

CIT 595 Spring 2016 Group Project

INTRODUCTION

In this project, you will work in a group to develop a system with “embedded” and “mobile” components. The goal of the project is to create a Pebble smartwatch application that is able to get data from and send control information to a remote sensor and display driven by an Arduino microcontroller.

REQUIREMENTS

System architecture

The system shall consist of three major components, described as follows:

- *Sensor and Display:* You will use a [Gravitech 7-Segment Shield](#) attached to an Arduino microcontroller. This component includes a temperature sensor, a seven-segment display (like a digital clock), and an RGB light. You may start with the Arduino program that is available in Canvas. This program reads the temperature and writes to the serial port once a second.
- *Middleware:* The Arduino shall be connected via USB to a Linux or Mac machine, which will run a C program that handles all communication between the sensor and the user interface.
- *User interface:* The user interface will be a Pebble smartwatch application, which communicates with the middleware over the Internet via an Android or iOS phone.

Note that because of firewall restrictions on the SEAS network, if you are running your middleware component on a Linux machine in the lab, you must use port 3001 for your server, and it can only be accessed by devices on one of the Penn networks. Please be sure to discuss this with a member of the teaching staff if you are unsure.

Functional requirements

Your system must fulfill *all* of the following requirements:

- The user should be able to see the most recent temperature sensor reading.
- The user should be able to see the average, low, and high temperature sensor reading for the past hour (if the sensor has been running for less than an hour, the user should be able to see the statistics for the time since the sensor started running).
- The user should be able to decide whether to see temperature readings (and statistics) either in Fahrenheit or Celsius. Additionally, the user should be able to change the 7-segment display on the sensor to show the temperature either in Fahrenheit or Celsius.
- The user should be able to put the sensor into a stand-by mode in which it is not reporting temperature readings (either to the user interface or to the display); however, during this time, the readings should be tracked for determining the high, low, and average. When the sensor is in stand-by mode, the user should also be able to tell it to resume reporting the readings.

- If the user interface becomes disconnected from the middleware (e.g. because of a network error or if the middleware stops running), an appropriate message should be shown to the user the next time it tries to connect.
- If the middleware cannot get a reading from the sensor (e.g. because the sensor is disconnected), an appropriate message should be shown to the user the next time the user interface tries to get a reading.

Note that you have complete freedom in how you implement these features in the user interface, i.e. what the user needs to do in order to see the data or send a command, and how it gets displayed.

In addition to the features described above, there should be a thread running on the middleware that waits/blocks until a user sitting at the terminal where the middleware was started enters the letter 'q'. Once it sees that, the middleware program should then terminate. As in Homework #4, it is okay if you require that one more request come in from the user interface before ending the program.

Additional features

In addition to the functional requirements listed above, your group must come up with *three* other (non-trivial) features to include in your system. At least one feature must involve sending a control message from the user interface to the display; at least one other feature must involve the processing and/or display of the data that is sent from the sensor to the user interface. Please discuss your additional features with the instructor or a TA before proceeding.

You may assume that all components are dedicated to each other, e.g. that your middleware component will never need to serve a different user interface, and that your user interface will never need to connect to a different server, so it is okay to hardcode addresses, etc. as needed.

MILESTONES

It is important that you make steady, continuous progress towards your final system. There are various milestones that your group should meet, as described below.

Pebble Tutorial (February 18)

During today's recitation session, your group should complete the Pebble tutorial and ensure that you can get a round-trip message sent from the watch to your server and then back.

This will not be graded, but you need to at least get the infrastructure in place so that you can get data sent between the components.

Milestone #1: Communication between Pebble and Arduino (March 17)

During today's recitation session, your group will have a brief (~5 minutes) meeting with

members of the instruction staff to demonstrate that it is possible for the Pebble watchapp to retrieve and display the most recent temperature reading from the Arduino.

During this meeting, you will be asked about the three additional features that you plan on implementing. These features need to be approved by a member of the instruction staff.

This milestone is worth 15% of the project grade.

Milestone #2: Prototype (April 7)

During today's recitation session, your group will have a 5-10 minute meeting with the members of the instruction staff. By this point, you should have much of the basic functionality completed, including showing the current temperature on the watch and sending some control signal from the watch to the Arduino.

During this meeting, be prepared to discuss issues such as:

- where does data processing (e.g., calculating the average temperature) occur?
- what is the structure of your communication protocols between the different components?
- what sorts of error-handling have you included in your code?

This milestone is worth 25% of the project grade.

Milestone #3: Final Presentation (before April 26)

Your group will present your system to members of the instruction staff and discuss your implementation decisions. At this point, the entire system should be working and you should have all features implemented. If not, you may present the project at a later date, but with a late penalty to be decided by the instructor.

This milestone is worth 40% of the project grade.

Final Deliverables (Apr 29)

Your group needs to submit a document that explains how your system works. Please include brief explanations of each of the following:

- what is the structure of the messages that are sent from the user interface to the middleware? give examples
- what is the structure of the messages that are sent from the middleware to the user interface? give examples
- what is the structure of the messages that are sent from the middleware to the sensor/display? give examples
- what is the structure of the messages that are sent from the sensor/display to the middleware? give examples
- how did you keep track of the average temperature? describe your algorithm and

- indicate which part of your code (including line numbers) implements this feature
- what are the three additional features that you implemented? indicate which parts of your code (including line numbers) implement these features

You should also write a short document explaining how someone would use your smartwatch application to access the features. Do not assume that it is “intuitive”!

This documentation is worth 20% of the project grade.

ACADEMIC HONESTY

You should be working with the other members of your group project team, obviously. However, you may not discuss or share code with students in any other groups, or with students who have taken this course in the past. If you want to use a third-party library, you **must** get permission from the instructor. Failure to do either of these will be considered academic dishonesty, and your team will receive a grade of 0 on this project. Which would be really, really bad.

If you run into problems, please ask a member of the teaching staff for help before trying to find solutions online!

SUBMISSION

All deliverables are due in Canvas by **Friday, April 29, 5:00pm**. Late submissions will be subject to a penalty determined by the instructor.

Your final deliverables should consist of the following:

Source code

Submit all Arduino, C, and Pebble source code in three *separate* zip files. If you use third-party libraries (which have been approved by the instructor!), only submit those in compiled form. Please be sure that all of your code is well-commented so it is easy for the grader to find where different features are implemented. Remember: happy graders give higher grades!

Documentation

Please create a *single* PDF that includes both the technical documentation and the user manual as described above.

Return all hardware

You must return the Arduino board, temperature sensor shield, USB cable, Pebble smartwatch (sorry!), and any other equipment before you can receive a grade for this assignment.

Updated: Thu Feb 25 2016, 8:46am