Generate Bootable Image File for FX3 with I2C EEPROM - KBA218344

Question: How do you generate an .img file for FX3 I2C booting with I2C EEPROM?

Answer: FX3 requires a specific bootable image format to boot the application firmware from I2C EEPROM. The possible boot options and bootable image formats for FX3 are documented in application note, AN76405**.**

By default, Eclipse, which is part of the FX3 SDK, generates the .elf file. Because FX3 requires a specified bootable image format, FX3 SDK invokes the Elf2img tool, which is provided with the FX3 SDK and is located at C:\Program Files (x86)\Cypress\EZ-USB FX3 SDK\1.3\util\elf2img, to build the .img file from the .elf file. Refer to the readme.txt in the directory for details about the elf2img tool usage. This article discusses the generating bootable image for all types of EEPROMs.

Section 5.3 of AN76405 describes the format of a bootable image file. The third byte of the bootable image file format, referred to as bImageCTL, is shown in Table 1. This explains to the bootloader the size, speed of the I2C EEPROM connected to FX3, and the type of file stored in EEPROM. Then, the bootloader generates the device address of the EEPROM by considering b3: b1 of the third byte.

Table 1. bImageCTL of Bootable Image File Format

Bit	Description
b0	0: execution binary file; 1: data file type
b3:b1 (I^2C Size)	7: 128 KB (Microchip) , 6: 64 KB (128K ATMEL ; 128K and 256K ST Electronics), 5: 32 KB, 4: 16 KB, 3: 8 KB, 2: 4 KB, 1 and 0: Reserved
b5:b4 (I^2C Speed)	00: 100 kHz, 01: 400 kHz, 10: 1 MHz, 11: Reserved
b7:b6	Bit7:6: Reserved; should be set to zero

The third byte needs to be modified based on the selected file type, EEPROM type, and desired speed. Assume that the desired file type is binary, and speed is 400 kHz; b0 is fixed to “0b” as the execution binary file and b5:b4 is fixed to “01b” for 400-kHz I2C speed. However, b3:b1 needs to be modified based on the size and manufacturer of the EEPROM. Let us consider the modification needed to create a bootable image file to boot from Microchip’s128 KB I2C EEPROM. The third byte value for the Microchip’s 128 KB I2C EEPROM and for the speed of 400 kHz will be “0x1E”.

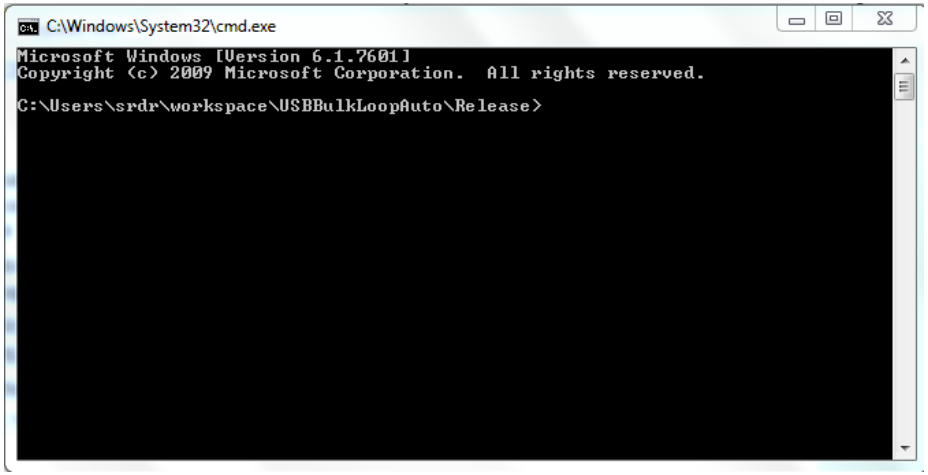
The third byte in the bootable image file format can be modified in the following ways:

1. **CMD Windows**

   a. Copy elf2img.exe (which is located at C:\Program Files (x86)\Cypress\EZ-USB FX3 SDK\1.3\util\elf2img in the default FX3 SDK installation path) into the same folder as the .elf file.

   b. Run CMD.exe as Administrator; navigate to the directory: C:\Users\srdr\workspace\USBBulkLoopAuto\Release, as shown in Figure 1

Figure 1. CMD.exe to the Directory



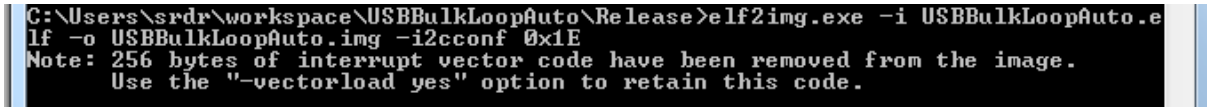
   c. Use the following command lines to generate binary image as shown in Figure 2.

```
elf2img.exe -I USBBulkLoopAuto.elf -o USBBulkLoopAuto.img -i2cconf 0x1E
```

   This shows that to generate a binary image file for USBBulkLoopAuto.elf, the output file USBBulkLoopAuto.img is the desired bootable image file for Microchip's128 KB I2C EEPROM.

   The third byte (bImageCTL) mapped for the selected EEPROM size and speed should be passed to **i2cconf** in the command line. This is reflected in the generated image file. In this example, **0x1E** is passed to the **i2cconf** for Microchip's 128 KB I2C EEPROM.

Figure 2. Command Line to Generate .img File for Microchip 128KB I2C EEPROM

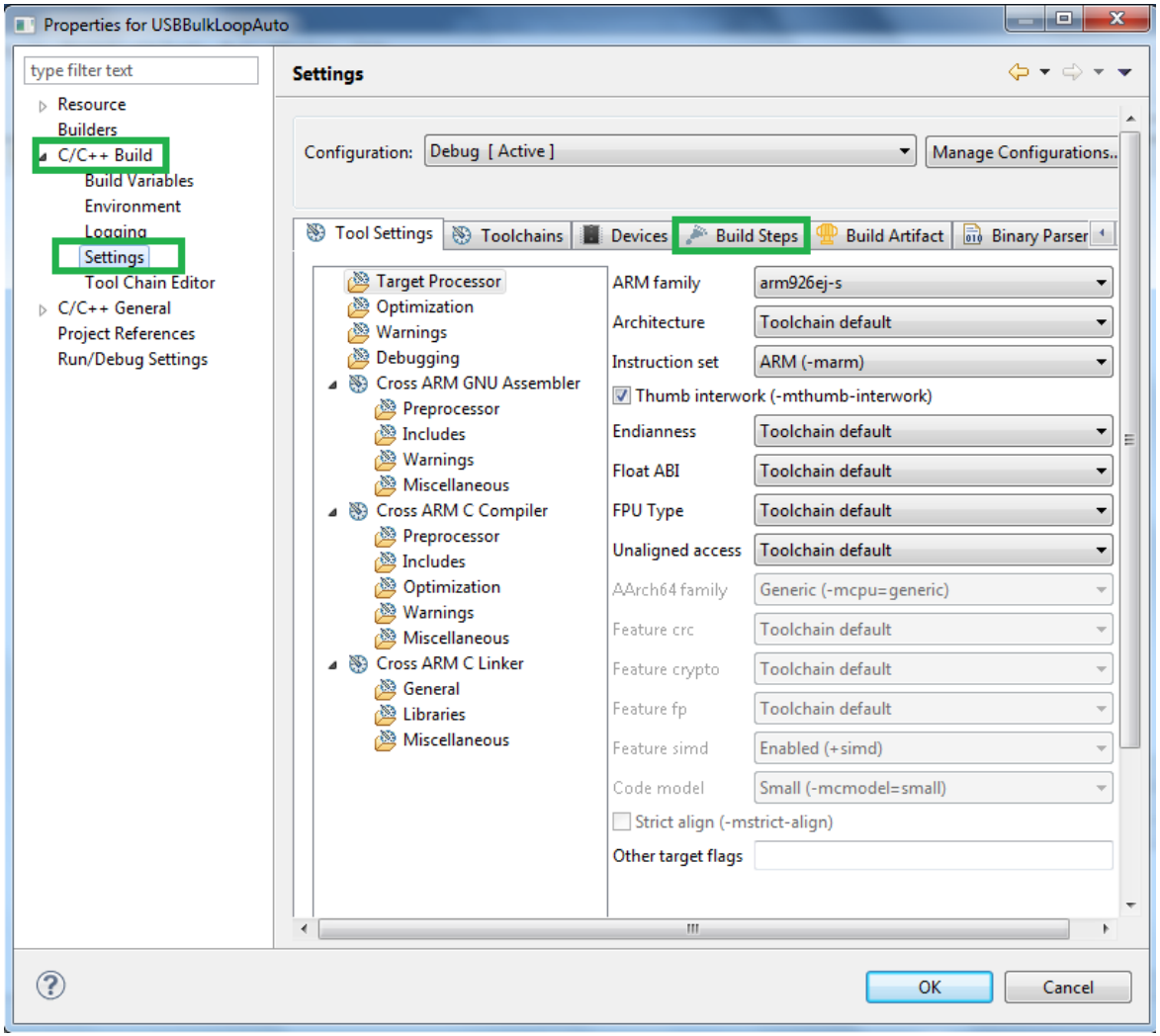


2. Eclipse IDE

Instead of using command lines, you can modify the post-build steps as explained here. This allows to build the desired image file from the SDK.

a. Right-click on the project and select **Properties**. The Properties window will open, as shown in Figure 3.

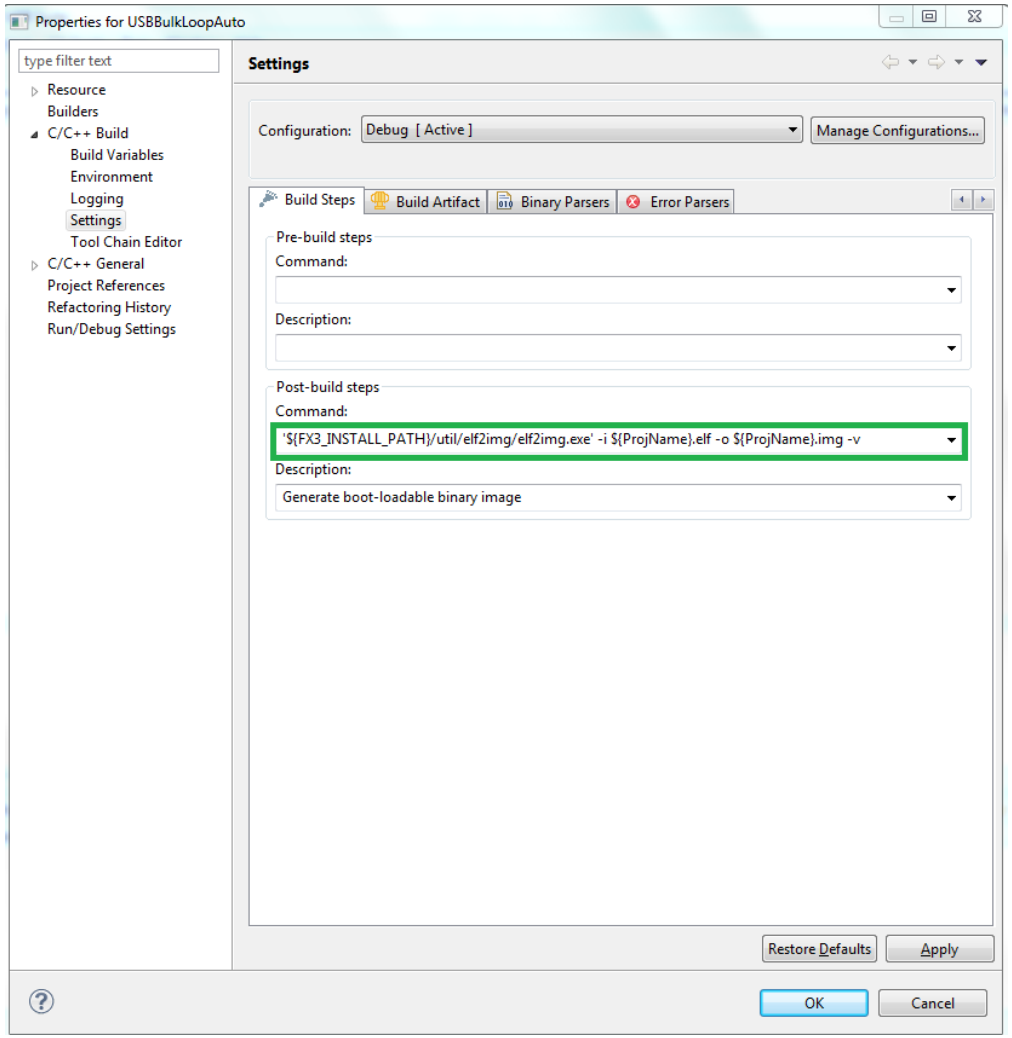
Figure 3. Properties Window in Eclipse



b. In the Properties window, select **C/C++ Bulid** > **Setting** > **Build Steps** as highlighted in Figure 3.

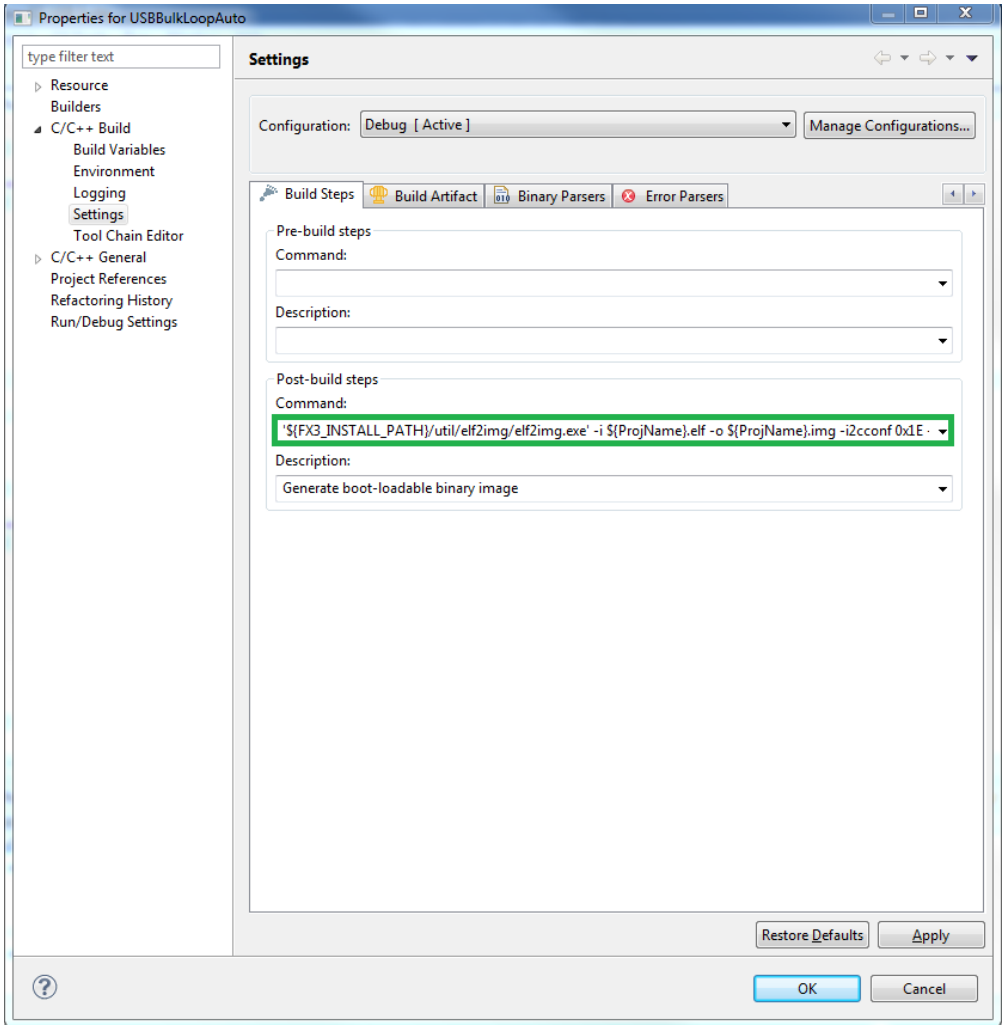
c. Add the I2C configuration parameter in the Command field as highlighted in Figure 4.

Figure 4. Command Lines in Eclipse



d. Follow the table and modify b3:b1 in the i2cconf parameter according to the selected I2C EEPROM in the post-build steps. This example is for Microchip's 128 KB I2C EEPROM, with file type as binary, and I2C speed as 400 kHz. Therefore, add "-i2cconf 0x1E" in the post-build settings as shown in Figure 5.

Figure 5. Add i2cconf 0x1E to Command Line



e. Click **Apply**; then, click **OK**. Build the project to generate the image file for I2C EEPROM (the build console is shown in Figure 6).