

Final Project Overview

In this lab assignment, you will do the following:

- Develop a significant final project individually or as part of a team.
- Demonstrate your project to the class and document your work in a professional final report.

Students should have their preliminary final project ideas formed by late February.

The PDR presentation submission for this assignment is due by Tuesday, March 20th.

The final demo presentation submission for this assignment is due by Tuesday, May 1st.

The lab demonstration of this assignment is due by Wednesday, May 2nd.

The supporting documentation for this project is due by 3:00 p.m. on Saturday, May 5th.

The final report for this project is due by 5:00 p.m. on Monday, May 7th.

This assignment is weighted as 28-35% of your course grade. This assignment is meant to give students the opportunity to explore an area which is of interest to them.

Lab Details

Hardware, Firmware, and Software

The **Due Date** for all final projects is **May 2nd**. By that day, each person or team must be prepared to demonstrate their final project hardware, firmware, and software, if applicable. Project demonstrations will be limited to approximately 7-10 minutes per project, so the team members should carefully plan how they will demonstrate that they met the project goals presented to the class earlier in the semester. Each team member should be prepared to answer questions related to his/her contribution to the project. Demonstrations will take place inside the lab.

Basic Final Project Guidelines:

- Students may work individually or in groups of up to three. Students may want to work with other students who have complementary skills; however, since the goal of the class is to learn as much as possible, students should consider accepting responsibility for a component of the project which forces them to learn new skills. Teams containing students from both ECEN 4613 and ECEN 5613 are allowed; however, all team members will be graded relative to the ECEN 5613 level standards.
- Final projects must be completed before the due date at the end of the semester. Choose a project which has several milestones. **Do not use an 'all or nothing' approach.** Make sure you have something working to demonstrate on the due date. Strongly consider showing incremental progress to the TA or instructor before the due date to guard against project failure on demo day.
- Projects must have sufficient complexity. Projects which are too easy will not be eligible for 'A' grades. There will be a difficulty factor and a quality factor associated with the grade for each project. Focus should be on electronics, firmware, and software, **not** on mechanical elements or packaging.
- Projects must include both a **new** hardware and a **new** firmware design component; however, students may choose to focus more heavily on either the hardware or firmware aspects of the project. The 8051 has limited bandwidth and projects should be defined with this in mind. Projects may also include a software component which allows communication between a host computer application and the project hardware and firmware, but the project focus should be on the embedded HW/FW.
- Students/teams will need to make a brief presentation (~5 minutes) at the preliminary design review (PDR) around week 10 of the semester. This presentation will give a brief project overview, define project milestones, and describe project deliverables. Project teams will need to generate a final report and give a project demonstration at the end of the semester.
- Projects must not pose a safety risk to students. In addition, projects must not make significant noise or otherwise disturb other students during development and during the presentation.

As examples of past 8051 projects, student teams have developed a SCSI interface to a storage device, an MP3 player (difficult) with an interface to a compact flash memory card, an EPROM programmer, a graphic calculator, a tank battle game, an embedded multitasking operating system, a robotic checkers game, a mobile robot, a home security system, a universal billboard programmer, a feedback control system, a web-based remote temperature sensor, an electronic battleship game, an X-10 home control system, a bar code scanner, a MIDI file player, and an infrared remote control sound effect generator.

Good final projects do not have to be expensive. Many excellent final projects cost less than \$10. Grades are not based on the cost of the project.

Suggestions for final projects include:

- Implement an embedded system using the low voltage Atmel AT89LP6440 (an 8051 derivative available in the 40-pin DIP package), or an Atmel AVR XMEGA processor, or an ARM-based processor. Obtain debug hardware and demonstrate how to use the built-in debugging capabilities of this chip. Demonstrate other capabilities, such as in-system programming and built-in ADC/DAC.
- Device Programmer: Design a programmer which can program a limited number of devices, such as a PLD, an EPROM or a programmable microcontroller, such as an 8751 or a Microchip PIC microcontroller. A beginning goal could be to program 1KB of an EPROM and then run an 8051 program from that programmed EPROM. More advanced goals could be to program multiple types of EPROMs, parallel EEPROMs, or programmable microcontrollers.
- Alternate Microcontroller: Design a project using an additional microcontroller, such as a Microchip PIC. This alternate microcontroller could communicate with the 8051 board built during the semester.
- Home Network: Design a system which will allow multiple devices to communicate using a protocol such as X-10 or RS-485. Each device on the system could store some particular pieces of data or control some element like a motor which could be accessed across the network.
- Storage Device Controller: Design an interface which can communicate with an external device, such as a floppy drive or optical storage device. Bandwidth limitations of the 8051 should be kept in mind if a project like this is attempted. **Projects should focus on functionality rather than performance.**
- Web Page Server: Create a web page server that responds to TCP/IP requests coming through a serial port attached to a modem which is in turn connected to a network. Alternately, design a system which monitors some device like a thermometer, and sends an e-mail message across the network.
- Operating System: Design a cooperative multi-tasking operating system which allows multiple tasks to share processor bandwidth. For students with a background in operating systems, this system could be extended to support real-time functionality, or to support forms of interprocess communication.
- MIDI Controller: Design a Musical Instrument Digital Interface which communicates with MIDI-enabled music equipment. This device could sequence different electronic instruments, merge MIDI channel data and filter unwanted MIDI events, etc.
- Digital Control System: Design a control system using DACs and amplifiers to drive motors, and a feedback mechanism such as an optical encoder or position sensor with an ADC. Balance a bar on its end (difficult). Maintain the position of an object on which a varying load is placed.
- Infrared Receiver and Transmitter: Design a system which can receive and decode IR transmissions from a remote control. The system could then translate the remote control's commands into an alternate format and retransmit the commands to a device (e.g. a radio, CD player, VCR, etc.)
- Plotter: Use stepper motors to control the movement of pens and paper and write a device driver to allow simple pictures to be drawn.
- USB Device: Design a low speed HID-class Universal Serial Bus peripheral. Perhaps use a USB interface chip or use an 8051 derivative which has an integrated USB interface. If this project is pursued, definitely consider a chip which integrates and automates as much USB functionality as possible. One suggested option is the Cypress EZ-USB 8051 derivative. The instructor has a development board and some components that might be of use. (Note: Difficult Project)

As a suggestion, students should determine an area which they find interesting and then try to structure a project around that area. The goal of the final project is more focused on learning something new and building something significant rather than designing and building something completely practical. One can get ideas for a project by looking through electronic parts catalogs, embedded systems magazines, application notes/data sheets, and walking through electronics surplus stores.

Projects could include the use of graphic LCD displays, input devices such as keypads or computer keyboards, ADCs, DACs, Flash memory, FPGAs, PLDs/PALs, and/or motors. One option for projects requiring higher bandwidth is to use a faster clock speed for the Atmel AT89C51RC2, either by using a higher frequency crystal or by utilizing the X2 mode of operation.

Stationary projects are preferable. Mobile projects such as robots or vehicles add complexity and typically require more mechanical design, which shouldn't be the focus of the project. A complex mechanical design reduces the time available to focus on the electronics and firmware development.

If you're requesting samples of particular parts, order DIP packages and consider whether the chip requires a 5V or 3.3V power supply. Many chips these days require 3.3V supplies and many are available only in surface mount packages; both of these characteristics make it more difficult to interface those parts to your existing 8051 circuitry. **Do not get packages such as ball grid arrays, since these will be impossible or very difficult for you to solder** - these packages are designed for solder reflow ovens.

Before ordering a specific part, take a look at the corresponding data sheet and verify that not only does the part have the functionality that you desire, but that you can understand the data sheet and any application notes for the part. Not all manufacturers provide high quality documentation. Poor documentation can make it difficult or impossible for you to be successful. Also take a look to see if there are any errata notes available which describe flaws in the silicon and any known work-arounds.

Order parts and samples early. Sample delivery can sometimes take several weeks or even months. If possible, you should have ordered and received your sample parts by the time you do your PDR presentation.

Final Project Preliminary Design Review (PDR) Requirements

Students/teams will need to make a brief presentation (~5 minutes) at the preliminary design review (PDR) around week 10 of the semester. This presentation will give a brief project overview, define project milestones, and describe project deliverables. You will be graded on your preparation for the PDR and for your presentation.

- Due to the large number of presentations, each team will be limited to approximately 4 minutes for the presentation, plus approximately 2 minutes for questions. The presentation should be carefully organized in order to maximize its quality given the time constraints. Each team member must present some portion of the information.
- Each presentation should be 3-5 slides, which will be submitted to the instructor electronically.
- The slides for this presentation do not have to be fancy. **Having good information content is much more important than having pretty slides.** Spend your time organizing your information; you're not being graded on the attractiveness of your slides.
- Each slide must have a footer which includes:
 1. the project name and names of the group members
 2. the slide/page number and the date
- The slides must indicate:
 1. New hardware elements of the project (a block diagram of your system would be interesting)
 2. New software elements of the project (indicate software structure or new modules to be written)
 3. Key milestones, including dates and current status (do you have all parts needed, etc.)
 4. Project deliverables, including a fall-back plan in the event of implementation problems
- The presentation file name must follow the following naming guideline. The name must begin with "S12_", followed by the student's last name (family name), followed by a brief identification of the final project. An example file name for a team of students named "John Adams" and "Thomas Jefferson" with a project related to the ZigBee protocol would be "S12_Adams_Jefferson_ZigBee.ppt".
- You must submit this assignment electronically **via CULearn by 8pm on the night before the PDR.** Only one submission is needed per team. Acceptable document formats include:
 1. Microsoft PowerPoint (.ppt or .pptx formats)
 2. Adobe PDF (.pdf) [full size high resolution, one slide per page, ensuring pictures are very clear]Other formats may be acceptable (e.g. if they can be easily converted to .ppt format), but you need to verify the alternate format with the instructor prior to submission. Be careful not to embed other files or expect applications (like Apple movies) to be available on the notebook computer used for display.
- Speak loudly and clearly during the presentation, to ensure that everyone can hear you.
- Do not speak exclusively to the instructor. Speak to the entire audience.
- Introduce your team members.
- Do not block the view of your audience. The team should stand to the side, so that everyone in the audience can see the projection screen.

After PDR, there are only a few more weeks left in the semester. You should either have already ordered parts for your final project, or you should be ready to order parts immediately after PDR is complete, to make sure parts arrive in time. Some parts may take several weeks to be delivered, so plan ahead. Before ordering parts, make sure good data sheets and application notes exist for the particular parts you plan to use. Lack of good documentation can cause project delays and/or failure.

Final Project Presentation/Demo Requirements

The **Due Date** for all final project demos is **May 2nd**.

- Each team member must participate in the final project presentation.
- The presentation file name must follow the following naming guideline. The name must begin with "S12_", followed by the student's last name (family name), followed by a brief identification of the final project. An example file name for a team of students named "John Adams" and "Thomas Jefferson" with a project related to the ZigBee protocol would be "S12_Adams_Jefferson_ZigBee.ppt".
- Each team must submit one copy of the presentation to the instructor electronically **via CULearn by 8pm on the night before the demo**. Acceptable document formats include:
 1. Microsoft PowerPoint (.ppt or .pptx formats)
 2. Adobe PDF (.pdf) [full size, one slide per page, ensuring pictures are very clear]Other formats may be acceptable (e.g. if they can be easily converted to .ppt format), but you need to verify the alternate format with the instructor prior to submission. Be careful not to embed other files or expect applications (like Apple movies) to be available on the notebook computer used for display.
- At least one slide showing a detailed block diagram of the project must be presented. There are enough people in the class that it will be impossible for everyone to get close to each project as it is being shown. Therefore, there will be an LCD projector in the room, and each group should show at least one slide that provides the whole class with an overview of the project. At a minimum, the slide should show a detailed block diagram of the hardware and/or software. More than one slide is fine; however, you have to carefully and wisely budget your time, since you get a very limited amount of time to show your work and to answer questions. **DO NOT USE ALL YOUR TIME SHOWING SLIDES AND THUS RUN OUT OF TIME FOR YOUR ACTUAL DEMO.** You do NOT need to generate paper handouts for the whole class.
- During the project demonstration, the project team should talk about design and implementation details. The team should discuss any major engineering challenges encountered during the implementation, and discuss what the team learned during the implementation.
- There may be as many as 20 projects presented during the final project demo. If each group uses 9 minutes (~2 minutes for slides, ~5 minutes for demo, ~1 minute for questions, 1 minute for transition between projects), then 3.5 hours will be required for the Wednesday night demos. Therefore, please be prepared. If you don't have much to show, you do not need to use up your 8-9 minutes, and that will allow the class to get through all the projects in a shorter amount of time. If you do not demo any functionality, then your total presentation time is limited to 5 minutes.
- Speak loudly when presenting your project. The room is large and there are a lot of people, and you need to make sure everyone can hear you.
- Actively listen while others are presenting. Don't have side conversations or be working.
- Schematics for the final project must be complete, and a copy of these schematics must be available during the final project demonstration. Have your schematics available (e.g. on the computer) during the presentation in case the instructor has any specific questions about your circuits. You do not need to turn these in during the demo.
- Don't be doing project development work on demo day.
- The lab stations must be shared among all the student project teams. After your demo, you may need to move your project in order to allow another project team access to the lab station.
- The instructor plans to have a digital camera at the demos and hopes to take a picture of each project and project team.

What to Submit by May 5th at 3:00 p.m.

1. Each team must submit a hard copy honor code pledge, similar to the pledge on every lab submission sheet. All students on the project team are expected to sign the honor code pledge and to identify any work that is not their own. The final project honor code sheet is available on the course web site.
2. Each project team must provide the instructor an electronic copy of all project files, including source code (both firmware and software in source format, not as a PDF), header files, makefiles, and any other documents, such as unique data sheets and app notes. Teams will provide the files via CULearn, CD/DVD, or by e-mail. See instructions from professor near the end of the semester.
3. Include your code and makefiles for each of the labs in a separate folder (as well as other interesting tools during the semester). Each person on a team must submit their code.
4. The instructor usually prefers a CD-R disc that contains all your files. If you provide a CD, then please write your name(s) and "Spring 2012" on the CD (careful, don't scratch the disc) and organize the files on the CD using folders. If working in a team, the team needs to submit only one CD that contains all the files listed below, with a separate folder for each student's independent work. Keep folder names relatively short (fewer than ~14 characters). A suggested folder structure is:
 - Data Sheets (data sheets, application notes, and other reference documentation)
 - Schematics (final project schematic source files, Orcad .DSN files, libraries, etc.)
 - Layouts (layout files, including libraries and Gerber files, if a PCB was constructed)
 - Code (final project source code with header files, makefiles, etc. **All C code must have .c extension.** Files must be in their original format and able to be compiled by the professor in Eclipse/SDCC.)
 - Utilities (any special free software tools that you might have used on your project)
 - Presentations (current topics, final project PDR, and final project demo presentation files)
 - Pictures (photos and/or videos related to the project)
 - Labs (Code and schematics for Labs 2-4, for each student on the team)
 - Homework (optional; include if you have any relevant or interesting files to submit)
 - Report (any rough drafts and supporting documents that exist at this point in time)
5. If providing the electronic files by e-mail (or CULearn), organize your files as described above and use WinZip to combine all your files before e-mailing. Try to limit e-mail size to a maximum of **3MB**. Do not e-mail files larger than 5MB without contacting the instructor first. Don't include other comments in the e-mail body that aren't related to the electronic files submission; **use a separate e-mail for all other comments.**

Other items to return by this date include:

- Tool Kit (in plastic bag)
 - **ZIF socket** (in protective container)
 - **NVRAM** (in protective container)
 - wire wrap tool with wire stripper
 - cutters
 - long needle nose pliers
 - serial cable (with the same number as the one signed out earlier in the semester)
 - power supply
 - floppy diskette
- LogicPort Logic Analyzer and all accessories**
- Money for parts you purchased with an IOU or for items signed out and subsequently lost or broke
- Other items you borrowed during the semester, such as digital logic probe, oscscope probes, etc.
- Documentation (Borrowed books, etc.)

What to Submit by May 7th at 5:00 p.m.

Final Project Report Requirements

- Each team must submit one copy of the final project report to the instructor electronically **via CULearn**. Acceptable document formats include Adobe PDF and/or Microsoft Word (.doc or .docx). The submission should be a single file which contains all parts of the report, including cover page, body, and appendices. If necessary, appendices may be included as separate files; however, all files submitted must be clearly identified, commented, and well organized. Your grade will depend on how well you organize the report and whether you make the report easy for the instructor to read and grade.
- The submission file name must follow the following naming guideline. The name must begin with "S12", followed by the student's last name (family name), followed the word "Report". An example file name for a team of students named "John Adams" and "Thomas Jefferson" with a project related to the ZigBee protocol would be "S12_Adams_Jefferson_Report.zip" or "S12_Adams_Jefferson_Report.docx".
- The final report must represent a professional effort, and must be typed on a word processor. The report must be well organized, easy to read, free from spelling/grammatical errors, and must include page numbers.
- The report must describe the project, and detail its design and implementation. Assume that the reader is familiar with the 8051 architecture and instruction set. Make sure that the report doesn't just identify what was designed, but also details the design considerations and what was learned.
- **Plagiarism will not be tolerated.** Copying material from the web or from data sheets without giving credit is plagiarism. Figures may be borrowed from data sheets or application notes, provided that credit for each figure is clearly given in the report (e.g. "Fig. 3 from ABC Company datasheet XYZ"). Borrowing a sentence or paragraph is reasonable (as long as credit is given), but do not borrow pages of material and include that in the body of your report. Students are expected to understand plagiarism and how to properly cite the work of others. Refer to the course web site for information regarding the CU Honor Code and Plagiarism, and refer to <http://en.wikipedia.org/wiki/Plagiarism>.
- The report must include a cover page. The cover page must include a descriptive title of the project, the team members' names, the course number and title, and the due date. This must be the front page of the report.
- One possible way to organize the report follows: cover page; table of contents; introduction and overview; technical description/details, results/error analysis; conclusions; future development ideas for the project; appendices. Hardware, firmware, and software design (if applicable) must be detailed. A template will be available on the course web site.
- The details section should include small figures, diagrams, or code fragments if needed; however, large schematics and code listings should be placed in an appendix. All schematics and appendices must be titled and numbered, and referred to by number in the report. **All schematics, labels, and graphics must be large enough and clear enough to be read easily.** You should verify this after you print out your report to a PDF file.
- Appendices should include copies of any special data sheets or references. The report does not need to include every page of an extremely long data sheet or application note (just include the most relevant pages in the report, and submit the entire data sheet or application note in electronic format as a supporting document). URLs for web sites containing data sheets, application notes, and other specs must be listed in the report.
- A full circuit schematic must be included in an appendix. The schematic should be done with a schematic capture program such as Orcad, which is available in the lab. Don't put too much information on any one sheet. Label all signal names and include pin numbers.
- Source code listings for project firmware and software must be included in an appendix. Code must

follow general good coding standards. Useful, descriptive comments must be included in the code. Use blank lines and white space wisely in your code to improve readability. It is suggested that code listings are printed "2-up", with two pages of code per side of paper. If you also print a hard copy, double sided printing is encouraged, as long as the resulting appendix is easy to read without requiring that the report be rotated 180 degrees to read successive pages of the code.

- A project parts/cost list (bill of materials or BOM) must be included. This BOM should indicate the source, part number, and cost of items specific to the final project, not parts used for Labs 1-4.
- Firmware, software, or hardware which is highly leveraged or copied from another source may be used if legal; however, credit **must** clearly be given to the original author/designer in code comments and project documentation. Student-created code and hardware will be counted more heavily in the project grade and should be clearly identified in the report. Company confidential information must not be included in the report, since these reports may be viewed by a wide population. **Each leveraged section must be clearly identified.**
- **Each code file header must contain the author's name and must identify whether any leveraged code is contained in the file.**
- **Before submitting your report, double check to make sure all the figures, schematics, and code are included and are legible!**
- All reports submitted will become the property of the instructor.