UCR EE/CS 120B

Custom Project

Introduction

You will work independently to design and implement a custom lab project on the ATmega1284 microprocessor. Due to the limited time during the summer session, you have the choice of two projects you can complete: Side scrolling game or Pong. Your implementation should involve concurrent synchSMs implemented with no variations from the structured techniques from the lecture and course book, unless specifically waived by the TA/Instructor.
*Note: If you want to work on a different project you **must** get permission from the TA and Instructor. Keep in mind that summer sessions have limited time.*

Your 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). *Note: A soft reset **does not** require the user removing power from the microcontroller.*

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

Pong The classic pong game consists of two paddles on opposite sides of the screen. The user is controlling one of the paddles while the computer controls the other paddle. There is a ball that bounces in between the paddles. The ball speeds up and changes angle (as well as direction) when it hits the corner of either paddle. If the ball hits the center of the paddle it continues at the same angle, but reverses direction.	**Side scroller** A side scrolling game consists of a character on one side of the screen that moves according to the user controls. When the user moves to the right, the screen "scrolls" along introducing new enemies and/or items. Alternatively the screen can scroll at a predetermined pace and the user is along for the ride. If the user attempts to go to the left they are blocked by the edge of the screen. If the user hits an item, wall, or enemy, the game should respond accordingly.

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 repurposing 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.

You are limited in your choices, but still must have the 3 build-upons for your project. Below are example build-upons for each project. The check marked build-upons are “required” for your project in that the teaching staff sees no way to complete the project without these build-upons.

Pong ✓ Game logic (with ball physics) ✓ LED matrix and shift registers ● Second player ● etc.	**Side scroller** ✓ Game logic ● Custom characters on LCD (½) ● EEPROM to save score (½) ● Second player (interacts with first) ● etc.

Demo video

Prepare a 70-110 second video (no shorter, no longer). Upload to YouTube with the EXACT title format:

- "UCR EE/CS 120B *Quarter Year* -- *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/linkedin-page/etc. A Google recruiter actually reached out to a former student for an interview because of one of these videos).

In-lab proposal

Submit to your TA which project you decide to work on.

Milestones

There will be two milestone check-off's, the first at the end of the week you start your project and the second about a week after you start (check the calendar for exact dates). For each milestone you will need to

1. Demo a working build-upon to your TA
2. Create a wiring diagram for how your breadboard is wired currently
3. Take a picture (cell-phone okay) of your breadboard
4. Commit to github with a tag indicating which milestone it is.

Final demo

There will be the final demo day on the last day of lab where you get to show off your final project to your TA's and fellow classmates.

Custom lab grade breakdown

- 5%: Proposal
- 20%: Milestones (10% per)
- 5%: Attendance
- 20%: Custom lab "Build-Upons" (Final build-upon and integration)
- 25%: Correct functionality when played (including when a user interacts unusually)
- 10%: Implementation using structured techniques (from lecture and course book)
- 10%: Report
- 5%: Demo video

Custom lab folder and submission

- Create a Google doc folder
    - Make sure the folder is sharable with the link (no login required); test with someone else in the class (or in incognito mode) to make sure it is shared properly.
    - Idea is to create something you can link to from your personal homepage, for future job searches.
- Add to the folder your custom project report named REPORT_lastname. Report will contain:
    - High level description of custom lab.
    - 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 **public** github with your code. For each source file provide
        - 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).
- Submit *URL* of custom lab **folder** on iLearn as a link.