Roman’s Storozhenko FW experience

“Smart home”. A startup

Smart lock

PIC16C84 8bit microcontroller.

Programming device for PIC

Ultraviolet eraser

When I was a student, my system programmer teacher recommended me as a summer-period volunteer to a startup creating smart home solutions. During this assignment I created multithread nano OS in PIC16C84. It was written by me completely from scratch using the assembly language due to memory limitations. There was 2 main threads with the following functions:

1) Power management of the PCB using PWM (pulse-width modulation) generation
2) DTMF signal handling for interact with the smart lock.

HW: Smart Lock, PIC16C84, Ultraviolet eraser lamp for resetting PIC’s EEPROM, Universal programming device for flashing PIC’s EEPROM

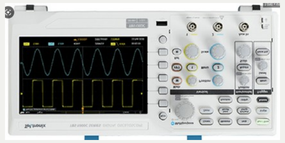
SK Hynix Memory Solutions Eastern Europe

I was a member of advanced algorithm team consisted of mathematicians and SW Engineers. We was working with the cutting-edge NAND solutions and apply the newest algorithms. My responsibilities was to exchange information with NAND devices, in particular extract it from the NAND HW in a linux driver and export it to the userspace. Also, I had to know HW and buses inside the NAND and sometimes debug its internal FW using a simulator.

Zorachka startup

Homam 64GB home security camera

LEDs



Oscilloscope

Bus analyzer

WiFi router

Homam 64GB consists of 3 PCBs each contains a microcontroller that manage this particular PCB. I have written 3 major versions for each of the 3 PCBs from scratch, that is, 9 FW images. There are a lot of FW functions in this camera: LED management, power management, PWM (HW and SW) management for LEDs, bootloader, GPIO management, FW images update management, I2C bus management…

During this development I faced a lot of debugging challenges. I used:

- Led indicator as a sign that code went to a particular branch when I was debugging issues with initial HW initialization.
- Bus analyzer when I was debugging interaction between the bootloader and user-space application providing the update for the FW image. My python updater utility was supplying an updated FW image to my bootloader using my own protocol over I2C.
- Oscilloscope when I was debugging correct software generation of pulse width modulated signal for leds.

But my experience in this project wasn't limited only by FW tasks. I participated in:

- development of the Yocto-based Homam OS Linux for ARM's CORTEX A9 Dual 600Mhz main camera's CPU. Specifically, I have written and refactored several yocto-recipes and written Linux kernel GPIO-driver from scratch in order to be able to communicate with my microcontroller's FW from one side and userspace utilities from another. It was used mostly by LED-management algorithms.
- preparing packages for SW Update embedded update system and images and some scripts for U-boot
- preparing different tests for WiFi alliance certification and installed on a router openWRT Linux distribution to handle this easily
- Integrate Apple HomeKit v3 in the userspace software

Links:

Homam 64GB on Zorachka

Homam 64GB on Amazon