

System Architecture for a Metro Platform Door Control Unit

Our client, a niche embedded systems company, developed the electronics and control logic behind **automatic platform doors** used in metro stations. These doors are synchronized to work in sync with train arrivals to ensure a controlled, reliable opening and closing of the doors.

The Challenge

While the engineering team had developed detailed system designs and component specifications, the existing documentation was fragmented, highly technical, and jargon-heavy.

- **Engineering diagrams** were distributed in various documents, and it was difficult to understand which one came before which.
- **SME inputs** had unexplained acronyms and domain-specific terms, making knowledge transfer to non-core teams difficult.
- There was **no single source of truth** outlining how the system behaved under normal and error conditions or how maintenance should be approached.
- The testing team was **outsourced**. The communication between our client and the test team was partial, and both the teams didn't have the complete information about the product - from installation to maintenance.

Our task was to consolidate and translate this complexity into a **cohesive, visual, and accessible System Architecture Document** that could be used by engineering teams, maintenance crews, and future developers.

Our Approach

Our approach was to build a document from start to finish which would be a single source of all information.

1. Start with a Visual Glossary: We developed a **pictorial glossary** of all hardware components and terms used by SMEs like the LED panel, connectors, cables with different configurations. This helped as a visual checklist of the components used in the installation too. Where relevant, we included labels on photos and diagrams to indicate where each component fit in the system.

2. Macro to Micro: First, we explained the **operational context** — the metro station environment, the position of the embedded system, and the door control triggers. Then we **zoomed in** on the specific electronic module developed by the client, covering hardware layout, interface dependencies, and internal control logic.

3. Document the System Behavior. The crux of the document involved capturing how the system **behaved in different states** - the normal open and closing of the door, and the error states and their meaning, how to do the manual intervention - all complete with state diagrams.

4. Capture Maintenance & Field Diagnostics: To support field engineers, we outlined the **preventive maintenance schedule**, listing which parts needed inspection or replacement at what intervals. For each hardware component, we added details like model numbers, access locations, and diagnostic procedures.

The Impact

- **Improved handover** to partner teams working on testing, integration, and installation.
- **Faster onboarding** of new engineers who could understand the system without multiple SME calls.
- **Reduced dependency** on tribal knowledge and verbal explanations from the core team.
- **Better field readiness:** Maintenance teams had one comprehensive guide instead of relying on scattered notes and PDF datasheets.

Key Takeaways

For complex embedded systems, **contextualization is key**. Once the context or the bigger picture of where the system was deployed or intended to be deployed, it was easier to explain the system and its behaviour. A **visual glossary** is essential when engineering language becomes too dense when different teams are involved. The systems document also included the maintenance path and was a complete reference manual for all audiences