Drasko DRASKOVIC

108 Rue Lecourbe, 75015 Paris, France
✉ drasko.draskovic@gmail.com ☎ +33 (0)6 73487750

OBJECTIVE

A software engineer position in a results-oriented company that seeks an ambitious and career-conscious person, where acquired skills and education will be utilized toward continued growth and advancement.

EXPERIENCE

NOKIA (ex Alcatel-Lucent) Paris, France
Senior Software Engineer
November 2014 — Present (3 years)

Work multi-standard LTE femto-cell, Linux based. Work on 5G. Work on Blockchain.

- Assured Quallcom deliveries of Linux BSP for multi-standard LTE femto-cells on projects for AT&T and T-Mobile.
- Implemented (in C++) 5G features for new NOKIA projects
- Obtained grant from internal NOKIA internal Venture Capital accelarator to bootstrap new bussiness around Blockchain technology

Devialet Paris, France
Software Architect
March 2013 — NOvember 2014 (1.5 years)

Work on system design of wireless digital audio system of next generation based on dual-core ARM Cortex-A9 FPGA with custom hard-RT Xenomai-patched Yocto Poky Linux kernel and dual-band 802.11abgn WiFi connection

- Participated in system design, both HW and SW. Electronic components selection and sourcing. SW architecture.
- Work on Linux system: wrote scripts to automatize OS creation, prototyped the system using Debian and Linaro Ubuntu on Pandaboard (vanilla kernel + debootstrap) and Wandboard (Freescale OpenEmbedded for i.MX6). Integrated Linux for Altera FPGA CycloneV board. Created and maintained platform build system based on Yocto and custom meta layer. Wrote Bitbake recipes and OPKG packages. Worked on Buildroot platform builder.

- Investigated synchronization over WiFi using NTP and PTP protocols. Implemented precise pulse-per-second GPIO kernel driver and used oscilloscope to observe and measure synchronization
- Proposed and integrated Xenomai hard-realtime kernel patches to obtain needed RT performances for audio processing and communication with FPGA cores
- Participated in creation of boot and update procedure specification. Compiled and integrated U-Boot for Altera Socfpga target and resolved problems with SPL preloader creation and FPGA sub-system load procedure

Sequans Communications Paris, France
Platform Software Engineer
June 2010 — March 2013 (2.5 years)

Work on platform enablement, system and device drivers for LTE multi-processor SoCs, based on ARM946E-S, MIPS24Kc and Lattice Mico32 architectures with eCos RTOS (for baseband CPU) and Linux (for application CPU) as a systems of choice.

- Board bring-up and HW/SW debugging (JTAG, oscilloscope, logic analyzer, ASIC debugging, HW modifications). Work on a new SoC designs from FPGA simulations and prototypes, through PCBs with first ASIC versions up to the mature chip product
- Compiled and prepared toolchains for ARMEB, MIPSEB and LM32 architectures based on Binutils, GCC and GDB. Tested and debugged solutions on all 3 architectures and integrated it into development build process.
- Fully ported uClinux to SQN3010 SoC application processor based on ARM946E-S. Fully ported Linux for MIPS, OpenWRT distribution to SQN3110 FPGA chip, and re-wrote specific configuration, Makefile, shell and Perl scripts and prepared a series of patches in a build system.
- Implemented series of Linux device drivers (GPIO, SDIO, ...) and low-level and userspace services. Created SW workarounds (driver quirks) for ASIC bugs (specification non-alignments) in third party IP cores
- Fully developed (from scratch) USIM physical layer driver and transport protocol layers of T=0 and T=1 protocols for Smart Card communication. Used Comprion and Microprocess SC sniffers/analyzers and debugged solution. Debugged ASIC bugs and created workarounds in SW. Debugged HW bugs using oscilloscope and logic analyzers. Designed (specification) and implemented low-level API and created a suite of unitary and productions tests
- Enabled OpenOCD open-source project for ARM946E-S and MIPS24Kc platforms, contributing patches to the community under GPL licence. Reverse-engineered Lattice Mico32 MonitorROM and third-party VHDL code. Adapted UrJTAG for very

low-level JTAG communication debugging. Wrote complex shell and TCL scripts for usage of cheap FTDI-based USB JTAG dongles and replacing expensive third-partner solutions, which significantly reduced development costs for the company Given a support and training to other engineers on using implemented solution

- Fully wrote (from the scratch) scatter-gather zero-copy solution for inter-processor communication based on parallel communication between eCos and uClinux network drivers, using SW circular FIFOs and HW pointers. Developed Linux application to test the implementation by receiving RF packets sent from the eNodeB, sent via eCos driver and received on Linux side, and observed traffic using Wireshark
- Implemented series of eCos RTOS drivers (watchdog, nework activity GPIO, flash handling...) and low-level services :
  - IQ Mismatch driver for calculation phase, gain and offset convergence and correction, and tested solution using tone generator and tweaking Maxim RF chip set-up. Implemented AT commands and services for controlling of the solution
  - Implemented driver for reading ISO images stored on the NOR flash and proper handling of sector handling and wear leveling. Implemented unitary test eCos application by creating ISO images on host Linux and turning USB dongles (dev boards) into ISO storage

  Modified bootloader code and NVRAM configuration in the binary form

Pace (ex Philips STB) Paris, France
Embedded Software Engineer
November 2009 — June 2010 (8 months)

Low level system programming for HD TV WiFi Zapper satelite set-top box solution based on ST7105 SoC with SH4 RISC processor, running embedded Linux as an OS of choice.

- Work on ethernet and USB drivers for U-Boot bootloader
- Bundled Linux image with intiramfs for use on USB key to simulate missing flash memory
- Rewrote U-Boot procedures and changed low-level initialization (poke table) in order to use it with proprietary bootcode with strong security constraints
- Wrote (in assembly language) bootstrap procedure for auto-decompression of system binary in specific format and implemented 29/32-bit addressing mode switch needed to boot Linux image. Implemented mechanism to pass Linux boot arguments from U-Boot to kernel via bootstrap code

- Debugged NAND Flash driver in U-Boot and resolved timing and Bad Block management issues. Integrated similar solution in Linux system using MTD Utils and JFFS2 filesystem
- Implemented MAC address detection from NOR Flash and appropriate environment setting in U-Boot and Linux
- Implemented assembly code which places Audio and Video companions (precompiled binaries) to correct shared memory locations before jumping to Linux entry point and freed necessary RAM memory by reducing root filesystem size
- Attended one week educational seminary and courses titled "Embedded Linux kernel and device drivers programing" organized by ORSYS group in Paris, France
- Wrote Perl script which merges several kernel modules into one (by renaming conflicting symbols) and implemented related mechanism via shell scripts and Makefiles, needed for correct dynamic loading of proprietary drivers
- Work on Linux kernel security hardening by re-configuring kernel, rootfs (Busybox), uClibc and user-space scripts to respect security constraints
- Implemented kernel module to read NVRAM and trigger userspace helper programs for mtd subsystem processing

SPiDCOM Technologies Paris, France
Embedded Software Engineer
December 2008 - November 2009 (1 year)

Firmware design and implementation for new SPiDCOM SPC300 HomePlug AV protocol (powerline communication) compliant SoC. Creation of BSP and Linux (with U-Boot) SW bundle for ARM based chip

- Implemented SDRAM controller configuration under U-Boot bootloader in ARM assembly language
- Designed (specification creation) a custom protocol for firmware update based on Ethernet-type Homeplug AV MME messages and implemented it in C. Wrote PC host client application (based on Linux raw sockets) to test and debug protocol implementation.
- Fully wrote MAC controller (Ethernet) driver and enabled TFTP image boot
- Enabled ARM MMU (caches and pagetable set-up, defined virtual to physical mapping, etc...) under U-Boot in order to enable DCache, which significantly speeded up Linux image transfer and all pre-boot operations
- Designed (specification creation) a custom protocol for FW update based on Ethernet-type Homeplug AV MME messages and implemented it in C. Wrote PC host client

application (based on Linux raw sockets) to test and debug protocol implementation

- Implemented (ARM assembly) code for autodetection of parameters stored on flash used on board wake-up and debuged implementation with jtag and GDB. Wrote OpenOCD jtag and GDB scripts to speed up debugging process
- Wrote (in C) Linux image boot procedure that supports dual image booting with candidate selection based on parameters stored in custom image headers. Wrote user space applications for creating these headers and modified Buildroot Makefiles to enable their pre-pending on image build
- Implemented (from scratch) library for manipulating ethernet type MME message structures and wrote unitary and functional tests in a form of multithreaded application (POSIX threads) using libcheck and TUN/TAP virtual Ethernet interface.
- Implemented (from scratch) library for system configuration and wrote unitary and functional tests

Texas Instruments Nice, France
Multimedia Software Engineer
July 2006 - December 2008 (2 year 7 months)

DSP Architecture and Applications Group

Module-level verification of In-Loop DeBlocking Filter and Motion Estimation modules for digital video coding (compliant to several modern standards: H.264, MPEG-4, etc.), part of the IVA Hardware Accelerator for OMAP4 platform

- Fully wrote C testbenches, embedding them in the reference decoder code (provided by third party vendors) and programmed filter C model; Development done on Windows (MS Visual C++ development environment) and Linux
- Ported all code to Linux and developed C interface and synchronization mechanism to enable IPC with Specman RTL simulation tool in order to test Verilog (hardware) code
- Wrote various Perl scripts and created Perl/Tk GUI application to enable automatic running of the tests, automatic logging and report creation.
- Defined and implemented (in C) functional coverage and algorithms to extract minimum set of testing conformance video bitstreams needed for full coverage

GSM/GPRS/EDGE L1 Software Engineer

Layer 1 Non-Regression testing and tool development

- Provided in depth support during GSM/GPRS/EDGE Layer1 Real-time embedded software development
- In charge of MCU and DSP L1 SW non-regression testing spread over several programs that used several TI OMAP platforms with ARM 9 / ARM 11 and TI C55x DSP
- Extensive laboratory experience - proficiency with various modern mobile telephony protocols test equipment (ANITE, RACAL, CRTU, CMU200, CRTP, etc.)
- Analyzed L1 Trace, troubleshooted L1 sofware in case of issues
- Fully developed GSM Voice audio loopback and BER test cases, implemented complete automation of GSM Voice test process for RACAL AIME 6103.
- Developed ANITE/Agilent SW applications (in C) for testing 2G/UMA (GAN) handovers

FNX Solutions Belgrade, Serbia
Software Engineer
August 2005 - July 2006 (1 year)

FNX SIERRA System (system for the management and processing of capital market transactions) development and debugging

- Worked on system coding in C under Solaris within the international team, remotely connected to servers in Philadelphia, USA
- Designed and implemented various business objects and data transfer objects used within a system for database connectivity. Implemented a crucial part of Oracle database port by adapting various libraries used for application/database communication.
- Thoroughly debugged and resolved issues for risk-management applications (system, network and GUI)
- Programmed and designed various database triggers, table and update scripts, stored procedures, etc., in SQL, Transact-SQL and PL/SQL, migrating system from Sybase to Oracle
- Fully wrote various Perl and BASH scripts and successfully implemented automatization of development process

Innovational Centre of School of Electrical Engeneering Belgrade, Serbia
Research Assistant
August 2004 - August 2005 (1 year 1 month)

LINSEC - Linux Security and Protection System, project introducing Mandatory Access Control (MAC) mechanism into Linux (as opposed to existing Discretionary Access Control mechanism)

- Linux kernel hacking (file system domain access control, socket access control)
- Ported the legacy system on 2.6.x "vanilla" kernel and resolved various errors due to GCC incompatibility issues
- Developed enhance for user-space tools for controlling the system patch

OPEN SOURCE PROJECTS

GNU and FOSS evangelist, implicated in many open source projects in domains of development, maintaining, documentation and support.

- **Mainflux** (`https://github.com/Mainflux/mainflux`)
  Mainflux IoT Platform
  - Project author and main achitect
  - Implemented reference version in Javascript (NodeJS) and Go
  - Set-up team of experts and rised community awarnes
  - Documented implementation and functionalities
  - Presented project on conferences all over the world
- **WeIO** (`http://www.we-io.net`)
  Project aims at bringing rapid prototyping and "internet of things" platform under FOSS licence for both HW and SW. Homebrew "from the scratch" implementation is employed and encouraged.
  - Project co-author and designer
  - Communication and co-operation with MicropendousX Open Hardware project on PCB design.
  - Electronic components selection and ordering and soldering of the HW development boards using PCB re-flow technique
  - Enabled OpenOCD low-level JTAG debugging. Wrote TCL configuration scripts and test for HW sanity checking.
  - Compiled and prepared toolchain based on GCC and Binutils and wrote build system as a combination of Bash shell scripts and set of Make files
  - Compiled/ported ARM CMSIS low-level libraries and drivers to NXP LPC1768

(ARM CORTEX-M3) architecture

- Ported NuttX RTOS to WeIO platform
- Ported Liux to WeIO platform and assured Linux BSP

- **OpenOCD** (`http://openocd.sourceforge.net/`)
  The Open On-Chip Debugger (OpenOCD) aims to provide debugging, in-system programming and boundary-scan testing for embedded target devices.
  - Added support for ARM946E-S architecture
  - Corrected MIPS32 code and added microcodes for cache handling
  - Documented MIPS32 implementation and functionalities

EDUCATION

Diploma Engineer (equivalent to M.Sc.)
School of Electrical Engineering, University of Belgrade
department of Electronics, Telecommunications and Control

Master Thesis in Audiotechnics : *"Analysis of Methodes And Applications for Artificial Reverberation"*, department of Telecommunications, School of Electrical Engeneering, Belgrade, Serbia

Student projects:

- RT application for railroad traffic signaling and organization written in C++ using API of a custom RT kernel
- Retriggerable timer with 7-seg display based on PIC16F84 uC written in assembly (MPlab IDE)
- 10-band audio equalizer schematic and layout implementation in Protel package
- JK flip-flop VLSI layout done in Magic VLSI tool

FOREIGN LANGUAGES

- English (excellent)
- French (good)
- Russian (understanding)
- Serbian (mother tongue)

PERSONAL INTERESTS

- Music playing (guitar) and sound and video production using GNU tools
- Contemporary and fine art photography, analogue and digital
- Karate (karateka and official member of French Karate Federation)
- Film, literature and philosophy