

FNU RAUNAK

rraunak@asu.edu | 602-587-6826 | www.linkedin.com/in/rraunakk | github.com/rraunak

EDUCATION

MS in Computer Engineering, Arizona State University, Tempe (**GPA: 3.83/4.0**) **Graduating May'21**
- Relevant Courses – Embedded Operating System Internals, Real-Time Embedded Systems, Computer Architecture 2, Mobile Systems Architecture, Distributed Multiprocessor Operating Systems.

B.Tech in Electrical and Electronics Engineering, MIT, Manipal (**GPA: 8.5/10**) **May'17**
- Relevant Courses – Microcontroller System Design, Embedded Systems, Computer Architecture.

EMPLOYMENT

Embedded Researcher, MPS Lab, Center for Embedded Systems, ASU **Jan'20–Dec'20**
- Collaborated with the professor, Aviral Shrivastava on a framework for fault injection setup in **Gem5** on **X86** and **ARM64** for the analysis of single-bit **soft error** mitigation techniques in the computer hardware.
- Implemented **Swift-R** and **Nemesis** state of the art techniques by developing a pass in the **LLVM** compiler.
- Used **Gem5**, **GDB**, and **Clang Toolchain** to debug the 2 software fault tolerance techniques.

Electrical Systems Engineer, Reliance Industries Limited, Mumbai, India **Aug'17–Jul '18**
- Conceptualized and spearheaded the project on a next-generation monitoring system by making use of the **ACS712** hall effect sensor for monitoring the current to protect the equipment from **overload**.
- Achieved an increase in efficiency by **15%** by **FMEA**, successful replace, and overhauling strategies.

TECHNICAL SKILLS

- Programming Languages: **C, Embedded C, C++, JAVA, OpenCL, Python3, Bash, Assembly**
- Others: **ARM, MATLAB, I2C, SPI, GPIO, PCI, SIMD Neon, TCP/IP, UDP, UEFI, BIOS, Timers, DMA, Linux, Kernel Programming, Bare-metal, Machine Learning, Arduino, USB Protocol, and Device Drivers.**

ACADEMIC PROJECTS

PWM Operation, Latency Measurement and Device Drivers in Zephyr RTOS **Spring'21**
- Used **PCA9685** driver of **x86** based galileo board to modulate **pwm** signals to control the brightness of the LED.
- Designed thread unlocking sequence using **mutex** and **semaphores** to estimate exact **context switch** latency.
- Configured a **device driver** to enable use through app in zephyr to calculate distance from the HC-SR04 sensor.

STM32 Custom Bootloader **Fall'20**
- Implemented a bootloader, which asks for a binary through **UART** and flashes it to update the firmware.
- Built different apps by altering vector table offset and linker script to allow the bootloader to jump on the app.

Generic Netlink Socket and SPI Device Driver **Spring'20**
- Programmed **LED SPI driver (MAX7219)** to display animation based on 5 distances calculated by the **HC-SR04**.
- Utilized **Netlink socket communication** to transfer 6 display patterns, SPI chip select, echo pin, trigger pin, and a distance measurement signal to and from the kernel space to display the animation sequence.
- Coded an **interrupt handler** to manage **SPI** and **gpio** operations, configured the pattern display and measurement operations to **asynchronous** using **kernel threads** known as **kthreads**.

Device Drivers for HC-SR04 Ultrasonic Sensor **Spring'20**
- Designed a Linux **miscellaneous driver** module to enable the user-space interface for HC-SR04.
- Configured the 2 echo and trigger signals as Linux **gpio pins**, directed the echo pin to trigger **gpio interrupts**, and differences of **TSC clock** was employed to get pulse width, to measure the distance.
- Developed a **platform driver**, which permitted any **HC-SR04** devices defined as **platform devices** to be instantiated and bound with the **platform driver** and enabled the **Sysfs interface**.

Red-Black Tree, Kprobes, and Dynamic Stack Dumping in Linux Kernel **Spring'20**
- Created a **character device driver** and multithreaded program using **pthreads** for Red-Black Tree.
- Devised 1 kernel module, which employed **kprobe API** to add or remove probes in the kernel.
- Built 2 new **system calls** to insert and remove **dump stack** in the execution path of kernel programs.

AIP Cache Replacement Policy **Fall'19**
- Synthesized **AIP** cache replacement policy based on a **reference counter** for **L2 cache** in the **Gem5** additionally, analyzed and compared it with **LRU** replacement policy on 5 different benchmarks.

Microcontroller Based Protection of 3-Phase Induction Motor **Spring'17**
- Engineered a protection system using the **AT89C51** microcontroller on the **Keil Uvision** platform.