ADHIP SHUKLA

adhips@vt.edu linkedin.com/in/adhip-shukla github.com/AdhipShukla **+1 (540) 605-0062**

Education

Virginia Tech **Aug 2023 – May 2025**

Master of Science in Computer Engineering *Blacksburg, VA, US*

- Courses – Real-Time Sys., Multiprocessor Prog., Operating Sys., Computer Arch., Compiler Optimizations
- Current GPA – 4.0

The LNM Institute of Information Technology **Aug 2016 – May 2020**

Bachelor of Technology in Mechanical Engineering *Jaipur, RJ, India*

Technical Skills

Languages: C, C++, SIMULINK, MATLAB, STATEFLOW, ARM Assembly, Bash Scripting
Operating Systems & Tools: FreeRTOS, Linux, Git, Makefile, GDB
Hardware: ARM Cortex Mx, STM32, BeagleBone Black, NXP S32Kx, Bosch - Infineon tri-core TC1793ESP32, Arduino
Communication Protocols: CAN, CANopen, J1936, UDS, Ethernet, SPI, I2C, UART

Experience

DORLECO **Jan 2021 – Jun 2023**

Lead Vehicle Controls Software Engineer (Model Based and Embedded) *Pune, India*

- Lead the development of VCU function, including state machines for Startup, Shutdown, and Sleep modes, Drive-Charge-Update-Fault modes, PRND Direction Determination, Torque Arbitration, Cruise Control, AC Compressor Control, Cooling Pumps & Fan Control, Power Steering Control, HVI L & HV Isolation Management, Fault Management (including data saving, DTC trigger, & safety checks), and Vehicle Speed & Odometer Calculation. Product Link.
- EEPROM library development and NVM memory management for NXP32K144.
- Developed user-application layer CAN abstraction library for NXP32K144 in Simulink.
- CAN, CANopen, and J1939 stack development for automotive and motor drive applications.
- Implemented PID, lead/lag compensators, feed-forward maps, filters for real-time applications. Performed stability analysis using Nyquist criterion and pole placement using root-locus.
- Performed requirements generation and traceability using Polarian, Simulink Requirements Manager.
- Developed MIL, SIL and PIL harness for vehicle and motor controls application in Simulink Test. Also, performed simulated HIL by running plant models on one ECU and controls on another.
- Requirements generation, TARA analysis, and DVP generation for CAN message authentication, secure diagnostics, and software signing for SOCs on Ford's ADAS ECU DAT 3.0 platform. Developed test cases using CAPL in vector CANoe.
- FOC, trapezoidal, and DTC controls development for PMSM, BLDC, and SynRM motors.

Recent Projects

STM32 Bare Metal Driver Development **GitHub**

- Developed GPIO, SPI, I2C, UART drivers from scratch for STM32 F411RE Nucleo board using STM32CudeIDE. Implemented separate APIs for interrupt service routines.
- Developed application source files and validated the drivers by establishing communication with ESP32 for various protocols.

Thread Safe Implementation of Data Structures in C **GitHub**

- Implemented spin lock, back-off lock, and queue based MCS & CLH locks using hardware atomic instructions Test & Set. Compared the correctness and performance for up to 64 threads.
- Implemented Lock-free and Wait-free queue and stack using atomic Compare and Swap instructions. Analyzed the performance against coarse and fine-grained spins locks.
- Developed CILK like work-stealing scheduler for up to 32 threads. The scheduler uses double-ended queue for each thread to manage the jobs. Compared the implementation for fine grained locking and lock-free versions.

FreeRTOS Modules Development for ESP32 **GitHub**

- Developed multi-threaded ESP32 application that samples data from STM32 over SPI and display the processed data on Nokia 5110 LCD using native FreeRTOS kernel objects.
- Implemented Lock-Free static circular queue using ESP32 specific C&S instruction. Unlike binary Semaphores and Mutexs, the implementation helps in completely avoiding priority inversions.

SSD Flash Translation Layer Implementation

- Implemented FTL for open source FEMU SSD simulator that allows block level mapping for physical and virtual page address.