CSE535 Mobile Computing, Fall 2007
Homework #2 - Basic TinyOS programming

Due:

This is a group project (Each group will have 3-4 members). The purpose of this assignment is to become familiar with programming on mica2 motes. This assignment will involve reading through the tinyos tutorial lessons 1 through 6 and uploading applications on the mica motes.

Introduction to TinyOS and Mica motes:

The Mica2 mote has a processor, radio, and leds on it. It also has provision for interfacing with sensor boards. MTS300CA is an example of a sensor board that can be interfaced with Mica2 motes. This sensor board contains temperature, light, acoustic and sounder sensor devices.

The motes can be programmed by attaching them to a Mote Interface Boards (MIB510). The MIB510 is interfaced by connecting it to the serial port. The Mica2 motes have an onboard flash that can be programmed. The mote that is to be programmed is connected to the 51 pin male connector on the board. Motes run a multithreaded operating system called TinyOS. TinyOS is based on component model. Each component declares the commands it uses and the events it will signal. A simple FIFO scheduler will be part of each program uploaded onto the mote. The program will consist of code for components that will be used. The components communicate with each other by passing commands. Events are usually initiated by hardware devices. Based on the event, the component related to that event will issue one or more commands to other components. TinyOS system, libraries, and applications are written in the NesC language. NesC has a C-like syntax. The machine to which the MIB is connected contains the TinyOS code, (i.e.) the code for the components. We write our code by using the component code that is already present in TinyOS. To compile the program, we use an ncc compiler. The output by default is called main.exe. Then avr-objcopy converts the exe file produced by ncc into a text format that can be used for programming the mote's flash.

To program the motes, first run the cygwin batch file located on the desktop. Cygwin is a linux like environment for Windows machines. Attach the mote to a MIB510, then attach the MIB510 to the PC via the serial port. Change directory in the cygwin window to /opt/tinyos-1.x/. This directory contains the source files for tynos. The /opt/tinyos-1.x/apps contains sample applications that can be run on motes. The apps directory contains a set of directories, each of which contains an application. To compile an application, change to the particular directory, and type "make mica2". If it is successful it will output the amount of RAM and ROM space the program will require. The output of the make command will be main.srec. Now this can be used for programming the flash memory. The make file has an install option that takes care of uploading the program onto the flash. The install option uses a program called "uisp" to upload the program onto the flash. The TinyOS directory also contains tiny os tutorial. It also contains manuals for motes, sensor boards and for tiny os.
To complete this assignment, install tinyos 1.1.11 which is available from tinyos.net (http://www.tinyos.net/dist-1.1.0/tinyos/windows/tinyos-1.1.11-3is.exe) and includes the cygwin installation on your own machine. You'll be able to examine the tinyos code, modify example application and compile them. TinyOS comes with a simulator called TOSSIM. You use TOSSIM to test the example programs. Lesson 5 of tinyos tutorial gives an introduction to TOSSIM.

**Querying the motes:**

In this assignment, we will get data from sensors based on queries submitted by a java program on a PC. This assignment will involve writing a tinyos application that will retrieve sensor data whenever it receives a request. Requests will be transmitted by a base station node (a node with TOSBase program) connected to a PC. The requests will be in the following format SenseMsg.h. The SenseMsg AM message will be used by the basestation to send a request to the motes and it will be used by the motes to send the data back to the base station.

Requests will be initiated by a java program that takes queries from the user. It prompts the user to enter the type of data to request and the id of the mote. Types of data that can be requested are 1) temperature 2) light reading 3) microphone input. The sensed data is returned as voltage values. The temperature voltage value can be converted into degrees Celsius, but this is not required for this programming assignment. (Hint: to sense data you will have to use some of the components that are in /opt/tinyos-1.x/tos/sensorboards/micasb directory.)

**Sample prompt:**

Please enter the data type followed by the mote id or 'q' to exit:
Valid data types
- t  temperature
- l  light
- m  microphone

```
t 1 <-------------User Input
```

The temperature value at mote 1 is : 167 <-----------------------Output

Please enter the data type followed by the mote id or 'q' to exit:
Valid data types
- t  temperature
- l  light
- m  microphone

```
l 2 <--------------------------User Input
```
The light reading at mote 2 is : 238 <--- Output

Please enter the data type followed by the mote id or 'q' to exit:
Valid data types
- t ------------- temperature
- l ------------- light
- m ------------- microphone

m 2 <--- User Input

The microphone reading at mote 2 is : 210 <--- Output

Please enter the data type followed by the mote id or 'q' to exit:
Valid data types
- t ------------- temperature
- l ------------- light
- m ------------- microphone

quit <--- User Input

Good Bye

**Deliverables:**
- Java program code that takes queries and displays sensed values
- NesC source codes for the sensing application
- Report which contains what you learned and what were problems you have faced.
  Also, you have to specify the contribution of each member in a report.

**Submission:**

Please zip all your files named by cse535_hw1_your group number.zip. Email to the TA (su.kim@asu.edu) with the full names, ASU IDs of all members, and a subject starting [cse535]. Please cc the email to yourself and submit a hardcopy of the received email at the start of the class.

**Demo:**

There will be a demo with the TA at IMPACT lab (BY517). The demo will be scheduled later.