

# Logic Architecture Layer (Generation 2 = Generation 3): FPGA image

The Xilinx FPGA itself on the Trenz Electronic USB FX2 modules is blank when powered off. To define an FPGA functionality, a Logic Architecture Layer (also called FPGA image or FPGA bitstream) should be defined and loaded into the device at power-on.

The Logic Architecture Layer is defined in a .mcs or .bit file (.bin file is also possible). The Logic Architecture Layer is defined (and .bit/.bin/.mcs are created) using the Xilinx software applications:

- XPS, if MicroBlaze is not used
- EDK (or both), if MicroBlaze is used.

The Logic Architecture Layer could be loaded from SPI Flash (it happens automatically at power on or on general reset) or from [JTAG](#) and/or [USB](#) connections. See [here](#) for a general introduction.



A TE USB FX2 module comes with SPI Flash preloaded with a reference design ([Reference Architecture Layer](#)).

Those preloaded image is just for getting started demo. For real designs:

- it is expected that these image could be replaced by the customer (custom Logic Architecture Layer);
- it is expected that these image could be used by the customer as starting point for the creation of a derived Logic Architecture Layer compatible with Reference Architecture Layer.

The use of preloaded [Reference Architecture Layer](#) is a user choice.



If the customer runs:

1. custom Logic Architecture Layer (on the FPGA, with or without MicroBlaze) and
2. the reference firmware (or derived compatible firmware) on FX2 USB microcontroller

the [USB FX2 API Commands](#) and [SPI Flash Commands](#) are available through C++/.NET [SW API Layer](#).

[MicroBlaze API Commands \(MB Commands\)](#) are available only if Reference Architecture Layer (or derived compatible Logic Architecture Layer). [GET\\_INTERRUPT command](#) and [SET\\_INTERRUPT command](#) (two [USB FX2 API Commands](#)) could be used without Reference Architecture Layer (or derived compatible Logic Architecture Layer), but they are usually usefull only with Reference Architecture Layer.

See [TE API Commands \(FW APIs\) dependencies to reference custom IP blocks](#).