Embedded System Design Mini-Project

Introduction:
Design is an extremely important part of the curriculum – virtually every employer of engineering graduates is interested in the nature and extent of design projects completed by their potential employees. In upper division lab courses such as ECE 362, it is therefore essential that students have the opportunity to complete open-ended, design-oriented exercises. This “Mini-Project” represents our effort to provide such an opportunity.

From a pedagogical point of view, this project is based on the premise: “The best way to learn about a topic is to write your own question and then proceed to answer it.” In other words, the process of formulating questions and exploring alternate solutions can help elucidate difficult concepts. To provide as much flexibility as possible concerning potential topics you may wish to explore, a wide range of Mini-Projects may be completed.

Mini-Project Functional Requirements:
The basic requirement is to design your own embedded system that makes meaningful (and hopefully creative) use of the 9S12C32 microcontroller resources – in particular, the on-chip peripherals. Also, your project must operate in a “turn-key” fashion, i.e., your code should be stored in the on-chip flash memory and start up directly out of reset (or automatically when powered up). Finally, the project should be “stand-alone” (self-contained) – any interface hardware required should be “packaged” appropriately. All components used must be supplied by your team.

Mini-Project Report Requirements:
The “core” of your written report (Sections 1-6, listed below) should be approximately 6 pages in length and printed in point size 12 with vertical spacing 18 (space-and-a-half). Your report should be organized as listed below (a Word skeleton file is provided on the course web site for your convenience):

- Evaluation Form/Cover Sheet
- Table of Contents
- Section 1: Introduction (a functional description of your project and the role played by each team member)
- Section 2: Interfacing (a description of the external interfaces implemented)
- Section 3: Resources (a description of how the on-chip resources are used and initialized)
- Section 4: Software (a description of software organization/structure)
- Section 5: Packaging (a description of the project packaging)
- Section 6: Summary and Conclusions
- Section 7: List of References (URLs, data sheets, application notes, etc.)
- Appendix A: Individual Contributions and Activity Logs
- Appendix B: Interface Schematic (and PCB Layout – if applicable)
- Appendix C: Software Flowcharts
- Appendix D: Project Packaging
Mini-Project Demonstration and Report Submission:
- Projects can be demonstrated to either professor any time the week of December 4—**the absolute deadline is NOON on Thursday, December 7.**
- On-line submission of report, code, and bonus credit files must be completed **no later than Friday, Dec 15, at NOON**

Bonus Credit Opportunities:
- Creation of a poster illustrating the device constructed and documenting how it was designed (**printed copy** of poster required for Design Showcase participation): **up to 1% bonus credit**
- Creation of a YouTube video illustrating project packaging and demonstrating functionality (**live hyperlink** to video must be included on Poster): **up to 1% bonus credit**
- Participation in Spark Challenge on Friday, December 8 (**prior submission** of both a poster and a video by deadline is required to qualify for participation); registration for and attendance at event is required: **up to 1% bonus credit**
- An **early completion bonus** for project functionality demonstration of **up to 0.5% bonus credit** will be awarded as follows:

<table>
<thead>
<tr>
<th>Project Demonstration Date</th>
<th>12/4</th>
<th>12/5</th>
<th>12/6</th>
<th>12/7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early Completion Bonus Awarded</td>
<td>0.5%</td>
<td>0.4%</td>
<td>0.3%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Team Responsibilities:
While the team of **two to four** members work together in order to complete the project, each member of the team is to play a specific role. Specific responsibilities are outlined below:

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
<th>Tasks</th>
<th>Document Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDP (team, documentation, packaging) Leader</td>
<td>Overall team coordination, final editing of documentation submitted, product packaging</td>
<td>Delegation of tasks, planning of team time, acquisition of resources, editing report sections together, citing references, assisting with debugging of hardware and software, packaging project</td>
<td>Introduction, Packaging, Summary/Conclusions, Activity Logs (App. A), Packaging (App. D),</td>
</tr>
<tr>
<td>Software Leader</td>
<td>Overall software planning and development</td>
<td>Coding (with comments) and debugging</td>
<td>Software Flowcharts (App. C), Software Source File (submitted on-line)</td>
</tr>
<tr>
<td>Interface Leader</td>
<td>Interface hardware, including OrCAD or Eagle schematic</td>
<td>Schematic entry, interface hardware construction</td>
<td>Interfacing, Schematic (App. B)</td>
</tr>
<tr>
<td>Peripheral Leader</td>
<td>Peripheral module utilization, initialization, and device drivers</td>
<td>Coding (with comments) and debugging</td>
<td>Peripherals, Software Source File (submitted on-line)</td>
</tr>
</tbody>
</table>

The above itemized tasks are to help balance and distribute responsibility for the project. Being the “software leader”, for example, means that you are to take the lead in planning, writing, and integrating the software. It does **not** necessarily mean that you are the only one developing and debugging the software, however.
Suggestions for Possible Ideas:
Meaningful, significant extensions to the last set of lab experiments are potential ideas:

- **Shot Clock** – add PWM-based “buzzer” and loudspeaker/amplifier, make shot clock setting programmable (write setting to flash memory so value can be “remembered” when power removed), add LCD display for shot clock operator, incorporate into basketball hoop assembly, play “themed music” when shot clock reaches certain values.

- **Digital Volt Meter** – add capability to convert all eight input channels, replace terminal with LCD display, add “range-select” capability, add current measurement capability, add impedance measurement capability, add (PWM generated) “continuity beep” capability.

- **Reaction Timer** – add a PWM-based audible alert (with programmable frequency and digital volume control), provide capability to select visual/auditory reaction time test, create new “pushbutton circuit” to eliminate sampling ambiguity, keep history of (say) ten best reaction times in flash memory and display them on start-up screen.

- **Motor Speed/Tachometer** – add “closed loop speed control” capability (i.e., run motor at a programmable “set point” speed), add capability to control motor speed based on frequency of an audio input signal (as an option), add audio input circuit/microphone preamp that allows “singing” to control motor speed (“the hummer”).

- **Programmable Waveform Generator** – synthesize (via table lookup) variable frequency sine, triangle, square, and sawtooth waveforms; reconstruct signals using PWM.

Additional Hardware Suggestions:
Many projects can be completed using the components found in various ECE parts kits. However, if the design of your project requires additional hardware, here is a listing of resources:

1. Contact the company that has developed the part – you may be able to obtain a sample. Make sure that you are able to obtain complete documentation on the part especially if it is relatively complex. *Allow for plenty of time.*

2. Order the part from a supplier – the following are recommended (always order at least “one extra”):
   - Jameco (jameco.com) – OK selection, great prices on small components
   - Digi-Key (digikey.com) – Excellent selection, a bit higher in price over Jameco
   - Acroname (acroname.com) – Great resource for interfacing (LCD, Ultrasonics)
   - All Electronics (allelectronics.com) – Lots of good quality surplus parts

3. Try to obtain the part locally (e.g., Lafayette Electronics Supply)

Careful planning and providing plenty of lead time is the key to a successful project.