

TE0950

Versal AI Edge で遊びたいなと思っても、あまり手軽なボードがない... と思っていた時に、Trenz のボード TE950-03-EGBE21C の在庫が Digkey にあるよ、と教えてもらって購入。

リファレンス

- ・ [TE0950 Test Board](#)
- ・ [TE0950 Basic Linux Example](#)

まずは動かしてみる

[TE0950 Basic Linux Example](#) から、

TE0950-test_board-vivado_2023.2-build_4_20240531092954.zip をダウンロード。

展開すると、test_board/prebuilt の下にビルド済みファイルがある。

file_location.txt に従って、

FAT32 でフォーマットされている microSD カードに TE950-03-EGBE21C 用のファイルをコピー。

```
cp boot_images/23_1lse_8gb/u-boot/* /media/miyo/9016-4EF8
cp -r os/petalinux/8GB/* /media/miyo/9016-4EF8
cp -r hardware/23_1lse_8gb/* /media/miyo/9016-4EF8
```

で、ボードの外側に近い方のマイクロ USB 端子を使ってホスト PC と接続して電源投入。無事に Linux が起動してきた。

```
root@Trenz: # uname -a
Linux Trenz 6.1.30-xilinx-v2023.2 #1 SMP Fri Sep 22 10:41:01 UTC 2023 aarch64 GNU/Linux
```

Ethernet ポートに LAN ケーブルをつなぐと DHCP で IP を取得。

ユーザ名 root / パスワード root で、ssh でログインできた。

自分でビルドしてみる

環境は

- ・ Vivado 2023.2.1 - /tools/Xilinx/Vivado/2023.2
 - ・ Vivado 2023.2.1 の場合は有償ライセンスが必要
 - ・ 合成以降を Vivado 2024.2.2 で置換することでライセンスなしでも構築可能
- ・ Petalinux 2023.2 - \$HOME/petalinux/2023.2

というもの。

まずは、[TE0950 Basic Linux Example](#) から、

TE0950-test_board_noprebuilt-vivado_2023.2-build_4_20240531092954.zip をダウンロード

展開すると、test_board ができる。

test_board/_readme.txt に Web ページ以上に細かく手順が書いてあるので読む。

私の場合、まず準備として、

```
chmod 755 _create_linux_setup.sh
```

としてスクリプトに実行権限を付与し , console/base_sh/design_basic_settings.sh を開いて

```
export ALTERNATIVE_PETALINUX_XSETTINGS= /petalinux/2023.2/settings.sh
```

と変更しておく .

で ,

```
./_create_linux_setup.sh
```

実行

```
-----TE Reference Design-----  
-----  
-- (d) Go to Documentation (Web Documentation)  
-- (x) Exit Batch (nothing is done!)  
-- (0) Module selection guide, project creation...  
-- (1) Create minimum setup of CMD-Files and exit Batch  
-- (2) Create maximum setup of CMD-Files and exit Batch  
-- (3) (internal only) Dev  
-- (g) Install Board Files from Xilinx Board Store (beta)  
-- (a) Start design with unsupported Vivado Version (beta)  
-----  
Select (ex.: '0' for module selection guide):
```

とかメニューがでてくるので ,

0

を選択 . 環境チェックが走ったあと ,

```
-----  
Select Module will be done in 2 steps:  
-----  
Step 1: (select column filter):  
-Change module list size (for small monitors only), press: 'full' or 'small'  
-Display current module list, press: 'L' or 'l'  
-Restore whole module list, press: 'R' or 'r'  
-Reduce List by ID, press: 'ID' or 'id' or insert ID columns value directly(filter step is bypassed  
and id number is used)  
-Reduce List by Article Number, press: 'AN' or 'an'  
-Reduce List by SoC/FPGA, press: 'FPGA' or 'fpga'  
-Reduce List by PCB REV, press: 'PCB' or 'pcb'  
-Reduce List by DDR, press: 'DDR' or 'ddr'  
-Reduce List by Flash, press: 'FLASH' or 'flash'  
-Reduce List by EMMC, press: 'EMMC' or 'emmc'  
-Reduce List by Others, press: 'OTHERS' or 'others'  
-Reduce List by Notes, press: 'NOTES' or 'notes'  
-Exit without selection, press: 'Q' or 'q'  
-----  
Please Enter Option:
```

と , 続いて選択肢のテーブルが表示される . まず ,

id

と入力すると ,

```
Last Input:<id>
Note: Input will be compared with list elements, wildcard * possible. Ex.*1*
Go back to top menu with 'q' or 'Q'
Step 2: Insert ID:
```

とでてくるので、利用するボードにあわせて、

```
4
```

を選択。

```
You like to start with this device? y/N
```

と問われるので、

```
y
```

と入力。

```
What would you like to do?
- Create and open delivery binary folder, press 0
- Create vivado project, press 1
- Both, press 2
```

とかでてくる。

```
1
```

を入力。Vivado が走ってプロジェクトの作成がはじまる。

しばらくすると、Vivado が GUI モードで起動してくるので、Generate Device Image をクリックして、FW を作る。

プロジェクトファイルは vivado/test_board.xpr にできている。

無事に FW ができたら、File Export Export Hardware... で xsa ファイルを作る。

Include device image にチェックをいれるのを忘れないように。

デフォルトのまますすめると、vivado/vsys_wrapper.xsa ができる。

なお、ビルドには Vivado の有償ライセンスが必要になる。

有償ライセンスがない場合、たとえば、2024.2 では、TE0950 に搭載されている XCVE2302 向けのビルドができるので、

Vivado を一旦終了した後で、Vivado 2024.2 で vivado/test_board.xpr を開いて、

プロジェクトのアップグレードと IP コアのアップデートをおこなうと無事にビルドができる。

次はPetalinux で u-boot や Linux など一式をビルドする。test_board/petalinux を作って作業することに。

(ここでは、Vivado 2024.2 を使って xsa を作った場合でも 2023.2 を使った)

まずは、test_board/ の下にいる状態で、

```
mkdir -p petalinux
cd petalinux
```

として作業ディレクトリを作成。
プロジェクトを

```
petalinux-create --type project --template versal --name vsys
cd vsys
```

で作成して、作成したプロジェクトのディレクトリに移動。
次に、作成した xsa ファイルを読み込む。

```
petalinux-config --get-hw-description ../../vivado/vsys_wrapper.xsa
```

とりあえず、config は、そのまま抜ける。
あらかじめ作ってある設定ファイルがあるので、雑に上書きしてみる。

```
pushd ../../os/petalinux/
tar cvf - project-spec | tar xvf - -C ../../petalinux/vsys/
popd
```

このままだと、xsa ファイルへの参照がおかしいので、再度読み込んでおく。

```
petalinux-config --get-hw-description ../../vivado/vsys_wrapper.xsa
```

準備がととのったので、ビルド。

```
petalinux-build
```

ビルドがおったらパッケージを作成して、必要なファイルを集める。

```
petalinux-package --boot --plm --psmfw --u-boot --dtb --force
mkdir -p SD; cp images/linux/{BOOT.BIN,boot.scr,image,rootfs.cpio.gz,u-boot} SD/
```

SD 以下のファイルを microSD カードに書き込む。
作成した microSD カードでブートすると、無事に Linux が起動してきた。