

# smxARM™

## Support for the ARM and Cortex Processor Families

*smxARM is the version of smx designed for the ARM and Cortex processor families. 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 ARM or Cortex processor. smxBSP and startup code are available for the following processors:

Vendor	Processor	Type
Atmel	AT91x408xx	7TDMI
Atmel	AT91M55800	7TDMI
Atmel	AT91CAP9	926EJ-S
Atmel	AT91SAM3U	Cortex-M3
Atmel	AT91SAM7A1/2/3	7TDMI
Atmel	AT91SAM7S/SE	7TDMI
Atmel	AT91SAM7X	7TDMI
Atmel	AT91RM9200	920T
Atmel	AT91SAM9260/1/3	926EJ-S
Atmel	AT91SAM9G20	926EJ-S
Atmel	AT91SAM9M10/G45	926EJ-S
Atmel	AT91SAM9RL64	926EJ-S
Atmel	AT91SAM9XE	926EJ-S
Cirrus Logic	EP93xx	920T
Energy Micro	EFM32G8xx/2xx	Cortex-M3
Freescale	i.MX1/L	920T
Freescale	i.MX31/31L	1136JF-S
Freescale	Kxx	Cortex-M4
Freescale	MAC71xx	7TDMI-S
NXP	LH754xx	7TDMI
NXP	LH79520/4/5	7TDMI
NXP	LH7A40x	922T
NXP	LPC17xx	Cortex-M3
NXP	LPC21/22/23/24xx	7TDMI-S
NXP	LPC288x	7TDMI-S
NXP	LPC29xx	968E-S
NXP	LPC313x/4x/5x	926EJ-S
NXP	LPC3180	926EJ-S

NXP	LPC32x0	926EJ-S
Samsung	S3C2410/40/43	920T
STMicro	STM32F101/2/3	Cortex-M3
STMicro	STM32F105/7	Cortex-M3
STMicro	STM32F2xx	Cortex-M3
STMicro	STR71x	7TDMI
STMicro	STR75x	7TDMI
STMicro	STR91x	966E-S
TI	AM18x/17x	926EJ-S
TI	AM35x	Cortex-A8
TI	LM3S5xxx/3xxx	Cortex-M3
TI	LM3S8xxx/6xxx/2xxx	Cortex-M3
TI	LM3S9xxx	Cortex-M3
TI	TMS470	7TDMI
TI	TMS570	Cortex-R4

See [www.smxrtos.com/processors](http://www.smxrtos.com/processors) for the latest information. Evaluation kits for these are available at [www.smxrtos.com/eval](http://www.smxrtos.com/eval).

### Development Tools Supported

- IAR Embedded Workbench v6.x, v5.x
- Rowley CrossWorks ARM v2.0, v1.7
- GNU C/C++

### Development System Requirements

- JTAG interface such as IAR J-Link/J-Trace, Lauterbach TRACE32, or Signum JTAGjet. For IAR, J-Link/J-Trace is recommended. JTAG or alternative is built into some 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 UARTs. 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 supporting new ones. 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 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.

## 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))

## Code Size

	IAR ARM	IAR Thumb	IAR Thumb-2
smx library (typical)	26.3	17.2	16.0
App core files + BSP + Base	14.7	11.7	7.5
C run-time library	5.1	3.7	2.8
Total	46.1	32.6	26.3

### Notes

1. Sizes in KB.
2. Typical size is the size of smx kernel functions needed by the Protosystem and several SMX products.
3. The library can be configured smaller if error checks and other options are disabled. If memory is tight, please request an Evaluation Kit and measure sizes for your case.

## Data Size

**smx global variables:** 600 bytes

**stack space:** num stacks \* stack size (typical stack size is 500 to 1000 bytes)

**heap space:** space for control blocks + LSR queue + heap stacks + 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 LSRs 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. It is used only by smxAware.