

[← Week 8 Overview](#)

## Assignment 8: The Fitbit - Part 1

### Logistics

**Assigned:** Wed, Oct 26

**Due:** Wed, Nov 2

### 1 The idea

This is the first part of a two part project to build a [Fitbit™](#) using your Arduino. Fitbits have a lot of different functions, but we are primarily interested in accomplishing three things:

- Tracking step count (pedometer)
- Tracking the amount of sleep a person gets
- Tracking temperature (the Fitbit doesn't actually do this, but we can!)

You have already completed some of these pieces over the previous assignments. This assignment will focus on two primary goals: tracking sleep, and creating a protocol that your Fitbit can use to communicate data back to the PC.

In the next assignment, you will build a health monitor that processes and displays the data you record from the sensors on the Arduino.

### 2 The assignment

#### 2.1 Tracking Sleep

So far, we've been using the accelerometer to track steps, but we can also use it to track sleeping patterns. Consider that when you are sleeping peacefully, your body has very little motion. If you toss and turn a lot in your sleep then you won't feel as rested when you wake up.

Think about how this would apply to the accelerometer: **we want to know when the values *aren't* changing** - almost the opposite of how we do step detection. Not only that, but for this purpose we also **care about movement in any direction**. If movement is detected on any one of the three accelerometer axes, then we know that we aren't getting a

nice, restful sleep pattern.

Since we're using the accelerometer to track steps already, we need a way to tell it to switch over to sleep tracking mode. A real fitbit allows you to tap it to change its setting. We will use a button instead. Wire up a button on your breadboard and connect it to one of the Arduino pins. You should also use an LED or two to indicate which mode you are in. This can be one of the LEDs built into the Arduino, or an LED that you wire up.

Next, you will need to update your code to implement this new sleep tracking mode. We want to know the total time that a person has been asleep since we've entered sleep mode. Create an Arduino sketch (or better yet, update your step count sketch) to read the button input, and switch to sleep mode if necessary. You'll then need to read the accelerometer values (remember that we are interested in all three axes!). If you detect little to no motion, we want to update the amount of time that the person has been asleep. If you detect motion, we don't want to update the timer.

Consider how much motion is "too much." Some motion during sleep is to be expected: people will continue to breathe while they are asleep, for example. We want our program to be fairly sensitive, but if it is too sensitive it won't record sleep at all, even though someone is asleep. Play around with various thresholds for motion until you find one that works.

Once you are finished, test the entire circuit. If we are in pedometer mode, we want to track the number of steps like we did in studio. If we are in sleep mode we want to track the amount of time the user has been asleep since entering sleep mode - we don't need to worry about tracking steps in this mode.

Finally, add a second button to your circuit that can reset the step count when it is pressed. Note that when you go into sleep mode, the step count should not change. The step count will only reset when the reset button is pressed.

## 2.2 Designing a Protocol

Recall that in Assignment 4, we gave you a protocol that you had to implement. It is now your turn to design and implement a protocol. Our protocol needs to be able to do the following:

- Debugging Strings      0x30 in UTF8 format
- Error Strings          0x31 in UTF8 format
- Send temperature sensor readings      0x32 converted filtered temperature reading, 4-byte float, degrees C
- Send accelerometer data
  - Step counts              0x33 step counts, 2-byte integer, since reset
  - Time spent asleep        0x34 time spent asleep, 4-byte integer, milliseconds since asleep
- Send the total time that the application has been running      0x35 total time application running, 4-byte integer, milliseconds

You are welcome to refer back to assignment for to help you with this part, however we want you to design your own protocol. This means coming up with a magic number and keys for each type of value you want to send.

magic number: 0x23 - ASCII '#'

Pay special attention to the accelerometer and temperature values: this is perhaps where your biggest choice lies. You could process the accelerometer data on the Arduino side, or send raw accelerometer data and process it on the java side - this choice is up to you. The same choice also exists with the temperature data.

Regardless of how you choose to set up your protocol, by the end of the assignment, you should be able to view debug strings, error strings, step count, time spent asleep, and temperature in the Java console and/or ViewInputStream.

### 3 The check-in

1. Commit your code and verify in your web browser that it is all there.
2. Check out with a TA.

### 4 The rubric

- Is the circuit wired properly?
  - Temperature sensor
  - Accelerometer
  - Two buttons
- Protocol Design
  - All pieces are present
  - Implemented properly on the Arduino side
  - Recieved and displayed properly on the Java side
- Buttons working properly
  - Sleep mode
  - Step reset
  - LED to indicate which mode the accelerometer is in
- Code style

Generated at 2016-10-27 10:34:58 -0500.

Page written by Doug Shook. Site design by [Ben Stolovitz](#).