HS 2008 Prof. Dr. Roger Wattenhofer, Nicolas Burri, Philipp Sommer

Ad Hoc And Sensor Networks
TinyOS Exercise

Assigned: October 13, 2008
Due: December 19, 2008

1 Exercise Setting

This is the hands-on exercise where you will learn how to program a sensor network. In particular, you will develop applications for the TinyNode sensor node using the TinyOS platform. To spread the project workload it is possible to do this exercise in groups of up to three people. Since special hardware is required we provide a workplace. We installed an online reservation system on the course website to facilitate lab management and to guarantee you a free workplace upon reservation. All the necessary equipment and the key for the lab have to be picked up at our office (ETZ G64.1) and must be returned completely after work. The exercise is due by the end of the semester. Nevertheless, it may be wise not to defer the work for too long since the lab schedule might be quite tight towards the end of term. Once you finished the exercise, you will present us your achievements and send us all relevant files by email.

You will use the TinyOS plug-in for Eclipse to develop your applications. An introduction on how to build a TinyOS project inside eclipse can be found at `http://tos-ide.ethz.ch/`. To get familiar with the TinyOS 2.x platform we strongly advice you to first work through the first three lessons of the online tutorial (*'Getting Started with TinyOS'*, *'Modules and the TinyOS Execution Model'* and *'Mote-mote radio communication'*) provided by the TinyOS community. The tutorial is available at (`http://docs.tinyos.net/index.php/TinyOS_Tutorials`). The tutorial assumes that a normal shell is used for compiling and flashing of the sensor applications. You can ignore these instructions as we are working with Eclipse. Please read through the tutorial at home such that you do not waste time in the lab.

Once you have done the tutorial you are ready to develop your own program. Below, two applications are described. As a guideline, the first one is easier. However, the second one is a bit more challenging but also more fun (who would not want to have a Morse telegraphy enabled sensor node :-)). For both applications code skeletons are provided. You are expected to be able to successfully implement at least the first one.

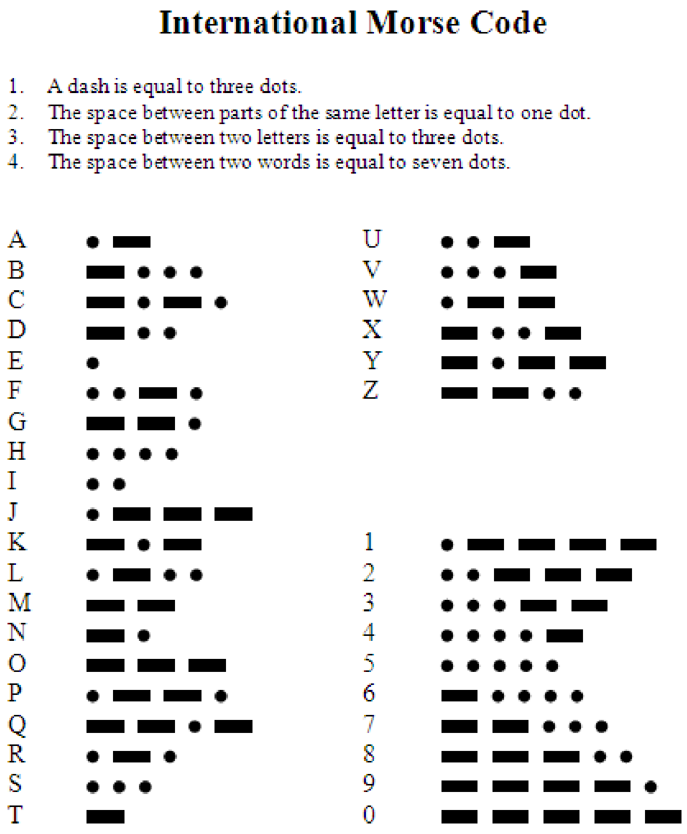
The lab computer is equipped with all needed software. We installed the Eclipse plug-in for TinyOS and created projects including skeletons for all the applications. We maintain a FAQ linked from the course website. If your questions are not already answered there feel free to ask one of us personally at ETZ G64.1 or via mail.

2 Application 1: Light Sensing

The *Light Sensing* application setup consists of two sensor nodes. Node A measures the ambient light intensity. Node B is used to visualize any intensity change by its LEDs. Therefore, A is sampling the light sensor periodically and broadcasts the corresponding value over the radio. On successful message transmission the green LED is toggled. In case of a transmission failure A turns on all LEDs. Node B listens for these messages. It then compares the current sensor reading with the last received value. If the light intensity change is within a given threshold the yellow LED is set. If the new intensity is getting lower the red LED is set. If the intensity is increasing the green LED is set. The code skeleton for node A is in the SensingSender Project. The files for node B can be found in the SensingReceiver Project. Note that the return value of a sensor reading is in millivolt. The output provides voltages from 0 mV (complete dark) up to around 1200 mV (direct sunlight).

3 Application 2: Morse Telegraph

For this application you are supposed to implement a Morse telegraph. As with the Light Sensing application the *Morse Telegraph* also uses two sensor nodes. The goal of this exercise is to transmit simple text messages from node A to node B using infrared light. Using a Java GUI the user enters a message. The message is first translated to the Morse alphabet and then the PC instructs node A to switch on or off the infrared LED according to the Morse codes. Node A's red LED is switched on or off simultaneously with the infrared LED so that we can see that the system is working correctly. Node B continuously samples the voltage at the infrared receiver, decodes the transmitted characters, and writes them on the serial port. Using `cutecom`, a standard terminal application connected to the corresponding serial port, you can see the result of your transmitted message (the TinyNode operates at a baud rate of 115200 and without flow control enabled). You will have to implement the receiver. That is, you will develop node B's code based on the given Skeleton (MorseReceiver project). The Java program on the PC and the code for node A (SimpleBlinker project) are provided by us. The Java GUI can be started by typing `java -jar Morser.jar` in the shell.



International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.

A	• ▬	U	• • ▬
B	▬ • • •	V	• • • ▬
C	▬ • ▬ •	W	• ▬ ▬
D	▬ • •	X	▬ • • ▬
E	•	Y	▬ • ▬ ▬
F	• • ▬ •	Z	▬ ▬ • •
G	▬ ▬ •		
H	• • • •		
I	• •		
J	• ▬ ▬ ▬		
K	▬ • ▬	1	• ▬ ▬ ▬ ▬
L	• ▬ • •	2	• • ▬ ▬ ▬
M	▬ ▬	3	• • • ▬ ▬
N	▬ •	4	• • • • ▬
O	▬ ▬ ▬	5	• • • • •
P	• ▬ ▬ •	6	▬ • • • •
Q	▬ ▬ • ▬	7	▬ ▬ • • •
R	• ▬ •	8	▬ ▬ ▬ • •
S	• • •	9	▬ ▬ ▬ ▬ •
T	▬	0	▬ ▬ ▬ ▬ ▬