SYSC 3010 Systems Development Lab

Term Project

The term project is an exercise in systems design, implementation, testing, communications and in teamwork. The project objective (what you actually will build) is being left to your imagination, subject to meeting the technical requirements lists below.

Technical Requirements

- The system developed shall make use of a minimum of two RPis and a "normal" host machine (i.e. a lab machine or your laptop.
    - One RPi must be connected in standalone mode, optionally with an internet connection.
    - The second RPi must be headless, optionally with internet provided through the network-equipped Linux lab machine..
- Either individually or collectively, the system or its programs shall include the following characteristics:
    - Manage more than one device.
    - [Ideally] Manage a feedback loop
    - Make use of a timing loop (periodic)
    - Provide local feedback (output) AND control (input) using PiFace
    - Provide local feedback (output) AND control (input) using Gertboard
    - Include an interrupt service routine, ideally more than one. [Next year, 2014]
- The host machine will run a Java application with the following characteristics:
    - Provides a graphical user interface, using the SWING framework
    - Makes use of the XML framework, either for persistence or for configuration
    - Uses the Model-View-Controller framework
    - Demonstrates one or more software design patterns.

Process Requirements

Each team shall follow an incremental iterative approach, marked by four milestones. A milestone shall be marked by something that can be tested and demonstrated. Within each milestone, elements of requirements, design, implementation and testing shall be present. Broad suggestions for the milestones follow, with specifics to be provided by the team during the first project meeting.

1. Complete data path
2. Primary feature(s)
3. Complete system
4. Re-factored optimized system

Team Requirements

Work alone, go fast. Work together, work far.

Each weekly lab shall include a formal meeting in which the development activities of the past week shall be summarized, major issues identified, and plans for the coming week constructed. Each team member will submit a timesheet for their work in the past week.

When reporting completed activities, *specific* accomplishments, observations, conclusions or decisions must be noted.

Poor Example: Researched the maximum distance attainable for wireless communication
Good Example: Conducted an experiment for the maximum transmission distance. Found it to be 10 meters.

Teams shall manage their collaboration through the GIT version control system. The team repository shall maintain all source code. All project documentation as well as the team minutes and timesheets must be maintained on CULearn in the Group's Book. Part of the final grade of the project will include an inspection of the GIT history

For each milestone, a different member of the team shall act as leader: setting the meeting agenda, writing and submitting the meeting minutes. Each team member shall submit their time sheets individually to the instructor.

Schedule

Draft Proposal: Monday, October 7, 7:00 AM
If necessary, Revised Proposal: Monday, October 14, 11:45 PM
Design Review Presentation: Monday October 21, Lecture

- All relevant documentation must be in CULearn book by this time.

Checkin: Wednesday, November 13, Lab – Must be able to demonstrate a tested component.

- Requirement: Pull a PREVIOUS test from GIT and run it.

Code Review Presentation: Monday, November 18, Lecture

- All relevant code must be drawn directly from GIT.

Final Demonstration: Wednesday, December 4, lab

- ALL documentation must be in the CULearn book, and all code must be on GIT. Anything submitted afterwards will not be marked.

Documentation Requirements

1. Project Proposal: Please see the guidelines for *a Project Proposal* at https://sites.google.com/a/sce.carleton.ca/sysc-engineering-project/ although ignore the suggestion of 10 pages; instead aim for 1-2 pages. Also, take advantage of the *Writing Tips*.
2. Design Document, broadly organized as follows:
   Chapter 1: Introduction (Basically, the project proposal)
   Chapter 2: Requirements (Technical specification)
   Chapter 3: Design (System and Class Diagrams)
   Chapter 4: Implementation (Explanations only as needed. Code itself should be on GIT.)
   Chapter 5: Testing (Test plan)
3. Weekly Meeting Minutes (aka Product Backlog[1] & Sprint Backlog[2]) (Provided on course webpage)
4. Self and Peer Assessment (Provided on course webpage): At each milestone, each team member will complete both a self-assessment and a peer-assessment for each teammate. These will be marked and will contribute toward the project's grade.

Submission

Each team will submit all documentation and source code as a version on the GIT repository. Documentation must include a README.txt will tell a reader what files are provided and how to execute the system.

[1] http://www.mountaingoatsoftware.com/scrum/product-backlog

[2] http://www.mountaingoatsoftware.com/scrum/sprint-backlog