

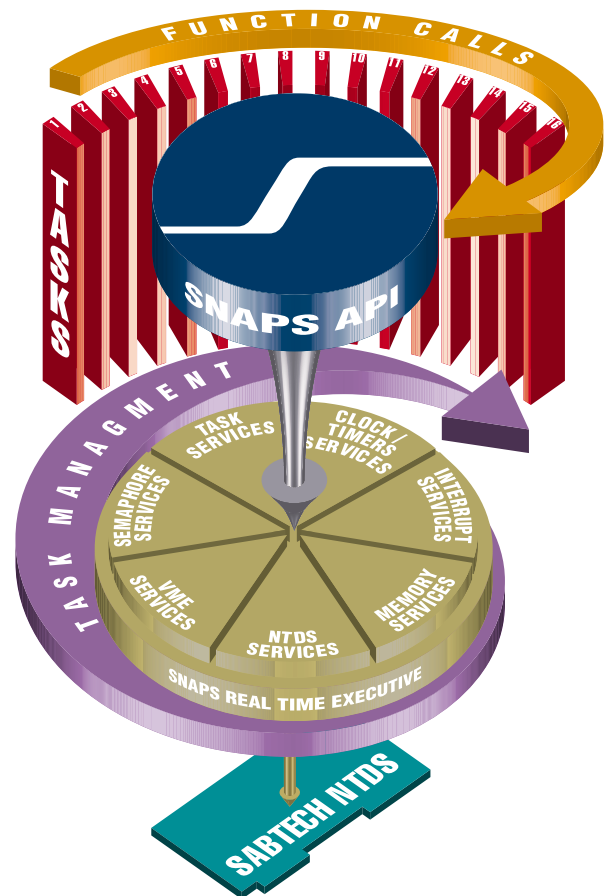
Snaps™

Description

Sabtech NTDS Application Programming Software (SNAPS™) is an application programming environment with tools to create C language programs that execute on all Sabtech VME NTDS boards. SNAPS™ multiplies the processing power of a VME system by taking advantage of the single board computer architecture of Sabtech's VME boards. Multiple boards running SNAPS™ programs become an *array of coprocessors*, greatly improving performance while reducing or even eliminating the need for additional CPU boards.

When a Sabtech VME NTDS board is loaded with a SNAPS™ program, it can run *16 concurrent tasks* in a cooperative real-time environment. SNAPS™ programs can be developed to distribute a system's workload by moving NTDS software-related functions from the Host system onto the NTDS boards. SNAPS™ applications can be used to:

- Respond faster (lower latency) on the NTDS channel with little or no host intervention
- Support NTDS messaging protocol onboard
- Perform NTDS checksums and onboard error recovery
- Filter or analyze data onboard
- Reduce VMEbus usage
- Perform virtually any task that can be executed on a general purpose microprocessor with standard C and the SNAPS™ API library



SNAPS™ Development Tools

SNAPS™ program development can be performed on a PC or a UNIX workstation. SNAPS™ is shipped on a diskette that includes the C language header files, link library, and a set of utilities to aid in the development and testing of the application. The user writes his program using standard C and SNAPS™ API functions and performs a build. The final output of the build process is a single binary file ready for upload to the Sabtech board. The utilities include a set of batch files and a program to upload the application into the Sabtech board's RAM over the VMEbus. Once loaded, the SNAPS™ application immediately begins running.

Up to 16 Sabtech boards can reside on a single VMEbus backplane with each board capable of running *16 concurrent tasks*. It's easy to see how Sabtech's VME NTDS I/O boards with SNAPS™ can be transformed into a powerful and sophisticated distributed processing system.

A powerful option is that the SNAPS™ binary file can be burned into any EPROM up to 1 MB and installed in the Sabtech board's User EPROM socket. At power up or reset, the embedded initialization code takes care of loading the program into RAM and starting the application.

Two NTDS APIs are provided by SNAPS™. Standard NTDS I/O provides support for concurrent activation of separate EI, EF, ID & OD buffers. *Low Level NTDS I/O* provides the capability to read EI & ID words into a single input buffer, or transmit EF and OD words from a single output buffer. *Low Level NTDS I/O* is especially useful for passive tap applications or where the order of transfer must be tightly controlled.

Application Debugging

All Sabtech VME NTDS boards have a built-in debugger that can be used for testing SNAPS™ programs. The debugger gives you the ability to load, store, dump, copy and compare onboard and offboard RAM, and provides access to local registers and programmable devices on the board. Execution of SNAPS™ programs can be controlled with commands such as *Go*, *Single Step* and *Breakpoint*. The C *printf* statement can also be used for debug since it "prints" to the RS-232 port.

SNAPS™ Runtime Environment

The real-time executive provides functions for controlling cooperative multitasking, clock/timer services, semaphores, and memory management. VME and NTDS operations can run from the *main()* task loop or as separate tasks controlled by the executive. The executive manages execution of up to *16 concurrent tasks* according to priority values from 1 to 255 assigned by the user. Configuration of the NTDS board is automatic at application start-up. SNAPS™ delivers everything that the application needs to perform its job in a runtime environment.

Preserving Your Investment

Sabtech understands the risks customers take in developing systems based on COTS components. Every system design requires commitment to a certain architecture and a resulting investment in hardware and software. Choices are often based upon the capabilities of certain COTS hardware and ultimately to a supplier's commitment to deliver and support that product through its life cycle in the fleet.

Sabtech alleviates this concern by designing all its VME products with compatibility in mind. SNAPS™ is no exception. Programs developed under SNAPS™ are *guaranteed* to be compatible with all current and future generations of Sabtech VME NTDS boards with nothing more than a quick recompile. SNAPS™ applications are also portable between Parallel and Serial Sabtech VME/NTDS boards with minimal changes to the configuration parameters.

As COTS usage in military systems continues to increase, SNAPS™ is the best way to build powerful, sophisticated NTDS systems while preserving your investment in hardware and software.

Features

- Real-time, interrupt driven, multitasking executive runs on the NTDS I/O board
- Supports up to 16 concurrent tasks
- Cooperative (not timed) task switching
- Default or user-assigned task priorities from 1-255
- Can generate and receive VME interrupts
- Handles all interrupt processing and task scheduling
- Automatic board configuration at application startup
- Includes C language NTDS API and Library
- Includes APIs for task control, semaphores, memory management, timers, VMEbus transfers and NTDS I/O
- Transfers memory between the Sabtech NTDS board and any VMEbus address in block or single word mode
- Supports fairness & multiple request levels on the VMEbus
- Supports concurrent activation of OD, EF, ID and EI
- Allows the same program to run on serial or parallel Sabtech VME NTDS boards
- Includes utility programs to: upload binary files, generate assembly language list files, compile and link test programs
- Requires Microtec Research C or C++ cross compiler (for Motorola 68xxx processors)

Task Control & Execution	Clock/Timer Functions	Semaphore Functions	Std. NTDS I/O Functions
taskStart taskExit exit setPriority releaseTask dispatch getCurrentTask taskStatus critEnter critExit	clkRateSet clkRateGet sysClkGet sysClkSet pauseTask delay sysUsecGet	semCreateBin semCreateCnt semDeleteBin semDeleteCnt semTakeBin semTakeCnt semGiveBin semGiveCnt semFlushBin semFlushCnt	ntdsConfigure initializeNtds startInputBuffer startOutputBuffer startEfBuffer startEiBuffer getNtdsStatus getNtdsCount resetNtdsChannel
Low Level NTDS I/O	Message Block Buffering	Memory Functions	Interrupt Processing
NtdsIn NtdsOut NtdsCancelIn ntdsCancelOut	readBlock32 readBlock16 readBlock8 writeBlock32 writeBlock16 writeBlock8	malloc memavl memmax getInputBuffer getOutputBuffer maxInputBuffer maxOutputBuffer getCardNumber memProbe	RegisterMailboxSem RegisterVmeIntrSem generateInterrupt

SNAPS™ API

Function

Calls