Yurii S.
Senior Embedded C/C++ Engineer

SUMMARY

- Senior Embedded C/C++ Engineer with 6+ years in embedded systems, IoT, and network protocol development
- Experience with embedded Linux, Linux kernel modules, and system-level architecture across Linux and STM32
- Develops software components and firmware in C/C++ and Python, including drivers, communication stacks, and control systems
- Skilled with TCP/IP, UDP, TLS, LWM2M, and real-time data pipelines; experience with low-latency video and telemetry systems
- Debugging and profiling skills (GDB, OpenOCD, Wireshark); works with cross-compilation, Yocto, and CI automation

TECHNICAL SKILLS

Main Technical Skills	C++, C, Linux Kernel, Embedded Linux, Python
Programming Languages	AWK, C, C++, PHP, Python
JavaScript Frameworks	Node.js
JavaScript Libraries and Tools	React Router
Security	Wireshark
UI Frameworks, Libraries, and Browsers	CSS, Gstreamer, HTML
Cloud Platforms, Services & Computing	DigitalOcean
Azure Cloud Services	Azure Logic Apps
SDK / API and Integrations	API, Collections API, LWM2M, MAVLink, PyMavLink, RESTful API
Mail / Network Protocols / Data transfer	ARP, DHCP, DMA, DNS, Linux Networking, RTP, RTSP, scp, SSL, tcpdump, TCP/IP, TLS, UDP, USB, WebRTC
Scripting and Command Line Interfaces	Bash, Make, Perl
QA, Test Automation, Security	BugZilla, gtest, HIL / SITL, Integration Tests, J-Unit, TestFlight, Unit Testing, WAF, Wireshark
Hardware	CAN, CPU, ECU, GPIO, HAL, I2C, JTAG, Oscilloscope, SPI, STM, UART, WiFi
Platforms	ChibiOS, FreeRTOS, IOT, MSP, NuttX, OpenWrt, STM32, Yocto

Deployment, CI/CD & Administration	CI/CD, Jenkins, OpenOCD
Third Party Tools / IDEs / SDK / Services	CMake, GDB, SystemView, valgrind
Virtualization, Containers and Orchestration	Docker, Qemu, VmWare
Operating Systems	Embedded Linux, FreeBSD, Linux, Linux Kernel, RTOS
Logging and Monitoring	ftrace, ltrace, perf, Strace
Version Control	Git
Web/App Servers, Middleware	IPC, V4L2, ZMQ
Collaboration, Task & Issue Tracking	Jira
Methodologies, Paradigms and Patterns	REST
iOS Libraries and Tools	TestFlight
Other Technical Skills	AirSim, Gazebo, Unreal Engine

WORK EXPERIENCE

Embedded Software Engineer, NRK - Embedded Linux/STM32 video and control platform

Duration: Oct. 2025 - now

Summary: Development of an embedded Linux and STM32 based video and control platform integrating low-latency video pipelines and flight stack for advanced control and telemetry. The project involves creating end-to-end application architecture and custom firmware for sensor acquisition and peripheral control.

Responsibilities:

- Creating end-to-end application architecture across Embedded Linux companion computer and STM32 co-processor
- Designing and implementing low-latency video pipeline based on GStreamer (capture, encode, RTP transport, sink)
- Writing custom Linux kernel drivers (V4L2, I2C, SPI, GPIO) and device-tree overlays for the carrier board
- Writing custom STM32 firmware modules for sensor acquisition, peripheral control and time synchronization
- Integrating PX4 flight stack with the Linux companion over MAVLink and shared-memory IPC
- Implementing TCP/IP-based command, telemetry and video control channels with reconnection / back-pressure logic
- Writing high efficiency C/C++ multi-platform code (Linux user-space, kernel modules, STM32 bare-metal)
- Implementing new features for the firmware
- Integrating new libraries

- Maintaining API layer between Linux services and STM32 firmware
- Debugging and bug-fixing the firmware
- Debugging Linux kernel and memory allocation (kmalloc/DMA-coherent buffers, page faults, OOM analysis)
- Profiling and tuning end-to-end video latency with GStreamer tracers, perf, ftrace and custom timestamp probes
- Diagnosing TCP/IP performance issues with Wireshark, tcpdump, ss and kernel networking counters
- Tracking down race conditions and deadlocks with GDB, ThreadSanitizer, strace and ltrace
- JTAG/SWD on-target debugging of STM32 with OpenOCD and SEGGER J-Link, plus SystemView for RTOS tracing
- HW-level debugging with oscilloscope and logic analyzer (SPI/I2C/UART bring-up, signal integrity)
- Reverse engineering of vendor binaries and undocumented register behavior
- Writing unit-tests and on-target integration tests
- Automating the build process (cross-compilation, Yocto/buildroot recipes, CI pipelines)

Technologies: C, C++, Python, CMake, Make, Bash, Embedded Linux, Yocto, buildroot, Linux kernel modules, device tree, STM32 (HAL, LL, bare-metal), FreeRTOS, GStreamer, V4L2, custom kernel drivers, PX4, MAVLink, TCP/IP, UDP, RTP, RTSP, GDB, OpenOCD, JTAG/SWD, SEGGER J-Link, SystemView, Wireshark, tcpdump, strace, ltrace, perf, ftrace, Valgrind, Oscilloscope, logic analyzer, ArduPilot.

Embedded Software Engineer, ArduPilot porting and custom IMU drivers

Duration: Mar. 2025 - Sep. 2025

Summary: Porting ArduPilot firmware to custom STM32 hardware platform including board bring-up and development of custom IMU drivers integrated into the ArduPilot sensor stack. The project focuses on sensor calibration, bias estimation, and integration with sensor fusion pipelines.

Responsibilities:

- Porting ArduPilot firmware to a custom STM32-based hardware platform (board definition, HAL, ChibiOS configuration)
- Board bring-up: clocks, power sequencing, bootloader, peripheral mapping, DMA channels
- Writing custom IMU drivers (SPI/I2C) integrated into the ArduPilot AP_HAL/ AP_InertialSensor stack
- Implementing sensor calibration, temperature compensation and bias estimation routines
- Integrating drivers into EKF/sensor fusion pipeline and validating against reference IMUs
- Writing high efficiency C/C++ code for STM32/ChibiOS
- Implementing new features for the firmware
- Integrating new libraries
- Maintaining API layer between HAL, drivers and vehicle code
- Debugging and bug-fixing the firmware
- Real-time debugging with GDB over JTAG/SWD, OpenOCD and SEGGER J-Link
- RTOS-level tracing with SEGGER SystemView (task switches, ISR latency, priority inversion)
- Diagnosing IMU noise/bias with Allan variance analysis and oscilloscope captures of SPI lines
- Optimizing DMA-based sensor reads and IRQ handlers to reduce CPU load and jitter

- Analyzing flight logs (DataFlash/.bin) to correlate firmware behavior with sensor anomalies
- Reverse engineering vendor IMU register maps and reference drivers from datasheets
- Writing unit-tests and HIL/SITL tests for new drivers and ported subsystems
- Automating the build process (waf, custom board targets, CI)

Technologies: C, C++, Python, CMake, Make, Bash, waf, STM32 (HAL, LL, bare-metal), ChibiOS, ArduPilot, PX4, SPI, I2C, UART, CAN, DMA, JTAG/SWD, MAVLink, SITL, HIL, GDB, OpenOCD, SEGGER J-Link, SystemView, oscilloscope, logic analyzer, TCP/IP, UDP.

Embedded Software Engineer, Automatic steering algorithm for FPV (ArduPilot based)

Duration: Apr. 2024 - Feb. 2025

Summary: Development of an automatic steering algorithm for FPV drones based on ArduPilot firmware, including simulation environment setup and multi-platform firmware development to enhance autonomous flight capabilities.

Responsibilities:

- Creating application architecture
- Setting up simulation environment (Gazebo, AirSim, Python Client, C++ binary)
- Writing high efficiency C++ multi-platform code
- Implementing new firmware features and integrating libraries
- Maintaining API layer
- Debugging and bug-fixing firmware and Linux kernel
- Reverse engineering and writing unit-tests
- Automating build process

Technologies: C, C++, Python, CMake, Make, Bash, Linux Kernel, ZMQ, PyMavLink, ArduPilot.

Embedded Software Engineer, Automatic steering algorithm for FPV (INav based)

Duration: Feb. 2023 - Mar. 2024

Summary: Development of an automatic steering algorithm for FPV drones based on INav firmware, including simulation environment setup and multi-platform firmware development to improve autonomous navigation.

Responsibilities:

- Creating application architecture
- Setting up simulation environment (AirSim, Python Client, C++ binary)
- Writing high efficiency C++ multi-platform code
- Implementing new firmware features and integrating libraries
- Maintaining API layer
- Debugging and bug-fixing firmware and Linux kernel
- Reverse engineering and writing unit-tests
- Automating build process

Technologies: C, C++, Python, CMake, Make, Bash, Linux Kernel, ZMQ, MSP, INav.

Embedded Software Engineer, Custom communication protocol for FPV

Duration: Feb. 2022 - Mar. 2023

Summary: Development of a custom communication protocol for FPV drones, focusing on high efficiency STM controller firmware and integration of new features to enhance communication reliability and performance.

Responsibilities:

- Writing high efficiency C code for STM controller
- Implementing new firmware features and integrating libraries
- Maintaining API layer
- Debugging and bug-fixing firmware and Linux kernel
- Reverse engineering and writing unit-tests
- Automating build process

Technologies: C, C++, Python, CMake, Make, Bash, Perl, AWK, GTest.

Embedded Software Engineer, Automotive ECU based on AUTOSAR stack

Duration: Feb. 2021 - Feb. 2022

Summary: Development and maintenance of firmware for automotive ECU based on AUTOSAR stack, focusing on feature implementation, debugging, and integration of new libraries to meet automotive standards.

Responsibilities:

- Implementing new firmware features and integrating libraries
- Maintaining API layer
- Debugging and bug-fixing firmware and Linux kernel
- Reverse engineering and writing unit-tests
- Automating build process

Technologies: C, C++, Python, CMake, Make, Bash, Perl, AWK, GTest.

Embedded Software Engineer, Firmware for WiFi routers, based on OpenWRT

Duration: Jan. 2019 - Feb. 2021

Summary: Development and maintenance of firmware for WiFi routers based on OpenWRT, including feature implementation, library integration, and debugging to improve router performance and network capabilities.

Responsibilities:

- Implementing new features and integrating new libraries
- Maintaining API layer
- Debugging and bug-fixing firmware and Linux kernel
- Reverse engineering and writing unit-tests
- Automating build process

Technologies: C, C++, Python, CMake, Make, Bash, Perl, AWK, PHP, REST API, Node.js, HTML basics, CSS basics, TCP/IP, UDP, TLS, SSL, SCP, DHCP, DNS, ARP, Linux networking and network devices APIs, IPC, Wireshark, GDB, Linux, Docker, VMWare, QEMU.

Embedded Software Engineer, WebRTC communication server

Duration: Sep. 2018 - Jan. 2019

Summary: Development and maintenance of a WebRTC communication server, focusing on server-side feature implementation, communication maintenance, and debugging to ensure reliable real-time communication services.

Responsibilities:

- Implementing new features on the server side
- Maintaining and upgrading the communication
- Maintaining API layer
- Debugging and bug-fixing
- Writing unit-tests
- Automating build process

Technologies: C, C++, Python, CMake, Make, Bash, Perl, AWK, PHP, TCP/IP, UDP, TLS, Linux networking and network devices APIs, IPC, Wireshark, GDB, Linux, Docker, VMWare.

Embedded Software Engineer, IoT solution, based on LWM2M protocol

Duration: Mar. 2018 - Sep. 2018

Summary: Development of an IoT solution based on the LWM2M protocol, including implementation of FOTA, custom LWM2M objects, server support, and custom communication protocols to enable efficient device management.

Responsibilities:

- Implementing new features including FOTA and custom LWM2M objects
- Supporting custom server and communication protocols
- Maintaining API layer
- Debugging and bug fixing
- Writing unit-tests

Technologies: C, C++, Python, CMake, Make, Bash, TCP/IP, UDP, TLS, SSL, LWM2M, Linux networking and network devices APIs, Wireshark, GDB.

System Engineer, Computer Center Hardware diagnostics and repairing

Duration: 2015 - 2017

Summary: Providing hardware diagnostics and repair services at a computer center, ensuring reliable operation and maintenance of computer hardware systems.

Responsibilities:

- Diagnosing hardware issues
- Repairing computer hardware

EDUCATION

- **Polytechnic National University**
  Semiconductor Electronics. Master degree
  2011 - 2017