THIS SPEC IS OBSOLETE

Spec No: 001-68914

Spec Title: EZ-USB(R) FX3 I2C BOOT OPTION - AN68914

Sunset Owner: SONIA GANDHI (OSG)

Replaced by: 001-76405

EZ-USB® FX3 I^2C Boot Option

AN68914

Author: Shruti Maheshwari

Associated Project: No
Associated Part Family: EZ-USB® FX3
Software Version: None
Associated Application Notes: None

Application Note Abstract

AN68914 describes the features of the Cypress EZ-USB® FX3 I^2C boot option. It covers the details of selecting I2C boot option, devices supported by the bootloader, and the format of boot image to be loaded on the EEPROM.

Introduction

Cypress EZ-USB FX3 is the next generation USB 3.0 peripheral controller, which provides highly integrated and flexible features that enable developers to add USB 3.0 functionality to any system.

EZ-USB FX3 has a fully configurable, parallel, general programmable interface called GPIF II, which can connect to an external processor, ASIC, or FPGA. The GPIF II is an enhanced version of the GPIF in FX2LP, Cypress's flagship USB 2.0 product. It provides easy and glue less connectivity to popular interfaces such as asynchronous SRAM, asynchronous and synchronous address data multiplexed interface, and many others.

FX3 supports many boot options including booting over I2C, SPI, and USB interfaces. This application note discusses booting from I2C memories.

FX3 Boot Options

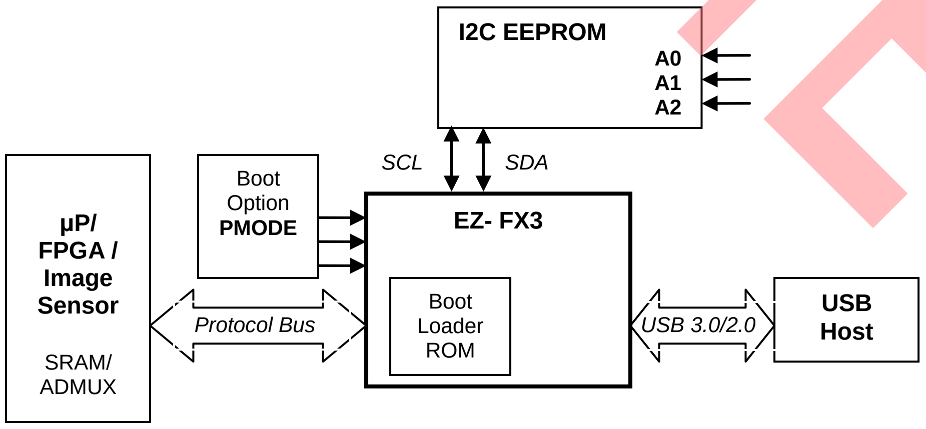
FX3 comes with a bootloader, which is the startup code for the ARM9 CPU that resides in masked ROM. The function of the bootloader is to download the FX3 firmware image from various interfaces such as I^2C EEPROM, SPI EEPROM, GPIF II asynchronous SRAM and asynchronous/synchronous ADMUX interfaces, and USB.

The boot options supported by the bootloader in EZ-USB FX3 are listed in Table 1.

Table 1. Boot Option Selection for FX3

PMODE Pins	Boot Option	USB Fallback
F00	Sync ADMUX (16-bit)	No
F01	Async ADMUX (16-bit)	No
F11	USB Boot	Yes
F0F	SRAM (16-bit)	No
1FF	I^2C	No
F1F	I^2C => USB	Yes
0F1	SPI => USB	Yes
Note: The shaded rows indicate pins that are relevant to this application note. Other combinations are reserved.		

Figure 1. High-Level Interface for FX3 Bootloader



PMODE Pins: The FX3 bootloader uses three FX3 pins to determine the booting options.

As shown in Table 1 on page 1, the three pins available can be driven high (1) or low (0) or no connection (F - Float).

The float detection capability of the bootloader provides 3^3 = 27 possible combinations that can be used as boot options.

This application note discusses the boot options highlighted in Table 1.

I^{2}C EEPROM Boot

PMODE Pins: 1FF

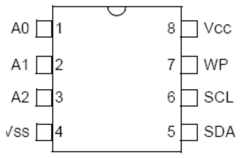
Features

- FX3 boots from I^{2}C EEPROM devices through a two-wire I^{2}C interface.
- EEPROM[1] device sizes supported are:
  - 32 Kilobit (Kb) or 4 Kilobyte (KB)
  - 64 Kb or 8 KB
  - 128 Kb or 16 KB
  - 256 Kb or 32 KB
  - 512 Kb or 64 KB
  - 1024 Kb or 128 KB
- Manufacturers supported are ATMEL and Microchip
- Supported boot frequencies are 100 kHz, 400 kHz, and 1 MHz. Note that when V_{IO5} is 1.2 V, the maximum operating frequency supported is 100 kHz. When V_{IO5} is 1.8 V, 2.5 V, or 3.3 V, the operating frequencies supported are 400 kHz and 1 MHz. (V_{IO5} is the I/O voltage for I^{2}C interface)
- Supports boot from multiple I^{2}C EEPROM devices of the same size. When I^{2}C EEPROM is smaller than the firmware image; multiple I^{2}C EEPROM devices must be used. The bootloader supports loading the image across multiple I^{2}C EEPROM chips. The bootloader can support up to eight I^{2}C EEPROM devices smaller than 128 Kbytes. The bootloader can support up to four I^{2}C EEPROM devices for a 128-Kbyte device.
- Only one firmware image is stored on I^{2}C EEPROM. No redundant images are allowed.
- Bootloader does not support the multimaster I^{2}C feature of FX3. Therefore, during the FX3 I^{2}C booting process, other I^{2}C masters should not show any activity on the I^{2}C bus.

Storing Image on EEPROM

The FX3 bootloader supports a master I^{2}C interface for external serial I^{2}C EPROM devices. The serial I^{2}C EEPROM can be used to store application specific code and data. The following diagram shows the pinout of a typical I^{2}C EEPROM.

Figure 2. Pinout of a Typical I^{2}C EEPROM



The I^{2}C EEPROM interface consists of two active wires: serial clock line (SCL) and serial data line (SDA).

WP is Write Protect and should be pulled low while writing the image on to EEPROM.

The A0, A1, and A2 pins are the address lines. They set the slave device address 000–111. This makes it possible to address eight I^{2}C EEPROMs of the same size. These lines should be pulled high or low based on the address required. Table 2 shows how eight 24LC256 EEPROM devices can be connected.

Table 2. 24LC256 EEPROM Device Connections

Device No.	Address Range	A2 A1 A0	Size
1	0x0000-0x7FFF	0 0 0	32 Kbytes
2	0x7FFF-0xFFFF	0 0 1	32 Kbytes
3	0xFFFF-0x17FFF	0 1 0	32 Kbytes
4	0x17FFF-0x1FFFF	0 1 1	32 Kbytes
5	0x1FFFF-0x27FFF	1 0 1	32 Kbytes
6	0x27FFF-0x2FFFF	1 1 0	32 Kbytes
7	0x2FFFF-0x37FFF	1 0 1	32 Kbytes
8	0x37FFF-0x3FFFF	1 1 1	32 Kbytes

For example, if the firmware code is 60 Kbyte, you must use two I^{2}C EEPROMs, with the first EEPROM having A<2:0> = 000 and second having A<2:0> = b001. The firmware image should be stored across the EEPROMs as a contiguous image as in a single I^{2}C EEPROM.

Important Points to Note on 128 KByte

In the case of a 128-Kbyte I^{2}C EEPROM, the addressing style is not standard. For example, Microchip uses pins A1 and A0 for chip select and pin A2 is unused. However, Atmel uses A2 and A1 for chip select whereas A0 is unused. Both these cases are handled by the bootloader. The addressing style can be indicated in the firmware header.

[1] Only support 2-byte I2C address. Single byte address is not supported for any I2C EEPROM size less than 32 Kbit

Table 3 shows how four Microchip 24LC1024 EEPROM devices can be connected.

Table 3. Microchip 24LC1024 EEPROM Device Connections

Device No.	Address Range	A2	A1	A0	Size
1	0x00000-0x1FFFF	Vcc	0	0	128 Kbytes
2	0x20000-0x3FFFF	Vcc	0	1	128 Kbytes
3	0x40000-0x5FFFF	Vcc	1	0	128 Kbytes
4	0x60000-0x7FFFF	Vcc	1	1	128 Kbytes

Table 4 shows how four Atmel 24C1024 EEPROM devices can be connected.

Table 4. ATMEL 24C1024 EEPROM Device Connections

Device No.	Address Range	A2	A1	A0	Size
1	0x00000-0x1FFFF	0	0	NC	128 Kbytes
2	0x20000-0x3FFFF	0	1	NC	128 Kbytes
3	0x40000-0x5FFFF	1	0	NC	128 Kbytes
4	0x60000-0x7FFFF	1	1	NC	128 Kbytes

Note: NC indicates No Connection

Boot Image Format

The firmware image should be stored on the EEPROM as follows:

Table 5. Firmware Image Storage Format

Binary Image Header	Length (16-bit)	Description
wSignature	1	Signature 2 bytes initialize with “CY” ASCII text
bImageCTL;	½	Bit0 = 0: execution binary file; 1: data file type Bit3:1 (I2C size) 7: 128KB (Micro chip) 6: 64KB (128K ATMEL) 5: 32KB 4: 16KB 3: 8KB 2: 4KB **Note** Options 1 and 0 are reserved for future usage. Unpredicted result will occurred when booting in these modes. Bit5:4(I2C speed): 00: 100KHz 01: 400KHz 10: 1MHz 11: 3.4MHz (reserved) **Note** Bootloader power-up default will be set at 100KHz and it will adjust the I2C speed if needed. Bit7:6: Reserved should be set to zero
bImageType;	½	bImageType=0xB0: normal FW binary image with checksum bImageType=0xB1: Reserved for security image type bImageType=0xB2: I2C boot with new VID and PID
dLength 0	2	1st section length, in long words (32-bit) When bImageType=0xB2, the dLength 0 will contain PID and VID. Boot Loader will ignore the rest of the any following data.
dAddress 0	2	1st sections address of Program Code not the I2C address. **Note** Internal ARM address is byte addressable, so the address for each section should be 32-bit align
dData[dLength 0]	dLength 0*2	All Image Code/Data also must be 32-bit align
…		More sections
dLength N	2	0x00000000 (Last record: termination section)
dAddress N	2	Should contain valid Program Entry (Normally, it should be the Start up code i.e. the RESET Vector)

Binary Image Header	Length (16-bit)	Description
		Note if bImageCTL.bit0 = 1, the Boot Loader will not transfer the execution to this Program Entry. If bImageCTL.bit0 = 0, the Boot Loader will transfer the execution to this Program Entry: This address should be in ITCM area or SYSTEM RAM area Boot Loader does not validate the Program Entry
dCheckSum	2	32-bit unsigned little endian checksum data will start from the 1st sections to termination section. The checksum will not include the dLength , dAddress and Image Header

Example: The binary image file is stored in the I²C EEPROM in the following order:

Byte0: “C”
Byte1: “Y”
Byte2: bImageCTL
Byte3: bImageType
…..
Byte N: Checksum of Image

Important Notes

- Bootloader default boot speed = 100 KHz; to change the speed from 100 KHz to 1 MHz, the bImageCTL<5:4> should be set to 10.
- To select I²C EEPROM size, the bImageCTL<3:1> should be used.

Checksum Calculation

The bootloader computes the checksum when loading the binary image I²C EEPROM. If the checksum does not match the one in the image, the bootloader does not transfer execution to the program entry.

The bootloader operates in little endian mode; for this reason, the checksum must also be computed in little endian mode.

The 32-bit unsigned little endian checksum data starts from the first sections to the termination section. The checksum does not include the dLength, dAddress and Image Header.

First Example Boot Image

The following image is stored only at one section in the system RAM of FX3 at the location 0x40008000:

```
Location1: 0xB0 0x1A ’Y’ ’C’          //CY Signature, 32KB EEPROM,400Khz,0xB0 Image
Location2:  0x00000004                //Image length =4
Location3:  0x40008000                // 1st section stored in SYSMEM RAM at  0x40008000
Location4: 0x12345678                 //Image starts
Location5: 0x9ABCDEF1
Location6: 0x23456789
Location7: 0xABCDEF12
Location8: 0x00000000                 //Termination of Image
Location9: 0x40008000                 //Jump to 0x40008000 on FX3 System RAM
Location 10: 0x7C048C04               //Check sum (0x12345678 + 0x9ABCDEF1 + 0x23456789 + 0xABCDEF12)
```

Second Example Boot Image

The following image is stored at two sections in the system RAM of FX3 at the location 0x40008000 and 0x40009000:

```
Location1: 0xB0 0x1A ’Y’ ’C’          //CY Signature, 32KB EEPROM,400Khz,0xB0 Image
Location2:  0x00000004                //Image length of section 1 =4
Location3:  0x40008000                // 1st section stored in SYSMEM RAM at 0x40008000
Location4: 0x12345678                 //Image starts (Section1)
Location5: 0x9ABCDEF1
Location6: 0x23456789
Location7: 0xABCDEF12                 //Section 1 ends
Location8:  0x00000002                //Image length of section 2 =2
```

Location9: 0x40009000	// 2nd section stored in SYSMEM RAM at 0x40009000
Location10: 0xDDCCBBAA	//Section 2 starts
Location11: 0x11223344	
Location12: 0x00000000	//Termination of Image
Location13 0x40008000	//Jump to 0x40008000 on FX3 System RAM
Location 14: 0x6AF37AF2	//Check sum (0x12345678 + 0x9ABCDEF1 + 0x23456789 + 0xABCDEF12+ 0xDDCCBBAA +0x11223344)

Similarly, you can have N sections of an image stored using one boot image.

The following section shows the checksum sample code:

```
// Checksum sample code
DWORD dCheckSum, dExpectedCheckSum;
WORD wSignature, wLen;
DWORD dAddress, i;
DWORD dImageBuf[512*1024];

fread(&wSignature,1,2,input_file); // read signature bytes
if (wSignature != 0x5943)          // check 'CY' signature byte
{
   printf("Invalid image");
   return fail;
}
fread(&i, 2, 1, input_file);       // skip 2 dummy bytes
dCheckSum = 0;
while (1)
{
   fread(&dLength,4,1,imput_file);  // read dLength
   fread(&dAddress,4,1,input_file); // read dAddress
   if (dLength==0) break;           // done
   // read sections
   fread(dImageBuf, 4, dLength,  input_file);
   for (i=0; i<dLength; i++) dCheckSum += dImageBuf[i];
}
// read pre-computed checksum data
fread(&dExpectedChecksum, 4, 1, input_file);
if  (dCheckSum != dExpectedCheckSum)
{
   printf("Fail to boot due to checksum error\n");
  return fail;
}
```

I^{2}C EEPROM Boot with USB Fallback

PMODE Pins: 1FF

In all USB Fallback modes ("=>USB"), USB is enumerated if 0xB2 boot is selected or an error occurs. After USB is enumerated, the external USB host can boot FX3 using USB Boot. I^{2}C EEPROM boot with USB Fallback (I2C => USB) is also used to store Vendor Identification (VID) and Product Identification (PID) for USB Boot.

I^{2}C EEPROM boot fails under the following conditions:

- I^{2}C address cycle or data cycle error.
- Invalid signature on FX3 firmware. Invalid image type.
- A special image type is used to denote that instead of the FX3 firmware image, data on EEPROM is the VID and PID for USB boot. This helps in having a new VID and PID for USB Boot.

Notes

- On USB boot, the bootloader supports only USB High Speed and USB Full Speed. USB30 Super speed is not supported.
- In the 0xB2 boot option, the USB descriptor uses the customer defined VID and PID as part of the 0xB2 image from I^{2}C EEPROM.
- If USB falls back when any error occurs during I^{2}C Boot, the USB descriptor uses the VID=0x04B4 and PID=0x00F3.
- The USB Device Descriptor is reported as BUS-power, which consumes around 200 mA. The FX3 chip itself consumes around 100 mA.

Example Image for boot with VID and PID

```
Location1: 0xB2 0x1A 'Y' 'C'        //CY Signature,32k EEPROM,400Khz,0xB2 Image
Location2: 0x04B40008               // VID = 0x04B4 | PID=0x0008
```

Summary

The details of the I^{2}C boot option supported by the FX3 bootloader have been discussed in this application note. The application note enables you to select an appropriate EEPROM device, store boot image in the EEPROM, and boot FX3 over the I^{2}C interface.

About the Author

Name:	Shruti Maheshwari
Title:	Systems Engineer Senior
Contact:	svrm@cypress.com

Document History

Document Title: EZ-USB® FX3 I^{2}C Boot Option – AN68914

Document Number: 001-68914

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3220026	SVRM	04/07/2011	New application note.
*A	3284799	SVRM	06/14/2011	Removed FX3 system level diagram. Added second example of Boot Image.
*B	3358203	SVRM	08/30/2011	Added new point in Features section about multimaster support during booting.
*C	3980631	OSG	04/24/2013	Obsolete document.

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