ZGP323ICE01ZEM

Z8 GP™ ZGP323 In-Circuit Emulator and Development Platform

User Manual

UM017503-0208

Revision History

Each instance in Revision History reflects a change to this document from its previous revision. For more details, refer to the corresponding pages and appropriate links in the table below.

Date	Revision Level	Description	Page No
February 2008	03	Updated Zilog logo, Zilog text, Disclaimer section, Table 1, Using an Event to Stop Execution, Single-Stepping Through a Program, Burn Code from the Current Project, and Burn Code from an Existing Hex File sections.	All
October 2004	02	Corrected Using J9 Pin 1, External Trigger In, Using J9 Pin 3, External Trigger Out sections.	22
October 2004	01	Original issue.	All

Table of Contents

Introduction

Zilog's Z8 GP™ In-Circuit Emulator (ICE) provides Z8 GP family chip emulation with a Trace and Event system for program debugging using Zilog Developer Studio II (ZDS II) development tools. The OTP programming module is used to burn your design on to OTP devices.

System Requirements

Table 1 lists the system requirements for running ZDS II.

Table 1. ZDS II System Requirements

Recommended Configuration	Minimum Configuration
PC running MS Windows XP, SP1	PC running MS Windows 98SE/Win2000–SP3/WinXP–SP1
Pentium III/500 MHz processor	Pentium II/233 MHz processor
128 MB RAM	96 MB RAM
110 MB hard disk space	25 MB hard disk space (documentation not included)
Super VGA video adapter	Super VGA video adapter
CD-ROM drive	CD-ROM drive
Ethernet port	Ethernet port
One or more RS-232 communications ports	One or more RS-232 communications ports
Internet browser (Internet Explorer or Netscape)	Internet browser (Internet Explorer or Netscape)

Software Installation

Follow the steps below to install ZDS II with ANSI C-Compiler:

1. **DemoShield** program available in the ZDS II installation CD launches automatically. If it does not automatically launch, go to the root of the CD-ROM and double-click the file `launch.exe`.
2. **DemoShield** provides several installation options to install ZDS II, select **Install ZDS II**. You can install other software and accompanying documentation later.
3. Follow the instructions on the screen to complete the installation.

To receive free technical support, register your software at www.zilog.com. To access the registration page, open the **Support** menu at the top of the web page and click **Product Registration**.

Hardware Installation

Z8 GP™ ZGP323 ICE and development platform features an Ethernet interface and an RS-232 serial port. Hardware installation consists of the following:

- Connecting Target Pod
- Connecting Z8 GP ICE to Target Pod
- Connecting Z8 GP ICE to OTP Programming Module (Optional)
- Connecting Z8 GP ICE to a PC

You have to reconfigure network settings on the PC or on the Crimzon ICE before using the emulator.

Connecting Target Pod

Z8 GP ICE kit is ideal for use with a Z8 GP ZGP323 evaluation board (see Figure 1 on page 3) as a target development board.

The 20-/28-/40-PDIP target pods plug into the associated PDIP sockets on the development platform. If you are using a different target development board, use an appropriate target pod and pin converter to connect the Z8 GP ICE to the board. For example, if your target board has a 20-SOIC socket, mate the 20-PDIP target pod onto the 20-PDIP to 20-SOIC converter. Then install the target pod and converter assembly into the board's 20-SOIC socket.

Note: *If you are not using a target development board, insert the 34-pin null target connector p/n 93c0086-001, into Z8 GP ICE target interface connector P16.*

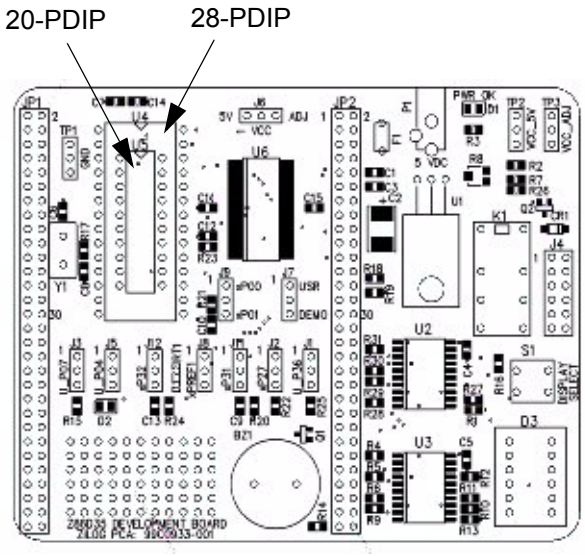

Figure 1. Z8 GP ZGP323 Evaluation Board

Connecting Z8 GP ICE to Target Pod

After installing the appropriate target pod (and converter, if required) onto the target development board, connect the Z8 GP ICE to the target pod as follows:

- For 40-PDIP target pod:
  - Connect the 16-circuit cable from P17 on the emulator to P17 on the 40-PDIP target pod.
  - Connect the 34-circuit cable from P16 on the emulator to P16 on the 40-PDIP target pod.
- For 20-/28-PDIP target pods:
  - Connect the 34-circuit cable from P16 on the emulator to P16 on the target pod. (Emulator connector P17 is not used).

Connecting Z8 GP ICE to OTP Programming Module (Optional)

After developing and debugging your software, follow the steps below to connect the Z8 GP ICE to the OTP programming module so that you can burn your code onto the OTP chip:

1. Connect the 40-circuit ribbon cable from the Z8 GP ICE OTP Programming connector to connector P1 on the OTP programmer module.
2. The 40-PDIP ZIF socket on the OTP programming module is designed to accept 40-PDIP OTP chips. The OTP programming adapters supplied with the Z8 GP ICE allow

you to adapt the ZIF socket to accept 20-/28-SOIC, 20-/28-/48-SSOP, and 20-/28-PDIP chip packages.

After installing the OTP chip into the ZIF socket (or programming adapter), you can program the chip using the instructions provided in OTP Programming on page 17.

Connecting Z8 GP ICE to a PC

Follow the steps below to connect the Z8 GP ICE to a host PC:

1. Connect a CAT-5 crossover cable from the PC to the Ethernet port on the Z8 GP ICE, see Figure 2.

Note: *You can also connect the emulator to an Ethernet hub using a standard CAT-5 patch cable.*

2. Connect the serial COM port on the PC to the SETUP serial port on the Crimzon ICE using the DB9-to-DB9 serial cable, see Figure 3.

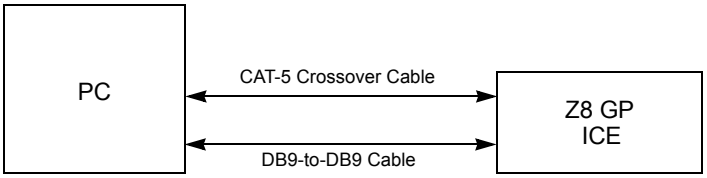

Figure 2. Connecting a PC to the Z8 GP ICE

Figure 3. Z8 GP ICE Rear Panel

3. Connect a 5 V DC power supply to the Z8 GP ICE. The 3.3 V DC power LED must illuminate, see Figure 3. Contact Zilog® support at www.zilog.com if there is any problem.

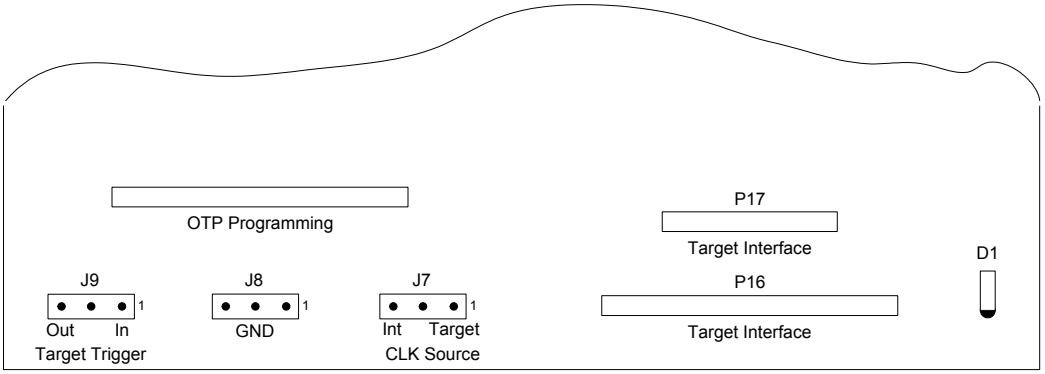

Figure 4. Z8 GP ICE Top View

Figure 5. Z8 GP ICE Front-Panel

Hardware Configuration

Z8 GP ICE configuration consists of selecting the emulator jumper options and setting up Ethernet communications between the emulator and your PC.

Jumper Settings on the Z8 GP ICE

There is one jumper on the Z8 GP ICE. Jumper J7 on the Z8 GP ICE allows you to select whether the emulator uses the target board clock or programmed using the programmable clock settings in ZDS II.

Table 2. Jumper J7 Settings on the Z8 GP ICE

Jumper Position	Function
1–2 (default)	Emulator uses the target clock.
2–3	Emulator uses the internal ZDS II programmable clock.

Setting Up Ethernet Communications

The default IP address and subnet mask of the Z8 GP ICE are 192.168.1.50 and 255.255.255.0 respectively. To enable communication between the PC running ZDS II and the Z8 GP ICE, you must either change the PC's Ethernet settings to match those of the Z8 GP ICE or vice versa. If using the PC in a stand-alone configuration, set the PC's IP address to 192.168.1.21 and its subnet mask to 255.255.255.0. For more details, see Changing the PC's Settings to Match the Z8 GP ICE on page 6. In a networked environment, set the Z8 GP ICE IP address and subnet mask to match the network setup. For more details, see Changing Z8 GP ICE Settings to Match the PC on page 10.

Changing the PC's Settings to Match the Z8 GP ICE

Follow the steps below to change the PC's settings:

Note: *The following instructions are for MS Windows XP. If your Windows OS is different, refer to your MS Windows OS online help for details.*

1. Open the **Windows Control Panel** and double-click **Network Connections** icon. The **Network Connections** window appears, see Figure 6.

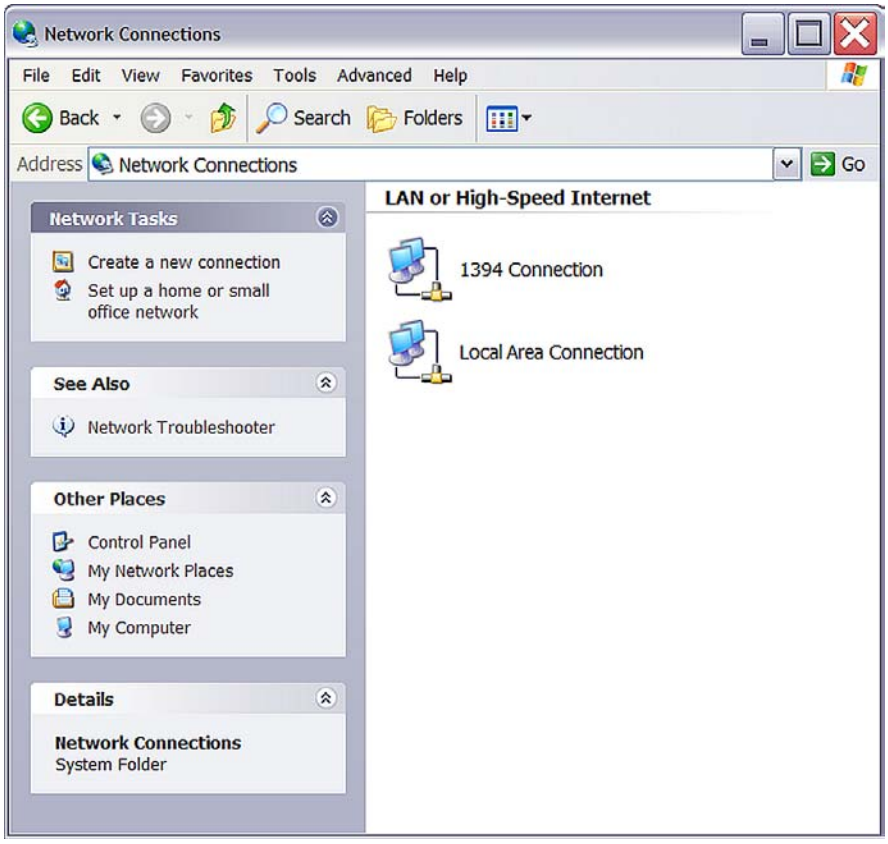

Figure 6. Network Connections Window

2. In the panel labeled **LAN or High-Speed Internet**, double-click **Local Area Connection** icon. The **Local Area Connection Status** window appears, see Figure 7.

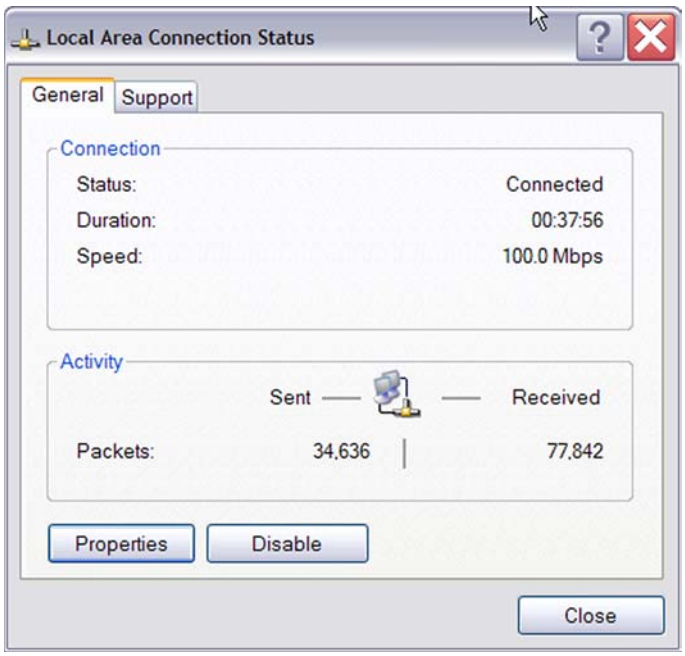

Figure 7. Local Area Connection Status Window

3. In the **Local Area Connection Status** window, click **Properties** button. The **Local Area Connection Properties** dialog appears, see Figure 8.

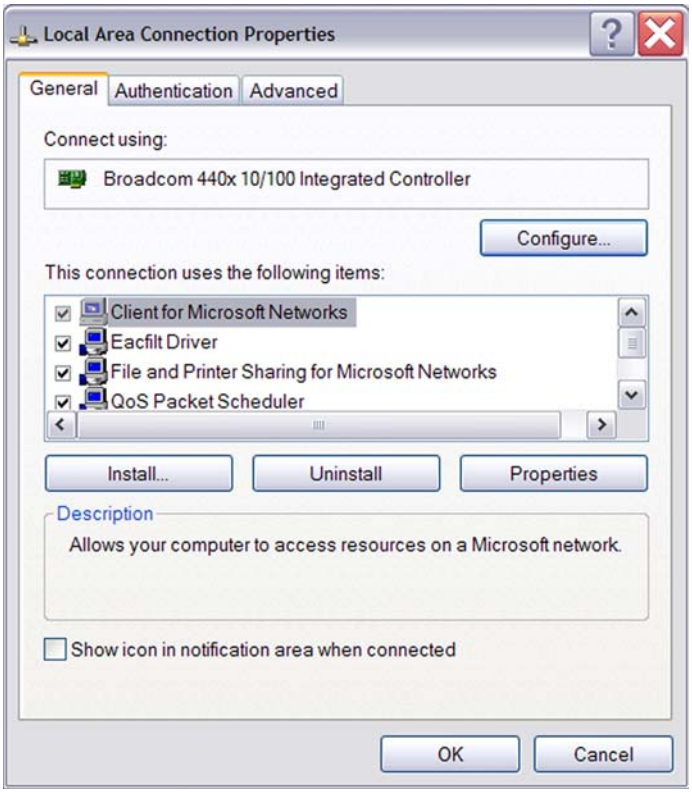

Figure 8. Local Area Connection Properties Dialog

4. Select **Internet Protocol (TCP/IP)** from the scroll down list, and click **Properties** button. The **Internet Protocol (TCP/IP) Properties** dialog appears, see Figure 9.

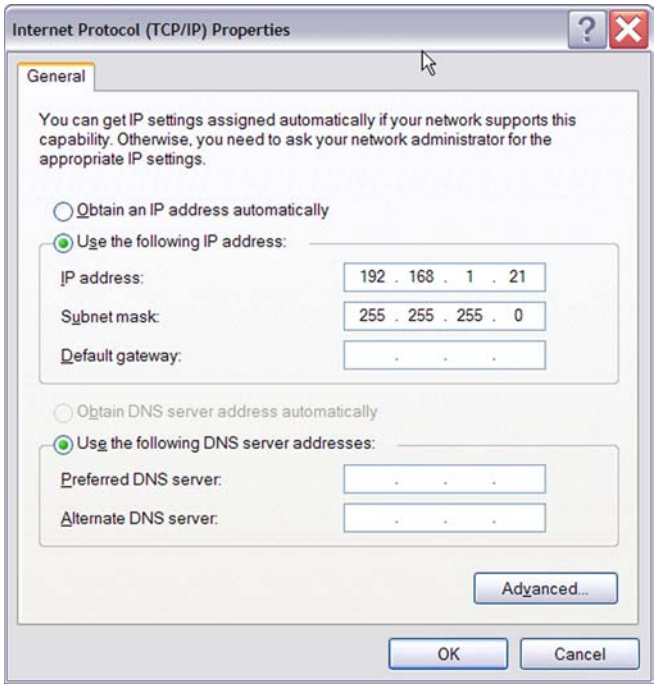

Figure 9. Internet Protocol Properties Dialog

5. Enter the IP address and subnet mask to match those shown in Figure 9. Leave the remaining fields blank. In this example, an IP address of 192.168.1.21 and a subnet mask of 255.255.255.0 are being assigned to the PC. These values connect the PC to the same network as the Z8 GP ICE unit.
6. Click **OK** and restart the PC.

Note: *To execute the sample project, see* Sample Project *on page 13*.

Changing Z8 GP ICE Settings to Match the PC

Follow the steps below to change the Z8 GP ICE Settings:

1. Connect the serial port of the PC to the Crimzon ICE serial port using the DB9-to-DB9 serial cable.
2. Launch HyperTerminal on the PC by selecting **Start** → **Programs** → **Accessories** → **Communications** → **HyperTerminal**. The **Connection Description** dialog appears.

3. Enter the name for a new connection in the **Connection Description** dialog, and click **OK** to open the **Connect To** dialog box.
4. In the **Connect To** dialog, set the **Connect Using** drop-down menu to match the COM port to which the Crimzon ICE is connected. Click **OK**.
5. A **COM Properties** dialog appears. Enter the following port settings and click **OK**. HyperTerminal should automatically attempt a connection. Otherwise, select **Call** → **Connect**.

Bits per second	57600
Data bits	8
Parity	None
Stop bits	2
Flow control	None

6. When the emulator is turned ON or reset, a Z8 GP ICE console boot-up message appears in the HyperTerminal. A typical boot-up message is provided below:

```
ZiLOG Z8 LXM ICE
Firmware Version 2.0, Build (Aug 22 2005 08:14:37)
Copyright (C) 2005 ZiLOG, Inc. All Rights Reserved.
Adding emac driver...
Attempting to establish Ethernet connection.
10 Mbps Half-Duplex Link established
IP Address: 10.1.7.95
IP Subnet: 10.1.0.0/255.255.0.0
IP Gateway: 10.1.1.254

Press 'Ctrl-Z' to enter configuration mode
```

7. Press Ctrl-z. The emulator command prompt appears:

   Z8 LXM ICE %

Note: *The emulator console prompt is not case-sensitive.*

Type `help` or `?` at the emulator command prompt to see the list of available commands. For information on the description of the complete Z8 GP ICE commands, see Appendix A—Z8 GP ICE Commands on page 23.

8. When you have finished configuring the emulator, type `exit` to exit the command shell.
9. Press **Alt**+**F4** to exit HyperTerminal.
10. Cycle the power on the Z8 GP ICE for the new settings to take effect.

The hardware is now configured and ready for application development.

Note: *To execute the sample project, see* Sample Project *on page 13.*

Sample Project

After installing the ZDS II software and setting up the hardware, you can execute the sample software project to verify proper emulator operation and to test with the Trace and Event system. This section describes how to run the emulator in the in-circuit mode.

Note: *If you run the emulator with a target attached, the emulator's voltage comparator is designed to serve as a target power sensor, and not as a precision voltage measurement device. If you set the Target VCC to match your target and the target's voltage drifts downward, the power sensor may no longer detect it. The emulator may therefore not connect to the target. In such cases, set the Target VCC voltage progressively lower until a proper connection is established.*

The sample project `ledblink_c.pro` is included in the ZDS II sample directory, located in:

```
c:\Program Files\ZiLOG\ZDSII_<product>_<version>\samples\<processor type>_<demo name>
```

Start ZDS II for the Z8 GP ICE Emulator by selecting **Start → ZiLOG ZDS II-Z8 GP Emulator Kit_**`<revision>` and follow the instructions below to run `ledblink_c.pro`, the sample project.

1. Use **File → Open Project** menu option to open the sample project file located at the following path:

   ```
   c:\Program Files\ZiLOG\ZDSII_Z8GP_Emulator_ <version>\samples\ZGP323_ledBlink\ledblink_c\src\ ledblink_c.pro
   ```

2. To open the source file, double-click main.c in the **Project Files** Window.
3. Select **Project → Settings**.
4. In the **General** tab, set the **CPU Family field** to ZGP323 and the **CPU field type** to ZGP323XXX2832.
5. In the **Debugger** tab select **Ethernet Driver** from the from the **Driver** drop-down menu and click **Configure** button.
6. The **Ethernet Configure Driver** dialog appears. The **IP Address** field displays a default IP address, 192.168.1.50. Enter the Z8 GP ICE IP address if it has been modified. Leave the Port setting at 4040.
7. Click **OK**.
8. In the **Debugger** tab, set the **Target** drop-down menu to ZGP323ICEXXZEM and click **Configure** button. The **Configure Target** window appears.
9. Set the **Voltage** drop-down menu to 3.0 V. If the emulator is connected to a custom target, select the voltage appropriate for the connected target.

10. In **Clock Source** section, select the **External** radio button.

Note: *Ensure that jumper J7 on the Z8 GP ICE is set with the shunt between 1-2 as described in* Table 2 *on page 6* *when executing the emulator in the incircuit mode.*

11. In the **Programming Option Bits** section, ensure that none of the options are selected.
12. Click **OK**.
13. Click **OK** in the **Project Settings** window and you will be prompted to rebuild the affected files, click **Yes** to rebuild the project. (You can also rebuild later by pressing F7.)

Note: *The following steps describe two ways to use the Trace and Event system. For more information on executing the Trace and Event system, refer to Zilog Developer Studio II—Crimzon and Z8 GP User Manual (UM0164) located in the docs directory of the ZDS II CD-ROM and ZDS II online help.*

Collecting a Trace

Follow the steps below to obtain a sample Trace:

1. Collect a simple trace by starting the program, stopping it, and viewing the Trace buffer. Click **Go** button in the toolbar, and then click **Break** button. The Trace buffer acts as a ring buffer that continuously fills and then overwrites itself until you stop execution.
2. Select the **Trace** window by selecting **View** → **Debug Windows** → **Trace** and click **Get Frames** to display the Trace information.

Using an Event to Stop Execution

Events allow you to stop execution based on more complex conditions than a simple instruction address.

The following events are available:

- Program counter position, with mask
- Data on Port0 (state of its pins), with mask
- Data on Port2 (state of its pins), with mask
- Data on Port3 (state of its three input pins), with mask
- External Trigger In (0 or 1)

Follow the steps below to setup and execute an event:

1. Select **Tools** → **Trace and Event System**. The **Trace and Event System** window appears, see Figure 10.

2. Check **Enable event system** check box and in the **Then:** section, check **Break** radio button.
3. In the **When:** section, check **Program Counter** check box and set **Program Counter** to 06B2 and **Mask** to FFFF.

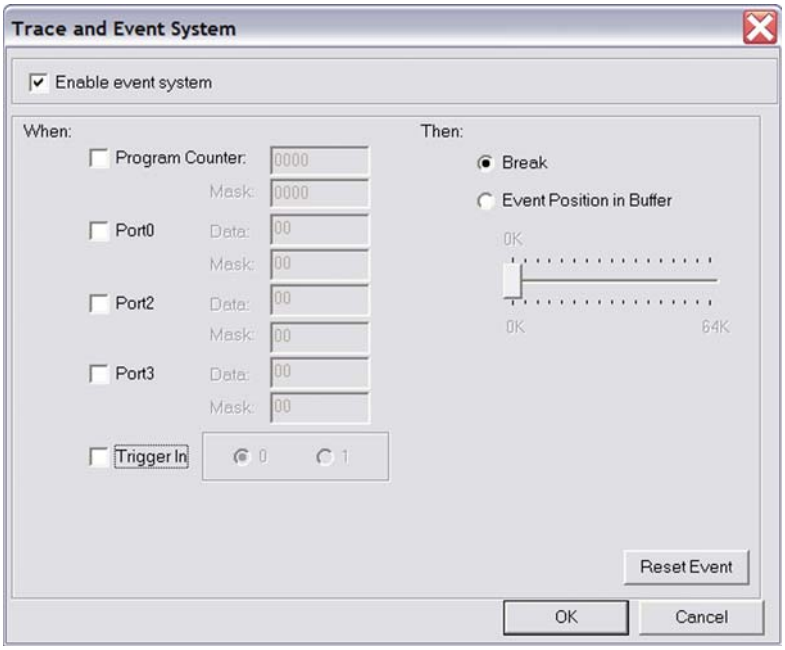

Figure 10. Trace and Event System Window

4. Click **OK**.
5. Open the Trace window by selecting **View** → **Debug Windows** → **Trace**.
6. In the **Trace** window, click **Clear Trace** button.
7. To reset the **Debugger** click **Reset** button in the toolbar, or select **Debug** → **Reset**.
8. Click **Go** button or select **Debug** → **Go** to run the **Debugger**.
9. When the program counter reaches 06B2, execution stops on event match.
10. Click **Get Frames** to display the Trace information.

Collecting Trace Before and After an Event

You can use the Trace and Event System to capture trace data before and after an event. Set up the events as described in Using an Event to Stop Execution on page 14. In the **Then:** section (see Figure 10), check **Event Position in Buffer** radio button instead of

Break. Use the slider bar to select the number of cycles from the 64 K buffer to be captured before and after the event. In this case, all cycles are traced until the event is detected, then the selected number of cycles after the event are collected. Execution stops after the cycles are collected. What remains in the trace buffer are the selected number of cycles after the event. The remainder of the 64 K frames contains cycles before the event occurred. If you move the slider completely to the left, only cycles before the event are captured. If you move the slider completely to the right, only cycles after the event are captured.

Single-Stepping Through a Program

ZDS II provides a simple mechanism for single-stepping through a program. Follow the steps below to single-step through a program:

1. Reset the program to `main()` by either clicking **Reset** icon or by selecting **Debug** → **Reset**. Set the Reset to `main()` option by selecting **Tools** → **Options**. In the **Options** window, select the **Debugger** tab and select the **Reset** to symbol 'main' check box.
2. To step through the program one instruction at a time, use F11 or click **Go** button in the **Debug** toolbar or select **Debug** → **Step Into**.

Peek/Poke Registers

Follow the steps below to read the emulator register contents:

1. ZDS II makes it easy for you to set and read emulator register con-tents. With the ledblink_c.pro project open and ZDS II connected to the emulator (target), select **View** → **Debug Windows** → **Registers**.
2. In the **Registers** window, double-click the value of any register and type in a new value.
3. Press **Enter**. The new value is displayed in red.

Refer to *Zilog Developer Studio II—Crimzon and Z8 GP User Manual (UM0164)* on the ZDS II CD-ROM and ZDS II online help for further information on setting and reading register values.

Peek/Poke Memory

Follow the steps below to set and read the peek/poke memory contents:

1. With the `ledblink_c.pro` project open and ZDS II connected to the emulator (target), select **View** → **Debug Windows** → **Memory**.
2. In **Memory** window, double-click the value you want to change and type in a new value. (Values begin in the second column after the **Address** column.)
3. Press **Enter**. The new value is displayed in red.

Refer to *Zilog Developer Studio II—Crimzon and Z8 GP User Manual (UM0164)* on the ZDS II CD-ROM and the ZDS II online help for further information on setting, filling, and reading memory.

OTP Programming

Use the Z8 GP ICE OTP Programming Module to burn your program onto a ZGP323 family chip. There are two ways to burn an OTP chip:

1. Burn Code from the Current Project
2. Burn Code from an Existing Hex File

Note: *Do not connect to the Z8 GP ICE when programming windowed CDIP parts. See* Burn Code from an Existing Hex File *on page 19 when programming windowed CDIP parts.*

Burn Code from the Current Project

Follow the steps below to burn code from the current project built in ZDS II (loaded in emulator RAM):

1. Connect the OTP programming module to the emulator as described in Connecting Z8 GP ICE to OTP Programming Module (Optional) on page 3.
2. Select the OTP chip to be burned and the appropriate package converter.
3. Install the package converter, if used, into the ZIF socket on the OTP programming module.
4. Install the OTP chip to be burned into the ZIF socket on the OTP programming adapter. Match pin 1 of the chip with pin 1 of the ZIF socket.
5. In ZDS II, open the project for the code to be burned onto the chip.

Note: *If you are in debugging mode, click* ***Debug*** → ***Stop Debugging*** *or press* ***Shift-F5*** *to stop debugging.*

6. Select **Tools** → **OTP Programming** to open the **OTP** window, see Figure 11.

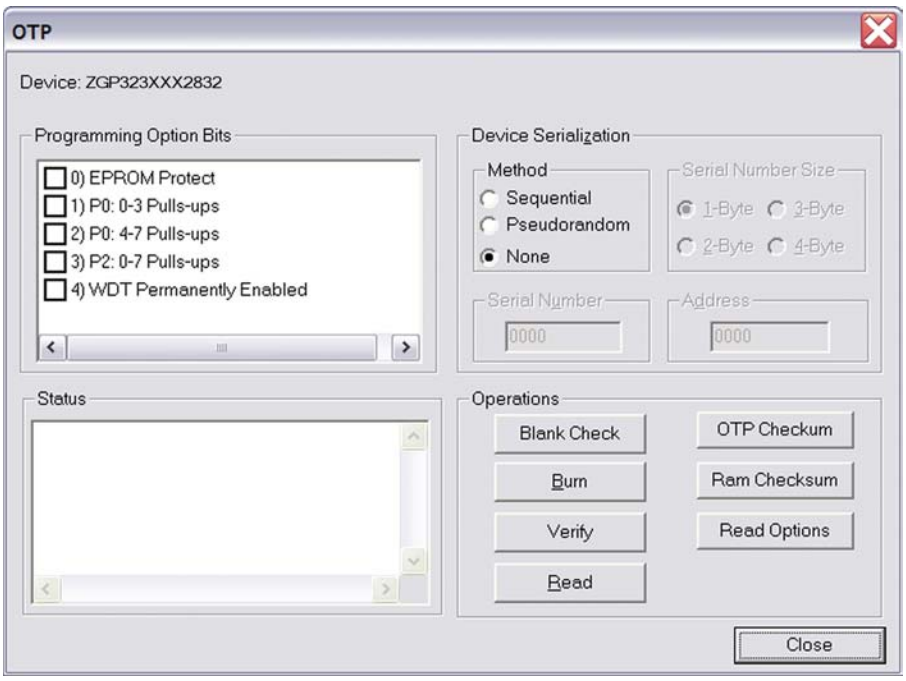

Figure 11. OTP Programming Window (Current ZDS II Project Example)

7. If the appropriate target device is not selected, select it from the **Device** drop-down menu.
8. If you do not want to pad the hex file, select the **None** button in the **Pad File With** area (see Figure 12 on page 20). Otherwise, select **FF**, **00**, or **Other** button. If you select **Other** button, type the hex value to pad the hex file with in the **Other** field.
9. Click **Ram Checksum** button to calculate the checksum of the data in emulator RAM. Use this to compare with the OTP checksum after burning.
10. Select the option bits to be programmed in the **Programming Option Bits** area.
11. Select **None** button in the **Method** panel of **Device Serialization** to leave the serial number blank.
12. To load a serial number:

- Select **Sequential** or **Pseudorandom** button in the **Method** panel. This determines how the serial number is incremented on subsequent burns.
- Select the size of the serial number (1, 2, 3, or 4 bytes) in the **Serial Number Size** area.
- Enter the starting serial number in the **Serial Number** field.

13. Click **Blank Check** to verify that the OTP chip is actually blank.
14. Click **Burn** to program the OTP chip with the contents of emulator RAM. The OTP chip content value is also verified.
15. When the burn is complete, click **OTP Checksum** to calculate the checksum of data on the OTP chip and compare it to the **RAM checksum** calculated earlier.
16. Click **Close** to close the OTP Programming window.

Burn Code from an Existing Hex File

Follow the steps below to load an existing hex file into emulator RAM and burn an OTP chip:

1. Connect the OTP programming module to the emulator as described in Connecting Z8 GP ICE to OTP Programming Module (Optional) on page 3.
2. Select the OTP chip to be burned and the appropriate package converter.
3. Install the package converter, if used, into the ZIF socket on the OTP programming module.
4. Install the OTP chip to be burned into the ZIF socket on the OTP programming adapter. Match pin 1 of the chip with pin 1 of the ZIF socket.

Note: *Stop any debugging process by selecting **Debug** → **Stop Debug**.*

5. In ZDS II, open the project for the code to be burned onto the chip.
6. Select **Tools** → **OTP Programming** to open the OTP window, see Figure 12 on page 20.

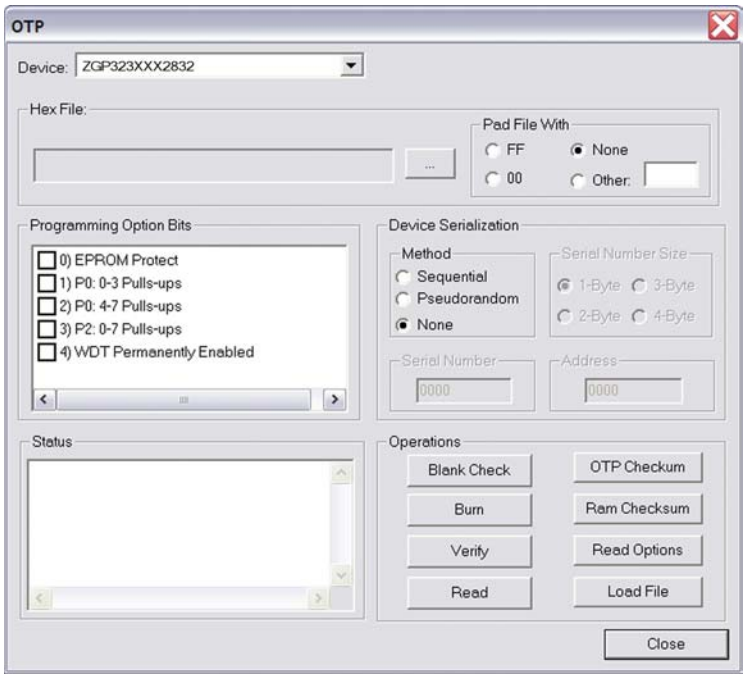

Figure 12. OTP Programming Window (Hex File Example)

7. Select the appropriate target device from the **Device** drop-down menu.
8. In the **Hex File:** section, click ... button and select the hex file to be programmed on to the OTP chip.
9. If you do not want to pad the hex file, select **None** button in the **Pad File With** panel. Otherwise, select **FF**, **00**, or **Other** button. If you select the **Other** button, type the hex value to pad the file with in the text field provided with **Other**.
10. Click **Load File** to load the hex file into emulator RAM.
11. Click **Ram Checksum** to calculate the checksum of the data in emulator RAM. Use this to compare with the OTP checksum after burning.
12. Select the option bits to program in the **Programming Option Bits** area.
13. Select **None** button in the **Method** panel of **Device Serialization** to leave the serial number blank.
14. To load a serial number:

- Select **Sequential** or **Pseudorandom** button. This determines how the serial number is incremented on subsequent burns.

- Select the size of the serial number (1, 2, 3, or 4 bytes) in the **Serial Number Size** area.
- Enter the starting serial number in the **Serial Number** field.
- In the **Address** field, enter the address of the serial number.

15. Click **Blank Check** to verify that the OTP chip is actually blank.
16. Click **Burn** to program the OTP chip with the contents of emulator RAM. The OTP chip contents is also verified.
17. When the burn is complete, click **OTP Checksum** to calculate the checksum of data on the OTP chip and compare it to the RAM check-sum calculated earlier.
18. Click **Close** to close the OTP Programming window.

LED Indicators

There are three sets of dual LED indicators on the Z8 GP ICE, as described below:

1. The dual ICE RUN LED on the front panel, see Figure 5 on page 5, indicates emulator status. If the top LED is ON, the emulator is executing the system code. When the top LED is off, emulation has stopped. If the bottom LED is ON, the emulator is not functioning properly, contact technical support for assistance.
2. The dual 3.3 V DC/1.8 V DC LED on the rear panel, see Figure 3 on page 4, indicates the status of internal voltages. Both LEDs are normally illuminated when power is connected.
3. The dual LAN/LINK LED on the rear panel, see Figure 3 on page 4, indicates Ethernet status. The LINK LED indicates that the Ethernet connection is live. The LAN LED indicates that data is being transferred across the connected network.

External Interface Connectors

There are five external interface connectors (P15, P16, P17, J8, and J9) on the Z8 GP ICE as explained below:

- Connectors P16 and P17 are used to connect the emulator to the target pod and adapter board assembly, see Figure 4 on page 5.
- The OTP Programming connector P15 is used to connect the emulator to the OTP programming module.
- Connector J8 on the emulator front panel (see Figure 4 and Figure 5 on page 5), provides a ground connection on all three pins.

- Connector J9 on the emulator front panel, see Figure 4 on page 5 and Figure 5 on page 5, provides access to the following functions:
  - Pin 3 provides a HIGH external trigger out for use in triggering a device such as a logic analyzer or oscilloscope. Pin 3 is under software control, and can be activated through the ZDS II Trace and Event system. The trigger can be set to toggle or pulse.
  - Pin 1 provides an input for an external HIGH or LOW trigger in, allowing use of an external trigger as an event for the ZDS II Trace and Event system.

Using J9 Pin 3, External Trigger Out

The Z8 GP ICE external trigger out feature is always enabled. Set your Trace and Event system parameters, before executing the code. When the set up event occurs, pin 3 of connector J9 goes HIGH and stays HIGH as long as the event is active. Longer the event window, longer the trigger out stays HIGH.

Using J9 Pin 1, External Trigger In

The Z8 GP ICE external trigger in feature can be enabled as follows:

1. With the ledblink_c.pro project open in ZDS II as described in Sample Project on page 13, select **Tools** → **Trace and Event System**.
2. In the **Trace and Event System** window, select an **Event** entry. In the **When** section, check **Trigger In** box.
3. Select either 0 or 1 to trigger on LOW or HIGH, respectively (edge-triggering not supported).
4. Click **OK** to set the **Trace and Event System** parameters. If you set trigger=1 in the Trace and Event System window, then a HIGH on pin 1 of connector J9 generates an event in the ZDS II **Trace and Event System**. If you set trigger=0, then a LOW on pin 1 of connector J9 generates an event.

Appendix A—Z8 GP ICE Commands

Table 3 lists the Z8 GP ICE Commands.

Table 3. Z8 GP ICE Commands

Command	Description and Options
?	Displays available emulator command shell options.
`debugport`	Configures the TCP port; usage: `debugport`—displays current setting `debugport tcp_port`—sets `debugport` to specified TCP port. **Example** `debugport` 4040—sets `debugport` to TCP port 4040.
`devs`	Not used.
`exit`	Exits the command shell.
`help`	Displays available emulator commands.
`ifconfig`	Configures the emulator network interface. Entering `ipconfig` with no options lists current configuration. The following command options are available: `i`—specifies an IP address `s`—specifies a subnet mask `g`—specifies a network gateway address `dhcp`—configures the emulator network interface to look for a dhcp host to obtain network settings **Example** `ifconfig i 192.168.1.1 s 255.255.255.0 g 192.165.1.254` configures the emulator to use IP address 192.168.1.1 on subnet 255.255.255.0 with gateway address 192.168.1.254.
`kbuf`	Not used.
`kill`	Not used.
`mem`	Not used.
`password`	Not used.

Table 3. Z8 GP ICE Commands (Continued)

Command	Description and Options
`ps`	Not used.
`reboot`	Reboots the emulator.
`restore`	Restores factory default network interface settings.

Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at http://www.zilog.com/kb.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at http://support.zilog.com.

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 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. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

ZiLOG:
ZGP323ICE01ZEM