

ECSE-426 Final Project Report Guidelines – Fall 2009

Objectives

The final project report should present a formal but concise summary of your team work in the realization of the ECSE-426 final project.

Overall Requirements

On the first page, an executive summary should briefly describe the project and clearly state which options were selected and delivered during the demonstration. The report length should be around 10 pages (5 pages if your team has only 2 students), excluding the source code/scripts/notes/performance data that should be included in the appendix.

Required Sections

Executive Summary

- Brief description of the project
- Listing of important features in bullet form (similar to a datasheet)
Examples : Graphical display format (2x24 character-based, graphical dot-matrix), software architecture (use of RTOS, state machine controller), Input device, wireless features (programmable channels, etc.). It should be written a bit like a marketing pitch and ideally should highlight the “best” features of your project.
- Team member's name and responsibilities in the project.
- A picture (or a few) of one of the gaming unit. One picture of the user interface in typical usage (e.g. showcase the menu system).

Hardware Overview

- Block diagram of the hardware interconnection of peripherals. This section should indicate which ports of the microcontroller was used to interconnect the various peripherals. If some interconnection passes through the CPLD, it should be clearly indicated and if the CPLD perform any logic operations, they should appear as a block in this diagram. The block diagram may cover an entire page. It should carry enough details to unambiguously describe the whole system (it must reflect what we can see in the source code). Analog circuitry may be illustrated on this diagram or abstracted in a block which may be described in a subsequent part of the report.
- Details of the wire-wrap. Since some peripherals are connected via wire-wrap, we expect to see at minimum, a small table listing the connectivity between the pin headers on McGumps and your peripheral. Example (J2-P10 => LCD-nWE).

Software Architecture

- A description of the overall software architecture. It should present a concise description (diagram is useful here too) of the major control elements and structures of your software architecture. This section should describe the files present in the project. If you use standard constructions such as buffers/queues or shared data structures, you should discuss those. If you used any mechanisms to

protect shared structures from concurrent access, this would be a good place to discuss your strategy.

- Details of each major software blocks. Here each team member should have the opportunity to write 1-1.5 page(s) describing their project contribution and design. If you implemented a specific algorithm or wish to make a point on the implementation, then you should refer to the appendix source code to direct the reader and showcase the implementation (e.g. our gaming unit made use of wavelet compression of images [appendix wave_compress.c, lines 200-7500]). The source code in appendix should be included and ideally, each line should be numbered. Quite a few tools are available to pretty-print C source code and append line numbers in front of every line. Another way would be to refer to a function name. Just make sure the reader can find the information efficiently.

- User interface details and survey. You should detail the user interface of your application and include the results from the mini-survey on its useability.

Problems or Unresolved Issues

This section should discuss any issues you encountered during development and if you found a solution or workaround. If any limitations exist in your design, you should describe it briefly and offer an explanation on how it could be fixed.

Conclusion

Your team can summarize their project experience and results. If any important limitations were found with the given hardware or if you have contributions for future improvements in the course project, you may include them here.

Appendix

The appendix should include ALL your source code, including modules that you may have retrieved from TI or the web and CPLD VHDL file(s). If you used FreeRTOS, you don't have to include the source code, but make sure you include the customized kernel configuration file (or any modified files from FreeRTOS) and clearly state which version/revision of the kernel you used. Ensure that code that was not produced by your team is clearly identified (either by leaving the original copyright header or by writing it at the top of the file).

Source code should ideally be included with line numbers which you may refer to in the body of the report. Using `enscript` is an easy way to produce very pretty source code (I use it in Linux, but should work in Cygwin). The command below will produce the required output (in PDF):

```
enscript --pretty-print --color --line-numbers -o hello.pdf hello.c [ more input files if needed ]
```

You may use any other program that can produce a clean PDF file from the source code.

The appendix may also include more pictures of your unit, relevant snapshots of the debug interface, LCD screen, and a team photo if you wish.

Submission

The ECSE-426 project report should be submitted electronically via WebCT by the team leader in each team. It is not required to print a paper copy. The report should be a self-contained PDF file. If you wish to submit a paper copy, you will have to notify the TAs that you have deposited it in the assignment box.