ECE230 term project: Embedded System Design

Assignment

For your term project, you will design an embedded system. That is, a microcontroller-based system which senses one or more inputs and produces one or more actuating outputs in response to the current environment or context. The term project is an open design project which you may choose. This is a group project, to be completed in groups of two (unless otherwise approved). There are a set of criteria which all projects must include which are listed below:

Requirements for Completion

- The project must use the MSP432P4111 (or other microcontroller with approval)
  - Additional microcontrollers may be used in project if needed (parallel process)
- The embedded circuit must include at least one sensor (input)
  - Pushbutton, analog signal, serial communication input, etc.
- The embedded circuit must include at least one actuator (output)
  - LED, LCD, motor, etc.
- The embedded project must make use of at least one timer
- The embedded project must make use of at least one interrupt

You may choose to implement whatever project you desire which meets these criteria. Keep in mind, however, that only a successfully demonstrated project can receive a grade. Therefore, you should approach your design in modular steps – have a simple version of your project which you will initially implement and verify, then add an additional level of complexity and repeat until the project is fully completed. Be sure to always save the most recently "fully-functional" iteration, so that if you cannot get the next feature addition to work you will at least have a functioning project to demo.

Grading

The assignment will be graded based on quality of implementation, as well as complexity and creativity of design, as shown in the rubric below. For quality, complexity & creativity, some metrics which will be especially considered include:

- Use of MSP432 peripherals that were not implemented in a lab exercise or project
- Use of MSP432 peripherals in a significantly different configuration (e.g. SPI configuration rather than I^2C)
- Implementation of interfaces or communication schemes which are unique to the course
- Unique application of course concepts
- Build quality
- Software quality

The grading rubric is shown in the following table. More detail of each deliverable is provided in the next section. See Moodle for due dates.

Deliverables	**Weight**
Project proposal	6%
Progress report	12%
Final project report	30%
Successful demonstration of project meeting requirements criteria	12%
Final source code	10%
Quality, Complexity & Creativity	30%

In addition to deliverable timelines shown, components of deliverables may require approval and your instructor may ask you to resubmit if necessary. For example, the project plan or test plan submitted in the progress report.

Deliverables

i. **Project proposal – submit to Gradescope**. This proposal will describe your project goal, along with some initial plans. An example proposal is available on Moodle, though it will not match all requirements below. The project proposal should include:
   - **Introduction** – briefly introduce the goal of the project
   - **Objectives and specifications** – indicate specifications for project and objectives for system, may include bulleted list.
   - **Conceptual design description** – system overview and description of proposed solution for the project. It is okay if your planned solution changes over the course of the project, changes to your plan can be indicated in later progress reports. A diagram of system overview would be useful.
   - **Tasks and priorities** – subdivision of project into tasks and subtasks necessary for completion. May be a bulleted list.
   - **Schedule bar-chart** – create a schedule for completion of the project. This may be represented as a bar chart indicating the timeline for each subtask. **Define modular phases**.
   - **Budget** – provide a list of parts/materials expected to be needed for completion of the project and expected cost.

ii. **Progress report – submit to Gradescope**. Report on progress to-date on project. Detail any changes to project design/goals, if applicable. Progress report should include:
   - **Current status** (percentage of work done and any delay) – detail work completed
     - Include team member contributions
   - **Current and future work** – include team member assignments
   - **Project plan** – software plan which should include flowcharts for each expected module
   - **Test plan** – complete test plan (requirement, test procedure, and pass criteria) for three well-defined and measurable requirements

iii. **Final project report – submit to Gradescope**. This report summarizes your project, and provides documentation of the design and usage of the system. Your final report will also serve as your verification document. Please **do not** include the entire source code in your report, there is a separate dropbox for that. You may include code snippets for reference as needed. The report should include:
   - **Introduction**
     - Include overview of how minimum criteria were met
   - **Work completed and past work**
     - Include references for all sources used (libraries, sample code, project ideas, etc.)
     - Detail **your** contributions (additions and modifications to work)
     - Specify additional features and considerations for complexity/quality score
   - **User's Manual** with photos or graphical illustration
   - **Hardware design** and implementation
     - Should include circuit diagram
   - **Software design** and implementation
     - Should include flowcharts
   - **Test plan, verification, and test results** (test plan previously approved from progress report)
   - **Bill of materials**
   - **References and acknowledgement** (as needed)

iv. **Final project demonstration – during final exam timeslot**. Demonstrate your final project and view other demonstrations. Plan to be present for the entire symposium.

v. **Final source code – submit to Gradescope**. Submit all final source code. Use good formatting and clean code of unnecessary commented-out lines. Include header and comments in code.

Notes

If you are thinking of having anything fabricated – talk to the ECE technicians now! They need advance notice.

It is OK to look for ideas on the Internet and to modify another project, however, **you must**

- Give credit to others (including links and/or references to original sources)
- Show how much is new and your own work
- This information should be included in report and source code
  - Omission of this information is **academic misconduct and grounds for failure.**

Define a project that can be done in 2+ weeks

- 48 person-hours (for two people)

Caution: work incrementally from simple implementation to complex

- Divide a project into a few phases: phase 1, phase 2, …

Always be ready to demonstrate a simpler and working prototype

Divide program into subroutines and work on one at a time

- **Keep software versions in separate project folders and back up projects to cloud storage**
  - Avoid losing programs due to computer failure
  - Avoid being unable to revert to working version of code

You may use TI BoosterPack plug-in modules

Limited list of examples of past projects

- Walkie-talkies
- Music players and instruments
- Bar code scanner and database
- Intrusion detection and alarm system
- Remote controlled vehicle and garage door
- Bluetooth controlled vehicle
- LED propeller display
- Portable multimeter
- Anti-gravity water fountain
- Laser harp
- 2-player LED target game
- Equalizer display
- Vending machine
- Claw game