



# Programming Assignment – 2

Topic: Plotting Sensor Values

Due: Mar 2, 2017 (Hard Deadline)

## 1. Overview – Plotting Sensor Values in Real-Time

The goal of this assignment is to plot sensor values, their mean, and their variance over time. The user will be able to choose which sensor he wants to view. Once the sensor type is chosen, the user will be taken to a second activity that presents sensor data in two formats: (a) raw values, means, and standard deviations over time, and (b) an animated image based on the current sensor reading.

## 2. First Activity – Sensor Chooser

- The first Activity should show at least two types of sensors:
  - Accelerometer (**must**)
  - At least one other sensor of your choice.
- The app shows the status of these sensors, i.e., present or not, and other info, e.g., max range, resolution, and min delay.
- When a user selects a sensor (e.g. by clicking the button), the user will be taken to the next Activity.

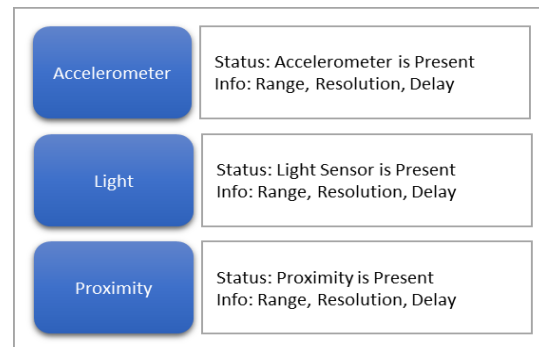


Fig 1. An example UI of the first Activity. You may use any other UI design to do the same.

## 3. Second Activity – Sensor Data Plotting (**must use Custom View**)

- A custom view (**must**) should plot the sensor values, a running mean of last 3-5 values, and a running standard deviation of last 3-5 values on the same plot.
- The X-axis will be time, and the Y-axis will contain the sensor values, means, and variances. That means, there will be 3 lines on the plot (of different colors).
- Note that: accelerometer values need to be calculated using  $\sqrt{A_x^2 + A_y^2 + A_z^2}$ .
- Because screen width is limited, once a line reaches the right edge of the plot, existing values need to be shifted one step to the left to fit in the new value.
- The plot will be updated in real-time, e.g., once every 100 ms. Use a time step for which the plot looks nice.
- Keep a back button so that we can go back to the first activity and choose another sensor.

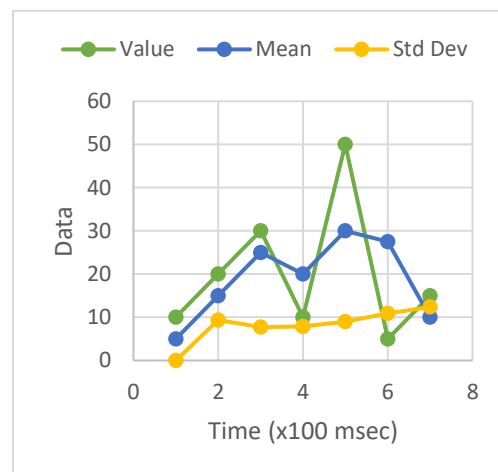


Fig 2. An example Custom View. Use appropriate X and Y-axis steps.

#### 4. Second Activity – Animation Based on Sensor Value (using Animation)

- Based on the current mean and/or variance of the sensor readings, you will show an appropriate animation.
- For example, you can define a threshold,  $T$  for the *mean light sensor value*. Based on this, you can define two types of lighting conditions: DARK, and BRIGHT. Now, you can use two interesting animations, one for each condition. Feel free to use more conditions.
- Similarly, using two thresholds for accelerometer variance, you can define three types of motion conditions (LOW, MEDIUM, and HIGH), and show appropriate animations for each condition.
- The animations must appear instantaneously, i.e. as soon as the value crosses a threshold.



**Fig 3.** An example animation scheme—changing the glow of the bulb based on sensor readings.

#### 5. Point Distribution

Tasks	COMP 590	COMP 790	Grading
1. A running app with a GUI showing at least 2 buttons, sensor status, and info.	10	10	
2. Touching a button brings a new Activity showing an XY plot.	15	15	
3. Plotting accelerometer values.	25	25	
4. Plotting accelerometer means and variances.	5 + 5	5 + 5	
5. Plotting the second sensor readings.	25 + 5 + 5	25 + 5 + 5	
6. Plot shifts to the left when the line reaches the right edge.	15	15	
7. Back button brings the app to the first Activity, and new sensors can be chosen.	5	5	
8. Animation for accelerometer value.	5	5	
9. Animation for another sensor. (Extra Credit)	5	5	
10. Dynamic adjustment of X- and Y-axis (Extra Credit)	5	5	
<b>Total:</b>	120 + 10	120 + 10	