Faculty of Mathematics and Computer Science

– Georg-August University Göttingen –

Telematics Group

Lab 7 - Sound Recognition

Sensorlab

Contents

1 Sound Recognition

As you may have already learned in some of the previous labs, there are sensor boards containing also sound sensors. This lab makes use of these sensors, with the aim of detecting clapping sounds.
You will use the MTS310CB sensorboard, which contains, among other things, a microphone and a buzzer. To get good support for streaming data from the microphone, it will be necessary to install some research code from the University of Szeged.

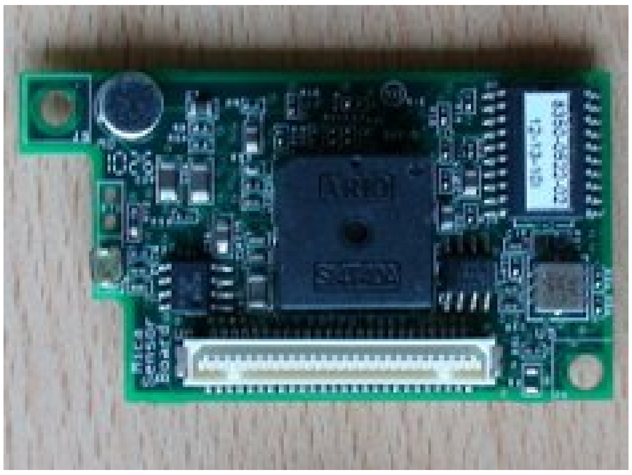
Figure 1: MTS310CB sensorboard

1.1 Setup

Unless the code is already installed, you can perform the installation using the following commands on the commandline:

```
mkdir -p $HOME/local # Create target directory
cd $HOME/local       # Enter directory
wget http://szte-wsn.cvs.sourceforge.net/viewvc/szte-wsn/tinyos/?view=tar \
   -O szte-wsn-tinyos.tar.gz  # Download from CVS
tar -xvzf szte-wsn-tinyos.tar.gz # Unpack
gedit $HOME/.bashrc  # Set up environment as follows:
```

To use the new code, a change to `.bashrc` has to be made. Look for line that looks like this:

```
export SZTETOSDIR=$HOME/local/tinyos/tos
```

This line sets an environment variable, which points the build system at the location of the files needed to build applications, which depend on the SZTE code. If such a line does not exist, add it. If it exists but the `SZTETOSDIR` variable points elsewhere, modify it accordingly.

1.2 Base station

Since for audio transmission, high bandwidth and bigger packet size than provided by the regular base station is required, you will have to setup a mote with the SZTE base station code. This can be found in `$HOME/local/tinyos/apps/BaseStation`. You can install it on a mote as usual. To receive the serial data sent by this base station, the `java` commands have to be modified to use a higher than default baud rate. For example, **instead of**:

```
java net.tinyos.tools.Listen -comm serial@/dev/ttyUSB1:iris
```

The default value for IRIS motes (`iris`) has to be replaced with the correct baud rate (230400):

```
java net.tinyos.tools.Listen -comm serial@/dev/ttyUSB1:230400
```

1.3 Microphone

You can find a microphone demonstration application in `$HOME/local/tinyos/apps/MicReadStreamTest`. Try installing it on a sensor mote. To receive the audio data on the PC, you should be able to use the `Capture` application in this directory:

```
java Capture -comm serial@/dev/ttyUSB1:230400
```

1.4 Assignment

There are two parts to this assignment.

1.4.1 Practical (35 Points)

Build a WSN that can detect sudden noises, such as hand clapping. Whenever a sudden noise is detected, the PC attached to the base station should output a notification either in the GUI or on the console (or both). In addition, led2 should be triggered whenever a sudden noise is detected.
Optionally, you can also look into using the buzzer on the MTS310CB board to give additional audio notification, but take care not to create a feedback loop.
To detect sudden noises, a possible approach might be to calculate a rolling average of the sensed audio samples, to smooth the input, and look for the difference between low and high points. If you decide to use this approach, describe why you think this is a good approach. Otherwise detail your reasoning too.

1.4.2 Theory (15 Points)

See if you can think of other approaches to analyze the signal. What pros and cons are there for processing the audio signal on the PC vs processing on a sensor mote (Note: IRIS sensor motes have 8KB or RAM and a sub-10MHz CPU with a word size of 8bit).
Could Fourier transforms be useful for analyzing sensed audio data? If so, why? Are there faster ways to calculate it?
Do you think calculating Fourier coefficients on a sensor mote is feasible, while the mote is capturing audio data in real time?
Consider and describe shortly if any other useful audio processing approaches come to mind, which could be used on motes or the PC/Basestation.