eZ80190 Development Kit

External Flash Loader

PUG001203-0203 Product User Guide

Introduction

The Flash Loader utility integrated within ZiLOG Developer Studio (ZDSII) provides the ability to erase and program Flash memory on the target eZ80190 Module via the eZ80® Development Platform. The ZDSII Integrated Development Environment (IDE) provides an intuitive interface for loading Flash. This interface guides the user with a file developed for the ZDSII IDE, and with language tools developed for the eZ80®.

The target-based external Flash Loader utility allows the user to reprogram Flash memory inside an eZ80® processor on a stand-alone target system. In this system, neither the ZDSII IDE nor ZPAKII is used. This target-based, external use model is primarily designed for field upgrades. The Flash Loader software runs on the eZ80190 device and interacts with the user across a serial port to download a `.hex` file for burning into internal eZ80® Flash memory.

This document discusses the target-based, external Flash Loader utility. For a discussion of the ZDSII IDE Flash Loader utility, please see the ZiLOG Developer Studio II—eZ80 User Manual (UM0123).

Features

- Flash programming support for external Flash memory using ZDSII
- Flash support to program the write-protected Boot Block using ZDSII
- Flash support to program user code via the onboard serial console port
- Support to modify the MAC address in Flash using ZDSII

Requirements

Software

- The ZiLOG Developer Studio II (ZDSII)—Integrated Development Environment includes the following features:
  - Editor
  - C-Compiler
  - Assembler
  - Librarian
  - Debugger

- Flash Loader Utility
- Flash Boot Block Utility
- A terminal emulation program, such as HyperTerminal

Hardware

- eZ80® Development Platform
- One RS232 cable @ 57.6Kbps, 8-N-2
- ZPAKII Debug Interface Tool, including:
  - ZDI Target Interface Module (TIM)
  - One 40-pin cable to connect TIM to ZPAKII
- One Ethernet cable
- One Ethernet hub with power supply
- One 5V power supply for ZPAKII
- One 9V power supply for the eZ80® Development Platform

For demonstration purposes, a MAC address is provided on the bottom of the eZ80® Development Platform.

Caution: Always use a grounding strap to prevent damage resulting from electrostatic discharge (ESD).

Settings

Terminal

The eZ80® Development Platform requires the specific terminal settings for using the Flash Loader Utility. Configure these settings with the following brief instruction.

1. Connect the Host PC to the console port (P2) of the eZ80® Development Platform.
2. Set the COM properties to:
   - 57600 baud
   - 8 bits
   - No parity
   - 2 stop bits

Jumpers

The jumper and its settings are listed below.

J7--Flash Write Enable ($\overline{\text{FlashWE}}$)

- Installed--the first 16KB Boot Block is vulnerable to modification
- Removed--the first 16KB Boot Block is write-protected

If the Flash WE jumper (J7) is not installed, the user cannot program the Flash memory boot block.

Memory Map

The Flash Loader Utility resides in the boot block section of memory, in the address range 000000h–003FFFh. This 16KB block is protected by removing the FlashWE jumper (J7) and is vulnerable to change when installed. The MAC address resides within this protected block. The next two 8KB parameter blocks, in the address range 004000h–007FFFh, are reserved. User code resides in a section of memory beginning at address 008000h and ending at address 0FFFFFh. This user code section contains 992KB of free code space. See Figure 1.

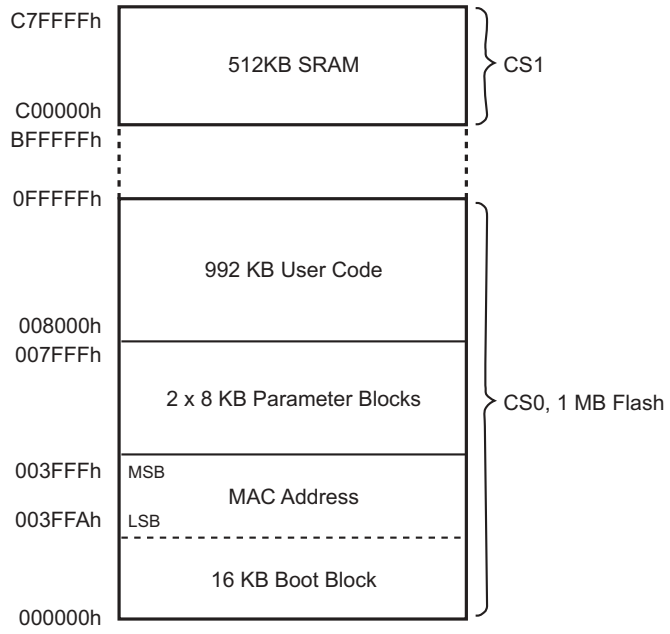

Figure 1. Memory Map

During reset, the eZ80® Development Platform maps Flash memory at address 000000h and places SRAM beyond the first 64KB of memory space. The Flash Loader changes the chip select configuration so that SRAM starts at address 000000h and Flash memory starts at address 800000h. As a result, the entire range of Flash memory is visible to the eZ80® CPU.

Flash and SRAM memories on the eZ80® Development Platform allocate the following address ranges while using the Flash Loader:

Flash Memory: 800000h-FFFFFFh (8MB, assuming all 8MB is physically present on the eZ80® Development Platform)

SRAM: 000000h-07FFFFh (512KB, limited to available SRAM on the eZ80® Development Platform, Rev B)

Note: The change in the memory range does not require configuring the user application to be built within this range. The Flash Loader manages these details on its own.

Installing the Flash Loader Utility

The eZ80® Development Platform is equipped with the Flash Loader Utility installed in the boot sector of Flash memory. In addition, the boot sector contains MAC address bytes (3FFAh--3FFFh).

If it subsequently becomes necessary to reinstall, update, or modify the Flash Loader utility, follow the steps in this section.

When reinstallation is complete, reboot the eZ80® with ZPAKII disconnected, and start the Flash Loader Utility with the eZ80® operating in STANDALONE mode.

Necessary Hardware

- eZ80® Development Platform
- ZPAKII , including:
  - ZDI Target Interface Module (TIM)
  - One 40-pin cable to connect TIM to ZPAKII
- One serial cable
- One Ethernet cable
- One Ethernet hub with power supply
- One 5V power supply for ZPAKII
- One 9V power supply for the eZ80® Development Platform

Procedure

The following steps demonstrate how to program the Flash Loader Utility into the protected Flash Boot Block.

Caution: Always use a grounding strap to prevent damage resulting from electrostatic discharge (ESD).

1. Connect ZPAKII to the eZ80® Development Platform using TIM and the ribbon cable.
2. Using the Ethernet cable and the hub, connect ZPAKII to the Network Interface Card of the PC running ZDSII.
3. Connect console port P2 of the eZ80® Development Platform to the COM port of the terminal PC.
4. Install the Flash WE jumper (J7).
5. Apply power to ZPAKII and to the eZ80® Development Platform.
6. Launch HyperTerminal**.** In the **Connect To** dialog of the HyperTerminal, select the COM port that connects to the eZ80® Development Platform. In the **Properties** dialog, set the COM port parameters to:
   - 57600 bps
   - 8 data bits
   - No parity
   - 2 stop bits
   - No flow control
7. From the File menu in ZDSII, choose **Open Project,** and navigate the path `..\Applications\eZ80190\flash_loader\190FlashLoader\190ZpakLoader`. Double-click the project file named `190DevPlat_Zpak_flash_Loader.pro`.
8. Choose **Project** → **Settings** and click the **Debugger** tab. Choose **ZPAKII** as the driver from the drop-down list and click **Configure ZPAKII.** In the **Configure ZPAKII** dialog box, enter the appropriate IP address and the port number for ZPAKII hardware. Click **OK** to close the **Configure ZPAKII** dialog box. Click **OK** to close the **Settings** dialog box.
9. In ZDS, choose **Connect** to establish communications with ZPAKII. Choose . **Download Code** to begin downloading the code to eZ80® Development Platform (alternatively, choose Reset to perform the same operation). A progress bar indicates the status of the download process.

10. After the download is complete, choose **Debug** → **Go** from the **Build** menu in ZDSII. The program runs and the following output appears in the HyperTerminal window:

```
Reset
eZ80 Flash Loader Utility
Version x.xx ZDS

Enter Mac Address >
```

Note: `x.xx` denotes the current version of ZDSII.

11. In the HyperTerminal window, enter the MAC address located on a label on the bottom of the eZ80® Development Platform. This MAC address is provided for demonstration purposes only.

Note: No backspace characters are allowed when entering the MAC address. In the event of a keyboard input error, enter any string of characters until the program prompts the user. At the prompt, press *M* to cause the `Enter MAC Address >` prompt to reappear.

Example Output

```
Enter Mac Address > 00:90:23:00:00:00
Type 'H' for help
eZ80190>
```

Note: The user can enter *H* at any time to list the available Flash Loader commands.

12. Press *L* to prepare the eZ80® Development Platform to receive code.

    If the $\overline{\text{FlashWE}}$ jumper (J7) is not installed, the user cannot program the Flash boot block.

Example Output

```
eZ80190> L
Start sending file via Xmodem protocol...
```

13. Select the eZ80190 Flash Loader hex file from the following path:

    `..\Applications\eZ80190\flash_loader\190FlashLoader\190Boot Loader\190DevPlat_eZ80flash_loader.hex.`

Choose the **Send File** option of HyperTerminal's **Transfer** menu and send the file using the XMODEM protocol.

A brief delay ensues before the download begins (approximately 15 seconds).

The Flash Loader Utility completes loading into the protected boot sector of Flash.

Note: Make sure that the $\overline{\text{FlashWE}}$ jumper (J7) is installed when programming the boot sector. This FlashWE jumper is intended only for the boot sector. When the Flash Loader Utility is programmed, the user can remove the jumper J7 before programming the remainder of Flash memory, to ensure that the boot block is protected from any accidental overwriting.

When the transfer is complete, the user is prompted with the following output:

```
Done
eZ80190>
```

14. Remove the $\overline{\text{FlashWE}}$ jumper, J7.
15. Stop ZDS and disconnect the ZPAKII from the eZ80® Development Platform.

The Flash Loader Utility now resides in the write-protected boot block of Flash memory. The user can now boot from Flash and load Flash without ZDSII.

Loading User Code

After the Flash Loader Utility code is written to Flash memory, it can be used to load the user code, starting at address `8000h`. For proper execution, the `.hex` file downloaded onto Flash must be properly built for the eZ80® Development Platform's memory configuration.

Necessary Hardware

- eZ80® Development Platform
- Power Supply
- Serial Cable

Procedure

The following steps demonstrate how to load the user code onto the eZ80® Development Platform.

Caution: Always use a grounding strap to prevent damage resulting from electrostatic discharge (ESD).

1. Connect console port P2 to the COM port of the terminal PC.
2. Ensure that jumper J7 is removed to disable Flash WRITE in the boot sector.
3. Apply power to the eZ80® Development Platform.
4. Launch HyperTerminal. In the **Connect To** dialog of HyperTerminal**,** select the COM port that connects to the eZ80® Development Platform. In the **Properties** dialog box, set the COM port parameters to the following values.
   - 57600 bps
   - 8 data bits
   - No parity
   - 2 stop bits
   - No flow control
5. Hold down the space bar of the terminal PC and reset the eZ80® Development Platform.

   The Flash Loader Utility checks the serial port for a space character when it is reset. If a space character is sampled when reset, the eZ80® CPU boots into the Flash Loader. If no space character is sampled at reset, the eZ80® CPU jumps to the user application at address `8000h`.

Example Output

```
Reset
eZ80 Flash Loader Utility
Version x.xx eZ80
Type 'H' for help
eZ80190>
x.xx denotes the current version.
```

In the example above, Flash Loader Utility is booted from the Flash boot block. If ZDSII initiates the program execution, `ZDS` appears following the version number, instead of `eZ80`.

Note: The user can enter *H* at any time to list the available Flash Loader commands.

6. Press *L* to load the user application into Flash memory.

Example Output

```
eZ80190> L
Start sending file via Xmodem protocol...
```

7. Select the appropriate `.hex` file. Choose the **Send File** option of the **Transfer** menu in the HyperTerminal and send the file using the XMODEM protocol. Make sure that the jumper J7 is removed when programming user code. This is to protect the boot sector from any accidental overwriting.

Example Output

```
Done
eZ80190>
```

The user code section now contains code and is ready to run.

8. Reset the eZ80® Development Platform to run user code starting at address `8000h`.

Demonstration Application

In order to demonstrate functions that can be used to write a user application, the Flash Loader Utility contains a Demonstration Application (DemoApp).

The DemoApp showcases the following:

1. It shows the arrangement for the Timer0, UART0, and UART1 interrupts. The entries for these interrupts feature associated ISR handlers. All the other entries for unused interrupts feature null ISRs.
2. The startup code shows stack initialization and chip select programming.
3. The main function (see main.c file) initializes the interrupt table with null ISRs for all interrupts (see interrupt.c file). Next, the timer interrupt initialization (see timer.c file) and the UART interrupt initialization (see stream.c file) routines place their ISR handlers into their respective entries in the interrupt table.
4. It shows how to handle RSTs and NMI (here, NMI is simulated using CALL %66).

The DemoApp files are located in the following paths:

- `..\Applications\eZ80190\flash_loader\Demo\190`. This directory contains eZ80190 processor-specific files only.
- `..\Applications\eZ80190\flash_loader\Demo\CommonFiles`. This directory contains files that are common to demo applications for other eZ80® processors.

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

ZiLOG Worldwide Headquarters
532 Race Street
San Jose, CA 95126
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Document Disclaimer