Exercise 2

Instructions:

• The exercises are to be done individually (direct copying of code is not allowed), but you can of course ask your friends for advice. Good place for discussing the assignments and the course in general is channel #momib at IRCNet.
• Some individuals will be asked to present their work to the group at the exercise sessions.
• Support for Android development is provided by the course staff.
• You can also use IOS, Windows phone, Meego etc., but with less support.
• Testing on real mobile devices is recommended, but not mandatory.

• Be prepared to demo and explain your solution at the exercise session 26.3.2013 at 14:15 in D123.
• Return the source code of your solution in a .zip, .tar.gz or tar.bz archive to the Moodle page of the course at https://moodle.helsinki.fi/course/view.php?id=9438 before 26.3.2013 14:15. Note that Moodle uses the University of Helsinki AD login and password, not the ones of the department of computer science. Please enclose the files you intend to return into a single folder that is named according to the format firstnamefamilyname, and include that folder into the archive. The maximum size of the archive is 20 MB.
• Getting the point from the exercise requires that you both return your solution to moodle and are present at the exercise session. If you cannot be present at the exercise session for some good reason, you can compensate for your absence by demoing your solution personally to the course assistant.

Assignment

Communicate the names of the sensors available on your mobile device to a server using the CoAP protocol. When the server receives the list of sensor names, it should send an acknowledgement message back to the mobile device.

Design your communication so that it complies to the CoAP conventions. Additionally, in order to create an uniform naming convention for the sensors, use the sensor type constants from http://developer.android.com/reference/android/hardware/Sensor.html as the baseline for your sensor names. Create the sensor names by removing the “TYPE_” prefix from the constant name and converting the string to lower case. For example, you should use “temperature” as the name of the temperature sensor in the CoAP communication. The same naming convention should also be followed on non-Android platforms.

Hints for developers:

The CoAP specification can be found in: http://datatracker.ietf.org/doc/draft-ietf-core-coap/.

Jcoap is an open-source Java implementation of CoAP http://code.google.com/p/jcoap/. Jcoap works on both Android and Java SE, and it also includes a simple CoAP server. A good way to work is to run the jcoap CoAP server on your computer, and use the jcoap client library on your mobile device or emulator.

Wireshark supports CoAP and it is a good tool for debugging the communication http://www.wireshark.org/docs/dftref/c/coap.html.