

ADC-11-Linux C Compiler, Assembler, Linker for Linux

Professional Linux Tools Include 12-Months Professional Technical Support

C Compiler

- Supports all popular 68HC11 derivatives
- Standard C syntax
- C interrupt support
- In-line assembly
- Generates assembly source
- Reentrant banking support
- 32-bit and 64-bit floating point
- Macro support with parameters

Macro Assembler

- 248 levels of nested conditions
- Generates relocatable code
- Linker supports 32-bit addressing
- Fourteen types of debug formats are directly generated from the Linker

The HMI-200-68HC11 In-Circuit Emulator for the 68HC11 is also part of our complete solution for 68HC11 development. Ask about our Complete Package for the HC11



HMI-200-68HC11 Emulator

C Compiler

The 2500AD 68HC11 C Compiler for Linux is designed specifically for tight, fast running embedded applications on an 68HC11. Yet the compiler is fully reentrant even when using the built in bank switching features. Floating point operations are processed using included libraries that handle 32-bit floats or 64-bit doubles. The number of accurate digits can be manually set at run time to allow you to execute the routines very quickly when accuracy is not the primary concern. Interrupt routines can also be fully coded in C.

While you may never need to use inline assembly code, the compiler does allow for assembly code, assembly directives and linker directives to be embedded directly into the C source code. Special Function Registers may be directly accessed via C source as well.

Macro Assembler

The professional grade 2500AD Macro Cross Assembler for the 68HC11 offers a complete set of macro and relocatability features. Macro capabilities add high level structure and modularity to your assembly code. Conditional expressions can be nested up to 248 levels and can exist inside or outside macro definitions. Macros can be used to assemble in completed sections of code based on multiple parameters.

The Linker handles full 32-bit addressing for bank switching. The relocatable nature of the assembler allows all code placements to occur at link time adding additional modularity to your coding. Commonly used modules may also be organized into libraries for selective linking based on usage. Fourteen types of debug formats are directly generated from the Linker. If your format is not explicitly supported, we may have an existing converter for you or can provide one to fit your needs. Our level of compatibility is incomparable.

The Complete Solution for Embedded Systems Development Tools

