

# RTOS for IoT (Scorpius)

Multi-task Real-Time OS for small embedded devices

## Overview

Scorpius is an ITRON-like real-time operating system.

High-speed task scheduling operation is provided with small-code size, making this operating system perfect for the needs below.

- To run multi-tasking on small embedded devices such as IOT devices where code size and memory limitations are critical
- To run high-speed task switching even with a powerless CPU
- To implement a multi-task environment without high-level knowledge about OS
- To implement power saving functions that are essential for IoT devices
- To implement a security function to IoT device
- To monitor and check a status of internal operations of the system such as task transitions.
- To consider some technical support for future expansion of software and execution analysis



## Features

- Task priority based scheduler
- 0.95 KB kernel size in the smallest configuration
- Approximately 1 us<sup>(Note 1)</sup> task switching
- Supports all service functions required by applications like small embedded devices
- Easy migration to various processors
- Supports power saving and security functions in the near future
- Trace-log function to visualize internal operations of system
- Basically provided for free (royalty not required)

## Specifications

Specifications	
Kernel Code Size <sup>(Note 2)</sup>	0.95 KBytes (task functions only) to 2.5 KBytes (all functions)
Memory size for Each Task management	Approximately 80 Bytes <sup>(Note 3)</sup>
Task Dispatch Time	4.2 us <sup>(Note 4)</sup> with ARM7/40 MHz, 1 us with Cortex-M3/80 MHz
Supported Service Calls <sup>(Note 5)</sup>	Task, Semaphore, Event Flag, Mailbox, Trace Log, Energy Saving <sup>(Note 6)</sup> , Memory Pool <sup>(Note 6)</sup> , Security <sup>(Note 6)</sup>
Trace Log Function	Compatible with TOPPERS Project trace log viewer TraceLogVisualizer <sup>(Note 7)</sup>

## Supported Devices

- ARM7
- Cortex-M3/M4
- Nios II (Standard)

## Deliverables

- Source code
- User's manual
- Reference environment

## Evaluation Environment

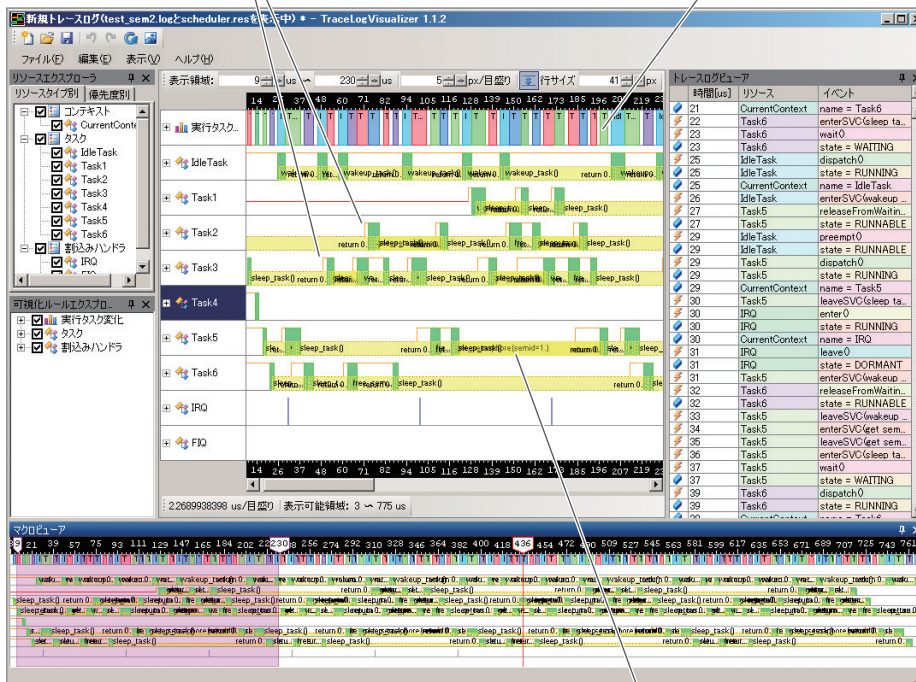
- Silicon Labs EFM32™ Pearl Gecko Starter Kit
- Silicon Labs EFM32™ Giant Gecko Starter Kit
- Analog Devices ADuCM320 Evaluation Board
- Terasic MAX® 10 Nios II Embedded Evaluation Kit

**ScorPiuS**

## Example Task Transition Display Using TraceLogVisualizer

Transition status of each task at a glance!<sup>(Note 8)</sup>

Processing time at a glance!<sup>(Notes 9, 10)</sup>



You can also check the status of each service call!

- (Note 1) Measured time required for dispatch processing (task context switching) at Cortex-M3/80 MHz operation.
- (Note 2) Compilation result for ARM7 core using Macnica environment.
- (Note 3) Memory area size required for every addition of a task.
- (Note 4) In the case of typical commercially available RTOS supporting similar CPUs, task dispatch time is around 5 us at 60 MHz
- (Note 5) Kernel can be created with minimum required functions only.
- (Note 6) Support planned in near future.
- (Note 7) Refer to <http://www.toppers.jp/en/index.html>
- (Note 8) Trace log function may change overall system processing order.
- (Note 9) Requires implementation of a timer function.
- (Note 10) Displays execution time including the trace log function code.