

# STREAMLINED DEVELOPMENT FOR FUTURE-READY SSDs

Datasheet

# Project objective

Implement garbage collection functionality for the client's SSD without direct access to its components. Develop a tool to simplify product development and updates by optimizing code management processes within the Register Abstraction Layer (RAL) for the SSD firmware.



# Result

The client received C-model representations for the ASIC firmware, simulating garbage collection functionality with all necessary interfaces, and incorporating SSD firmware features and interactions with other blocks. Using simulation while hardware components were under development promoted a rapid, cost-saving process with eliminated project downtime.

Also, they received an efficient tool to generate C header files for Register Abstraction Layer (RAL) for the SSD firmware. This promoted simplified code management, consistency, and reuse, which accelerated the development process and further updates.

## Scope of work

- ❖ Functional capabilities, mock-ups of screens and reports, and detailed use cases for the Garbage Collection Engine (GCE)
- ❖ Design specification, implementation, and testing of the GCE
- ❖ GCE Read and Write cursors implementation for efficient data processing and placement
- ❖ Workflow diagram, instructions, and configuration file structure for the RAL generator tool
- ❖ User-friendly templates creation to fit unique coding styles and expectations
- ❖ Implementation of the RAL generator tool

## Activities

- ❖ Requirements definition (onsite)
- ❖ Architecture design
- ❖ Software development
- ❖ Unit & system testing

# About the project

## Technologies

- ❖ C
- ❖ ANSI C
- ❖ Python
- ❖ SAS



## Project size

- ❖ 4 Software Engineers

## Duration



## Platforms

- ❖ ASIC
- ❖ Linux