

## “Assignment 1 - Step Counter” by Prerna Chikersal

### Disclaimer/ References:

I referred to the following for the basic structure of our app: <http://www.gadgetsaint.com/android/create-pedometer-step-counter-android/>. I implemented by own algorithm though.

I used the following library to build a real-time plot for the accelerometer data: <https://github.com/appsthatmatter/GraphView>. And I referred to the following example code to implement real-time plotting: [https://github.com/appsthatmatter/GraphView-Demos/blob/master/app/src/main/java/com/jjoe64/graphview\\_demos/examples/RealtimeScrolling.java](https://github.com/appsthatmatter/GraphView-Demos/blob/master/app/src/main/java/com/jjoe64/graphview_demos/examples/RealtimeScrolling.java).

### Description:

Here is how the accelerometer data was processed to derive step count:

1. For each new sensor value (x, y, z), magnitude m was calculated.
2. Magnitude values over *ACCEL\_BUFFER\_SIZE* (set to 50) number of samples were stored in a global buffer.
3. A local buffer which was a copy of the global buffer was created. All calculations moving forward were performed on this local buffer.
4. Mean of the buffer was subtracted from each element of the buffer. This process is called demeaning.
5. Median filtering is also performed on the buffer with a window size of *MEDIAN\_FILTER\_WIN* (set to 8) number of samples. Median filtering also reduces the size of the buffer such that  $\text{new\_length\_of\_buffer} = \text{old\_length\_of\_buffer} - \text{MEDIAN\_FILTER\_WIN} + 1$ .
6. Zero-crossings are counted next. However, only zero-crossings where the absolute maximum value (like amplitude) between previous and current zero-crossing is above a certain threshold are counted. This threshold is *ZERO\_CROSSING\_AMP\_THRES* (set to 12).
7. If final zero-crossings are greater than 2 ('cause we only want to count either +ve to -ve or -ve to +ve but not both) and *ACCEL\_STEP\_EVERY* (set to 40) number of samples have passed since the last set of values for which steps were counted, we increment step count by 1. Note that *ACCEL\_STEP\_EVERY* (=40) is the step size after which we calculate new steps. So, steps are calculated over buffer size of 50 with step size of 40. Hence only 10 samples are common between any two buffers from which new steps are calculated.

The following parameters are empirically tuned:

*ACCEL\_BUFFER\_SIZE* = 50

*ACCEL\_STEP\_EVERY* = 40

*MEDIAN\_FILTER\_WIN* = 8

*ZERO\_CROSSING\_AMP\_THRES* = 12

Results are especially sensitive to *ZERO\_CROSSING\_AMP\_THRES*, and this parameter depends on how a person walks and holds the phone.