Niket Naidu

San Diego, CA | niketnaiduus@gmail.com
https://www.linkedin.com/in/niket-naidu | github.com/coder137

SKILLS

- **Programming Languages:** Rust, C++17, C, Python, Dart
- **OS & Tools:** Zephyr OS, FreeRTOS, Linux, CMake, Git, Flutter
- **Network & Wireless Protocols:** gRPC, MQTT, HTTP, TCP/IP, UDP, BLE, OpenThread, GPS
- **Hardware Interfaces:** I2C, SPI, CAN, UART
- **Processors:** ARM Cortex (M7/M4) (STM32F746ZG/STM32L4), TI (TM4C123G/CC1352), NXP LPC4078, ESP8266/12E/32 Wi-Fi Chip, Raspberry Pi, Beaglebone Black, BG96

WORK EXPERIENCE

Qualcomm Technologies
Senior Engineer - Cloud AI Hardware 10/2023 – Ongoing

- Currently working on accelerating AI inference performance on Qualcomm GPUs to enable real-time, on-device execution of complex neural networks.
- Authored high-performance rust modules for encryption, logging, and created vital Rust bindings for the Qualcomm Cloud AI software stack to enable advanced AI workloads.
- Pioneered new strategies to enhance codebase concurrency and parallelism, as well as creating novel visualizations with distributed tracing to optimize performance.

Engineer - Wireless 07/2021 – 9/2023

- Previously worked on a 4G LTE project, integrating a Qualcomm Modem with Oneweb's LEO satellite grid as part of the ML1 and FW team.

TuringSense
Firmware Engineer Intern 12/2020 – 05/2021

- Architectured performant drivers (UART, SPI, TWI) during board bring-up on the Nordic ecosystem.
- Designed 3 unique software prototypes to speed up BLE transfer speeds from 10kbps to 1.4mbps.

Blue River Technology
System Software Intern 05/2020 – 08/2020

- Implemented algorithms for a real-time robotics system part of the See and Spray Technology by leveraging C++ 17 and Python in collaboration with the team. Improved log storage and speed efficiency.
- Created a python tool to automatically sync and upload logs to the AWS cloud, improving the productivity of the team by 40 percent. Provided feature updates as per JIRA requests.

San Jose State University
Research Assistant 10/2019 – 05/2020

- Successfully developed two embedded prototypes for firefighters in high-risk environments, integrating wireless technologies like LTE, BLE, and various sensors with AWS IoT Core.
- Achieved a significant 50 USD reduction in firmware costs per device and a 20% improvement in power efficiency for the second prototype.

Spartan Hyperloop, San Jose State University
Control and Communications Lead 09/2019 – 05/2021

- Improved the state logic to control the Hyperloop pod by utilizing the STM32 NUCLEO-F746ZG Board with MBED OS and STM32 HAL.
- Researched the Ethernet driver to communicate with the Off pod laptop over UDP at 50Hz and CAN to communicate with the onboard peripheral systems.

EDUCATION

Masters of Science in Computer Engineering, 2019-2021, San Jose State University, California, USA
Bachelors of Technology in Electronics and Communication, 2013-2017, Amity University, INDIA

PROJECTS

Connected and Distributed Sensing System for Healthcare [Link] 08/2020 – 05/2021
AIM: To create a peer-to-peer mesh-based network using Google's OpenThread framework to monitor large crowds, as well as to collect and forward data to healthcare personnel for further analysis and diagnosis.

- Architectured the OpenThread CLI project for Texas Instruments CC1352R1 and CC1352P2 chips using CMake with the ARM GCC toolchain, OpenOCD debugging, and unit-testing frameworks.
- Interfaced the SGP30, SPEC CO sensors with the onboard sensors on the B-L475E-IOT01A STM32 board using the MBED OS.
- Configured the STM32 board as an I2C Slave device for the CC1352R1 board while simultaneously sending data over TCP to a local server for data storage.

Enterprise Firmware platform development [Link] 02/2020 – 01/2021
AIM: To create an enterprise embedded software stack from scratch for the STM32 devices by using the ARM GCC toolchain.

- Added custom linker script support for C and C++ and their respective standard libraries along with ARM startup, bss, and data initialization.
- Integrated critical third-party software such as CMSIS and FreeRTOS while conforming to a layer-based project structure.
- The industry-leading CMake build system is used to compile, flash, and debug the target board using the ST-Link Programmer, OpenOCD, and GDB.
- Platform agnostic interfaces and hardware bring-up for the STM32L4 and LPC4078 devices.

CANster Truck - CAN-based Autonomous Vehicle [Link] 02/2020 – 05/2020
AIM: To construct an autonomous car based on the CAN protocol which can navigate from the source destination to its selected destination using Sensors, Actuators, and GPS.

- Each controller was designed as a self-contained module that handled individual tasks i.e Sensor Bridge Node, Geological Node, Driver Node, and Motor Node.
- Message passing between each node (SJTwo Board) was done over CAN (Controller Area Network).
- Primarily developed the Geological Controller and interfaced with the GPS, Bluetooth, and Compass hardware.
- Implemented the Haversine algorithm, Waypoints algorithm, and the Driver Controller logic.

Infinity Mirror MP3 Player [Link] 10/2019 – 12/2019
AIM: To create an MP3 Player with an LED Matrix and OLED Screen for output. A Music based game was to be created and playable on the LED Matrix.

- OLED Drivers using the u8g2 library were written and tested on the in-house SJTwo Board (NXP 4078).
- APDS-9960 Gesture Sensor and Accelerometer Sensor (MMA8452Q) drivers were implemented to get user input for the interactive game.
- A mobile application was created using Flutter and Bluetooth was used to send Accelerometer and Joystick sensor data to the SJTwo Board.

Generate PUF using NAND Flash 09/2019 – 12/2019
AIM: To generate a Hardware-based PUF/Secure Key by simulating Read and Write Disturb on an SLC NAND Flash.

- NAND Flash K9F1G08U0E was a bit banged using the B-L475E-IOT01A board.
- Program Disturb was successfully recreated on the SLC Flash and the PUF was generated after running the Program Disturb for 10000 cycles.
- An IEEE-based research paper was authored and submitted for the CMPE 240 course.

Green Homes Project [Link] 02/2019 – 06/2019
AIM: To build an IoT-based Plant Watering system that can be controlled remotely as well as perform watering tasks automatically.

- Successfully built and integrated the frontend (Mobile App), backend (Parse Server), and Slave devices (NodeMCU – ESP12E) over the real-time WebSocket connection.
- Backend systems were configured with ACLs to handle multiple clients using the same device as well as to add multiple slave devices under the same client.
- Slave devices had an external RTC DS3231 to store alarm information for watering functionality.