



Embedded in Life

An IXYS Company

Z8 Encore!® Microcontroller

Zilog Developer Studio II

Product Brief

PB009708-1010

Zilog Developer Studio II

Introduction

Zilog Developer Studio II (ZDS II) Integrated Development Environment is a complete stand-alone system that provides a state-of-the-art development environment. Based on standard Windows user interfaces, ZDS II integrates a language-sensitive editor, project manager, C-Compiler, assembler, linker, librarian, and source-level symbolic debugger to provide a development solution specifically tailored to the Z8 Encore!® line of microcontrollers.

ZDS II Products Integrated for the Z8 Encore!®

Assembler	ez8asm
Compiler	ez8cc
Linker	ez8link
Librarian	ez8lib
Simulator	Yes
Flash Loader	Yes
Host Communication	RS-232, USB, Ethernet

Figure 1 illustrates a typical display interface showing many of the features of ZDS II.

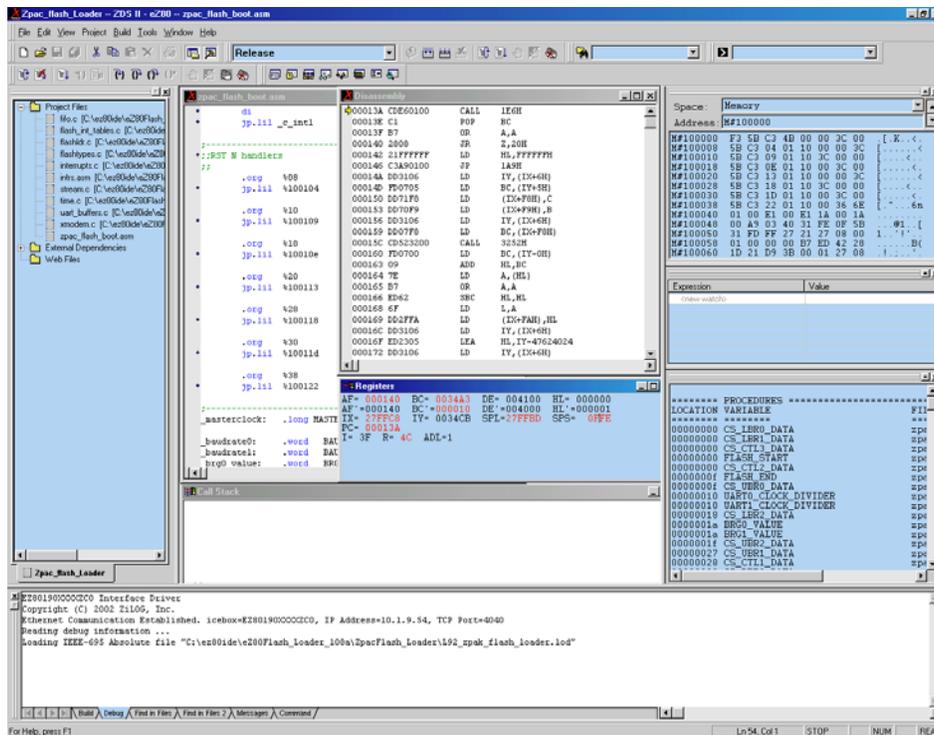


Figure 1. ZDS II's Easy-To-Use Interface

Easy-To-Use Interface

ZDS II provides a standard user interface with intuitive, easy-to-use controls commonly found in Windows®-based environments. The system contains an integrated set of windows, document views, menus, and toolbars to create, test and refine applications without having to alternate between different systems.

Flexible and Adaptable Design Capabilities

Designed to use the multithreading capability of the host operating system, multiple operations can be performed efficiently and easily with ZDS II. Working in this type of multitasking environment allows file editing while simultaneously compiling code.

Save Valuable Time

The basic components of ZDS II are an assembler, eZ8 C-Compiler, linker, debugger, and editor. Each of these modules plays an important role in helping a developer compile a product in record time.

Incorporating auto-completion and auto-insert for braces and quotes, the Intelligent Editor allows users to read and write code faster, navigate intelligently and identify and correct mistakes.

Combining a simple interface and flexible design elements results in full command of design applications and quick time-to-market.

Features

- Integrated Project Management—create, add, or remove project files
- Intelligent Editor Support that allows you to read and write code faster, navigate intelligently and identify and correct mistakes
- Color configuration per syntax style
- *Find*, *Find in Files* and *Replace* capabilities
- Optimizing ANSI eZ8 C-Compiler
- Compile, Build, and Rebuild All capabilities

- Download, Execute, Debug, and Analyze
- Language-sensitive editor
- Print the disassembly, call stack, symbol, memory and register window outputs for future reference
- Symbolic source-level debugging for C and assembly languages
- Online Help
- Full-featured assembler and linker
- Interleaved source and disassembly
- Makefile generation
- Scripting language

Specifications

- Zilog Developer Studio (ZDS II)
- Configurable settings for:

CPU	Target
Compiler	Debugger
Assembler	Debug Tool
Linker	Workspace
Editor options (syntax coloring, tabs, foldings, etc.)	
- Debugger:
 - Source level, symbolic assembly debugging
 - eZ8 C-Compiler
- Supported Zilog processor family: Zilog's Z8 Encore!® Microcontrollers
- Host operating systems: Windows 2000, Windows XP Pro (32-bit), Windows Vista, Windows 7
- Supported hardware: Zilog Z8 Encore!® Microcontroller On-Chip Debugger devices

IDE Window Details

Zilog ZDS II features several windows that allow viewing various aspects of ZDS II tools while working with a project.

Project Workspace Window

The Project Workspace window displays project files. It uses a graphical hierarchy to display a project's source, dependency, and object files. This graphic tree displays separate folders for source, web, and dependency files. These files display in the Project Workspace window when the project is first opened.

Editor Window

The Editor window allows viewing and editing of the files in a project. It includes a chroma-code language-sensitive editor that displays instructions, comments, and assembler directives in different colors. It also includes built-in search capabilities. Double-click on an error message in the output window to find the line containing the error.

Output Windows

Use the Output windows to view comments and errors when building or compiling a project. Output windows contain tabs for Build, Debug and Find, Message, and Command Output functions. The Build tab displays the results of the assembler/linker process. The Debug tab displays error messages during the Debug and Build processes. The two Find files hold the results of the Find in Files Command, depending on which Find in Files tab the user chooses. The Messages Output window displays informational messages intended for the

user. The Command Output window displays all text messages and output generated by the ZDS II IDE as it executes commands.

The following windows output can be printed: Disassembly, call stack, symbol, memory and register. Just click on the window and press CTRL+P (or right-click on the window and select "print").

Status Bar

The Status Bar displays the current status of program execution. Status messages include STOP, STEP, and RUN. STOP mode halts the program. STEP mode allows Step Into, Step Over, and Step Out commands.

Debug Windows

ZDS II features several specialized windows that allow viewing information that helps to debug applications. These windows are only available when running in Debug mode. Select the Debug Windows submenu and the name of the Debug window or click on the appropriate Debug window's icon in the toolbar. Figure 2 illustrates the Debug Window.

Developers can use the following commands and tools:

- Go, Break, Reset, Step Into, Step Over, Step Out, and Set Next Instruction commands
- Run or Jump to Cursor commands
- Downloading code to Zilog emulators
- Inserting or removing breakpoints
- Enable/Disable (All) breakpoints

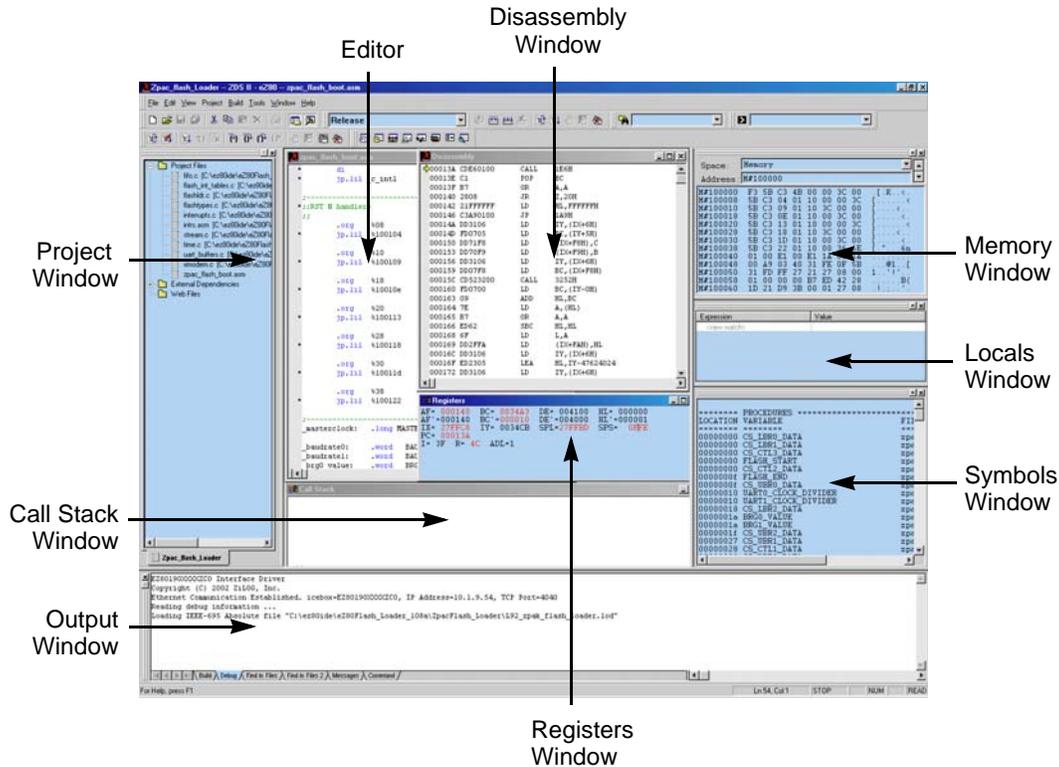


Figure 2. Debug Windows—Example

Table 1 lists the Debug windows and functions.

Table 1. Functions of the Debug Window

Debug Window	Function
Registers	Displays the contents of all registers.
Clock	Displays the number of states executed since the last reset.
Memory	Displays data located in program memory.
Watch	Displays all the variables and their values defined using the WATCH command. If the variable is not in scope, the variable is not displayed.
Locals	Displays all local variables that are currently in scope.
Call Stack	Displays the functions whose frames have been pushed onto the stack.
Symbols	Displays the address for each symbol in the program.
Disassembly	Displays the assembly code associated with the code displayed in the Code window. For each line in this window, the address location, the machine code, the assembly instruction and its operands display.
Special Function Registers	Displays the contents of the special function registers for the selected target.

eZ8 C-Compiler

Introduction

The eZ8 C-Compiler is an optimizing ANSI C-Compiler. The eZ8 C-Compiler meets ANSI's definition of a *freestanding implementation* (code can be developed for another machine; also, it is not assumed that operating system services will be available at run time), with the exception that *doubles* (object types) are 32 bits. In accordance with the definition of a freestanding implementation, the compiler accepts programs which confine the use of the features of the ANSI standard library to the contents of the standard headers: `<float.h>`,

`<limits.h>` and `<stddef.h>`. This release supports more of the standard library than is required of a freestanding implementation.

The compiler runs on the Windows 2000, Windows XP Pro (32-bit), Windows Vista or Windows 7 operating systems and can be invoked either through the command line or within the Zilog Developer Studio Integrated Development Environment. There are several language extensions supported in this version of the compiler, including interrupt functions and memory space accesses.

Figure 3 illustrates an example of the eZ8 C-Compiler development flow.

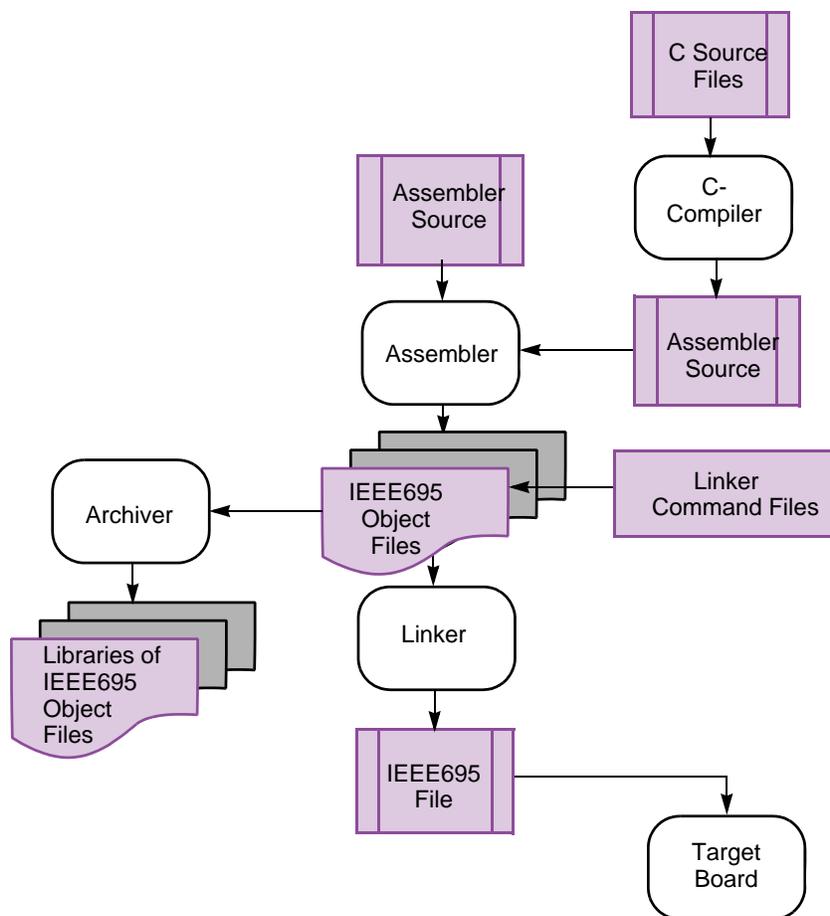


Figure 3. C-Compiler Development Flow

Software Features

- Highly optimized ANSI C-Compiler
- ANSI C run time library
 - String manipulation
 - Character classification and conversion
 - Buffer manipulation
 - Data conversion
 - Floating point and trigonometric functions
- Single precision IEEE floating point support
- Extensions for interrupt service routines
- Support for in-line assembly
- Intrinsic functions that are in-line expanded
- Copies initialized data from ROM to RAM
- The compiler, assembler, linker, and librarian can be used within ZDS or individually on the command line
- Optimizations:
 - Constant folding
 - Constant propagation
 - Copy propagation
 - Common sub-expression elimination

Supported Host System Configuration

- Windows 2000, Windows XP Pro(32 Bit), Windows Vista, Windows 7
- Pentium IV, 1.2GHz processor or higher

- 512MB RAM or more
- 50MB hard disk space or more
- Super VGA Video adapter
- CD-RM
- One or more RS-232 ports
- USB Port
- Ethernet Port

Supported Zilog Processor Families

Zilog Z8 Encore![®]

Technical Support

- <http://support.Zilog.com/>
- Toll-free telephone, U.S.: 1.877.945.6427
- International telephone: 512.306.4067
- Facsimile: 512.306.4072

Distribution Contents

- C-Compiler
- Assembler, linker and archive executables
- Header files
- Libraries
- Sample files and/or projects
- C-Compiler user manual in PDF format
- Release documentation with change logs
- ZDS configuration files

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

Zilog Worldwide Headquarters

1590 Buckeye Drive
Milpitas, CA 95035
Telephone: 408.513.1500
Fax: 408.365.8535
www.Zilog.com

Document Disclaimer

Zilog is a registered trademark of Zilog Inc. in the United States and in other countries. All other products and/or service names mentioned herein may be trademarks of the companies with which they are associated.

©2010 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. Zilog, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. Zilog ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval Zilog, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses or other rights are conveyed, implicitly or otherwise, by this document under any intellectual property rights.