Internship Program – Software & Engineering Tracks

1. Program Overview

Our internship program provides university students with hands-on experience in real industrial product development. Interns will work directly with senior engineers, participate in agile workflows, and contribute to real features, tools, and research activities.

The program offers four specialized tracks:

- Embedded Software Engineering
- Automation Test Engineering

Each track offers in-depth training, mentorship, and project-based learning tailored to the required technical skillsets.

2. Benefits for Students

- Real-world experience working with industrial hardware, scanners, embedded systems, Android frameworks, and enterprise software.
- Mentorship from experienced engineers.
- Hands-on project assignments enabling interns to contribute to production-level software.
- Soft skills development: agile methodologies, documentation practices, teamwork, code reviews.
- Opportunity for full-time employment for high-performing interns.
- Exposure to international engineering processes.
- Allowance: 6,000,000 VND/Month

3. General Requirements for Students

- Currently pursuing a degree in Computer Science, Computer Engineering, Electronics
- Engineering, Mechatronics, or related fields.
- Basic understanding of programming concepts and data structures.
- Familiar with software versioning system as Git
- Willingness to learn new technologies.
- Good communication skills.
- Passion for building real products and solving technical problems.

4. Technical Topics

4.1. EMBEDDED SOFTWARE ENGINEERING INTERNSHIP

4.1.1 Prefered Background

- Proficiency in C/C++.

Datalogic INTERNAL

- Understanding microcontrollers and hardware fundamentals - UART, SPI, I2C.
- RTOS and Embedded Linux fundamentals.

4.1.2 Topics

4.1.2.1 Software Development Process Improvement

Goal

- Improve the overall software development process by enhancing and harmonizing development scripts, implementing unit testing, and integrating static analysis.

Descriptions

- Aanalyzing, developing, and standardizing scripts to automate workflows and increase development efficiency.
- Designing and implementing unit tests to strengthen code quality and ensure reliable functionality across software modules.
- Integrating and optimizing static analysis tools to enforce coding standards and maintain high-quality, maintainable code.

4.1.2.2 Feasibility Study for Performance-Enhancing Scanner Features

Goal

- Evaluate the feasibility of new intelligent features—such as auto-learn, background auto-detection, and reading-surface detection—to enhance HHS scanner performance.

Descriptions

- Researching and analyzing advanced algorithm or methods that enable scanners to automatically learn optimal settings based on captured data.
- Examining techniques for detecting background or reading surfaces to improve decoding accuracy under varying environments.
- Explores technical constraints, performance impacts, and potential implementation paths for integrating these features into current HHS scanner platforms.

4.2. AUTOMATION TEST ENGINEERING INTERNSHIP

4.2.1 Prefered Background

Technical skills:

- Proven ability in test case design, defect reporting, and automation scripting.
- Solid programming skills in Python, with knowledge of OOP and design patterns.
- Familiarity with Linux/Windows operations, version control (Git), and REST APIs.
- Basic experience or strong interest in working with LLMs.

- Nice to have: RobotFramework/Pytest, Embedded systems knowledge, UI test automation (Selenium/Appium), and DevOps tools (Docker/Jenkins/Kubernetes).

Soft skills:

- Good communication to ask for support when needed.
- Time management to meet deadlines.
- Problem-solving skills to creatively and independently come up with solutions.
- Adaptability and flexibility to adjust to changing work conditions and keep up with changing industry standards.
- Ability to accept feedback and to work in team

4.2.2 Topics

4.2.2.1 Local LLM for Code Documentation & Review

- **Goals:** Build a secure, local LLM station for automated documentation generation and code review.
- **Description:** Deploy an open-source Large Language Model (LLM) entirely on local internal infrastructure to ensure data privacy. Develop an automated pipeline that scans the test automation codebase (Python, Robot Framework) to generate technical documentation, automatically write function docstrings, and assist engineers with automated code reviews. This will significantly reduce manual maintenance time and ensure documentation is always up-to-date.

4.2.2.2 Test Pipeline Integration for Jazz Triggered Execution and Automated Result

- **Goals:** Integrate the test pipeline to run Robot Framework test cases triggered from Jazz and automatically upload test results back to Jazz through REST APIs, reducing manual execution and reporting and improving result traceability.
- **Description:** Study the REST APIs supported by the internal Jazz system and integrate an automated upload step into the existing test pipeline. Enable users to trigger the pipeline from Jazz to run selected Robot Framework test cases, passing the required test case identifiers and execution parameters to the pipeline. Map the selected Jazz test cases to the corresponding automated tests and track each execution request. Read Robot Framework output.xml files, extract test execution details, and map each result to the corresponding test case in Jazz. Automatically upload execution status and relevant result details after each test run, with links or attachments to logs and reports where supported. Configure authentication and project and test case mapping, handle upload failures, and prevent duplicate uploads. Validate the integration with sample pipeline runs and provide configuration and usage documentation.

Datalogic INTERNAL