Niket Naidu

San Diego, CA | (669) 212-1601 | niketnaiduus@gmail.com
linkedin.com/in/niket-naidu-30090a134 | github.com/coder137

SKILLS

- **Programming Languages:** C, C++17, Rust, Python
- **32-bit Microcontrollers:** NXP LPC4078, ESP8266/12E/32 WiFi Chip, ARM Cortex M4 STM32L4, ARM Cortex M7 STM32F746ZG, TI TM4C123G, CC1352R1, CC1352P2
- **Microprocessors:** Raspberry Pi, Beaglebone Black, BG96
- **Protocols:** UART, SPI, I2C, CAN, HTTP, MQTT, TCP/IP, UDP, BLE, GPS, OpenThread
- **Software:** Linux, FreeRTOS, Zephyr OS, OpenOCD, GDB, CMake, Bazel, Git, Flutter, Android
- **Tools:** Logic Analyzer, PCAN-USB, BUSMASTER, Keil µVision, STM32CubeIDE, VSCode, Eclipse

WORK EXPERIENCE

Software Engineer 07/2021 – Ongoing
Qualcomm Technologies
- Implemented various features and robust bug fixes as part of the Middle Layer (ML1) and Firmware Layer (FW) teams in the LTE and Satellite communication project.
- Collaborated with cross functional teams to gather information, while building a feature from scratch as an individual contributor for the ML1 and FW teams.
- Proactively improved the Logging system to automatically visualize system interactions in an asynchronous context, reducing developer debugging time by several hours.
- Recently Promoted to Senior Software Engineer after contributing at Qualcomm for 2 years.

Firmware Engineer Intern 12/2020 – 05/2021
TuringSense
- 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.

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

Research Assistant 10/2019 – 05/2020
San Jose State University
- Engineered 2 Embedded prototypes for Firefighters in high-risk environments utilizing wireless protocols and technologies (LTE, BLE, GPS, Heartrate, and CO Sensors).
- Reduced firmware costs by 50 USD per device and improved power consumption by 20 percent for the second prototype.
- Integrated the STM32L4 (B-L475E-IOT01A with onboard sensors), Carbon Monoxide, and Heartbeat sensor with the AWS IoT Core using Cellular Modem (BG96 and AT&T Module).
- The NRF Thingy 91 development platform was used with the Zephyr OS and inbuilt LTE module.

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

For the latest projects visit github.com/coder137

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 – Ongoing
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.