

Optimizing RTOS - ENEA OSEck

AREA OF COMPETENCES: Embedded Systems and Microelectronic

INDUSTRY: Telco

CHALLENGE

ENEAA

For a constant improvement of its product OSEck, Enea needed a large team of skilled engineers that will manage to offload the development effort for their OSEck (OSE Compact Kernel), a DSP-optimized version of Enea's full-featured OSE RTOS. Occupying as little as 8 kbytes of memory, OSEck delivers fully-preemptive, event-driven real-time response and features built-in error detection and handling.

This combination makes OSEck ideal for telecom, datacom, automotive, industrial control, medical and mil/aero applications with tight memory constraints that require reliable real-time control and signal processing. IP Devel team is responsible of continuous enhancement of the product by adding new features, up-dating and creating new generation of the product by implementing new competences.

SOLUTION

Working closely with Enea, we jointly developed a plan to assemble a team of experienced embedded software engineers with various expertises like driver and firmware development and RTOS knowledge. The new organization group mirrored the existing ENEA R&D organization and was connected directly with existing Enea development teams. They were immediately involved in on-going and coming projects. We and Enea team used the newest prototypes and hardware that will support the new processors/hardware. Our team is still involved in the continuous improvement of OSEck RTOS.



The effort we participated included a series of tasks:

- Porting OSEck 3.2.x on different DSP architectures (Freescale Starcore Family, Texas Instruments c64x+ Family, LSI STARPRO, ARM, etc);
- Modifying the Kernel to better fit latest hardware;
- Developing new board support package;
- Up-dating pool/buffer management modules;
- Adding data and instructions cache support;
- Adding (hardware) memory management unit support;
- Up-dating test/benchmark systems;
- Driver optimization to use hardware acceleration via integrating 3rd party CSL (Chip Support Library).

The technologies that we used were:

- Hardware: Freescale Starcore Family, Texas Instruments c64x+ Family, LSI STARPRO, ARM.
- Software: Code Warrior IDE, Code Composer, Lauterbach

As main benefits we can mention significant cost savings at final cost of implementation processes; access to experienced resources and know-how; management and optimal resources allocation.