

[← Week 9 Overview](#)

## Assignment 9: The Fitbit - Part 2

### Logistics

**Assigned:** Wed, Nov 2

**Due:** Wed, Nov 9

### 1 The idea

This is the second part of a two part project to build a Fitbit™ using your Arduino. In this part you will combine the data collected on the Arduino in [Assignment 8](#) with the graph from [Studio 9](#). When completed, you should have a device that collects a variety of data and relays it to a desktop for graphical display.

### 2 The assignment

Update your graphical display from Studio 9 so it's dynamic. I.e., it should always graph the most recent 45 seconds of accelerometer data. If the fitbit is in pedometer mode, you should also annotate the graph to show when a step (peak) is detected on the plot. This should only be shown in pedometer mode, these annotations should not appear during sleep mode.

In addition to graphing the accelerometer data, the StdDraw window should also be updated to include a few simple text indicators: \* The number of steps taken since the last reset \* The current rate of steps (in steps per hour) since the last reset \* The current temperature \* The total amount of sleep Each of these should be clearly labeled and should include units.

### 3 Hints

\*StdDraw can "double buffer". Typically StdDraw does each drawing command directly to the screen. Since each takes time to execute, the user can see each step of updating the screen. This usually leads to poor performance and/or a "jumpy" animation. Double buffering allows for smoother animations by applying all graphical updates to a hidden windows (the second buffer) and then switching to it all at once, which gives the appearance of all the updates happening simultaneously. In order to enable double buffering with StdDraw you will want to call the `StdDraw.enableDoubleBuffering()` command once. Then you will want to repeatedly call `StdDraw.show()` when you want to display the updates — nothing will be displayed at all until you `show()` your updates, so be sure

to call `show()` when all content has been drawn.

- Dynamic displays of data can be tricky. It may be best to create one or more methods that draw the display given arbitrary data and then just repeatedly call them with the most recent data.

#### 4 The check-in

1. Commit your code and verify in your web browser that it is all there.
2. Check out with a TA. You can create any files needed, but you should follow the same basic structure used in prior assignments. All your assignment code should be in `assignments/assignment10.fitbit`. If you make any modifications to the Arduino code used in Assignment 9, you should also include it in `assignments/assignment10.Fitbit`.

#### 5 The rubric

- Is the circuit wired properly?
  - Temperature sensor
  - Accelerometer
  - Two buttons
- Protocol Design
  - All pieces are present
  - Implemented properly on the Arduino side
- PC (Java-based) display
  - Does the graph show the last 45 seconds of data
  - Are axis labeled
  - Are units shown
  - Are tic marks and values shown so that the user can estimate the values for a particular point
    - Do the tic marks update dynamically if appropriate (i.e., does time update)
  - Is there a label, data, and units shown for the step count
  - Is there a label, data, and units shown for the step rate
  - Do step counts/rates reset appropriately
  - Is there a label, data, and units shown for temperature
  - Is there a label, data, and units shown for total sleep
  - Are peak annotations shown in pedometer mode (and not sleep mode)?
- Code style

Generated at 2016-11-02 16:15:43 -0500.

Page written by Bill Siever. Site design by [Ben Stolovitz](#).