

AN15456

Author: Kristopher Young

Associated Project: No

Associated Part Family: CY7C68013A, CY7C64713

Application Note Abstract

This application note outlines a process that catches many of the most likely causes of EZ-USB FX2LP™ and EZ-USB FX1™ hardware problems. It also facilitates the process of catching potential problems before building a board and assists in the debugging when getting a board up and running.

Introduction

The EZ-USB FX2LP and EZ-USB FX1 devices are an excellent solution for adding a high performance high-speed or full-speed USB connection to a design. While these devices are fairly easy to use, sometimes getting a new board up and running is tricky due to the various details that need to be carefully followed to get the hardware functional. This application note not only helps catch potential problems before building a board, but also helps debug the hardware once the board is created. The goal is to get the board ready to download firmware and begin the firmware development process.

Note In this application note, FX2LP is used to refer to both the FX2LP and the FX1 devices except where noted.

Hardware Checklist

The following is a list of items that are critical for successful operation of the FX2LP. Go through this checklist before creating a printed circuit board using the FX2LP. If a board already built is not behaving properly, go through this list to verify that all the items are being implemented correctly on the target.

1. All power pins (AVCC or VCC) are powered to a proper voltage level (3.0V to 3.6V).
2. VCC ramp up time is at least 200 μ s, with a maximum ramp rate of 18V per millisecond.

Most boards have enough capacitance on VCC to meet this requirement. Many regulators also ramp up VCC fairly slowly. However, occasionally a board has an especially fast ramp up time. In such a case, extra capacitance on VCC slows it down to meet the requirements.

3. When using a crystal, reset remains asserted for at least 5 ms past the point that VCC reaches 3.0V. When using an external clock source, this hold time is at least 200 μ s. If reset is asserted when the FX2LP is already powered, the reset is held asserted for at least 200 μ s.

4. The oscillator on the FX2LP requires a crystal or resonator with a load capacitance of 12 pF and a frequency of 24 MHz (\pm 100 ppm). The crystal is required to handle a drive level of at least 500 μ W. Two crystals used successfully with the FX2LP are the eCera™ FX2400026 and the Ecliptek™ EC-12-24.000M. 12 pF load capacitors are required between each crystal pin and ground.

5. The RESERVED pin is grounded.

The RESERVED pin is a test mode pin. If it is not grounded, the FX2LP is placed into a test mode and does not operate correctly.

6. Make certain the SCL and SDA pull up resistors are installed.

If there are no pull ups on SCL and SDA, the serial interface engine (SIE) hangs up immediately on booting. The FX2LP looks for a serial EEPROM on SCL and SDA when booting to determine the mode of operation. If the lines are pulled up with no EEPROM present, the SIE enumerates as the default USB device that enables connection to the CyConsole application. However, with no pull up resistors, the FX2LP believes the bus is controlled by another master and waits indefinitely for the other master to release the bus. The recommended value of the pull up resistors is 2.2K ohms. A method of temporarily breaking the connection between SDA of the FX2LP and SDA of the EEPROM should be included on the board, especially during development. This enables the reprogramming of a corrupt EEPROM without having to remove it from the board. This disconnect can be a jumper, switch, or a removable zero ohm resistor.

7. EA is tied to ground unless external Flash or ROM is connected to the external memory bus.

If EA is tied high, the FX2LP immediately attempts to boot from code stored in external memory. If there is no code in external memory, the device cannot boot.

- The WAKEUP# pin is tied high or low and is not left floating.

A floating WAKEUP# pin causes erratic suspend behavior. Holding the WAKEUP# pin low inhibits the FX2LP from entering suspend mode when no USB traffic is detected. WAKEUP# is typically tied high with a 10K or 100K ohm resistor.

Bringing Up the Board

When bringing up the board of a new FX2LP design, start with a board that has only a minimum set of components installed. Populate the board with a minimum number of components as follows:

- FX2LP or FX1 chip.
- 3.3V regulation circuit connected to all AVDD and AVCC pins.
- Two pull up resistors (2.2K ohm, or other value compliant to the I²C™ specification), one each on the SCL and SDA lines.
- EEPROM is blank or is not installed.
- Connect the WAKEUP# line to ensure it is not floating. Pulling it high with a 10K or 100K ohm resistor is preferred for easier debugging.
- EA pin is connected to logic low.
- RESERVED pin connected to ground/low.
- Reset circuit that provides the required minimum reset timing.
- Crystal oscillator circuit (including crystal and load capacitors) or external clock source.

The purpose of hardware debugging is to get the default SIE of the FX2LP chip communicating with the EZ-USB interface within the CyConsole application running on a PC. The default SIE is how the device boots up when there is no EEPROM connected, or the connected EEPROM is blank. Connectivity to CyConsole is an indication that the FX2LP is running and the USB path is functional. The CyConsole is also used to execute firmware on the chip for testing and development.

When the board is ready for testing, connect it to a PC on which the CY3684/CY3674 development kit is installed. If everything is working properly, CyConsole shows a connection to the device. If enumeration is not successful, determine the cause of the problem using the following procedure.

To help solve an FX2LP enumeration problem, use an oscilloscope to look for a 12 MHz signal on the CLKOUT pin while the board is powering up. The following four scenarios of CLKOUT behavior helps direct the debugging efforts in the right direction:

- CLKOUT never outputs 12 MHz.

In this situation, the FX2LP is either not powering up correctly, or it is not receiving a reference clock. Check the VCC/AVCC pins, VCC ramp rate, crystal oscillator circuit, and power on reset timing. Also double check to make sure the RESERVED pin is grounded.

- CLKOUT outputs 12 MHz when powering up, but the 12 MHz disappears after a short period of time.

The cause of this behavior is that the part is entering suspend mode. Check to make sure the WAKEUP# pin is not floating. Suspend mode happens if the FX2LP is not seeing the regular SOF signals from the host and WAKEUP# is tied high. Make sure the unit is plugged into the host and that D+ and D- connect from the host to the device with the correct polarity and with no external series resistors in between. Suspend mode can also occur if the reference clock (either driven externally or from the crystal oscillator) is outside the tolerated accuracy range. Check the frequency of CLKOUT to make sure it is 12 MHz +/- 100 ppm.

- CLKOUT continues to output 12 MHz, but enumeration fails.

If the WAKEUP# pin is tied low (inhibiting suspend), the device may not be seeing the signals from the host. Please check all the items in the previous case where CLKOUT is active but quickly turns off.

If the WAKEUP# pin is tied high (or if it is tied low and you have double checked all the previous items), check to make sure there are pull ups on SCL and SDA and no EEPROM is connected. Also make sure EA is tied low.

If these things check out and all the other checklist items are okay from the first section, then there is a good chance that there is some sort of problem with the operating system drivers. The next section describes some techniques to debug these kinds of driver problems using the Windows® Device Manager.

- CLKOUT outputs 48 MHz, the device enumerates, but does not connect to CyConsole.

This is a symptom of the DVB-T driver issue discussed in the following section. Windows Update may have automatically bound the default SIE to a different driver and loaded third party firmware into the device. Read for instructions on how to clear this driver and reattach to the proper Cypress driver.

Drivers and the Device Manager

Even if the hardware is working properly, it is possible for the device to not get linked properly to CYUSB.SYS. If this link is not made, CyConsole is unable to connect to the device. The Windows Device Manager is used to examine these kinds of problems, and is sometimes used to fix them.

The Device Manager is found under the System icon in the Windows control panel. On Windows 2000, 2003, XP, and Vista, a shortcut is to type “*Devmgmt.msc*” into the “Run...” box under the Start menu. As a USB hardware developer, there are times when this utility is used frequently. Therefore, it’s a good idea to create a shortcut for this useful utility for fast and convenient access in the future.

In the Device Manager, all the USB devices are seen connected to the PC in a list under “Universal Serial Bus controllers”. On operating systems later than Windows 2000, you see the VID and PID for a connected device under “Device Instance ID” in the Details tab of the properties for the device. Windows 2000 does not have this feature, so to observe the VID and PID in Windows 2000, you have to find a copy of the USBCheck utility (previously available on usb.org) or the USBView utility from the Windows 2000 DDK. A black exclamation point within a yellow circle (commonly called a “bang”) indicates that a device is having some sort of difficulty. A bang on a USB device usually has something to do either with a problem during enumeration or a driver issue.

If enumeration to the default SIE is successful, the Device Manager shows a USB device connected called “Cypress EZ-USB FX2LP - EEPROM missing”, with VID=0x04B4 and PID=0x8613. The FX1 is similar, except with a PID of 0x6473. This indicates that the default SIE of the device is enumerated properly. If it appears this way in the Device Manager but is still not showing up in the CyConsole, then for some reason it could not be linked to the CYUSB.SYS driver. To fix this, click on “Update Driver...” on the Driver tab and browse to `\$windows\inf\cyusbpre.inf` (where \$windows is the directory where Windows is installed, usually either “windows” or “winnt”). Once the device becomes associated with the CYUSB.SYS driver, it is recognized by CyConsole.

If you get a “USB Device Not Recognized” error (see Figure 1) when the device is first plugged in and the Device Manager shows an Unknown device has connected with a zero for both the VID and PID (see Figure 2), the host has detected the connection of a device via the device’s internal pull up assertion on D+ but the first step of enumeration (reading the VID and PID from the device) was not successful. This problem can be caused by the hardware problems mentioned in the previous section, by a corrupt EEPROM, or by a firmware error—if code is loaded from a connected EEPROM or flash.

Figure 1. Error Message Due to Enumeration Failure

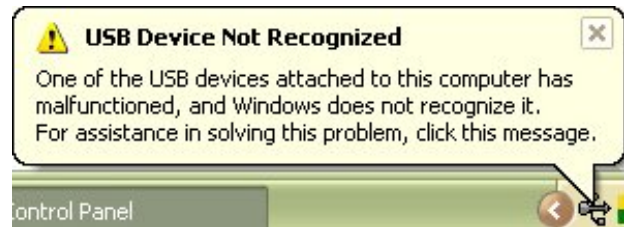
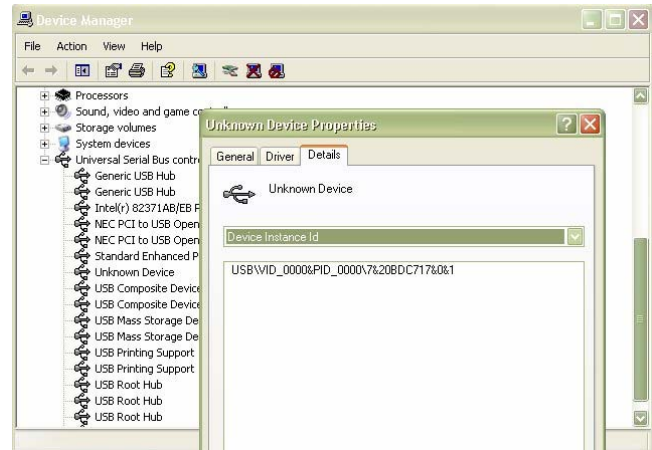


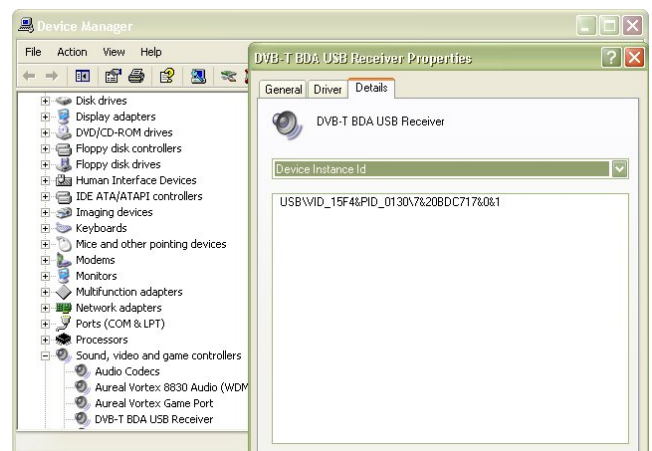
Figure 2. Unknown Device with a Zero VID and PID



There is also a potential problem because the VID and PID for the FX2LP EEPROM missing configuration were inadvertently put into the Windows Driver Update as associated with a third party device (DVB-T). If the device is linked to this other driver (see Figure 3), you must clean out the registry and then manually bind the device to the CYUSB.SYS driver.

Note This only affects the FX2LP. The FX1 is not susceptible to this issue since its VID/PID combination is not bound to this driver through the Windows update.

Figure 3. FX2LP Bound to the DVB-T Driver



The process required to fix the DVB-T driver issue follows:

1. Disconnect your board from the PC’s USB port.

- If you are inexperienced with registry editing, back up the registry before this step. Go into the registry (by typing “**regedit**” in **Start...Run**) and find the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB` folder. Locate the key/folder labeled “`Vid_04b4&Pid_8613`”. Right click on this key and set the security permissions to enable full control. Delete the key and any subkeys. Repeat these steps for:

`HKEY_LOCAL_MACHINE\SYSTEM\ControlSetxxx\Enum\USB` (where xxx is a three digit number from 001 to 004). These alternate control sets may or may not be present.

- Find the `oemxx.inf` file in the `\$windows\inf` directory (where xx is a two digit number) that contains the following line:

```
%CEBDALDR.DeviceDesc%=LOADER.Device,    →
USB\VID_04B4&PID_8613
```

Delete this INF file, and the `oemxx.PNF` file, where xx is the same two digit number as the INF filename.

- Reconnect your board to the PC’s USB port. When prompted, choose “**Install from a list or specific location**”, then in the next window choose “**Don’t search. I will choose the driver to install**”. If you are asked for hardware types, choose “**show all devices**”. On the next prompt, click “**Have Disk...**” and choose `\$windows\inf\CYUSBPRE.INF`. If you see an unsigned driver warning, click “**Continue Anyway**”.
- Once the process completes, the device manager should show the FX2LP with EEPROM missing, and should indicate that it is bound to `cyusb.sys`. You are ready to run CyConsole.

CyConsole and the EZ-USB Interface

The FX2LP with a blank or no EEPROM installed is bound to the `CYUSB.SYS` driver. The CyConsole is used to get descriptors, program the EEPROM, or download a program to the FX2LP RAM. To do this, go to the Options menu of the CyConsole application and choose the EZ-USB interface. For more information on using CyConsole, refer to `$dvc\doc\general\CyConsole.PDF` (where \$dvc is the installation directory of the CY3684/CY3674 development kit).

To perform a thorough test of the hardware, load the `bulkloop` example firmware onto the FX2LP on the board and run it. This is done with the download button in CyConsole (select `bulkloop.hex` from the `$dvc\Examples` directory). The `bulkloop` example configures the FX2LP to perform USB bulk transfer loopback. `CyBulk.exe`, is an application installed in the `$dvc\bin` directory that is associated with the `bulkloop` example. `CyBulk.exe` sends bulk data to the FX2LP and tests the data returned to make sure it is the same as was sent. Running the `bulkloop` example verifies that the FX2LP is operating properly and the communication path to the USB host is clear.

Conclusion

The FX2LP high speed USB peripheral controller and FX1 full speed USB peripheral controllers are powerful devices that offer a significant amount of functionality. When doing a design with the EZ-USB FX2LP or FX1, refer to the guidelines of this application note before sending the board for manufacture. If there are problems getting the board up and running the process outlined in this application note enables you to solve most problems quickly and move on to the important task of firmware development.

EZ-USB FX2LP, and EZ-USB FX1 are trademarks of Cypress Semiconductor Corp. I2C is the trademark of Philips Electronics. Ecliptek is the trademark of Ecliptek Corporation. eCera is the trademark of SaRonix-eCERA Corporation. Windows is a register trademark of Microsoft Corporation. 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, 2007. 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.