FTDI Programmer User Manual

Kunal Chakraborty
kunalc@cdot.in

January 21, 2019

0.1 Introduction

FTDI-programmer is an application written in pure python to access/program on board devices through FTDI devices using jtag/spi/i2c/gpio interfaces. JTAG programming is done through svf file. It can be used in place of any external emulator. The application is built on PyFtdi driver.

The source code is compatible with both Windows and Linux system. However this manual is written from a windows user's point of view.

This is a Beta version and hence the application has couple of limitations as listed in section 0.3

0.2 Features

- On board device programming through JTAG interface. It needs svf file for JTAG programming.
- TI UCD device programming through I2C interface. It needs SMBUS csv file programming. This file can be generated by Fusion tool.
- SPI Flash programming.(to be implemented)
- On board device access thhrough spi/i2c/gpio interfaces. (to be implemented)

0.3 Limitation

1. I2C mode
   (a) FTDI devices does not support multimaster and clock stretching. Hence this application will not work with slave devices which requires clock stretching. In that case application can be run on lower frequency so that target device may get enough time to respond. However there is a workaround given by FTDI by connecting I2C SCL to a separate gpio of FTDI device so that FTDI device can read the clock line. This feature is not supported in present version FTDI-programmer.

    (b) Highest I2C slave address supported is 0x78. It is the limitation from pyftdi driver. No workaround is provided.

2. JTAG mode

    (a) Target device must be the only device in JTAG chain because SVF parser of FTDI-programmer does not support header and trailer addition. This is kept as Future Development.

    (b) SVF file verification takes longer than expected. This is because pyftdi driver takes long time to send data to application after reading from usb. This is kept as Future Development.

    (c) Only Max-V CPLD programming is supported. If any other device is detected, FTDI-programmer will flag a warning and continue with programming. However the integrity of the programming is not validated.

3. Others

    (a) MS Windows detects every channel of FTDI device as a separate usb device with same VID and PID. Hence FTDI-programmer can access only single channel of FTDI device. So user must enable a single lib-usb driver as per FTDI channel number as discussed in section **??**

0.4 Installation

All the required python packages are compiled and supplied with FTDI-programmer. However it needs a low level backend driver libusb to talk to FTDI device. You must have admin privilege to carry out below steps. Step by step installation guide is given below.

1. **libusb installation** An easy way to install libusb backend on windows is Zadig. Download the latest version of Zadig Run zadig.exe and you should get a dialog as shown below Go to Options – List All Devices. Then select FTDI-channel0 from drop down option. Next select libusb-win32 as driver. Finally, click install driver. Follow the same step to install driver on FTDI-channel1

2. **Get FTDI-programmer** Download a fresh copy of FTDI-programmer from 'https://github.com/kunalcdot/FTDI-programmer'. The application is under /bin folder. Run FTDI-programmer.exe to start the application.

0.5 Programming Guide

0.5.1 JTAG Programming

pre requisite – svf file zadig channel selection

0.5.2 JTAG Programming

0.6 Future Development

1. All programming file parser to be updated with 'regular expression' module. It should support all svf standard command.
2. JTAG read time needs to be reduced.
3. SPI Flash Programming option to be developed
4. An utility needs to be provided to read/write on board devices through I2C/SPI/GPIO etc

0.7 Troubleshooting

Ensure only one FTDI device is connected and single ftdi channel is enabled. Refer to section **??** for more details.