UCR EE/CS 120B

Custom Project

Introduction

You will work independently to design and implement your own custom lab, your own creative embedded systems invention, on the ATmega1284 microprocessor. This invention can be a game or some useful device prototype. You can build upon 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 from the lecture and course book. You may use RIBS to help generate C code.

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).

(Some examples from summer session can be found here and here, also just search YouTube for “UCR EE/CS 120B”).

Custom lab “Build-Upons”

A full “Build-Upon” score requires at least 3 individual “Build-Upons”. 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.

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 (½ complexity). 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.

Project Proposal

Prepare and submit a proposal for your custom lab project, including three proposed build-upons. If you desire, you may include a few alternative strategies and/or alternative build-upons, deferring a final decision until later. The following template is a good example of a custom lab project proposal.

The proposal is due at the end of Week 5 and should be uploaded to Gradescope. The instructor and/or graders will provide feedback as quickly as possible. The feedback will either approve your project (including build-upons) or make suggestions for changes to ensure that the complexity and challenges inherent in the proposal are appropriate for a 3-week end-of-quarter project.

Project Demo Videos

Each student is required to produce two project prototype demo videos (at the end of Weeks 9 and 10) that demonstrate intermediate progress on the project. Each student is also required to produce a final project demo video (at the end of Week 10).

The first two videos should be uploaded directly to Gradescope (uploading a URL is fine). For the final video, project source code should be uploaded to Gradescope, and the video can be linked from the header file, similar to the lab assignments.

The first prototype video (Week 8) should demonstrate your progress, including partial project functionality and at least 1 “Build-Upon.” The prototype should run and be usable, even if key components or subsystems have not yet been integrated.

The second prototype video (Week 9) should demonstrate further progress, including intermediate project functionality and at least 2 “Build-Upons.” The prototype should run and be usable, and have substantially more components or subsystems integrated than the first demo video.

The final demonstration video (Week 10) aims to demonstrate full system functionality, including all 3 “Build-Upons.”

Custom Lab Project Grade Breakdown

5%	Custom Lab Project Proposal
5%	First Project Demo Video (Week 8)
5%	Second Project Demo Video (Week 9)
10%	Final Project Demo Video (Week 10)
20%	Correct functionality when played (including when a user interacts unusually)
30%	Custom lab “Build-Upons” (**must** be observable in the final demo video)
15%	Implementation using structured techniques (from lecture and course book)
10%	Project Report

Custom Lab Project Folder

- Create a Google doc folder
    - 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 basic idea is to create something you can link to from your personal homepage, for future job searches.
- Add your custom project report named REPORT_lastname to the folder
    - 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.).
    - Links to all three demo videos.
    - Link to each source file and a few-word summary of file's purpose.
    - Short description for a resume which shows the technologies you learned.
- Add to the folder image/drawings explaining how components were connected to the microcontroller (e.g. which pins you used).
- Add to the folder all source files (.sm, .c, .h).
    - Be sure to cite sources, including IEEE and Internet found code.

Custom Lab Project Submission

- Submit your source code to Gradescope, just like a regular lab assignment
- In the commented header, include URL links to your final project demo video and your Google Drive folder

Final Project Demonstration Video

Prepare a 70-110 second video (no shorter, no longer). If you are proud of your project, you should upload the video to YouTube with the EXACT title format:

- "UCR EE/CS 120B Fall 2021 -- *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”. (When searching for jobs, considering linking to this video from your resume/web-page/facebook-page/etc. Google recruiters have reached out to former students for interviews because of their videos).

If you prefer to keep your video private, that’s also fine. **The only requirement is that your source code header file links to a video that the grader can view.**