

MCUEZQSG0508/D
FEBRUARY 1998

**MCUez MMDS or MMEVS
for HC05/08
ezSTART GUIDE**

Important Notice to Users

While every effort has been made to ensure the accuracy of all information in this document, Motorola assumes no liability to any party for any loss or damage caused by errors or omissions or by statements resulting from negligence, accident, or any other cause. Motorola further assumes no liability arising out of the application or use of any information, product, or system described herein; nor any liability for incidental or consequential damages arising from the use of this document. Motorola disclaims all warranties regarding the information contained herein, whether expressed, implied, or statutory, including implied warranties of merchantability or fitness for a particular purpose. Motorola makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use, or sell equipment constructed in accordance with this description.

Information contained in this document applies to REVision (1) MCUez.

The computer program contains material copyrighted by Motorola Inc., first published 1997, and may be used only under a license such as the License For Computer Programs (Article 14) contained in Motorola's Terms and Conditions of Sale, Rev. 1/79.

Trademarks

This document includes these trademarks:

MCUez is a trademark of Motorola Inc.

The MCUez development, emulation, and debugging application is based on Hi-Wave, a software technology developed by HIWARE. Hi-Wave is a registered trademark of HIWARE AG.

Adobe Acrobat is a trademark of Adobe Systems Incorporated.

Windows is a trademark of Microsoft Corporation.

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION	1-1
1.2 SYSTEM REQUIREMENTS	1-1
1.3 SOFTWARE COMPONENTS.....	1-1
1.4 MCUEZ DOCUMENTATION	1-2
1.5 INSTALLING MCUEZ.....	1-2

CHAPTER 2 USING MCUEZ

2.1 INTRODUCTION	2-1
2.2 LAUNCHING THE MCUEZ SHELL	2-1
2.3 USING MCUEZ SHELL CONFIGURATION	2-2
2.3.1 Configuration Dialog	2-2
2.3.2 Setting The Project Directory	2-2
2.3.3 Current Configuration Dialog	2-3
2.3.4 Adding A New Tool.....	2-4
2.3.5 Adding A New Project.....	2-6
2.4 USING THE MCUEZ ASSEMBLER.....	2-6
2.4.1 Choosing An Editor	2-6
2.4.2 Generating .abs Files From Absolute Files	2-7
2.4.3 Generating .o Files From Relocatable Files.....	2-9
2.4.4 Assembler Error Feedback From Editor	2-11
2.5 USING MCUEZ LINKER.....	2-11
2.5.1 Linking Code Segments.....	2-11
2.5.2 Linker Error Feedback From Editor.....	2-13
2.6 USING MCUEZ DEBUGGER	2-13
2.6.1 Starting The Target Hardware	2-13
2.6.2 Starting The Debugger.....	2-13
2.6.3 Setting Communication Device Specifications	2-15
2.6.4 Loading An Application And Using Commands From Run Menu	2-16
2.6.5 Debugging Features	2-17
2.6.5.1 Set Breakpoints	2-17
2.6.5.2 Drag and Drop	2-18
2.6.5.3 Command Window.....	2-19

2.6.5.4 Real-Time Data And Memory Window Update.	2-19
2.6.5.5 Bus Trace	2-21
2.6.5.6 Multiple Windows	2-24
2.6.5.7 Target Signals.	2-24
2.6.5.8 Changing Debugger Window Layout.	2-25
2.6.6 Reset.cmd/Startup.cmd	2-25

FIGURES

Figure 2-1. MCUez Shell Toolbar	2-1
Figure 2-2. Configuration Dialog	2-2
Figure 2-3. Current Configuration Dialog	2-3
Figure 2-4. Tools Tab.	2-4
Figure 2-5. New Tool Dialog	2-4
Figure 2-6. New Tool In Current Configuration	2-5
Figure 2-7. MCUez Shell Toolbar With New Tool Added	2-5
Figure 2-8. Project Directory Dialog	2-6
Figure 2-9. Selecting Advanced From Assembler Menu.	2-7
Figure 2-10. Output Tab Of Advanced Options Settings Dialog.	2-7
Figure 2-11. Selecting Assemble From File Menu	2-8
Figure 2-12. Select File To Assemble Dialog	2-8
Figure 2-13. Selecting Advanced From Assembler Menu.	2-9
Figure 2-14. Output Tab Of Advanced Options Settings Dialog.	2-9
Figure 2-15. Selecting Assemble From File Menu	2-10
Figure 2-16. Select File To Assemble Dialog	2-10
Figure 2-17. Selecting Advanced From Linker Menu.	2-11
Figure 2-18. Advanced Options Settings Dialog	2-12
Figure 2-19. Selecting Link From File Menu	2-12
Figure 2-20. Debugger Loading Firmware	2-13
Figure 2-21. Selecting Memory Map From MMDS0508 Menu	2-14
Figure 2-22. Memory Configuration Dialog	2-14
Figure 2-23. Selecting Communication From MMDS0508 Menu	2-15
Figure 2-24. Communication Device Specification Dialog.	2-15
Figure 2-25. Selecting Load From MMDS0508 Menu	2-16
Figure 2-26. Run Menu Options	2-16
Figure 2-27. Breakpoints Popup Window	2-17
Figure 2-28. Red Arrow Indicator In Instructions	2-18
Figure 2-29. Updating Memory Contents From Command Window	2-19
Figure 2-30. Selecting Periodical For Data Window.	2-20
Figure 2-31. Selecting Periodical For Memory Window.	2-20
Figure 2-32. Selecting Arm Analyzer From Trace Menu	2-21



Figure 2-33. Selecting Setup... From Trace Menu.....	2-21
Figure 2-34. Bus Analyzer Configuration Dialog.....	2-22
Figure 2-35. Sequencer Tab Section.....	2-22
Figure 2-36. Bus Analyzer Information In Trace Window.....	2-23
Figure 2-37. Graphical View Of Bus Analyzer Information.....	2-23
Figure 2-38. Zoom In Of Graphical View.....	2-23
Figure 2-39. Multiple Memory Windows.....	2-24
Figure 2-40. Target Signals Dialog.....	2-24
Figure 2-41. Layout Options - Store And Load.....	2-25



CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

This ezStart Guide will help you quickly install the MCUez package and develop/debug your HC05/08 applications through MCUez.

The MCuez package includes software and documentation.

1.2 SYSTEM REQUIREMENTS

The MCUez software package requires:

- PC with an 80486 or better microprocessor
- At least 16MB of RAM (recommended)
- CD-ROM disk drive
- Hard disk with 18 MB of available storage
- Microsoft Windows (WIN95 or WINNT)

1.3 SOFTWARE COMPONENTS

The MCUez software package contains:

- ez Configuration: Allows you to configure your project with different editors and setup environment variables.
- Motpad Editor: Default editor for working with the assembler and linker.
- ez Assembler: Supports Absolute and Relocatable assembly. The Assembler is ELF/DWARF 2.0 object format compliant and works with P&E HC05 assembly codes from “P&E converter to MCUez assembly”.
- ez Linker: Smart linker that allows maximum memory usage by your applications. The Linker is ELF/DWARF 2.0 debugging information compliant.
- ez Debugger: Supports assembly source-level debugging. Future releases will support C source-level debugging for any compiler that is ELF/DWARF 2.0 compliant.
- On-Line help: Available for each software component.

1.4 MCUEZ DOCUMENTATION

The MCUEz documentation is available on-line as Adobe Acrobat (“.pdf”) files. The MCUEz installation process installs the documentation files along with the MCUEz software. The documentation includes this ezStart Guide to help you install and use the software, and a manual for each of the software components.

1.5 INSTALLING MCUEZ

To correctly install the MCUEz software package:

1. Insert the MCUEz CD-ROM in the appropriate drive.
2. Access the Start/Run menu from the task bar.
3. Run SETUP . EXE from the CD-ROM drive.
4. Follow the steps presented by the installation script (be sure to select the MCUEz for the MCU family you intend to use).
5. If you intend to use the P&E Assembler Converter, copy “peconv . exe” and “pecvtip . txt” from the 3.5” disk labelled “Converter from P&E to MCUEz Assembly”. Put “peconv . exe” and “pecvtip . txt” in the \prog subdirectory of the directory created when you installed MCUEz.

CHAPTER 2

USING MCUEZ

2.1 INTRODUCTION

This chapter teaches you how to use the MCUez Shell Configuration, Assembler, Linker, and Debugger by guiding you through examples. The MCUez installation process includes the creation of demonstration files. For the examples in this chapter, you will use two example assembly files:

- fiboorg.asm - an absolute assembly file
- fibo.asm - a relocatable assembly file

The examples are stored in the ...\\demo\\wmmds05a subdirectory of the directory created when you installed MCUez. In the examples in this chapter, the installation directory is C:\\MCUez\\MCUez05. If you installed MCUez to a different location, substitute that location.

2.2 LAUNCHING THE MCUEZ SHELL

You can launch the MCUez Shell with either of two methods:

- Double-click the MCUez Shell icon:
- Select MCUez Shell from the Start/Programs menu.

When you complete one of these methods, the MCUez Shell Toolbar appears:



Figure 2-1. MCUez Shell Toolbar

2.3 USING MCUEZ SHELL CONFIGURATION

MCUez Tools search for project information in the default directory determined in the Shell Configuration. Because you can determine the installation location of the MCUez package and you may wish to install different families of the MCUez development tools at different times, the project directory defaults to the location of the last development to be selected or installed. For example, if the HC08 family of MCUez development tools is installed last, the default project directory will be set to `..\MCUez08\Demo\Wmmds08a`. In that case, if you needed to use MCUez HC05 applications, you would have to switch the default project directory to `..\MCUez05\Demo\Wmmds05a`. To switch default directories, follow the example in Setting The Project Directory section.

2.3.1 Configuration Dialog

The Configuration dialog helps you set up a new project or update an existing project and controls the environment settings. The Project Directory contains `project.ini` and `default.env` the files used to store the settings you assign with the Configuration dialog.

2.3.2 Setting The Project Directory

In this example, you will set the Project Directory:

1. Activate the MCUez Shell.
2. Select the MCUez icon (the first button from the left) on the Shell Toolbar.
3. The Configuration dialog opens:

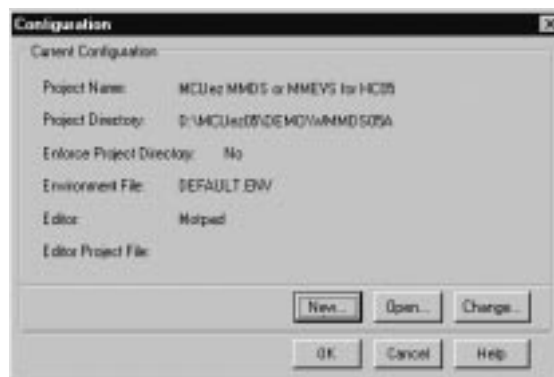


Figure 2-2. Configuration Dialog

4. The Project Directory should be set to the desired directory (in this case, `C:\MCUez\MCUez05\demo\wmmds05a`).
5. If you need to open another Project Directory that uses the same shell:
 - a. Select the Open button.
 - b. The Select Directory dialog opens.
 - c. Choose the correct Project Directory.
 - d. Select the OK button.

6. If you need to open a project that was not created by the MCUEz Shell, you can:
 - Copy project.ini and default.env from the C:\MCUEz\MCUEz05\demo\wmmnds... directory appropriate to your project.
 - Create your project directory using the shell configuration from the default setting environment (the default setting environment is C:\MCUEz\MCUEz05\demo\wmmnds05a or C:\MCUEz\MCUEz05\demo\wmmnds08a).
7. If you need to create a new Project Directory:
 - a. Select the New... button.
 - b. The Project Directory dialog opens.
 - c. Type the correct path for the Project Directory or choose the Browse button to manually search for the correct Project Directory.
 - d. Select the OK button.

2.3.3 Current Configuration Dialog

The Current Configuration dialog allows you to choose different tab sections that allow you to change the project name and select different editors, tools, paths, and environment variables.

To open the Current Configuration dialog:

1. Activate the MCUEz Shell.
2. Select the MCUEz icon from the Toolbar.
3. The Configuration dialog opens.
4. Select the Change... button.
5. The Current Configuration dialog opens:

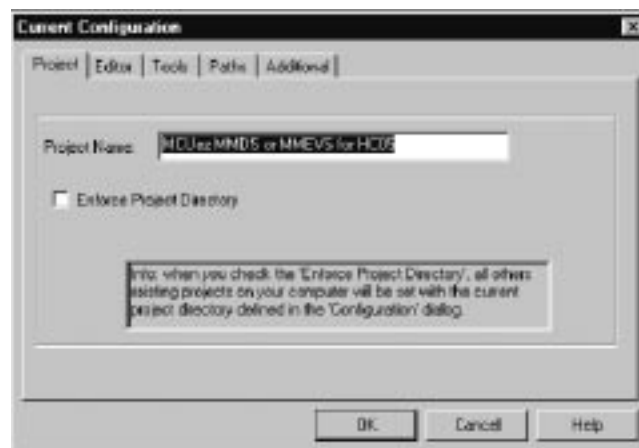


Figure 2-3. Current Configuration Dialog

6. Use the tabs to view and edit the configuration settings.

2.3.4 Adding A New Tool

This section demonstrates how the Current Configuration dialog can be used to change the configuration. In this example, you use the Current Configuration dialog to add a new tool (the P&E Assembler Converter) to the MCUEz Toolbar (the example assumes you have installed the P&E Assembler Converter package. If you have not, the example will not function):

1. Activate the MCUEz Shell
2. Open the Configuration dialog.
3. Select the Tools tab.
4. The Tools tab section appears:

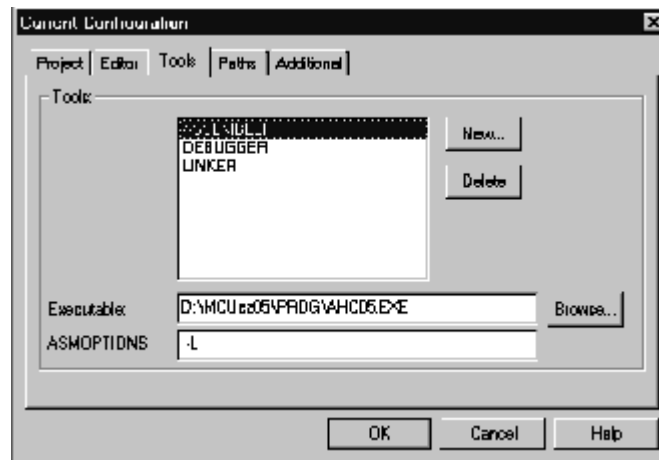


Figure 2-4. Tools Tab

5. Select New Button.
6. The New Tool dialog appears:

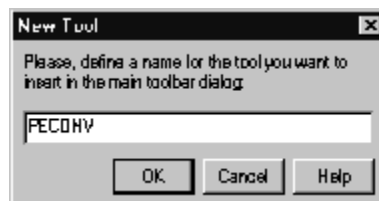


Figure 2-5. New Tool Dialog

7. Type the name of the desired tool in the dialog box. (For this example, use “peconv”).

8. Select the OK button.
9. The new application appears in the Tools tab section:

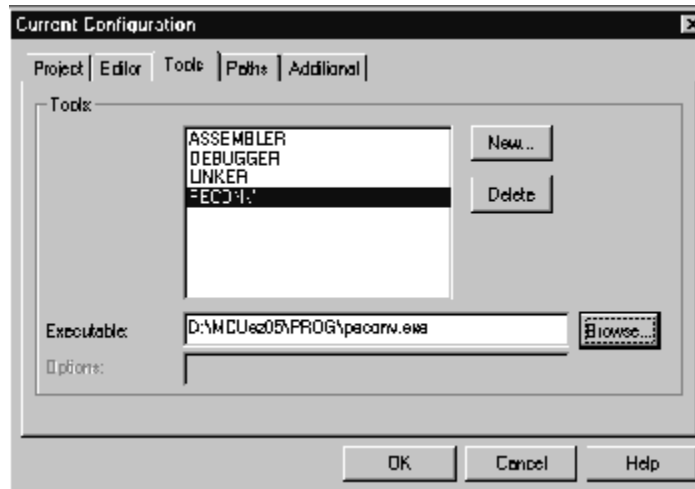


Figure 2-6. New Tool In Current Configuration

10. Select the Browse button.
11. A standard Browse dialog appears.
12. Locate and choose the executable file associated with the new application (in the example, peconv.exe from C:\MCUez05\PROG).
13. Select the OK button in the Browse dialog.
14. Select the OK button in the Tools tab section.
15. Select the OK button in the Configuration dialog.
16. The project.ini and default.env files are updated and saved.
17. The new tool appears in the MCUez Shell Toolbar:



Figure 2-7. MCUez Shell Toolbar With New Tool Added

2.3.5 Adding A New Project

Follow these steps to create a new project:

1. Activate the MCUEz Shell
2. Open the Configuration dialog.
3. Select the New button.
4. The Project Directory dialog appears:

:

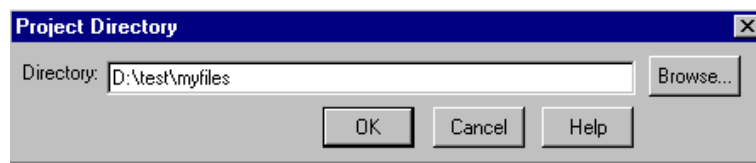


Figure 2-8. Project Directory Dialog

5. Specify the new project path.
6. Select OK.
7. The New Configuration dialog appears.
8. Use the New Configuration dialog to set the editor, tools, paths, and other environmental variables. If you create a new project with the New Configuration dialog, the new project will keep the previous project's settings for editor, tools, and environmental variables. If the previous project is MCUEz default `..\..\demo\wmmdsxa`, the new project will keep default configuration settings from the demonstration project.

2.4 USING THE MCUEZ ASSEMBLER

This sections explains how to use the MCUEz Assembler and its functions.

2.4.1 Choosing An Editor

You can select the default editor from the Editor Tab in the Current Configuration dialog. The default editor is originally Motpad. Motpad has better features than Notepad or Wordpad and should be used if you do not use Winedit or Codewright.

2.4.2 Generating .abs Files From Absolute Files

Follow these step to create an .abs file from absolute assembly code (you can skip steps 4-10 if the absolute file is already assembled):

1. Activate the MCUEz Shell.
2. Select the ezASM icon.
3. The Assembler dialog opens.
4. Select Advanced from the Assembler pull-down menu:

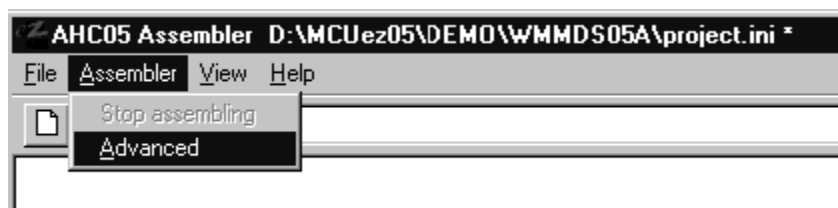


Figure 2-9. Selecting Advanced From Assembler Menu

5. The Advanced Options Settings Dialog opens.
6. Select the Output tab.
7. The Output tab section appears:

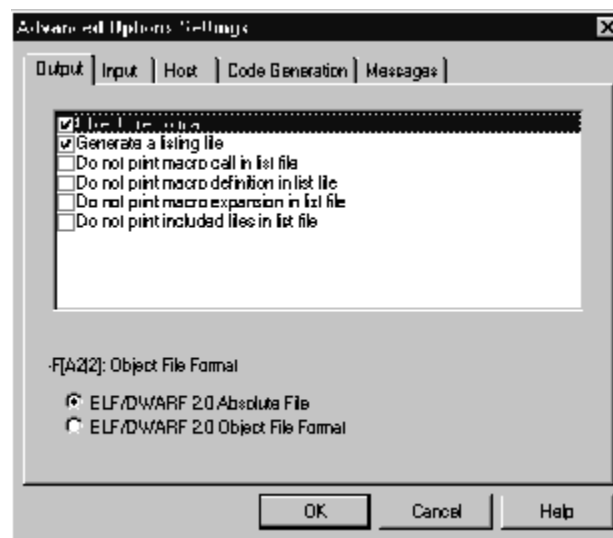


Figure 2-10. Output Tab Of Advanced Options Settings Dialog

8. Choose the Object File Format box (for this example, the Generate A Listing File box and the ELF/DWARF 2.0 Absolute File radio button are checked as the default setting).
9. Select the OK button.
10. The Advanced Options Settings dialog closes.
11. Select Assemble from the File pull-down menu.

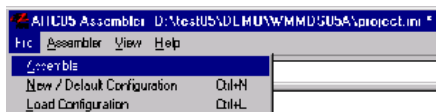


Figure 2-11. Selecting Assemble From File Menu

12. The Select File To Assemble dialog appears:

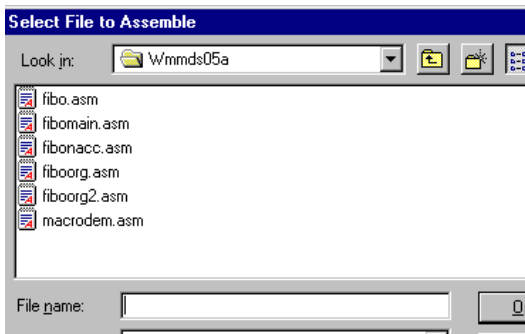


Figure 2-12. Select File To Assemble Dialog

13. Choose fiboorg.asm (an absolute file).
14. Select the OK button.
15. MCUEz Assembler generates four output files:
 - An S record file (.sx)
 - A debugging source file (.dbg)
 - A list file (.lst)
 - An absolute file with debugging information (.abs)

You can also assemble a file by dragging it from Windows Explorer and dropping it in the Assembler window. If you use the “Drag and Drop” method, you can only assemble one file at a time. To assemble multiple files at the same time, use the Assembler window command line. In the command line, type the names of the files you would like to assemble and press Enter.

2.4.3 Generating .o Files From Relocatable Files

Follow these steps to create an .o file from relocatable assembly code:

1. Activate the MCUEz Shell.
2. Select the ezASM icon.
3. The Assembler dialog opens.
4. Select Advanced from the Assembler pull-down menu:

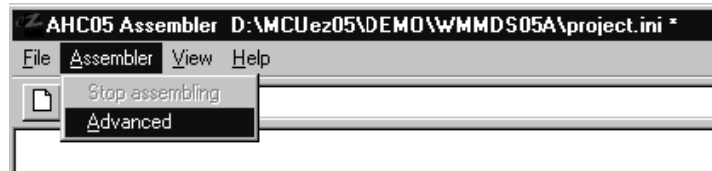


Figure 2-13. Selecting Advanced From Assembler Menu

5. The Advanced Options Settings Dialog opens.
6. Select the Output tab.
7. The Output tab section appears:

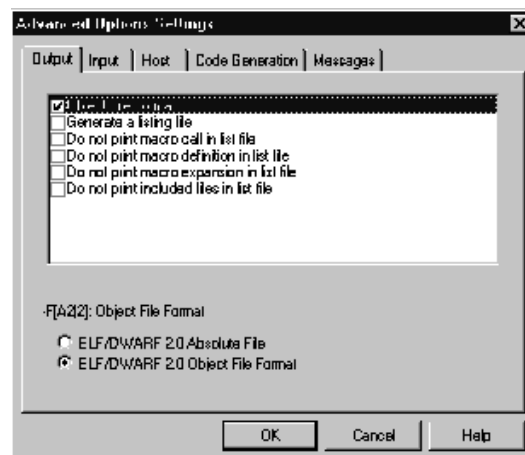


Figure 2-14. Output Tab Of Advanced Options Settings Dialog

8. Choose the Object File Format box.
9. Choose the ELF/DWARF 2.0 Object File Format radio button.
10. Select the OK button.
11. The Advanced Options Settings dialog closes.
12. Select Assemble from the File pull-down menu.

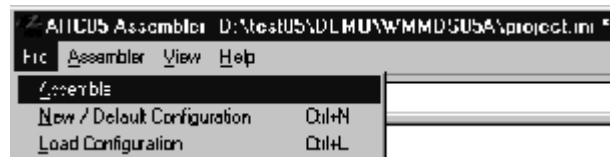


Figure 2-15. Selecting Assemble From File Menu

13. The Select File To Assemble dialog appears:

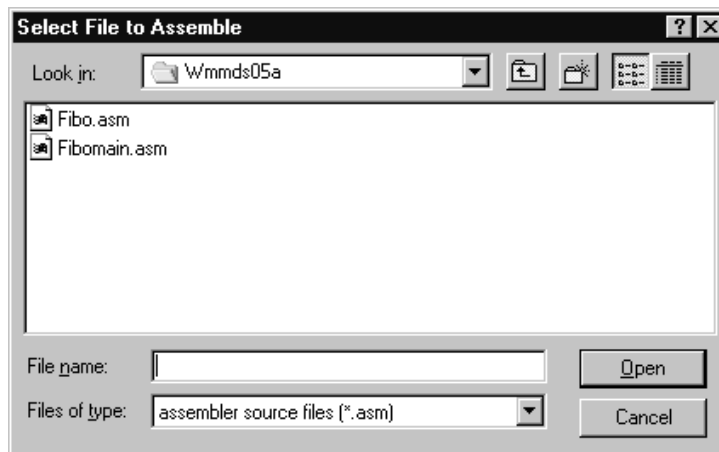


Figure 2-16. Select File To Assemble Dialog

14. Choose fibo.asm (a relocatable file).
15. Select the OK button.
16. MCUEZ Assembler generates an object file (.o).\

2.4.4 Assembler Error Feedback From Editor

If you are using Motpad (the editor supplied with the MCUEz package) and there are errors found in the assembly codes after assembling, you can click an error message from the assembly window, and the editor will automatically open the file and indicate the error in the code. Notepad and Wordpad do not support in-line searching, so the error will not be indicated.

2.5 USING MCUEZ LINKER

This section explains the MCUEz Linker.

2.5.1 Linking Code Segments

This example teaches you how to use the MCUEz Linker to link code segments:

1. Create/edit a linker file using Motpad editor. Linker files (.prm) define the Segments and Placement of the code to be linked. Segments specify memory types and location. Placement indicates where to place the codes and data in the memory Segments. Use fibo.prm in this example:

```
LINK fibo.abs
NAMES fibo.o END

SEGMENTS
  Z_RAM = READ_WRITE 0x0080 TO 0x00EF;
  MY_ROM = READ_ONLY 0x0B00 TO 0x0BFF;
END
PLACEMENT
  .data INTO Z_RAM;
  .text INTO MY_ROM;
```

2. Activate the MCUEz Shell
3. Select the ezLink icon from the Toolbar.
4. Select Advanced from the Linker pull-down menu:

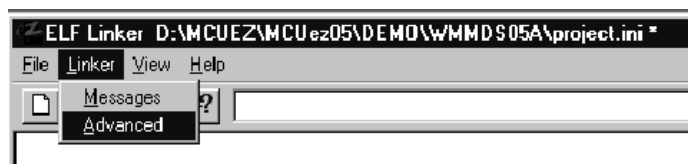


Figure 2-17. Selecting Advanced From Linker Menu

5. The Advanced Options Settings dialog appears:

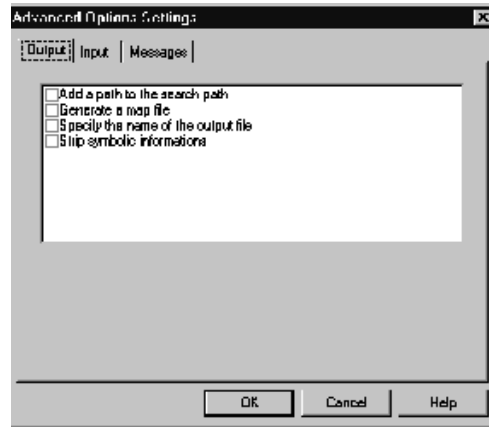


Figure 2-18. Advanced Options Settings Dialog

6. Choose appropriate Output, Input, and Message settings (use default values for example).
7. Select the OK button.
8. The Advanced Options Settings dialog closes.
9. Select Link from the File pull-down menu:



Figure 2-19. Selecting Link From File Menu

10. MCUEz Linker generates three files:
 - Motorola S record (.sx)
 - Absolute file with debugging information (.abs)
 - Object code memory match (.map)

You can also use the Linker by dragging a file from Windows Explorer and dropping it in the linker window.

The Linker command line can also be used to link files. In the command line, type the addresses of the files you would like to link.

2.5.2 Linker Error Feedback From Editor

If you are using Motpad (the editor supplied with the MCUEz package) and there are errors found in the link file after linking, you can click an error message from the Linker window, and the editor will automatically open the file and indicate the error in the code. Notepad and Wordpad do not support in-line searching, so the error will not be indicated.

2.6 USING MCUEZ DEBUGGER

This section teaches you how to use the MCUEz Debugger and its features. To use the Debugger, you must start the target hardware and the Debugger.

2.6.1 Starting The Target Hardware

To start the target hardware:

1. Set M68HC05B32 Emulation Board to MMDS05.
2. Connect Host and MMDS05 with serial cable to Comport.
3. Turn power on.

2.6.2 Starting The Debugger

To start the Debugger:

1. Activate the MCUEz Shell (be sure the target hardware is on).
2. Select the ezDebug icon.
3. The Debugger dialog opens and the Debugger automatically loads firmware:

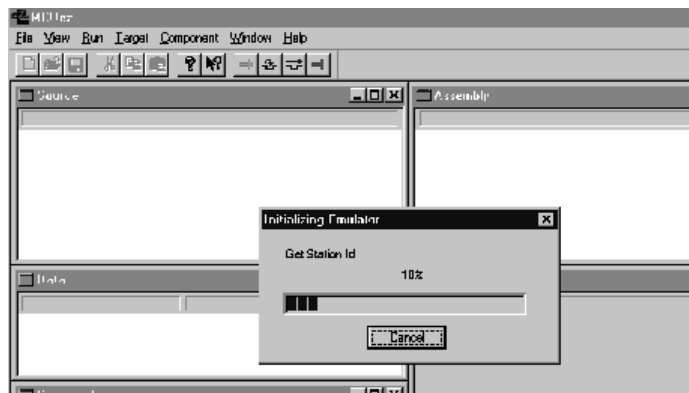


Figure 2-20. Debugger Loading Firmware

4. The Debugger automatically loads a MEM file to configure the device memory map on the MMDS. If the Debugger does not locate a MEM file, a dialog opens and prompts you to insert the MEM file. Some emulator boards support multiple derivatives with the same MCU ID number, but different MEM files. In this case, the Debugger will automatically load a default MEM file to represent the multiple derivatives when the Debugger starts.
5. After the MEM file is successfully loaded, select Memory Map from the MMDS0508 pull-down menu:

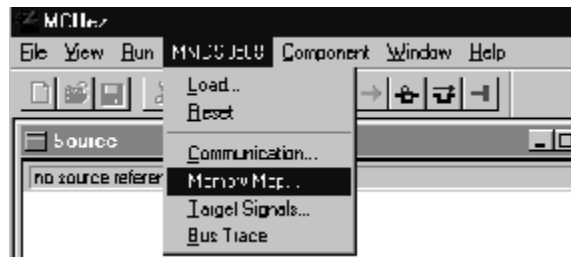


Figure 2-21. Selecting Memory Map From MMDS0508 Menu

6. The Memory Configuration dialog appears:

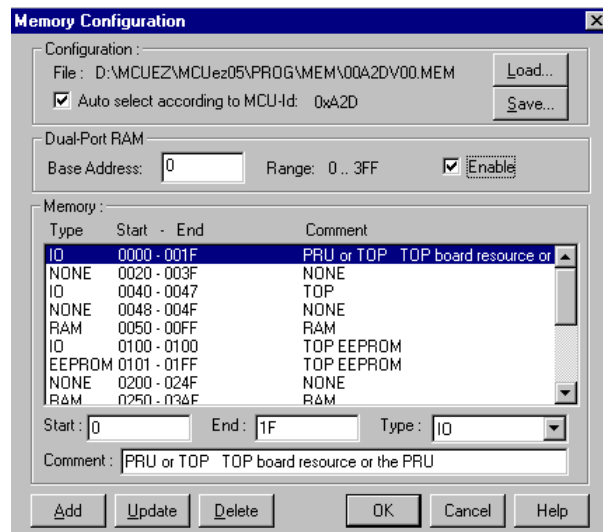


Figure 2-22. Memory Configuration Dialog

7. Load a specific MEM file or modify the memory map directly.
8. To save the new configuration, select the Save... button. Save changes to a name other than the name of the original MEM file.
9. Select the OK button.

2.6.3 Setting Communication Device Specifications

This section shows you how to use the Communication Device Specification dialog to control the baudrate, comport, and Show Protocol command:

1. Choose Communication from the MMDS0508 pull-down menu:

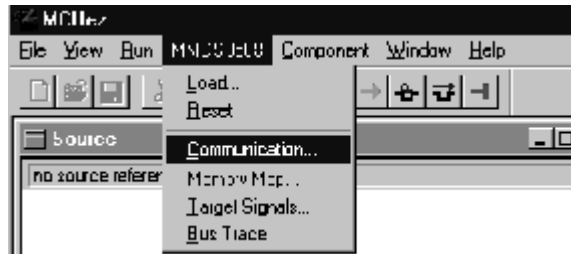


Figure 2-23. Selecting Communication From MMDS0508 Menu

2. The Communication Device Specification dialog opens:



Figure 2-24. Communication Device Specification Dialog

3. Use the Communication Device Specification dialog to set the baudrate, communication device (comport), and Show Protocol command. The Show Protocol command displays asynchronous messages that return from the target hardware when the debugging interface is issued. Show Protocol should normally be turned off (unchecked).
4. Select the OK button to implement your changes.

2.6.4 Loading An Application And Using Commands From Run Menu

This example teaches how to load an application and use commands from the Run menu:

1. Activate the MCUEz Shell.
2. Select the ezDebug icon.
3. Choose Load... from the MMDS0508 pull-down menu:

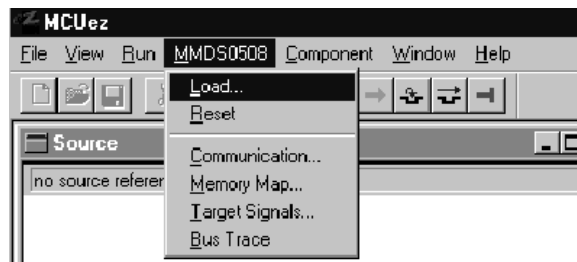


Figure 2-25. Selecting Load From MMDS0508 Menu

4. Load application into hardware (for this example, load fibo.abs from C:\MCUEz\MCUEz05\demo\wmmds05a).
5. Use the Run pull-down menu commands (Start/Continue, Halt, Assembly Step, and Assembly Step Over) to control code processing. Run icons can also control processing:

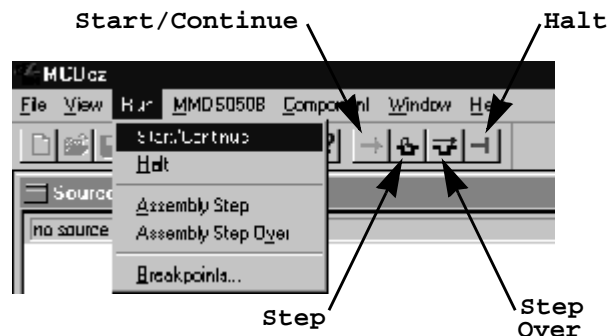


Figure 2-26. Run Menu Options

NOTE

The memory map for fibo.abs is defined as 0x0080 TO 0x00EF (RAM) and 0x0B00 TO 0x0BFF (ROM). Check the Memory Configuration dialog to ensure your current memory map matches fibo.abs. If not, change fibo.prm (see the Using MCUEz Linker section) to allocate different RAM and ROM locations according to the emulated device memory configuration from the Memory Configuration dialog.

2.6.5 Debugging Features

To make the debugging process easier, the MCUEz Debugger offers these features/options:

- Set Breakpoint
- Drag and Drop
- Command Window
- Real-Time Data and Memory Window Update
- Bus Trace (with MMDS)
- Multiple Windows
- Target Signals

2.6.5.1 Set Breakpoints

To set breakpoints:

1. Load an application.
2. Move the mouse arrow to the Assembly or Source window.
3. Click the right mouse button.
4. The breakpoints popup window opens:

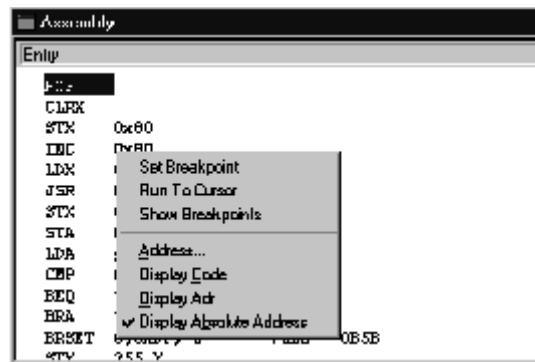


Figure 2-27. Breakpoints Popup Window

5. Select Set Breakpoint to set a breakpoint at the point indicated by the red arrow:

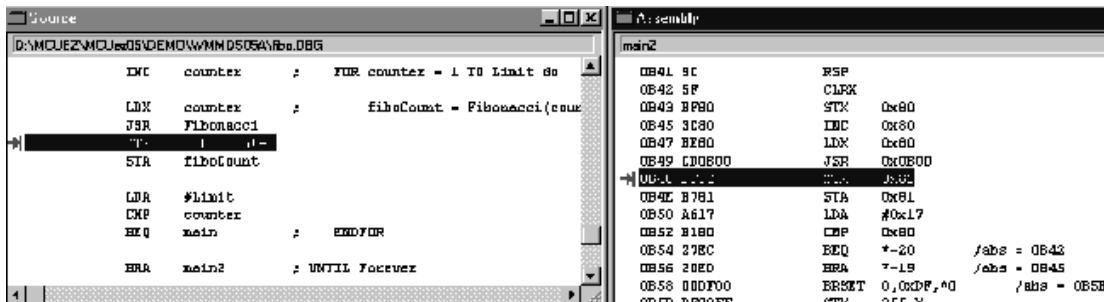


Figure 2-28. Red Arrow Indicator In Instructions

6. Run the application.
7. The application stops at the breakpoint.

The breakpoint popup window can also be used to display code and addresses. To display code, follow the steps to set a breakpoint but select Display Code and Address in the popup. To delete a breakpoint, move the mouse arrow to the breakpoint in the instructions and click the right mouse button.

2.6.5.2 Drag and Drop

The Debugger provides a powerful “Drag and Drop” utility to view different debugging information. The Drag and Drop feature works much like the Windows standard. For example, to see the opcode and address for an instruction or line of code:

1. Locate the instruction you would like to view.
2. Highlight the instruction.
3. Drag the selected instruction to the Memory window.
4. Drop the instruction in the Memory window.
5. The opcode and address for the instruction appear in the Memory window.

You can drag memory content from the Memory window to the Assembly window to view instructions. You can also drag a variable from the Data window to the Memory window to view its address and contents. Some drag and drop combinations are *not* valid:

- From a Source window to a Memory window
- From a Memory window to a Source window
- From a Memory window to a Data window
- From a Memory window to a Register window
- From a Module window to a Command window
- From a Command window to a Module window
- From a Module window to a Register window

2.6.5.3 Command Window

Most of the Debugger features available through the graphic user interface (GUI) can be performed with commands in a Command window. MCUEz Debugger uses two major sets of debugging commands:

- MCUEz engine commands
- Hardware specific commands

The Bus Analyzer is an example of a hardware specific command. To view available commands, type “help” from the in> prompt in a Command window.

Type “wb 0x80..0x85 0x11 0x22 0x33 0x44 0x55 0x66” at the Command prompt to modify bytes from 0x80..0x85. Memory contents from 0x80 to 0x85 are updated:



Figure 2-29. Updating Memory Contents From Command Window

You can implement several Debugger commands at once by creating a script file (e.g. test.cf) and running it with the command, “cf test.cf”.

The Command window also allows you to view I/O registers and memory maps in a specific manner. For example, the command “rd EE” displays I/O registers for EEPROM. The names and contents of the EE registers appear in the Command window. To see all the definition registers in terms of register name and content, type “rd *” at the prompt.

2.6.5.4 Real-Time Data And Memory Window Update

The Real-Time Data and Memory Window Update features allow you to view variable changes in real-time when the variables are mapped to Dual Port memory locations. To use the Real-Time feature:

1. Move the mouse arrow to a Data window.
2. Press the right mouse button.
3. The real-time popup window appears.

4. Select Periodical from the Mode submenu:

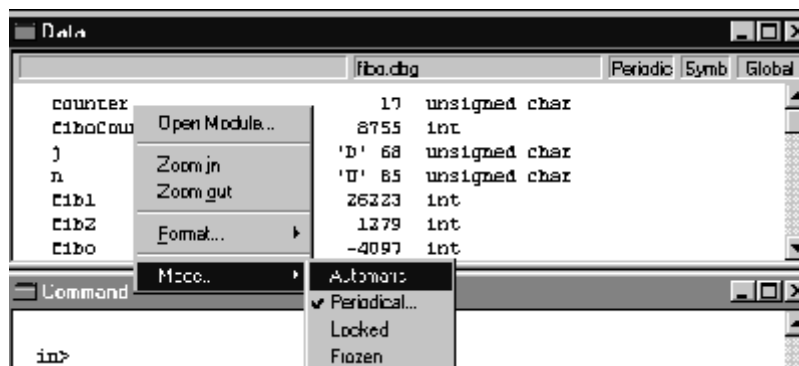


Figure 2-30. Selecting Periodical For Data Window

5. The Update Rate dialog appears.
6. Choose the appropriate update frequency.
7. Select the OK button.
8. The Update Rate dialog closes.
9. Select the Start/Continue icon.
10. As the application runs, the variables changes at the update rate you determined.

You can also view variable content changes in real-time from the memory window. To view real-time variable changes in the memory window:

1. Move the mouse arrow to a Memory window and press the right mouse button.
2. The real-time popup window appears.
3. Select Periodical from the Mode submenu:

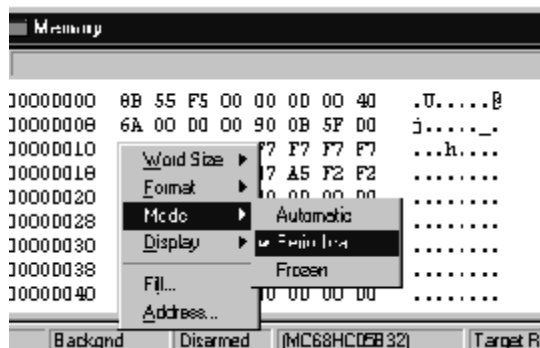


Figure 2-31. Selecting Periodical For Memory Window

4. The Update Rate dialog appears.
5. Choose the appropriate update frequency.
6. Select the OK button.
7. The Update Rate dialog closes.
8. Select the Start/Continue icon.
9. As the application runs, the variables changes at the update rate you determined.

2.6.5.5 Bus Trace

The Bus Trace feature can only be used with the MMDS. Bus Trace collects bus transaction information in terms of instruction access abilities (R/W) and bus cycle. MCUEz Debugger provides different options for viewing and setting the bus analyzer from the Trace menu:

1. Select Arm Analyzer from the Trace menu:



Figure 2-32. Selecting Arm Analyzer From Trace Menu

2. Select Setup from the Trace menu:

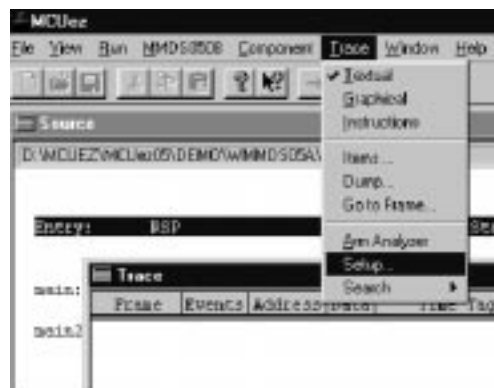


Figure 2-33. Selecting Setup... From Trace Menu

- The Bus Analyzer Configuration dialog opens:

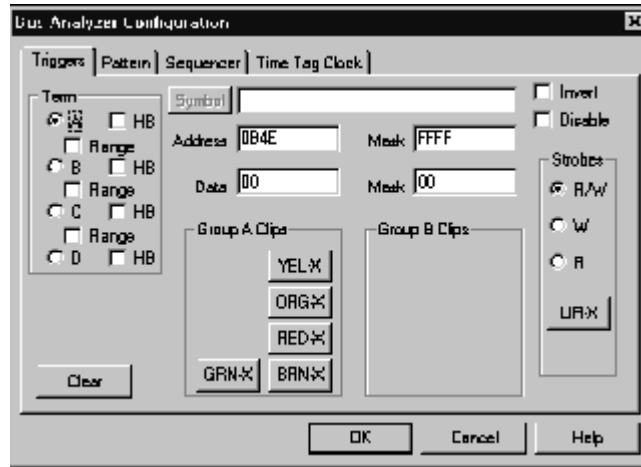


Figure 2-34. Bus Analyzer Configuration Dialog

- Set Address to 0B4E, Mask to FFFF, and uncheck the Disable box.
- Select the Sequencer tab.
- The Sequencer tab section appears:

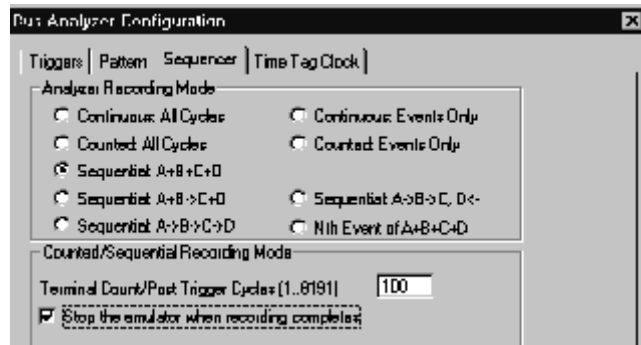


Figure 2-35. Sequencer Tab Section

- Set Analyzer Recording Mode to Sequential A+B+C+D, Terminal Count/Post Trigger Cycles to 100, and check the “Stop the emulator when recording complete” box.
- Select the OK button.
- The Bus Analyzer Configuration dialog closes.
- Select the Start/Continue icon.
- The analyzer trigger will stop the application and collect address, data, time tag, instructions for each bus cycle:

Frame	Events	Address	Data	Time Tag	INSTRUCTION	R/W	A-Gry	A-Ylt	A-Blu	A-Gzn	A-Yel
92	----	0082	FF	93.00 us		R	0	0	0	0	0
93	----	0082	01	94.00 us		w	0	0	0	0	0
94	A---	0B4E	E7	95.00 us	STA 0x81	R	0	0	0	0	0
95	----	0B4F	81	96.00 us	*	R	0	0	0	0	0
96	----	0081	FE	97.00 us		R	0	0	0	0	0
97	----	0081	00	98.00 us		w	0	0	0	0	0
98	----	0B50	A6	99.00 us	LDA #0x17	R	0	0	0	0	0

Figure 2-36. Bus Analyzer Information In Trace Window

To view the bus analyzer information in Graphical mode:

1. Select Graphical option from the Trace menu or popup window (activate the popup window with a right mouse button click in a Trace window):

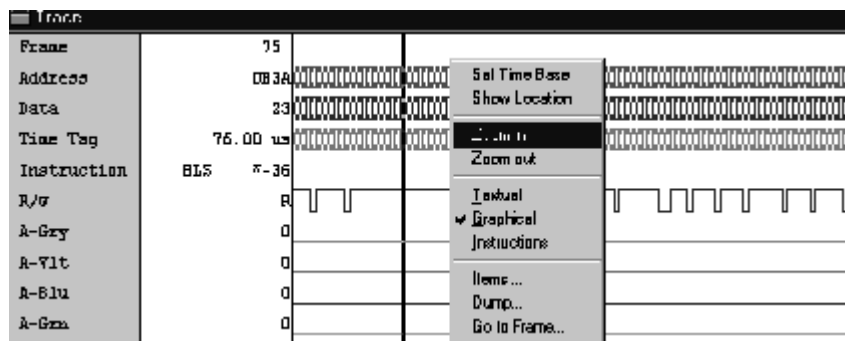


Figure 2-37. Graphical View Of Bus Analyzer Information

2. To view detailed content of each bus cycle, select Zoom In from the Trace popup window:

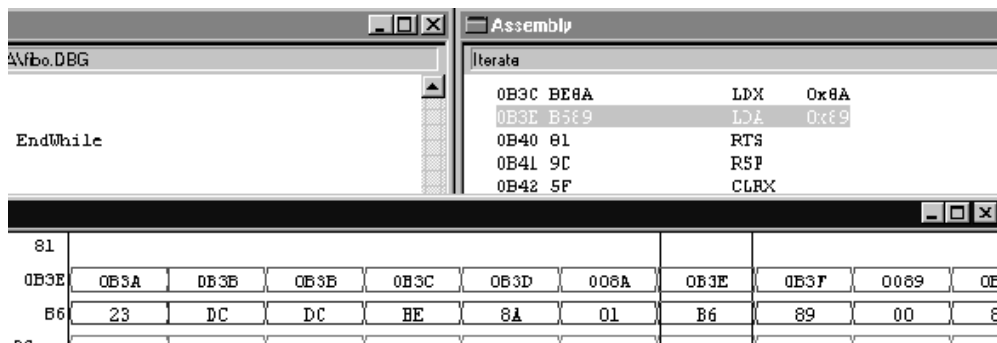


Figure 2-38. Zoom In Of Graphical View

2.6.5.6 Multiple Windows

MCUez Debugger allows multiple instances of windows. To open a second Memory window, select Open from the Component pull-down menu. A second Memory window, labeled Memory:2, appears:

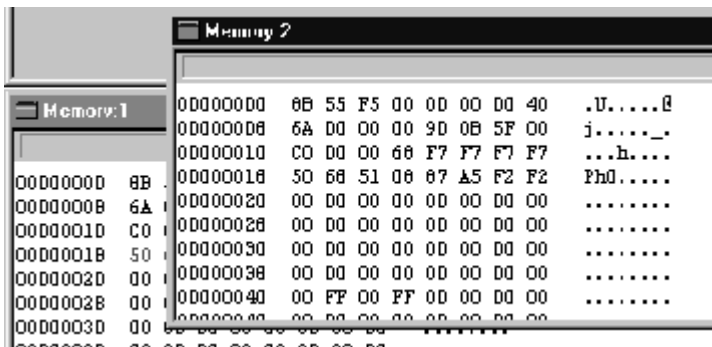


Figure 2-39. Multiple Memory Windows

2.6.5.7 Target Signals

The Target Signals dialog allows you to set target signals to “Reset in” (reset the emulating MCU) or “Reset out” (reset the emulating MCU and the target system). To open the Target Signals dialog, select Target Signals from the MMDS0508 pull-down menu:



Figure 2-40. Target Signals Dialog

2.6.5.8 Changing Debugger Window Layout

To reconfigure MCUEz Debugger layout, do one of the following:

- Modify “Layout Out” entry from *Project.ini* in C:\MCUEz05\Demo\Wmmds05a.
- Select Open from the Component pull-down menu and select a new component. Choose Store from the Layout options of the Window pull-down menu to save the entire layout to a *.hwl file. To open a *.hwl file, choose Load from the Layout options:

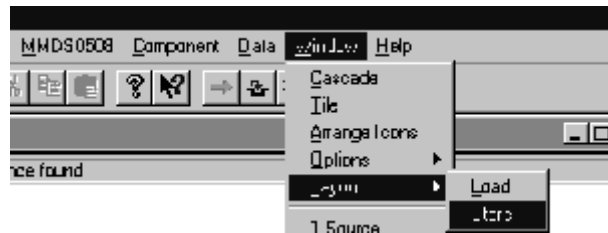


Figure 2-41. Layout Options - Store And Load

2.6.6 Reset.cmd/Startup.cmd

MCUEz provides preprocessing files that allow you to initialize the reset status of the target system. You can create a “reset.cmd” or “startup.cmd” text file that contains commands to initialize registers before you execute the code. If you enter the commands in “reset.cmd”, each time Reset is selected from the menu, the debugger will execute the reset.cmd file and perform an initialization. If you enter the commands in “startup.cmd”, the target will be initialized by the startup.cmd file each time the debugger application launches.

The reset.cmd and startup.cmd files need to be placed into your Working Directory.

