Navis

A Web-Based Platform for Firmware Project Management

Berkay Kösebay (Project Manager), Djordje Momčilović (Project Manager), Jasper Brandsma, Kabeer Jamal, Andras Berkly, Tycho Louws, Arda Dinsoy, Asil Koröğlu, Fatih Dikmen, Eduardo Borges Marinho de Oliveira, Wessel van der Wijst, Bart van Aert, Caner Murzoğlu

The Problem: Local Build Complexities in Embedded Development

Firmware projects are often hindered by the complexity of setting up the build environment, leading to frequent issues.

- Manual toolchain installation is required for each system.
- Build environments vary by operating system, causing inconsistencies.
- Firmware often compiles successfully on one machine but fails on another.
- Debugging build failures related to environment mismatches.

These inconsistencies disrupt team collaboration and reduce the overall productivity in firmware development workflows, consuming valuable development time and effort.

Our Solution: Navis – One Platform, All You Need

Navis solves these challenges by offering a fully web-based development environment for Zephyr firmware projects. Through a browser, users can manage projects, write code in the code editor, and trigger builds in secure containerized environments, all without installing any software locally. Build reproducibility is guaranteed, and developers can access their projects from any machine. Features include:

- **Project Management Dashboard:** Organize and switch between projects seamlessly.

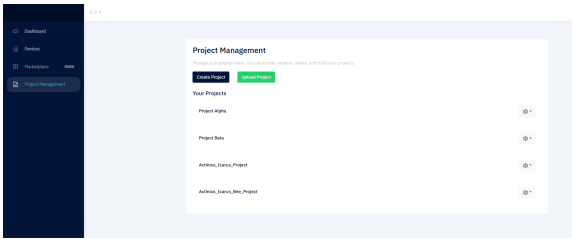

Figure 1: Project management dashboard.

- **Code editor:** Code using a browser-integrated version of VSCode with syntax highlighting and file navigation.

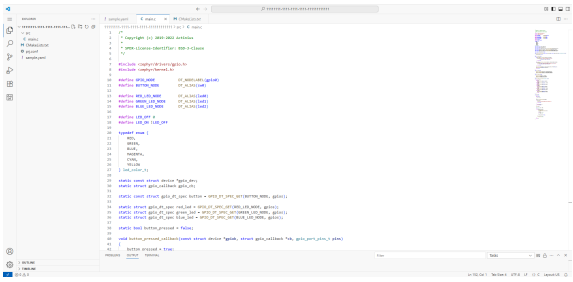

Figure 2: Code editor page.

- **One-Click Build:** Builds are executed inside ephemeral containers, ensuring consistency and security.
- **Output Access:** Logs and binaries are made available for inspection and download after each build.

Navis abstracts away system-level complexities so developers can focus on writing code, not setting up environments.

Software tools and architecture

The development and operation of Navis rely on a few software tools, each playing a role in different parts of the Navis architecture.

- **React**: Allow users to manage their projects in the dashboard through a responsive interface.
- **Node.js**: Handles the communication between the backend and the database.
- **WebDAV**: Enables a file-storage server.
- **Visual Studio Code**: The code editor is developed using Code-OSS, the open-source foundation of Visual Studio Code, including features such as file tree navigation and syntax highlighting.
- **nginx**: Used to proxy the VS Code Web development server.
- **Docker, nRF Connect SDK and Kubernetes**: While firmware builds are executed within isolated Docker containers using nRF Connect SDK, Kubernetes is responsible for managing these containerized builds.
- **PostgreSQL**: Manages and stores user data associated with Navis.

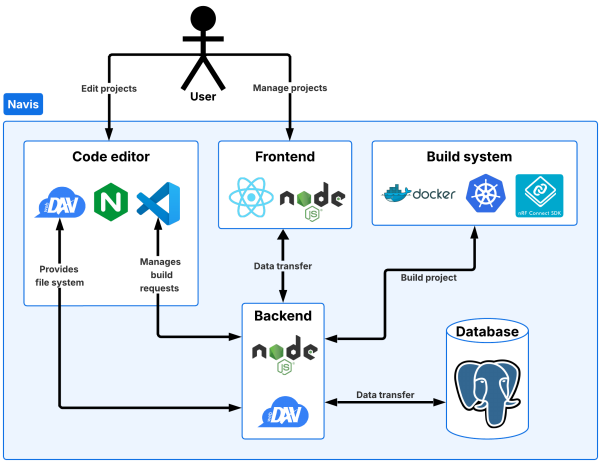

Figure 3: Architecture diagram.

What's next

While Navis provides a solid foundation for firmware development, several features can be included to enhance its capabilities, such as introducing user roles to manage access levels for project members or adding support for encrypted secrets, allowing sensitive information to be stored safely and to be securely referenced by other files.
These additions would not only improve the platform's security and team collaboration but also adapt Navis to support large-scale firmware projects.

Department of Mathematics and Computer Sciences