INSTRUCTIONS TO PERFORM THE SOUND TEST ON SODAQ SARA N211

1) Connect the Loudness Sensor to the Sara board:
   VCC pin —> 5V;
   GND pin —>GND;
   SIG pin —> A0 (whatever analog pin is ok);
   NC pin (Not connected) —> no connection (sometimes i keep that pin connected to A0 pin because of the Sound Sensor configuration, and it seems to continue to work)

2) Adjust the potentiometer on the sensor:
   - set up the Arduino IDE for the Sara board, following instructions at https://learn.sodaq.com/getting_started/ ;
   - import and load the code at http://wiki.seeedstudio.com/Grove-Loudness_Sensor/ in the Arduino IDE; (I also attached the .ino file so you can take that one)
   - look at the Serial Monitor while the Loudness Sensor is listening to the least noise possibile (quite room). Values on the monitor should oscillate between two values (e.g. 135-145). I think that it will never give us a stable value, because of istaneneous changes of pressure level on the microphone.
   - **Adjust the potentiometer on the sensor to get results on the monitor between 205-220 in "quiet mode", so with no noise**. (We can choose any interval, I propose that one because it seems to be a mean interval between the two I measured by adjusting the potentiometer to their extreme positions [135-145] and [265-280]).

3) Measurement of different noises:
   - IMPORTANT: Due to the different volume of the computer speakers, sounds in the files could be louder or lower so we have to "calibrate" the speaker volume to ensure the same test conditions to everyone. I suggest doing these steps:
     - play the sound file and put a Sound Level Meter App close to the computer speaker, to read the SPL value of that sound in dB. (* and **)
     - adjust the speaker volume to get the dB value that you can read below, when playing the sound at that precise frequency (e.g. play the sine wave sound at 200 Hz and adjust the speaker volume to read 58.8 dB on the phone app)
   - play the files that I attach with this guide and note down the results that you see on the Serial monitor. As said before, these results will oscillate between two values. Maybe a code that makes an average could be useful to get a unique result.

Information about the structure of the sound file “sinusoid_sequence” that I built:
This file is structured as follows:

0 to 10 s : SINUSOIDAL WAVE AT 200 Hz —— 58.8 dB
10 to 20 s: SILENCE
20 to 30 s : SINUSOIDAL WAVE AT 300 Hz —— 70 dB
30 to 40 s: SILENCE
40 to 50 s : SINUSOIDAL WAVE AT 440 Hz ——68.2 dB
50 to 60 s: SILENCE
60 to 70 s : SINUSOIDAL WAVE AT 500 Hz —— 66.5 dB
70 to 80 s: SILENCE
80 to 90 s : SINUSOIDAL WAVE AT 2000 Hz —— 72.3 dB
90 to 100 s: SILENCE
100 to 110 s : SINUSOIDAL WAVE AT 3000 Hz —— 72 dB
110 to 120 s: SILENCE
120 to 130 s : SINUSOIDAL WAVE AT 5000 Hz —— 26.8 dB
130 to — s: SILENCE

* : The dB value on the phone App should be constant when play a pure tone as those one above.

**: I also attach single files of different tones. I think that these one are more simple to use when we have to calibrate the speaker volume.

The other file named White Noise contains a 30min-long white noise, **I set up the speaker volume to get 54 dB on the phone app.**

The last file is “20680__acclivity__testtones”, that I found in the Internet and that could be a good benchmark because contains a lot of pure sound at different frequencies. **I set up the speaker volume to get 50-51 dB on the phone app when playing the 100 Hz sound at t = = 45s from the beginning of the file.**