



Release Notes for CrossCore Embedded Studio 2.9.0

1 Table of Contents

1	Table of Contents.....	2
2	Introduction	3
3	New and Noteworthy	4
3.1	Ubuntu Support	4
3.2	Add option to generate link map file to ARM (Cortex-A/M).....	4
3.3	FFTA accelerator functions for inverse FFT giving real output.....	4
4	Support for New Silicon Revisions	6
5	Changes That Might Impact Backwards Compatibility.....	7
5.1	GCC ARM Embedded toolchain for Cortex-M parts updated	7
5.2	GNU ARM toolchain for Cortex-A parts updated	7
5.3	Changes to part support switches in Cortex-A toolchain	8
5.4	Default breakpoints added for Cortex-A debug sessions	8
5.5	Assembler .IMPORT directive now searches assembly source directory first	8
5.6	elfloader -bcode defaults changed and ea0274 now a warning	8
5.7	ARM (Cortex-A/M) LD scripts are pre-processed.....	9
5.8	Current working directory for launch configurations using GDB/OpenOCD or GDB/QEMU has been changed	9
5.9	By default the \$PC is set to program(s) entry address when launching a debug session.....	10
5.10	Silicon revision 'none' has been deprecated.....	10
5.11	Assembler option for omitting the .commandLine section.....	10
5.12	Symmetric and more precise FFT twiddle tables for SHARC.....	10
5.13	MODE1:RND32 now set conditionally in SHARC interrupt handling	10
5.14	Deprecation of Micrium μ C/OS and μ C/USB demo libraries	10
6	Known Issues.....	12

2 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.9.0. You can find the release notes for older releases in the docs sub-directory of your CCES installation as well as an Installation Guide which will help you install this release.

3 New and Noteworthy

3.1 Ubuntu Support

CrossCore Embedded Studio (CCES) is no longer supported running on Ubuntu 14.04 versions. Support for Ubuntu 16.04 32-bit versions is unchanged for CCES 2.9.0 but will be removed in a future CCES release.

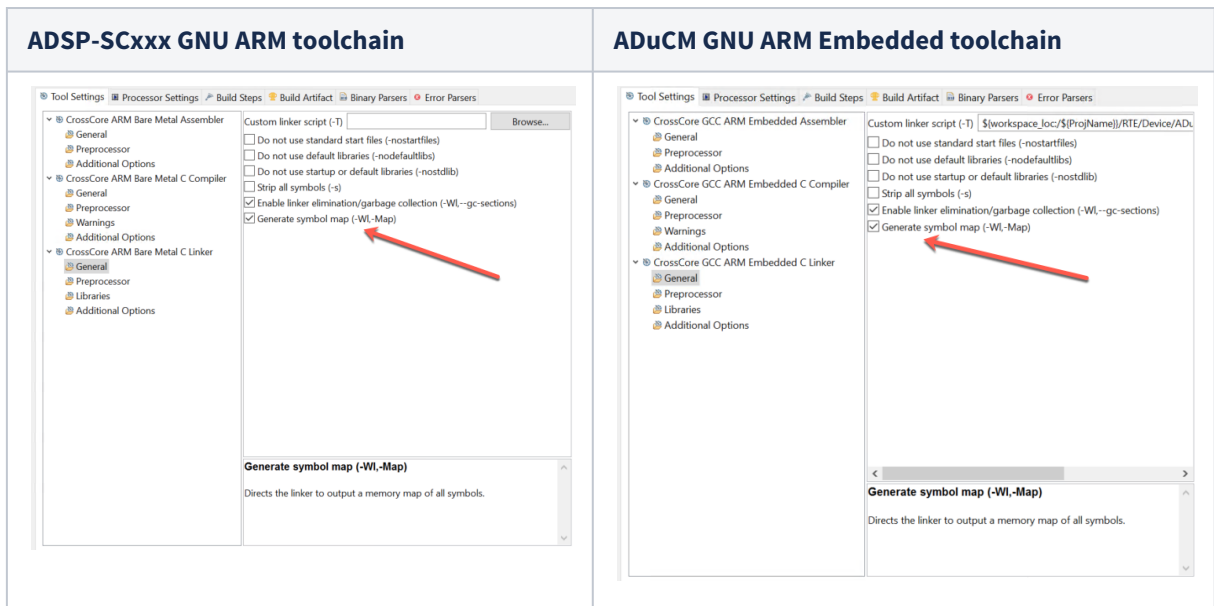
CCES can be installed and used on Ubuntu 18.04 64-bit. To install and use CCES on Ubuntu 64-bit, the following 32-bit compatibility libraries need to be installed:

```
sudo dpkg --add-architecture i386 && sudo apt-get update && sudo apt-get install -y \  
libc6:i386 libcurses5:i386 libstdc++6:i386 libgtk2.0-0:i386 libxtst6:i386 \  
gtk2-engines-murrine:i386 libcanberra-gtk-module:i386 gtk2-engines:i386
```

3.2 Add option to generate link map file to ARM (Cortex-A/M)

A new option to output a memory map of all the symbols has been added to the Linker | General category for ADSP-SCxxx ARM (Cortex-A) and ADuCM ARM (Cortex-M) projects.

When the new `Generate symbol map` option is checked, the command line switch `-Wl, -Map=${ProjName}.map` is added to the ARM Linker command line.



3.3 FFTA accelerator functions for inverse FFT giving real output

The following functions have been added to the `libfftacc` library:

- `accel_irfft_small`
- `accel_irfft_small_pipe`
- `accel_irfft_small_windowed`

- `accel_irfft_small_windowed_pipe`
- `accel_irfft_large`
- `accel_irfft_large_pipe`
- `accel_irfft_large_windowed`
- `accel_irfft_large_windowed_pipe`
- `accel_irfft_large_set_tcbs`
- `accel_irfft_large_set_tcbs_strided`
- `accel_irfft_large_windowed_set_tcbs`
- `accel_irfft_large_windowed_set_tcbs_strided`

These functions are used for computing an inverse FFT in cases where only the real part of the result is required. Please see the documentation at [C/C++ Library Manual for SHARC® Processors > DSP Run-Time Library > DSP Run-Time Library Reference](#) for further information.

4 Support for New Silicon Revisions

CCES 2.9.0 provides support for revision 1.2 of the ADSP-SC58x/ADSP-2158x SHARC family DSPs and revision 0.2 of the ADSP-SC57x/ADSP-2157x SHARC family DSPs.

5 Changes That Might Impact Backwards Compatibility

5.1 GCC ARM Embedded toolchain for Cortex-M parts updated

The toolchain for ARM Cortex-M parts, such as ADuCM36x, ADuCM302x and ADuCM4050 processors, has been updated to the gcc-arm-none-eabi-7-2017-q4-major release. This version introduces support for the ARMv8-R architecture and for ARM Cortex-R52 parts.

The following components are used in this version:

- gcc : ARM/embedded-7-branch revision 255204 (7.2.1 instead of 6.3.1)
[svn://gcc.gnu.org/svn/gcc/branches/ARM/embedded-7-branch/](https://svn.gnu.org/svn/gcc/branches/ARM/embedded-7-branch/)
- binutils : 2.29 mainline (instead of 2.8)
[git://sourceware.org/git/binutils-gdb.git](https://sourceware.org/git/binutils-gdb.git) commit
b23dc97fe237a1d9e850d7cbeee066183a00630b
- newlib and newlib-nano
[git://sourceware.org/git/newlib-cygwin.git](https://sourceware.org/git/newlib-cygwin.git) commit
76bd5cab331a873ac422fdb7ba5fe79abea94f0
- gdb : 8.0 mainline (instead of 7.12.1)
[git://sourceware.org/git/binutils-gdb.git](https://sourceware.org/git/binutils-gdb.git) commit
b23dc97fe237a1d9e850d7cbeee066183a00630b

Known issues:

- Thumb1 code size regression due to new register allocation: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=59535

5.2 GNU ARM toolchain for Cortex-A parts updated

The GNU toolchain for the ARM Cortex-A cores on ADSP-SC5xx parts has been updated. It is installed in directory ARM/arm-none-eabi of the CCES install.

The following components are used in this version:

- gcc-linaro 7.2-2017.11 (instead of 4.8-2014.04)
- binutils 2.30 (instead of 2.24.51)
- newlib 3.0.0 (instead of 2.1.0)
- gdb 8.1 (instead of 7.8.1)

The GCC update changes the default language standards supported for C and C++:

- The default for C is `-std=gnu11`, meaning ISO C11 with GNU extensions. Previously it was `-std=gnu89`.
- The default for C++ is `-std=gnu++14`, meaning ISO C++14 with GNU extensions. Previously it was `-std=gnu++98`.

Where necessary, the previous language standard can be restored by adding the appropriate `-std` option to the compiler additional options in CCES project tool settings.

The switch to gnu11 for C changes the default semantics for non-static inline functions: previously a definition of a non-static inline function would both make the function available for inlining and create an

out-of-line global definition of the function that can be linked against when the function is not inlined. Now, as per the semantics introduced in C99, creating an out-of-line definition of an inline function additionally requires the `extern` keyword. The original inline semantics can be restored using `switch -fgnu89-inline`.

The switch to `gnu11` also enables some warnings by default, most notably, warnings about invoking a function without declaration and declaring a function without return type.

Further details about the changes in language support, particularly regarding C++, can be found in the following documents:

- https://gcc.gnu.org/gcc-5/porting_to.html
- https://gcc.gnu.org/gcc-6/porting_to.html
- https://gcc.gnu.org/gcc-7/porting_to.html

5.3 Changes to part support switches in Cortex-A toolchain

As before, the GNU toolchain for the ARM cores on ADSP-SC5xx parts defaults to building for ADSP-SC589 when no `-mproc` option for specifying an ADI target processor is given. However, if a `-march` or `-mcpu` option for specifying a target ARM architecture or non-ADI CPU type is given, they will now be respected when compiling or assembling, whereas previously they were silently ignored or triggered an error.

Please note though that no runtime libraries are provided for targets other than ADSP-SC5xx processors, which means that linking for such targets will fail unless required runtime libraries are provided separately.

5.4 Default breakpoints added for Cortex-A debug sessions

In order to ease debugging of failed assertions and runtime errors, default breakpoints have been added for symbols `abort` and `__fatal_error` in debug sessions for ADSP-SC5xx Cortex-A cores.

Moreover, the implementation of `__fatal_error` has been amended to send signal SIGABRT before looping, in order to cause a failed program to abort rather than hang when run using the `qemu-system-arm` command.

5.5 Assembler .IMPORT directive now searches assembly source directory first

The Blackfin and SHARC assemblers' `.IMPORT` directive will now look for the imported file relative to the location of the importing assembly source file prior to looking relative to the current working directory and searching the include paths. This is to allow a file in the same directory as the assembly source to be imported straightforwardly when the working directory is different from the source directory.

However, that does mean that if a file of the same name exists both in the source directory and in the working directory or the include search paths, the `.IMPORT` directive will now pick up the file in the source directory whereas previously it would have picked up the other one. This matches the behavior of the `#include` preprocessor directive.

5.6 elfloader -bcode defaults changed and ea0274 now a warning

Using CCES 2.8.0 and earlier the elfloader tool for ADSP-BF70x, ADSP-SC5xx and ADSP-215xx parts would raise error `ld0274` if `-bcode` was not used even for boot modes that don't read the BCODE value.

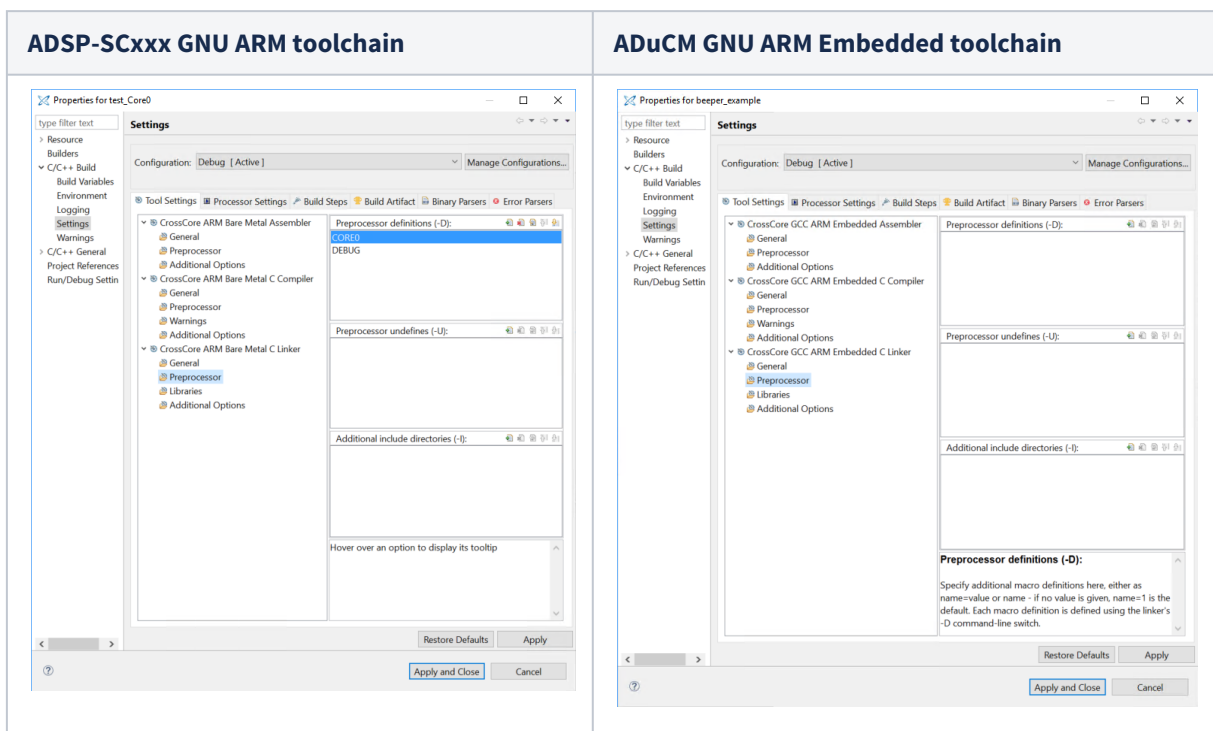
In CCES 2.9.0 elfloader support for ADSP-BF70x, ADSP-SC57x and ADSP-SC5xx parts has changed as follows:

- ld0274 is now a warning and is only raised when creating loader files for the SPIMASTER boot mode as that's the only mode that uses BCODE. For other boot modes -bcode can be used but there will be no warning or error when it is not used.
- The value put in the BCODE field for loader file block headers now defaults to 1 instead of 0 when -bcode is not used. If you need to compare loader files created with CCEs 2.9.0 and earlier releases you may need to use -bcode 0 to avoid seeing this change as a difference

5.7 ARM (Cortex-A/M) LD scripts are pre-processed

LD script used to link ADSP-SCxxx ARM (Cortex-A) and ADuCM ARM (Cortex-M) projects are now pre-processed before being added to the ARM Linker command line.

A new Preprocessor category has also been added to the Link Tool Settings:



5.8 Current working directory for launch configurations using GDB/OpenOCD or GDB/QEMU has been changed

The current working directory for a GDB/OpenOCD or GDB/QEMU debug session is now set to the directory containing the program that you are debugging.

5.9 By default the \$PC is set to program(s) entry address when launching a debug session

If you specify a program to load and run in your Debug launch configuration and the program's entry address is non-zero, then the IDE will set the \$PC to that entry address before running the program.

5.10 Silicon revision 'none' has been deprecated

Silicon revision 'none', which prevents workarounds for silicon errata being applied, has been deprecated. It will continue to work under CrossCore Embedded Studio 2.9.0 but shall be removed in a subsequent version. Applications should use the revision of the processor being used, or 'any'.

5.11 Assembler option for omitting the .commandLine section

Assembler option *-no-cl-sec* has been added to omit the *.commandLine* section from the assembled object file. That section records the command line that the assembler was invoked with.

This option can be used when compiling C/C++ files (using option *-flags-asm -no-cl-sec*) to avoid unnecessary object file differences due to randomized temporary assembly file names being recorded into the *.commandLine* section. The option is used in the C/C++ runtime library build.

5.12 Symmetric and more precise FFT twiddle tables for SHARC

The FFT twiddle table generation functions *twidfft*, *twidfft* and *accel_twidfft* as well as pre-defined twiddle tables have been updated to be fully symmetric and more precise, in order to reduce crosstalk in the *rfft_2* function and generally reduce noise introduced in the FFTs.

On SHARC+ processors, the twiddle tables now are correctly rounded to 32-bit floating-point precision. On previous SHARC processors, they are now within 2 LSBs (least significant bits) of the precise mathematical results, with an average error of around 0.3 LSBs. The tables now fully observe symmetry $\cos(x) = \sin(\pi+x)$ between their real and imaginary parts.

5.13 MODE1:RND32 now set conditionally in SHARC interrupt handling

Previously the RND32 (rounding for 32-bit floating-point) bit in MODE1 (Mode Control 1) SHARC register would be set unconditionally for interrupt handlers to disable extended precision 40-bit floats. In CCES 2.9.0 the interrupt dispatchers have been updated to not set MODE1:RND32 if the corresponding RND32 bit in MMASK (Mode1 Mask Register) is set.

5.14 Deprecation of Micrium μ C/OS and μ C/USB demo libraries

CrossCore[®] Embedded Studio provides pre-built versions of the μ C/OS-III real-time operating system and μ C/USB Device[™] Stack libraries for evaluation purposes. This functionality will be removed from a future version of CrossCore[®] Embedded Studio. Similar functionality can be achieved by installing the μ C/OS-III Add-In and μ C/USB Device[™] Stack for CrossCore[®] Embedded Studio. μ C/OS-III Add-In for CrossCore

Embedded Studio has an evaluation license. User need to purchase the license for μ C/USB Device™ Stack for CrossCore® Embedded Studio, though.

6 Known Issues

Build failures may be seen while using Ethernet driver libraries for ARM Core. Users need to add Ethernet driver in their project by adding "system.svc → Device Drivers and System Services → On-chip Peripheral drivers → Ethernet Driver for Cortex-A" plugin in order to use the same.

For the latest anomalies please consult our [Software and Tools Anomalies Search](#) page.