

Project Report 3

**MoteComm: Implementing Active message
communication between mote and zigbee board.**

Submitted By

**Trushant A Kalyanpur
Manan M Shah**

North Carolina State University

Table of Contents

Milestones Proposed in the previous report:.....	3
1st week (Nov 1 – 7).....	3
2nd week (Nov 8-14)	3
3 rd week (Nov 15-21).....	3
4 th week (Nov 22-28).....	3
Milestones Achieved:.....	3
1) Establishing communication between the M16C board and the Mote:.....	3
2) The data packet received on the mote is transferred over the UART and displayed on the PC:.....	4
Problems Faced.....	4
1) Transferring data from the Zigbee board to the mote:.....	4
2) Comparing the message formats received by the board and the Java program:.....	4
Proposed Milestones:	5
1 st Week (Nov 15-21).....	5
2 nd Week (Nov 22-28).....	5
Individual Member contribution:	5
Trushant:	5
Manan:	5

Milestones Proposed in the previous report:

In the previous report we had proposed the following milestones:

1st week (Nov 1 – 7)

We will be to try to send some data from the mote to the zigbee board which will continuously generate interrupts.

2nd week (Nov 8-14)

The next step will be to send some data from the zigbee to the mote and try to parse the various fields received.

3rd week (Nov 15-21)

We will try to send data from the zigbee board to the mote and interpret it.

4th week (Nov 22-28)

We will write a sample application for bidirectional communication between the mote and the zigbee board.

We have worked on the following since the previous milestone:

Milestones Achieved:

1) Establishing communication between the M16C board and the Mote:

When we submitted the last milestone we were unable to establish communication between the M16C board and the mote. The problem was corrected and the following two reasons caused it:

1. The first was that the Zigbee board which we were using was faulty. Since we had two Zigbee boards we tried to transfer data between the two and we observed that only one of them functioned correctly as the coordinator in the Personal Area Network (PAN).
2. The second reason was that the two devices were communicating on a different channel.

The mote was using Channel 11 whereas the board was using Channel 26 for communication.

A sample application was written to blink the LED's on the M16C board upon reception of any data from the mote. The mote was used to transmit data continuously and we verified the communication by the glowing LED's on the M16C side. Upon reception of a MAC data frame, the MAC layer on the M16C side issues a call to the application layer via the *mcpsDataIndication()* function. We controlled the blinking of

the LED's in this function in which we toggled the green LED upon multiple entry into the *mcpDataIndication()* function.

A challenge we now face is to try and extract the same message which is transmitted by the motes since the packet structures on both sides is different.

2) The data packet received on the mote is transferred over the UART and displayed on the PC:

We integrated the receiver application with the UARTComm, to transmit and display the data packet received by the mote on the PC. The mote now successfully receives the packet from the other mote and displays the packet contents on the data screen.

The first time we used the GenericComm interface to send the data to the PC, however it became difficult to differentiate between the data packet received from the Java program and the data packet received from the radio interface. So we used the UARTComm interface to send and receive messages to the PC and then Generic Comm Interface to send and receive packets from the radio interface. We had to write a new interface which could handle the passing of packets from the RfMote component to the TempToUart component.

One interesting problem we faced while transferring the packets is that the Java program can only accept 24 bytes at a time so we had to divide the packet into 2 parts. In the first part we transmit the fields of the structure and in the second part we transmit the data part of the structure.

Problems Faced

1) Transferring data from the Zigbee board to the mote:

When data is transferred from the Zigbee board to the mote, the mote does not seem to generate any interrupts. We tried checking the frequency and it was observed that the frequency was the same. The next part we noticed was that mote was discarding some types of packets of shorter lengths received from the radio interface, so we commented out that portion of the code but yet there were no interrupts generated.

Solution: We propose to study the method in which the packet is formatted when it is transmitted from the mote and at the same time we also want to compare it the way it is done on the zigbee board. We will also be comparing the chip initialization procedures to check the differences and see if it makes a difference in packet reception.

2) Comparing the message formats received by the board and the Java program:

On the M16C side we used the KD30 debugger to view the content of the packet received by the M16C board. On the mote side we used the Java program to examine the content of the TOS message sent by the mote. The Java program shows a byte in the payload field being incremented during successive transmission and the packet on the M16C side also shows a particular byte being incremented. We were unable to compare the data on either side since the message structures were significantly different.

Solution: We will have to study the format in which the packet is exactly transmitted and the format in which it is received and possibly modify the data structure on one of the sides.

Proposed Milestones:

1st Week (Nov 15-21)

- 1) Study how the mote transmits the data packet on its side and make the Zigbee board match the same format.
- 2) Study the datasheet of the radio transmission chip used and study how two chips actually communicate and then determine if the chips on both the sides are in the same mode of communication.
- 3) Extract the TOS message information on the M16C side.

2nd Week (Nov 22-28)

Write an application to transfer data between the mote and the Zigbee board.

Individual Member contribution:

Trushant:

Milestone 1: Played a key role in studying of the zigbee code and identifying the functions involved when a message is received or sent.(Milestones 5 through 7)

Milestone 2: Played a key role in identifying the problems on the board side, he identified the problems associated with the zigbee board identified during the last milestone; the interrupts which are triggered when the data was received. Currently trying to determine how to parse the data packet received from the zigbee board and working studying the MAC layer for identifying problems which occur when the packet is transmitted from the board to the PC.

Manan:

Milestone 1: Manan studied the Active message layer on the mote, extracting the C code and finding the data structure transmitted by the mote. (Milestones 1 through 4).

Milestone 2: Manan studied the active message layer on the moteside and integrated the application for transferring the data to the PC from the mote; modified the java application of HW2 to parse the packet and display the contents on the screen. Currently investigating how the radio chip is initialized and the manner in which the packet is transmitted from the mote.

The updated webpage can be viewed at <http://www4.ncsu.edu/~mmsah/Project.htm>