eZ80Acclaim!™ Development Kits

External Flash Loader

PUG001603-1003 **Product User Guide**

Introduction

This Product User Guide describes how to use the External Flash Loader application with eZ80Acclaim!™ microcontrollers.The External Flash Loader is a target-based application that programs a user application into the external Flash device present on the eZ80® Development Platform when using serial communication protocols and a terminal application such as a HyperTerminal. The External Flash Loader application is written for the Micron Flash device present on the eZ80® Development Platform.

This External Flash Loader application should not be confused with the Integrated Flash Loader available with the ZDS II-IDE (selected with the **Tools** → **Flash Loader** menu option). Although both of these Flash Loader applications are used for the same purpose (programming an external Flash device), the target-based External Flash Loader described in this Product User Guide is designed for the following customer-driven scenarios:

- When manufacturing a product requires download of an executable image into a final product
- When updating an executable image after the product is released to their customer base (field upgrades)
- When supporting other types of Flash devices is required. This support is made possible by merely replacing the External Flash Loader's Flash driver library with the customer's own version

Features

The External Flash Loader application offers the following features:

- Provides a simple mechanism for burning the user application's executable (`.hex`) file using ZDS II-IDE[1], and HyperTerminal via serial communication
- Two build configurations:
  - Debug configuration, in which the user can download the External Flash Loader application into SRAM on the eZ80® Development Platform using ZDS II-IDE, and execute the application from within SRAM
  - Release configuration, in which the user can download the External Flash Loader application to internal Flash memory on an eZ80Acclaim!™ MCU, using the Integrated Flash Loader already present in ZDS II. When the

1. ZDS II-IDE is the ZiLOG Developer's Studio, an integrated development environment.

External Flash Loader application is loaded into internal Flash memory, it can subsequently be used without ZDS II

- Upon RESET, the External Flash Loader application is capable of executing the user application residing in the external Flash
- Provides a method to set the EMAC address required for TCP/IP stack users
- Uses the XModem protocol for transfer of user code via serial communication, which makes the transfer fast and reliable

Note: The External Flash Loader application source files are included in the release; these files can be used to make additional modifications.

Requirements

Software

- The ZiLOG Developer Studio II—Integrated Development Environment includes:
  - Editor
  - C-Compiler
  - Assembler
  - Librarian
  - Debugger
  - Integrated Flash Loader utility
- A terminal emulation program, such as HyperTerminal

Hardware

- eZ80® Development Platform
- One RS-232 cable
- ZPAK II, including:
  - ZDI Target Interface Module (TIM)
  - One 40-pin ribbon cable to connect the TIM to ZPAK II
- One Ethernet cable
- One Ethernet hub with power supply
- One 5V 1000 mA power supply for ZPAK II
- One 9V 1200 mA power supply for the eZ80® Development Platform

Note: An EMAC address is silkscreened on the bottom of each eZ80® Development Platform. Use this unique address while testing the External Flash Loader application.

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

Settings

Terminal

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

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
   – No flow control

Jumpers

The ability to write to Flash memory must be enabled via the Flash Write Enable jumper, J7. The two possible settings for this FlashWE jumper are:

- Removed—the first 16KB boot block is write-protected
- Installed—the first 16KB boot block is vulnerable to modification

If the $\overline{\text{FlashWE}}$ jumper is not installed, the boot block of Flash memory cannot be programmed.

Note: All other blocks in Flash memory are vulnerable to changes.

Memory Map

The eZ80® Development Platform features 1MB of external Flash memory that is tied to the first chip select (CS0) of the eZ80Acclaim!™ microcontroller. Flash memory is provided by the Micron MT28F008B3 Flash device (see the appropriate Micron specification for more details), which contains a 16KB boot block at its lowest address. This block is hardware write-protected when the J7 jumper is removed.

The next two 8KB blocks are the parameter blocks. The parameter blocks are followed by eight user code blocks. Of these, the first 96KB user code block is followed by the remaining seven 128KB blocks. The memory map for the Micron Flash device is depicted in Figure 1.

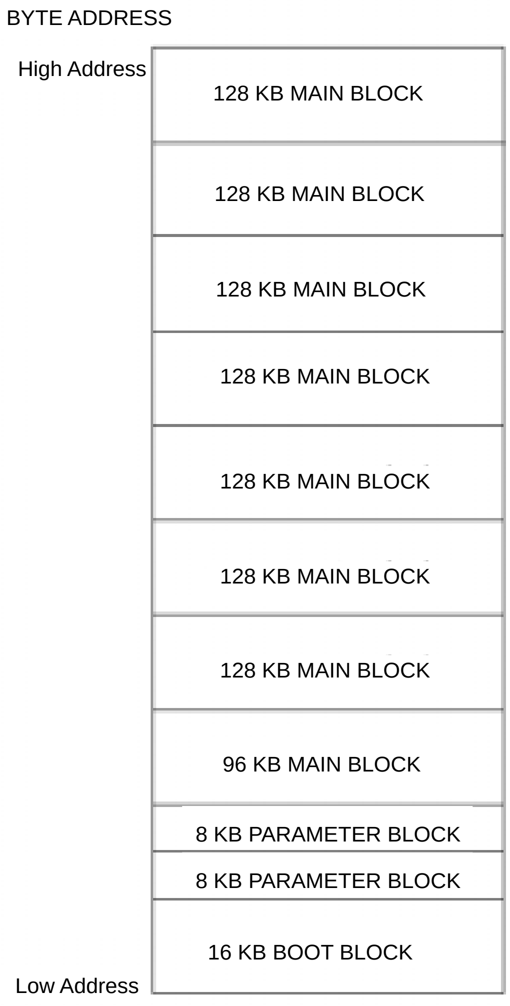

Figure 1. Micron Flash Device Memory Map

The memory map used for the two build configurations that are available for the External Flash Loader application, Debug and Release, are explained in the following section.

Debug Configuration

The Debug configuration of the External Flash Loader is useful for debugging the application when modifications are made either to the External Flash Loader or the user application to be programmed into Flash.

When building the External Flash Loader with the Debug configuration, internal Flash memory is disabled. The external SRAM occupies the lowest address space and external Flash memory is moved to the higher address space. The External Flash Loader code is downloaded to SRAM using ZDS II, via a ZPAK II Debug Interface tool (ZDI). For the eZ80F91 MCU, the memory spaces, sizes, and ranges are set to the values shown in Table 1.

Table 1. Debug Configuration Memory Spaces (eZ80F91)

Memory	Size	Range
Flash (internal)	disabled	—
SRAM	512 KB	0x000000–0x07FFFF
Flash (external)	1 MB	0x800000–0x8FFFFF

Release Configuration

The Release configuration of the External Flash Loader is used to generate the `.hex` file that resides within internal Flash, and it is used to program the user application into external Flash on the eZ80® Development Platform.

When building the External Flash Loader with the Release configuration, internal Flash memory is enabled, and it occupies the lowest address space. External SRAM and external Flash memory are moved to the higher address space. The External Flash Loader code is downloaded to internal Flash memory. For the eZ80F91 MCU, the memory spaces, sizes, and ranges are set to the values shown in Table 2.

Table 2. Release Configuration Memory Spaces (eZ80F91)

Memory	Size	Range
Flash (internal)	256	0x000000–0x03FFFF
SRAM	512 KB	0xC00000–0xC7FFFF
Flash (external)	1 MB	0x800000–0x8FFFFF

Building the External Flash Loader Application

This section describes how to build the External Flash Loader application with the Debug and Release configurations. To build the External Flash Loader application in either of the two configurations, follow the steps provided below.

1. Launch ZDS II. From the **File** menu in ZDS II, choose **Open Project**, and navigate to the path `..\FlashApp\FlashLoaderApp\src`. Double-click the project file pertaining to the target (for example: `F91_Flash_Loader.pro`).
2. Go to **Build** → **Set Active Configuration** and select either **Debug–FlashLoader/Release–FlashLoader** from the list as the active build configuration.
3. Choose **Build** → **Rebuild All** to build the project. The final executable file is generated in the path: `..\FlashApp\FlashLoaderApp\src`. If the chosen active build configuration is **Debug–FlashLoader**, a `.lod` file is generated. If the chosen active build configuration is **Release–FlashLoader**, a `.hex` file is generated.

Note: While building the Debug configuration, the Flash Library[1] file, `FlashLoaderD.lib`, is included, and while building the Release configuration, the Flash Library file, `FlashLoader.lib`, is included.

The chip select settings for the Debug and Release configurations are provided in Tables 3 and 4.

Table 3. Chip Select Setting for the Debug Configuration

Internal Flash	Disabled		
CS0 (external Flash)	Lower Bound = 80;	Upper Bound = 8F;	Control Reg = E8
CS1 (external SRAM)	Lower Bound = 00;	Upper Bound = 07;	Control Reg = 28
CS2	Lower Bound = 00;	Upper Bound = 00;	Control Reg = 00
CS3	Lower Bound = 00;	Upper Bound = 00;	Control Reg = 00

Table 4. Chip Select Setting for the Release Configuration

Internal Flash	Enabled;	Flash Page = 00	
CS0 (external Flash)	Lower Bound = 80;	Upper Bound = 8F;	Control Reg = E8
CS1 (external SRAM)	Lower Bound=C0;	Upper Bound=C7;	Control Reg=28
CS2	Lower Bound=00;	Upper Bound=00;	Control Reg=00
CS3	Lower Bound=00;	Upper Bound=00;	Control Reg=00

1. The Flash Library is functionally a Flash device driver that is used to program into Micron's Flash device. The Flash Library is available in two configurations-Debug and Release. See the Flash Library API Reference Manual (RM0013) for more details.

External Flash Loader Installation

To install the Flash Loader application into internal Flash memory on the eZ80Acclaim!™ MCU, follow the steps provided below.

1. Build the Release configuration of the External Flash Loader as discussed in the Building the External Flash Loader Application sectionon page 6.
2. Connect ZPAKII to the eZ80® Development Platform using the TIM and the ribbon cable.
3. Using the Ethernet cable and the hub, connect ZPAKII to the Network Interface Card of the PC running ZDSII.
4. Apply power to the ZPAKII unit and to the eZ80® Development Platform.
5. In ZDSII, choose **Project** → **Settings** and click the **Debugger** tab. Click **Configure ZPAKII**. In the **Configure ZPAKII** dialog box, enter the appropriate IP address and the port number for the ZPAKII hardware. Click **OK** to close the **Configure ZPAKII** dialog box. Click **OK** to close the **Settings** dialog box.
6. Choose the **Flash Loader** option from the **Tools** menu. The **Flash Loader Processor** dialog box appears. Note that the message `[Info]Internal Flash Detected` appears in the **Status** field.
7. From the **Flash Options** panel, choose **Internal**.
8. In the **File** option, browse for the `.hex` file generated during the Build procedure.
9. Set **Internal Base Offset** to **0**.
10. Check the **Erase Flash Before Burning** option so that the Integrated Flash Loader erases the internal Flash before programming.
11. Click the **Fast Burn** button in the **Flash Controls** panel. The Integrated Flash Loader first erases the internal Flash device and then programs the chosen `.hex` file into the internal Flash device.

As a result of the preceding steps, the Release configuration of the External Flash Loader is programmed into internal Flash on the eZ80Acclaim!™ MCU.

Loading User Code

The procedure for programming external Flash memory using the Debug and Release configurations is explained in this section.

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

Debug Configuration

The Debug configuration of the External Flash Loader requires ZDS II-IDE and a ZPAK II Debug Interface tool for programming the external Flash device.

Procedure

1. Connect ZPAK II to the eZ80® Development Platform using the TIM and the ribbon cable.
2. Using the Ethernet cable and the hub, connect ZPAK II to the Network Interface Card of the PC running ZDS II.
3. Connect console port P2 of the eZ80® Development Platform, to the COM port of the terminal PC.
4. Install the $\overline{\text{FlashWE}}$ jumper (J7) on the eZ80® Development Platform, to unprotect the boot block.
5. Apply power to the ZPAK II unit and to the eZ80® Development Platform.
6. Launch **HyperTerminal.** In the **Connect To** dialog box of the **HyperTerminal**, select the COM port connected to the eZ80® Development Platform. In the **COM 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 ZDS II, choose **Open Project,** and navigate to the path `..\FlashApp\FlashLoaderApp\src`. Double-click the project file pertaining to the target (example: `F91_Flash_Loader.pro`).
8. Choose **Project** → **Settings** and click the **Debugger** tab. Choose **ZPAK II** as the driver from the drop-down list and click **Configure ZPAK II.** In the **Configure ZPAK II** dialog box, enter the appropriate IP address and the port number for the ZPAK II hardware. Click **OK** to close the **Configure ZPAK II** dialog box. Click **OK** to close the **Settings** dialog box.
9. In ZDS II, choose **Connect** to establish communications with ZPAK II. Choose **Download Code** to begin downloading code to the eZ80® Development Platform (alternatively, choosing the **Reset** command also performs the above operations). A progress bar indicates the status of the download process.
10. After the download is complete, choose **Debug** → **Go** from the **Build** menu in ZDS II. The program is executed and the following example output appears in the **HyperTerminal** window:

```
Reset
eZ80 Flash Loader Utility
Version x.xx ZDS
Type 'H' for help
eZ80F91>
```

(x.xx denotes the current version)

11. Press **L** to begin the process to program the user `.hex` file into the external Flash device. Remember that the boot block of the Flash can be programmed only if the Write Enable jumper (J7) is installed. The following output appears on the HyperTerminal screen:

    ```
    eZ80F91> L
    Start sending file via Xmodem protocol...
    ```

12. In the **HyperTerminal** window, go to **Transfer** → **Send File.** The **Send File** dialog box appears. Select the `.hex` user program from the **Filename Browse** option, and select **XMODEM** for the **Protocol** option. Click the **Send** button. A brief delay (approximately 15 seconds) ensues before the download begins. The Flash Loader utility programs the user application into the external Flash memory.
13. When the transfer is complete, the user is prompted with the following (sample) output:

    ```
    Done
    eZ80F91>
    ```

14. Remove the Write Enable jumper (J7).
15. To view the contents of Flash memory, use the **Display Memory** option in the Flash Loader menu.
16. Exit ZDS II and disconnect ZPAK II from the eZ80® Development Platform.

The user code now resides in the external Flash memory.

Release Configuration

The Release configuration of the External Flash Loader is a target-resident utility. It resides in the internal Flash memory of the eZ80Acclaim!™ MCU and programs the user code into the external Flash device.

Procedure

1. Connect console port P2 of the eZ80® Development Platform to the COM port of the terminal PC.

2. Install the $\overline{\text{FlashWE}}$ jumper, J7, onto the eZ80® Development Platform to unprotect the boot block.
3. Apply power to the eZ80® Development Platform.
4. Launch **HyperTerminal**. In the **Connect To** dialog box, select the **COM** port that the eZ80® Development Platform is connected to. In the **COM Properties** dialog box, set the COM port parameters to:
   - 57600 bps
   - 8 data bits
   - No parity
   - 2 stop bits
   - No flow control

5. Press and hold the spacebar on the PC and push the RESET button on the bottom of the eZ80® Development Platform. The External Flash Loader utility checks the serial port for a space character when it is RESET. If a space character is sampled, the eZ80® CPU boots into the External Flash Loader. If no space character is sampled at RESET, the eZ80® CPU jumps to the user application. The following sample output appears:

   ```
   Reset
   eZ80 Flash Loader Utility
   Version x.xx eZ80
   Type 'H' for help
   eZ80F91>
   ```

   (x.xx denotes the current version of the Flash Loader utility)

6. Follow Steps 11 to 16 of the Debug Configuration section to load the user program.

The user code now resides within external Flash memory.

Executing the User Code

At RESET, control is transferred to the External Flash Loader. If the spacebar character is not detected, the Flash Loader jumps to external Flash memory at location `0x000000`. The Flash Loader can be configured to jump to a different location (for example, to the starting address of the user application) by changing this location. This jump location can be changed only at compile time and not at run time. The symbol **START_ADDRESS** is defined for this purpose in the `key-scan.asm` file, which resides in the path `..\FlashLoaderApp\src`.

After changing the start location, the target-specific project in the above path must be rebuilt before using the External Flash Loader. Subsequently, the External Flash Loader disables internal Flash memory.

Notes: When control is transferred to the user application within external Flash memory, the Flash Loader sets the chip select CS0 to occupy the memory range `0x000000–0x0FFFFF`.

Programming the EMAC Address

Many user applications, especially those that contain a TCP/IP stack, require an EMAC address.

An EMAC address can be included as part of the application code or programmed separately into Flash memory. The External Flash Loader (Debug/Release configurations) not only allows the user to program EMAC address bytes, but also allows the user to modify the EMAC address when required.

In addition, while programming the EMAC address bytes, the External Flash Loader ensures that all other data in the block remain intact. The procedure for programming the EMAC address is provided below.

Procedure

1. Follow the procedure described in the Loading User Code section on page 7 to display the `eZ80F91>` prompt in the **HyperTerminal** window.
2. Enter **M** to program the EMAC address. A sample output is shown below.

```
eZ80F91> M
Enter Mac Address >
```

3. In the **HyperTerminal** window, enter the EMAC address. For best results, use the EMAC address printed on the bottom of the eZ80® Development Platform.

Note: The External Flash Loader automatically inserts a colon between address bytes to make them more readable. No backspace characters are allowed when entering the EMAC address. In the event of a keyboard input error, enter a non-hex character to quit programming the EMAC address. 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
eZ80F91>
```

When programming the EMAC address is complete and the External Flash Loader is ready to accept a new command, it displays a prompt in the HyperTerminal window. If an error occurs, the External Flash Loader displays an appropriate error message before prompting for a new command.

Changing the EMAC Address Location

By default, the Flash Loader programs the EMAC address to start at address location `0x3FFA`. The Flash Loader can be configured to program the EMAC address to start at a different location. This new location can be specified only at compile time and not at run time.

Table 5 lists the macro definitions that are defined in the `main.h` file, which can be found in the path `..\FlashLoaderApp\header`. These macro definitions are used while programming the EMAC address.

Table 5. Macro Definitions for EMAC Programming

Parameter	Description	Default Value
EMACADDRESSBASE	Location in the Flash memory	0x003FFA
EMACBLOCK_SIZE	Size of the block to which location belong	0x4000
EMACBLOCK_BASE	Start address of the block	0x000000

To change the EMAC address location during compile time, all of the above macro definitions must be changed. For example, to change the locations to accommodate the range `0x007FFA`–`0x007FFF`, the macro definition values are:

```
EMACADDRESSBASE  =  0x007FFA
EMACBLOCK_SIZE   =  0x2000
EMACBLOCK_BASE   =  0x006000
```

See the Memory Map section on page 3 for details about the Flash device memory map.

After changing the EMAC address location, the target-specific project in the path `..\FlashLoaderApp\src` must be rebuilt before using the External Flash Loader.

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