Blackhawk™ Tailwind™

TI DSP Platform Support:

TMS320C645x, TMS320C6414T/15T/16T, TMS320C642x, TMS320C6410/12/13/18, TMS320DM646x, TMS320C644x, TMS320DM643x TMS320DM64x

Tailwind™ is an extension to the Texas Instruments (TI) DSP/BIOS™ Real-Time Kernel that provides a robust POSIX® application programming interface (API). POSIX, or Portable Operating System Interface, is a set of standards specified under the IEEE Std. 1003 that governs how to write application source code so that is it portable between operating systems (OSs).

IEEE Std. 1003.1 (also adopted as ISO Std. 9945) and is referred to, in short, as "POSIX.1". It specifies application programming interfaces (APIs) at the source level in order to enable source code portability. The following is a sampling of services and extensions offered by POSIX (please refer to the standard for a complete list).

POSIX.1, Core Services

Process Creation and Control	Timers
Signals	File and Directory Operations
Floating Point Exceptions	Pipes
Segmentation Violations	C Library (Standard C)
Illegal Instructions	I/O Port Interface and Control
Bus Errors	

POSIX.1b, Real-Time Extensions

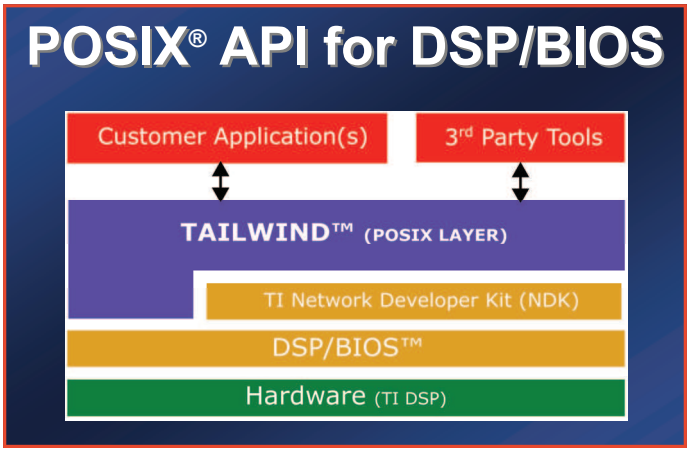
Priority Scheduling	Message Passing
Real-Time Signals	Shared Memory
Clocks and Timers	Asynch and Synch I/O
Semaphores	Memory Locking Interface

POSIX.1c, Threads Extension

Thread Creation, Control, Cleanup	Thread Synchronization
Thread Scheduling	Signal Handling

Tailwind's POSIX layer also provides the complete set of Software Communication Architecture (SCA) required POSIX interfaces. The SCA sets the rules that, if followed, will allow the growing market of wireless networks and devices to communicate. The operating environment (OE), as defined by the SCA standard, consists of a POSIX operating system, TCP/IP, Common Object Request Broker Architecture (CORBA) and SCA core framework.

Software Defined Radio (SDR) applications implement the SCA. SDR technology enables dynamically configurable and upgradable software that is being applied in the both commercial arena (public safety radios, general radios - walkie-talkies) and in military applications, such as Joint Tactical Radio Systems (JTRS). POSIX is also an increasing requirement in medical and industrial applications.



Features and Benefits

Standardized OS Interface

- Well-defined and standardized operating system services provide for the design of reliable and robust application software.
- Application compliance to POSIX ensures smooth software portability between hardware platforms.
- Standard operating system interfaces enhance programmers' productivity by encouraging them to comply with proven and efficient programming practices.
- It is the core of DoD's mandated Joint Technical Architecture (JTA) OS Services.

Featured APIs

Tailwind provides the complete set of SCA-required POSIX interfaces, along with numerous other services and capabilities, e.g. timeout support for blocking functions. These APIs include:

POSIX Threads Base API	POSIX File/Directory API
POSIX Threads Safe API	POSIX File/Device I/O API
POSIX Signals API	POSIX File Attributes API
POSIX Clock and Timers API	POSIX C Language Support API
POSIX Semaphores API	POSIX C Language Math API
POSIX Thread-Safe API	POSIX Sockets API
POSIX File System API	

Memory Overhead

The following table presents the approximate total memory consumption of Tailwind for the C64x+ platform, not including the sockets library.

Segment	**Byte Size** (hex)	**Byte Size** (decimal)
.const	0xb57	2,903
.far	0x944	2,372
.cinit	0x738	1,842
.bss	0x64	100
.switch	0x14	204
.text	0x1d3e0	119,776
Total	**0x1f02b**	**127,019**

In addition, every active thread requires a descriptor of about 600 bytes. Of course, all dynamically created objects like semaphores and files consume some additional memory.

SCA Conformance

The table below presents the relationship matrix between the SCA-mandated functionality and the services provided by Tailwind. It is evident that Tailwind offers a much richer set of APIs than required, facilitating the porting of POSIX-relying middleware such as CORBA.

AEP Unit of Functionality	SCA Mandated	Tailwind
POSIX_C_LANG_JUMP	2	2
POSIX_C_LANG_MATH	22	100+
POSIX_C_LANG_SUPPORT	54	89
POSIX_DEVICE_IO	33	40
POSIX_DEVICE_SPECIFIC	0	1
POSIX_FD_MGMT	6	12
POSIX_FILE_ATTRIBUTES	0	3
POSIX_FILE_SYSTEM	22	33
POSIX_JOB_CONTROL	0	0
POSIX_MULTI_PROCESS	***0***	7
POSIX_NETWORKING	0	25
POSIX_PIPE	0	1
POSIX_SEMAPHORES	9	10
POSIX_SIGNAL_JUMP	0	2
POSIX_SIGNALS	15	15
POSIX_SINGLE_PROCESS	0	3
POSIX_THREADS_BASE	40	74
POSIX_THREAD_SAFE_FUNC	7	8
POSIX_TIMERS	9	9

DSP/BIOS Connection

The features listed below demonstrate the relationship *Tailwind* has with DSP/BIOS, and is also shown in **Figure 1.**

Threads and Tasks

Every POSIX thread is represented by exactly one DSP/BIOS task. POSIX threads are blocked by setting their priority to minus one.

Thread Priorities

While DSP/BIOS supports only 16 task priorities, Tailwind provides 32. The priority zero, as in DSP/BIOS, is implicitly round-robin scheduled to allow for BIOS idle processing.

Thread Scheduling

Tailwind supports both FIFO and round-robin scheduling of POSIX threads.

PRD Function

Tailwind needs a periodic function to be called implemented by DSP/BIOS PRD interface. This function is used to support timeouts for blocking functions, timers and thread round-robin scheduling.

HOOK Function

Tailwind needs a task switch hook function to be called implemented by DSP/BIOS HOOK interface. This function is used to support asynchronous signal interruption and cancellation, as well as thread exception context switch for C++ with exceptions.

Memory Management

Tailwind relies on DSP/BIOS malloc()/free() for all dynamically allocated storage.

POSIX and DSP/BIOS Mixing

Although largely technically possible, mixing DSP/BIOS and POSIX API in the same application is not advisable.

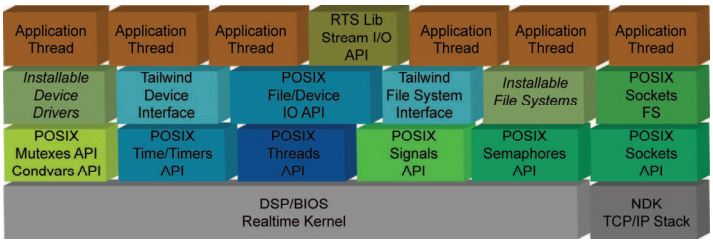


Figure 1—Tailwind Block Diagram

TI DSP Support and Requirements

- C64x big and little endian
- C64x+ big and little endian
- C6000 Compiler versions 6.0.x and 6.1.x
- Socket library for C6400 and C6400+ big and little endian (NDK 1.93/1.94)
- DSK6455 and DSK6416 Target Boards

Part Number: BH-TW-6400

Available Distributions

- Evaluation (limited runtime)
- Standard Distribution, which includes binaries, link libraries, examples
- Full Source-Code Distribution

ABOUT BLACKHAWK

EWA Technologies, Inc. is a major global player in the application of the IEEE 1149.x standards (known as boundary-scan or JTAG) to state-of-the-art products that will support the design, prototyping and production of the next generation of complex technology products. EWA Technologies consists of two operating units: the Blackhawk™ division in Mount Laurel, NJ, and Corelis, Inc. in Cerritos, CA. The Blackhawk™ division is a leader in providing hardware and software for the rapid development of digital signal processor-based (DSP) applications for a wide variety of vertical markets. Blackhawk™ is a Texas Instruments® (TI) DSP Third Party provider for development hardware, advanced JTAG emulators, Real-Time Operating Systems, design services and consulting.

123 Gaither Drive • Mt. Laurel, NJ 08054 • 877-983-4514 • www.blackhawk-dsp.com

Blackhawk Powering DSP Development™