

Custom firmware

It is possible to write and download a custom firmware for the TE USB FX2 module.



If the custom firmware is malfunctioning/broken, the FX2 microcontroller (after loading) will stall and the [USB connection \(with the host computer\)](#) will be unresponsive.



If the user is unsure about the suitability of a new firmware (wherever it is the source), it's a good idea to load it into RAM first to make sure it is not totally malfunctioning/broken.

In any case, even an EEPROM written with a malfunctioning/broken firmware could be easily corrected with an implicit two step [recovery tool \(TE USB FX2 Firmware Recovery\)](#) or a explicit two step recovery boot.

Firmware modification dependencies

Host computer's SW will be	To do so, the developer should create a custom FX2 microcontroller's firmware	To do so, the developer should modify the FPGA's MicroBlaze software
able to use custom USB FX2 API Commands and reference USB FX2 API Commands	compatible with the reference firmware;	no change of FPGA's MicroBlaze software should be necessary
able to use SPI Flash Commands ,	that keep a compatible FLASH_WRITE_COMMAND command in FW;	no change of FPGA's MicroBlaze software should be necessary
able to use custom MB Commands	no change of FX2 microcontroller's firmware should be necessary;	modify MicroBlaze API Commands in the <code>i2c_slave_int_handler()</code> function in interrupt.c running on FPGA's MicroBlaze (using the demo project as starting point);
use the interrupts associated with MB Commands in a different way (aka avoid polling for example)	create a custom FX2 microcontroller's firmware that modify GET_INTERRUPT command, SET_INTERRUPT command,	modify the <code>i2c_slave_int_handler()</code> function in interrupt.c running on FPGA's MicroBlaze and (maybe) modify the VHDL XPS_I2C_SLAVE custom IP block
use Slave Parallel (SelectMAP) Mode and/or Slave Serial Mode	that load the configuration data from a source (SPI Flash, USB connection or B2B connection) and write the retrieved configuration data in the FPGA.	no change of FPGA's MicroBlaze software should be necessary, but some HW modifications (desoldering/resoldering) are necessary for TE0630 and TE0300 module

Firmware modification dependencies

The first two point will (probably) require modifications only to USB FX2 microcontroller's firmware.

The third point will (probably) requires modifications only to FPGA's MicroBlaze software.

The fourth point will requires (minor) modifications to USB FX2 microcontroller's firmware and (not so minor) modifications to FPGA's MicroBlaze software and (maybe) VHDL of XPS_I2C_SLAVE custom IP block.



If the user modify the [USB FX2 API Commands](#) in a not compatible way, the legacy [SW API Layer's](#) functions (or a part of them) will be unavailable.