CS 3468
Lab 4

Jared Wallace

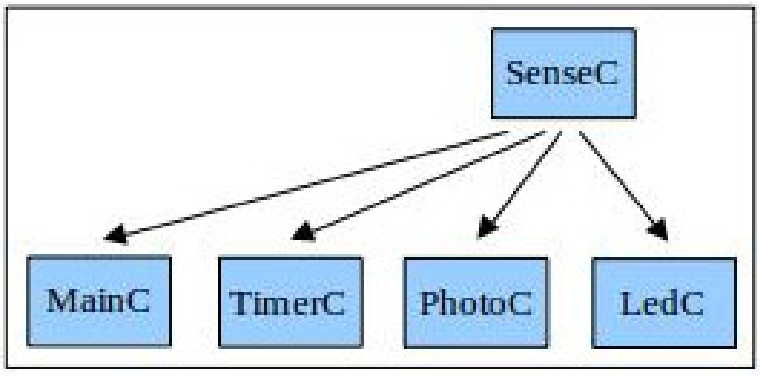
Objectives

1. Learn to use a sensor to sense the environment
2. Learn to read and analyze sensing data

Tasks

Sense Light

1. Besides communication, sensing is the other major function of a sensor. Depending on the actual hardware components, a sensor may sense light, temperature, presence of a magnetic field, humidity, sound, acceleration, etc. The diagram below shows a sensor application that senses the light level once per second.



The application has seven components in total, and the component PhotoC is what actually handles the sensing of light levels.

- System component MainC

- System component new TimerMilliC() as TimerC
- System component PhotoC
- System component LedC
- Application component SenseC

2. Now, open the API link `http://www.tinyos.net/tinyos-2.x/doc/nesdoc/micaz/`. Find the API documents of the first five system components and their interfaces. Implement a sensor application such that:

   - The sensor senses the light every 2 seconds.
   - The red LED lights up when the application is sensing.
   - The green LED indicates that the intensity of the light is beyond a certain threshold (0x180).
   - The yellow LED indicates that the intensity of the light is below a certain threshold (0x180).
   - The lab instructor will give you a threshold to use as a reference. You may need to adjust the threshold according to your environment.

3. Compile and run your application, and show your lab instructor

Send data back to the computer

1. Now, use the radio transmission component (like we did last week) to send the data back to a base station connected to the computer.

2. The application should have the following workflow:

   - The sensor samples the light intensity every 2 seconds.
   - The sensor computes an average of the light intensity measured every 6 seconds, and sends that average back to the base station.
   - The red LED should light up to indicate when it is sensing. It should be off when the data is being transmitted out.
   - The green LED should light to indicate that the intensity of the light is beyond a certain threshold (0x180).
   - The yellow LED ishould light to indicate that the intensity of the light is below a certain threshold (0x180).

3. Follow the lab instructors example, and run a Java program to read and display the data being sent back. Demonstrate your application for the lab instructor.

Model the sensor and the environment

1. The lab instructor will turn off the lights in the room, and give you a set of simple calibration tools.

2. Your sensor measures the intensity of the background light. Read and record three sets of received data in Table 1. If the deviation is more than 10 percent of the average, please redo the measurement.

Data 1	**Data 2**	**Data 3**	**Average**	**Deviation**

3. Turn on a light source and move it around your sensor. Measure the light intensity at various positions. Read and record three data measurements at each position, and record them in the following table. If the deviation is more than 10 percent of the average, please redo the measurements.

Angle	**Distance (ft)**	**Data 1**	**Data 2**	**Data 3**	**Average**	**Deviation**
0	0.5					
	1.0					
	1.5					
	2.0					
	3.0					
30	0.5					
	1.0					
	1.5					
	2.0					
	3.0					
60	0.5					
	1.0					
	1.5					
	2.0					
	3.0					
90	0.5					
	1.0					
	1.5					
	2.0					
	3.0					

Lab Report

1. Please demonstrate your program to the lab instructor and let him check your code at the end of the current lab project.
2. Your project report is due at the beginning of the next lab.
3. Grading criteria
   - Demonstration, 15 percent
   - Code, 15 percent
   - Report, 70 percent

Report instructions

Format:

1. Include your name and ID in the first page
2. Font size of at least 10pt
3. Single spaced
4. Maximum of 5 pages (I will take points off for exceeding this without any good reason)
5. Please submit as PDF online, and turn in a hard copy

Content:

1. Introduction (10 percent of the report grade) Please summarize the task of this lab and what you have learned in the lab
2. Implementation (30 percent of the report grade) Please describe in detail how your made your program. Show what components you used, how they are wired, and through what interfaces.
3. Experiment (30 percent of the report grade) Please describe in detail what can be observed from your program and explain how said observed behavior is a result of your code (and not happy coincidence).

StackSort connects to StackOverflow, searches for 'sort a list', and downloads and runs code snippets until the list is sorted.