

Programming Assignment 1

ComS/CprE 454/554 Spring 2007

Total: 100 points

Due: Monday, February 19, 11:59pm

Problem Description

Suppose there are n ($n = 2, 3, \dots, 6$) motes whose addresses are $0, 1, \dots, n-1$, respectively. Each mote maintains a local *virtual clock*. Specifically, once a mote is turned on, its clock value is initialized to an arbitrary integer between 0 and 7; then, every 1 sec, the value is decreased by 1; if the value is 0, decreasing it will result in 7. If these motes have the same initial value and are turned on at exactly the same time, their clocks will keep synchronized naturally. In practice, however, this may not be the case. In this assignment, you will develop a program to synchronize the clocks of the n motes, regardless of the order they are turned on. Here, we assume any two motes can communicate with each other.

A simple algorithm works as follows:

- After mote 0 starts, it sends its current clock reading to mote 1 every certain time interval (the repeating is needed since mote 1 may not be up yet when the first several such messages are sent out), until mote 1 receives the reading and sends back an acknowledgement. This way, the clocks of mote 0 and mote 1 are synchronized.
- After mote 1 has been synchronized with mote 0, it sends its current clock reading to mote 2 every certain interval until mote 2 receives the reading and sends back an acknowledgement.
-

You may implement the above algorithm, or design and implement a better algorithm (i.e., synchronizing motes more quickly, incurring less messages, etc.) to achieve the same goal. Note that, the above algorithm is not efficient if the motes are turned on in an arbitrary order. The groups of two graduate students are strongly encouraged to try better solutions.

To facilitate observation and debugging, you are also required to:

- Let every mote show its current clock reading using LEDs. The required mapping between clock readings and LEDs reading is as follows:

Clock reading	Red LED	Green LED	Yellow (Blue) LED
0	Off	Off	Off
1	Off	Off	On
2	Off	On	Off
3	Off	On	On
4	On	Off	Off
5	On	Off	On
6	On	On	Off
7	On	On	On

- Let every mote send its current clock reading via UART to PC, every 1 second. So that the readings of a mote can also be displayed by “java.net.tinyos.tools.Listen” whenever it is attached to a PC.

We suggest you to first test your program with TOSSIM. To simulate initial inconsistency in time clock, you may set different initial clock readings for different motes. Then, you can test your code with 2 motes that your group has. After that, you may borrow extra motes temporarily from the instructor or from other groups to test the cases with more than 2 motes.

Submission

- You should zip the directory of your source code and submit it via WebCT.
- You should ensure that, once your submission is expanded under <tos>/apps directory, it can be compiled correctly. Otherwise, you may get no point.
- Your submission should be named as assignment-1-XXXXX-YYYYY.zip or assignment-1-XXXXX-YYYYY.tar.gz, where XXXXX and YYYYY are the last names of the students.
- One group only needs to submit one copy.
- The submission should include a readme file briefly explaining your solution; adequate comments between your codes are strongly encouraged to improve code readability.