

Sidharth Mehta

sidharthmehta@outlook.com | (+1) 919-771-4390 | www.linkedin.com/in/sidharthmehta1996/

Education

North Carolina State University, Raleigh, NC

Anticipated May 2021

Master of Science

Major: Computer Engineering

CGPA: 4.0

Coursework: Real Time Embedded System, Embedded System Optimization, Embedded Linux, Compilers, Microprocessor Architecture, ASIC and FPGA design with Verilog

YMCA University, Faridabad, Haryana, India

June 2018

Bachelor of Technology

Major: Electronic and Communication Engineering

CGPA: 8.83/10

Work Experience

EMTECH Foundation, New Delhi, India

June 2017 – November 2017

Embedded Systems Intern

- Schematic and PCB design for various projects using Altium designer.
- Writing firmware for projects based on PIC and STM32 microcontroller.
- Proof-reading and quality maintenance of technical documents.
- Assisting students undergoing embedded systems training at EMTECH Foundation.

Skills

Programming: C, C++, Assembly, Python, Verilog, C#, RTOS, LLVM

Protocols: I2C, SPI, USB, UART

Electronics: Hardware prototyping, ARM Cortex M, Raspberry Pi, AD2 Logic analyzer

Tools: Linux, Git, Perf, Ghidra, Keil, OrCad PSpice, Altium Designer

Projects

- **Speed optimization of Linux based Embedded Systems** – Speedup a Spherical Geometry calculation code by 4.95x on a Cortex-A72 processor. Used perf to generate profiling info and tweaked code to take benefit of ARM ISA.
- **Memory size optimization of Embedded Systems** – Reduced the amount of memory needed for an RTOS-based application on Cortex-M based processor. Reduction of RAM and ROM by 2348 bytes and 3000 bytes, respectively.
- **Speed optimization of Embedded System** – Achieved 4.3x improvement in execution speed of a JPEG decode and display program for images fetched from µSD card. Optimizations include operating SPI at higher baud rate, use of fast GPIOs, caching of data read from SD card and tweaks in control flow to reduce number of instructions.
- **Latency analysis of GPIO on Raspberry Pi 4** – Wrote programs in Bash, sysfs with C++ and C Loadable Kernel Module to detect a change in input and generate a pulse output. Measured latency using logic analyzer to compare performance of the programs for power save and performance CPU governors.
- **Performance analysis of Linux system** – Benchmarked performance of N body gravitational simulator written in different programming languages (C, C++, Python). In addition, created a python program using matplotlib to log the temperature and CPU frequency on Raspberry Pi 4 and display it in a graphical format.
- **Motion detector** – Communicated with accelerometer over I2C to update motion data in a shared data structure and use this data to change color of RGB LED via non preemptive scheduling. Improved responsiveness by breaking tasks into Finite state machine.
- **Touchscreen control of LED brightness** – Shared access of ADC using Preemptive scheduling to get touch input from touchscreen and current measurement from buck converter in a time critical manner. Project also involved displaying current plot on a display.
- **LLVM bit-code generator** – Implemented C-- a subset of C programming language using parser generators Flex & Bison to generate code in LLVM IR.