ECE4703 B Term 2008
Project Code and Laboratory Report Guidelines

Documenting and communicating your work to others is an essential part of the development process and is something that all engineers must master. In ECE4703, the laboratory report is a tool for you to capture the work you did on the assignment and to communicate this work to the Instructor and Teaching Assistant. A good laboratory report will remain useful long after the exercise, even the course, has completed. This section of the assignment lays out the general guidelines for laboratory reports and project code submissions in ECE4703.

1 Report Guidelines

All laboratory reports in ECE4703 should follow the following format:

1. One **Cover Page** including the course number, the date, a title, and the report authors.

2. A concise **Introduction** including the problem description as well as goals and specifications. Do not just copy and paste the problem description from the assignment. Use your own words and try to motivate the reader into reading the rest of the report. You may want to even conclude the introduction section with a highlight from your solution and/or results.

3. **Background**. This section should include a concise discussion of any background information (especially theory) that is necessary to understand the methods, solution, and/or results. You can assume that the reader has a junior-level ECE education in the sense that you don't need to explain every little detail, e.g. Fourier analysis. You should, however, highlight any special theory or background material that will help the reader to better understand your work.

4. **Methods**. This section should include information about how you developed and tested your solution to the problem. Any tradeoffs that were considered should be discussed here. Any special techniques that were critical to the solution should be discussed here. This shouldn't be written as a chronological diary of your work but, rather, as a logical justification of how you came to your final solution.

5. **Problem Solution**. This section presents the specifics of your final solution. Your solution should be presented at two levels: first an overview and then the details. The overview is necessary to provide context to the reader. The overview should contain high-level flow charts and/or state diagrams to allow the reader to get an overall understanding of your solution. From this overview, you can then document the details of each specific part of your overall solution. You are encouraged to use diagrams liberally to illustrate details of your design. You don't have to justify your choices here — that was already done in the methods section.

6. **Results**. This is where you present your results as well as answers to any specific questions in the assignment. You are encouraged to try out more than the minimum asked in the lab assignment. Always explain the precise conditions of each test, discuss what the results mean, and provide at least some intuition as to why they make sense. Results without explanation have little value. Where possible, refer to the theory in your background section (or cite a textbook) to justify your results. Use the appropriate technique to most effectively communicate your results: sometimes tables are the best way but it may also be appropriate to include plots generated in Code Composer Studio and/or screenshots from the oscilloscope.

7. **Conclusions**. In this section, you should summarize your accomplishments, document any lessons learned and any insight gained, highlight any particular struggles you had in developing the solution, and even suggest directions for future research and/or development. Did you uncover any errors in the problem description? If so what did you do about them? If you were allowed different constraints in the laboratory could you have designed a better, faster, or cheaper system? If so, how?

8. **Appendices** (not always required). Appendices are a good place to put things that help to document your work but don't contribute to the overall flow and readability of the report. If you collected a lot of data and it fills a table that takes up 3 pages, it's probably better to just discuss a summary of the results in the Results section and refer to the table in the appendix.

9. **References**. You should document the reference sources you used. That way, if you ever need to find the information again, you'll know where to go. Web references are ok but are less preferable to printed references, e.g. textbook or published papers.

10. **Code**. Do not put full code listings in your report. It is ok to put code snippets in your report to illustrate important details of your methods or solution. The full projects, however, should be compressed into .zip files and submitted electronically as described below.

You should use 1.5 line spacing for your report to allow for written comments from the grader. Break large sections into smaller subsections (and even subsubsections) as necessary to improve readability. Your grade is not based on the length of the report; a concise, clearly written report with the key results is much better than a long, wordy, confusing report.

Your report should be of professional quality and, in addition to being a pleasure to read, should look nice. All figures must be numbered, have a descriptive caption, and referenced in the text. All tables must be numbered, have a descriptive caption, and referenced in the text. Good visualizations are important. Sloppy diagrams, plots with missing axis labels, plots with axis dimensions that don't make sense, and plots that fail to show the important features of the results will receive little or no credit.

Your report should be submitted to the Teaching Assistant and Instructor by email in .pdf format (not .doc format) prior to the deadline stated on the laboratory assignment. You will need to make sure the report is less than 10MB, otherwise email with the report as an attachment may not be delivered. If your report is larger than 10MB, it is probably because you have included one or more large figures. You can resample these figures (and perhaps change the color depth) to reduce their size if necessary.

And finally, don't plagiarize.

2 Project Code Submission Guidelines

In addition to submitting your report, you are also required to submit one .zip file for each Code Composer Studio project in the laboratory assignment as well as any Matlab code required in the assignment. Your CCS projects should be in individual subdirectories of the

```
C:\CCStudio_v3.1\myprojects\
```

directory. You should compress each of your project subdirectories into individual .zip files and submit these to the Teaching Assistant and Instructor by email prior to the deadline stated on the laboratory assignment. You should also attach Matlab .m files, as required per the assignment, to your project code submission email. These do not have to be compressed.

The Teaching Assistant and the Instructor will test your CCS projects to be sure that they build and that they function correctly. To be safe, prior to submitting your project code, you may want to take your .zip files and install them in the `myprojects` directory on another computer to ensure that they build without errors and function correctly. Your Matlab projects will also be tested for correct functionality.

All of your code will also be read and graded on its readability. Commenting your code is a good practice even if you are the only one who will ever see it, and it is critical when you are working in teams. It is very important that you liberally comment any C, ASM, and Matlab code that you submit to the Teaching Assistant and Instructor. It is recommended that you place a header comment on the first few lines of any code that you write. The header should contain the course number, the date, the authors' names, and at least one sentence describing what the code does. Feel free to add more to your header (you could describe the function and permissible range of certain parameters, for example) as appropriate.

All code that you submit in ECE4703 must represent your own work. Do not copy code from other teams or any source other than the course notes or the examples in the course textbook.