

CYDBä

CY3654 Development Kit

User's Guide



CYPRESS

Version 1.00

CYDB User's Guide:

©1998-2000 Cypress Semiconductor Corporation. All Rights Reserved

CYDB Software:

©1998-2000 Cypress Semiconductor Corporation. All Rights Reserved

Microsoft Windows 98:

©1998 Microsoft Corporation. All Rights Reserved

Printing Revision: 1.00

The use and copying of the CYDB User's Guide and Software are subject to the License Agreement included with the software. Any other use is prohibited.

CYDB™ is a trademark of Cypress Semiconductor Corporation.
Windows® 98® is a registered trademark of Microsoft Corporation.
Windows NT™ is a registered trademark of Microsoft Corporation.

Contents

Title Page.....	i
Copyright Page.....	iii
Contents	iv
List of Figures	viii
Conventions	1
Document Conventions	3
Manual Organization.....	4
Chapter 1 Introduction.....	5
1.1 Overview	7
1.2 Getting Started.....	7
1.2.1 System Requirements	7
1.2.2 Help Sources.....	8
1.2.3 Technical Support	8
1.2.4 Features Overview	9
1.3 Installation.....	10
1.3.1 Hardware Installation.....	10
Physical Description.....	10
Functional Description	13
Installation Process.....	14
1.3.2 Software Installation	15
Physical Description.....	15
Functional Description	16
Installation Process.....	16
1.3.3 Installation Verification	17
Launching the Program for the First Time	17
Default Configuration – What Do You See?.....	18

Chapter 2	Operations.....	19
2.1	Overview	21
2.1.1	How the Hardware and Software Work Together	21
2.2	Connect to Target	22
2.3	Configuration.....	23
2.3.1	Target Chip Configuration – Loading Personality	23
2.3.2	Emulator Configuration – Entering Stand-alone Mode	26
2.4	Debugger Operation.....	28
2.4.1	How to View/Modify/Refresh ...	28
	Data Memory Window.....	28
	Program Memory Window.....	30
	CPU Registers Window.....	31
	IO Registers Window	32
	Memory FIFO Window	34
	Breakpoints Window	35
2.4.2	Application Operation	37
	Downloading an Application.....	37
	Running an Application	39
	Resetting an Application ...	41
	Stepping.....	41
	Working with Breakpoints.....	42
	Stopping.....	42
2.5	Customizing Your Workspace	43
2.5.1	Customizing Toolbars and Commands.....	43

	Toolbars.....	44
	Commands.....	47
2.5.2	Saving and Opening a Workspace	49
	Save Your Workspace	49
	Open a Workspace	51
	Workspace Options.....	54
Chapter 3	Appendices	55
	Appendix A Acronyms and Abbreviations....	57
	Appendix B Index.....	59
	Appendix C Detailed Hardware Description.	61
	Platform Board	61

List of Figures

Figure 1 – Typical Hardware Setup.....	10
Figure 2 – CY3654 Platform Board	11
Figure 3 –Personality Board, Flex Cable Assembly, Target Probe Adapters	11
Figure 4 – Serial Cable	12
Figure 5 – AC Adapter	12
Figure 6 –Personality Board on the CY 3654 Platform Board with Flex Cable and Applications Board	14
Figure 7 – The CYDB Main Window	18
Figure 8 – Connect to Target Dialog Box 1	22
Figure 9 – Connect to Target Dialog Box 2	23
Figure 10 – Configure Target/Emulator Dialog Box.....	24
Figure 11 – Configuring Development Board Message.....	25

Figure 12 – Configure Development Board Emulator
 Dialog Box 26

Figure 13 – Error Message..... 27

Figure 14 – Development Board Not Ready Dialog
 Box 27

Figure 15 – Data Memory Window..... 29

Figure 16 – Program Memory Window..... 30

Figure 17 – CPU Registers Window..... 31

Figure 18 – IO Registers Window 33

Figure 19 – Memory FIFO Window 34

Figure 20 – Breakpoints Window 35

Figure 21 – Download Dialog Box..... 37

Figure 22 – Select File Dialog Box 38

Figure 23 – Open Dialog Box 39

Figure 24 – .LST File Window 40

Figure 25 – Customize Dialog Box – Toolbars 44

Figure 26 – New Toolbar Dialog Box 46

Figure 27 – Customize Dialog Box – Commands..... 47

Figure 28 – Save Workspace Dialog Box..... 50

Figure 29 – Open Workspace Dialog box..... 52

Figure 30 – Recent Workspaces Sub-menu..... 53

Figure 31 – Workspace Options Dialog Box..... 54

Conventions

Contents

Document Conventions	3
Manual Organization	4

Document Conventions

The following table describes each typographic convention used in this manual.

Example of Convention	Description
Setup	Entries to be typed by the user appear in bold. For example, “Type setup at the prompt.”
Insert menu	Button, icon, menu, and command names appear in bold title caps.
<i>Variable</i>	Italic letters indicate placeholders for information the user supplies. Italic formatting is also used for book titles, figure headings, and occasionally for emphasis.
Set task_pri ori ty = 10 Set task_stack_si ze = 40000	This font is used for code.
CY.EXE	Words in all capital letters indicate filenames. All capital letters are also used for directory and folder names.
ENTER	Small capital letters are used for the names of keys and key sequences, such as ENTER and CTRL+R.
ALT+F1	A plus sign (+) between key names indicates a combination of keys. For example, ALT+F1 means to hold down the ALT key while pressing the F1 key.
ALT,F,S,ENTER	A comma (,) between key names indicates that the individual keys are pressed in sequence.

Example of Convention	Description
DOWN ARROW	Individual direction keys are referred to by the direction of the arrow on the key top (LEFT, RIGHT, UP, or DOWN). The phrase “arrow keys” is used when describing these keys collectively.
BACKSPACE, HOME	Other navigational keys are referred to by their specific names.
Universal Serial Bus (USB)	Acronyms are spelled out the first time they are used. For a complete list of acronyms used in this manual, refer to Appendix A – Acronyms and Abbreviations.

Manual Organization

This manual is composed of three chapters:

Chapter 1 is the INTRODUCTION. This chapter covers the basic capabilities of the system, the system requirements, where to look for help, and who to call for technical support. It also includes step-by-step instructions for installing the hardware and software.

Chapter 2 is OPERATIONS. This chapter describes the debugger environment and how to operate the hardware and software.

Chapter 3 contains the APPENDICES. In the appendices, you will find definitions for acronyms and abbreviations found in this document, the index, and the detailed hardware description.

Chapter 1

Introduction

Contents

Overview	7
Getting Started	7
Installation	10

1.1 Overview

The CYDB Debugger/CY3654 Development Kit is an easy-to-use kit for application code development. It is a combination of hardware and software designed to allow firmware developers to design, debug, test, and emulate the firmware of microcontrollers for a variety of target applications.

1.2 Getting Started

This section will help you get started using the CYDB Debugger/CY3654 Development Kit. It covers what is required to install and run CYDB, what documentation is available (print and online), where to call for technical support, and a basic overview of the CYDB features.

1.2.1 System Requirements

To install and run the CYDB Debugger/CY3654 Development Kit, your computer must have at least the basic Windows 95/NT setup. Minimum requirements include:

- Windows® 95® or later operating system or Windows NT™ Workstation operating system version 4.0 or later.
- An Intel® Pentium® or higher CPU.
- At least 16 megabytes (MB) of Random Access Memory (RAM).
- A personal computer (PC) hard disk or shared network drive with at least 20 MB of hard memory available.

- A video graphics adapter (VGA) monitor.
- A CD-ROM drive.
- A serial input/output (I/O) port.
- A mouse or other suitable pointing device. (The software can be operated without a mouse. However, using a mouse facilitates use of the software.)

1.2.2 Help Sources

The following help sources are available:

- *CYDB User's Guide*.
- *CYDB Reference Manual*.
- Online help.

1.2.3 Technical Support

For technical support:

Web Site: <http://www.cypress.com>

E-mail: CYAPPS@cypress.com

1.2.4 Features Overview

The CYDB Debugging environment offers many features that facilitate debugging and emulating. Below is a list of some of these features. For a more detailed description of the hardware and software, refer to section 1.3, “Installation” or Appendix C “Detailed Hardware Description.”

- C and assembler source level debugging.
- Target cable (flex cable assembly) and AC adapter are included.
- Integrated Development Environment (IDE) software for the development kit is included.
- Personality boards provide IO emulation characteristics of the emulated device.
- The optional Intelligent Tracing Trigger System (ITTS) board can add trace and complex breakpoint features to the base unit. (The ITTS board is sold separately.)

1.3 Installation

This section describes the hardware and software included in the CYDB Debugger/CY3654 Development Kit and the installation of each.

1.3.1 Hardware Installation

This section describes the hardware included in the development kit and its installation. The following diagram shows the typical hardware setup:

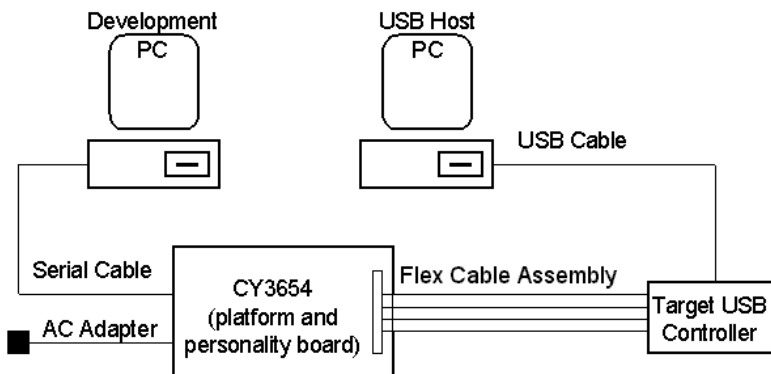


Figure 1 – Typical Hardware Setup

Physical Description

The following is a list of hardware included in the development kit:

- CY3654 Platform Board (see Figure 2).
- Personality Board (see Figure 3).
- Flex cable assembly (see Figure 3).
- Target probe adapter (see Figure 3).
- 4-foot Serial Cable (see Figure 4).
- AC Adapter (see Figure 5).

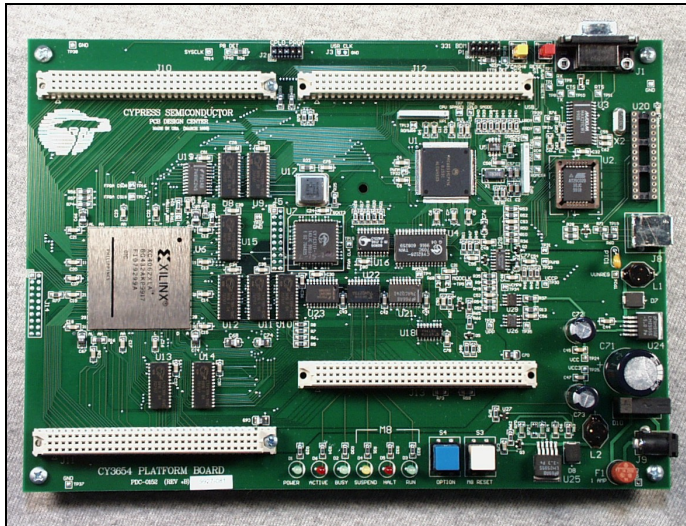


Figure 2 – CY3654 Platform Board

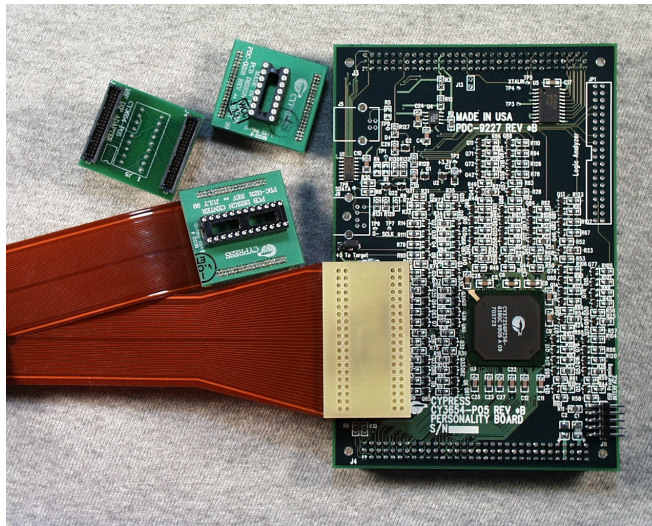


Figure 3 –Personality Board, Flex Cable Assembly, Target Probe Adapters

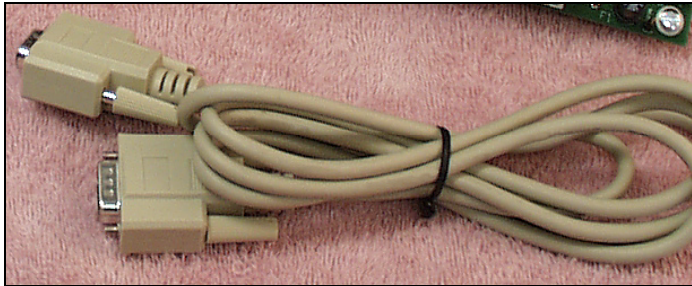


Figure 4 – Serial Cable

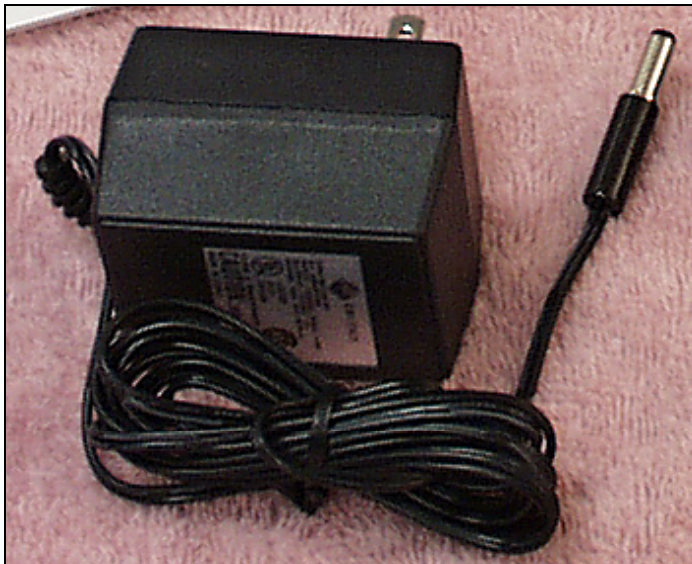


Figure 5 – AC Adapter

Functional Description

Platform Board

The CY3654 platform board contains all hardware and firmware required to run the emulator. It is fully software configurable so that the user is not required to customize the board with DIP switches or jumpers. It has connectors so that the personality board and ITTS can be easily attached. The basic features include downloading and running user programs, setting and stopping on user specified breakpoints, viewing and monitoring CPU registers and RAM, and several other basic emulator functions.

The emulator communicates with the host PC with a standard serial interface that is capable of running at speeds from 9.6K to 115.2K baud. A separate system processor is used to maximize up/download speeds.

Personality Board

The personality board converts the generic connections on the platform board to the correct pin-out required for the packages used by its specific USB microcontroller. A single personality board may have several target probe adapters to support multiple packages. The personality board makes it possible to add emulation hardware to the system to augment functions that can not be emulated by the platform board alone. This changes the characteristics of the emulator to reflect the features of a specific microcontroller.

Intelligent Tracing Trigger System (ITTS) Option

The ITTS option adds trace and complex breakpoint features to the base unit. Address, data, control signals, and I/O bus patterns may be selected by the user to create trace points and set breakpoints. Logical “and/or” combinations of these patterns may be selected to generate

these trigger points. The trace memory is 32Kx32 bits, which equates to approximately a 10K instruction trace. The ITTS option helps the user pinpoint the problem instead of requiring thousands of lines of trace to be read by the user.

Installation Process

To install the hardware,

- Plug the personality board onto the platform board as shown in Figure 6.

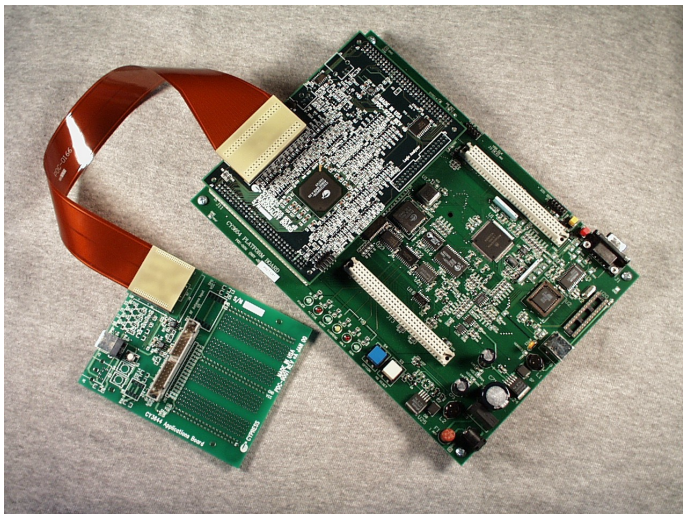


Figure 6 –Personality Board on the CY 3654 Platform Board (with Flex Cable and Application Board)

- Plug the female end of the serial cable to the serial port on the back of the development PC.

- Plug the male end of the serial cable to the serial port at the upper right corner of the platform board.
- Plug the large end of the flex cable assembly to the personality board as shown in Figure 6.



Important

When you plug the assembly to the board, it is important that you make sure the flex cable goes away from the board. If you plug it in backwards, it can damage the board.

- Plug the other end of the flex cable assembly into the appropriate target probe adapter.
- Plug the target probe adapter into the target device.
- Plug the transformer end of the AC Adapter into a 110v outlet.
- Plug the other end of the AC Adapter into the power jack on the lower right corner of the platform board.

1.3.2 Software Installation

This section describes the CYDB software and its installation.

Physical Description

The software for CYDB consists of one compact disc (CD).

Functional Description

Integrated Development Environment (IDE)

This interface provides the following functions to the user:

- Software downloading.
- Viewing and editing memory and registers.
- Setting breakpoints.

Installation Process

To install the software,

- Insert the CD into the CD-ROM drive.
 - If the CD-ROM drive is set to auto-run, the setup program will run automatically.
 - If the CD-ROM drive is not set to auto-run, in Explorer, double-click SETUP.EXE from the directory window for your CD-ROM drive.
- Follow the instructions on your screen.

1.3.3 Installation Verification

To verify that the software has loaded properly, perform the following:

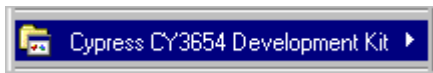
Launching the Program for the First Time



On the taskbar, click the **Start** button. The **Start** menu appears.



Click **Programs**. The Programs sub-menu appears.



Click Cypress CY3654 Development Kit. The Cypress CY3654 Development Kit sub-menu appears.



Click Cypress **Debugger**. The CYDB Main Window appears (see [Figure 7](#)).



Note

Each time you restart CYDB, the workspace displayed is the last workspace you were working in when you last closed the CYDB Main Window. Refer to the [“Saving and Opening a Workspace”](#) section and the [“Workspace Options”](#) section.

Default Configuration – What Do You See?

When the CYDB Main Window opens the first time, the workspace configuration is set to default. The default configuration consists of the Menu bar with all menu items available, the Tool bar which displays all tools (although some are unavailable), and the Status bar.

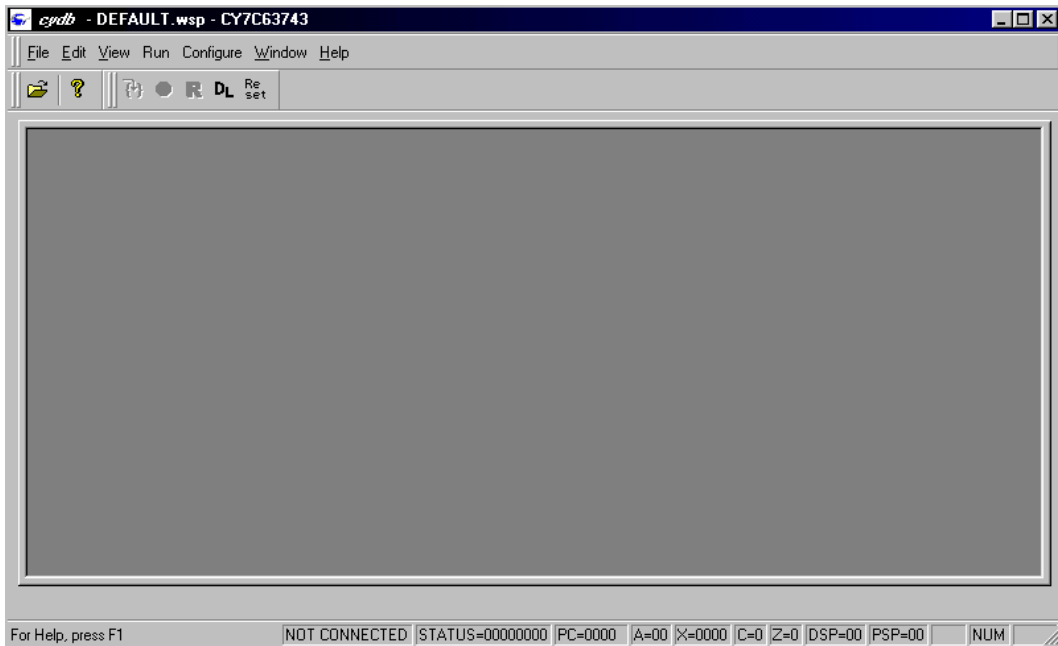


Figure 7 – The CYDB Main Window

- To change the workspace configuration, refer to section 2.5, “Customizing Your Workspace.”
- To save and open a new workspace configuration, refer to “Saving and Opening a Workspace” under section 2.5.2, “Operation.”
- After you install CYDB, you may also need to configure your target USB controller. Refer to Section 2.3, “Configuration.”

Chapter 2

Operations

Contents

Overview	21
Connect to Target	22
Configuration	23
Debugger Operation	28
Customizing Your Workspace	43

2.1 Overview

This section describes the operation of CYDB. It gives you step-by-step instructions for configuring and running the debugger and customizing your workspace.

2.1.1 How the Hardware and Software Work Together

The CYDB debugging environment is your connection between the “host” system where CYDB is run and the emulator system where your programs run. The emulator has the ability to configure the emulator for multiple microcontrollers.

The emulator governs the communication between your host and target, so you can debug without consuming target resources. It can reproduce the behavior of real or theoretical target hardware on your host platform. Then, you can debug software before hardware is available or while target hardware is not connected to your host. You can emulate any device: mouse, keyboard, joystick, scanner, printer, camera, zip drive, etc.

2.2 Connect to Target

Before you configure your hardware, you must connect to the emulator:



- In the CYDB Main Window, click **Run** on the menu bar. The **Run** menu appears.



- Click **Connect**. The Connect to Target dialog box 1 appears:

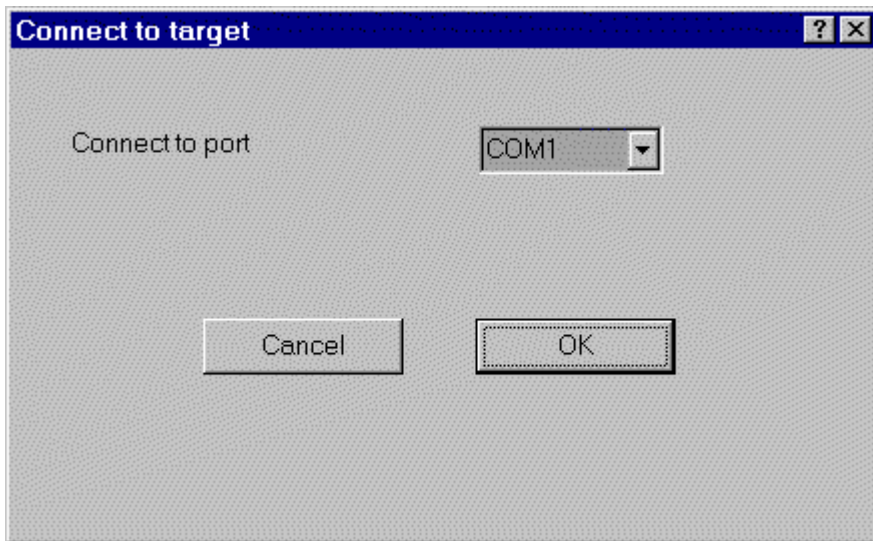
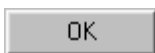


Figure 8 – Connect to Target Dialog Box 1

- Select the communications port you want to connect to.
- Click **OK** to confirm the port selection and connect to the emulator. The Connect to Target dialog box 2 appears (see Figure 9).



- To exit without connecting, click **Cancel**.

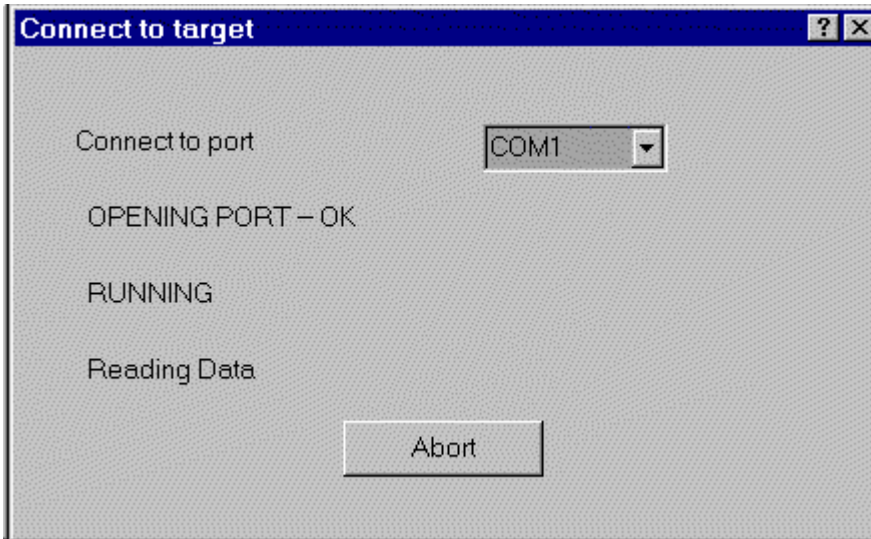


Figure 9 – Connect to Target Dialog Box 2

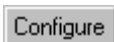


- Click **Abort** if you want to abort the connection process. The dialog box disappears when the connection is complete or if the connection times out.

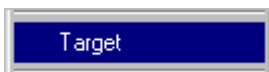
2.3 Configuration

2.3.1 Target Chip Configuration – Loading Personality

To set the connections to your hardware:



- In the CYDB Main Window, click **Configure** on the menu bar. The **Configure** menu appears.



- Click **Target**. The Configure Target/Emulator dialog box appears:

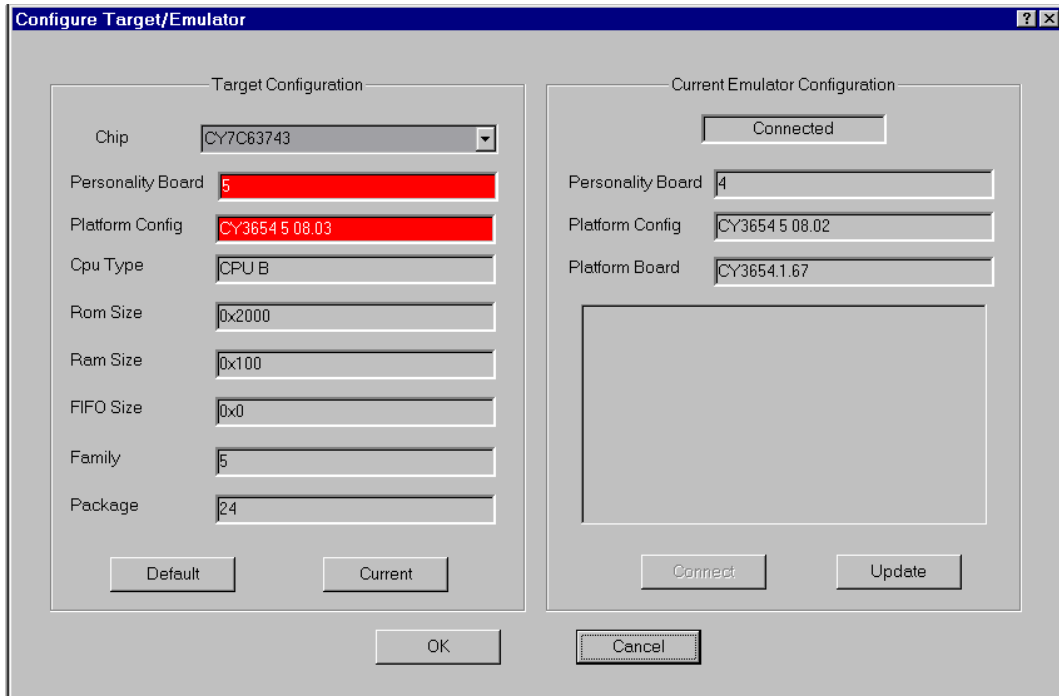

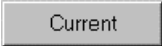


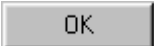


Figure 10 – Configure Target/Emulator Dialog Box

The Target Configuration section (the left side of the workspace) shows the configuration for the selected chip. It shows the required personality board and platform board configurations (these are highlighted if updating is required). It also shows the properties of the selected microcontroller.

The Current Emulator Configuration section (the right side of the workspace) shows the configuration of the currently connected platform board.

- In the pull-down list, select the microcontroller you want to emulate. The default is **CY7C63743**.

-  • Displays the configuration for the default chip.
-  • Displays the configuration for the chip currently selected in the workspace.
-  • Click **Connect** to connect to the development board (if you are not already connected).
-  • Click **Update** to update the chip configuration.
 - If the Development Board is not ready, the Development Board Not Ready dialog box appears (see Figure 14).
 - Follow the Instructions and click **OK**. The Configuring Development Board Message appears:
- 

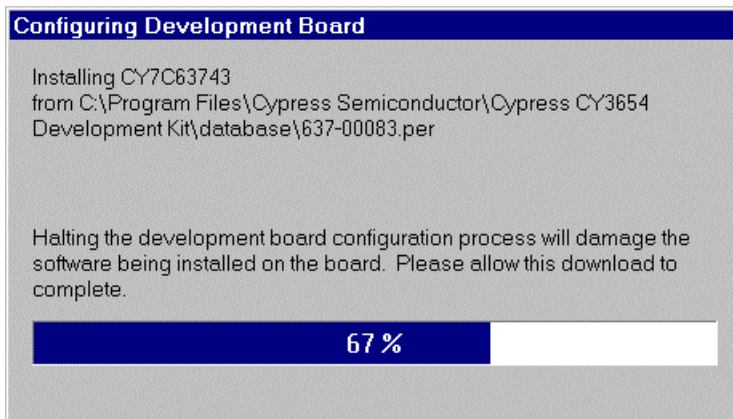
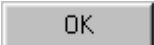



Figure 11 – Configuring Development Board Message

- Wait for the download to complete. When it is finished, the message disappears and you can:
 -  • Click **OK** to confirm the microcontroller selection.
 -  ➤ Click **Cancel** to exit without selecting a microcontroller.

2.3.2 Emulator Configuration – Entering Stand-alone Mode

To upgrade the firmware on the Platform Board, perform the following:



- In the CYDB Main Window, click **Configure** on the menu bar. The **Configure** menu appears.



- Click **Update Firmware**. The **Configure Development Board Emulator** dialog box appears:

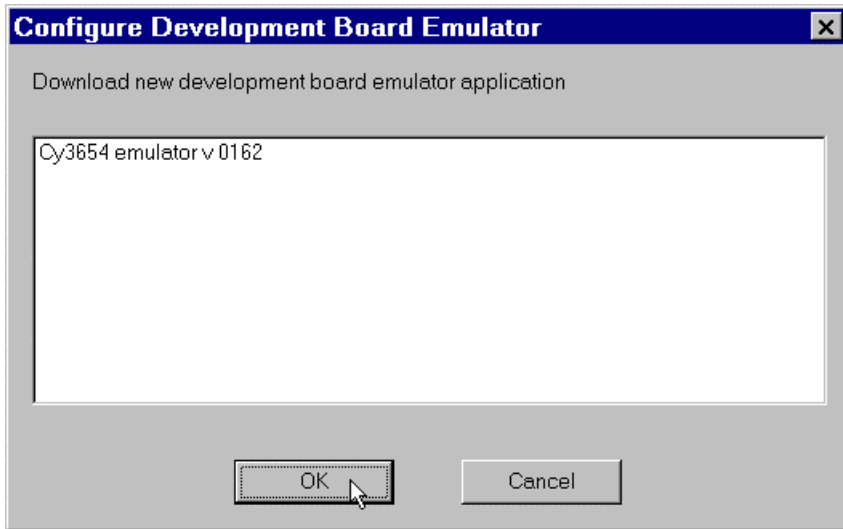
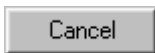


Figure 12 – Configure Development Board Emulator Dialog Box



- To exit without selecting an emulator configuration, click **Cancel**.



- Select the desired emulator configuration and click **OK**. The emulator is now configured.

- If the development board is not ready for configuration, the following error message appears:



Figure 13 – Error Message



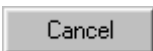
- Click **OK**. The Development Board Not Ready Dialog Box appears:



Figure 14 – Development Board Not Ready Dialog Box



- + Follow the instructions in the dialog box and then click **OK** to continue.



- + To return to the CYDB Main Window, click **Cancel**.

2.4 Debugger Operation

This section describes the operation of the Debugger software. In this section you will learn how to view, modify, and refresh the different windows available in the CYDB Main Window, and how to run the debugger.

2.4.1 How to View/Modify/Refresh

There are five windows available in the CYDB Main Window. This section describes these windows and tells you how to view, modify, and refresh them.

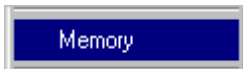
Data Memory Window

The Data Memory window displays the contents of the RAM. Each **Value** field represents a location in data memory. For example, the cell in the 0x000090 row and the third value column from the left contains the program code for address 0x000092.

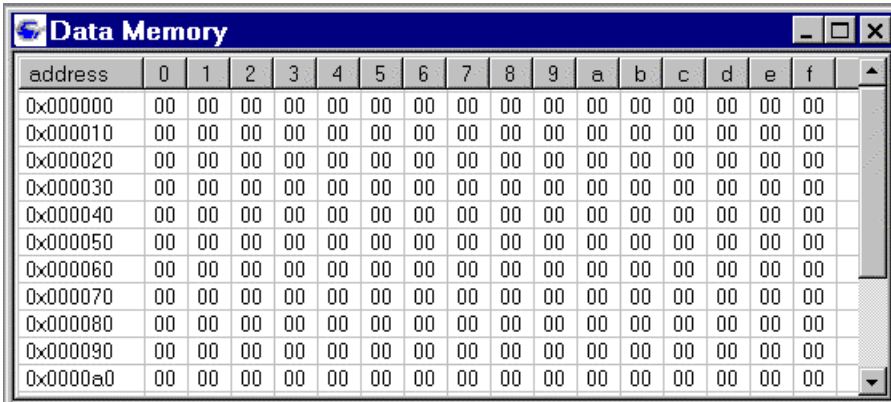
View



- In the CYDB Main Window, click **View**. The view menu appears.



- Click **Memory**. The Data Memory window appears:



address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0000a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 15 – Data Memory Window

Modify

You can only modify data in the **Value** fields. You can not modify data in the **Address** field. To make a change to the RAM program:

- Double click the cell you want to change.
- Type the new value in that field.
 - Press [Enter] to save the change.
 - Press [Esc] to cancel the change.

Program Memory Window

The Program Memory window displays the contents of the ROM. Each **Value** field represents a different line of programming. For example, the cell in the 0x0000d0 row and the third value column from the right contains the program code for line 0x0000dd.

View



- In the CYDB Main Window, click **View**. The view menu appears.



- Click **Program**. The Program Memory window appears:

address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0000a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 16 – Program Memory Window

Modify

You can only modify data in the **Value** fields. You can not modify data in the **Address** field. To make a change to the ROM program:

- Double click the cell you want to change.
- Type the new value in that field.
 - Press [Enter] to save the change.
 - Press [Esc] to cancel the change.

CPU Registers Window

The CPU Registers window allows you to view and modify the value for each register on the processor.

View



- At the CYDB Main Window, click **View** on the menu bar. The View menu appears.



- Click **Registers**. The CPU Registers window appears:

register	value	
A	00	
X	00	
PC	00	
PSP	00	
DSP	00	

Figure 17 – CPU Registers Window

Modify

You can only modify data in the **Value** field. You can not modify data in the **Register** field. To make a change to a CPU register:

- Double click the value you want to change.
- Type the new value in that field.
 - Press [Enter] to save the change.
 - Press [Esc] to cancel the change.



Note

For information pertaining to the definition of and the possible values for each register, refer to the *Cypress USB Development Kit Assembler Guide*.

IO Registers Window

The IO Registers window allows you to view and modify the value for each IO port.

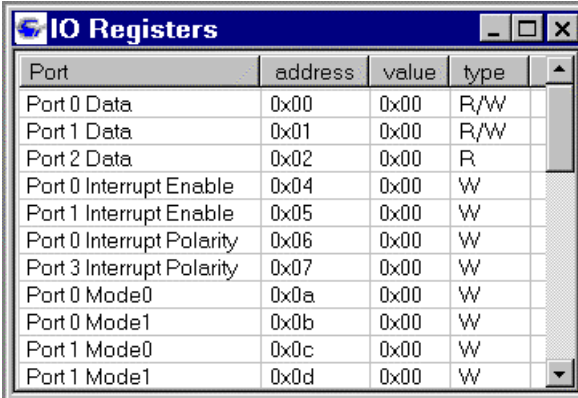
View



- At the CYDB Main Window, click **View** on the menu bar. The View menu appears.



- Click **IO Registers**. The IO Registers window appears:



Port	address	value	type
Port 0 Data	0x00	0x00	R/W
Port 1 Data	0x01	0x00	R/W
Port 2 Data	0x02	0x00	R
Port 0 Interrupt Enable	0x04	0x00	W
Port 1 Interrupt Enable	0x05	0x00	W
Port 0 Interrupt Polarity	0x06	0x00	W
Port 3 Interrupt Polarity	0x07	0x00	W
Port 0 Mode0	0x0a	0x00	W
Port 0 Mode1	0x0b	0x00	W
Port 1 Mode0	0x0c	0x00	W
Port 1 Mode1	0x0d	0x00	W

Figure 18 – IO Registers Window

Modify

You can only modify data in the **Value** field. You can not modify data in the **Port**, **Address**, or **Type** fields. To make a change to an IO register:

- Double click the value you want to change.
- Type the new value in that field.
 - Press [Enter] to save the change.
 - Press [Esc] to cancel the change.



Note

For more information pertaining to the IO registers, refer to the *Cypress USB Development Kit Assembler Guide*.

Memory FIFO Window

The Memory FIFO window displays the contents of the FIFO. Each **Val** field represents a different line of programming. For example, the cell in the 0x0000d0 row and the third value column from the right contains the program code for line 0x0000dd.



Note

Not all supported devices have FIFOs.

View



- In the CYDB Main Window, click **View**. The view menu appears.



- Click **FIFO**. The Memory FIFO window appears:

address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0000a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 19 – Memory FIFO Window

Modify

You can only modify data in the **Val** fields. You can not modify data in the **Address** field. To make a change to the FIFO:

- Double click the cell you want to change.
- Type the new value in that field.
 - Press [Enter] to save the change.
 - Press [Esc] to cancel the change.

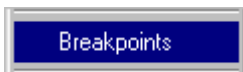
Breakpoints Window

The Breakpoints window contains only the address field. This field shows the address of each existing breakpoint to be planted in memory and triggered when executed.

View



- At the CYDB Main Window, click **View** on the menu bar. The View menu appears.



- Click **Breakpoints**. The Breakpoints window appears:

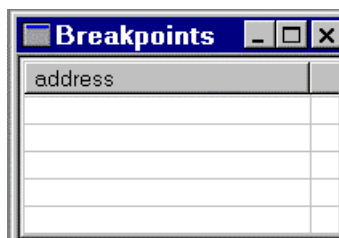


Figure 20 – Breakpoints Window

Modify

There are two ways to modify the data in the **address** field:

1. Set or clear the breakpoint in the program's .LST file (see "Running an Application" and "Working with Breakpoints").
 - When you set a breakpoint in the .LST file, the program counter position (i.e., 0x000092) of that breakpoint appears in the **address** field in the Breakpoints window.
 - When you clear a breakpoint in the program's .LST file, the address of that breakpoint is removed from the **address** field of the Breakpoints window.
2. Set or clear the breakpoint in the Breakpoints window:
 - Double click the address you want to change or click in an empty cell.
 - Delete the address to clear the breakpoint or type the address of the new breakpoint.
 - When you add, change, or delete an address in the Breakpoints window, the modification is reflected in the program's .LST file.

2.4.2 Application Operation

This section tells you how to download an application to the emulator, reset an application, step, and run, set breakpoints, and stop.



Note

Before you begin downloading and running an application, make sure the platform board is properly connected to the development PC and the target USB controller (refer to section 1.3, “Installation”).

Downloading an Application



- In the CYDB Main Window, click **Run** on the menu bar. The **Run** menu appears.



- Click **download**. The Download dialog box appears:

➤ You can also click the Download () button.

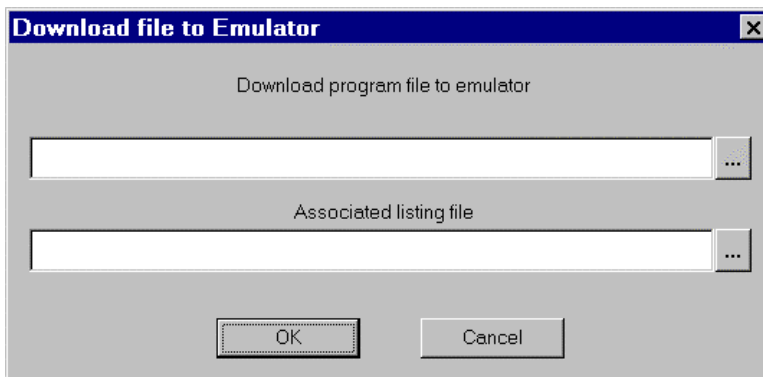


Figure 21 – Download Dialog Box



- Click the browse button. The Select File dialog box appears:
 - If you know the path and filename of the application you want to download, you can type it in the text box instead of browsing for it.

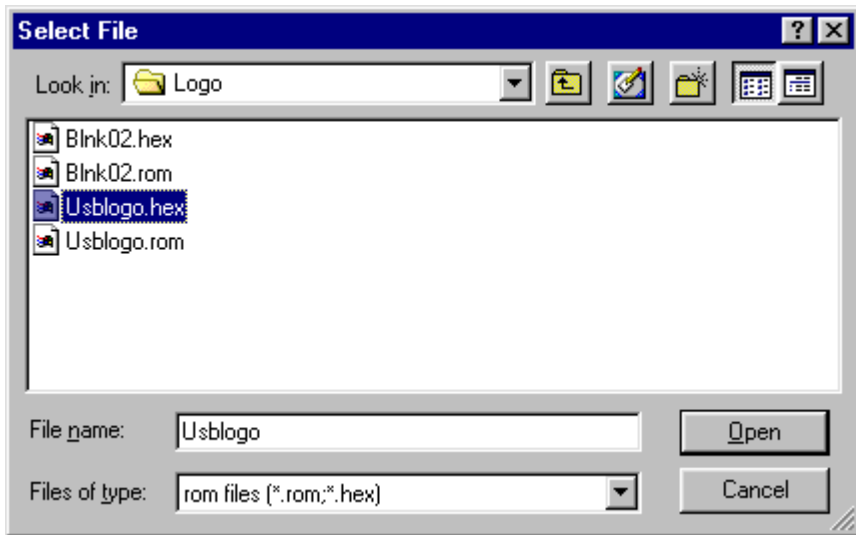


Figure 22 – Select File Dialog Box

- Select the application you want to download and click **Open**. The Download dialog box reappears with the application's path and filename entered in the text box.
 - Applications have the .ROM or .HEX file extension.

- At the Download dialog box, click **OK**. The program downloads to the emulator, and the .LST file is updated and opened.

- To exit the **Download** or **Select File** dialog box without selecting an application, click **Cancel**.

Running an Application

Once you have loaded an application, you typically want to see your program run. The Run command starts the program at the current program counter position and runs the program until a breakpoint is encountered, or it finishes.

- Open the .LST file for the downloaded application:

File

- In the CYDB Main Window, click **File** on the menu bar. The **File** menu appears.



- Click **Open**. The Open dialog box appears:

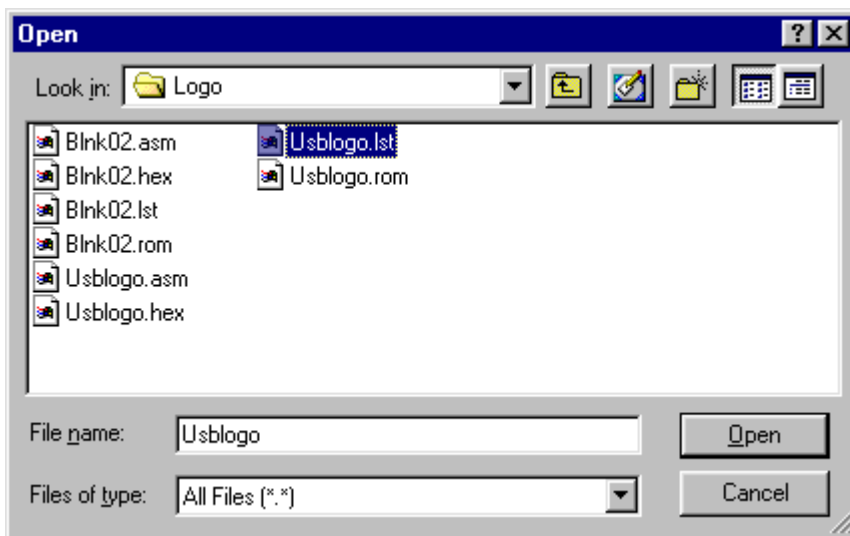


Figure 23 – Open Dialog Box

- Select the path to the .LST file you want and click **Open**. The .LST file window appears:

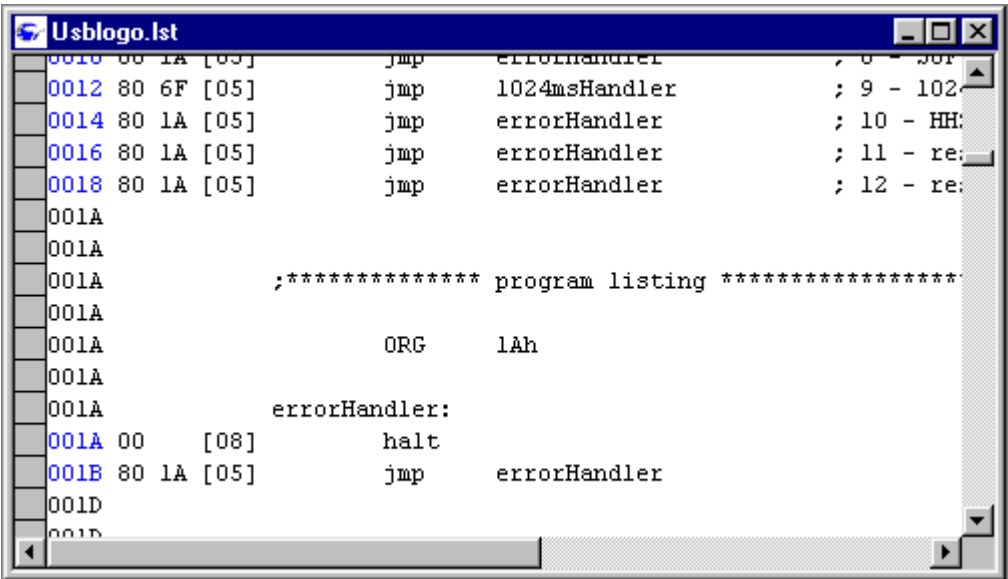


Figure 24 – .LST File Window

- Run the program:



- Click the **Run** button in the toolbar.

Or

- From the **Run** menu, select **Run**.

The debugger monitors the states of the emulator. If the microcontroller halts because of a breakpoint, or executes a halt instruction, the **Stop** button (■) becomes inactive and the **Run** button (R) becomes active again. The .LST file window then reflects the new program counter.

Resetting an Application

When the application is stopped, you can start it again at the point where you left it, or you can *reset* it. When you reset it, the program counter moves to the beginning of the program (address 0) and resets the target hardware if it is connected.

- To reset a program, select **Reset** from the **Run** menu.

Stepping

When you debug a program, you often want to run it a small portion at a time. This is called “stepping”. A step corresponds to one instruction.

“Stepping” runs the program until the next instruction. If the program counter starts on a function call, “step” calls the function, then stops at the first instruction within the function.

- To step:



- Click the **Step** button in the toolbar.

Or

- From the **Run** menu, select **Step**.

Working with Breakpoints

Breakpoints stop your program at strategic points so you can then look at the current state of your target. Most debugging activities either directly or indirectly use breakpoints. There are two ways to set or clear a breakpoint. You can add, change, or delete the address of the breakpoint in the Breakpoints window (see "Breakpoints Window") or you can perform the following:

- To set a breakpoint,
 - In the .LST file window, double-click on the line where you want the program to stop. A red dot appears in the left margin and the first four digits of the address are highlighted blue.
- To clear a breakpoint,
 - Double-click on the instruction where you set the breakpoint. The red dot disappears and the highlighted text reverts to black.

Stopping

While the program is running, you can stop it at any time. The program counter positions itself on the next instruction that will be executed when you start the program again. To stop a running program,



- Click the **Stop** button in the toolbar.

Or

- From the **Run** menu, select **Stop**.

2.5 Customizing Your Workspace

“Workspace” refers to the area within the CYDB Main Window. This section tells you how to customize your workspace settings, how to save your settings to a workspace file, and how to load a saved workspace. Saving a workspace also *remembers* the previous configuration, including executable files and listing files.

2.5.1 Customizing Toolbars and Commands

You can customize your workspace by selecting which toolbars and command buttons to display, and then saving the workspace settings. To begin customizing your workspace, perform the following.

A rectangular button with a light gray border and the text "Configure" in a standard font.

- In the CYDB Main Window, click **Configure** on the menu bar. The **Configure** menu appears.

A blue rectangular menu item with a white border, the text "Workspace" in white, and a white right-pointing arrow.

- Click **Workspace**. The **Workspace** sub-menu appears.

A blue rectangular button with a white border, the text "Customize..." in white, and a white underline under the letter 'C'.

- Click **Customize**. The Customize dialog box appears with the **Toolbars** tab selected:

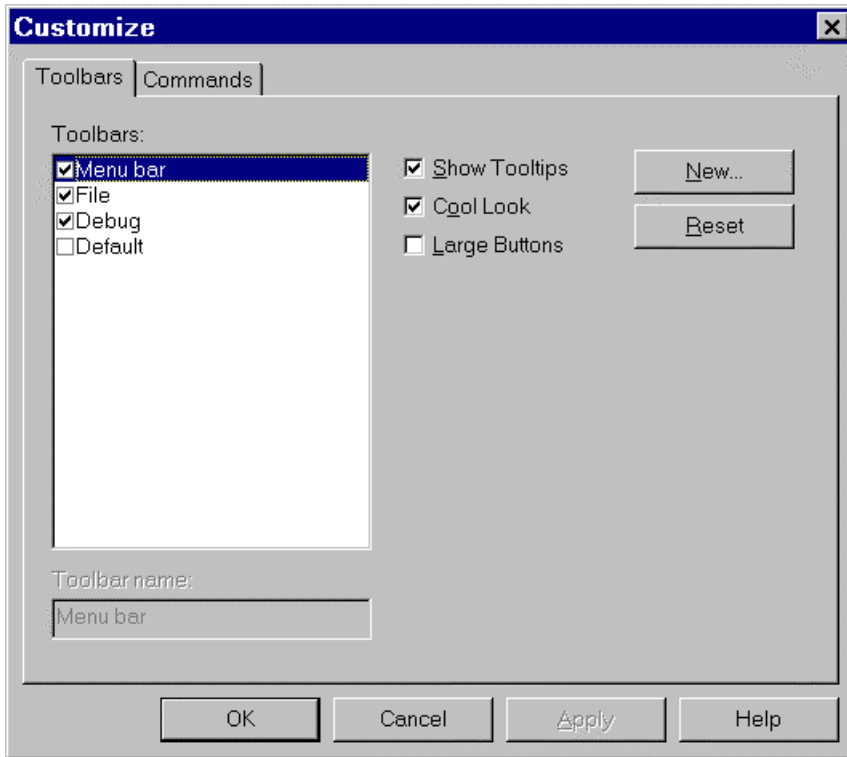


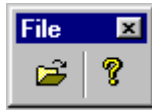
Figure 25 – Customize Dialog Box – Toolbars

Toolbars

To add or remove a toolbar from the CYDB Main Window, perform the following:

- In the Toolbars group box, select one or more check box.
 - The Menu bar check box is always selected. You can not clear the selection.
 - Select the Default check box to enable the File, Edit, and Debug toolbars.

- Select the File check box to enable the File toolbar:

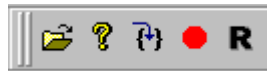


- Select the Debug check box to enable the Debug toolbar:



- Select the Show Tooltips check box if you want to display ToolTips when the mouse pointer is over a button.
- Select the Cool Look check box if you want to change the appearance of the menu bar and toolbar(s):

Cool Look (R)



Regular Look (R)



- Select the Large Buttons check box if you want to make the command buttons appear larger:



Create a New Toolbar



- In the Toolbars group box, click **New**. The New Toolbar dialog box appears:

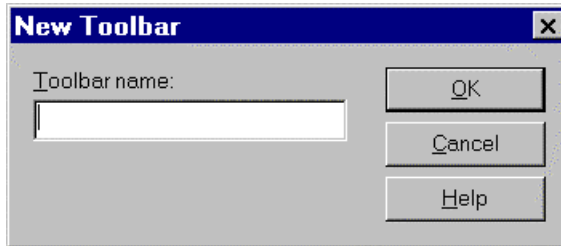


Figure 26 – New Toolbar Dialog Box



- Type the name you want to give your new toolbar and click **OK**. The new toolbar name appears in the Toolbars group box and the new toolbar appears in the CYDB Main Window.

Delete a New Toolbar

A rectangular button with a light gray background and a thin border. The text "Delete" is centered on the button in a dark gray font.

- In the Toolbars group box, click on the name of the new toolbar you want to delete and click **Delete**. The New Toolbar name is removed from the Toolbars group box and the new toolbar is removed from the CYDB Main Window.

Commands

- In the Customize dialog box, click the **Commands** tab. The Commands tabbed page appears:

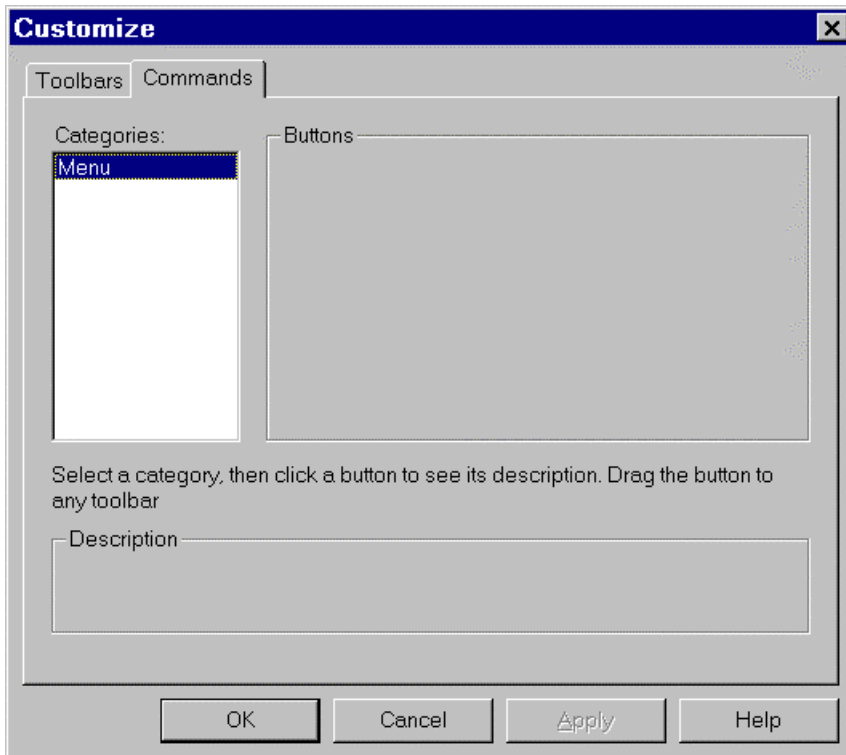


Figure 27 – Customize Dialog Box – Commands

- In the Categories list box, select a category.
 - The only category available at this time is “Menu.”
 - Click a menu name and drag it to the menu bar.
 - To remove a menu name from the menu bar, drag it back to the Customize dialog box.

Reset Changes to the Menu Bar

If you add a number of menu bar items, you can reset the changes before you click **OK**. To undo your changes, perform the following:

- Click the **Toolbars** tab. The Toolbars tabbed page appears.
- Click **Reset**. The toolbars and menu bar return to their previous configuration.

Save Changes to the Menu Bar



- Once you have made all the changes to the workspace, click **OK**.



Note

Clicking **Cancel** will only undo the changes made to the Show Tooltips, Cool Look, and Large Buttons check boxes before closing the Customize dialog box.

2.5.2 Saving and Opening a Workspace

Once you have customized your workspace, you can save your settings in a workspace file. Once a workspace file is created, you can open the file to apply those workspace settings to your current workspace.



Note

Each time you restart CYDB, the workspace displayed is the last workspace you were working in when you last closed the CYDB Main Window. Refer to the “Open a Workspace” section.

Save Your Workspace

When you launch the program for the first time, the workspace file DEFAULT.WSP provides the default workspace settings. After you make changes to the workspace settings, there are two ways to save the workspace. You can save the settings as your default workspace or as a new workspace to which you assign a name.

Save Workspace—Default



- In the CYDB Main Window, click **File** on the menu bar. The **File** menu appears.



- Click **Save Workspace**. The workspace settings are saved as your default workspace.

 **Note**

If you have a workspace open that is not the default workspace and you click **Save Workspace**, the workspace is saved as its given filename. It is *not* saved as the default file.

Save Workspace As—New



- In the CYDB Main Window, click **File** on the menu bar. The **File** menu appears.



- Click **Save Workspace As**. The Save Workspace Dialog box appears:

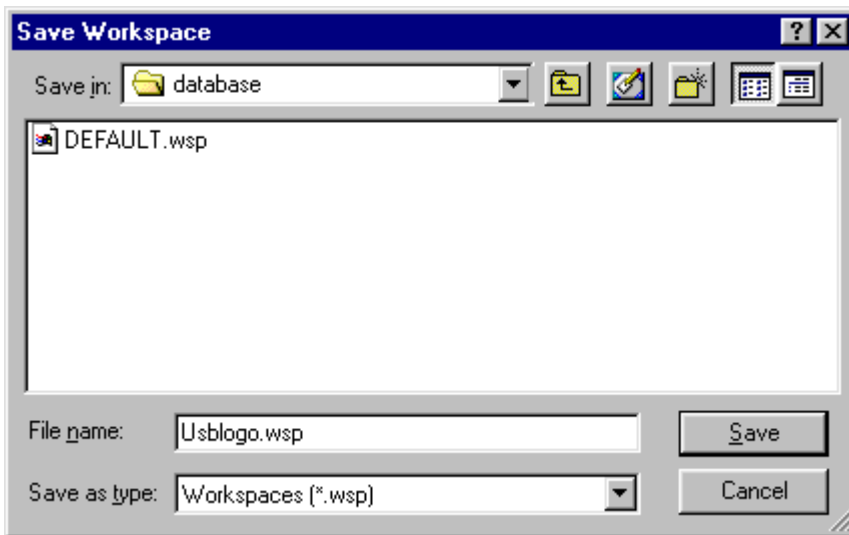


Figure 28 – Save Workspace Dialog Box

- Type the name you want to give the workspace in the File Name box.
- If you have made changes to an already existing workspace file, type or select that filename.

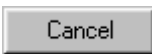


Note

DEFAULT.WSP is the default workspace file. Do not type or select this filename unless you want to change your default workspace settings.



- Click **Save**. The workspace is now saved.



- To close the Save Workspace dialog box without saving the file, click **Cancel**.

Open a Workspace

There are two ways to open a workspace. You can select a workspace file from the Open Workspace dialog box or from the list of Recent Workspaces.

Open Workspace



- In the CYDB Main Window, click **File** on the menu bar. The **File** menu appears.



- Click **Open Workspace**. The Open Workspace Dialog box appears:

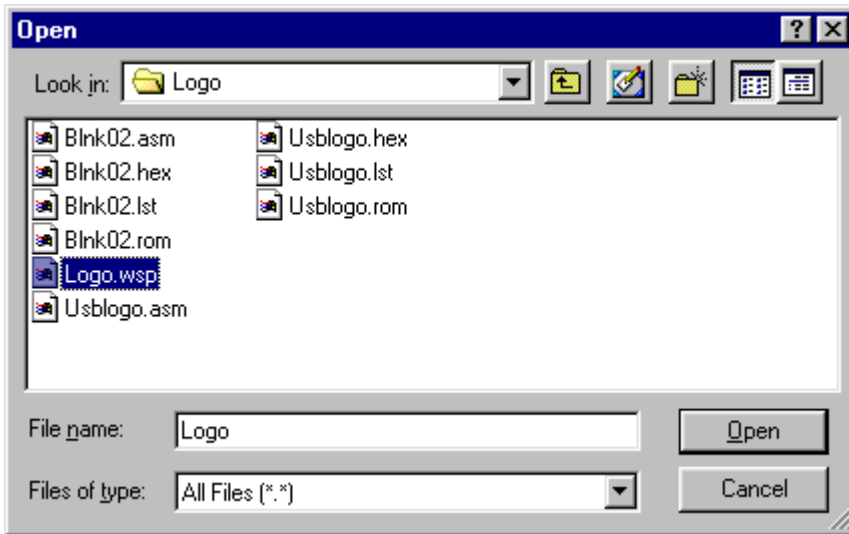
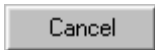


Figure 29 – Open Workspace Dialog box



- Select the path to the workspace you want to open and click **Open**.



- To exit the **Open Workspace** dialog box without selecting a workspace, click **Cancel**.

Recent Workspaces



- In the CYDB Main Window, click **File** on the menu bar. The **File** menu appears.



- Click **Recent Workspace**. The Recent Workspaces Sub-menu appears:



Figure 30 – Recent Workspaces Sub-menu

- Click the workspace file you want to open.
 - Remember, the DEFAULT.WSP file is the default.

Workspace Options

The Workspace Options dialog box allows you two options: Connect on start and Load Last Workspace.

- From the **Workspace** sub-menu, click Options. The Workspace Options dialog box appears:

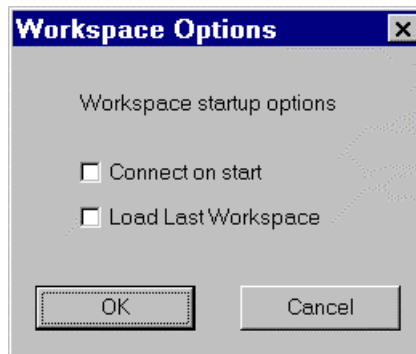
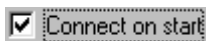
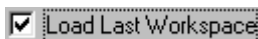


Figure 31 – Workspace Options Dialog Box



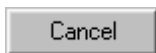
- Select **Connect on start** to automatically connect to the development board upon opening the CYDB Main Window.



- Select **Load Last Workspace** to automatically open the workspace you last used before closing the CYDB Main Menu.



- Click **OK** to apply the options.



- Click **Cancel** to exit the Workspace Options dialog box without selecting an option.

Chapter 3

Appendices

Contents

Acronyms and Abbreviations	57
Index	59
Detailed Hardware Description	61

Appendix A Acronyms and Abbreviations

The following is an alphabetical list of acronyms and abbreviations found in this document, and their definitions.

Term	Definition
AC	Alternating Current
C	A programming language
CD	Compact Disc
CD-ROM	Compact Disc Read-Only Memory
com	Communications
CPU	Central Processing Unit
DIP	Dual Inline Package
FAQs	Frequently Asked Questions
FIFO	First In First Out
FPGA	Field Programmable Gate Array
IDE	Integrated Development Environment
I/O	Input/Output
ITTS	Intelligent Tracing Trigger System
K	Kilo- (one thousand)
KB	Kilobyte
M	Mega- (one million)
PC	Personal Computer
PDF	Portable Document Format
RAM	Random Access Memory
ROM	Read Only Memory
SRAM	Static Random Access Memory
USB	Universal Serial Bus

Term	Definition
v	volt
VGA	Video Graphics Adapter
www	World Wide Web

Appendix B Index

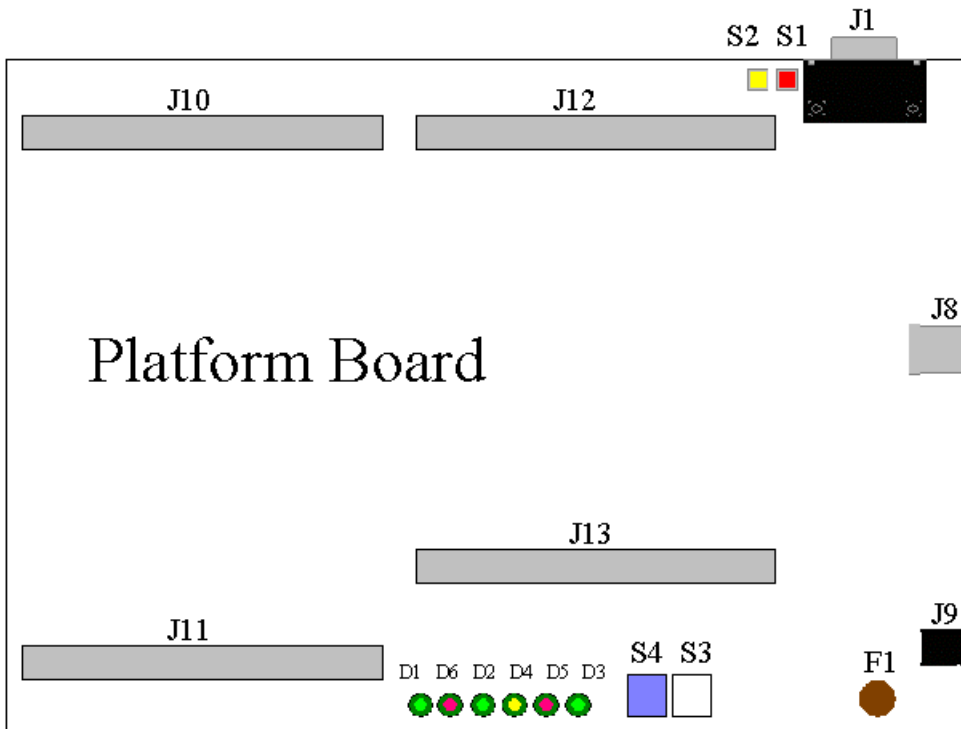
- Abbreviations, 61
- AC adapter, 9
- AC Adapter, 12, 15
- Acronyms, 61
- Appendices, 59
- application, 7, 37, 38, 39, 41, 52, 54
- arrow keys, 4
- breakpoint, 9, 13, 39, 42
- breakpoints, 13
- Breakpoints, 42
- Breakpoints Window, 35
- camera, 21
- CD-ROM, 16
- Commands**, 43, 47
- Communications**, 61
- Configuration, 18, 23
- Configure**, 23, 26, 43
- Configure Development Board
 - Emulator Dialog Box, 26
- CPU registers, 13
- CPU Registers Window, 31
- Customizing, 18, 43
- CYDB debugging environment, 21
- CYDB Main Window, 18, 22, 23,
 - 28, 43, 44, 46, 47, 49, 51, 53
- Data Memory Window, 28, 29
- debug, 7, 21, 41
- Debug, 44, 45
- Debugger, 15, 28
- Debugger Operation, 28
- Default Configuration, 18
- Development, 9
- Development Board**, 26
 - Development Board Not Ready Dialog Box, 27
- Development Kit, 7, 9, 21
- development PC, 14
- DIP switches, 13
- Document Conventions, 3
- downloading, 13
- Downloading, 37
- emulator, 13, 21
- Emulator Configuration, 26
- emulator functions, 13
- firmware, 7, 13
- flex cable assembly, 9, 15
- Flex cable assembly, 10
- Flex Cable Assembly, 11
- Getting Started, 7
- hardware, 10
- Hardware Installation, 10
- Help Sources, 8
- host PC, 13
- I/O bus, 13
- IDE, 9
- Installation, 9, 10, 14, 16, 17
- Installation Verification, 17
- Integrated Development Environment (IDE), 16
- IO Registers Window, 32, 33
- ITTS, 9, 13
- ITTS option, 13
- joystick, 21
- jumpers, 13
- keyboard, 21
- Memory FIFO Window, 34

- Microcontroller Selection, 23
- Modify, 28
- monitoring, 13
- mouse, 21
- Open Workspace, 51, 52, 53
- personality board, 13, 14, 15
- Personality board, 9
- Personality Board, 10, 11, 13, 14
- pin-out, 13
- platform board, 13, 14, 15, 37
- Platform Board, 10, 11, 13, 14
- printer, 21
- Program Memory Window, 30
- RAM, 13
- Recent Workspaces, 51
- Refresh, 28
- Reset**, 40, 41, 48
- Resetting, 41
- Run**, 39, 40, 41, 42
- running, 13
- Running, 39
- Save Workspace**, 49, 50, 51
- Save Workspace As, 50
- scanner, 21
- Serial Cable, 12
- setting, 13
- SETUP.EXE, 16
- Show Tooltips, 45, 48
- Software Installation, 15
- Status bar, 18
- Step In**, 41
- Stepping, 41
- Stop**, 42
- stopping, 13
- Stopping, 42
- System Requirements, 7
- target, 7, 15, 18, 21, 37, 41, 42
- Target cable, 9
- Target Chip Configuration Dialog
 - Box, 22, 23, 24, 25
- target hardware, 21
- target probe adapter, 15
- Technical Support, 8
- theoretical target hardware, 21
- Toolbars**, 43, 44, 46, 47, 48
- trace, 9, 13
- trace memory, 13
- USB, 18, 37, 61
- VGA, 7, 62
- View, 28
- viewing, 13
- Workspace**, 18, 43, 49, 51, 52, 53, 54
- workspace configuration, 18
- XX.WSP, 53, 54
- zip drive, 21

Appendix C Detailed Hardware Description

Platform Board

The following diagram is an approximation of the components on the CY3654 Platform Board. Each component is described in the table following the diagram.



Jumpers	J1	Serial Port
	J8	USB Connector (not used)
	J9	Power Connector
	J10/J11	Personality Board Connectors
	J12/J13	ITTS Option Board Connectors (future option)

Switches	S1	Emulator Reset
	S2	Flash Mode
	S3	M8 Reset
	S4	Not used
LEDs	D1	Power – on when power is supplied
	D2	Busy – on when executing host command or board init.
	D3	Run – on when user program is running
	D4	Suspend – on when M8 processor is in suspend mode
	D5	Halt – on when user program is halted
	D6	Active – blinks when board is active