COE3DY4 Project Report

Group number
Student names
McMaster email addresses
Date

Your project report must be pushed to your COE3DY4 project repo (in the `doc` sub-folder) before the end of day on Friday April 8. The report must use letter-size page format with a 12pt font size, single spacing, and minimum margins of 1.75 cm. There is a total page limit of 10 pages (including all figures and tables). The project report should be formatted as described below.

1 Introduction

Use a short paragraph to summarize the scope and objectives of this project, both from the application standpoint, as well as the practical set-up on which the software implementation of a radio receiver is expected to run in real-time.

2 Project overview

This section should contain a few paragraphs focused on the big picture of the project, such as frequency modulation (FM), software defined radios (SDRs), basic building blocks, e.g. finite impulse response (FIR) filters, FM demodulators, phase-locked loops (PLLs), re-samplers, radio data system (RDS), ..., and how these blocks are chained in a signal-flow graph to implement an FM receiver in software. No specific details concerning your implementation choices should be included in this section. Rather, the details from this section should provide the generic background needed to explain your specific implementation in the following section.

While no hard limits are imposed on page counts for individual sections, the recommendation is to use at most one page for the first two sections.

3 Implementation details

This section is concerned with your **own** implementation. You should start actually with the labs because some of the most basic building blocks, such as impulse response generation and digital filtering, have been developed at that stage of the course; then gradually proceed to the project work. Elaborate on how you have progressed step-by-step through the spec.

A normal expectation is that, for each path in the signal-flow graph, the real-time implementation has been preceded by modeling. Explain your validation strategy and how you have iterated between modeling and implementation. In particular, as you discuss a particular section from the spec, provide details on how you have addressed conceptual problems, e.g., incorrect filter settings, vs implementation-specific errors, e.g., wrong sizing of arrays in C++. Needless to say,

the potential space of both conceptual problems and implementation-specific errors is limitless and you have to discuss the ones based on your group's experience. Another expectation is that the knowledge, understanding, experience, ... that you have developed in an earlier stage of the project has been leveraged in order to proceed faster in the later stages of the project.

Use as many examples, figures, ..., as you see fit to articulate your modeling, design and validation experience. Be specific about the unique bugs that you needed to deal with (either of conceptual or implementation nature) and the thought process you have been through in order to address them. If you believe that you have produced any out-of-ordinary (or, even better, innovative) solutions, by all means, take your time to explain them.

This section is expected to be the longest section in your report. Use sub-sections as you see fit.

4 Analysis and measurements

The aim of this section is to demonstrate that you have a solid understanding of why your implementation works in real-time on an embedded hardware platform like Raspberry Pi 4 (RaspPi). It will also help you better appreciate the limits of what can be implemented on a particular hardware platform. Assuming a default number of filter taps N_{taps} equal to 101, the following analysis is requested:

- Analyze how many multiplications and accumulations are needed per audio sample (for each of the modes 0 to 3 for both mono and stereo);
- Analyze how many multiplications and accumulations are needed to produce one bit in the RDS path (for both modes 0 and 2, assuming you have attempted this part of the project);
- Apply the same analysis as above for non-linear operations (`atan2`/`sin`/`cos`/...).

You must perform runtime measurements on the RaspPi for different code sections and relate your analysis to these measurements. Note, code sections should be the basic building blocks from the signal flow graphs from the project document, e.g., each of the FIR filters, the FM demodulator, the PLLs, ... The runtime measurements and the above requested analysis for N_{taps} equal to 101 should be done for all of the supported modes of operation, i.e., modes 0 to 3 for audio and modes 0 and 2 for RDS, assuming you have attempted this part. Note, for the measurements to be meaningful you must also report the total number of seconds for which the I/Q samples (that are piped into your SDR software) have been acquired. It is to be expected that the absolute values for the measured runtimes for each code section will be different for the RaspPi and your local host; nonetheless their relative sizes to each other can be used to relate them to your analysis.

To better appreciate the impact of the number of filter taps on both the signal quality and the runtime, you should carry out an additional investigation for N_{taps} equal to 13 and 301. Measure the runtimes on the RaspPi and try to explain them based on the analytically expected number of multiplications and accumulations. You should also observe the quality of the audio, as well as the ability to synchronize the frames in the RDS bitstream (assuming you have completed this part). Provide an interpretation for the differences that you observe when you vary the number of filter taps N_{taps} from 101 to 13 and from 101 to 301. Note, for this particular investigation it is sufficient to discuss the impact of changing N_{taps} only in the mode 0 of operation.

The recommendation is to use approx two pages for this section. You should use tables to summarize your analysis and runtimes before interpreting them in the body of the report.

5 Proposal for improvement

As any project reaches its final stages, there are always ideas for improvement. Note, by improvement it is **not** meant completion of the work-in-progress. Rather, considering the current status of your project, provide comments on how the existing work can be improved to provide a better user experience or increased productivity or higher performance.

Comment in one paragraph on at least two features to be added to your software system: at least one feature should benefit the users of your system; at least one feature should better your own productivity when expanding/validating/improving it. Note, for each feature there has to be a logical explanation on why it is technically feasible to add it.

Comment in another paragraph on how you can further improve the runtime performance of your system, in such way that it can still run in real-time even when the hardware platform is of lesser capability. Provide technical justifications.

The expectation is to have a logical path from the current status of your project to the proposed improvements. How "creative" is the proposal is of lesser importance than the depth of the technical argument that is provided.

6 Project activity

Provide a table with eight rows, one for each week of the project (labeled as Feb 14 ... Apr 4) to explain the following:

- What was the progress in each of the eight weeks of the project (for example, midway through week 1 the project spec has been released and .., week 2 was the midterm recess and ..., throughout week 7 we have finalized the source code and ..., during week 8 we wrote this report and ...);
- What were the contributions of individual group members in each week of the project. As you summarize each contribution, you should refer back to the implementation section.

While facts need to be agreed on (someone has either worked or not on a particular task), opinions/viewpoints/... can differ (individuals can perceive their own contribution or the contribution of others in quite different ways). As the assessment process runs its due course, the discrepancy in viewpoints is expected to be resolved.

Each individual group member has the right **and** the responsibility to provide his/her own contribution statement. While other group members may or may not agree with some viewpoints, the onus is on the individual whose name is associated with the tasks associated with his/her name in the above table to assess the statement's accuracy in the submitted version of the project report.

7 Conclusion

Use this section to summarize your overall learning experience. One paragraph should be sufficient.

8 References

Provide a list of all your references (project spec, class notes, web resources, ...).