

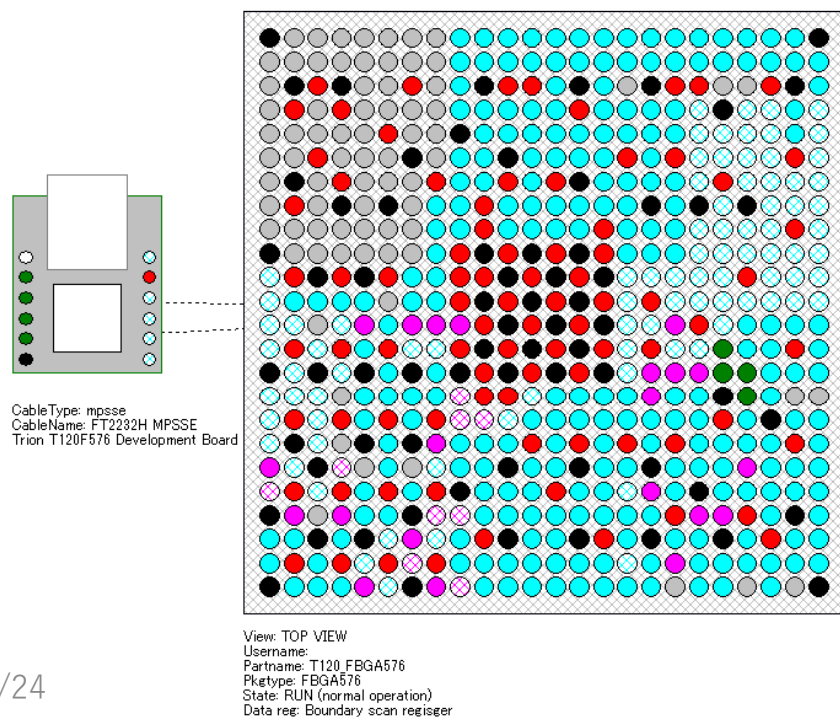
Effinixに書き込むための JTAGプロトコルの話

2023/6/24

なひたふ、@nahitafu

自己紹介

- 特殊電子回路株式会社
- MITOUJTAG(みとうジェイタグ)
というJTAGツールを作っています



なひたふ @nahitafu

目的

- Efinityの書き込み環境があまりにもひどいので、MITOUJTAGからTrionの書き込みができるようにしたい
- そのためにJTAG書き込みシーケンスを解読する

購入したボード T20とT120



※シリアル番号が外箱の裏に書いてあるので、外箱は捨てないようにしましょう

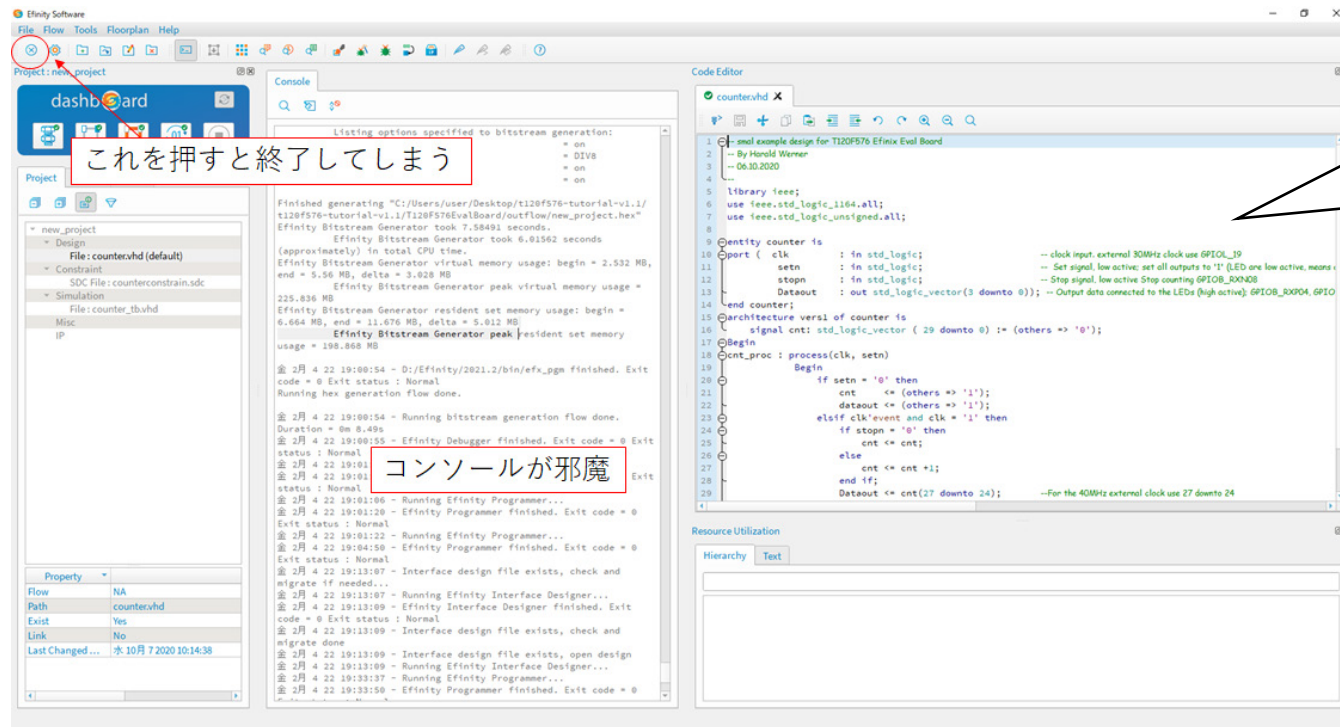
(C)2023 なひたふ

書き込みのためのJTAGアダプタは？

- ボード上にFT2232がのっていてMPSSEで書き込めるらしい
- MPSSEはFTDIの提供するUSB-JTAG

Efinity(開発環境)を動かしてみる

なんじゃこりゃ!?



作業エリアが
せまーい

※左上には「新規作成」とか「開く」があるはずなのに、いきなり閉じるとは、使わせたくないのか？

ピン定義がやりにくい・・・

- 正直、やってくれるか！という感じ

The screenshot shows the Efinity Interface Designer software. The main window is titled "Resource Assigner" and displays a table of GPIO instances. The table has columns for Instance, Package Pin, Resource, I/O Bank, Alt Conn, Features, Clock Region, and Pad. The instances listed are Dataout[0] through Dataout[4], and a clk instance. The Resource column shows GPIO_104, GPIO_105, GPIO_117, GPIO_118, and GPIO_153. The I/O Bank column shows 3A_3B_3C, 3A_3B_3C, 3A_3B_3C, 3A_3B_3C, and 3D_3E. The Alt Conn column shows None, None, GCTRL, GCTRL, and None. The Features column shows DDIO, DDIO, DDIO, DDIO, and DDIO. The Clock Region column shows R2, R2, R1, R1, and R0. The Pad column shows GPIO_104_CD125, GPIO_105_CD124, GPIO_117_CTRL14, GPIO_118_CTRL13, and GPIO_153.

Instance	Package Pin	Resource	I/O Bank	Alt Conn	Features	Clock Region	Pad
Dataout[0]	D14	GPIO_104	3A_3B_3C	None	DDIO	R2	GPIO_104_CD125
Dataout[1]	E13	GPIO_105	GPIO_104	None	DDIO	R2	GPIO_105_CD124
Dataout[2]	G13	GPIO_117	3A_3B_3C	GCTRL	DDIO	R1	GPIO_117_CTRL14
Dataout[3]	F14	GPIO_118	3A_3B_3C	GCTRL	DDIO	R1	GPIO_118_CTRL13
Dataout[4]	N14	GPIO_153	3D_3E	None	DDIO	R0	GPIO_153
clk	E8	GPIO_75	1D_1E	PLL_CLKIN	None	T	GPIO_75_PLLIN1

Below the table is the Design Explorer window, which shows a tree view of the design components. The Design Summary window is also visible, showing properties such as Name, Device, Package, Timing Model, Location, and Version.

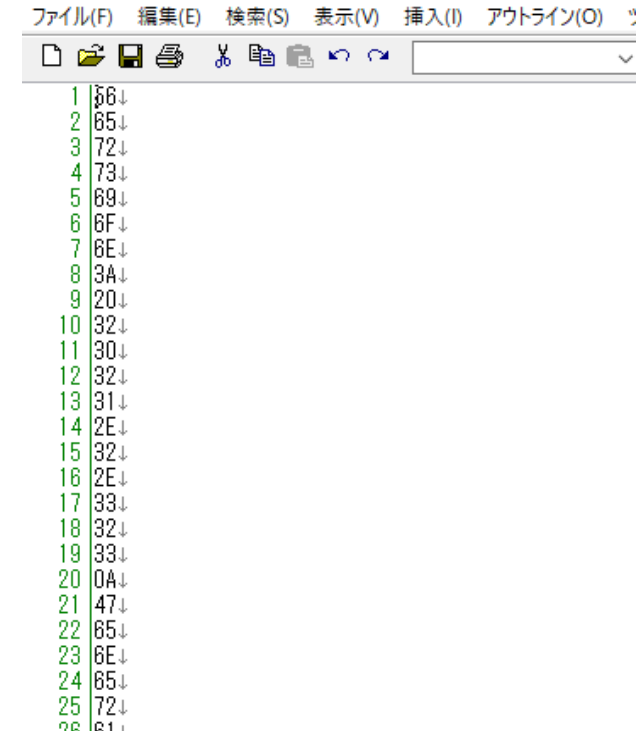
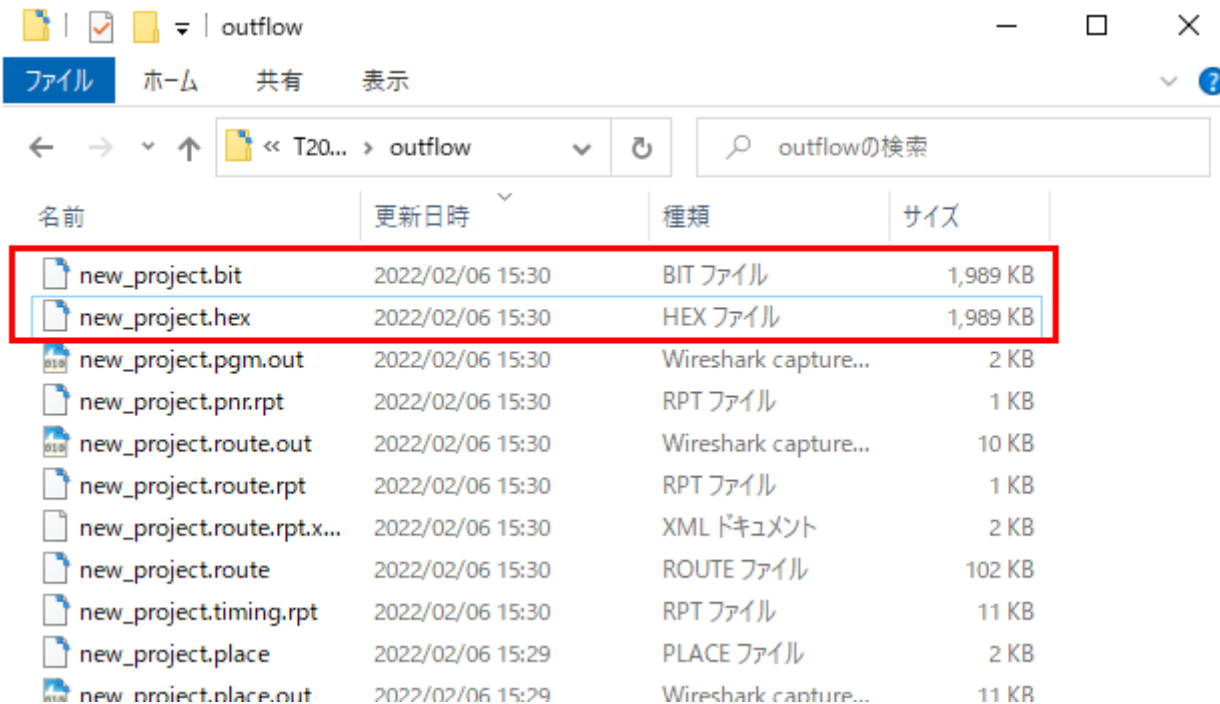
```
<efxpt:gpio_info device_def="T20F256">
<efxpt:gpio name="Dataout[0]" gpio_def="GPIO_104"
mode="output" bus_name="Dataout" is_lvds_gpio="false"
io_standard="3.3 V LVTTTL / LVCMOS">
<efxpt:output_config name="Dataout[0]" name_ddio_lo=""
register_option="none" clock_name=""
is_clock_inverted="false" is_slew_rate="false"
tied_option="none" ddio_type="none" drive_strength="3"/>
</efxpt:gpio>
```

ピン定義ファイルはXMLだ！

この表に入力していくのはしんどい

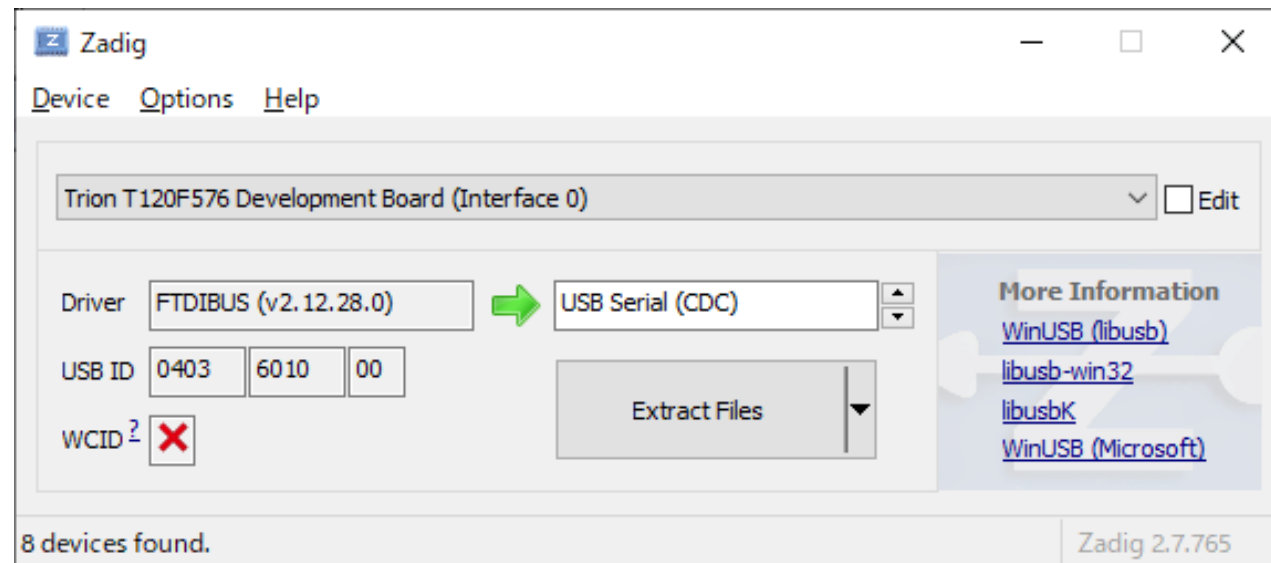
Efinityが出力するファイル

- bitとhexがあるが、中身は同じ
- 16進数がASCII文字で記述され、2桁ごとに改行



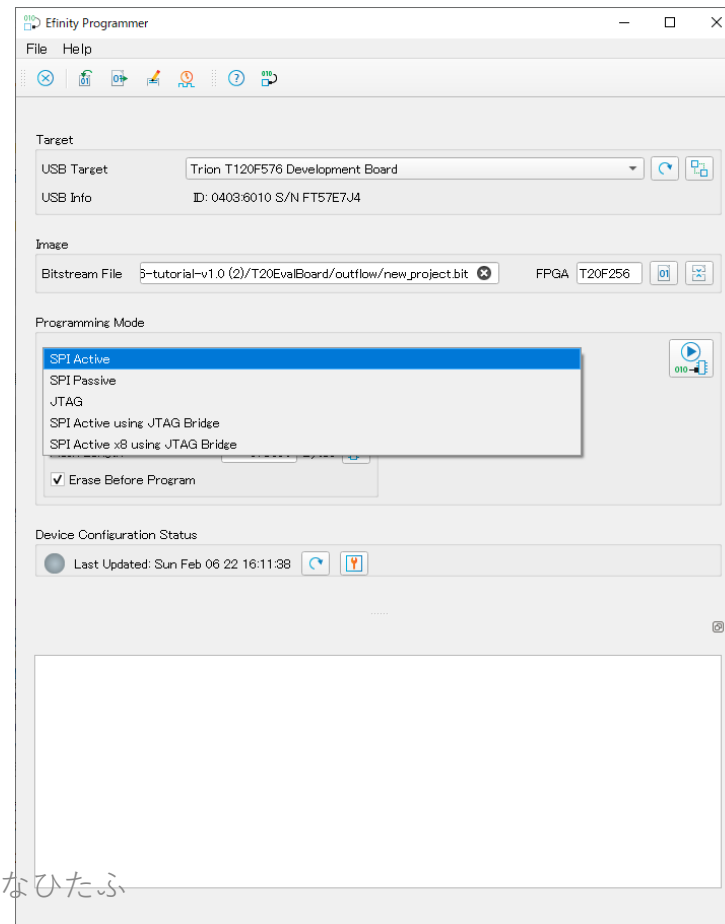
Efinityの標準書き込み方法

- 「Zadig」とかいう危険ツールを使う
- FTDIのドライバをlibusbで置き換えてしまうので、他のアプリでFTDIが使えなくなる！
- マウスとかキーボードを置き換えると詰む



ツールがすぐにクラッシュする

- JTAGモードを選択するだけでクラッシュする
- SPIに書き込むしかない！？



ツールがクラッシュする原因

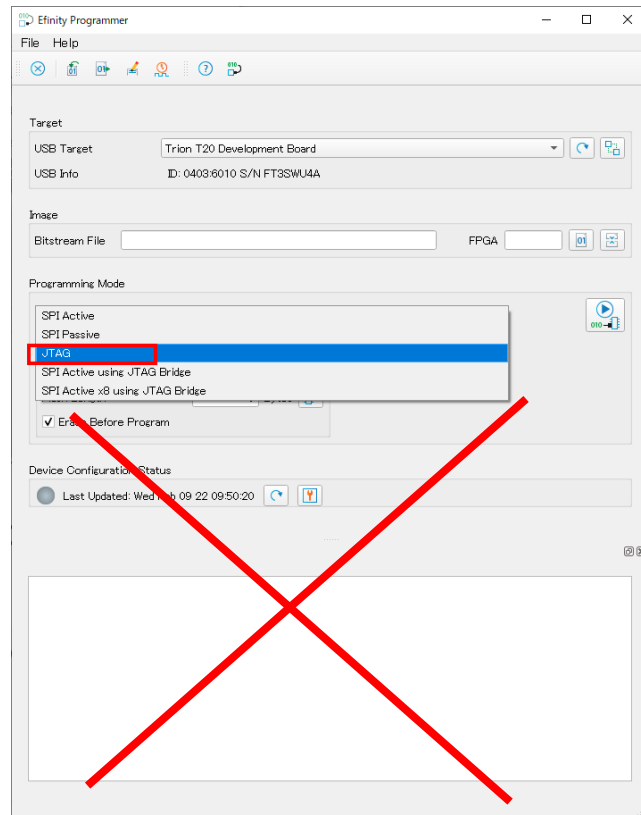
- Pythonで書かれているのでソースコードを読める
- D:\Efinity\2021.2\pgm\bin\efx_pgm\main_window.py の1021行目
- ここでidcodeに正しい値が入らない。
- チェーンの定義ファイルをXMLで書いておけばクラッシュはしなくなる・・・が書き込み時にクラッシュするので結局書けない。

```
try:  
    idcode = self.read_idcode(None, 1)  
    self.cb_chip_num.addItem('0x%08x' % int(idcode))  
except ValueError as exc:  
    self.logger.debug(str(exc))  
    self.cb_chip_num.addItem('Default')
```

クラッシュしているコード

ツール間での書き込み成否

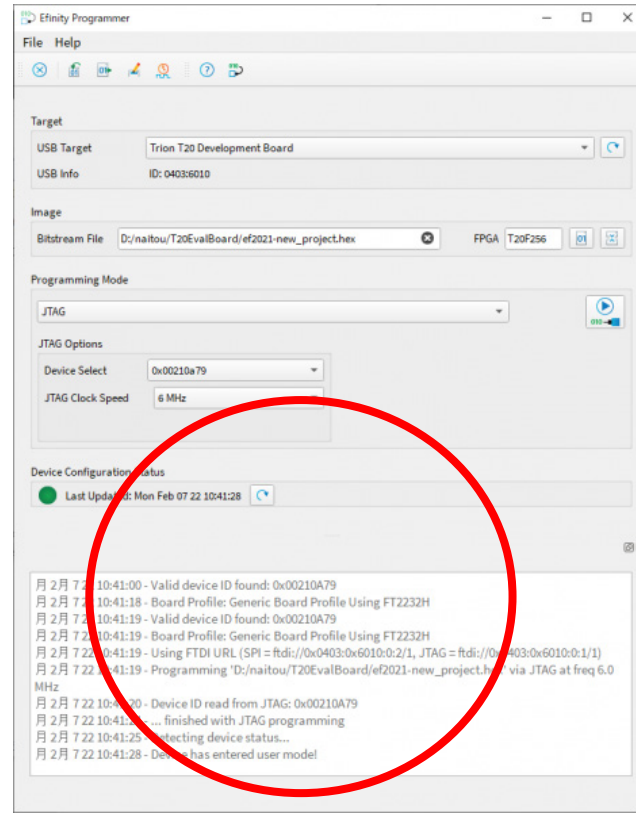
最新 v2021.2.323



クラッシュするのでダメ

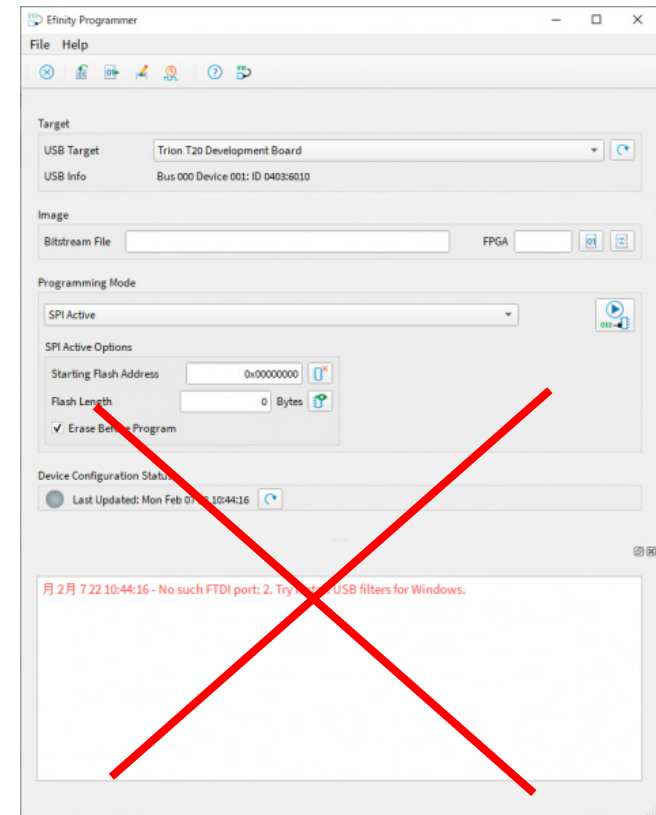
1つ前 v2021.1.165

2つ前 v2020.2.299



JTAG書き込みできる

3つ前 v2020.1.140



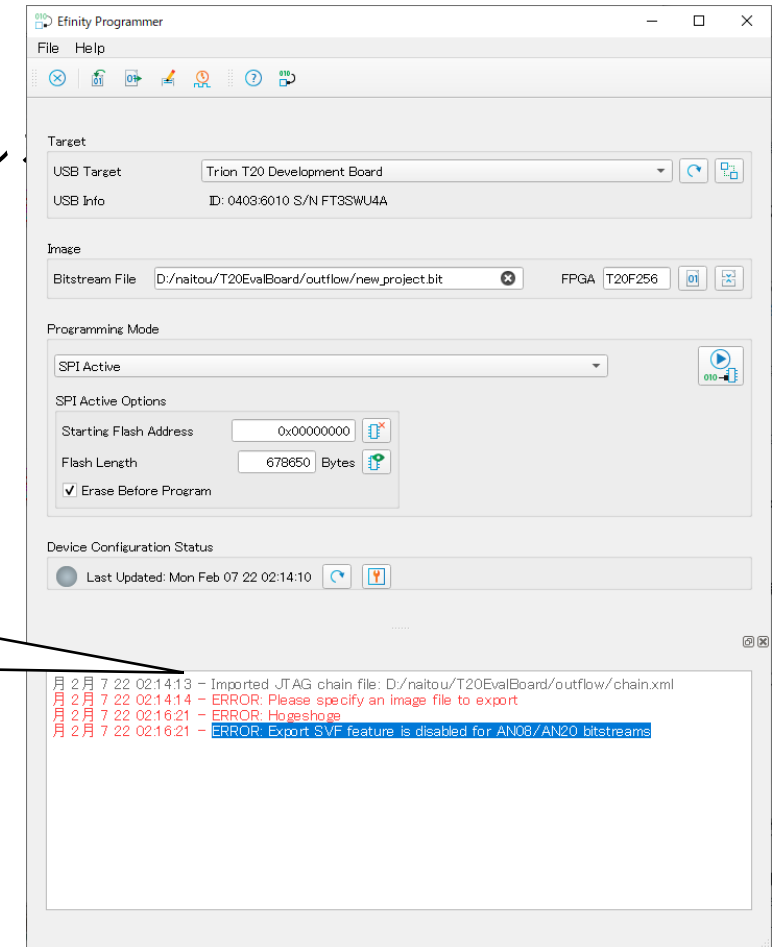
FTDIを認識しない

書き込みツールを解析しよう

- まずはSVFを吐かせたい
- T8~T20のデバイスに対してはSVFファイルさせないようになっている。ケチ！
- GUIツールでは無理っぽい

ERROR: Export SVF feature is disabled for AN08/AN20 bitstreams

※SVFはローレベルのJTAG手順が書かれたファイルで、これをみればプロトコルは一目瞭然。
※いろいろなBitファイルからSVFファイル吐かせてみて、書き込み手順を推測するのがリバエンの常套手段。



CUIツールなら・・・

- D:\Efinity\2021.2\pgm\bin\efx_pgm に移動し、
export_bitstream.pyという怪しげなツールを起動してみる

```
python38 export_bitstream.py hex_to_svf  
D:\naitou\T20EvalBoard\outflow\new_project.bit  
D:\naitou\T20EvalBoard\outflow\test.svf
```

※EfinityのツールはPythonで書かれていて、ソースコードが読めるし、改造もできる。
pgm\binにはたくさんのコードがあってまさに宝庫。

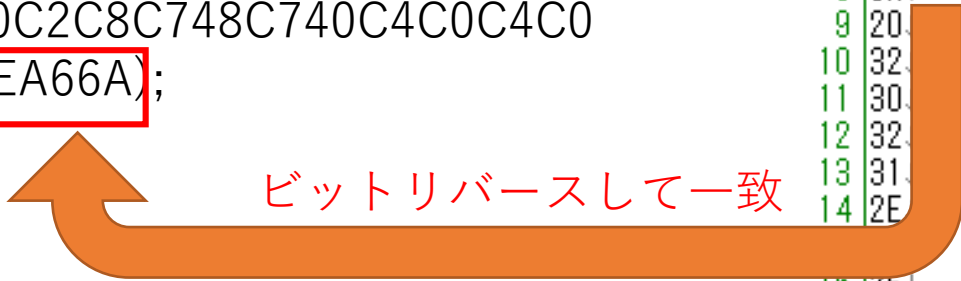
HEX(bit)ファイルとの対応

SDR 1000 TDI

```
(5C162E2696EA506CAC4C620C4C2A045C  
A6C6966EA6225076F6964E2A045C9E3696  
B6866250264E86F64236866EA20C4C2A3A  
AEF62E9686763A5C22045C2EC6A656F64E  
0A50504C4C0C4C048CAC5C8C8C5C0C8C  
04EC0C0446A6620476F6B2045C26A62E86  
4EA676A6E2500C2C8C748C740C4C0C4C0  
45C76F696CE4EA66A);
```



66↓
65↓
72↓
73↓
69↓
6F↓
6E↓



ビットリバーズして一致

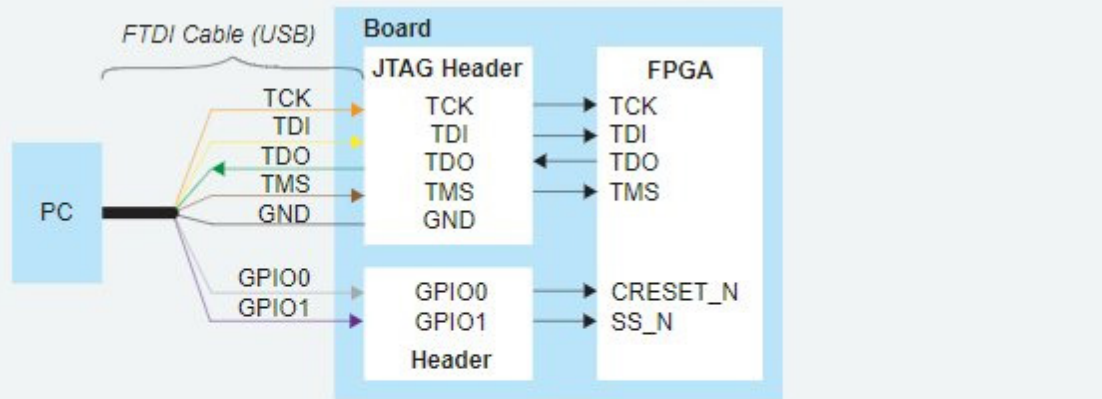
JTAG書き込みシーケンスの核心は？

- IRレジスタに4を書き込んで、
- **ビットリバーズしたHEXファイルを1000バイトごとに区切ってDRに流せばよい？**
- 最後に1000ビットくらいの0を送る
- IRレジスタに7を書き込む。

T120では書き込みできて、
T20では書き込みできない・・・なぜ？

T20ではCRESET_NとSS_Nが必要

Figure 2. Connect JTAG Cable to JTAG Pins plus CRESET_N



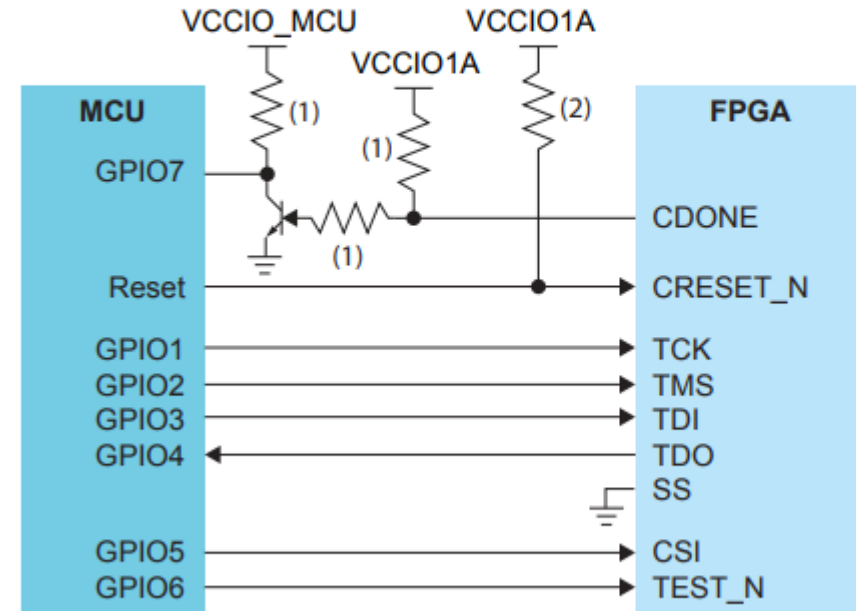
Note: This figure uses the CRESET_N and SS_N pins in addition to setup is only needed for normal JTAG programming. You can use for programming with SPI Active using JTAG Bridge mode or for

※データシートによればSSとCRESETは必要
そうである

T4 (All), T8 (All), T13 (All) and T20 (W80, F169, and F256)

These FPGAs require you to connect the SS pin to ground and have some additional configuration sequence requirements.

Figure 4: Single FPGA

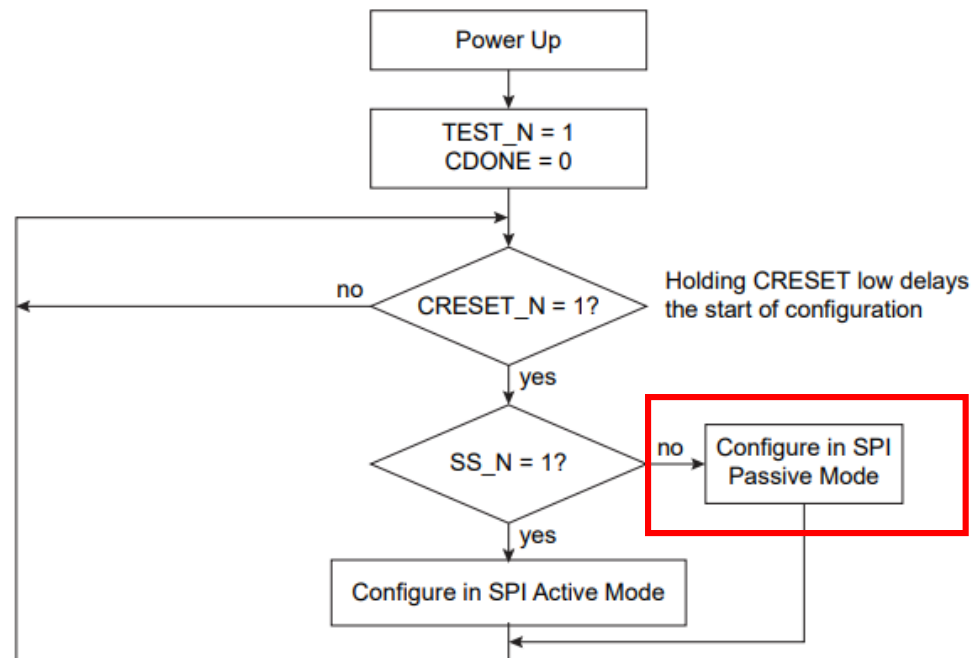


1. 100 kΩ resistor
2. 10 kΩ resistor

Passive Modeに入れる理由

- SPIコンフィギュレーションが走らないようにPassive Modeに入れておいた状態で、JTAGの信号を流しこむ

Figure 19: Configuration Flow Diagram

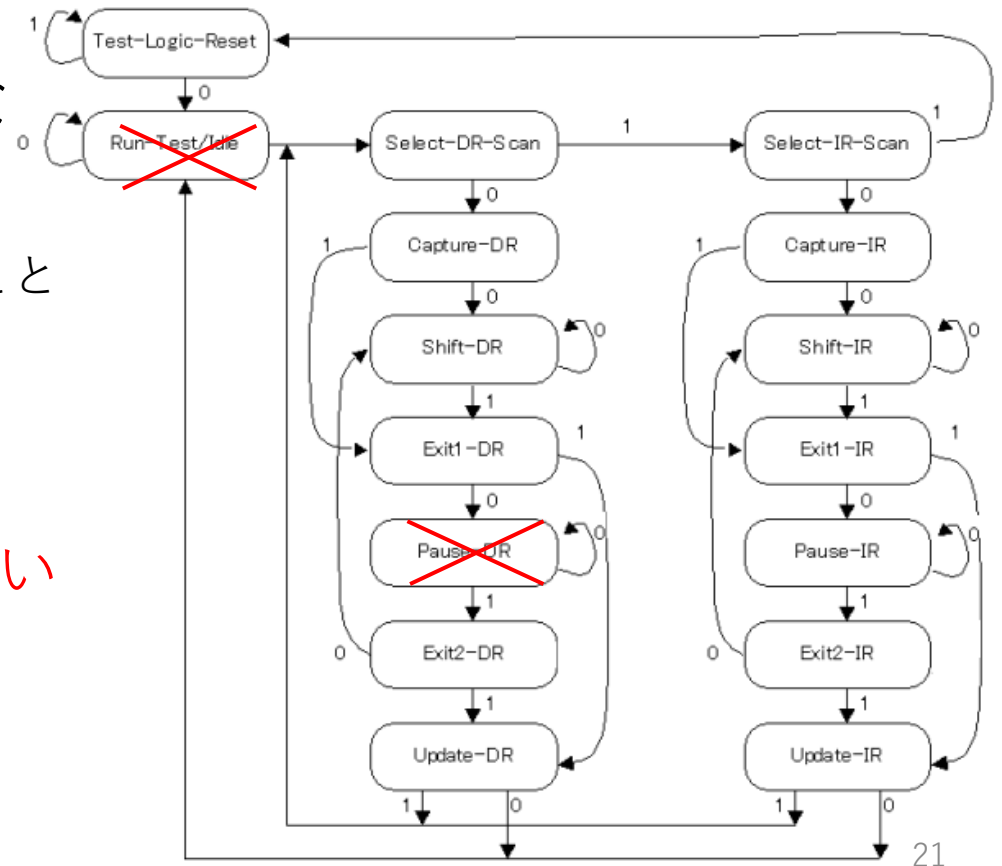


※Passive Modeというのは、SPI端子を通じてクロックとMISOが与えられるのを待っている状態。ActiveModeというのはFPGAがクロックを出してMISOからデータが入ってくるのを待っている状態。つまり、PassiveModeに入れて、無理やりコンフィギュレーションを止めている

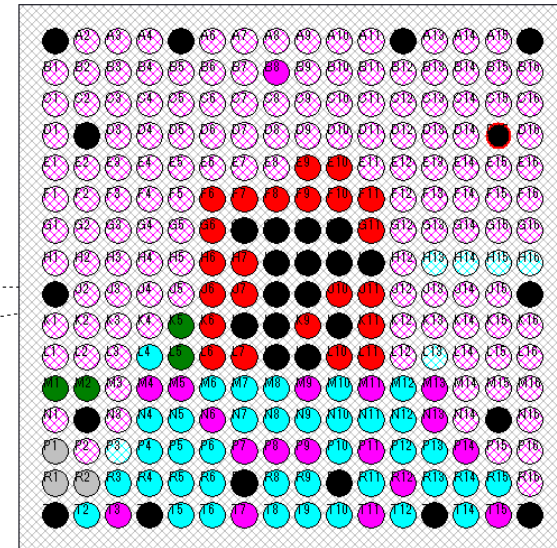
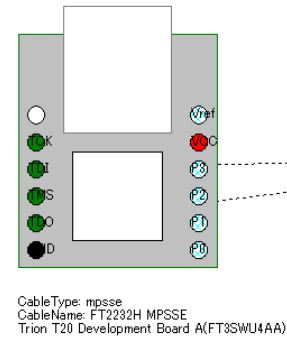
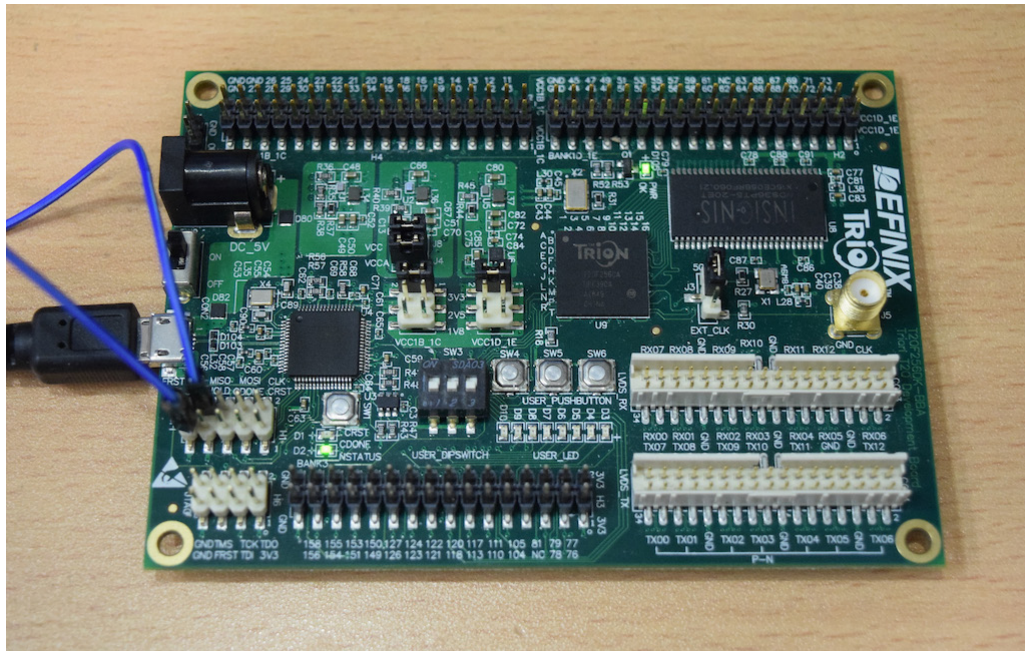
T20に書き込めなかった原因

- T20ではデータ送信中にPAUSE_DRやIDLEに遷移してはいけない
- SVFファイルに書かれていたような分割して送信するスキームは不可
※Shift-DRから出ずにすべてのビットを送れということ

T120ではPAUSEやIDLEに行っても問題ない



書き込みデモ



View: TOP VIEW
Username:
Partname: T20, FBGA256
Pkgtype: FBGA256
State: UNKNWON
Data reg: Other scan register

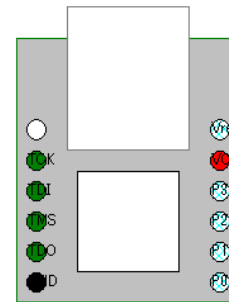
※青い線はSS_NをGNDにつないでいます
リセットボタンを手動で押してから書き込みます
JTAGから現在のコンフィグを消すことができません

まとめ

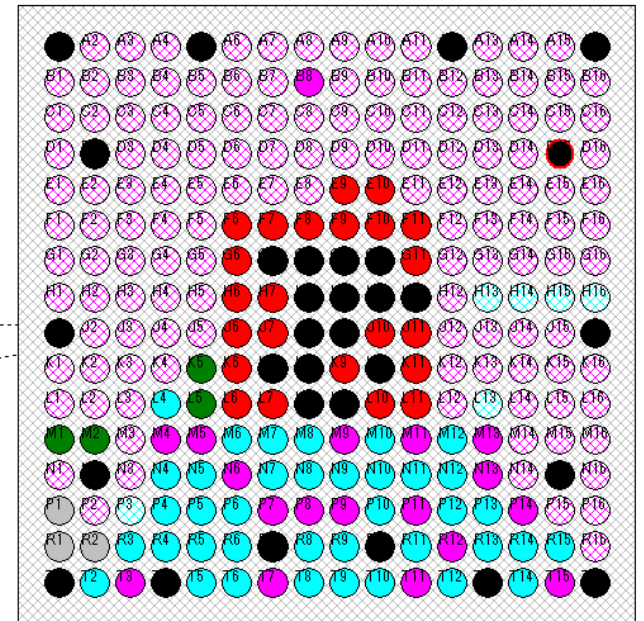
- T20とT120では、おそらくアーキテクチャが全く違う。もしかすると、T8～T20の小規模系デバイスはみんな同じ？
- Efinixツールが生成したSVFファイルが間違っていた。T20では分割してはいけない。だからSVF生成ができないのだろう。
- T20ではCRESET_NとSS_Nが必要。コンフィグモードピンが無いので、SPIコンフィグを止めて（外からSPIコンフィグデータが与えられるのを待たせて）、JTAGで書き込むようになっている
- もうZadigでドライバの置き換えなくてもよい

Efinixの開発環境が楽になったよ

- MITOUJTAGで、T20やT120に快適に書き込みができるようになった。
- もうEfinity Programmerで書き込む必要はありません。
- XILINXが手に入るようになったので、Efinixへの興味は急速に・・・



CableType: mpsse
CableName: FT232RL MPSSE
Trion T20 Development Board A(FT3SWU4AA)



View: TOP VIEW
Username:
Partname: T20_FBG256
Pkgtype: FBGA256
State: UNKNWON
Data reg: Other scan register