

Final Project (Part 1)

CS 3410 Systems Programming

Due Date: Monday Mar 27, 2017 at 6:00 PM

You may work in teams of *up to* **three (3)** members from your lab section

Background Information

You will build your own fitness monitoring device that will be capable of running some basic analytics on the information. There will be two parts to this system: data collection and data processing. You will use the following sensors:

- Heart Rate Sensor
- Real time clock
- One display device:
 - Matrix Display
 - 2x 7 Segment LED
 - LCD Screen
- One environment sensor:
 - Temperature Sensor
 - Humidity Sensor
 - Barometric Pressure Sensor

Your Arduino will send sensor reading to a host program (written in C) running on a UNIX-like system. The two programs will communicate over the Arduino's serial connection. The host program will store a histogram-like structure of readings which can be easily queried and outlier readings can be found. It is not difficult to imagine an extension that would allow your system to call *911* if it detected something anomalous. We will build this system in 4 parts over the remainder of the semester.

Specification (Part 1)

We will begin using two sensors: the heart rate sensor and display device of your choosing. You will write an Arduino sketch that reads in the current heart rate and outputs it to the display device. Whenever the display device is showing real-time data, you should light an LED (and turn it off otherwise). Additionally, the Arduino should flash an LED at the same rate as the detected heartbeat.

You will also write a C program to run on a UNIX host that communicates with the Arduino. The program will expose a command prompt that supports the following commands:

- **show X:** Set the output device to show **X** as the current heart rate instead of the current real-time value. This will be useful while debugging.
- **pause:** Pause the output and keep the display device showing the current reading
- **resume:** Show the real-time heart rate on the display device. This should be the default mode of the system
- **exit:** Exits the host program

Deliverables and Grading

- Push your code to GitHub (a link will be posted on Blackboard) before the deadline. The following things should be in a directory named **part1**:
 - One Arduino sketch
 - The C source code (and appropriate header files) of your host program. Be sure to include a **Makefile** so the grader can build your code. Do not forget, your code *MUST* compile with the following gcc flags: `-std=c99 -Wall -pedantic -Werror`. You will lose credit if your code compiles, but only without those extra flags.
 - One Fritzing¹ file that shows the circuit you built.
 - A document explaining how your are adapting the heart rate sensor readings to your display device. Also include any other information you deem relevant.
- You will be giving a demo of your circuits before class on the day of the deadline. Be sure to bring everything you need to show it off.
- Be sure to write clear and concise commit messages outlining what has been done.
- Write clean and simple code, using comments to explain what is not intuitive. If the grader cannot understand your code, you will lose credit on the assignment.
- Draw clean Fritzing schematics. If the grade cannot understand them, you will lose credit as well.
- Be sure your code compiles! If your sketches do not compile, you will receive **no credit**. It is better to submit a working sketch that only does a subset of the requirements than a broken one that attempts to do them all.

Table 1: Grading Rubric

Category	Percentage
Demo	70%
Compilation with <code>-Wall -pedantic -Werror</code>	10%
Code Quality (including Makefile)	10%
Schematic	5%
Write-Up	5%

¹<http://fritzing.org/home/>