

# ONLINE Final Project (Part 1)

## CS 3410 Systems Programming

**Due Date:** Tuesday Mar 24, 2020 at 11:50 PM

### Project: Simulated Signal Analyzer

#### Background Information

You will build a simulated sensing device that will be capable of running some basic analytics on the (simulated) information. There will be three main parts to this system:

1. data collection, pre-processing, and communication;
2. data processing and storage;
3. data analysis.

You will use the following hardware setup:

- One Arduino
- One Computer with (real or virtual) Ubuntu OS.
- Serial connection between the two.

Your Arduino will send sensor readings 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. In addition, you will build a database of relevant information for possible data analysis.

We will build this system in parts over the remainder of the semester.

#### Specification (Part 1)

You will be given an array of simulated sensor readings that you will send in a loop from the Arduino to the host computer through Serial. These should be sent at a rate of one sensor reading every second. Whenever the Arduino is done sending the reading, you should light an LED (and turn it off otherwise).

You will also write a C program to run on a UNIX host that communicates with the Arduino. This program should print the fake sensor reading that was sent from the Arduino (once per second), and a menu of options for the user to input. The command prompt should support the following commands:

- **blink <X>**: Cause the onboard LED for the arduino to blink **X** times during a second.
- **pause**: Pause the Arduino sensing program and keep the LED on.
- **resume**: Cause the Arduino to send a fake sensor reading to the host, print it on the terminal, and blink the LED once a second after the reading is sent. Blinking once a second should be the default mode of the system
- **exit**: Exits the host program

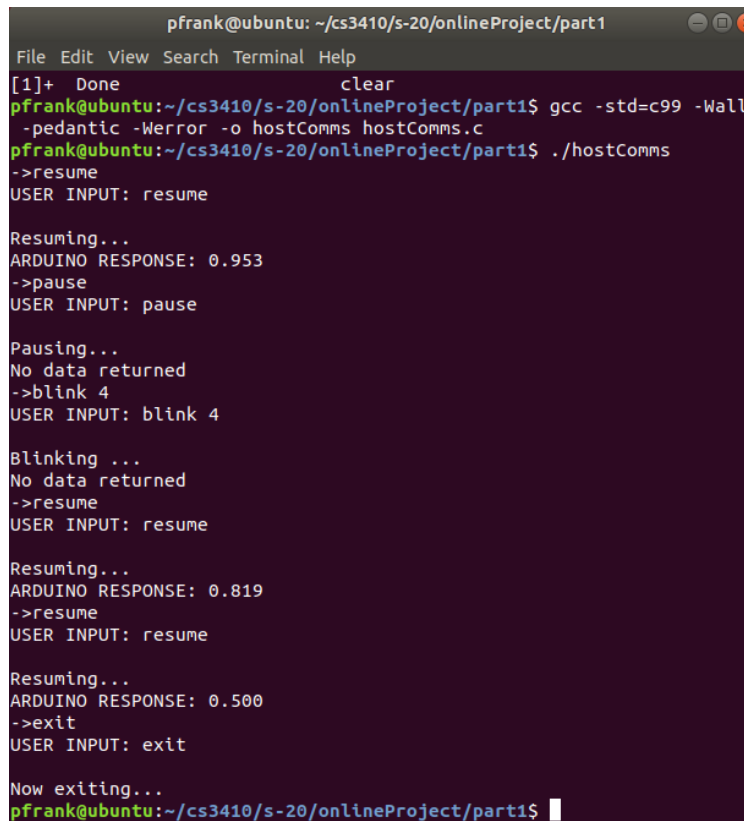
## Deliverables and Grading

- 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` (this might be updated later on). You will lose credit if your code compiles, but only without those extra flags.
- You must create a one page Write-Up explaining how your protocol works in a way that is easily followed by a casual reader. You may use state-diagrams or flowcharts to illustrate the steps followed.
- 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.
- 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%
Write-Up	10%

An example sequence of command output is shown in figure 1:



```
pfrank@ubuntu: ~/cs3410/s-20/onlineProject/part1
File Edit View Search Terminal Help
[1]+  Done                  clear
pfrank@ubuntu:~/cs3410/s-20/onlineProject/part1$ gcc -std=c99 -Wall
-pedantic -Werror -o hostComms hostComms.c
pfrank@ubuntu:~/cs3410/s-20/onlineProject/part1$ ./hostComms
->resume
USER INPUT: resume

Resuming...
ARDUINO RESPONSE: 0.953
->pause
USER INPUT: pause

Pausing...
No data returned
->blink 4
USER INPUT: blink 4

Blinking ...
No data returned
->resume
USER INPUT: resume

Resuming...
ARDUINO RESPONSE: 0.819
->resume
USER INPUT: resume

Resuming...
ARDUINO RESPONSE: 0.500
->exit
USER INPUT: exit

Now exiting...
pfrank@ubuntu:~/cs3410/s-20/onlineProject/part1$
```

Figure 1: Figure 1: Example output in the host terminal