
TASKING M16C Toolchain v3.0r1

RELEASE NOTE

Introduction

This release note describes the new M16C v3.0r1 toolchain.

The M16C toolchain v3.0 consists of a full embedded development toolchain, supporting:

- Renesas M16C/10, M16C/20, M16C/60 and R8C/tiny series of microprocessors.
- ISO C99, ANSI C90, MISRA C, C++, EC++ and assembly language, with M16C specific extensions.
- M16C/R8C development boards from Renesas, Glyn and Oaks.
- Debugging in the Crossview Pro source level debugger, using the M16C/R8C simulator, TASKING rom monitor or Renesas emulator.

The M16C toolchain v3.0 is based on new TASKING compiler and linker technology.

Features

The compiler offers full compliance with the latest ISO C 1999 standard, for instance the new datatypes `_Bool`, `long long` and `_Complex` are supported, code and declarations can be mixed, and arrays can be of variable length.

Besides the familiar small and large memory models, v3.0 introduces the medium memory model. This memory model maps naturally to the typical M16C derivative: ROM memory is available in the far space, and RAM memory is available in the near space. When using the medium memory model, constants and string literals will be allocated in the far space, while variables will end up in the near space.

Another novelty is the `__paged` qualifier. `__paged` qualified objects will be allocated in the `far` space, but will not overlap a 64k page boundary. In this way, pointer arithmetic will be more efficient. Constants, string literals and pointers are implicitly `__paged` qualified in the new medium memory model.

The new linker offers many advanced optimizations like removal of unused sections. It uses a new Linker Script Language (LSL), that is normally auto-generated from the Embedded Development Environment (EDE). The default output format has changed to ELF/Dwarf, the IEEE-695 output format is no longer supported.

Migration from previous TASKING M16C toolchains

With the upgrade to the new technology it was not possible to maintain full backwards compatibility with v2.3. However, we have included a conversion option to help you migrate projects from a previous

TASKING M16C toolchain.

Project options are automatically converted when loading an older project in the TASKING M16C v3.0 EDE.

But also the C-language has changed. Besides supporting the new ISO-99 keywords/extensions the main difference is the use of double instead of single underscores for language extensions, like memory space qualifiers and intrinsic functions. For example the bit type specifier is `__bit` instead of `_bit`. Also the `#pragma asm` construction has been replaced with the more powerful `__asm()` construction.

To ease conversion from existing ANSI-C 90 M16C code to ISO-C99 M16C code a conversion utility is added to the EDE. This tool can be started using the  button in the EDE toolbar. This utility allows conversion of a single module or all modules in the active project.

As the new toolchain is based on new technology, all known compiler and linker bugs in older M16C toolchains are solved in this release.

License

To obtain a license file, please contact your local sales / support center:

<http://www.tasking.com/about/contacts/>.

Support

Please use the EDE "Prepare an E-mail" item on the Help - Technical Support menu to report problems with this product.

Visit website <http://www.tasking.com/support/m16c/> to get the latest news about this product.

[TASKING M16C Support web site](#)

[TASKING M16C Product Page](#)

[TASKING Home Page](#)

[TASKING Developers' Forum](#)

[Copyright 2003-2004 Altium BV](#)