Product Research

sdmay25-23:
James Joseph
Samuel Lickteig
Alix Noble
Andrew Sand
Owen Sauser

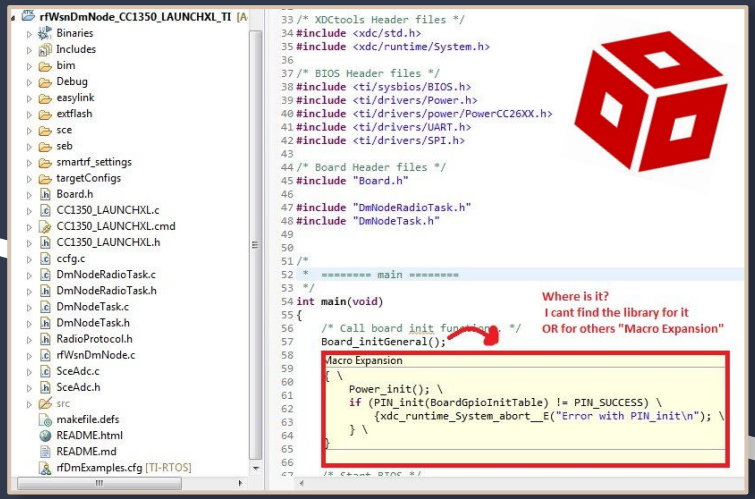
Code Critiquer System for the C Language and Embedded C

Project Overview

- Current state of project (continuation of sdmay24-34) is a web-based critiquer tool
  - Students upload C files to tool
  - Files are statically analyzed to search for antipatterns
  - Tool generates feedback
  - Students use feedback to improve skills
- Will modify current system and/or develop new prototypes
- Ideally tailored for CPR E 288

Problem Statement

- Static Code Analysis is a challenging problem
- Many off-the-shelf solutions
  - Many leave a lot to be desired
  - They are not bespoke for CPR E 288 usage
  - No off-the-shelf “perfect combination” for what the project client needs
- Client needs a code critiquer that can...
  - Be accessed by students and instructors
  - Provide beginner-oriented feedback
  - Ability to give embedded and datasheet-focused feedback
  - Potentially integrate with Code Composer Studio



Client Needs

- Client needs a code critiquer that can...
  - Be accessed by students and instructors
  - Provide beginner-oriented feedback
  - Ability to give embedded and datasheet-focused feedback
  - Potentially integrate with Code Composer Studio

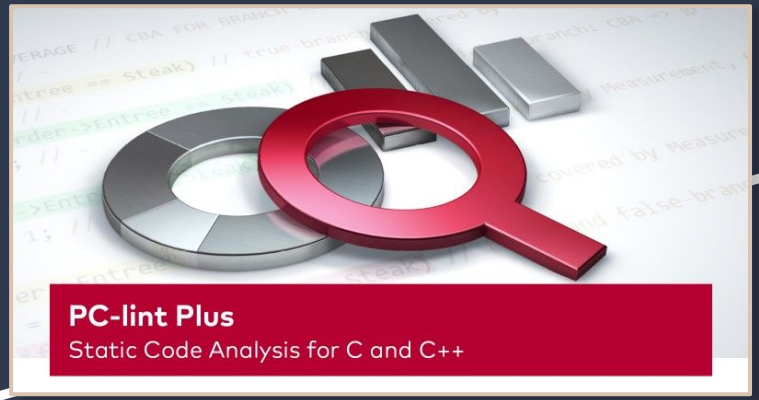
Related Products

MTU Projects by Dr. Ureel

- Michigan Tech University has multiple Code Critiquers for different programming languages.
- They built prototypes for both MatLab and Python code
    - sdmay24-34 began this project based on these
- Products are just prototypes and not widely available

Related Products

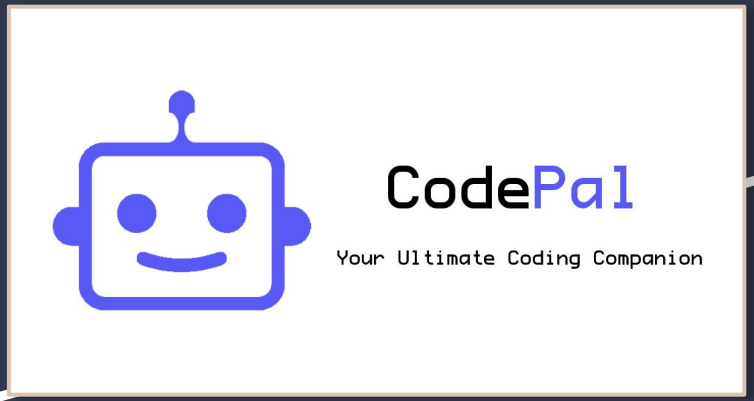
PC-lint Plus



- Uses static analysis to analyze C and C++ code
- Identifies security issues as well as bad coding practices
- Runs locally rather than easily hostable
- Costs money

Related Products

CodePal.Ai



- Uses artificial intelligence to critique the provided C code
- It gives detailed information on the syntax/structure, code readability, functionality, and the output method
- Due to its reliance on AI, it can get some things wrong
- It has a tiered pay structure and the free tier is quite limiting

Market Gap

- Tailoring suggestions to the Datasheet used in CPR E 288
- Reliability and Confidence in methodology and results (compared to AI)
- Beginner-friendly explanations
- Tailored to lessons and focus of coursework
- Lack of off-the-shelf critiquers for embedded C

Conclusions from Product Research

- Off-the-shelf products are good for general use
- Too general for usage in CPR E 288
- May provide good starting points or inspiration

New Ideas from Product Research

- Idea: Integrating the CPR E 288 datasheet into an existing solution
  - Potentially create a plugin or extension
- Idea: Compiler output explainer
  - Many compiler errors can be cryptic to beginners
- Idea: Code to English
  - Describing what code does or influences based on analysis

Any Questions, Suggestions, or Comments?