TinyOS Lab – An Intrusion Detection System

Laboration in Wireless Sensor Network Programming
Department of Information Technology, Uppsala University

Student 1
Name: 
UpUnet-S ID: 
Course instance (e.g. Datakom DV1): 

Student 2
Name: 
UpUnet-S ID: 
Course instance (e.g. Datakom DV1): 

Agreement
We have independently worked on the following assignment solution.
We have both taken part in creating the solution, according to the assignment specification.

Sign 1: 

Sign 2: 

Date: 
1 Formalities

This assignment should be performed in groups of two. You are encouraged to use available resources such as online code examples. However, you are not allowed to use code from other groups doing the assignment. If you discuss with another group, please state so on the front page to give them credit and avoid suspicion of cheating. The assignment is divided into two tasks. The first task is mandatory to pass, while the second is optional for extra credits. The teaching assistants will prioritize to assist groups working on the first task.

This lab will be graded during the lab session. In order to pass, you need to demonstrate your application to the TAs. See the individual task descriptions for what you need to demonstrate.

2 Setup and test the development environment

2.1 Setting up the virtual machine

In order to get started, you need to setup a virtual machine that has the necessary tools for building a TinyOS program. Here’s how to proceed:

- The virtual machine image is located on a CD-ROM available during the lab session.
- Copy it from the CD-ROM to your local harddisk, e.g., to the Desktop and unzip it.
- Start Oracle VirtualBox.
- Click the New button to create a new virtual machine.
- Choose any name for the virtual machine. Set the OS to Linux, Ubuntu.
- Select a base memory size of 1024 MB.
- In the following dialog, Choose Use existing hard disk, and select the image you have copied in the first step.
- Finish the setup and boot the virtual machine.

After the virtual machine has booted, you should be presented with a Gnome desktop. In case you are asked for a user name and password, the name of the default user is ubuntu, with password reverse.
2.2 Attaching a sensor node to the virtual machine

After you have attached a sensor node to the host machine using a USB cable, you still need to inform the virtual machine about the node. To do this, click the USB icon on the bottom right of the VirtualBox window and select Zolertia Z1. If you want to attach multiple nodes, you will have to repeat this step for each node. Your node(s) should now be available in the virtual machine.

You can verify that the node is connected by running the command `dmesg` in a shell on the virtual machine. For each attached node, the last few lines should contain a message similar to:

```
[1492904.809075] usb 1-5.1: cp210x converter now attached to ttyUSB1
```

Notice that if you unplug the USB cable from the PC and plug it in again, you will have to repeat this procedure to make the nodes available in the virtual machine again.

2.3 Preparing the source code

The source code for the clicker program can be found under `/home/ubuntu/tinyos-lab/clicker`. Before you modify it, compile it and program the nodes to make sure that your toolchain works. First you need to configure your program to use the assigned channel and active message type. You set the proper channel that you are assigned in the Makefile in the program source directory. You need to modify the lines:

```
CFLAGS +=-DCC2420_DEF_CHANNEL=17
CFLAGS +=-DCC2420_DEF_RFPOWER=3
```

in the files `ClickerClient/Makefile` and `ClickerBaseStation/Makefile`.

To set the assigned active message type, you will also need to change the line:

```
enum { AM_CLICKER_MSG = 42 };
```

in both `ClickerClient/ClickerMsg.h` and `ClickerBaseStation/ClickerMsg.h`.

Set the values according to the following table:

<table>
<thead>
<tr>
<th>Active message type</th>
<th>Channel</th>
<th>Transmission power</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM_CLICKER_MSG</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

Demonstrate to a teaching assistant (TA) that you have programmed the nodes and that the clicker program works. You will also need to show the modified Makefiles.

Setup OK: ____________________________________________
3 Introduction

The goal of the assignment is to write your first program in TinyOS. You will start with an existing program: The clicker application that was introduced in the lecture. The clicker application counts clicks from different wireless sensor nodes and visualizes the number of clicks on the LEDs.

Your goal is to change the clicker application to a simple intrusion detection application. In this application, a node should send a packet to a base station whenever it is shaken. You will have two Zolertia Z1 nodes to test your code.

4 Task 1

You are to write an intrusion detection program. There should be two types of nodes: A base station that will visualize alarms using its LEDs, and client nodes that will send an alarm packet whenever they are being shaken.

The client nodes should use accelerometer data to detect movement. The accelerometer is a sensor which measures its acceleration, which suits the purpose to detect movement. Your task is to modify the clicker application into an intrusion detector. A client node should send alarms when it is shaken and the base station node should visualize that an alarm is received. You need to demonstrate to a TA that your application works and the TA will sign your report.

Here are some hints:

- To learn how to use the accelerometer, check the Zolertia example page: http://zolertia.sourceforge.net/wiki/index.php/Code_Examples_with_Z1
- It is enough to check the acceleration along one axis. A sampling rate of 10 Hz, i.e., reading the accelerometer 10 times a second is also sufficient for the simple intrusion detection system.
- The accelerometer driver provides the interface Read<uint16_t>. However, accelerometer readings are signed. Cast values that you get from the readDone(...) events to signed integers.
- A bug in the accelerometer driver in TinyOS may cause starting the driver to fail. If this happens, hold the node’s reset button for two seconds. The driver should initialize correctly again.

Task 1 OK: ________________________________
5 Task 2

This task is optional, but can give extra credits towards your grade. Now you should extend the intrusion detection system built in Task 1 to handle multiple client nodes. The base station should visualize the number of detections in the system using the LEDs. If the client nodes are not shaken for some time, the LEDs should turn off. Before you start to code, sketch a solution and discuss it with a TA.

The demonstration is to shake one or two nodes and show that the base station turn on and off LEDs accordingly. Show this to a TA.

Task 2 OK: ________________________________