



CONTACT

- Støvring, Denmark
- dm@embiox.dk
- embiox.dk
- linkedin.com/in/dmandrup

CORE EXPERTISE

Embedded platforms

Embedded Linux · Yocto / BitBake · BSP and board bring-up · U-Boot / systemd · Secure boot · safe and secure A/B update

Software and real-time

C / C++ · Python / shell · RTOS / PREEMPT_RT · Timing and scheduling analysis · ftrace / perf / Perfetto · Cache-aware lock-free queues

Communication and interfaces

TCP / UDP / TLS / DTLS · MQTT / Protobuf / ZeroMQ / SNMP · Modbus · BLE / NB-IoT / LoRa · SPI / I2C / UART / RS-485 · DPDK

Delivery and lifecycle

Architecture and design · Docker / multi-stage builds · Reproducible development and CI · CMake / Meson / Make · Jenkins / CI pipelines · Software, system, and production test · Release and update engineering · SBOM / CVE / VEX

EMBEDDED LINUX · YOCTO AND BSPS · REAL-TIME SYSTEMS · MICROCONTROLLERS · COMMUNICATION PROTOCOLS · REPRODUCIBLE DELIVERY · PRODUCT LIFECYCLE

Daniel Mandrup

Embedded Software Engineer · Consultant



Founder, Embiox ApS

PROFILE

Embedded software engineer with 10+ years of professional experience developing product software across embedded Linux, Yocto, microcontrollers, real-time systems, and communication protocols.

I take technical responsibility for complete, maintainable embedded-software deliveries: architecture, hardware integration, platform and application development, reproducible builds, testing, releases, secure updates, and lifecycle support. My work spans new product development and complex existing systems where reliability, performance, traceability, and clear ownership matter.

My engineering approach is based on disciplined simplicity: clear layers, decoupled components, explicit interfaces, controlled dependencies, and decisions grounded in measurement and reproducible evidence.

EXPERIENCE

Embiox ApS

Nov 2021 - present

Founder / Embedded Software Consultant

Responsible for complete embedded-software deliveries and specialist engagements spanning architecture, platform integration, implementation, build and release infrastructure, and lifecycle support.

SELECTED EMBIOX ENGAGEMENTS

Satellite eNB - real-time Linux

Embedded Linux · PREEMPT_RT · C/C++ · performance analysis · DPDK

Through system tuning, in-depth timing analysis, and long-running practical tests, identified and validated the most effective approach for sustaining a 1 ms user-space interval timer on the an Arm Cortex-A72-based system. Across hundreds of millions of samples, interval lateness relative to the ideal target time exceeded 10 μ s only about once per 10 million samples; while extremely rare outliers remains bounded below 50 μ s.

Connected embedded Linux gateway

Raspberry Pi CM4 · Yocto · C · Modbus · LTE / Ethernet

Developed an industrial sensor gateway on Raspberry Pi CM4, acquiring data over Modbus and forwarding it via either Ethernet or LTE. The delivery also included an event-driven C application framework and Yocto meta-layers later reused as foundations for broader platform work.

Medical-device extension-antenna firmware

ATTiny1626 · embedded C · verification · production integration

Developed and integrated ATTiny1626 firmware for an extension antenna used in a medical product, including verification and production integration for a small but reliability-critical subsystem.

EDUCATION

M.Sc. Electrical Engineering
Networks and Distributed Systems
Aalborg University · 2016

B.Sc. Electronic Engineering and IT
Signal Processing
Aalborg University · 2014

LANGUAGES

Danish
Native

English
Professional working proficiency

EMBOX ENGAGEMENTS - CONTINUED

Embedded GUI platforms

Raspberry Pi · Boot2Qt · Yocto · Electron · HTML/CSS/JavaScript

Configured Boot2Qt-based Raspberry Pi distributions for Qt GUI applications, then developed an open-source Yocto-based Raspberry Pi CM4 proof of concept launching an Electron/JavaScript GUI. This removed the commercial Qt/Boot2Qt runtime-licensing dependency while enabling development with widely adopted HTML, CSS, and JavaScript technologies and tooling.

Docker build optimisation

Docker · multi-stage builds · build cache · reproducibility

Optimised Dockerfiles and build contexts through improved layer ordering, cache use, and multi-stage builds, reducing build time and image size while improving deployment efficiency and reproducibility.

ADDITIONAL CONSULTING WORK

- **Low-latency Linux IPC:** Implemented cache-aware, lock-free queues in C for low-latency, high-throughput data exchange between Linux user-space processes.
- **BLE integration:** Implemented a compact BLE integration on Raspberry Pi CM4 using Python and Bleak.
- **Real-time sensor processing:** Designed sampling, synchronization and interpolation of CO₂, O₂, and flow data for an indirect-calorimetry product and its resting-energy-expenditure calculations.
- **TCP/TLS software and diagnostics:** Diagnosed and extended asynchronous TCP/IP and TLS software implemented in C++ using Boost.Asio/Beast.
- **Platform bring-up and recovery:** Diagnosed and resolved an urgent boot failure on a Xilinx Zynq UltraScale+ platform.
- **Production-test integration:** Implemented microcontroller production-test features compatible with established test equipment and workflows.
- **RP2040 sensor HAL:** Developed a hardware abstraction layer for reliable, efficient sensor acquisition and processing on the RP2040 microcontroller.

PREVIOUS EMPLOYMENT

Sky Labs Aalborg A/S

Nov 2020 - Nov 2021

Software Engineer

Contributed to the RDK-based live-TV platform for Sky and Comcast, including low-level DVB-S tuner integration, Yocto, and a large multi-company build environment.

- Served as the Yocto specialist, maintaining and extending build tooling and resolving complex integration issues across a large codebase.
- Established Docker/Jenkins-based automated build infrastructure supporting CI/CD.

RTX A/S

Aug 2016 - Nov 2020

Embedded Software Engineer

Developed embedded and wireless products for industrial healthcare, spanning embedded Linux platforms, RTOS applications, communication protocols, and system integration.

- Contributed to a Yocto-based embedded Linux platform and secure firmware-update infrastructure.
- Established reproducible build and release procedures using Docker and archived source inputs.
- Worked with requirements, software and system test, and production-test integration.