Linux_FUT

Introduction

Linux_FUT is a C application for the GNU/Linux operating system based on Trenz Electronic API commands and libusb C API to program USB FX2 microcontroller firmware and FPGA bitstreams.

> **libusb and libusbx merged**
>
> libusb and libusbx release will be officially merged from v1.0.19. These two project/libraries are already actually ("unofficially") merged.

Linux_FUT can be downloaded from our GiiHub repository.

> **Generation 2 or Generation 3 support; it depends on a #define**
>
> Device Ids of the usb (lsusb): a C construct (#define) is used to choose between Generation 2 (DEWESoft device) and Generation 3 (Trenz Elecronic device).
>
> The user should compile and use the Linux_FUT.c file with the correct #define uncommented or the SW tool will not work.

Linux_FUT.c

```
// TE USB FX2: generation 2.0
// #define VENDOR_ID 0x0547
// #define PRODUCT_ID 0x1002


// TE USB FX2: generation 3.0
#define VENDOR_ID 0x0BD0
#define PRODUCT_ID 0x0300
```

Firmware Upgrade/Change and/or FPGA Configuration

Linux_FUT can be used (with Generation 2 and/or Generation 3 firmware) for:

- firmware update (USB EEPROM programming while the USB FX2 microcontroller is running with Trenz Electronic VID/PID): you should use a .bin file but a .iic file (EZ-USB FX2LP USB FX2 microcontroller firmware) should also work;
- FPGA configuration (SPI Flash and FPGA programming): only .bin file is supported.

> **Necessary files**
>
> The two files needed for firmware update and FPGA configuration are usb.bin and fpga.bin.
>
> In the past, these two files were normally packed together in a file with the extension .fwu . This file was a zip-file.
>
> The usb.bin was delivered by the company Trenz Electronic.
>
> The FPGA.bin was generated with the Xilinx ISE by the user.

Firmware Recovery

Linux_FUT cannot be used for firmware recovery. Firmware recovery is necessary when USB connection (with host computer) is nonresponsive.

Terminology: USB FX2 unresponsive aka USB FX2 microcontroller stall

USB FX2 microcontroller stall term is used to describe the situation where your TE USB FX2 module's FPGA works normally (you are still able to connect to FPGA using JTAG connection) but you are unable to connect to it via USB connection despite having the correct USB drivers installed.

Writing wrong (or corrupted) firmware to EEPROM will (normally) bring the USB FX2 microcontroller to a stall when the EEPROM firmware is running in FX2 microcontroller's RAM.

FX2 microcontroller's stall will hinder/prevent host computer USB connection (USB connection unresponsive) with TE USB FX2 module.

To bring FX2 microcontroller's back out of this stall and re-establish the host computer USB connection with TE USB FX2 module two firmware recovery procedure are possible:

- use an implicit two step recovery boot (Windows);
- use an explicit two step recovery boot (mainly Linux, for Windows the implicit two step recovery boot (is recommended).

Under Windows, the user should instead use CyControl, CyConsole or OpenFutNet for a implicit two step recovery boot. After this procedure, the user could use Linux_FUT again.

Under Linux, the user should instead use fx2loader (see also here) or fxload (see also here) for an explicit two step recovery boot. After this procedure, the user could use Linux_FUT again.

It's a good idea to load (and test) custom firmware into FX2 microcontroller's RAM before writing the new firmware into EEPROM; in this way the user/developer could safely test the new firmware.