

AN45197

Author: Vihang Trivedi

Associated Project: No

Associated Part Family: CY7C68013A, CY7C68014A, CY7C68015A, CY7C68016A

[GET FREE SAMPLES HERE](#)

Software Version: None

Associated Application Notes: None

Application Note Abstract

Hex2bix is a program used to convert a .hex file to a raw binary, A51, or IIC format. This application note describes how to use the Hex2bix conversion utility for successful file conversion.

Introduction

Firmware in the .hex file format is loaded only into the internal RAM of the EZ-USB FX2LP™. It cannot be directly loaded into the EEPROM using the '.hex' format and must be converted to an IIC format. The Hex2Bix program is useful for this purpose.

The Hex2bix Conversion Utility

Although the Hex2bix is a simple conversion utility, a few aspects must be considered. The program is run from the command prompt after browsing to the location of the executable file. Typing "hex2bix -h" and hitting the 'Enter' key provides a list of command line options. This is the information needed to use the Hex2bix conversion utility. Figure 1 shows the results for "hex2bix -h" command.

Figure 1. Different Options in Using the Hex2bix Command

```
HEX2BIX [-AEIBRW?] [-IC] [-S symbol] [-M memsize] [-C Config0Byte] [-F firstByte]
[-O filename] [-X filename address] Source

Source - Input filename
A      - Output file in the A51 file format
B      - Output file in the BIX (raw binary) format (Default)
BI     - Input file in the BIX (raw binary) format (hex is default)
C      - Config0 BYTE for AN2200 and FX2 (Default = 0x04)
E      - Create .IIC file for External RAM. Prepends 0x200 byte loader that
        loads at 0x1e00
F      - First byte (0xB0, 0xB2, 0xB6, 0xC0, 0xC2) (Default = 0xB2)
H!?    - Display this help screen
I      - Output file in the IIC file format (sets -R)
IC     - Output Compressed IIC file (sets -R)
Ig     - Output Contiguous IIC file (without gaps) (sets -R)
M      - Memory size, used as BIX and IIC out file size. (Default=8K)
O      - Output filename
P      - Product ID (Default = 2131)
R      - Append bootloader block to release reset
S      - Public symbol name for linking
U      - Vendor ID (Default = 0x0547)
X      - Append Xtra data at fixed address in EEPROM. Input file is raw binary,
        address in hex.

C:\Cypress\USB\bin>
```

Figure 1 shows which extensions are used in the command to get the respective file formats. Detailed explanation on each of them is as follows.

- Depending on the entire command, the Hex2bix can convert the hex file to an A51 file format, a raw binary format, or an IIC file format. '-A', '-B', and '-I' extensions are used for output files in A51, raw binary, and IIC formats respectively. The default input file is usually a hex file.
- If the input file is in raw binary format, the extension '-BI' is used when typing the entire command.
- The '-E' option is used to create a file for the external RAM in IIC format.
- The '-R' option is used to append the boot load block to release reset.
- The '-H?' option is used to get the help menu shown in Figure 1.
- The extension '-IC' is used when a compressed IIC file is required as an output.
- '-S' followed by a symbol, is used to link a public symbol name.
- '-M' is used to specify the memory size. The default memory size is 8 KB. If the file size is more than 8 KB, it is specified using the '-M' extension followed by the actual memory size of the file.
- The default Config0 byte for FX2 devices is 0x04. If this needs to be changed, the '-C' extension is used followed by the byte. The Config0 byte can also be set as follows: 0x40—sets the DISCON bit to 1 and the device comes up as "disconnected"; 0x01—allows the IIC bus to run at 400 KHz.
- The '-F' option must be used to set the first byte to either 0xC0 or 0xC2. If it is not set to either of these bytes, then it is set to the default byte 0xB2. This is for

another part (AN2200) and is not recognized by the FX2LP.

- The input file (source) and output file ('-O') names must be specified with the correct file extension.
- The default PID and VID are 2131 and 0x0547 respectively. They can be changed, if required, by using the '-P' or '-V' commands followed by the desired PID or VID.
- Appending extra data at a fixed memory address in EEPROM is possible by using the '-X' extension followed by the filename and the memory address. Using these extensions in the correct way is important for successful file conversion.

Note The -P and -V feature, and not the default VID and PID used by the Hex2bix conversion utility, must be used to generate the final EEPROM image.

Getting Started

Before getting started, ensure that the CY3684 EZ-USB FX2LP Development Kit is completely installed on the computer. The Hex2bix utility tool is typically found under C:\Cypress\USB\bin. The program searches for the hex file in the directory where the program is located. Therefore, the hex file must be placed in the folder where program is located for the conversion to take place. Alternatively, copy the executable file, *Hex2bix.exe*, from C:\Cypress\USB\bin and place it in the directory where the hex file is located.

File Conversion Steps

The Hex2bix utility is run from the command prompt. The steps to convert a file are as follows:

1. Open the command prompt.
2. Browse to the folder where the *Hex2bix.exe* file exists. The default location for the Hex2bix file is C:\Cypress\USB\bin.
3. Enter the command for the conversion depending on the output file format, input file format, first byte, Config0 byte, and other requirements. A command extension is used only if applicable. A command can look as follows:

```
hex2bix -i myfile.iic myfile.hex
```

This means that the file *myfile.hex* must be converted into the IIC file format with the root name and IIC file extension, *myfile.iic*.

4. The output file is stored in the same folder as the input file. After entering the command, a series of responses is displayed, ending with "Conversion completed successfully."

An Example

Figure 2 and Figure 3 illustrate details of a successful file conversion. Figure 2 shows browsing to the correct source location along with the Hex2bix command used. Figure 3 shows the output of the Hex2bix utility in the command prompt. The command used in this example is:

```
hex2bix -i -f 0xC2 bulkloop.iic bulkloop.hex
```

In the command, the source file is *bulkloop.hex* and the output file format is IIC ('-I' extension is used); the output file name is *bulkloop.iic*. Because the output file is to be burnt into the EEPROM of the CY3684 EZ-USB FX2LP Development Kit, the EEPROM must have 0xC2 in the first byte to indicate a C2 load. The C2 load is an indicator to load the program that follows and execute it. Hence, the '-F' extension is used in this command to set the first byte of the output file as 0xC2.

Note The EZ-USB family has three different boot load scenarios. They are C0 load, C2 load, and No EEPROM. During power on reset, if the EZ-USB detects an EEPROM connected to its IIC bus with a value of 0xC0 at address 0, it automatically copies the VID, PID, and DID from the EEPROM to its internal storage. This data is then used by the EZ-USB device when the host sends out a *Get Descriptor-Device* request. This is called C0 load; only the custom VID, PID, DID, and the configuration byte are loaded to internal storage. If the value at address 0 is 0xC2, then it loads all the data in the EEPROM to the on-chip RAM. This is called C2 load; the entire firmware in the EEPROM is loaded onto the on-chip RAM and executed. During power on reset, if there is no serial EEPROM present, or an EEPROM is present with the first byte being neither 0xC0 nor 0xC2, it is called No EEPROM. In this case, the EZ-USB enumerates as a default USB device, loads the default ID bytes, and identifies itself with a VID of 04B4 and PID of 8613 which are reserved for the Cypress development environment.

Figure 2. Hex2bix Command Example

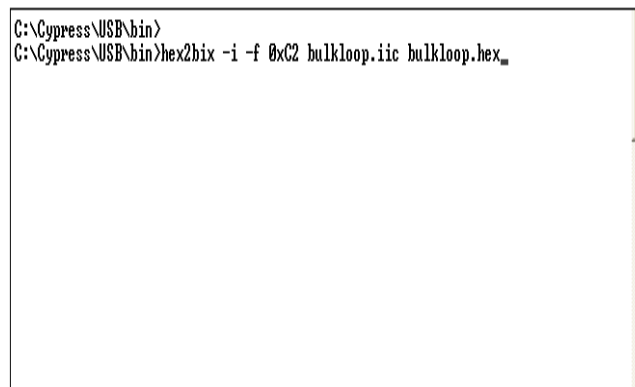


Figure 3. Hex2bix Program Output in Command Prompt

```

C:\Cypress\USB\bin>
C:\Cypress\USB\bin>hex2bix -i -f 0xC2 bulkloop.iic bulkloop.hex
Intel Hex file to EZ-USB Binary file conversion utility
Copyright (c) 1997-2005, Cypress Semiconductor Inc.
2266 Bytes written.
Total Code Bytes = 2225
Conversion completed successfully.

C:\Cypress\USB\bin>

```

As shown in [Figure 3](#), the output of the Hex2bix program in the command prompt also shows the number of bytes written and total code bytes before the message about successful completion of file conversion.

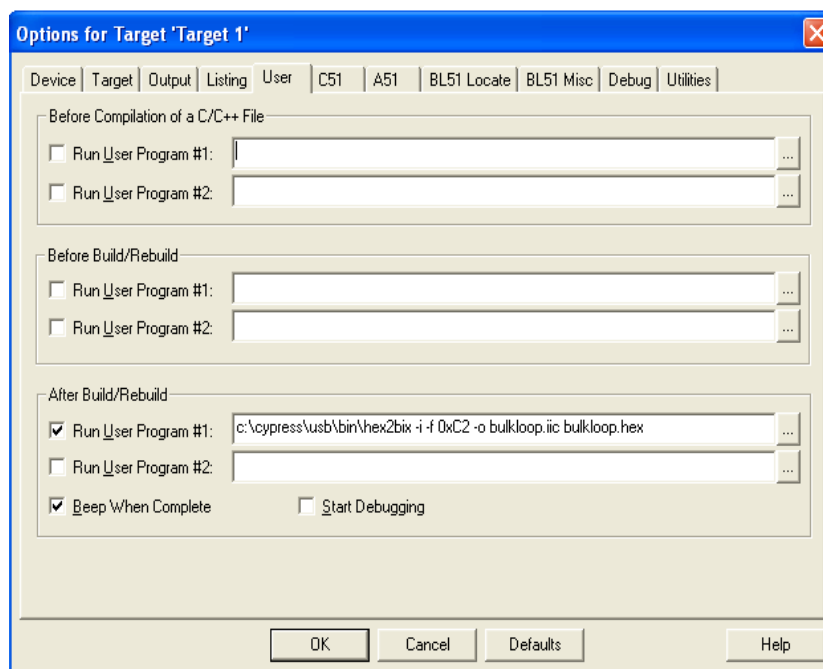
Note This type of Hex2bix conversion creates a file that can be loaded to the internal memory. To load that image to the external memory, a pre-loader must be used.

Alternative Method of Using Hex2bix Conversion Utility

The Hex2bix conversion utility is used from the command prompt. Another way of using it is with the Keil μ Vision3 or μ Vision2 Integrated Development Environment. This can be done at any stage of the project build. [Figure 4](#) and [Figure 5](#) show screenshots of the Keil μ Vision3 tool. The steps to use the tool are as follows:

1. In the "Project Workspace" section, right click **Target 1** and click **Options for Target 'Target 1'**.
2. A new window is displayed with all the options. In this window, click the User tab to view a screen as shown in [Figure 4](#). (The files, *bulkloop.hex* and *bulkloop.iic*, are used as an example).
3. Check the **Run User Program #1:** box under "After Build/Rebuild". Enter the following in the textbox: "c:\cypress\usb\bin\hex2bix -i -f 0xC2 -o bulkloop.iic bulkloop.hex". This indicates the location of the Hex2bix conversion utility followed by the actual command. There is no need to move the hex file physically to the folder where the Hex2bix conversion utility is located or vice versa.
4. Click the **OK** button. When the project is ready to be built, click the **Rebuild All Target Files** icon.
5. The output window shows the progress of the program compilation. After compiling, scroll up the output window to view the comment that was set in Step 3. The output window screen is shown in [Figure 5](#).

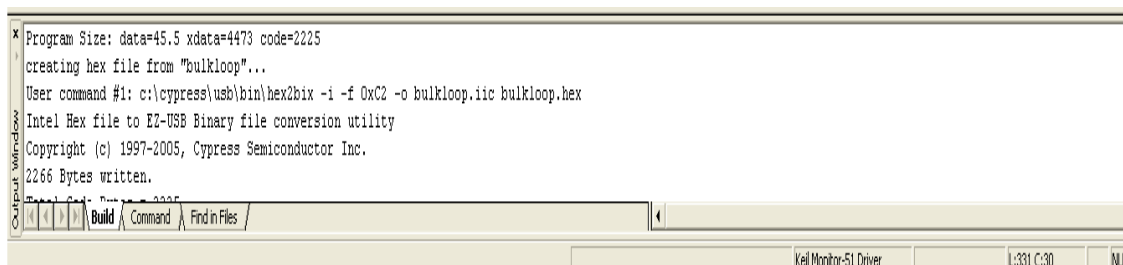
Figure 4. User Tab Under Options for Target "Target 1"



These steps also perform a successful Hex2bix file conversion. The output file in IIC format is saved in the same folder as the input file (in hex format). This method is simpler because only the location of the Hex2bix conversion utility needs to be specified along with the command line options used for a typical Hex2bix conversion. After it is set

up, the file path need not be modified again. The output file in IIC format is created when the entire project is rebuilt, along with the other files.

Figure 5. Output Window in Keil μ Vision3 IDE After Hex2bix File Conversion



```

X Program Size: data=45.5 xdata=4473 code=2225
creating hex file from "bulkloop"...
User command #1: c:\cypress\usb\bin\hex2bix -i -f 0xC2 -o bulkloop.iic bulkloop.hex
Intel Hex file to E2-USB Binary file conversion utility
Copyright (c) 1997-2005, Cypress Semiconductor Inc.
2266 Bytes written.

```

Summary

By using the options provided by the Hex2bix conversion utility correctly, a hex file can be converted to the desired IIC, A51, or BIX format. The Hex2bix program is convenient for quick conversion of a hex file.

About the Author

Name: Vihang Trivedi
Title: Applications Engineer
Background: Vihang Trivedi supports USB products at Cypress Semiconductor in San Diego, CA. He holds a Masters degree in Electrical Engineering from California State University, Chico.
Contact: vgt@cypress.com

Document History

Document Title: Using the Hex2bix Conversion Utility

Document Number: 001-45197

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2251966	VGT	03/26/08	New application note

EZ-USB FX2LP is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2008. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.