Smart Sampling Ambulatory Stress Sensor System

Overview
Stress is linked to a broad range of health outcomes and is a very important risk factor for cardiovascular disease. The current approach to ambulatory stress research involves randomly sampling behaviors and physiological events during an individual’s daily life. While this collects good data, this technique has no way of assuring that a stressful event is recorded. The current system would objectively measure physiological and physical activity and use a “smart” algorithm to analyze this data.

Objectives
- Create the smart sampling system by either creating the sensor or purchasing a commercial grade device
- Create a system to collect and store the data on the smart phone
- Develop a predictive algorithm that activates application based on data that is indicative of a stressful event to allow for automatic startup of the current stress survey application.

Approach
- We spoke with our sponsor to determine what expectations he had for the system design.
- The MyBasis watch was chosen as the sensor, but an NDA is still pending.
- System level schematics were created to provide future programmers with a general outline of how the program should be coded.
- The schematics describe the “Baseline”, “Calibration”, and “Active Week” programs.
- We used the “Active Week” schematic to create a Matlab program simulating the adaptive algorithm.
- The MATLAB program also helped us develop a base for the final Android Application.
- The application was created to read data, analyze the data, and trigger another application.
- Log files of real heart rate data were used to test the applications ability to read the file, analyze whether the data was indicative of a stress event, and open the separate application after analysis.

Outcomes
- Subjects will possess an easy to use system that requires minimal user interface.
- System will only record results of stressful events.
- Each system will adapt to subjects typical heart rate and acceleration variability when stressed.
- Future programmers will easily be able to understand the functionality of the program and translate our schematics and code into a final application.