
OPEN SOURCE CONTRIBUTIONS - LINUX KERNEL PATCHSETS

- **Applied to Mainline:** initial support for "suniv" Allwinner new ARM9 SoC
- Add support for DMA and audio codec of F1C100s
- Timer & SPI support for Allwinner suniv F1C100s

EDUCATION

- **Gebze Technical University** Kocaeli, Turkey
Ph.D. in Computer Science; GPA: -/4.00 July 2020 – Present
- **Gebze Technical University** Kocaeli, Turkey
Master of Science in Computer Science; GPA: 3.85/4.00 Feb. 2018 – July 2020
 - **Courses Taken:** Deep Learning, Symbolic Computation, Robot Control Theory, Non-Linear Control Theory, Advanced Computer Architecture, Algorithm Analysis and Design, Special Topics on Algorithms
 - **Scientific Preparation (Undergrad Level):** Object Oriented Programming, Data Structures, Operating Systems, Computer Architecture, Discrete Mathematics
- **Istanbul Technical University** Istanbul, Turkey
Bachelor of Engineering in Electronics and Communication Sep. 2011 – Feb. 2016

EXPERIENCE

- **Gebze Technical University** Kocaeli, Turkey
Research Assistant Feb. 2018 - Present

Zenom Real-time Simulator: Improved Zenom simulation software to handle hardware targets. Also developed ZenomCore for resource limited environments. (**Qt, Non-Linear Control**)

Haptic Delta Robot: Designed hardware to control a delta robot and developed a software to experiment different control techniques on the device. (**MCU, Linear Control**)

Swarm Unmanned Aerial Vehicle: Designed both hardware and software of a telemetry modem and an RSSI sensor capable of measuring the strength of RF signals in the specified band. (**RFIC, Embedded**)
- **Otokar** Sakarya, Turkey
Software Engineer August 2016 - Feb. 2018

Border Surveillance and Reconnaissance Vehicle: At one of our software when we try to implement a new feature, old ones started to broke. We had tremendous time pressure. I stepped back and decided to change the whole core code. Introduced a hierarchical state machine that manages the device. Since states managed correctly, code size shrunk and started to behave correctly. Later on, I revised all other devices used in this vehicle platform to use state machines. (**Embedded, C++, Qt, Yocto**)

Programmable Socket Simulator: Because we used actual hardware it was very hard to try all code changes to make sure we didn't break anything. Wrote a simulator to understand the devices in the vehicle better and fix bugs. The Result was faster development and hunting long-lasting bugs. (**Python**)
- **CTech** Istanbul, Turkey
Software Engineer July 2015 - August 2016

UAV Modem: I met Linux with this project. Hardware was given to me to write a device driver. Surprised to find that the driver already existed, and found that with a little tweaking it just started to work. The more surprising thing was how powerful the shell was. Without even writing a line of code it was so easy to connect little things to create a powerful program that utilizes my hardware. Later I wrote a device driver for a custom FPGA block. (**Linux Kernel, Device Drivers**)

Linux Training: Took Linux training from Nazim Koc (uCanLinux.com). (**U-Boot, Linux Kernel, Busybox compilation and building RootFS**)

MAJOR PROJECTS

- **Voice Transceiver Device:** An embedded Linux system with Opus codec and Alsa. Designed the hardware, and coded necessary **ALSA SoC driver, SoC's DMA Driver, RFIC device driver** and userspace software. Build the device tree and RootFS. This was the project where I created several patchsets and sent them to the mainline Linux Kernel. (**Linux Driver Development, Cross-compiling, Porting**)