



# Cosmic Software C Cross Compiler for STMicroelectronics ST10/Super 10

**C**osmic's C cross compiler, *cxst10* for STMicroelectronics' ST10 and Super10 family of microcontrollers, incorporates over ten years of innovative design and development effort.

*cxst10* is field tested, reliable, and incorporates many features that help ensure your embedded ST10 design meets and exceeds performance specifications.

Cosmic's **cxst10 Compiler** package includes: an optimizing C cross compiler, macro assembler, linker, librarian, hex file generator, debugging support utilities and ANSI run-time library objects and source code. **CXST10** is available for Windows 95/98/NT4/2000/XP, PC-Linux, SUN Solaris and HP-UX.

The Compiler package, *cxst10* for Windows includes Cosmic's own **IDEA** integrated project manager and editor. IDEA is designed specifically for embedded development and provides comprehensive C language support along with the low level features needed to utilize every feature of the ST10 and Super10 microcontroller families.

## Key Features

Supports All ST10/Super10 Microcontrollers

ANSI/ISO C Language Implementation

Extensions for Embedded Systems

Global and Processor-Specific Optimizations

Automated Checksum facility

Moveable Code feature for Bootloader

Smart Linker

C support for Bit Variables

C support for Interrupt Handlers

In-Line Assembly

Absolute addressing feature for I/O

Single Precision Float Support

Absolute C and Assembly Listings

IEEE-695 Debug support

Royalty-Free Library Source Code

First Year of Support Service Included

No Charge Upgrades

Also supports Infineon **C16X** and **XC16X** families

## Microcontroller Specific Design

*cxst10* is designed specifically for the STMicroelectronics **ST10** family of microcontrollers; all **ST10** family processors are supported. A **special code generator and optimizer** targeted for the **ST10** family eliminates the overhead and complexity of a more generic compiler. You also get header file support for many of the popular **ST10** peripherals, so you can access their objects by name either at the C or assembly language levels.

## ANSI / ISO Standard C

This implementation conforms with the **ANSI and ISO Standard C** specifications which helps you protect your software investment by aiding code portability and reliability.

## C Runtime Support

C runtime support consists of a subset of the standard ANSI library, and is provided in C source form with the binary package so you are free to modify library routines to match your needs. The basic library set includes the support functions required by a typical embedded system application. All runtime library functions are **ROMable**.

Runtime library functions include :

- Character handling.
- Mathematical functions.
- Formatted serial input/output.
- String handling.
- Memory management.

The package provides both an **integer-only library** as well as the standard **single precision floating point library**. This allows you to select the smaller and faster integer-only functions, if your application does not require floating point support.

## Optimizations

*cxst10* includes global and microcontroller specific optimizations to give your application maximum chance of meeting and exceeding its performance specifications. You retain control over optimizations via compile-time options and keyword extensions to ANSI C, so you can fine tune your application code to match your design specification:

- *cxst10* supports global optimizations which allow it to optimize whole C functions as well as C statements.
- Peephole optimizer further optimizes *cxst10*'s output by replacing inefficient code sequences with optimal code sequences for the **ST10**.
- The **ST10 bit instructions** are used extensively for bit operations and bit variables.
- Strict single-precision (32-bit) floating point arithmetic and math functions. Floating point numbers are represented as in the IEEE754 Floating Point Standard.
- Other optimizations include: branch shortening logic, jump-to-jump elimination, constant folding, elimination of unreachable code, removal of redundant loads/stores, and switch statement optimizations.

## Controller Specific Extensions to ANSI C

*cxst10* includes several extensions to the ANSI C standard which have been designed specifically to give you maximum control of your application at the C level and to simplify the job of writing C code for your embedded ST10 design :

- You can define in-line assembly using `_asm()` to insert assembly instructions directly in your C code to avoid the overhead of calling assembly language subroutines.
- Also you can use `#asm/#endasm` to insert assembly instructions directly in your C code.
- You can define C functions as interrupt handlers using the `@interrupt` keyword. Compiler saves volatile registers for handling exceptions and interrupts.
- You can define a C object or C function to have an absolute address at the C-level, using the `@<address>` syntax appended to you data definition; this is useful for interrupt handlers written in C and for defining memory mapped I/O.
- You can define `char`- and `int`-sized bitfields, and select bit numbering from right-to-left or left-to-right.

## Additional Compiler Features

- Full C and assembly source-level debugging support.
- Bit variables (`_Bool`) which can be either allocated in the bit addressable memory section, or mapped to any bit of bit addressable SFR's.
- Absolute and relocatable listing file output, with interspersed C, assembly language and object code; optionally, you can include compiler errors and compiler optimization comments.
- Extensive and useful compile-time error diagnostics.

- Fast compile and assemble time.
- Full user control over include file path(s), and placement of output object, listing and error file(s).
- All objects are relocatable and host independent. (i.e. files can be compiled on a workstation and debugged on a PC).
- Function code and switch tables are generated into the code (`.text`) section. Constant data such as string constants and `const` data are generated into a separate program (`.const`) section.
- Initialized static data can be located separately from uninitialized data or data initialized to zero.
- All function code is never self-modifying, including structure assignment and function calls, so it can be shared and placed in ROM.
- Code is generated as a symbolic assembly language file so you can examine compiler output.
- *cxst10* creates all its tables dynamically on the heap, allowing very large source files to be compiled.

## Assembler

The Cosmic ST10 assembler, *cast10*, supports macros, conditional assembly, includes, branch optimizations, expression evaluation, relocatable or absolute output, relocatable arithmetic, listing files and cross references. *Cast10* supports C syntax for `#includes` and `#defines` so include files can be shared between C and Assembly modules. The assembler also generates debug information, so that Cosmic's ZAP debugger can perform full source-level debug at the assembly language level.

## Linker

The Cosmic linker, *clnk*, combines relocatable object files created by the assembler, selectively loading from libraries of object files made with the librarian, *clib*, to create an executable format file.

*clnk* features:

- Flexible and extensive user-control over the linking process and selective placement of user application code and data program sections.
- Generation of memory map information to assist debugging.
- All symbols and relocation items can be made absolute to prelocate code that will be linked in elsewhere.
- Symbols can be defined, or aliased, from the Linker command File.

## Librarian

The Cosmic librarian, *clib*, is a development aid which allows you to collect related files into one named library file, for more convenient storage. *clib* provides the functions necessary to build and maintain object module libraries. The most obvious use for *clib* is to collect related object files into separate named library files, for scanning by the linker. The linker loads from a library only those modules needed to satisfy outstanding references.

## Object Module Inspector

---

The Cosmic object module inspector, *cobj*, allows you to examine library and relocatable object files for symbol table and file information. This information is an essential aid to program debugging.

- Symbol table cross referencing.
- Section sizes of the individual program sections can be printed for object and library files.
- Program segment map: lists all program segments, their sizes, absolute addresses and offsets.

## Absolute Hex File Generator

---

The Cosmic hex file generator, *chex*, translates executable images produced by the linker to one of several hexadecimal interchange formats for use with most common In-Circuit

Emulators and PROM programmers :

- Standard Intel hex format.
- Freescale S-record format.
- Rebiasing of text and data section load addresses. Allows you to generate hex files to load anywhere and execute anywhere in the target system address space.

## Absolute C and Assembly Listings

---

Paginated listings can be produced to assist program understanding. Listings can include original C source code with interspersed assembly code and absolute object code. Optionally, you can include compiler errors and optimization comments.

## Debugging Utilities

---

The cross compiler package includes utility programs which provide listings for all debug and map file information. The *clst* utility creates listings showing the C source files that were compiled to obtain the relocatable or executable files. The *cprd* utility extracts and prints information on the name, type, storage class and address of program static data, function arguments and function automatic data.

## Third Party Debugging Support

---

You can use *cxst10* or *cast10* with ZAP/SIM or with the debuggers provided by most popular In-Circuit Emulator manufacturers Lauterbach and Hitex. *Cxst10* also supports additional debugging formats including IEEE-695 format.

## Packaging

---

All compiler packages are provided on standard CD-ROM with on-line user documentation in Adobe PDF format.

## Support Services

---

All Cosmic Software products come with the first year of support included in the price. You will receive a courteous and prompt service from our technical support staff and **you retain control of the severity of the problem** i.e. if it's a problem that is critical to your project we guarantee you a response time of one to three business days depending on the severity of the problem. Service is provided during normal business hours via email, fax or telephone and is unlimited while you have a valid annual support agreement. New releases of the software are provided free of charge to support customers.

## Ordering Information

---

*cxst10* package product codes are as follows :

[Host System Product Code](#) :

|                           |                   |
|---------------------------|-------------------|
| PC (DOS/Windows)          | <b>CXST10-PC</b>  |
| SUN SPARC (SunOS/Solaris) | <b>CXST10-SUN</b> |
| HP9000(HPUX)              | <b>CXST10-HP</b>  |

Please contact our sales department or local distributor for license options, discounts and delivery information.

## Other Cosmic Software Products

---

Cosmic Software products focus on 8, 16 and 32-bit microcontrollers. C-Compiler/debugger support is available for a wide range of target processors. For more information on the ZAP C and assembler source-level debugger, ask for the ZAP Product Description and demo disk.

## Tool Customization Services

---

Some customers have special tool needs and through Cosmic's tool customization service, you have the ability to control the core tool technology to help solve your technical and/or business problems. Cosmic works closely with you to understand, define and implement technical solutions according to your needs and schedule.



*Supporting Embedded Innovation  
Since 1983*

**For more Information please contact one of our  
corporate offices or visit our website:**



**Cosmic Software, Inc.**

400 West Cummings Park, Suite 6000  
Woburn, MA 01801-6512 USA  
Phone: +1 781 932 2556 Fax: +1 781 932 2557  
Email: [sales@cosmic-us.com](mailto:sales@cosmic-us.com)  
web: [www.cosmic-software.com](http://www.cosmic-software.com)



**Cosmic Software France**

33 Rue Le Corbusier, Europarc Creteil  
94035 Creteil Cedex France  
Phone: +33 1 43 99 53 90 Fax: +33 1 43 99 14 83  
Email: [mailto:sales@cosmic.fr](mailto:mailto:sales@cosmic.fr)  
web: [www.cosmic.fr](http://www.cosmic.fr)



**Cosmic Software UK**

Oakwood House  
Wield Road, Medstead  
Alton, Hampshire  
GU34 5NJ, U.K.  
Phone: +44 1420 563498 Fax: +44 1420 561946  
Email: [sales@cosmic.co.uk](mailto:sales@cosmic.co.uk)



**Cosmic Software GmbH**

Rohrackerstr 68 D-70329 Stuttgart Germany  
Tel.+49 711 420 4062 Fax +49 711 420 4068  
Email: [sales@cosmic-software.de](mailto:sales@cosmic-software.de)  
web: [www.cosmic-software.de](http://www.cosmic-software.de)