

# Andrei Liungrin

✉ liungrin@proton.me | 🌐 aalun.ru | 📞 (919) 462-6552

## SUMMARY

---

<b>Experience</b>	5 years
<b>English</b>	Advanced (C1)
<b>Location</b>	Open to relocation
<b>Languages</b>	C, C++, Python, Tcl, Rust, Bash
<b>Tools</b>	OpenOCD, KiCad, PulseView, Uni-T oscilloscopes
<b>Collaboration</b>	Git, Gerrit, Confluence, Azure Devops, Redmine, Jira

## EXPERIENCE

---

### Auriga LLC 1 year (July 2024 - July 2025) Systems Programmer

- A major consulting/outstaff company.
- Client is an R&D department of a multinational cybersecurity corporation.
- Worked on a completely custom secure micro-kernel operating system.
- My team was responsible for bringing it up on new mobile devices.
- Ported Android device drivers without vendor documentation by reverse-engineering source code and instrumenting traces, extending the custom compatibility layer when necessary.
- Pioneered reverse-engineering of the LTE call stack in the vendor's Android kernel and user-space.
- Analyzed undocumented Java/C++ code-base, added instrumentation and traced control flows with ADB to deliver a comprehensive porting strategy for other teams.
- Pioneered automated HIL regression tests in the client's CI pipeline, blocking more than 5 faulty patches and saving weeks of debugging.
- Implemented auto-flashing and a new way of reliable log extraction to detect driver setup errors, extending undocumented CMake build infrastructure and internal frameworks in collaboration with multiple departments.
- Developed firmware for an STM32-based USB oscilloscope (200kHz) used by QA for per-function power profiling. Gathered requirements, created a C data-acquisition library and integrated it into the pipeline to support power consumption optimizations for the performance team.

### PizzaBot LLC 4 years (April 2020 - May 2024) Embedded Programmer

- An early stage local startup developing a robotic kiosk capable of cooking a fresh pizza in four minutes without human interaction.
- Led a switch to STM32 from low-power AVR controllers to simplify development and make movements smoother and more responsive. Created build infrastructure and helped colleagues pick up the new platform.
- On-boarded two college interns, teaching embedded fundamentals, reviewing code and giving design feedback, which enabled them to contribute production-ready firmware components within three months with minimal supervision.
- Developed a time-critical IPC system to coordinate 12 controllers with a sub-millisecond precision (STM32, USART, DMA, RS-485):

- Each CNC machine in the kiosk had 11 independent modular controllers - one per motor or sensor.
- Completely IRQ-driven not to interfere with precise stepper timings.
- DMA for all USART transmissions together with Idle-line detection and the hardware CRC unit to save processor cycles.
- Graph-like API for the main controller with resource locks and wait queues which allows declarative construction of complex command flows with sequential and parallel branches to simplify common use-cases.
- Local tests with mocks for hardware layers and an automated HIL test framework to simplify regression testing. Worked with a senior college to develop a comprehensive set of test cases.
- Developed a scalable IPC system based on TCP/IP (STM32, HSEM, LwIP, Ethernet, iptables):
  - The first generation of hardware had five CNC machines with high-power STM32H7 controllers coordinated from the main computer.
  - Simple interface with callbacks on a microcontroller triggered by commands sent from a Python program.
  - Running on a second core and communicated with the main one using hardware semaphores and shared memory.
  - Configured networking on a debian computer, debugged connectivity with `tcpdump` and even analyzed signal integrity with an oscilloscope.
- Developed a stepper-motor control system (STM32, Timers, PWM, EXTI):
  - Controlled five motors in a non-blocking mode with IRQs and state-machines.
  - Smooth S-curve acceleration profile and synchronized linear movements at frequencies up to 100kHz.
  - Optimized code by analyzing assembly and instrumenting with an oscilloscope.
- Developed a control system for a Universal Robots robotic arm (Python):
  - It allowed to create smooth continuous movements from multiple short pre-programmed components.
  - Created a parser and a serializer for a proprietary script language.
  - Configured a simulation package to automatically measure and adjust movement durations and synchronize with a custom external seventh-axis.

## PROJECTS

---

- Authored educational in-depth blog articles on embedded systems, including [Writing a disassembler for ARMv6](#).
- Awarded 2nd place at the World Robot Olympiad 2019 (Innopolis) for a humanoid robot with a custom face recognition system (OpenCV, machine learning), speech synthesis (Sphinx) and motorized arms controlled with a Raspberry Pi.