

Laboratory 3: "Compiler optimization"

Pre-lab Submittal:	None.
Demonstration and Submittal:	Due by the end of the day of the Week 4 lab session. Late submittals will be penalized per the course syllabus.

Objectives

- Gain experience setting optimization levels in a compiler.
- Observe and analyze the effects of various optimization settings on the code size produced, as well as execution speed.

This is an individual lab. Each student must independently complete this assignment. While discussing ideas and potential solutions with your classmates is permitted, sharing code is prohibited.

All software must be written in C using the CCS IDE.

For this laboratory, you will explore the optimization performance of the TI Arm Compiler. You will need to write C code to run on the MSP432 system, then experiment with different optimization settings and observe the performance in terms of code size and speed.

Specifications:

- Implement a time measurement mechanism in the code (e.g., measure the time to go through the main loop). The best mechanism for this would be toggling an output pin (i.e., set high at start of loop, set low at end) and measuring the time on a scope.
- Make the code large enough and complicated enough to provide opportunities for optimization. Incorporate as many of the optimization techniques discussed in class as you can in the C code, i.e., put things in the code that the techniques discussed can optimize. Use comments to identify each of these techniques in your code. Examples of things you might include:
 - Nested loops, with inner loop small and having minimal iterations
 - Calculations that could be simplified
 - Use some small functions
 - Etc.
- Code size can be found in the MODULE SUMMARY section of the 'project_name.map' file located in the Debug directory of your project.
- Take data (use spreadsheet provided) for the following minimum set of optimization settings:
 - Optimization = off, size/speed = 0 (baseline results)
 - For Optimization settings of 0, 2, 4, look at size/speed settings of 0, 2, 3, 5 (12 combinations).
 - Report both code size and speed for each combination, as well as percent change.
- **Note: Part of the grade will depend on how many of the techniques are incorporated into the code, so try to get as many included as you can.**

Submittal

In your lab report, include:

- Printed copies of your source code
- Data collected – in the table provided
- Describe why you included the things you put in your code, i.e., why did you think each section could be optimized by the compiler.