01.30.2019

Memo

To
JP Students

From
Prof. Healy

CC

Re
Project Plan Rubric

This is an updated rubric to help you re-submit your **project plan** at the end of the winter term. Use it to reflect changes and to fix errors from your fall term plan submission. Make sure to include **schematic drawings** and other artifacts that you had not completed at the time you submitted your plan in the fall.

You can find the specific errors to be corrected in the original graded submission that I returned to you earlier this month.

Timely Submission:

7 pts - Submit by 5:00 p.m. Friday, week 10 of the winter term.

Format/Organization:

8 pts – Word processed and double spaced, including a cover page, table of contents, list of illustrations, and list of tables. Use page numbers, and headings for each section.

Spelling Errors:

5 pts - Each spelling error incurs 1 point off up to total of 5 points.

Grammatical Errors:

5 pts - Each grammatical error incurs 1 point up to a total of 5 points.

Break Down into Modules / Subassemblies:

10 pts – Breakdown should reflect a manageable modular approach. For instance power supply, then Bluetooth communications, microcontroller, etc. Your breakdown may have changed since the drafting of your fall plan. Show those changes here.

Hierarchical Design Diagrams, Schematics, and Notes on Housing Design

20 pts – One or more block diagrams depicting your modular breakdown. (Fundamental tasks at the bottom. Final integration and testing toward the top.) Also include your wiring schematics and notes and diagrams related to your housing design in this section. Make sure you include schematics that the team has developed since the fall term. Use conventional schematic symbols.

Review the verbal descriptions for each of the individual blocks for clarity. Ask yourselves, “are these descriptions meaningful?”

Recognize that your algorithms probably have an inherent hierarchical structure as well. Use this same technique to explain this part of the development process as well: big picture problem at the top, broken down into a series of sub-problems. Then simpler and simpler problems as you descend the structure with the most mundane tasking at the bottom (the “base of the pyramid.”)

Each block diagram, each schematic, and every other figure ARE NOT stand-alone entities. Provide precise explanations for each, and make sure they accompany, or “caption” each of the figures. Use the feedback I provided earlier and refine your descriptions as necessary. **Detailed diagram descriptions are worth an additional 10 points.**

Test Data:

15 pts – Present data in accordance with the test plan you developed last term. This may require test plan modification that in all likelihood will force you to expand that plan beyond what you wrote up in the fall.

Embedded system software is generally light and lean. This is all the more reason you need to make sure it’s working properly:

- Verify interrupts are working correctly.
- Examine memory and register utilization.
- Ensure the target works robustly in isolation even though you’ve used cross-compiler debugging tools.

For the purpose of testing, you may have to modify your software to provide observability and control not normally associated with embedded systems; you can create conditional group blocks in your code by using directives such as #ifdef and #ifndef.

Test for all conceivable use cases. List these use cases in this section. Consider your upcoming design review and try to fill in the blanks for this question: “Did you consider ________, and did you test for ________?”

You may want to write additional software that serves to automate your testing – a.k.a. a test bench program. Use watchdog timers to verify proper execution. Be aware that manually tracing variables through your code has been shown to be very effective not only in identifying bugs but also in determining required tests and testing parameters. On the other hand, automated verification helps manage large quantities of data generated in more complex designs.

In the field you'll used test generation programs, automated design flow tools, and sophisticated tracking tools to ensure all test steps have been performed. Develop your own tracking process to insure your team completes the important testing steps that you've made the effort to identify.

On the hardware side, you could build a Thevenin model or Thevenin Equivalent circuit where the output is modeled as a voltage source in series with a resistor. Having this as a standard can help when troubleshooting, when testing for repeatability, or in the event of a component failure. Remember that CMOS logic zero is at Ground, or zero volts.

Be mindful of ground loops and the effect this can have on signal quality.

And finally measure that which can be measured, and display the numerical data in this section. Focus on that which relates to the quality and reliability of your prototype:

- Execution speed of programs
- Operational voltage and amperage boundaries
- Demonstrated battery life, if applicable
- Environmental extremes: temperature, humidity, dirt/particulate

Update your Parts List:

5 pts – Many of you have made scope changes and in some cases have switched out major components in your designs. Use this section to reflect those changes.

Software Requirements:

10 pts – Update your written software specification describing the function of each software piece. Include artifacts of the modeling tools that you used (UML or flowcharts, for example). Provide pseudo code that shows your process of software development. Provide some sample code, describing its function and where it fits in to the overall design.

While it would be impractical to provide comprehensive details in this section, I do expect to see **several pages** of artifacts along with the written spec described in the opening statement of the previous paragraph. Remember, I want to see *your development process*.

Individual Team Member Assignments:

5 pts – Tell me how the individual assignments and responsibilities of each team member has fallen out through the current term. This may or may not have changed since December.