



**ZNEO32! Family of Microcontrollers**

**ZNEO32!® Flash Loader**

**User Manual and Installation Guide**

UM028001-1016

**ZNEO32!**  
*32 Bit Microcontrollers*



**Warning:** DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

---

## **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**

©2016 Zilog, Inc. All rights reserved. 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concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded.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.

ZNEO32! is a trademark or registered trademark of Zilog, Inc. All other product or service names are the property of their respective owners.

## Revision History

Each instance in this document's revision history reflects a change from its previous edition. To learn more, refer to the corresponding page(s) or appropriate links furnished in the table below.

<b>Date</b>	<b>Revision Level</b>	<b>Description</b>	<b>Page</b>
Oct 2016	01	Original issue.	n/a

## Overview

Zilog's ZNEO32! Flash loader features the ability to write to Flash and inspect different parts of the device. It offers a Production mode to enable programming of multiple devices at the same time. This Flash loader also includes the ability to individually serialize every part automatically.

The Flash loader offers two modes:

- Interactive mode – provides complete access to the device
- Production mode – designed to automate programming devices using a configuration project file

The ZNEO32! parts have a built-in Stage 1 boot loader. This Stage 1 boot loader uses the UART0 and SPI0 peripherals to receive the Stage 2 code and writes it directly to the SRAM. After the code has been uploaded to SRAM, the device starts the execution of the Stage 2 code. The Flash loader utilizes the boot loader to download the Stage 2 code, which includes Flash algorithms and controls. The devices can be placed into the boot loader mode by pulling the Boot Mode pin to ground. When the Stage 1 boot loader is not active, the UART0 and SPI0 are available for normal use.

## System Requirements

The following operating systems are supported:

- Windows 7 (32 bit)
- Windows 7 (64 bit)
- Windows 8 (64 bit)
- Windows 8 (32 bit)
- Windows 10 (32 bit)
- Windows 10 (64 bit)

The following operating systems are not supported; however, they are expected to work:

- Windows XP (32 bit)
- Windows XP (64 bit)
- Windows Vista (32 bit)
- Windows Vista (64 bit)

Minimum system requirements for the ZNEO32! Flash loader include:

- 1 GB RAM
- 1 MB hard drive space available (for executable)
- VGA
- USB or RS232 connector\*

---

► **Note:** \*A USB to UART converter or RS232 connector is required for connection to the ZNEO32! devices, depending on the hardware and the configuration. The system must be able to connect to the hardware using standard Windows Asynchronous Communication (COM ports), which may require drivers to be installed in Windows.

---

## Operation

This section describes the Flash loader operations.

### Interactive Mode

The Flash loader's Interactive mode provides the ability to perform the following functions:

- Program Flash
- Verify Flash
- Erase pages or the entire chip
- Write data at a specific location
- Read Flash
- Read SRAM
- Read registers

The Interactive mode screen is shown in Figure 1.

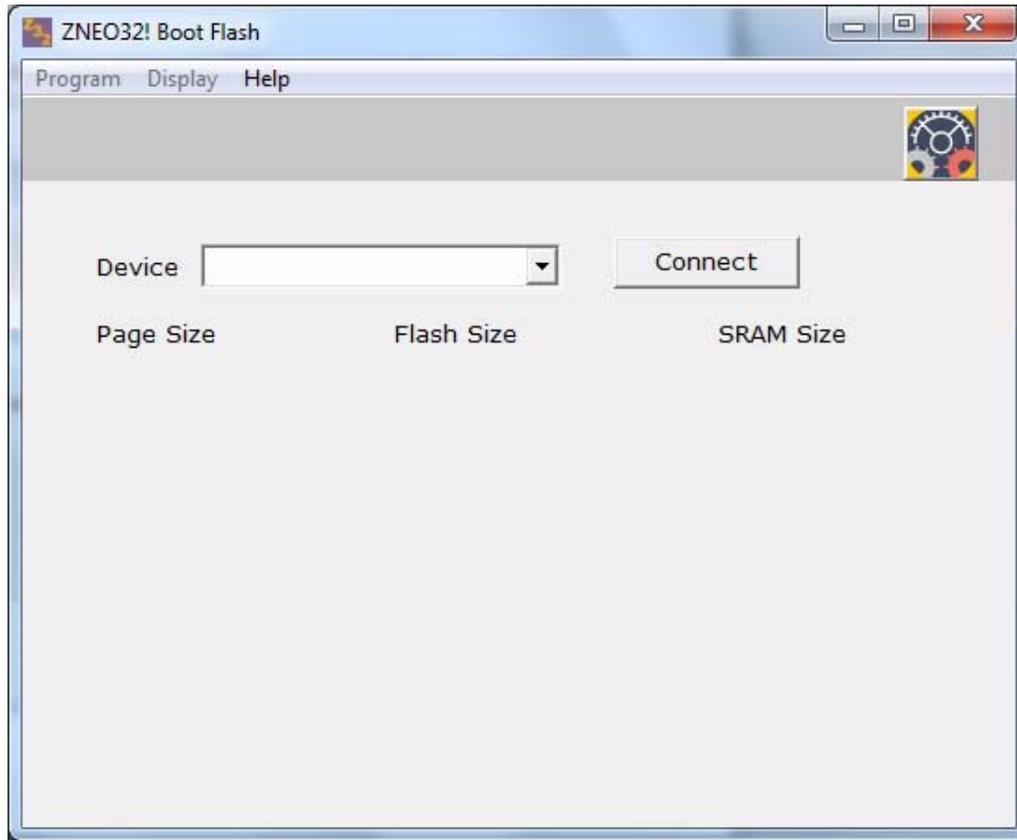


Figure 1. Interactive Mode Screen

## Connecting to the Device

When using Interactive mode, you must be connected to a device. To connect to a device:

1. Connect the device to a COM port of the PC.
2. Ensure that the Boot mode pin is grounded and either power up or reset the device.
3. From the drop down box, select the device that is connected, as shown in Figures 2 and 3.

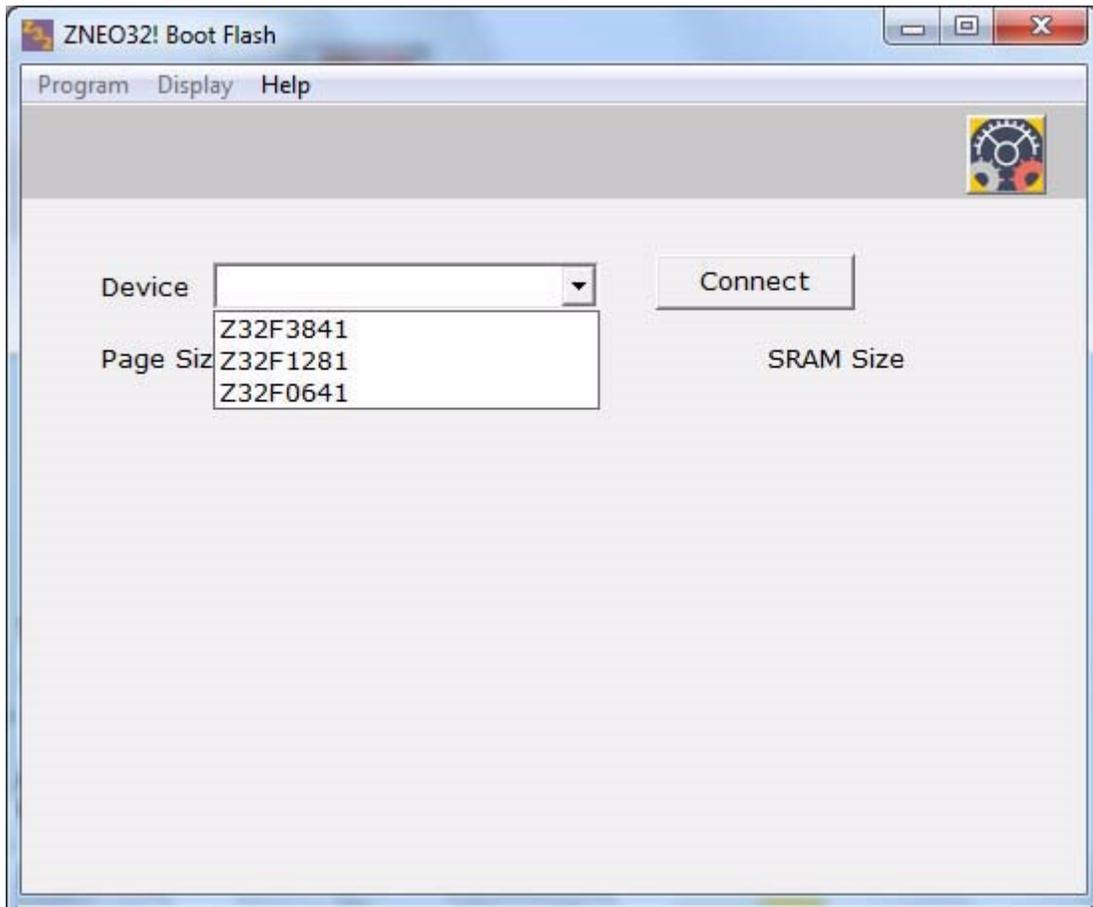


Figure 2. Selecting the Device

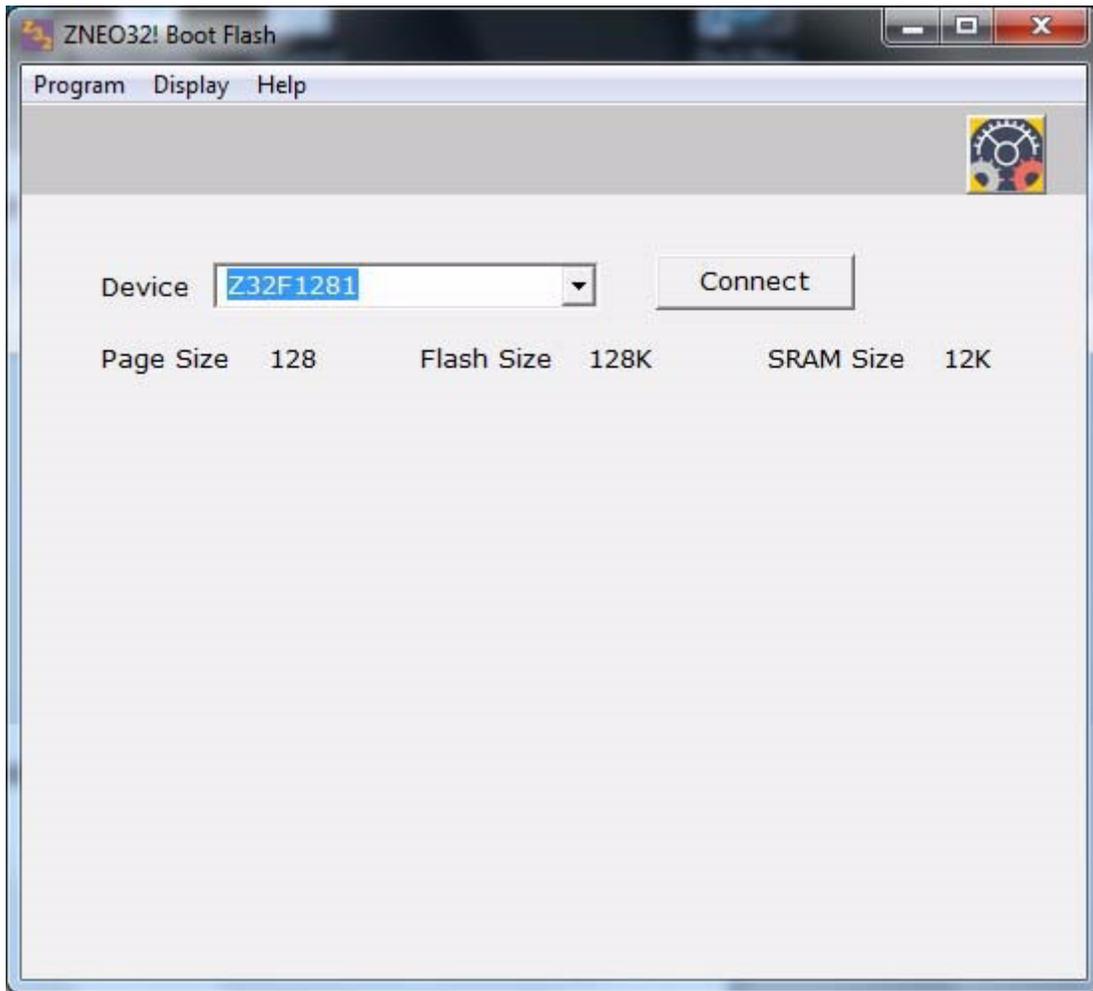


Figure 3. Device Selected

4. The Flash and SRAM information will be displayed. Click the **Connect** button.
5. The system interrogates the available COM ports to find a device to connect to. If multiple devices are available, the system selects the first one found. When the device is found, the system downloads the Stage 2 boot loader after checking to ensure that it is not already running.  
After the device is prepared and connected, the Connect button changes to Disconnect, and the menu and other buttons become available., as shown in Figure 4.

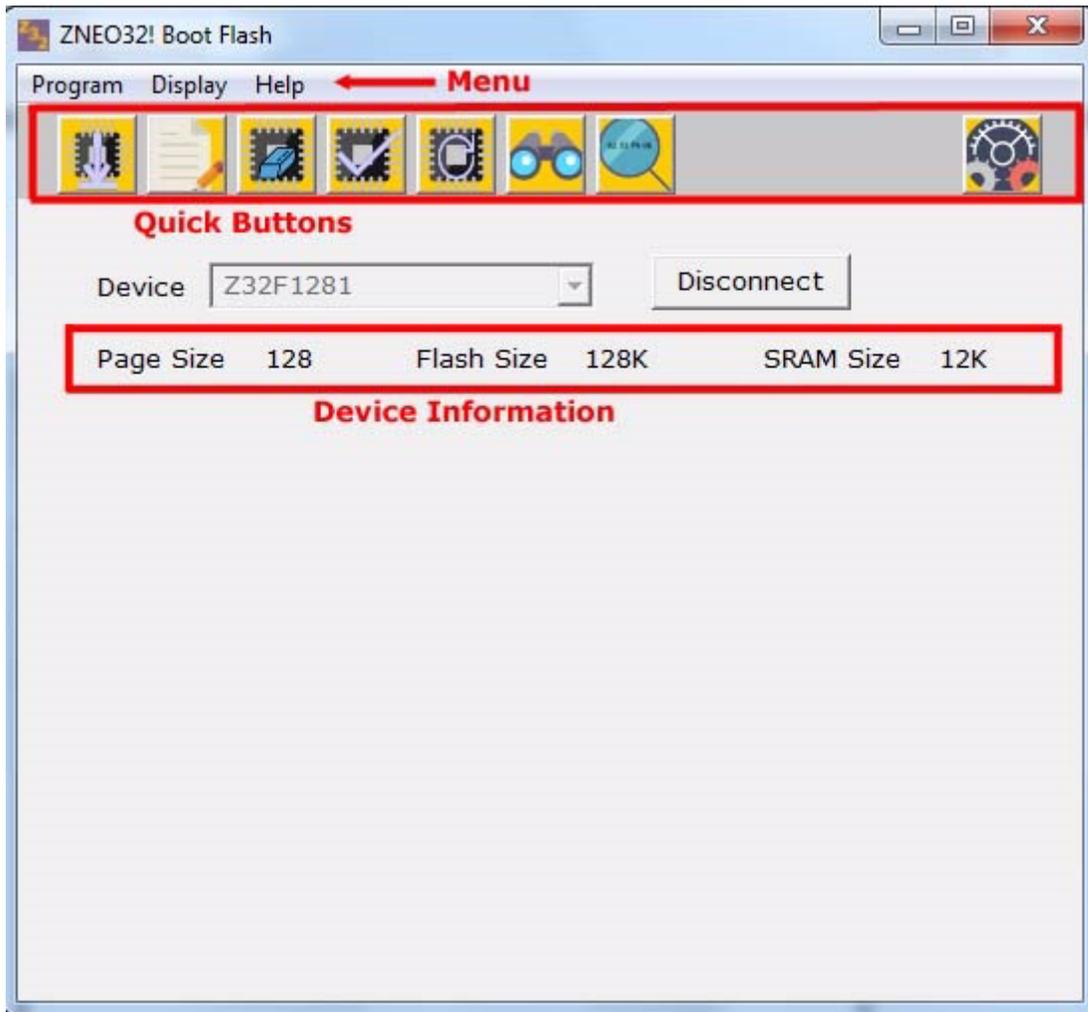


Figure 4. Connected to Device

## Quick Buttons

The Quick buttons bar offers an easy way to switch between different operations within the Interactive mode, and between the Interactive and Production modes. Hover over a button to see its description.

To change to Production mode, select the Production icon located on the far right of the Quick Buttons bar.



## Program Device



The Program Device section allows a user to:

- configure how the device is to be programmed
- program the device

To enable the window, a user must select an image file to write to the device. The supported image files include ELF executable files and Binary (.BIN) files. The ELF executable file is the typical output from most of the ARM toolchains.

---

► **Note:** Typical ELF executable file extensions from the ARM Toolchains include:  
GCC - \*.elf  
Keil MDK - \*.axf  
IAR Workbench - \*.out

---

To program the device:

1. Click the **Open** button to select a file from the screen shown in Figure 5.

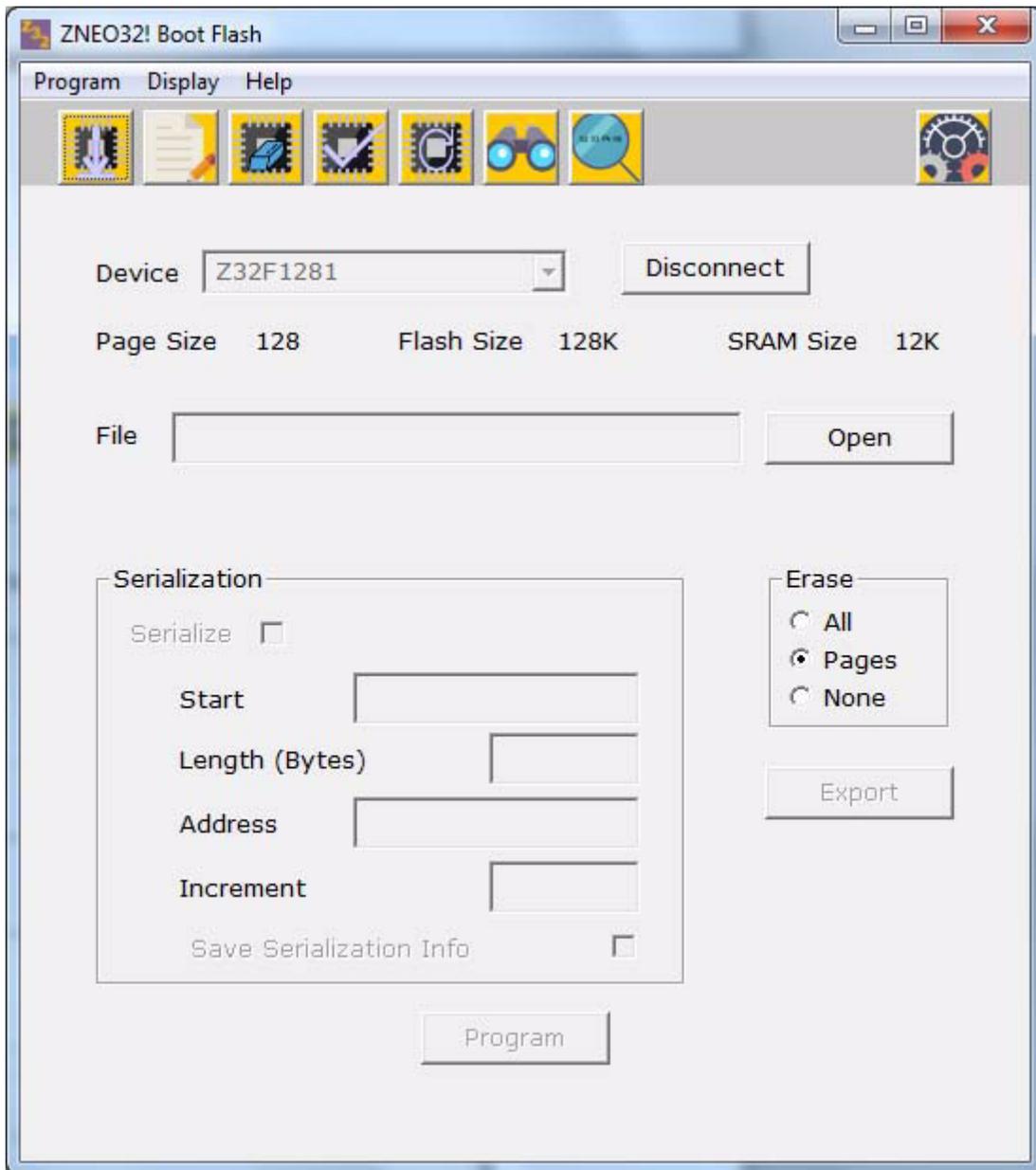


Figure 5. Program the Device

2. From the **Open File** dialog (see Figure 6), select a file and click **Open**.

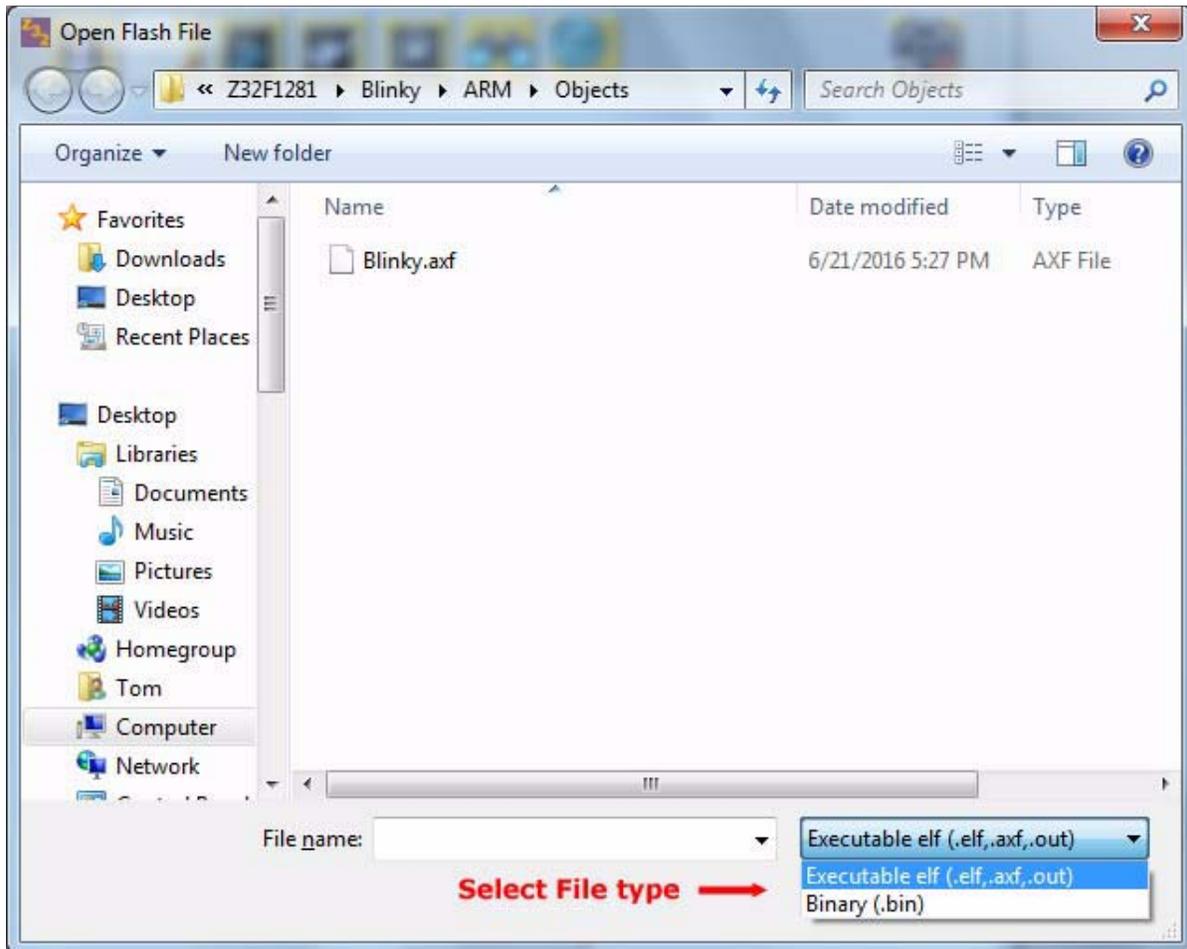


Figure 6. Open File Dialog

3. The system processes the file and creates a Flash image to be downloaded. If the system fails to create a Flash image, the user is notified of the reason for the failure.
4. After the Flash image is created, the system checks for any stored serialization information. If the selected file was used in the past with the same target with serialization, the serialization section is updated with the last used information for that file/device combination. The Erase section and Program button are also enabled, as shown in Figure 7.

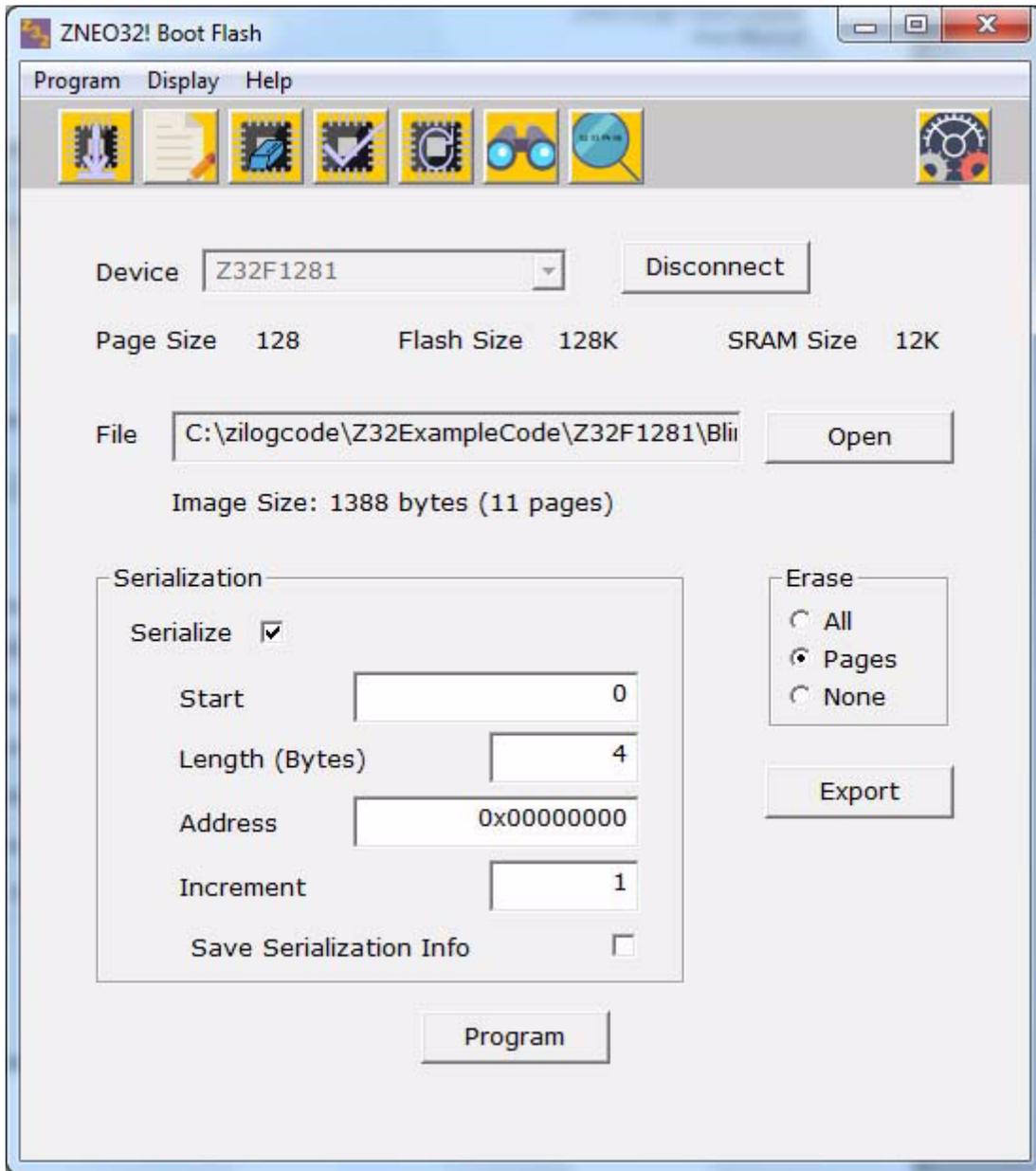


Figure 7. Program Device Information

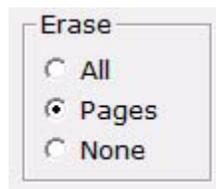
## Erase Area

The Erase section allows a user to select whether to completely erase (**All**) the device or only erase the pages that are being written (**Pages**). The user can also select not to erase anything (**None**).

---

► **Note:** The device can only be written in an erased state; therefore, programming may fail if the erase operation is not performed.

---



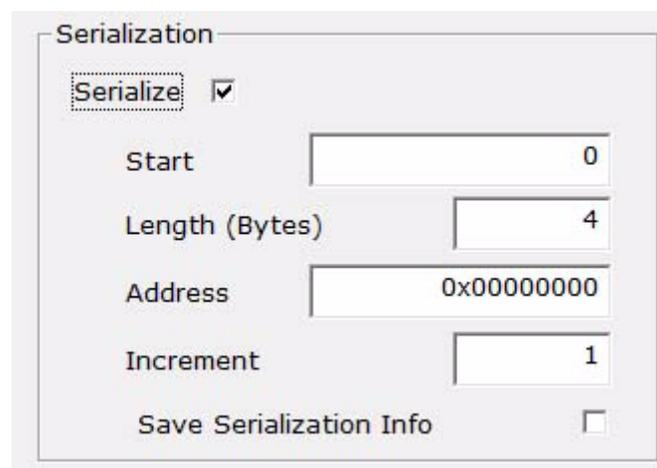
The image shows a dialog box titled "Erase" with three radio button options: "All", "Pages", and "None". The "Pages" option is selected, indicated by a filled circle next to it.

Figure 8. Erase Detail

## Serialization Area

The Serialization section allows a user to provide a way to serialize the device, so that it can be differentiated from other devices. This is necessary for products that require a unique identifier for the device, such as tracking devices, wireless controllers, and MAC addresses.

The Serialization area is shown in Figure 9.



The image shows a dialog box titled "Serialization" with a "Serialize" checkbox checked. Below it are four input fields: "Start" with value 0, "Length (Bytes)" with value 4, "Address" with value 0x00000000, and "Increment" with value 1. At the bottom is a "Save Serialization Info" checkbox which is unchecked.

Figure 9. Serialization Detail

To serialize the part, check the **Serialize** box.

**Start.** This field holds the serialization value that will be written to the part. The serialization value can be between 1-4 bytes long and is automatically incremented by the increment value field.

**Length (Bytes).** This field holds the length of the serialization value to be written.

---

► **Note:** If the address is outside of the Flash image, the length will always be 4 bytes, regardless of the value in the Length field.

---

**Address.** This field contains the address to write the serialization value. This address can be within the Flash image or outside of the Flash image and must be within the Flash area. This allows users to place the serialization number somewhere in Flash, outside of the Flash image for easy access, regardless of changes in code.

The system saves the serialization information using the complete file name (including the path) and the target as the unique identifier, allowing a user to have multiple serializations for each of their products without having to remember the last values for each of them.

---

► **Note:** The serialization information is saved only if the **Save Serialization Info** checkbox is selected. If the address is outside of the Flash image, the address value must be on the 32-bit word boundary.

---

**Increment.** Upon successful programming of the serialization information, the serialization value (Start) is incremented by this value.

**Save Serialization Info.** To save the serialization information, check this box. If this box is not checked, the Start value is not incremented and the information is not saved.

**Export Button.** The Export button allows the current information (file, device, serialization, and erase) for a project file to be used by Production mode. The serialization information is saved regardless of the Save Serialization Info checkbox selection.

Click the **Export** button to open the **Save Project File** dialog shown in Figure 10.

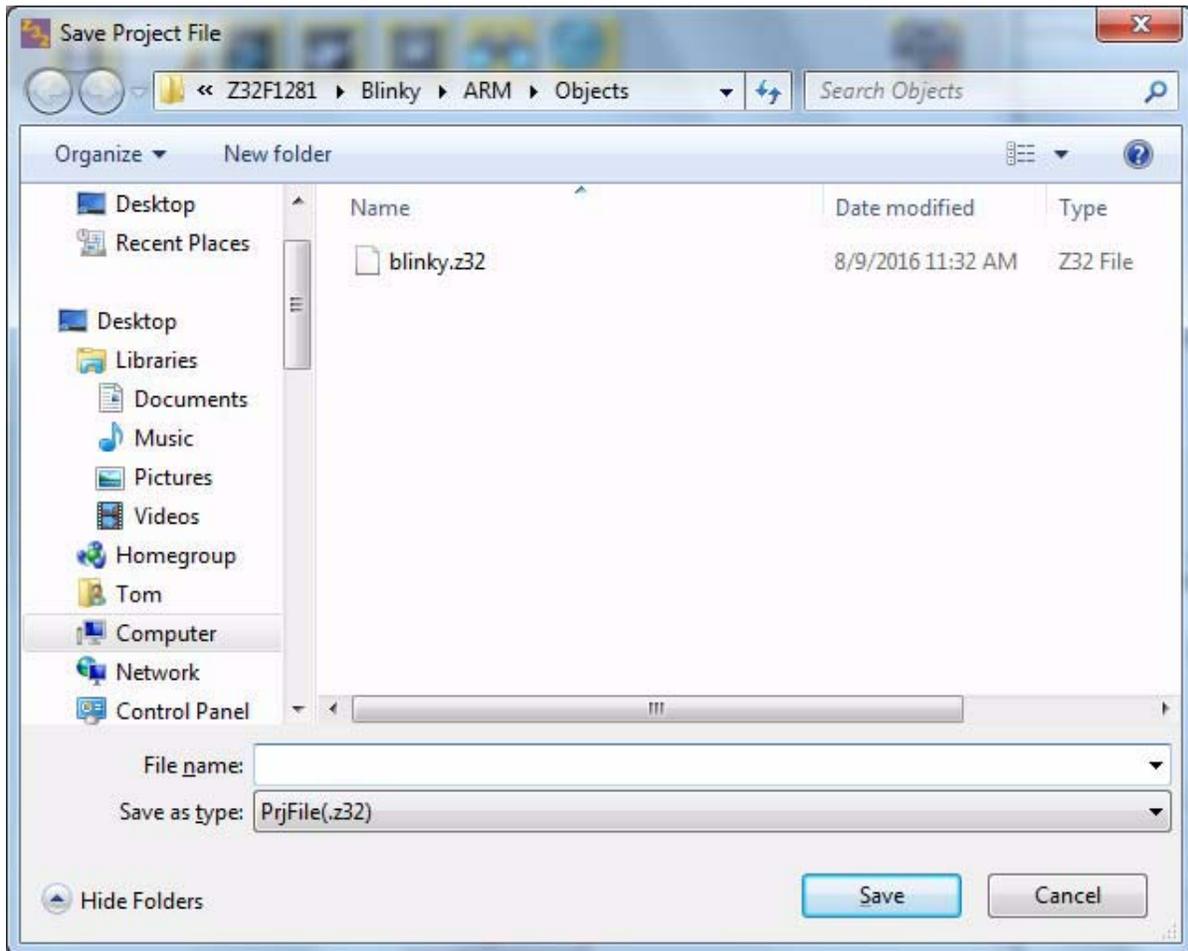


Figure 10. Save Project Dialog

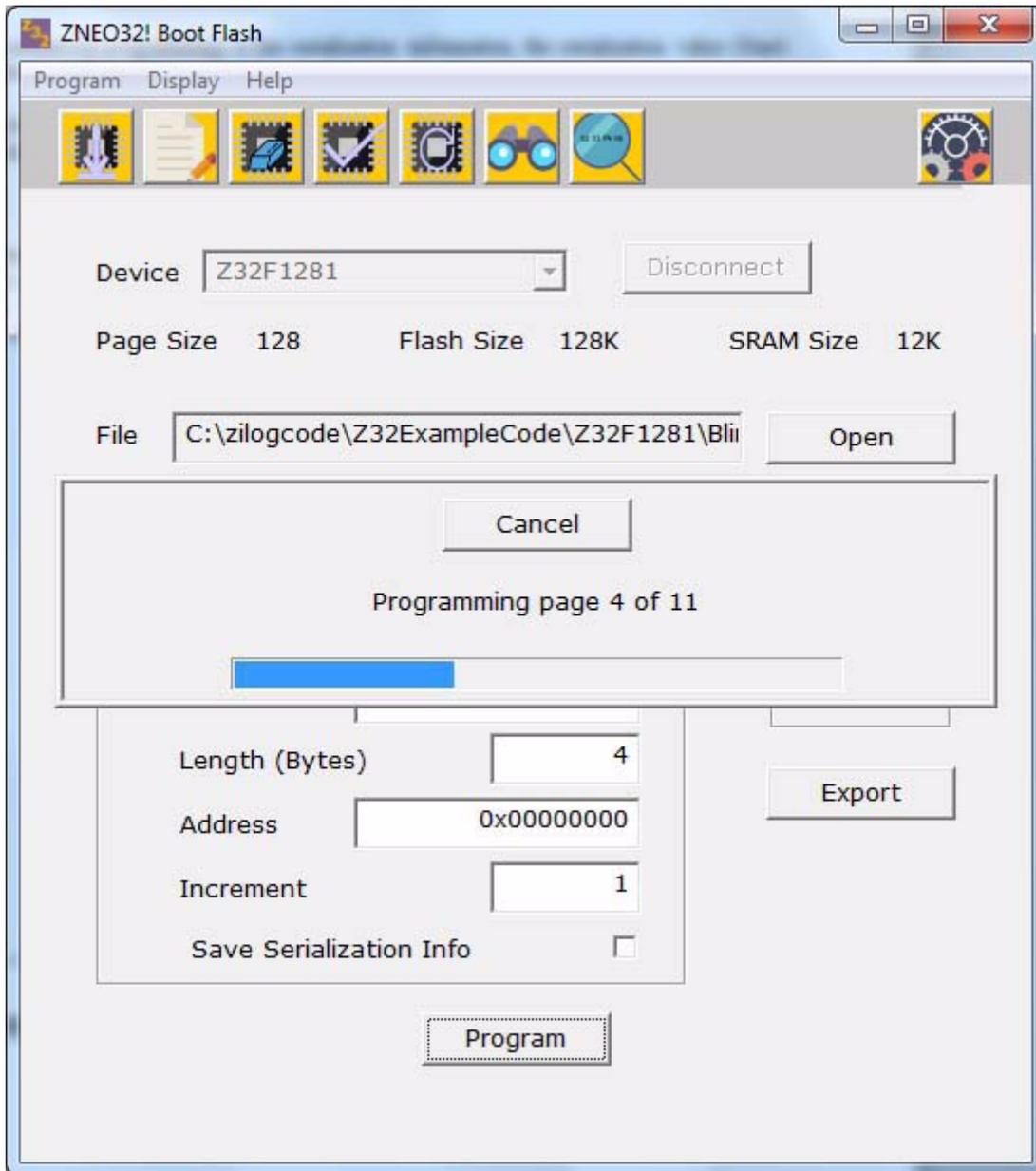
Select or type in a file name to save the project file, then press the **Save** button.

---

► **Note:** When using Production mode, the image file must be in the same directory as the project file.

---

**Program Button:** When Program is selected, the system attempts to write the selected Flash image to the part after erasing (if selected). A progress bar is presented during execution, as shown in Figure 11.



**Figure 11. Programming Device Execution**

The progress bar includes a Cancel button, if you wish to cancel during the programming process. Canceling only stops the programming, and does not restore the device's Flash back prior to programming.

After the programming is complete, a message box is displayed with the results, as shown in Figure 12. Click **OK** to continue.



Figure 12. Programming Complete Message

## Program Data



The Program Data section allows a user to enter up to 256 bytes and write the data to a specified address in the Flash memory of the device. Figure 13 shows the Program Data section.

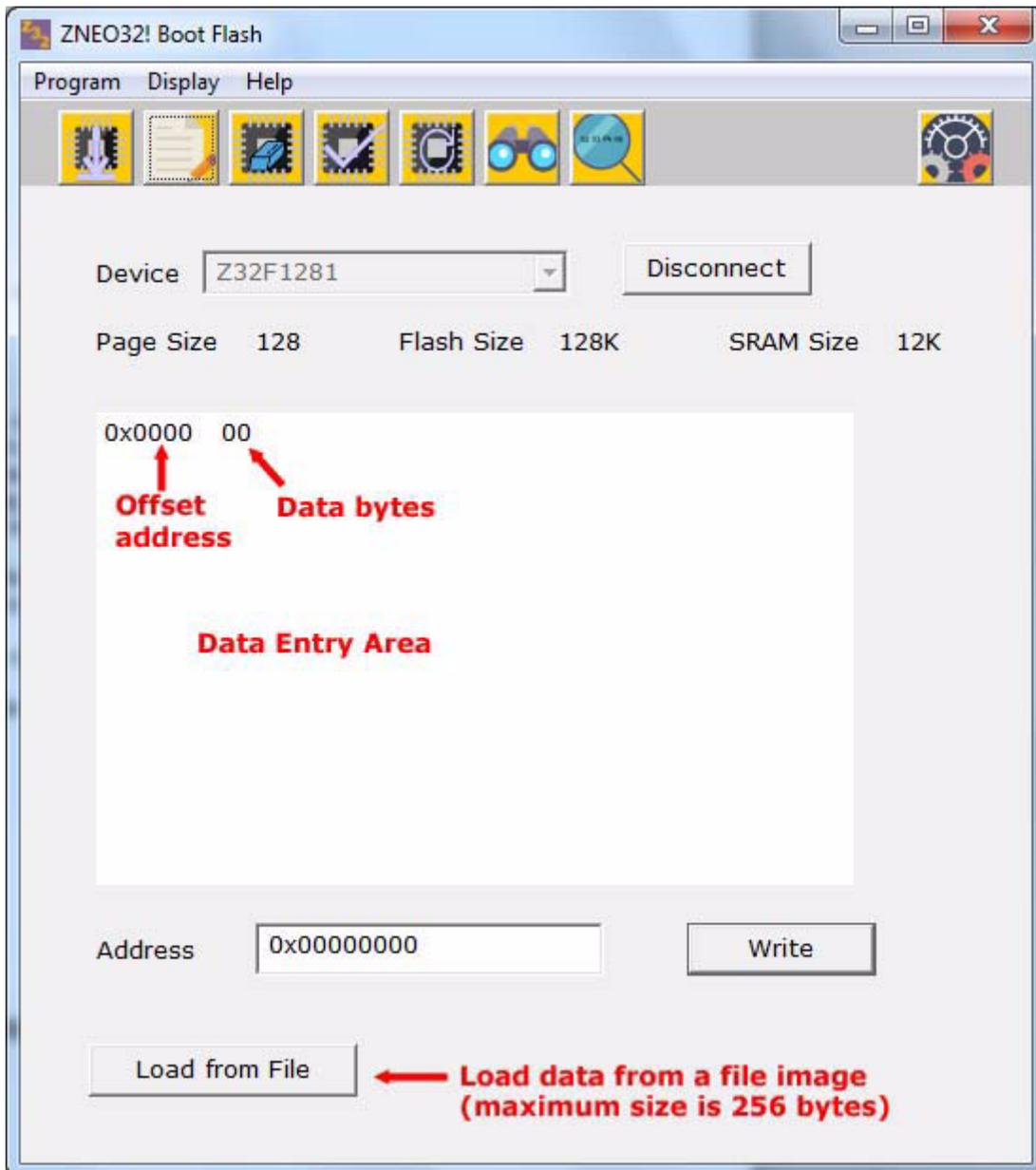


Figure 13. Program Data

**Data Entry area.** The Data Entry area provides the ability to enter the hexadecimal values to be written to the device.

#### To add data for editing

1. Right click any blank area (not between data bytes).
2. An Add menu is displayed; select an option (see Figure 14). Any options that would exceed the buffer length will be unavailable.

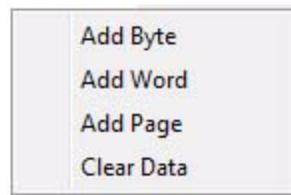


Figure 14. Add Context Menu

#### To edit data

1. Left click any data byte to highlight it.
2. Enter the desired hexadecimal value.
3. To advance to the next byte, press the **Tab** key; to go back 1 byte, press the **Shift+ Tab** keys. Press **Enter** when the entry is completed.

#### To insert or delete a byte

1. Without selecting the data byte, right click the data byte and select the desired action from the menu, shown in Figure 15.

---

► **Note:** If a data byte is selected, the standard Windows **Properties** menu will appear.

---

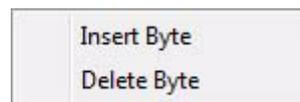


Figure 15. Insert Context Menu

**Load from File.** This feature provides the ability to load the Data Entry area with data from an image file. The image file cannot exceed 256 bytes.

#### To load data from an image file

1. Click the **Load from File** button.

2. The **Open File** dialog, shown in Figure 16, appears. Select the file type (typically, a .bin file) and select the file to be loaded.

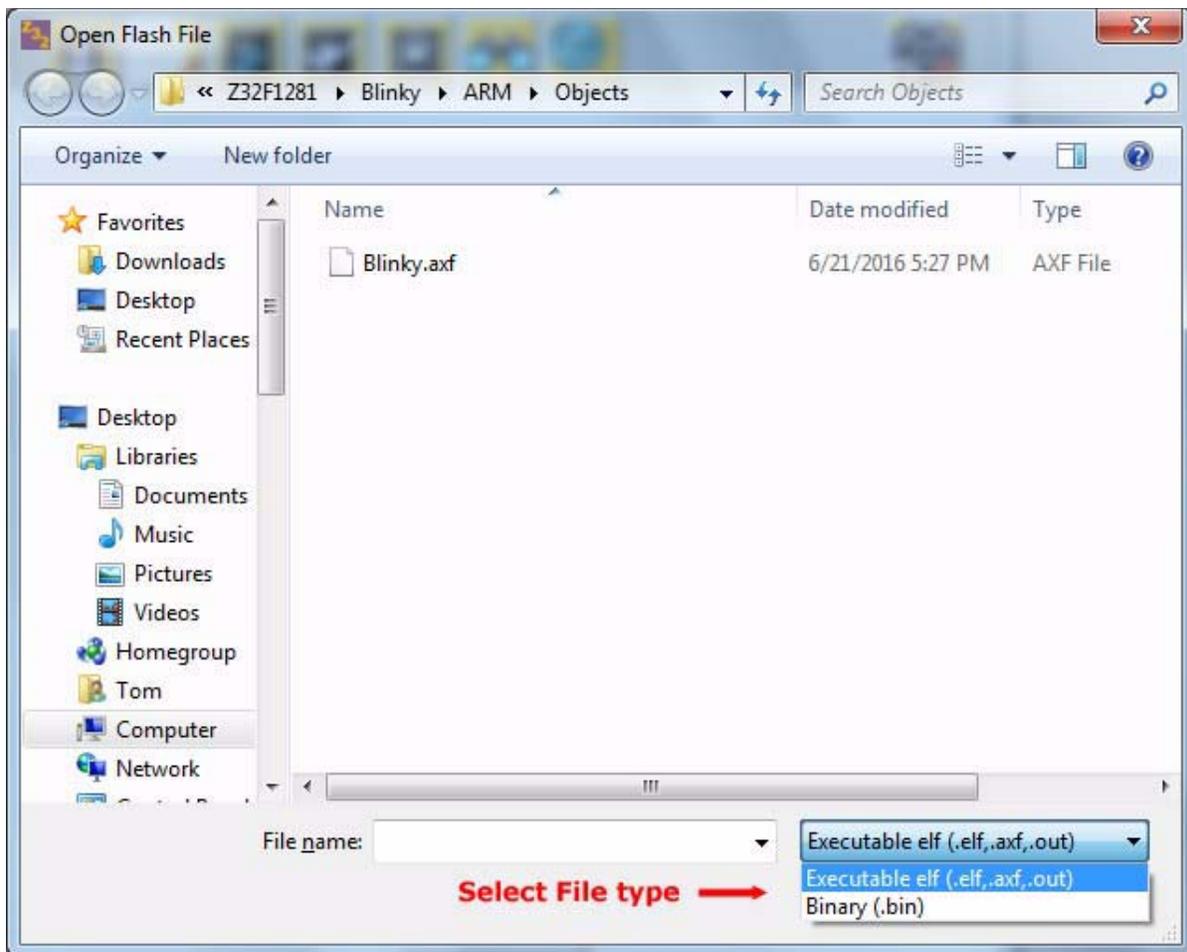


Figure 16. Open File Dialog

3. If the file is under 256 bytes and can be processed, the data will appear in the Data Entry area; otherwise, an error message is displayed.

When the data entry is complete, enter the address in the Flash memory to write to, and click the **Write** button to write the data to the device.

---

► **Note:** For the Write operation to be successful, the location should be erased prior to writing the data to the device.

---

## Erase Device



The Erase Device section allows users to erase all or selected pages of the chip. Figure 17 shows the Erase Device screen.

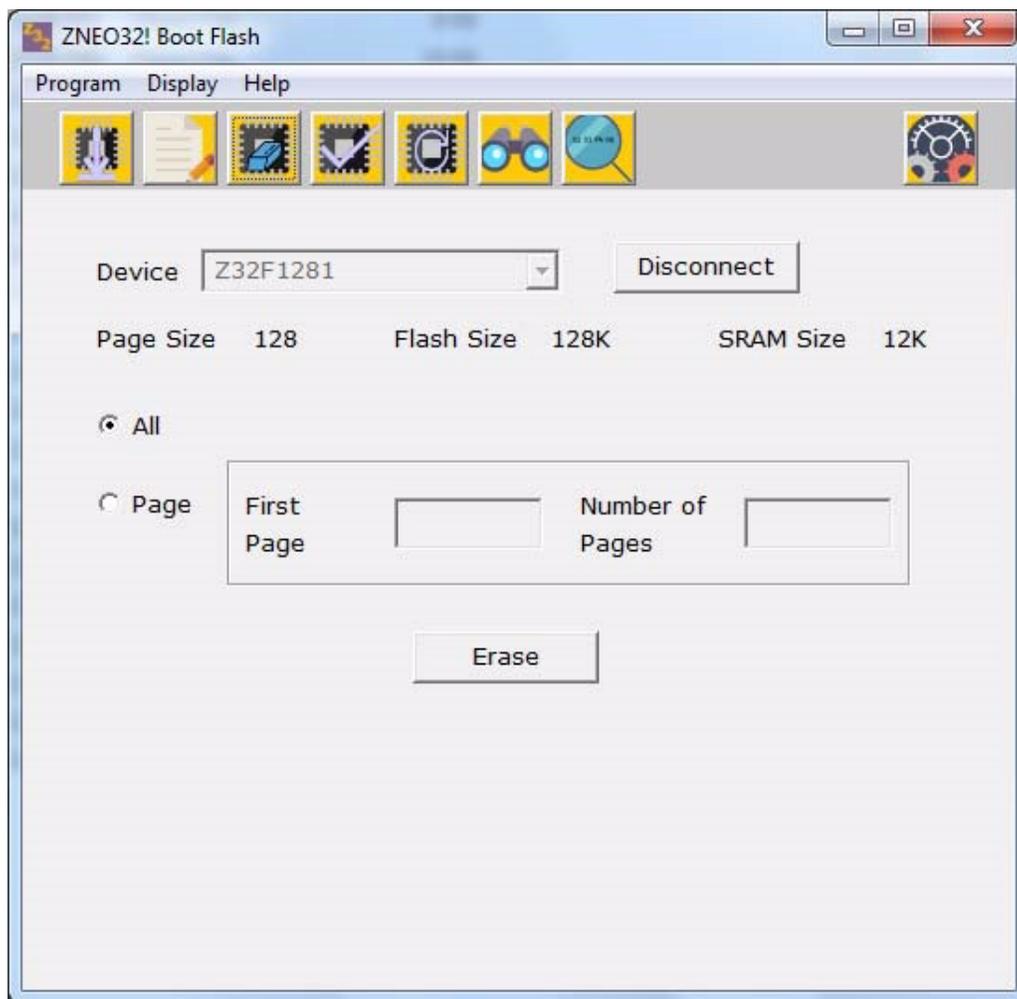


Figure 17. Erase Page

- To erase the entire chip, select **All** and click the **Erase** button.
- To erase selected pages, select **Page** and enter the first page to erase (page numbering starts at 0) and the number of pages to erase.

---

► **Note:** If the number of pages exceeds the Flash size, the system will automatically change the number of pages to the end of Flash memory.

---

To erase, click the **Erase** button. The progress window shows the progress of the erase operation, as shown in Figure 18. A **Cancel** button is available to cancel the process if necessary.

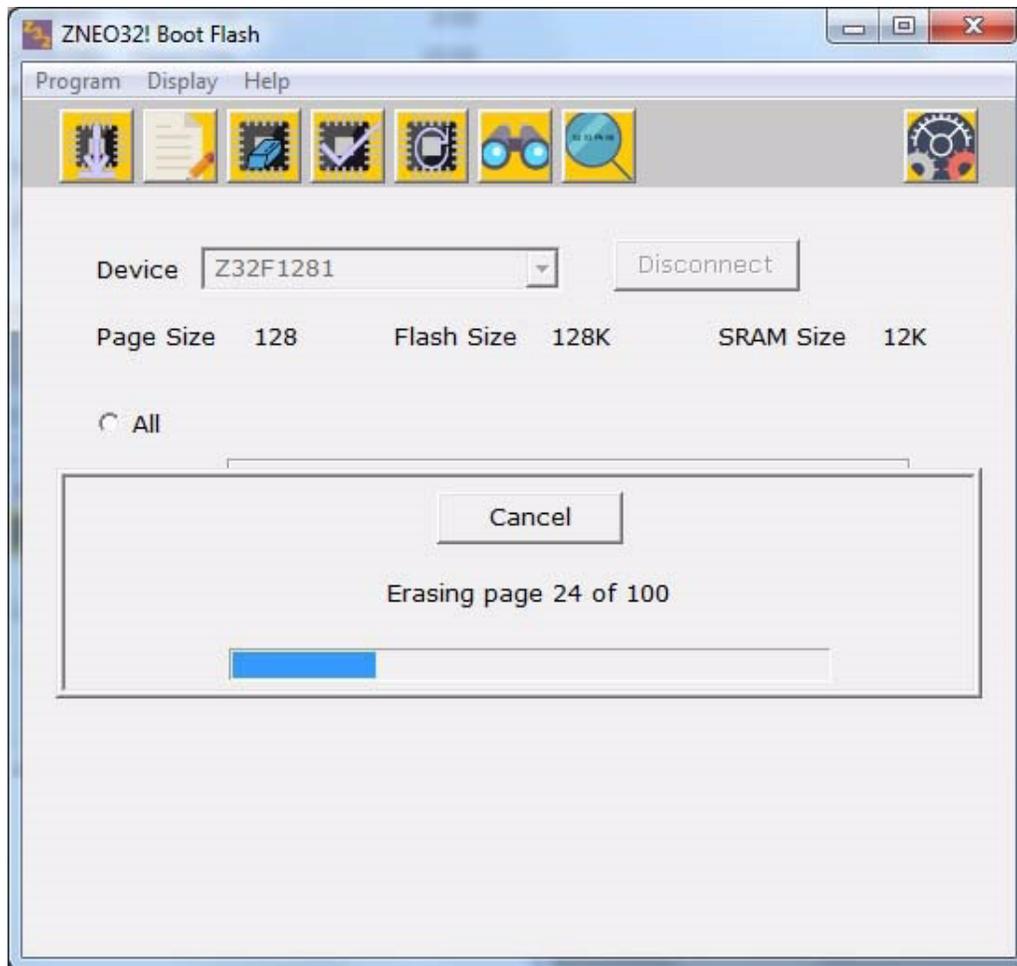


Figure 18. Erasing the Device

## Verify



The Verify section provides the ability to verify that the programmed device has not changed from the selected Flash image.

1. Click **Open** to select a file to load the Flash image to compare to. See Figure 19.

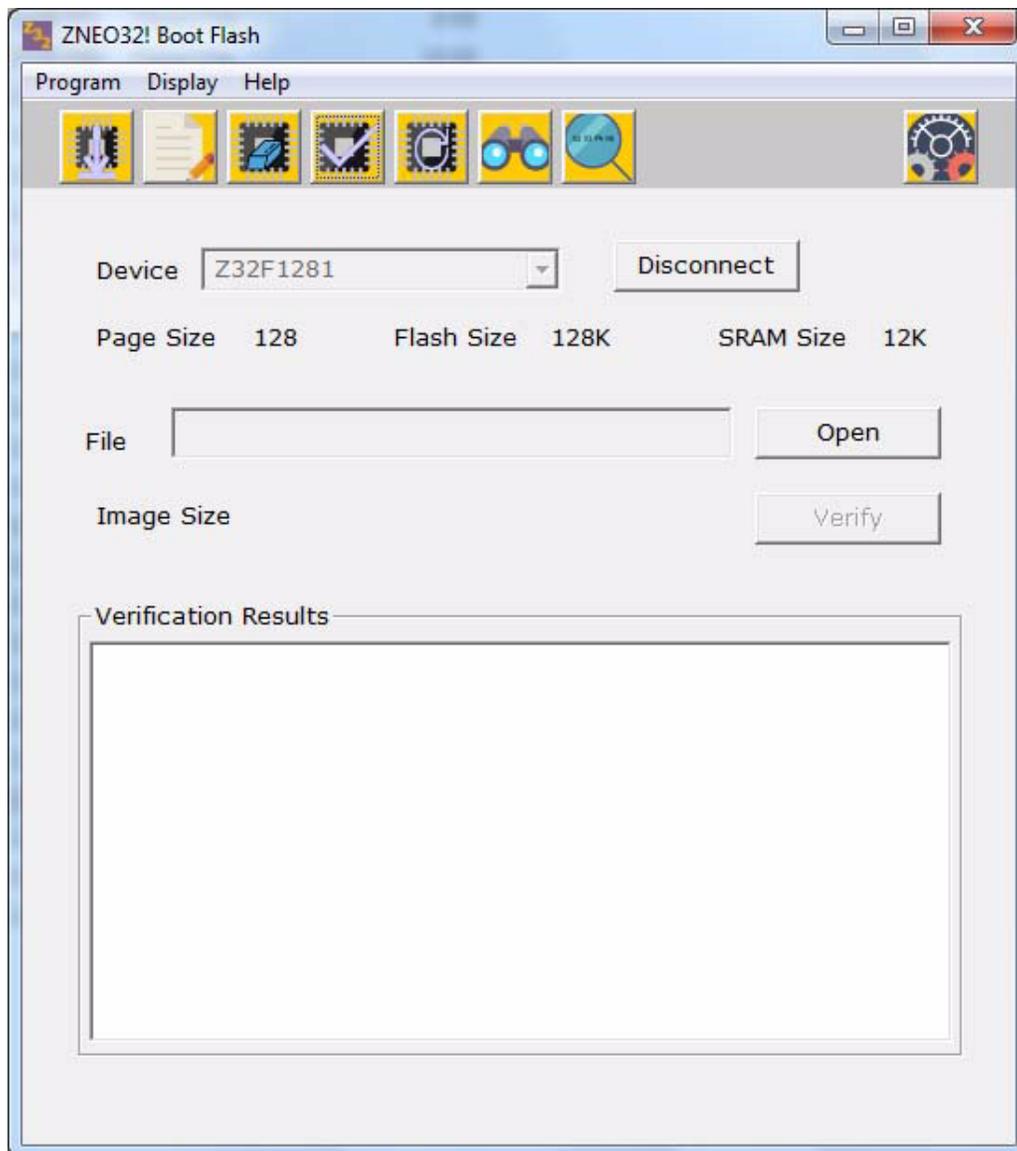


Figure 19. Verify Screen

2. In the **Open File** dialog that appears, shown in Figure 20, select the image file and click **Open**.

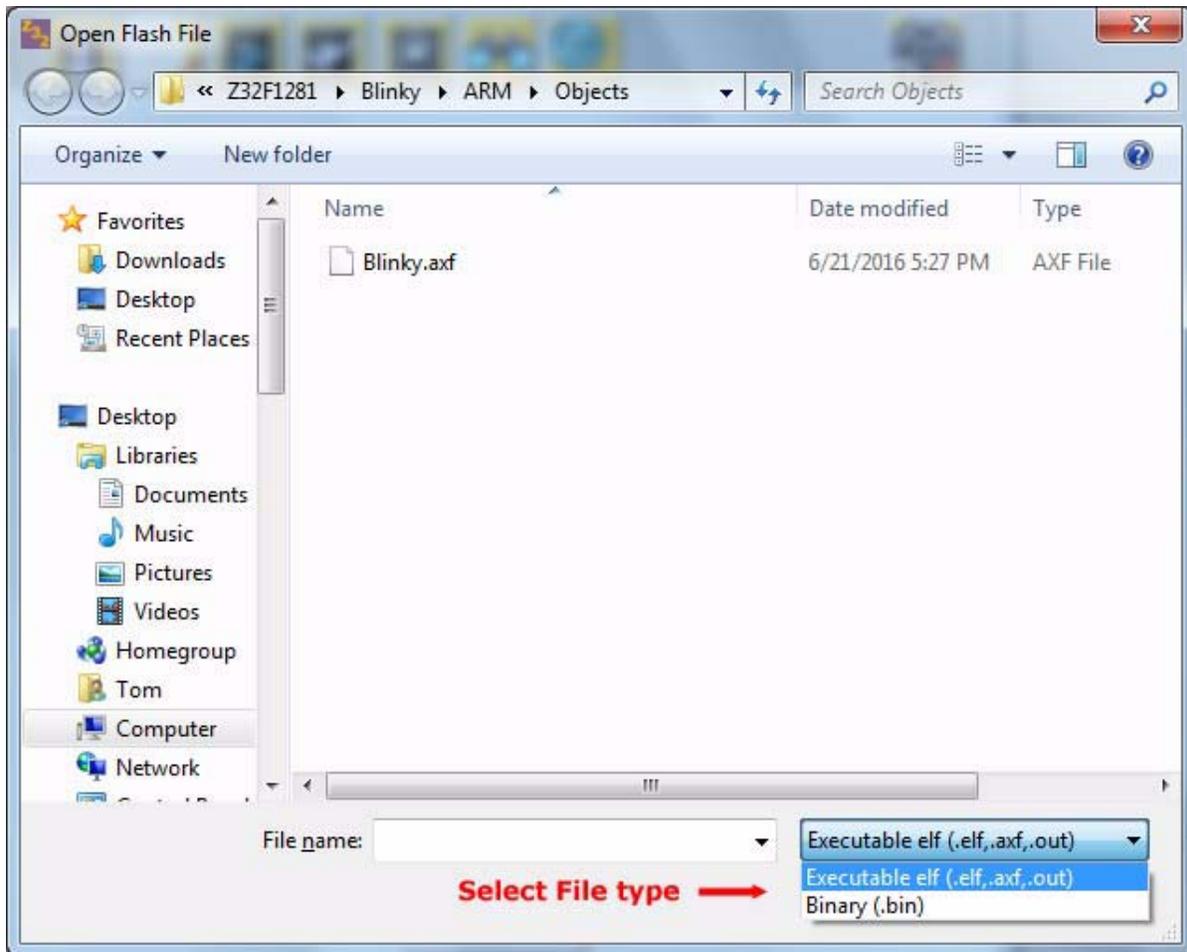


Figure 20. Open File Dialog

3. Click **Verify** on the screen shown in Figure 21.

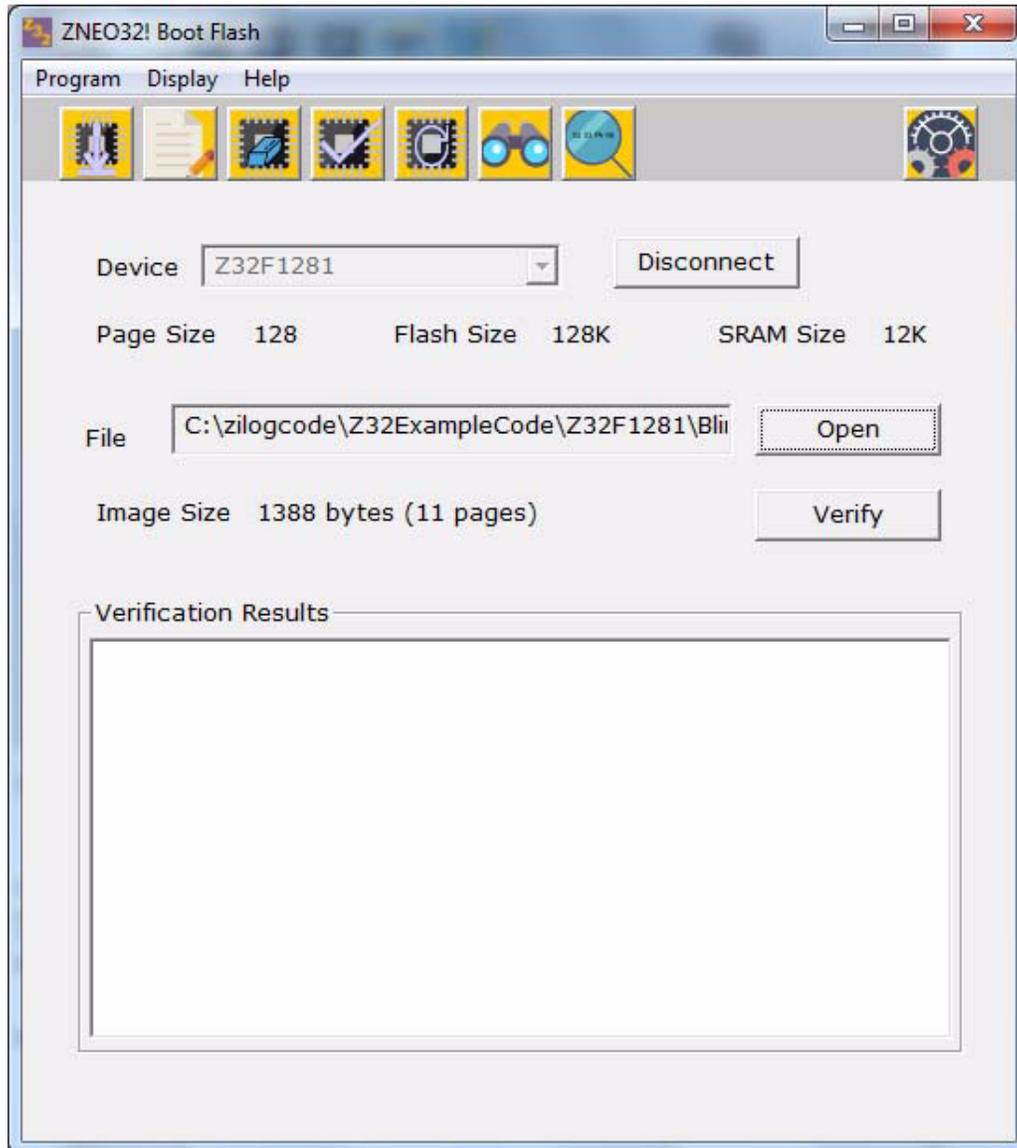
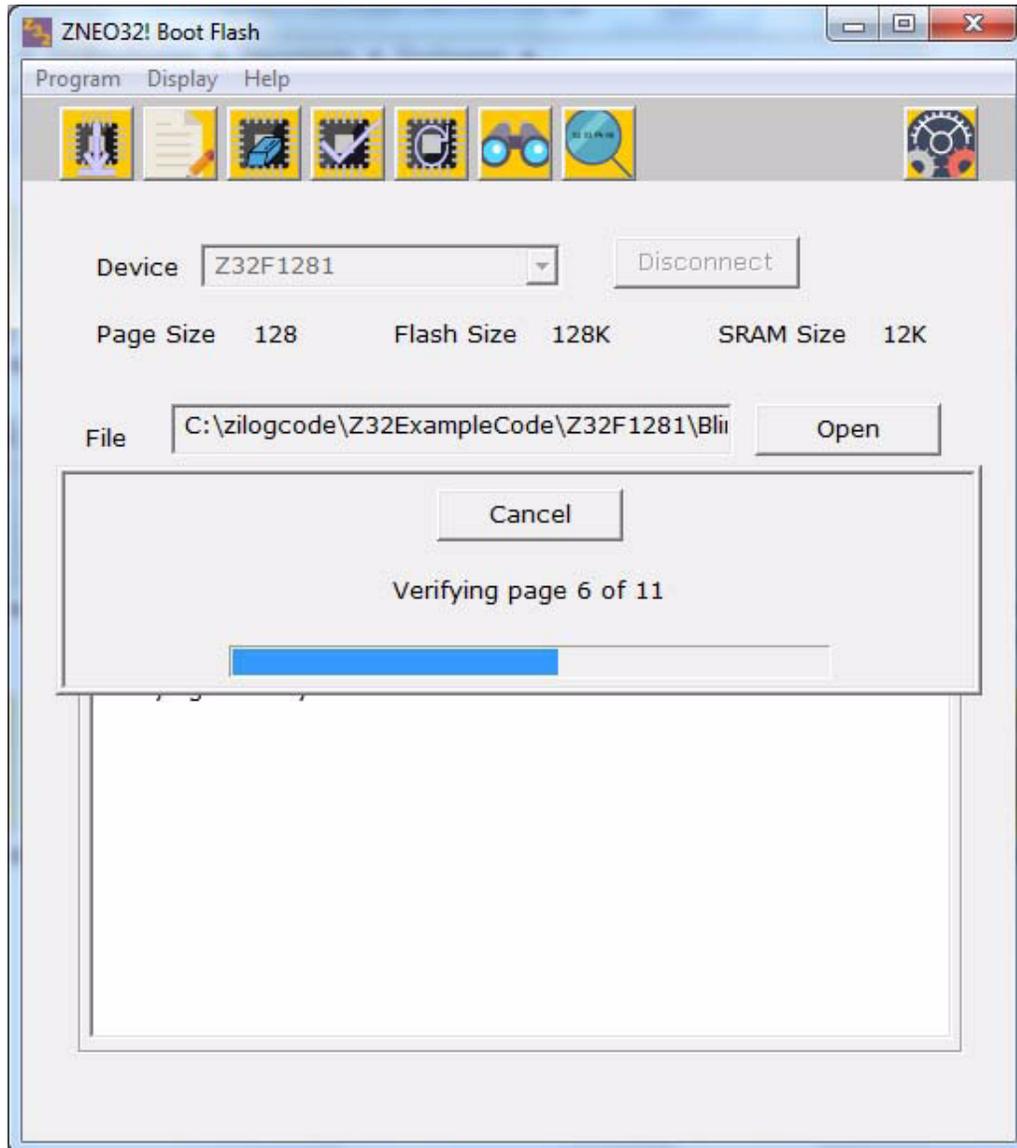


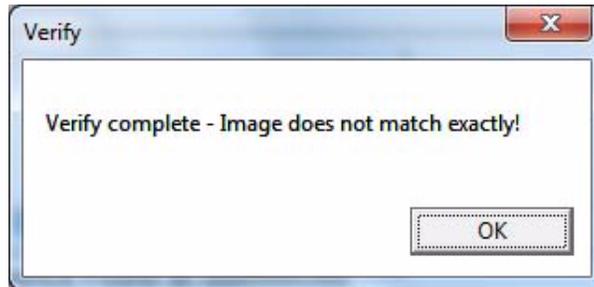
Figure 21. Ready to Verify

4. The system starts the verification process and shows the status bar of the progress made, as shown in Figure 22.



**Figure 22. Verifying**

5. When the verification is complete, the system displays a message with the result of the verification, as shown in Figure 23.



**Figure 23. Result Message**

6. The result details appear in the status box, as shown in Figure 24.

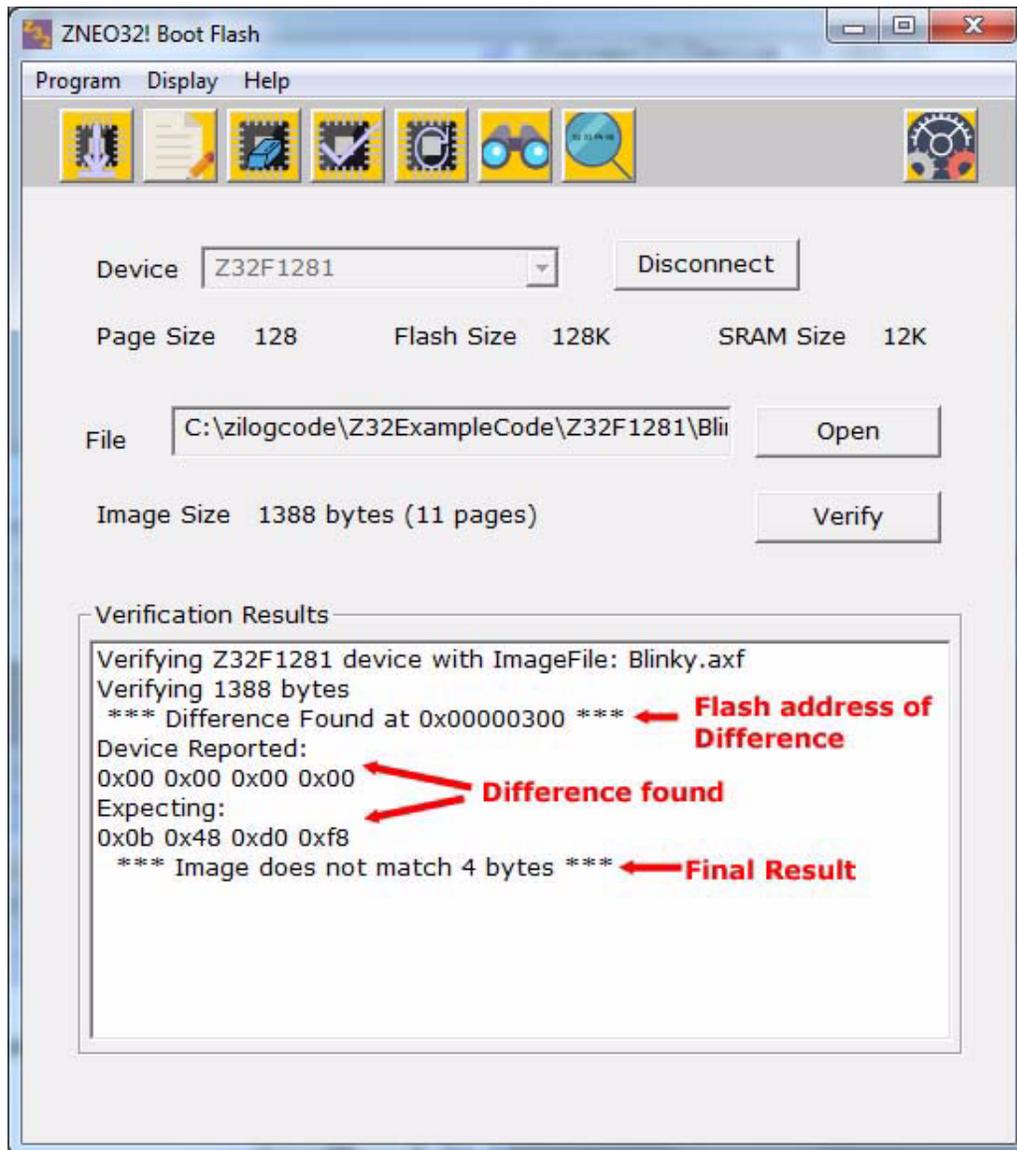


Figure 24. Result Details

When the system encounters differences between the device and the image file, the location of the Flash address where the differences exist is displayed. Additionally, the difference between actual device bytes compared to the bytes expected by the Flash image is also shown.

The verification fails if more than 32 bytes are different; otherwise, it *conditionally* fails, to allow the user to determine if the changes are acceptable. This feature provides the abil-

ity to check the entire image even though a serialization value is different, or if the application uses in-programming Flash writing of specific areas.

## Reset Device



The Reset Device functionality provides the ability to reset the device.

1. Remove the Boot Pin from Ground and reset to have the device start operating the Flash code that was downloaded.
2. When selecting the reset device, the system displays the message shown in Figure 25.

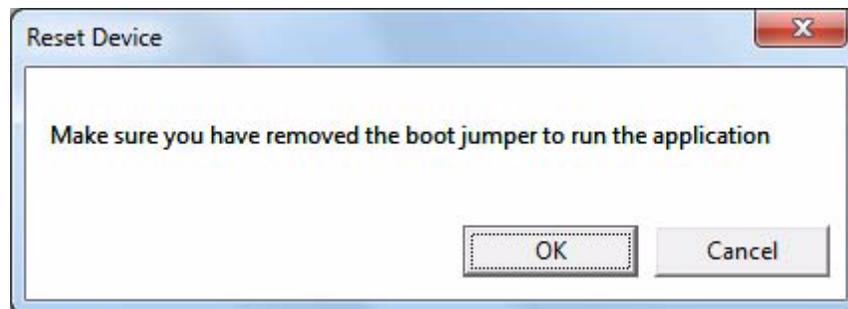


Figure 25. Reset Device Message

3. Click **OK** to reset the device.
4. After reset, the system disconnects from the device.

## Display Page



The Display Page data provides the ability to review the binary information on the selected page.

1. On the screen shown in Figure 26, enter the page number (page numbering starts with 0) and click **Load**.

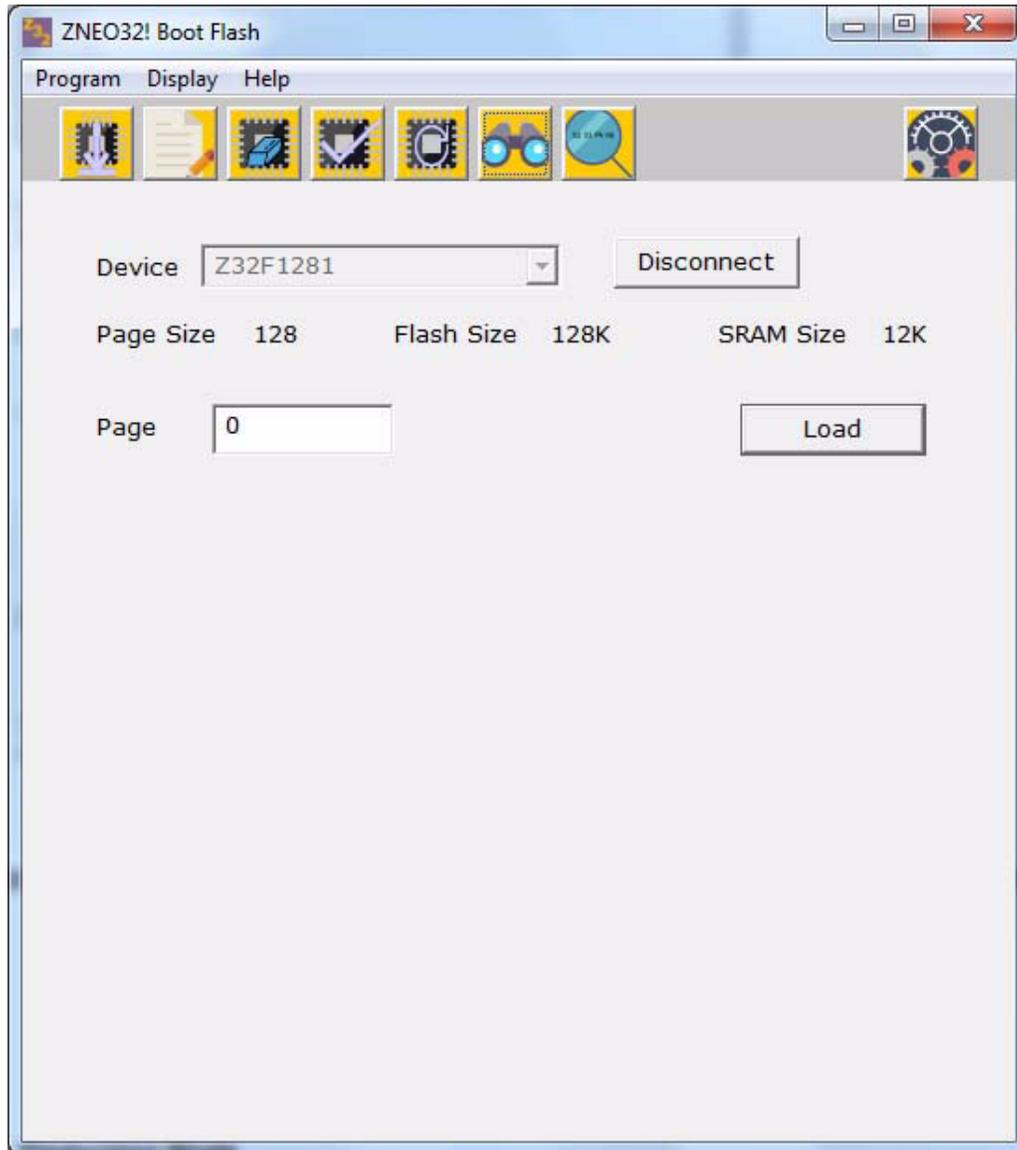


Figure 26. Display Page

2. The system retrieves the data from the device and displays the bytes with the address location, as shown in Figure 27.

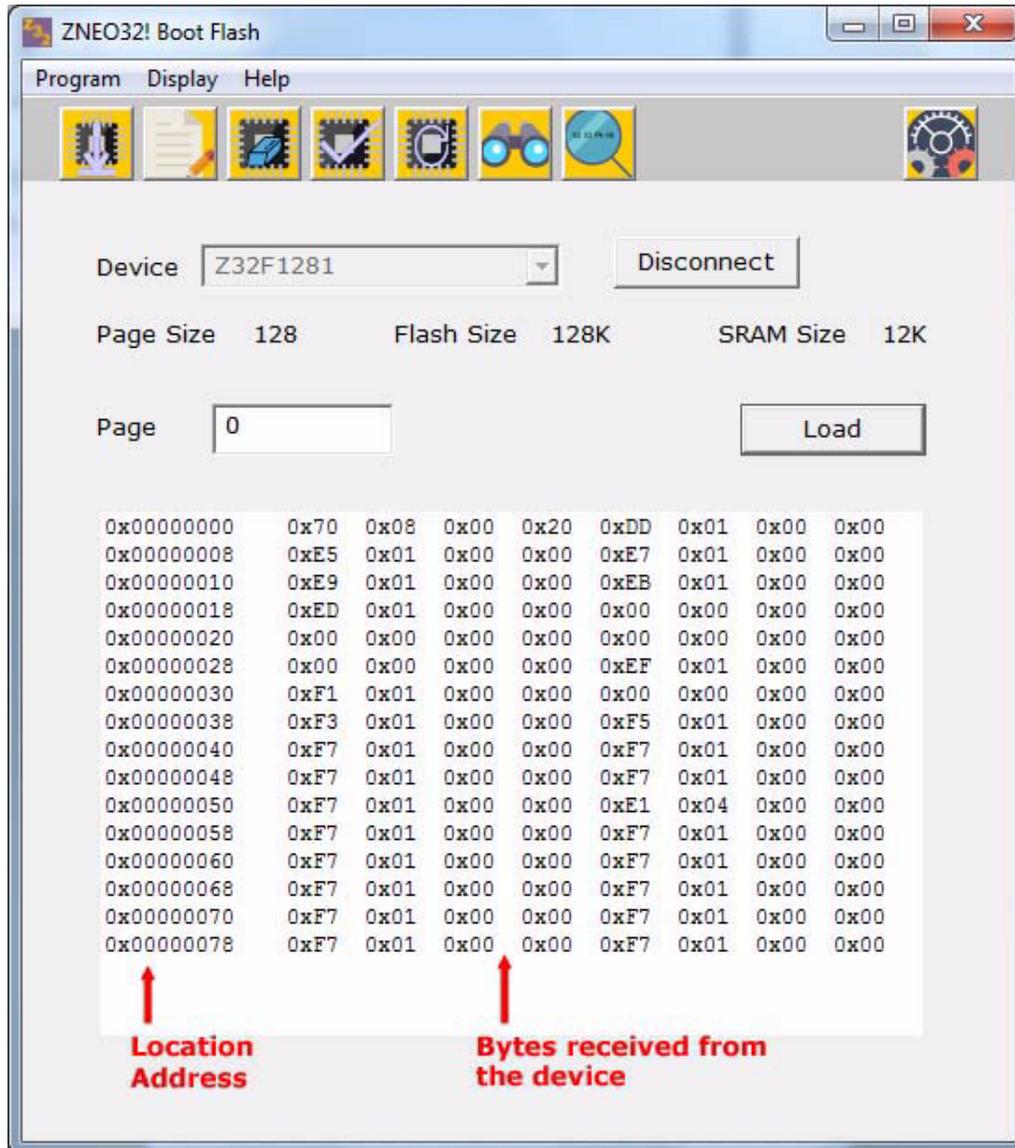


Figure 27. Page Displayed

## Display Address



The Display Address functionality enables the inspection of any location within the device. The maximum length is 4096 bytes, which provides the ability to inspect all readable registers, SRAM, and Flash.

1. On the screen shown in Figure 28, enter the address and the number of bytes to retrieve, then click **Load**.

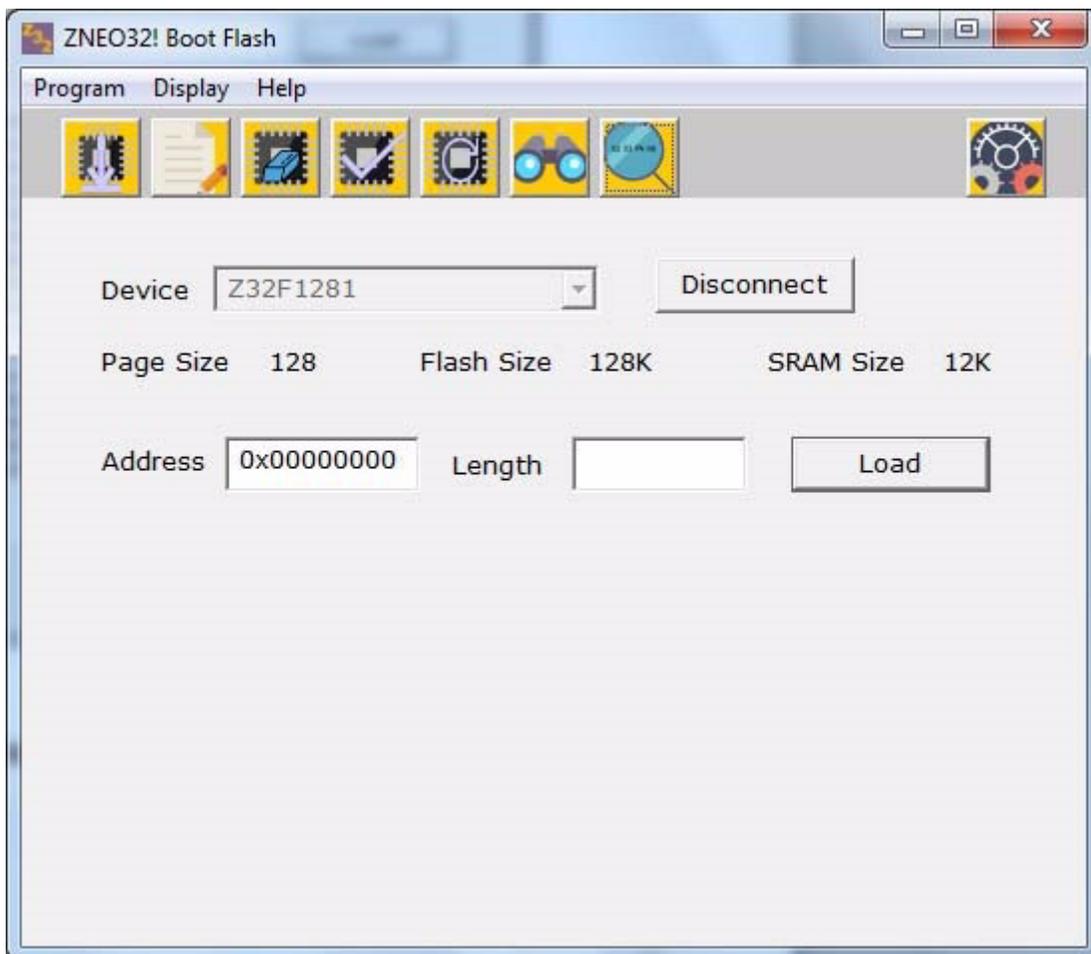


Figure 28. Display Address

2. The results are displayed on the screen shown in Figure 29.

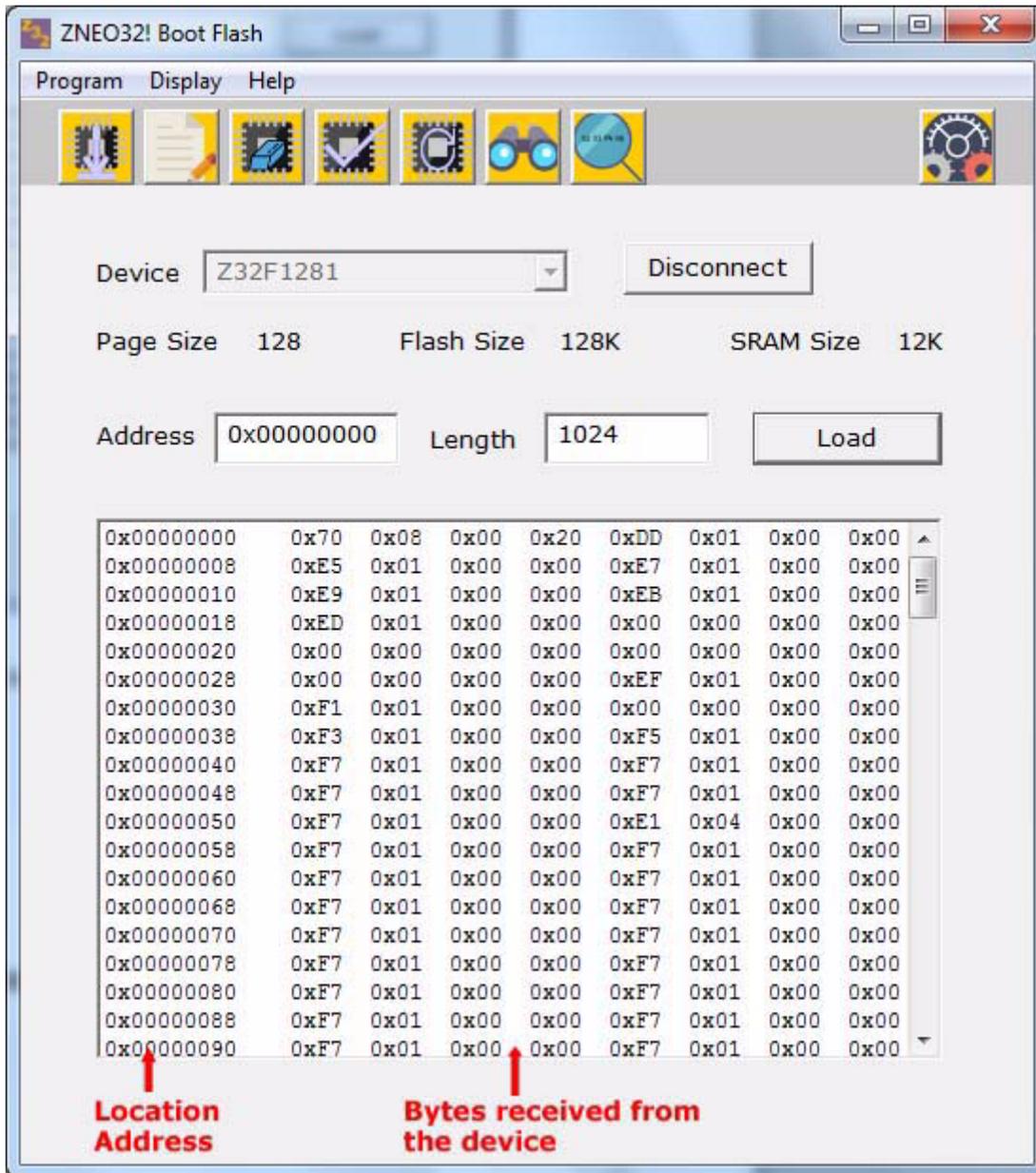


Figure 29. Display Address Results

## Production Mode

Production mode uses a project file (.z32) to program one or more parts at the same time. This mode also keeps track of the serialization value and increments the serialization value every time a device is successfully programmed (if serialization is used).

Upon clicking the program button, the Production mode checks for all connected devices, and programs each of the devices found until all are programmed or programming is cancelled.

The project file is created when the Program Device Export button is clicked in Interactive mode. The image file to be programmed must be in the same folder as the project file.

To change the mode to Interactive mode, select the Interactive icon



located on the far right of the Quick Buttons bar. The Production mode screen is shown in Figure 30.

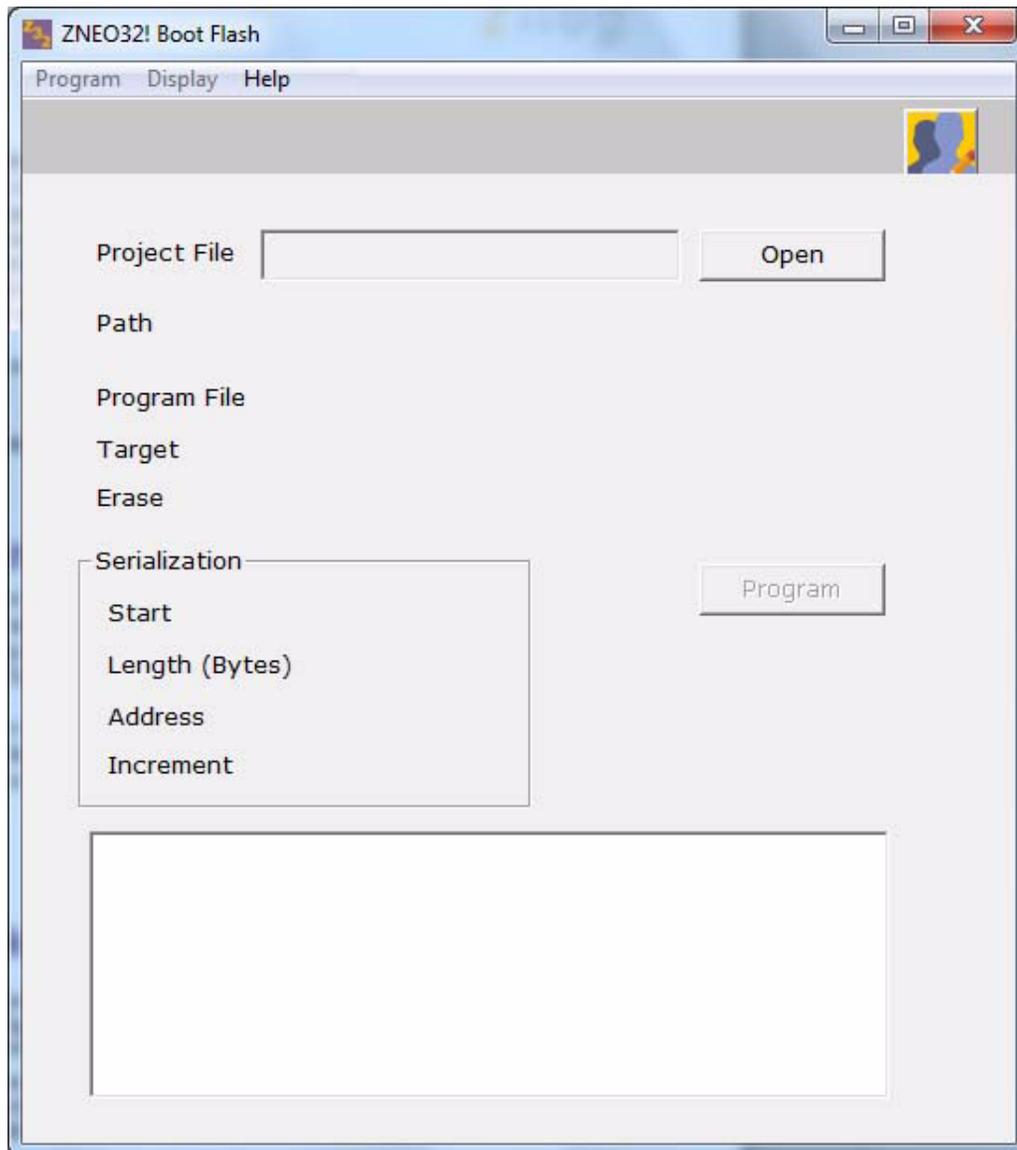


Figure 30. Production Mode Screen

1. To program devices in Production mode, click the **Open** button to open a project file.

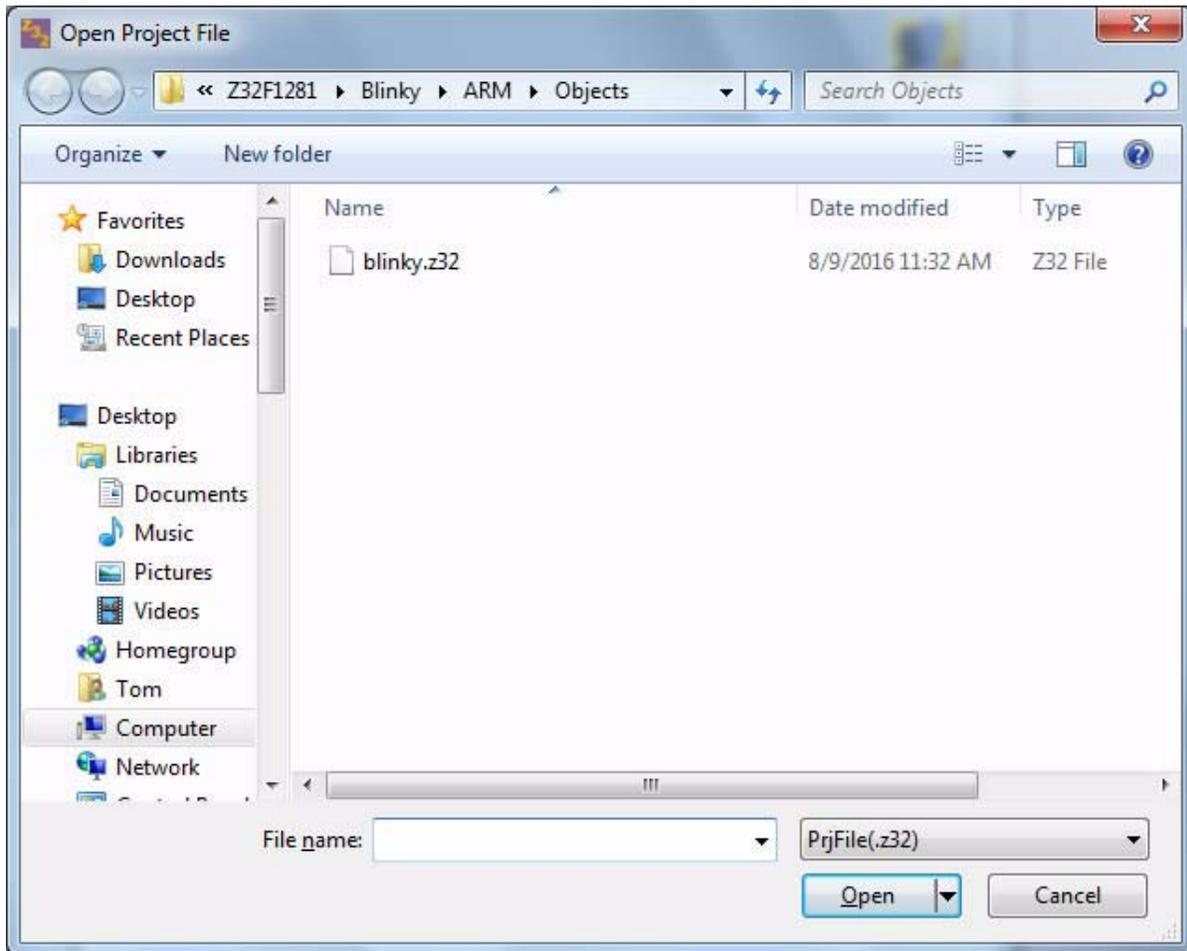


Figure 31. Open Project File

2. Select the project file to load. Project files have a .z32 extension. The system opens the file and populates the Production Mode screen with all the information in the project file, ready to program. If an error occurs, the system displays an error message.

---

► **Note:** The image file specified in the project file must be in the same directory as the project file.

---

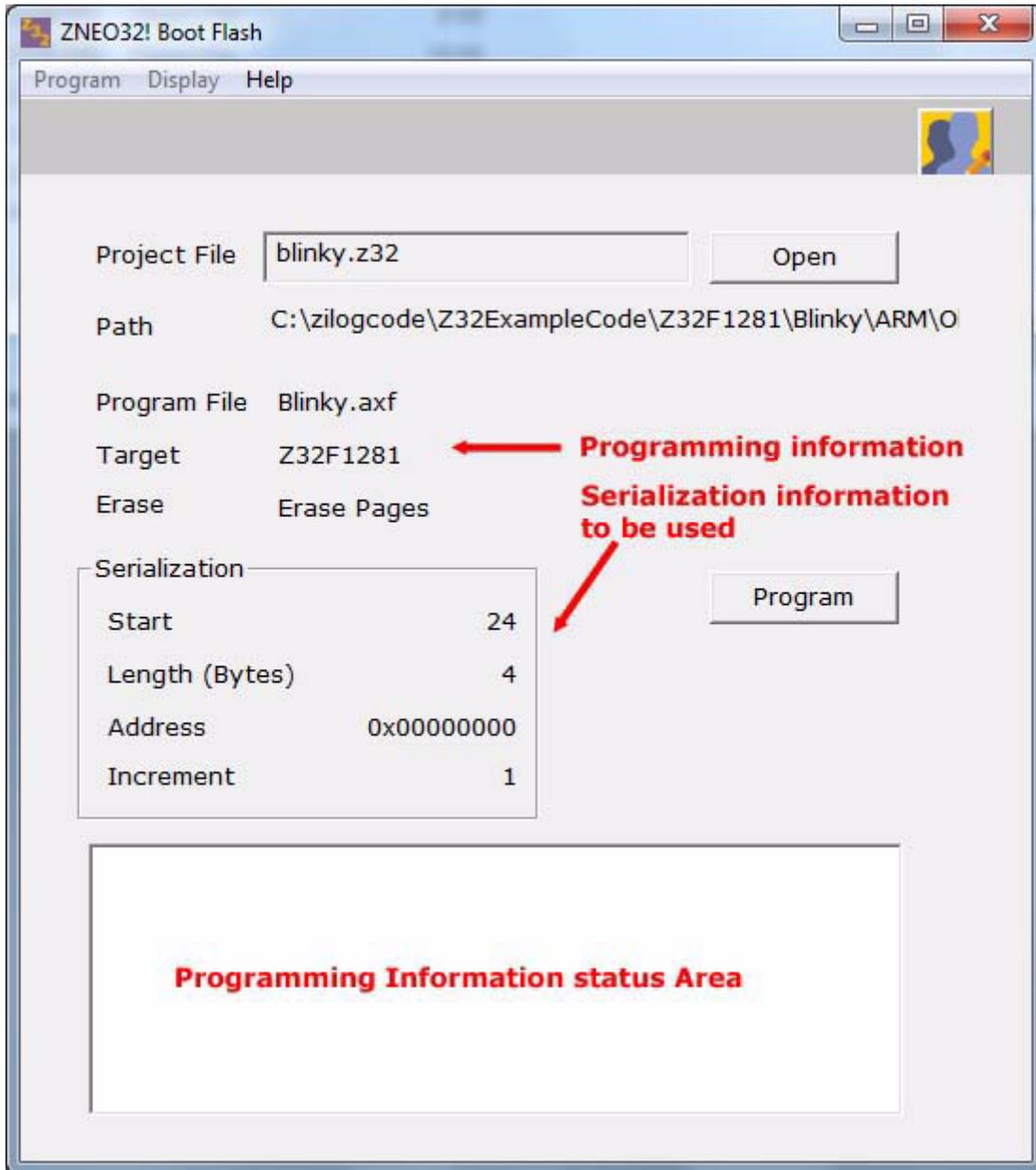


Figure 32. Project File Loaded Screen

---

► **Note:** No editing of the information is allowed. To change the data, return to Interactive mode, select **Program Device**, populate the correct information, and export the project file. Return to Production Mode and open the changed project file.

---

3. Connect all the devices to be programmed. All devices must be the same target device as displayed in the Target field.
4. Click the **Program** button. The system starts programming each of the devices that are found. A progress window is displayed and the results of the tasks are shown in the Programming Information status area, as seen in Figure 33.

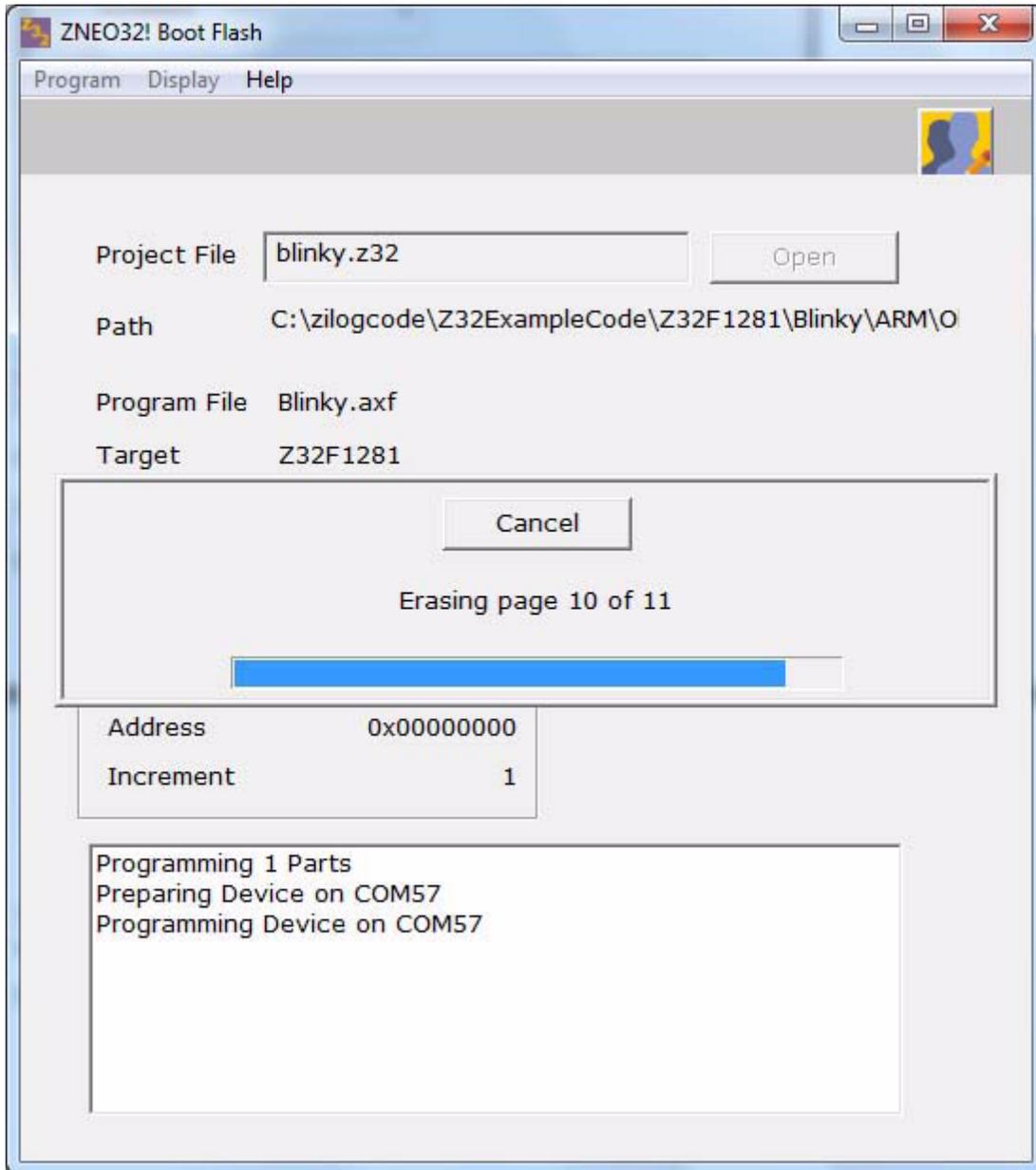


Figure 33. Programming Information Screen

5. Each part is programmed until no additional parts are found. If serialization is being used, the serialization start value is incremented and saved every time the programming is successful. If programming multiple devices at the same time, each device

will have a different serialization value. Figure 34 shows the completed programming screen.

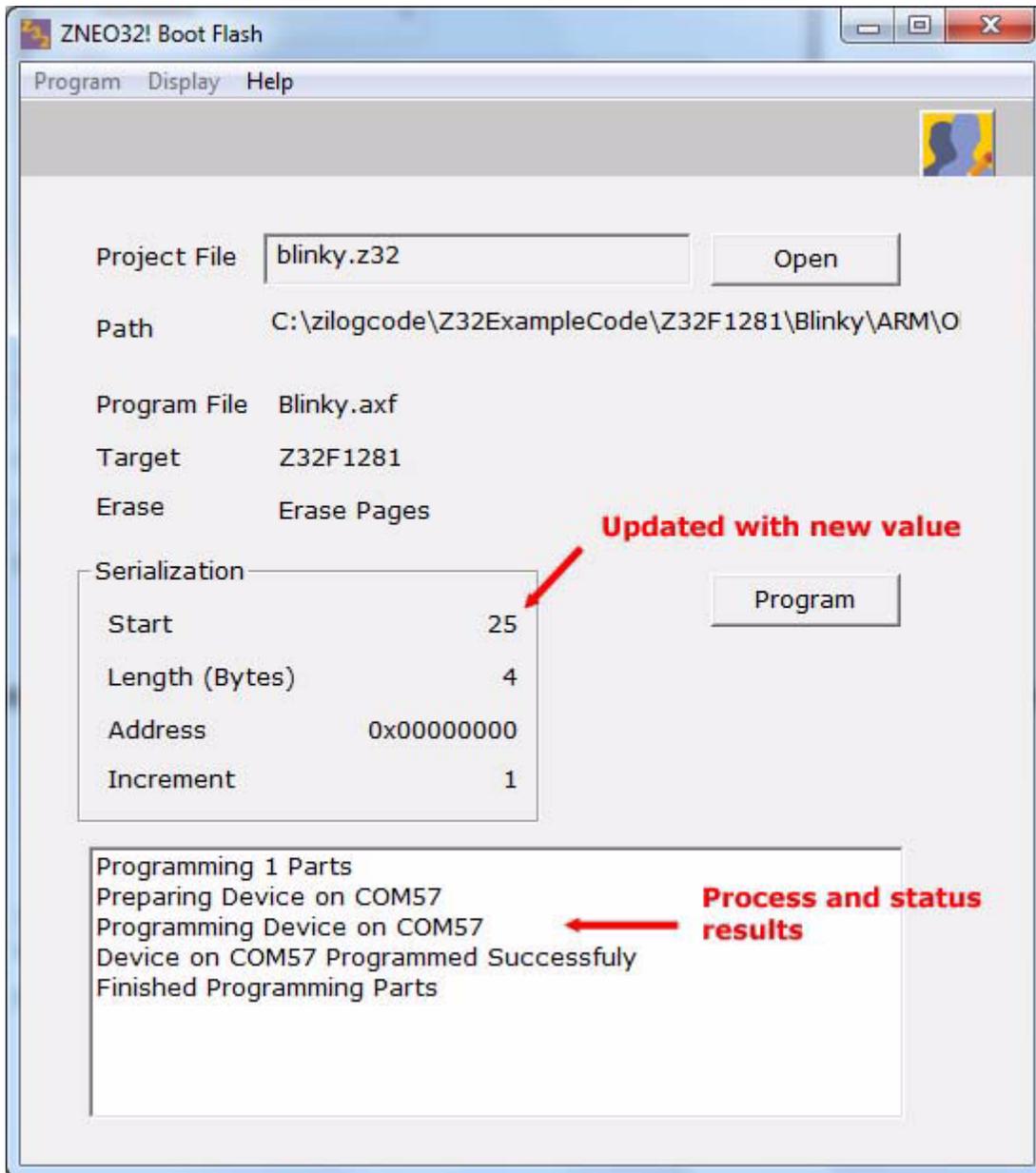


Figure 34. Programming Complete Screen

In Figure 34, the Programming Information status area shows the following details:

- The first line states the number of parts found by the system to program.
- The second line starts preparing the device by downloading the stage 2 boot code in preparation for programming.
- After the previous step is successful, programming starts, as seen from line three.
- The part is erased as defined by the Erase field, then programmed. The result of the programming is displayed in the line four.
- The last line shows that the Programming procedure exited correctly and signals that the process is completed.

The devices can be removed from the COM port and new ones connected up. Simply press the Program button to continue programming the same project file.

## Troubleshooting

The following issues may be encountered while using the ZNEO32! Flash loader:

- When trying to connect, an error message stating “No Boot connection found...” is displayed.  

Typically, this error is caused because a device is not in Boot mode. To place the device in Boot mode, connect the Boot Mode Pin to VSS (digital low) and reset or power up the device. The device must be connected via the device’s UART0 to the COM port of the PC. Depending on the operating system, the device may need to be reset after powering up.
- When trying to connect, an error message stating “Failed to connect to device” is displayed.  

Typically, this error is caused because the device is not responding to the request, which could be the result of a wrong device being selected, using an unsupported device, or a device in a hard fault. Try removing power and then power up again.

## Frequently Asked Questions

This section lists some of users' frequently asked questions when using the ZNEO32! Flash loader.

**Q.** What is the difference between *register* and *memory location*?

**A.** In this context, nothing. The configuration registers are memory locations. See the device's datasheet for the addresses of the different registers' memory locations.

**Q.** When I write to a register, why does the resulting value not match what was requested to be written?

**A.** Some registers do not allow writing to the register and/or some of the bits.

**Q.** Can different devices be programmed at the same time with Production programming?

**A.** Production programming programs all the devices that are on the COM ports; however, they must be the same device specified in the .z32 project file. Each .z32 project file includes only one device that can be programmed.

## Documentation

The documents associated with the ZNEO32! Flash Loader are listed in Table 1. Each of these documents can be obtained from the Zilog website by clicking the link associated with its Document Number.

**Table 1. ZNEO32! Flash Loader Documentation**

Document	Description
<a href="#">UM0280</a>	ZNEO32! Flash Loader User Manual and Installation Guide
<a href="#">PS0344</a>	Z32F064 Product Specification
<a href="#">PS0345</a>	Z32F128 Product Specification
<a href="#">PS0346</a>	Z32F384 Product Specification
<a href="#">FL0183</a>	ZNEO32! Evaluation Kit Insert

## Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at <http://support.zilog.com>.

To learn more about this product, find additional documentation, or to discover other facts about Zilog product offerings, please visit the [Zilog Knowledge Base](#) or consider participating in the [Zilog Forum](#).

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at <http://www.zilog.com/>