EE/CS 120B: Introduction to Embedded Systems
University of California, Riverside
Winter 2015

Custom Lab Project

Introduction

You will work independently to design and implement your own custom lab project, your own creative embedded systems invention, on the ATmega1284 microprocessor. This invention can be a game or some useful device prototype. You may use components from previous labs and new components if you wish. Buying a new component is not required. As much as possible, your implementation should involve concurrent synchSMs implemented with no variations from the structured techniques in PES. You may use RIBS to help generate C code, although you will need to modify the generated code to be compatible with the ATmega1284 API.

If implementing a game, the game should include a way of accumulating and displaying a score, a way of winning and/or losing, informational messages to the user, and a button that can be pressed at any time to start over (a soft reset to the game). You can find previous examples by searching YouTube for “UCR EE/CS 120B.”

Custom lab “Build-Upons”

One of the most important components of the custom lab is the notion of a “Build-Upon.” These “Build-Upons” can come from two general domains: hardware and software. You may build-upon previous lab exercises, or create new software and/or use new hardware. A full “Build-Upon” score requires at least 3 individual “Build-Upons”.

Hardware: Use a non-trivial hardware component (new component or reuse an old component in a new way). For example, the lab kit’s LCD has the ability to write to each pixel on the screen, not just a single character at a time. Another example is re-purposing your speaker to be a microphone.

Software: Points can also be awarded for non-trivial state machines. For example, implementing a fast fourier transform (FFT) in software. Some game logic requires one or more non-trivial concurrent state machines.

A software and hardware example is having the microcontroller communicate with another device via USART, such as a desktop/laptop or another microcontroller.

Custom Lab Project Schedule

Week 7:	5/15/15	Project proposal due (11:59pm)
Week 8:		TAs provide feedback on project demonstration (in lab)
Week 9:		First prototype demonstration to TA (in lab)
Week 10:		Second prototype demonstration to TA (in lab)
	6/5/15	Final project demonstration to TA (WCH 336; time TBA)
	6/6/15	Custom lab project report and demonstration video submitted on iLearn (11:59pm)

Project Proposal

Each student is required to submit a short project proposal via iLearn. Project proposals are due on Friday, May 15th, at 11:59pm. The length of the project proposal may range from 2-3 paragraphs to 2 pages at an absolute maximum. The proposal should provide a conceptual overview of the project (e.g., describe the game or application), list the components that will be used, and provide a list of three or more "Build-upons" that the student expects to implement.

The TAs will review the project proposals and provide written or verbal feedback to each student in the following laboratory section (Tuesday, February 24th). The feedback will assess the overall quality and feasibility of the proposal. On the one hand, the objective is to ensure that the proposed project can be completed realistically in a 3-week timeframe; on the other hand, the proposed project must entail sufficient design complexity to warrant three weeks of implementation work—it should certainly have more detail and depth than a typical one-week lab assignment we have had throughout the quarter.

Students who submit project proposals after the deadline may receive feedback at the discretion of the TA, but will not receive credit for the submitted proposal (which counts toward to the total custom lab grade).

Demonstrations and Prototyping

Students are expected to work continuously on their project during the 3-week project period and weekly demonstrations of progress are required.

The first project demonstration (Week 8) should establish a working subset of the overall system functionality, and should include at least one "Build-upon."

The second project demonstration (Week 9) should show that the overall system functionality is nearly complete and should include at least two "Build-upons."

The final project demonstration (Week 10) should show complete system functionality and at least three "Build-upons."

Although the final project demonstration is scheduled for Friday, 6/5/15, students may demo early with permission of the TA. With 120 students in the class, there is only time for one project demonstration per student; afterward, the demonstration is final.

Demos after (6/5/15) are not allowed, except for mitigating circumstances (with documentation) and/or by discretion of the instructor or TA.

Custom Lab Report, Source Code, and Demo Video

Students are expected to write and submit a custom lab report and working code, similar to previous laboratory exercises.

The laboratory report should be submitted via iLearn and is due on 6/6/15 at 11:59pm.

The laboratory report should include a description of the custom laboratory project as proposed (accounting for revisions due to TA feedback), followed by a description of how much of the proposed project was actually implemented. The report should include a description of the testing strategy, how the project was tested, and a list of notable bugs that were discovered and fixed due to systematic testing. The report should also include a list of any bugs or other known problems with the project at the time of the final project demonstration.

Along with the laboratory report, students should also submit working source code as described in the "Laboratory Policies" document, which is available on iLearn.

The source code is also due on 6/6/15 at 11:59pm.

Lastly, each student is required to prepare a 70-110 video (no shorter, no longer). After preparing the video, upload it to YouTube with the EXACT title format:

- "UCR EE/CS 120B Spring 2014 -- *Firstname Lastname* -- *Custom lab title summarizing functionality*"

Your video should be publicly viewable. The video should demo your invention, highlighting all of your "Build-Upons", adding captions/annotations (YouTube provides the ability to add those items) to describe/highlight important features. While the video should have sound, ***the video should be fully understandable without sound***.

The video is also due on 6/6/15 at 11:59pm.

When searching for jobs, considering linking to this video from your resume/web-page/facebook-page/etc. A Google recruiter actually reached out to a former student for an interview because of one of these videos.

Submission Format

The submission should be a single file (.zip, .tar.gz, etc.) containing a folder containing the following files:

- A 1-page custom lab summary named REPORT_lastname.pdf (or .doc, etc.)
  - High level description of custom lab (derived from your proposal).
  - User guide (Rules, controls, and any special considerations).
  - Technologies and components used in custom lab (AVR Studio 6, ATmega1284, etc.).
  - Link to demo video.
  - Link to each source file and a few-word summary of file's purpose.
- Any image/drawings explaining how components were connected to the microcontroller (e.g. which pins you used).
- All source files (.sm, .c, .h).
  - Be sure to cite sources, include IEEE and Internet found code.

For self-promotional and marketing purposes, you may want to create a Google doc folder as well. This way, you can send a link to the project to any recruiters or hiring managers that you encounter in your professional life. This is neither required nor expected for CS/EE 120B.

- Make sure the folder is sharable with the link (no login required); test with someone else in the class to make sure it is shared properly.
- The idea is to create something you can link to from your personal homepage, for future job searches.

Custom Lab Project Grading Rubric

40% (80 points):	Custom lab “Build-Upons” (**must** demo final product to get these points)
20% (40 points):	Correct functionality when played (including unusual user interactions)
10% (20 points):	Implementation using PES' structured programming techniques
10% (20 points):	Demo video
5% (10 points):	First project demonstration (Week 8)
5% (10 points):	Second project demonstration (Week 9)
5% (10 points):	Submitted custom lab report, source code, etc.
5% (10 points):	Attendance

Anticipating the Unexpected

Sometimes projects do not work out as planned. If, for any reason, you wish to change your project substantially (e.g., change to a different game, alter or replace one or more “Build-upons,” etc.), you must first obtain approval from your TA. Submit your list of proposed changes in writing along with a short justification for each major change. The TA has the discretion to permit or forbid the proposed changes.