Mesih Veysi Kılınç

https://github.com/MesihK

mesihkilinc@gmail.com

+90-543-864-2148

I am an Electronic Engineer doing Masters in Computer Science. Interested in Linux Kernel and love doing embedded related works. Fascinated by programming and started to code during the first year of high school, doing it since then.

OPEN SOURCE CONTRIBUTIONS - LINUX KERNEL

- **Applied:** 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
  *Master of Science in **Computer Science**; GPA: 4.00/4.00* — *Feb. 2018 – Present*
  - **Scientific preparation**: Successfully finished 5 courses under scientific preparation to Computer Science which are: *Object Oriented Programming, Data Structures, Operating Systems, Computer Architecture, Discrete Mathematics*

- **Istanbul Technical University** — Istanbul, Turkey
  *Bachelor of Engineering in **Electronics and Communication**; GPA: 2.71/4.00* — *Sep. 2011 – Feb. 2016*

EXPERIENCE

- **Gebze Technical University** — Kocaeli, Turkey
  Research Assistant — *Feb 2018 - Present*
  - **Haptic Delta Robot**: Designed hardware to control a delta robot. Developed software to try control techniques on device.
  - **Swarm Unmanned Aerial Vehicle**: Designed hardware and developed software for a UAV telemetry modem and an RSSI Sensor which can measure the strength of RF signals in the specified band.

- **Otokar** — Sakarya, Turkey
  Software Engineer — *August 2016 - Feb 2018*
  - **Border Surveillance and Reconnaissance Vehicle**: We were developing management software for the radar. At some point, no matter how hard we tried we couldn't solve problems. 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 radar. 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**)
  - **CLI Simulator**: Because we used actual hardware it was very hard to try all code changes to make sure we didn't break anything. I decided to write a radar simulator to understand it correctly and fix bugs. This paved the way to faster development and allowed us to hunt 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. Suprised 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 (**Linux Kernel, Device Drivers**)
  - **Linux Training**: Took Linux training from Nazim Koc (uCanLinux.com). Gained knowledge on **U-Boot, Linux Kernel, Busybox compilation and building RootFS**.

MAJOR PROJECTS

- **Real-time Video Compression Device**: Coded software for an H264 video compression device. It is based on IMX6 SoC and also uses real-time patched Linux and GStreamer. Build the required device tree and userspace software that is based on GStreamer (**Gstreamer, Yocto**)
- **Voice Transceiver Device**: A device capable of recording and sending voice at ISM band. It is an embedded Linux system that utilizes Opus codec and Alsa. Designed and produced the hardware and coded necessary **ALSA SoC driver, SoC's DMA Driver, RFIC device driver** and userspace software. Build the device tree and RootFS.
- **Cansat Competition, Abilene TX**: Designed hardware and coded software according to requirements of the competition. It uses Cortex-M based MCU.
- **Turksat 3USAT Cube Satellite, ITU**: Coded software for onboard computer of the satellite.