

smxCF™

Support for the Freescale ColdFire Processor Family

smxCF is the version of smx designed for the ColdFire processor family. It has many features to facilitate ColdFire development, which are discussed below. Features common to all versions of smx are detailed in the main smx brochure.

Processors Supported

smxCF will run on any of the ColdFire processors. We have put particular emphasis on these processor/board combinations:

- 548x Freescale M5485EVB
- 547x Freescale M5475EVB
- 5445x Freescale M54455EVB
- 5407 Freescale M5407C3
- 532x/7x Freescale M5329EVB
- 5307 Freescale M5307C3
- 528x Freescale M5282EVB and Avnet ADS-MCF5282-EVL
- 5275/74 Freescale M5275EVB
- 5272 Freescale M5272C3
- 5271/70 Freescale M5271EVB
- 5251/50/53 Freescale M5251C3
- 5249 Freescale M5249C3
- 523x Freescale M523xEVB
- 52277/4 Freescale M52277EVB
- 5216/14 Freescale M5282EVB
- 5213/12/11 Freescale M5213EVB
- 5208/7 Freescale M5208EVB
- 5206e Freescale M5206EC3

See www.smxrtos.com/processors for the latest information. Evaluation kits are available for all boards listed above, at www.smxrtos.com/eval.

Development Tools Supported

- Metrowerks CodeWarrior
- Diab Data C/C++ Compiler and visionCLICK.

Development System Requirements

- Windows 9x, 2000, ME, NT 4, or XP
- BDM interface (e.g. P&E Micro “wiggler” or Multilink; typically supplied with evaluation boards)

smxCF Development Kit Contents

- Pre-built smxCF kernel library (ELF)
- Source code platform (Protosystem) for an easy start (configured for an eval board)
- smxBSP
- SMX Quick Start, smx Target Guide, smx User’s Guide, and smx Reference Manual
- Site development license
- Royalty-free license for one developed product

smxBSP and Startup Code

The Protosystem supplied with smxCF includes BSP routines, startup code, and drivers for on-chip peripherals, such as timers and UART’s. smxFS, smxNS, smxUSB, etc. have drivers to support on-chip and external controllers. For CodeWarrior, we supply project files based on the stationery supplied and augment it with our own code for common evaluation boards. For your project, you would start with the closest BSP and adapt it, as necessary for your board. See the smxBSP brochure for more information.

Easy Change to/from Other Processors

smxCF shares the same code base with smx86, smxARM, and smxPPC. Therefore it is easy to migrate between smxCF and other processor versions of smx. If you have experience with smx on one processor then you are already down the learning curve for new projects.

Debugger Support

smxCF supports symbolic debugging for any debugger or emulator that can read the ELF/DWARF or COFF file format. In particular, CodeWarrior's integrated debugger and visionCLICK are recommended. smxAware supports these two debuggers.

smxAware is a DLL that adds smx kernel-awareness to the debugger. With it, the debugger is aware of all tasks and smx objects running in the system, and you can:

- Display information about kernel specific objects such as tasks, lsrs, semaphores,

exchanges, messages, events, heaps, stacks, etc. from an entry added to the main menu.

- View errors, profiling, and other diagnostic information.
- View a graphical window that shows event timelines, CPU usage, and stack usage.
- View a Thread window and Registers for each task. Shows where suspended tasks will resume. (CodeWarrior only)
- Display a trace log created by simple string markers output by the code.
- Set task-specific breakpoints. The breakpoint will be triggered only if it is hit while the specified task is running. (Not all debuggers support this.)

Please refer to the smxAware datasheet for further information. Also, the User's Guide is available on our web site.

Performance

max interrupt latency: 58 clocks (==0.88 microsec on 66 MHz CPU; 0.29 microsec on 200 MHz CPU)
isr to lsr time: 6.4 microsec (measured on 66 MHz 5282 running in SRAM; 2.1 microsec at 200 MHz)
task switch time: 11.4 microsec (measured on 66 MHz 5282 running in SRAM; 3.6 microsec at 200 MHz)

RAM Usage

1. **smx global variables:** 575 bytes
2. **stack space:** num stacks * stack size (typical stack size is 2000 bytes)
3. **heap space:** space for control blocks + lsr queue + heap stacks (if any) + error buffer + handle table

Notes

1. Space for control blocks depends on the number of smx objects used. Control blocks range from 12 to 76 bytes.
2. The lsr queue size is the number of lsr's that can be enqueued * 8 bytes per entry. Typically 20 to 100 elements.
3. The error buffer is optional. Its size is the number of entries * 12 bytes per entry.
4. The event buffer is optional. Its size is the number of entries * 24 bytes per entry.
5. The handle table is optional. Its size is the number of entries * 8 bytes per entry. The handle table is used only by smxAware.

