

Lab 1 – Introduction to TinyOS and nesC

Learning objectives

The main objective of this lab is to make the student familiar with the development environment for sensor platforms using TinyOS and understand the basic concepts of its programming language nesC.

Specifically, after completing the lab the student shall be able to:

- Explain the component and interface concept and how to wire them in order to create an application.
- Understand the concurrency model of TinyOS being familiarised with the concepts atomic, commands and events.
- Program a simple application to run in a real hardware sensor node.

Assignment 1: Blink the led

You are requested to complete the application skeleton BlinkTask in order to make blink a led on the sensor node. The led has to be 1 second switched on and 9 seconds off.

First, you will find out the needed components and interfaces for the application and wire them together. Information about the interfaces and components can be found in the TinyOS directory (/courses/TDDI07/tinyos-main/tos).

You will use only one timer to control the delays. The timer should be fired only when it is time for the led to change. Finally, compile and test your solution in the sensor node.

Assignment 2: Sensing application

You are requested to complete the application skeleton PhotoMeter. The application has to measure the illuminance using a sensor and display the result on the leds. For that purpose, the application has to sample data every 50 milliseconds from the sensor (PhotoC component in the skeleton) and display the value in the leds according to the following rule: values in range 0-33% out of the maximum will turn the led0 on, 33-66% will turn the led1 on and values greater than 66% will turn on the led2. Only one led should be on in each interval. The maximum will be 1024 (RANGE in the skeleton). You will store 10 values in a circular buffer and display the average of the last 10 values.

| | | |
|--------------|---------------|----------------|
| Led0 – 0-33% | Led1 – 33-66% | Led2 – 66-100% |
|--------------|---------------|----------------|

Read interface:

The illuminance data is provided by the Read interface in 16 bits. Printf can help you to debug the sensor readings if needed (see additional resources). Study the read interface in the TinyOS directory and write code for handling its event. When the reading is successful, store the data and use a separate task to process it.

Circular buffer:

You are required to use a circular buffer implemented as an array in order to store the data readings. This circular buffer will be used in the data processing task and therefore you have to synchronise the access to the variables.

Data processing:

The calculations and manipulations of the leds should not be performed within any event handler. Performing heavy calculations in an event handler means further events occurring during the calculations will be blocked/delayed/lost. This decreases the responsiveness and accuracy of the system.

Therefore, you are required to use a separate task to process the data as it is provided in the skeleton (processData). If any data is available in the buffer it will perform the display function, where the calculations and manipulations of the leds have to be done. You will display the average of the 10 last values in the leds.

Optional: the sample values have to be converted to illuminance (Lux). For that you need to use the following formula:

$$Illuminance = (0.625 * 10^6) * \frac{\left(\frac{sample}{4096} * 1.5\right)}{100}$$

Data collection:

You are required to show the value that you display in the leds in the terminal using the printf() function. You can show the average or a single sample. The printf() function buffers the data and it does not write the data until its buffer is more than half full. You can use printf() after calling printf() to directly send the data.

In order to display the values in a graph, you will need to redirect the output of the PrintfClient program to a file and plot it using the LiveGraph application (see the LiveGraph folder in the lab1 folder). Your application should only output the value in order to plot it later. The output can be redirected with the following command:

```
java net.tinyos.tools.PrintfClient -comm serial@/dev/ttyUSB0:telosb > yourfile.csv
```

To run livegraph follow the instructions in the document readme.txt which is located in tinyos-lab/Lab1/LiveGraph/.