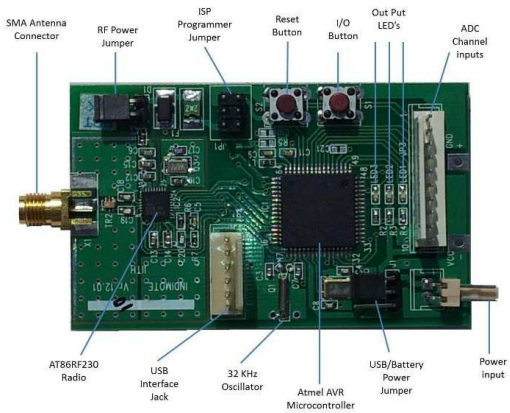

Fig. 1: IITH-mote

In this demo, we use IIT-H motes as nodes in the network (see Fig. 1). IITH-mote is a 2.4 GHz mote used for enabling low-power WSNs. It works on 8 MHz atmega microcontrollers with 8 KB RAM. It has a 6 pin usb interface jack for serial communication and uses IEEE 802.15.4 [2] compliant RF (Radio Frequency) Transceiver for radio communication. IIT-H mote supports two ways of communication. One is wireless communication through RF transceiver. The other way is wired serial communication through Universal Asynchronous Receiver/Transmitter (UART) [3] port. IITH-mote can be programmed using TinyOS [4] or assembly language. We use TinyOS platform for programming the motes.

TinyOS is a free and open source software used for embedded systems. One special feature of TinyOS is, it is a component-based operating system. We can structure the problem into component implementations and reuse them. This is the optimal way to design and combine the code. NesC (Network Embedded System C) [5] is the programming language used in this embedded operating system. TinyOS supports a variety of microcontroller families and radio chips. It efficiently manages the limited power available and renders excellent support for wireless networking.

II. Demo

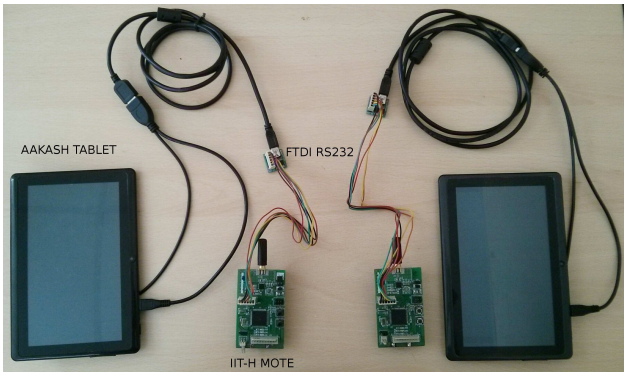

Fig. 2: Text messaging between two android devices (Aakash tablet) through IIT-H motes

Fig.2 shows peer to peer communication between android devices.Here the android devices used are Aakash tablets [6] The communication between motes is based on the Zigbee standard. The IITH-mote is connected to the android device through microUSB serial port.

A. Radio communication

Radio communication between the IIT-H motes is done through packets. This packet based communication is completely controlled by the TinyOS application.

TinyOS application: A basic TinyOS application for radio communication contains 4 files.
1. Configuration file
2. Module file
3. Make file
4. Header file

Configuration file contains the components that are used on IIT-H mote. Module file contains variables, code logic and interfaces. Makefile allows platform selection and includes rules required to make the application. Header file contains packet structure which is important. Following is an example of packet structure.

```
#ifndef PACKET_STRUCTURE_H
#define PACKET_STRUCTURE_H
typedef nx_struct AmMsg {
nx_uint8_t counter;
} AmMsg;
#endif /* PACKET_STRUCTURE_H */
```

In the above example, packet contains a single identifier named 'counter' of `nx_uint8_t` data type. Data type `nx_uint8_t` in TinyOS is equivalent to unsigned character data type. The default maximum payload size of a packet in TinyOS is 28 bytes. In this demo we use maximum payload size.

B. Serial communication

Serial communication is helpful in communicating with external devices. In general, a WSN will have a gateway for collecting data, observing network traffic and sending commands to the network. A gateway is nothing but a computing device connected with a node of that WSN. Here the communication between node and computing device is serial (UART) communication. TinyOS provides two different types of serial communication. They are packet based and packet less serial communication.

1) Packet based serial communication: In Packet based serial communication, both computing device and mote communicate using packets. TinyOS platform on computing device provides different tools for packet based serial communication. These tools help in creating, sending packets to the mote, parsing the packet sent by mote and analyzing the data.

But the disadvantage of this mode is that there are no sophisticated tools in android platform for supporting packet based communication. Hence we use packet less serial communication between IIT-H mote and tablet.

2) Packet less serial communication: In this mode raw data sent from tablet is converted to serial data by RS232 FTDI chip. It does not follow any packet structure. Every serial communication will have a baud rate and IITH-mote supports a baud rate of 57600. For this purpose an android application called B2B was developed for this demo. This application is developed based on a demo application 'TN 147 java D2XX' [7] provided by FTDI. B2B will automatically detect the device whenever mote is connected to the tablet. This application has default settings of baud rate, data bits and parity bit to be used. B2B writes the message to the micro usb port byte by byte. RS232 chip receives the bytes and convert them into UART format and sends byte by byte serially to the mote.

C. Message structure

While creating the WSN each node is given a unique identification called node ID with which user is identified. This node ID is used as source address and destination address while messaging. B2B application provides a text field for writing the message. After completion of writing, touch send button next to the text field to send the message. Text messaging follows a message structure shown in Fig. 3.

Fig. 3: Message structure

III. Advantages

ZigBee provides low cost and low power connectivity devices where lifetime lasts for months but does not provide bit rate as high as those enabled by Bluetooth. But the bit rate provided by Zigbee is sufficient for text chatting. Zigbee uses an unlicensed globally compatible ISM band ranging from 2.4 GHz to 2.4835 GHz. So a low cost communication system can be developed for use in residential areas. Since the android platform is now interfaced with WSNs, it should be possible to incorporate this in mobile devices on a large scale at low cost.

IV. Conclusion

In this demo we implemented a peer to peer message chatting application in android platform over wireless sensor networks. Since we use low power nodes with IEEE 802.15.4 compliant RF Transceiver for radio communication, the range of a node is limited to few tens of meters. The range of the network can be increased by using multiple hops in between. We can implement different topologies with good networking to cover larger areas. The realization of such a network in residential areas should be the focus of future research.

V. Acknowledgment

We thank Department of Electronics and Information Technology, Govt of India for funding this research.

References

[1] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over sensor networks," *Real-Time Systems Symposium*, 2006.

[2] J. T. Adams, "An Introduction to IEEE STD 802.15.4," *Aerospace conference*, December 2006.

[3] http://tutorial.cytron.com.my/2012/02/16/uart-universal-asynchronous-receiver-and-transmitter/.

[4] P. Levis and D. Gay, *TinyOS Programming*, 1st ed. Cambridge University Press, July 2009.

[5] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesc language: A holistic approach to networked embedded systems." [Online]. Available: http://www.tinyos.net/papers/nesc.pdf

[6] http://www.akashtablet.com/akashview.php.

[7] http://www.ftdichip.com/Support/SoftwareExamples/Android_JAVA_D2XX_Projects.htm.