Assignment #2 – TinyOS programming

Due 13/12, at 17:00
Please put your assignment answer in Philippe Bonnets mailbox

Overall description
You task is to design and program a TinyOS program to run on the 13192EVB that utilizes the temperature and light sensor on the sensor board, and provides feedback on the sensor readings, both on the serial output (USB) and the LEDs.

Prerequisites
You must install TinyOS on a PC of your choice, running either Windows or Linux with an USB port. The PC must have access to the Internet, more specifically it must be able to make HTTP requests against www.distlab.dk on port 80. If you do not have access to a PC, you can use one in the projektlab.

We will provide you with the following:
- A 13192EVB platform with sensor board mounted.
- A boot loader host program for Linux and Windows.
- A TinyOS 13192EVB tree that you must use instead or alongside your installed TinyOS tree
- For Windows: A set of drivers for the FTDI chip.
- For Linux: Source code to a simple terminal program that you can use instead of minicom

All software will be provided at http://www.distlab.dk/v04-336/ – you must use username: student, and password: sNs42 to access the files. Please do not distribute these files to people outside your group! The files includes a README file that provides more information about the compilation process. Please consult this file as soon as possible.

Assignment
Your assignment is to develop a TinyOS program that sample the light sensor at 5Hz, the temperature sensor at 1Hz, and output this information to the UART (sci module).

Additionally, the “red” led on the board should be on when the light is off and the temperature is greater than 22 degrees C, and off at all other times.

As part of developing this program, you will have to calibrate the light sensor such that you are able to report whether the light is on or off, and calibrate the temperature sensor such that you can translate the readings into degrees C.

Your report should include, detailed information about your program analysis and your calibration process. We expect you to hand in 1) report that includes at least a printed copy of your source code, component graph, and 2) the sensorboard programmed with your application.

Hints
- Start early! You will face a number of challenges that you should try to resolve as quickly as possible. You work with real hardware, and sometimes hardware will fail. Test your sensor as
best as you can (see below) and try to convince yourself that your sensor node works as expected. Also: The compilation process is tricky, involving compilation external to your system. We have tried to make the process as smooth and bugfree as possible, but without a doubt some of you will be able to expose shortcomings in the system. If you believe you have found a problem, do not hesitate to mail madsdyd@diku.dk about it. Include all the information you can think of, inclusive the source code to the application you are trying to compile, etc. The discovery of bugs/flaws in the compile system and the submission of informative bug reports will add positively to the evaluation of your report. Only the first group that submits a bug will be credited, of course. Bugs will be fixed/reported on the homepage during the assignment period, and the software repository updated as needed.

- Discuss with other students on the newsgroup diku.sensor-networks.
- Due to the limitation in UART bandwith, which is effectively around 240 bps, you should consider the encoding of data sent over the uart. We expect to be able to make sense of the information by attaching the sensor node to a standard terminal emulation program such as hyperterm (Windows), minicom or miniterm.
- Note that recent NesC distributions include editing modes for emacs and vi, as well as for the popular kde applications such as kate, kwrite and kdevelop.
- The sensor board you will be provided with, is programmed with a program that you can use to verify that your setup is working: Connect the board to an usb port, it should start flashing leds. Run a terminal program against the usb port, you should get output that indicates the sensor readings. Cover the node with your hand, the light level should change. Touch the temperature sensor, the temperature reading should change.