

smxARM™

Support for the ARM Processor Family

smxARM is the version of smx designed for the ARM processor family. It has many features to facilitate ARM development, which are discussed below. Features common to all versions of smx are detailed in the smx datasheet and smx Special Features brochure.

Processors Supported

smxARM will run on any ARM7, ARM9, ARM11, or Cortex-M3 processor. smxBSP and startup code are available for the following processors:

| | | |
|---------|-------------|-----------|
| STMicro | STM32F105/7 | Cortex-M3 |
| STMicro | STR71x | 7TDMI |
| STMicro | STR75x | 7TDMI |
| STMicro | STR91x | 966E-S |
| TI | TMS470 | 7TDMI |

| Vendor | Processor | Type |
|--------------|------------------|-----------|
| Atmel | AT91x408xx | 7TDMI |
| Atmel | AT91M55800 | 7TDMI |
| Atmel | AT91SAM7A1/2/3 | 7TDMI |
| Atmel | AT91SAM7S/SE | 7TDMI |
| Atmel | AT91SAM7X | 7TDMI |
| Atmel | AT91RM9200 | 920T |
| Atmel | AT91SAM9260/1/3 | 926EJ-S |
| Atmel | AT91SAM9RL64 | 926EJ-S |
| Atmel | AT91SAM9XE | 926EJ-S |
| Cirrus Logic | EP93xx | 920T |
| Freescale | i.MX1/L | 920T |
| Freescale | i.MX31/31L | 1136JF-S |
| Freescale | MAC71xx | 7TDMI-S |
| Luminary | LM3S3xxx/5xxx | Cortex-M3 |
| Luminary | LM3S8xxx | Cortex-M3 |
| Luminary | LM3S9B9x | Cortex-M3 |
| NXP | LH754xx | 7TDMI |
| NXP | LH79520/4/5 | 7TDMI |
| NXP | LH7A40x | 922T |
| NXP | LPC21/22/23/24xx | 7TDMI-S |
| NXP | LPC288x | 7TDMI-S |
| NXP | LPC29xx | 968E-S |
| NXP | LPC313x | 926EJ-S |
| NXP | LPC3180 | 926EJ-S |
| NXP | LPC32x0 | 926EJ-S |
| Samsung | S3C2410/40/43 | 920T |
| STMicro | STM32F101/2/3 | Cortex-M3 |

See www.smxrtdos.com/processors for the latest information. Evaluation kits for these are available at www.smxrtdos.com/eval.

Development Tools Supported

- IAR Embedded Workbench v5.30
- Rowley CrossWorks ARM v1.7
- CodeSourcery Sourcery G++ ARM v4.3
- GNU C/C++

Development System Requirements

- Windows 9x, 2000, ME, NT 4, or XP
- JTAG interface such as IAR J-Link/J-Trace, Lauterbach TRACE32, or Signum JTAGjet. For IAR, J-Link/J-Trace is recommended. JTAG not needed for Luminary eval boards.

smxARM Development Kit Contents

- Pre-built smxARM kernel library
- Source code platform (Protosystem) for an easy start (configured for an eval board)
- smxBSP for the selected processor
- 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 smxARM development kit includes smxBSP, 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. Also included is a project file for the EWARM or CrossWorks IDE to begin your application. If your processor is not in the table above, contact us — we are steadily adding new supported processors. For a non-supported processor, you can start with the closest smxBSP and adapt it. See the smxBSP datasheet for more information. We recommend that you purchase the development board listed on our website for your processor. Use this to get a quick start, then modify smxBSP and the startup code for your custom board.

Easy Upgrade to/from Other Processors

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

Performance

max interrupt latency: 78 clocks (== 0.4 microsec on 200 MHz ARM)
 task switch time: 8.5 microsec (measured on 200 MHz ARM922T (NXP LH7A400))

RAM Usage

1. **smx global variables:** 575 bytes
2. **stack space:** num stacks * stack size (typical stack size is 500 to 1000 bytes)
3. **heap space:** space for control blocks + lsr queue + heap stacks (if any) + error buffer + event 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, and smxDLM.

the learning curve for a new project using a different processor.

Debugger Support

smxARM supports symbolic debugging for the IAR and CrossWorks debuggers. smxARM also supports Lauterbach TRACE32 and Signum's JTAGjet + Chameleon debugger. smxAware is a DLL that adds smx kernel-awareness to the debugger. IAR C-SPY is supported, but CrossWorks cannot be. 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.
- Display a trace log created by simple string markers output by the code.

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

smxARM Function Sizes (v3.7)

| smx function | iar ARM | iar Thumb | smx function | iar ARM | iar Thumb |
|-----------------|------------|--------------|---------------------|--------------|--------------|
| bump_msg | 516 | 304 | + nheapinix | 296 | 196 |
| bump_task | 1092 | 608 | nheapsetx | 336 | 216 |
| clear_q | 1944 | 1224 | nheapwalkx | 512 | 304 |
| clear_mutex | 560 | 332 | + nmallocx | 444 | 280 |
| count | 764 | 460 | nreallocx | 460 | 296 |
| count_stop | 948 | 528 | num_in_pipe | 196 | 112 |
| create_cx | 184 | 116 | pget_char | 1080 | 612 |
| create_dpool | 952 | 588 | pget_char_stop | 1324 | 712 |
| create_eq | 152 | 96 | pput_char | 1076 | 612 |
| create_et | 116 | 70 | pput_char_stop | 1328 | 716 |
| + create_fbpool | 160 | 92 | put_msg | 304 | 172 |
| create_hmsgn | 380 | 236 | qsize | 624 | 356 |
| create_hpooln | 560 | 344 | read_timer | 252 | 144 |
| create_mutex | 324 | 196 | + receive | 1180 | 692 |
| + create_rq | 240 | 152 | receive_stop | 1128 | 644 |
| + create_sema | 368 | 244 | rel_all_blocks | 188 | 116 |
| + create_taskn | 472 | 304 | rel_all_msgs | 252 | 152 |
| + create_xchga | 520 | 368 | rel_block | 132 | 80 |
| delete_cx | 268 | 160 | + rel_fblock | 148 | 88 |
| delete_eq | 212 | 128 | rel_mutex | 904 | 540 |
| delete_et | 492 | 292 | reset_flags | 108 | 68 |
| delete_hmsg | 336 | 200 | + resume | 716 | 412 |
| delete_hpool | 448 | 260 | resume_pget | 552 | 304 |
| delete_mutex | 188 | 116 | resume_pput | 568 | 312 |
| delete_sem | 316 | 188 | + sendx | 944 | 568 |
| delete_task | 764 | 448 | set_flags | 464 | 292 |
| delete_xchg | 368 | 212 | + signalx | 944 | 540 |
| dq_msg | 228 | 132 | sleepx | 420 | 244 |
| + error manager | 1388 | 856 | sleep_stop | 412 | 240 |
| find_next | 340 | 176 | sort_fblocks | 304 | 184 |
| find_pool | 268 | 160 | + startx | 984 | 578 |
| free_mutex | 568 | 336 | start_timer | 356 | 228 |
| get_block | 260 | 168 | stop | 828 | 480 |
| + get_fblocks | 468 | 276 | stop_timer | 296 | 176 |
| get_msg | 220 | 136 | suspend | 804 | 464 |
| get_mutex | 1024 | 584 | + test | 696 | 396 |
| + go_smx | 2428 | 1632 | test_flags | 748 | 456 |
| handle table | 1136 | 720 | test_flags_stop | 740 | 460 |
| hook | 176 | 112 | test_stop | 1084 | 544 |
| + invoke | 188 | 120 | unhook | 148 | 92 |
| + keep_time | 820 | 492 | + unlockx | 180 | 116 |
| locate | 436 | 260 | + scheduler | 3032 | 1928 |
| lsrs_on | 44 | 28 | | | |
| + misc1 | 1100 | 704 | | | |
| ncallocx | 56 | 32 | + min kernel | 17716 | 11034 |
| nfreex | 236 | 144 | max kernel | 52548 | 31478 |
| nheapchkx | 28 | 22 | | | |

Notes

1. **iar** = IAR Embedded Workbench v4.30A.
2. smx v3.7.0 library built with full optimizations for speed (not size). SMX_EVB = 0; SMX_PROFILING = 0. Test Mode enabled (smx error testing).