

Firmware Layer (Generation 2 = Generation 3, only VID/PID are different)

Generation 2 and Generation 3 technology stacks share the same reference firmware but with two different VID/PID couples:

- 0547/1002 (Generation 2);
- 0BD0/0300 (Generation 3).

The firmware used (running on the FX2 microcontroller) is normally (EEPROM switch enabled) retrieved from the EEPROM when a [reset](#) occurs. It is also possible to directly write the RAM of the FX2 microcontroller to change the current running firmware.



A general introduction to USB bulk transaction and FX2 microcontroller firmware is available [here](#).

TE USB FX2 modules are shipped with a preloaded FX2 USB microcontroller reference firmware inside the EEPROM. For a description see [here](#).

The preloaded firmware exposes Trenz Electronic Firmware API ([TE API Commands \(FW APIs\)](#)) to the host computer software (through USB).

To change the EEPROM content ([reference firmware](#)) from Generation 2 (VID/PID 0547/1002) to Generation 3 (VID/PID 0BD0/0300), you shall follow this [procedure](#). You may also watch this [video play list](#).

For an introduction to USB FX2 microcontroller's [firmware configuration](#), [recovery boot](#) and [update boot](#).



For Custom FX2 microcontroller's firmware considerations and advices see [here](#).

Firmware loading and/or update comparison table

EEPROM switch ⁽²⁾ status when a reset occurs	disabled ⁽³⁾	enabled ⁽⁴⁾		
generation	2/3/custom	2	3	custom
boot mode	firmware recovery boot	regular or firmware update boot	regular or firmware update boot	regular or firmware update boot
VID	04B4	0547	0BD0	any, but 0BD0 is advised
PID	8613	1002	0300	any, but 0300 is advised
device enumerated	Cypress generic USB	DEWESoft TE03xx USB	Trenz Electronic TE USB FX2	any or Trenz Electronic TE USB FX2
required driver (Windows OS)	recovery driver (Cypress default, using Cypress or TE SW libraries) or WinUSB (using libusb library) ^(5a)	regular driver (Gen 2, using Cypress or TE SW libraries) or WinUSB (using libusb library) ^(5a)	regular driver (Gen 3, using Cypress or TE SW libraries) or WinUSB (using libusb library) ^(5a)	custom (using custom or libusb library), WinUSB (using libusb library) ^(5a) or regular driver (Gen 3, using Cypress or TE SW libraries)
firmware source code	Hardware Layer only ⁽⁶⁾	Source code available here	Source code available here	Custom firmware ⁽⁷⁾
firmware binary files	Hardware Layer only ⁽⁶⁾	Binary files available here	Binary files available here	Custom firmware ⁽⁷⁾

recovery tool (Windows OS)	Cypress USB Console, Cypress USB Control Center, C# OpenFutNet (recommended) ^(5b) , fx2loader (see also here)	not applicable	not applicable	not applicable
update tool (Windows OS)	not applicable		Cypress USB Console, Cypress USB Control Center , Python Open_FUT (Gen 3), C# OpenFutNet (recommended) ^(5b)	nothing or (only if custom firmware is compatible with TE API Commands (FW APIs)): Cypress USB Console, Cypress USB Control Center , Python Open_FUT (Gen 3), C# OpenFutNet (recommended) ^(5b)

		Python Open_FUT (Gen2) ^(5b)		
recovery tool (Linux OS)	fx2loader (see also here) or fxload (see also here)	not applicable	not applicable	not applicable
update tool (Linux OS)	not applicable	Linux_FUT	Linux_FUT	Linux_FUT (only if custom firmware is compatible with TE API Commands (FW APIs))

Firmware loading and/or update comparison table.

- (1) The TE USB FX2 module (in recovery boot) is served by the [Cypress USB Generic Driver](#); this driver allows Cypress firmware update tools (Cypress Console, Cypress Control Center) and OpenFutNet to work with the Cypress EZ-USB FX2 microcontroller on the module.
- (2) The EEPROM switch connect (enable)/disconnect (disable) the connection between EEPROM and FX2 microcontroller.
- (3) EEPROM connection disabled when TE USB FX2 module turned on, then EEPROM connection enabled when TE USB FX2 module is inserted in the USB port of the computer (aka plugged in, aka attached)
- (4) EEPROM connection enabled when TE USB FX2 module turned on, then EEPROM connection enabled when TE USB FX2 module is inserted in the USB port of the computer (aka plugged in, aka attached)
- (5a) WinUSB driver and Zadig should be used only if [libusb](#) or [libusbx](#) are the libraries used by the user under Windows OS. See WinUSB driver installation ([libusb](#) and [libusbx](#)). See also [Zadig](#).
- (5b) If [libusb](#) or [libusbx](#) are used (with WinUSB driver and/or Zadig), the software tools [Cypress USB Console](#), [Cypress USB Control Center](#), [Python Open_FUT \(Gen2\)](#), [Python Open_FUT \(Gen 3\)](#) and [C# OpenFutNet](#) could not be directly used. The code of these tools should be rewritten using [libusb](#) or [libusbx](#) instead of Cypress libraries (CyAPI.lib and CyUSB.dll) and the derived Trenez Electronic libraries ([TE_USB_FX2_CyAPI.dll](#) and [TE_USB_FX2_CyUSB.dll](#)). See [Linux_FUT](#) as a starting point for a new software tool using [libusb](#) or [libusbx](#) under Windows OS.



These libraries are not yet officially supported under Windows, but they are already used with TE USB FX2 modules under Linux (see the community contribution [Linux_FUT](#)). Zadig is not yet officially supported under Windows.

- (6) To default, the FX2 hardware (i.e with EEPROM isolated => hardware's first stage loader) enumerates the USB FX2 microcontroller chip as VID=0x04B4 (Cypress) and PID=0x8613 (FX2), and provides support for loading firmware into RAM. To default, using the FX2 hardware (i.e with EEPROM isolated => hardware's first stage loader) it is possible to write a new firmware in RAM but NOT in EEPROM. The USB FX2 microcontroller hardware's first stage loader (supporting the 0xA0 vendor request) can't write into external memory (EEPROM for example). Configurations that put firmware into external memory thus need a second stage loader in RAM (normally Vend_Ax.hex).
- (7) See section "[Custom FX2 microcontroller's firmware](#)".



EEPROM connection shall always be enabled during EEPROM programming.



Using [Cypress firmware update tools](#) with generation **2** modules does require a recovery boot, in order to force enumeration as [Cypress USB Generic Driver](#).

On the other side, using [Cypress firmware update tools](#) with generation **3** modules may not require a recovery boot (if a Generation 3 [reference firmware](#) is running on USB FX2 microcontroller), but only an update boot. This is possible because the original [Cypress USB Generic Driver](#) (in case of a recovery boot) and the [Trenz Electronic USB FX2 device driver](#) (in case of a regular boot) are both Cypress driver. [Trenz Electronic USB FX2 driver](#) derives from the original [Cypress USB Generic Driver](#).