PUMPKIN™
REAL-TIME SOFTWARE
750 Naples Street • San Francisco, CA 94112 • (415) 584-6360 • http://www.pumpkininc.com

RM-CCS2000
Reference Manual

Salvo Compiler Reference Manual – TI's Code Composer Studio 'C2000

Salvo™
The RTOS that runs in tiny places.™

created by Andrew E. Kalman on May 3, 2003 updated on Jul 24, 2003

Introduction

This manual is intended for Salvo users who are targeting TI's (http://www.ti.com/) TMS320C2000 DSPs with TI's Code Composer (CC) 'C2000 and Code Composer Studio (CCS) 'C2000 C compiler.

Note Since the two compilers and their IDEs are essentially identical, Code Composer Studio will be used throughout this manual to refer to both products. Where necessary, differences will be identified.

Related Documents

The following Salvo documents should be used in conjunction with this manual when building Salvo applications with TI's Code Composer Studio 'C2000 C compiler:

Salvo User Manual
Application Note AN-21

Example Projects

Example Salvo projects for use with TI's Code Composer 'C2000 C compiler, the CC IDE and Spectrum Digital's (http://www.spectrumdigital.com/) eZdsp™ LF2407A development kit can be found in the:

```
\salvo\ex\ex1\sysaa
\salvo\tut\tu1\sysaa
\salvo\tut\tu2\sysaa
\salvo\tut\tu3\sysaa
\salvo\tut\tu4\sysaa
\salvo\tut\tu5\sysaa
\salvo\tut\tu6\sysaa
```

directories of every Salvo for TI's TMS320C2000 DSPs distribution. Similarly, example Salvo projects for use with TI's Code Composer Studio 'C2000 C compiler, the CCS IDE and 'C28x DSPs can be found in the:

```
\salvo\ex\ex1\sysw
\salvo\tut\tu1\sysw
\salvo\tut\tu2\sysw
\salvo\tut\tu3\sysw
\salvo\tut\tu4\sysw
```

```
\salvo\tut\tu5\sysw
\salvo\tut\tu6\sysw
```

directories of every Salvo for TI's TMS320C2000 DSPs distribution.

Features

Table 1 illustrates important features of Salvo's port to TI's Code Composer Studio 'C2000 C compiler.

general	
available distributions	Salvo Lite, LE & Pro for TI's TMS320C2000 DSPs
supported targets	all of TI's C240x and C28x DSPs
header file(s)	`porttic2000.h`
other target-specific file(s)	`porttic24x.asm`, `porttic28x.asm`
project subdirectory name(s)	`SYSW`, `SYSAA`
salvocfg.h	
compiler auto-detected?	yes[1]
libraries	
`\salvo\lib` subdirectory	`tic2000`
context switching	
method	function-based via `OSDispatch()`
`_OSLabel()` required?	no
size of auto variables and function parameters in tasks	total size must not exceed 254 16-bit bytes
memory & registers	
addressing mode ('C28x)	direct addressing used, therefore Salvo objects must reside within first 4M of data space
interrupts	
controlled via	`INTM` bit
interrupt status preserved in critical sections?	yes
method used	saved in memory via `OSEnterCritical()`
nesting limit	16 levels
alternate methods possible?	no
debugging	
source-level debugging with Pro library builds?	yes
compiler	
bitfield packing support?	no
printf() / %p support?	yes / yes
va_arg() support?	yes

Table 1: Features of Salvo Port to TI's Code Composer Studio 'C2000 C Compiler

Compiler Optimizations

Incompatible Optimizations – 'C240x

The highest allowable setting for the compiler's Optimization Level *for source (*.c) modules containing Salvo tasks* is Optimization Level 1: Local [-o1]. Higher levels (e.g. Global [-o2] and File [-o3]) will cause problems with the Salvo context switcher.

The highest allowable setting for the compiler's Inline Options is Intrinsic Operators Only [-x1]. Higher levels (e.g. Full [-x2]) will cause problems with the Salvo context switcher.

Note In cases where the effects of optimization are otherwise beneficial to an application, it is recommended that individual source code (*.c) modules be dedicated to holding Salvo tasks and no other functions. Thus, the scope of disabling the optimizations can be limited to just those modules that contain Salvo tasks, thereby allowing the use of this optimization on other functions where it is appropriate.

Libraries

Nomenclature

The Salvo libraries for TI's Code Composer Studio 'C2000 C compiler follow the naming convention shown in Figure 1.

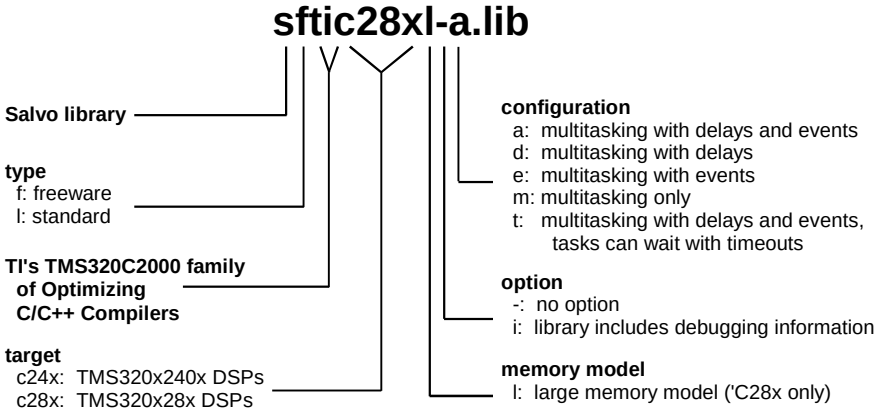

Figure 1: Salvo Library Nomenclature – TI's Code Composer Studio 'C2000 C Compiler

Type

Salvo Lite distributions contain *freeware* libraries. All other Salvo distributions contain *standard* libraries. See the *Libraries* chapter of the *Salvo User Manual* for more information on library types.

Target

Each library is intended for one or more specific processors. Table 2 lists the correct library for each C2000 DSP.

target code	processor(s)
`24x:`	all TMS320x240x (e.g. TMS320LC2401A, TMS320LF2407A, etc.)
`28x:`	all TMS320x28x (e.g. TMS320F2810, TMS320F2812, etc.)

Table 2: Processors for Salvo Libraries – TI's Code Composer Studio 'C2000 C Compiler Options

Memory Model

Where applicable, only the large memory model is supported.

Option

Salvo Pro users can select between two sets of libraries – standard libraries, and standard libraries incorporating source-level debugging information.[2] The latter have been built with TI's Code Composer Studio 'C2000 C compiler's `-g` command-line option. This adds source-level debugging information to the libraries, making them ideal for source-level debugging and stepping in the CCS debugger. To use these libraries, simply select one that includes the debugging information (e.g. `sltic28xlit.lib`) instead of one without (e.g. `sltic28xl-t.lib`) in your CCS project.

Configuration

Different library configurations are provided for different Salvo distributions and to enable the user to minimize the Salvo kernel's footprint. See the *Libraries* chapter of the *Salvo User Manual* for more information on library configurations.

Build Settings

Salvo's libraries for TI's Code Composer Studio 'C2000 C compiler are built using the default settings outlined in the *Libraries* chapter of the *Salvo User Manual*. Target-specific settings and overrides are listed in Table 3.

compiled limits	
max. number of tasks	3
max. number of events	5
max. number of event flags	1
max. number of message queues	1
target-specific settings	
delay sizes	8 bits
watchdog timer	not affected
system tick counter	available, 32 bits

Table 3: Build Settings and Overrides for Salvo Libraries for TI's Code Composer Studio 'C2000 C Compiler

Note The compiled limits for tasks, events, etc. in Salvo libraries can be overridden to be less (all Salvo distributions) or more (all Salvo distributions except Salvo Lite) than the library default. See the *Libraries* chapter of the *Salvo User Manual* for more information.

Available Libraries

There are 30 Salvo libraries for TI's Code Composer and Code Composer Studio 'C2000 C compilers. Each Salvo for TI's TMS320C2000 DSPs distribution contains the Salvo libraries of the lesser distributions beneath it.

Target-Specific Salvo Source Files

Due to architectural and instruction-set differences, one of two different source files, `porttic24x.asm` or `porttic28x.asm`, is required for Salvo Pro source code builds. Use the one appropriate for your target as per the target code nomenclature shown in Table 2.

salvocfg.h Examples

Below are examples of `salvocfg.h` project configuration files for various different Salvo for TI's TMS320C20000 DSPs targeting the TMS320C2812.

Note When overriding the default number of tasks, events, etc. in a Salvo library build, `OSTASKS` and `OSEVENTS` (respectively) *must also be defined* in the project's `salvocfg.h`. If left undefined, the default values (see Table 3) will be used.

Salvo Lite Library Build

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE          OSF
#define OSLIBRARY_CONFIG        OSA
```

Listing 1: Example salvocfg.h for Library Build Using sftic28xl-a.a

Salvo LE & Pro Library Build

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE          OSL
#define OSLIBRARY_CONFIG        OSA
```

Listing 2: Example salvocfg.h for Library Build Using sltic28xl-a.a or sltic28xlia.a

Salvo Pro Source-Code Build

```
#define OSENABLE_IDLING_HOOK    TRUE
#define OSENABLE_SEMAPHORES     TRUE
#define OSEVENTS                1
#define OSTASKS                 3
```

Listing 3: Example salvocfg.h for Source-Code Build

Performance

Memory Usage

Salvo Test System SYSW – 'F2812

tutorial memory usage[3]	total ROM[4]	total RAM[5]
`tu1lite`	312	23
`tu2lite`	370	23
`tu3lite`	388	24
`tu4lite`	627	34
`tu5lite`	878	48
`tu6lite`	946	50
`tu6pro`	957	50

Table 4: ROM and RAM requirements for Salvo Applications built with TI's Code Composer Studio 'C2000 C Compiler

---

1 This is done automatically through the `_TMS320C2XX` and `__TMS320C28XX__` symbols defined by the compilers.

2 The Salvo libraries provided with Salvo Lite and LE do not contain CCS-debugger-compatible debugging information because this requires the inclusion of source file listings.

3 Salvo v3.2.0-b with CCS v2.2.0.

4 In bytes. Sum of `.cinit`, `.text`, `.reset`, and `vectors` sections.

5 In bytes, `.ebss` section. Includes only Salvo and application `.obj` modules (no stack, run-time modules, etc.).