

Profile Summary

- Embedded Systems Firmware Developer with over **1+ year** of experience in embedded systems, specializing in low-level driver development for SD Host Controller 3.0 interfaces. Strong foundation in C, embedded C programming and hardware-software integration, with expertise in developing firmware drivers across both Pre-Silicon and Post-Silicon stages. Proficient in low-speed communication protocols such as I2C and UART. Experienced in collaborating with hardware and product engineering teams to ensure seamless driver integration and system functionality. Committed to delivering efficient, reliable, and maintainable firmware for embedded platforms.
- Currently Working with “**AZONIK Solutions Pvt Ltd**” from June **2024** to **till date**.

Professional Experience

- Developed firmware driver for SD Host Controller (SDHC) supporting UHS-I, DMA/PIO transfers, and interrupt-based communication.
- Designed and tested low-level drivers for SPI, I2C, and UART protocols, ensuring error handling and protocol compliance.
- Conducted Pre- and Post-Silicon phases of firmware driver using automated test scripts in Embedded C and Python.
- Debugged protocol and firmware issues using oscilloscopes, logic analysers, JTAG, and Protocol analysers.
- Utilized JIRA for issue tracking, defect reporting, test case management, and collaboration with cross-functional teams.
- Participated in hardware bring-up by understanding SoC schematics and interpreting IP specifications for driver integration.
- Implemented fault detection and recovery mechanisms in firmware for robust communication under real-time constraints.
- Supported debugging of silicon bugs and IP-level failures in coordination with silicon validation and hardware teams.
- Contributed to low-power feature implementation, including UART wake-up and sleep mode control in embedded systems.
- Maintained version control using Git, and documented code and debug logs for traceability and team reviews.

Skills Summary

- **Protocols:** UART, I2C & I2S, SPI, CAN, SDHC.
- **Operating Systems:** Windows, Linux (Ubuntu).
- **Architectures:** 8-bit, 16-bit, 32-bit (Intel-89S51, ARM -cortex M4 series)
- **Debugging Tools:** DMM, Oscilloscope, Logic Analyzer, Protocol Analyzer's, JTAG.
- **Programming:** C, Embedded C, Assembly.
- **Scripting:** Python (Basic).
- **Software IDEs:** Keil, STM32cubeIDE, Notepad++, GIT, GCC, GDB debugger, VScode.

Education

- **B. Tech (CSE)/**, MIC College of Technology with [8.21/10], 2021 – 2024.
- **Diploma in ECE**, Govt. polytechnic with 83%, 2018 – 2021.
- **S.S.C**, All Saints EM High School with [9.5/10], 2017 – 2018.

Work Experience

Firmware Development SDHC3.0

Roles & Responsibilities:

- Developed and tested firmware for SDHC interface compliance, Implemented state machines for card initialization.
- Developed and tested firmware as per SD specifications, Ensured interoperability.
- Integrated DMA for high-speed transfers (lower power usage) and implemented firmware optimizations.
- Implemented and tested multi-block transfer logic in firmware.
- Wrote ISR to manage errors, collaborated with hardware/validation teams to debug transfer issues.

Key Development Features:

- Card Detection and Hot-Swap.
- Power Management.
- Error Handling & Recovery.
- Compatibility & Compliance.
- Performance Optimization.
- Multi-block Read/Write Enhancements.

Embedded Driver Development for I2C & UART Protocol

I2C & UART Roles & Responsibilities (Development Work)

- Developed firmware drivers for I2C (Master/Slave) and UART (Serial Communication) across multiple MCU platforms.
- Integrated peripherals such as sensors, EEPROMs, RTCs (I2C), and external modules (UART) into embedded systems.
- Configured communication parameters (I2C timing: Standard/Fast modes; UART: baud rate, parity, stop/data bits).
- Implemented multiple transfer methods – polling, interrupts, and DMA – for optimized throughput and efficiency.
- Design and developed error handling for NACKs, arbitration loss, bus faults (I2C), and framing/overflow/parity errors (UART).
- Verified and debugged communication using logic analysers, oscilloscopes, and serial monitors.
- Integrated drivers with RTOS and bare-metal systems, supporting asynchronous and real-time data handling.
- Ensured compliance with protocol specifications (I2C timing diagrams, UART standards) and documented APIs for team usage.

Key Development Features – I2C Protocol

- Implemented I2C Master/Slave modes with 7/10-bit addressing.
- Enabled DMA & interrupt transfers with robust error recovery.
- Supported multi-device/multi-master communication.
- Optimized Fast Mode (400 kHz) timing.
- Developed scalable, modular I2C driver.

Key Development Features – UART Protocol

- Implemented interrupt & DMA-based UART with ring buffer for non-blocking, high-speed streaming.
- Supported multiple ports, auto baud detection, and RTS/CTS flow control.
- Added console logging, low-power UART, and stable transfer up to 1 Mbps.
- Enabled Modbus, GPS NMEA, and other protocol support.