**Name**

**Email**

**Phone Number**

**Professional Summary**

* Architected and developed robust, low-level embedded software for a multi-platform smart device ecosystem, leveraging Embedded C++ to ensure optimal performance and resource utilization in a constrained hardware environment.
* Led the integration of the RDK (Reference Design Kit) platform, customizing core components and building a new set of modules for device management, ensuring seamless interoperability and feature parity across diverse OEM hardware.
* Engineered a scalable device provisioning system using Yocto-based custom Linux distributions, streamlining the build process and enabling automated, over-the-air (OTA) updates for a fleet of over 10 million deployed devices.
* Designed and implemented a high-throughput, low-latency communication layer using TCP/IP sockets and gRPC, facilitating secure and efficient data exchange between embedded devices and a cloud-based microservices architecture.
* Authored and maintained a suite of unit tests using Google Test and Google Mock, achieving a comprehensive code coverage metric of over 95%, which significantly reduced regression bugs and accelerated the release cycle.
* Implemented advanced networking protocols including DNS and DHCP client/server functionality, ensuring reliable and dynamic network configuration for a complex embedded system.
* Optimized core system performance by refactoring critical code paths, employing advanced multi-threading and synchronization primitives (mutexes, condition variables), and resolving complex race conditions and deadlocks.
* Integrated a robust logging and monitoring framework with Splunk and Grafana, enabling real-time diagnostics and proactive issue resolution for production systems.
* Managed the end-to-end software development lifecycle in a Scrum/Agile environment, utilizing Jira for task management and Confluence for documentation, consistently meeting project deadlines and sprint goals.
* Implemented and validated memory management strategies, including custom allocators and leak detection using Valgrind, to maintain a stable and reliable system over extended operation.
* Orchestrated code reviews and collaboration using Git and Gerrit, enforcing strict coding standards and ensuring high-quality, maintainable code contributions across the team.
* Leveraged POSIX threads and Linux system calls to develop a high-performance, event-driven subsystem for real-time data processing and device control.
* Built and maintained a cross-compilation toolchain and CI/CD pipelines using Jenkins and Bash scripting to automate builds, testing, and deployment to multiple hardware targets.
* Contributed to security hardening of the embedded Linux platform by implementing secure boot, device authentication using X.509 certificates, and encrypted communication channels.
* Authored comprehensive technical documentation, including API specifications and system design documents, to facilitate knowledge transfer and onboard new team members effectively.

### Technical Skills

* **Languages:** Embedded C++, C, Bash, Python, POSIX
* **Frameworks/Libraries:** RDK, Yocto, Bitbake, Google Test, Google Mock, Protobuf, gRPC, CMake, Mbed
* **Networking Protocols:** TCP/IP, UDP, DNS, DHCP, mDNS, IPv4, IPv6, MQTT, HTTP, TLS/SSL
* **Operating Systems:** Embedded Linux, RTOS
* **DevOps/CI-CD:** Git, Gerrit, Jenkins, GitLab CI/CD, JIRA, Confluence
* **Debugging/Profiling:** GDB, Valgrind, Wireshark, Splunk, Grafana
* **Methodologies:** Agile, Scrum, Waterfall
* **Tools/Utilities:** VIM, VS Code, Make, Docker, VirtualBox, Cross-compilers, Buildroot

**Professional Experience**

**Software Engineer, Embedded Systems**

**Client : | Duration :**

* Designed and implemented the core embedded C++ application for a new smart home hub, which served as a central communication gateway, responsible for managing interactions with various peripheral IoT devices.
* Developed a custom TCP/IP stack over a low-power wireless protocol, optimizing for minimal latency and power consumption, which was critical for extending device battery life.
* Architected and integrated a gRPC-based control plane, enabling secure and efficient remote management of the hub and connected devices from a centralized cloud platform.
* Engineered a multi-threaded scheduler using POSIX threads to handle simultaneous data streams from multiple sensors, ensuring real-time responsiveness and preventing data loss under high load.
* Implemented a robust firmware over-the-air (FOTA) update mechanism that utilized a dual-partition approach to ensure atomic updates and rollbacks, minimizing device downtime and risk of bricking.
* Developed unit tests and integration tests using Google Test and CTest, achieving a high level of code coverage and automated regression testing as part of the CI/CD pipeline.
* Configured and built custom Linux kernel images and root file systems for the embedded hardware using Yocto Project and Bitbake, tailoring the distribution to the exact hardware and software requirements.
* Authored Bash scripts to automate build processes, cross-compilation, and the packaging of firmware images for mass production.
* Utilized debugging tools such as GDB and performance analysis tools like Valgrind to diagnose complex memory leaks and threading issues in a production environment.
* Collaborated with a global team using Gerrit for code reviews and Git for version control, ensuring all code adhered to enterprise-level coding standards.
* Integrated a custom logging subsystem that interfaced with Splunk, enabling detailed, remote diagnostics and post-mortem analysis of field devices.
* Managed dependencies and build configurations using CMake, ensuring a portable and maintainable build system across different development environments.
* Designed and implemented a network discovery protocol based on mDNS and DNS-SD, allowing the smart hub to dynamically discover and connect to new IoT devices on the local network.
* Contributed to the security architecture by implementing TLS/SSL for encrypted communication and integrating a secure element for hardware-based key storage.

**Environment:** Embedded C++, C, POSIX, Linux, Yocto, Bitbake, RDK, TCP/IP, gRPC, Protobuf, Git, Gerrit, Jenkins, Splunk, Jira, Confluence, GDB, Valgrind, Google Test, CMake, Bash, X.509, Secure Boot, Scrum/Agile

**Senior Embedded Software Engineer**

**Client : | Duration :**

* Developed and optimized the core software for a set-top box platform, focusing on the RDK (Reference Design Kit) video stack, including media playback, DRM integration, and graphics rendering.
* Customized the Yocto-based Linux distribution for the set-top box hardware, adding proprietary drivers and optimizing the kernel for multimedia performance and low power standby modes.
* Implemented a robust TCP/IP networking stack to handle high-volume streaming data and a variety of network configuration protocols, including DHCP and DNS.
* Migrated a legacy build system to Yocto, which streamlined the compilation process, improved build reproducibility, and reduced build times by over 40%.
* Engineered a custom inter-process communication (IPC) layer using shared memory and message queues to facilitate communication between different software components of the RDK stack.
* Authored highly optimized C++ code for the video rendering pipeline, achieving a smooth and glitch-free user experience on constrained hardware.
* Debugged and resolved complex system-level issues related to memory corruption and multi-threading race conditions using Valgrind and custom memory profilers.
* Developed and maintained the device driver and hardware abstraction layer (HAL) for a new generation of remote controls, including implementing support for Bluetooth LE.
* Integrated and configured security features, including secure boot and device attestation, to prevent unauthorized firmware modifications and ensure platform integrity.
* Collaborated with hardware and QA teams to triage and resolve issues, using Jira for bug tracking and Git for code management.
* Designed and implemented a system for logging and telemetry data collection, which was then consumed by a centralized Splunk server for performance monitoring and diagnostic purposes.
* Contributed to the development of a testing framework using Google Test and CTest, allowing for automated regression testing of key RDK components.
* Automated build and deployment processes using Jenkins pipelines and Bash scripting, creating a robust CI/CD workflow for nightly builds.
* Provided technical leadership and mentorship to junior engineers, conducting code reviews via Gerrit and ensuring adherence to best practices.

**Environment:** Embedded C++, C, RDK, Yocto, Bitbake, TCP/IP, Linux, Git, Gerrit, Jenkins, Jira, Valgrind, GDB, Google Test, CMake, Bash, Scrum/Agile

**Principal Software Engineer**

**Client : | Duration :**

* Architected and led the development of a new embedded platform for industrial control systems, which required real-time performance and high reliability.
* Designed and implemented a custom messaging protocol on top of TCP/IP to ensure low-latency communication and data integrity in a noisy factory environment.
* Engineered a fault-tolerant system with redundant components and a custom watchdog mechanism to ensure continuous operation, critical for production line uptime.
* Utilized Yocto to create a custom, minimal Linux distribution, stripping out unnecessary services and optimizing the boot time to just a few seconds.
* Implemented advanced network diagnostics and logging, enabling remote troubleshooting and monitoring of devices deployed in the field.
* Developed a comprehensive unit testing suite using Google Test to validate the correctness of the core control logic, which was the foundation of the system's reliability.
* Integrated a system for over-the-air firmware updates, utilizing a robust and secure mechanism to deploy new features and security patches.
* Managed source code in a highly disciplined manner using Git, with all changes going through a rigorous review process via Gerrit.
* Used profiling tools like Valgrind to optimize memory usage and eliminate leaks, ensuring the system could run reliably for months without a reboot.
* Created detailed technical specifications and design documents, serving as the single source of truth for the development and QA teams.
* Contributed to the design of the hardware platform, providing critical feedback on component selection and performance requirements.
* Led the integration of a security subsystem, implementing cryptographic modules for data encryption and secure key exchange.
* Wrote complex Bash scripts to automate the entire build and deployment pipeline, from code compilation to flashing the final firmware image.
* Mentored junior engineers and conducted training sessions on embedded systems development, Linux internals, and C++ best practices.

**Environment:** C, Embedded C++, Linux, Yocto, TCP/IP, Git, Gerrit, Jenkins, Splunk, Jira, Valgrind, GDB, Google Test, Bash, Scrum/Agile

**Software Engineer, Embedded Systems**

**Client : | Duration :**

* Developed and maintained the embedded C++ application for a commercial-grade network router, responsible for managing network traffic and device configurations.
* Implemented a custom firewall and NAT traversal module using low-level Linux system calls and netfilter hooks, ensuring high performance and security.
* Engineered a multi-threaded daemon to handle concurrent network connections and device management requests, utilizing POSIX threads and advanced synchronization primitives.
* Designed and developed a configuration management system using Protobuf, enabling the router to be configured remotely via a cloud-based service.
* Contributed to the porting of the router's software stack to a new hardware platform, which involved significant changes to the BSP and device drivers.
* Utilized the Yocto Project to build a minimal and secure Linux distribution for the router, optimizing for boot time and memory footprint.
* Participated in code reviews and bug triage, using Gerrit and Git to ensure high code quality and to track all changes.
* Authored unit and integration tests using a custom framework and Google Test to validate the correctness of new features before deployment.
* Implemented a robust logging and monitoring solution that integrated with Prometheus, allowing for real-time performance tracking and alerting.
* Used debugging and profiling tools such as GDB and Valgrind to resolve complex issues related to race conditions and resource leaks.
* Created a build automation script using Bash to streamline the cross-compilation and firmware packaging process for different hardware revisions.
* Integrated a secure bootloader and cryptographic modules to protect the router from tampering and unauthorized firmware updates.
* Collaborated with the DevOps team to set up a CI/CD pipeline in Jenkins, which automated the nightly builds and regression tests.
* Supported the QA team by providing detailed technical insights and assisting in the reproduction and resolution of critical bugs.

**Environment:** Embedded C++, C, Linux, Yocto, TCP/IP, Protobuf, Git, Gerrit, Jenkins, Prometheus, Jira, Confluence, GDB, Valgrind, Google Test, Bash, Scrum/Agile

**Software Engineer**

**Client : | Duration :**

* Developed and maintained the firmware for a connected car telematics unit, focusing on communication with the car's internal CAN bus and a cellular modem.
* Implemented a custom TCP/IP-based communication protocol for reliable and secure data transfer between the telematics unit and a cloud backend.
* Engineered a multi-threaded application to handle concurrent tasks, including GPS data acquisition, sensor readings, and cellular modem management, using POSIX threads.
* \*\*Utilized Embedded C++ to write efficient and memory-conscious code, which was crucial for the resource-constrained hardware of the telematics unit.
* Ported a legacy C application to a modern C++ architecture, which improved code maintainability and introduced new features more rapidly.
* Authored a suite of unit and integration tests using Google Test to validate the correctness of the new C++ components.
* Managed version control using Git and participated in regular code reviews to ensure all code met quality standards.
* Used debugging tools like GDB and memory analysis tools like Valgrind to identify and resolve critical bugs and memory leaks.
* Integrated a security framework to protect sensitive telematics data, including implementing encryption and authentication protocols.
* Wrote and maintained Bash scripts for automating development tasks, including compilation and flashing the device.
* Collaborated with a cross-functional team to define product requirements, review technical designs, and troubleshoot issues.
* Developed a custom bootloader that handled secure boot and firmware updates, ensuring the device could be updated securely in the field.
* Implemented a power management system to optimize battery consumption when the vehicle was turned off, which was a critical product requirement.
* Contributed to the design and architecture of the next-generation telematics unit, focusing on scalability and modularity.

**Environment:** Embedded C++, C, Linux, TCP/IP, Git, GDB, Valgrind, Google Test, Bash, Scrum/Agile