

Alireza Jafari

Senior Software Engineer — Full Stack Developer — Senior C++ Engineer

 [Github Profile](#) |  [LinkedIn Profile](#) |  [Website](#) |  alirezajafari28@gmail.com
 +61 447 876 134

Experienced Software Developer with 7+ years of expertise in **C++** development, specializing in **modern C++ standards, Qt, and QML**. Proven track record of designing and delivering complex applications across a range of industries, including embedded systems, medical devices, avionics, automotive, and more. Skilled in optimizing performance, debugging, and contributing to software architecture. Strong collaborative background working in fast-paced, multi-disciplinary teams.

Experience

September 2024 – January 2025

Senior C++ Developer at Graph Inc. (Full time)

- Worked as part of a professional **C++** team developing a Windows-based **EDR** (Endpoint Detection and Response) solution
- Developed and integrated new features to enhance the product, including a rotation logger to buffer system data during offline periods, using encryption and compression to securely store logs on disk
- Developed a parallel-threaded system to transmit buffered logs to the server using gRPC once connectivity was restored
- Optimized performance and fixed critical bugs to improve system reliability and efficiency

February 2024 - April 2025

Senior Qt/C++ Developer at Med Fanavaran Plus (Part time)

- Contributed to the development of a medical-grade **ultrasound sonography** imaging device as part of a collaborative engineering team.
- Involved in the design and development of various software functionalities
- Developed **touch panel** functionalities and **measurement**, requiring mathematical and geometric computations for accurate data processing and visualization
- Contributing to the overall software architecture and design.
- Using **Qt C++ and QML** extensively for development, with the core application running on Qt Widgets
- Installed and configured a **self-hosted GitLab server** as version control, improving team collaboration and development workflow.

Dec 2023 – May 2024

Qt/C++ Developer at Piaraz (Contractor)

- Sole **C++** developer for a Qt-based application used with avionics hardware (**ARINC 429** logger)
- Delivered a complete software solution with **Modbus/TCP** communication for control and data analysis
- Developed 50+ packet parsers to process received packets and display real-time parameter information.

Dec 2018 – Dec 2023

Software Developer at DevSpec (Full time)

- Primarily develop software using **Qt C++ and QML**, focusing on user-space applications for devices running embedded Linux
- Collaborate with a small team of specialists, each focusing on different layers including OS and hardware
- Worked on projects that required additional skills including HTML/CSS, Node.js, and Qt Widgets
- ...

Education

M.Sc in Electrical Engineering – Digital Electronics

2016 - 2018

Amirkabir University of Technology, Tehran

Thesis title: Video Stabilization and Image Enhancement in shaky video files

B.Sc in Electrical Engineering – Telecommunications

2011 - 2015

Isfahan University of Technology, Isfahan

Thesis title : GSM Packet Analysis Using Software Defined Radio (SDR)

Skills

Main Skills

- **Modern C++ (11/14/17)**
Multithreading, STL, Socket Programming, Lock-free Programming, OOP, SOLID, Patterns
- **Qt Framework**
- **QML**
- **Version Control**
Git, GitLab, GitHub, Git Flow
- **Unit Testing**
Google Test

Adaptable Skills

C++20, Linux(UserSpace), JavaScript, Matlab, HTML/CSS, SQL, QMake, REST APIs, GLSL, YAML, SQLite, gRPC, winRPC, Named Pipe, Linux D-Bus, ZMQ, QWidget, Linux Kernel Space, Linux build systems (Build-Root), Qt-Wayland, Node.js, Android, Google Test gMock, Jira, Confluence

Notable Projects

QBlueprint (2025) (open-source) [\[GitHub\]](#) [\[YouTube\]](#)

A high-performance visual programming tool built with **Qt C++ and QML**, inspired by **Unreal Engine's Blueprint system**. It enables users to create complex logic flows using drag-and-drop nodes, including mathematical, logical, and IO operations (TCP/UDP, HTTP, file read/write, chart visualizations). It also supports a powerful scripting node that allows users to execute JavaScript at runtime for advanced data processing. Most blocks are modeled after Unreal Engine counterparts, and the system includes support for a custom recursive heterogeneous array type. Additional features include blueprint save/load (JSON), multi-page workspace, theme

switching, zooming, and extensibility for future node types. This is an early-stage, fast-growing personal project developed in my spare time, and it is evolving rapidly with frequent feature additions and improvements.

Ultrasonography Imaging Software (2025)

Worked as a **Senior C++ developer** on an advanced ultrasound sonography imaging machine, using **Qt C++ and QML** for both UI and system-level development. Responsibilities included implementing measurement tools, touch panel features, and system integration. Developed over 500 measurement functions—including spline, trace, ellipse, angle, and volume tools—addressing complex **mathematical and geometrical** challenges to ensure medical-grade accuracy. Built a **custom virtual keyboard** from scratch with support for multiple pages, body labels, body markers, multilingual input, and other advanced features. Designed a **magnifier** tool to help clinicians place points with high precision. Most measurement and GUI elements were inspired by Philips ultrasound systems, resulting in a modern, user-friendly interface. Also contributed to several other components that enhanced overall system performance and usability.

HMI Launcher Desktop Environment (Industrial Tablet) (2023)

Sole developer of a comprehensive launcher application for an embedded Linux-based HMI, designed in full compliance with **XDG specifications**. Using **Qt, C++, QML**, and the **Wayland protocol**, I built a custom windowing system with robust task switching and a complete lightweight desktop environment. While some features were inspired by KDE Plasma, I studied its source code extensively to understand its design patterns; however, due to the heavy dependencies within KDE modules, direct integration was not feasible. As a result, the entire system was developed from scratch with **Qt as the only dependency**, ensuring minimal footprint and full control. Key applications—including a file browser, image viewer, text editor, and settings manager—were all built in-house. The launcher replicated the intuitive feel of an Android interface, offering features such as quick settings, global search, and a customizable home screen.

Custom HMI Designer for Embedded SoC (2023)

Designed and developed a lightweight visual design environment for a low-cost SoC-based HMI running customized embedded Linux, aimed at industrial use cases. The goal was to enable users—without programming knowledge—to create GUI screens and control logic, similar to tools like **Delta's DOPSoft**. I extended and customized the **Qt Designer** application by adding plugins for logic building and Modbus (serial) communication. Significant modifications were also made to Qt Designer's core to adapt it to the project's constraints. On the HMI device, I developed a runtime application that could read, parse, and execute the GUI and logic files generated by the customized designer, enabling dynamic and responsive interface behavior directly on the device.

Avionics ARINC 429 Logger (2022)

As the sole developer, I created a comprehensive **Qt C++/QML** application for **ARINC 429** data analysis via Modbus TCP. The software, used with commercial avionics hardware and inspired by **Ballard's CoPilot**, enabled full hardware configuration, IO management, and real-time parsing of over 50 ARINC 429 packet types. It featured interactive charting via Qt WebEngine and CanvasJS with high-performance C++ data feeds, supported packet replay and database logging, and allowed users to generate custom TX packets manually or through CSV import. Additionally, the application provided tools for defining channel mappings, equipment setup, and parser selection.

Digital Instrument Cluster (2021) (Video)

As the sole software developer for a modern digital instrument cluster, I designed and implemented a high-performance application using **Qt, QML, and C++**. Drawing inspiration from premium automotive brands like BMW and Mercedes-Benz, I developed multiple user-selectable themes to deliver a polished and intuitive experience. The system covered all core functionalities of a digital cluster, including diagnostic support via UDS, OSEK-NM networking, and manufacturer-specific protocol integrations. The final product was tailored for IKCO TARA and SAIPA SHAHIN/ARIA vehicles, interfacing with their CAN matrix data. Despite running on a single-core SoC under embedded Linux, the application consistently achieved 60 FPS with fast boot times, demonstrating its optimization and responsiveness. This project was also recognized in the prestigious Khwarizmi Youth Award competition under the Practical Innovation category.

** More details and visual examples of my projects are available on my website: <https://alirezajafari.ir>*