

# CSCE 515 Computer Network Programming

## Homework 1

Discuss in class on Oct. 6, 2008

On the course website, please find the file `tinyosw1.tar` which contains five directories of files. The directories are as follows:

- **Forwarder/** - A very simple TinyOS program to use as a starting point. All the program does is listen for a message and then broadcast the same message once it hears anything. You can think this as a program in application layer.
- **system/** - Has the full set of general system files for TinyOS. These are architecture independent, and hence are not a specific mote platform. In the process of compiling, if a file has the same name as the one in the specific mote platform, the file in the specific mote platform will be chosen. So always search for files in the specific platform directory first, e.g. **mica2** in this homework.
- **type/** - Has the header file `AM.h`, which defines the packet format. Note the format defined in `AM.h` contains both the fields that will be sent on the radio and will not be sent on the radio.
- **mica2/** - Has the files specifically for Mica2 motes. This is where most of the interesting radio stuff is.
- **Interfaces/** - Has the files where you can find the specification of interfaces.

These files correspond to the communication (transmission and reception) functions in TinyOS 1.x. This is not the complete TinyOS 1.x (many files were removed in order to allow you to focus on relevant pieces). You can find the complete TinyOS 1.x at: <http://tinuos.cvs.sourceforge.net/tinuos/tinuos-1.x/?sortdir=down>

You should explore these files and attempt to reconstruct the flow and interconnection of the code/functions. It is not necessary (or expected) that you try to compile this code. Specifically, you should explain and turn in the answer for the following questions in written (Hint: the answer is in the file `mica2/CC1000RadioIntM.nc`. Check the following function `async event result_t SpiByteFifo.dataReady(uint8_t data_in)`):

1. What type of framing is used? How is the start and end detected? What is the format of data frame?
2. How and where is CRC checking performed? What is the code polynomial used? What is the checksummed message for 1011 using the selected polynomial?
3. How and where is carrier sensing performed? Is there any backoff operation? If yes, what is the rule of backoff?

To help you get started, you might examine:

- <http://www.tinyos.net/tinyos-1.x/doc/stack.pdf> - radio stack of the MICA mote. This is NOT the same as in the MICA 2, but there are some similarities (enough to decipher what you need).
- <http://www.tinyos.net/tinyos-1.x/doc/tutorial/index.html> - TinyOS tutorial. Lesson 4 covers the radio, but doesn't discuss MICA2 specifically.
- <http://nesc.csourceforge.net/papers/nesc-ref.pdf> - NesC language reference manual. NesC is the language used to implement TinyOS.