

## PYNQ のビルド

<https://wasa-labo.com/wp/?p=612> がだいぶ古かったのでアップデート

[https://pynq.readthedocs.io/en/latest/pynq\\_sd\\_card.html](https://pynq.readthedocs.io/en/latest/pynq_sd_card.html) が元ネタ

### Vivado のインストール

Vivado 2022.1 をインストールして、2022.1.1 と 2022.1.2 までアップデートしておく。

/tools/Xilinx/Vivado/2022.1 とかにインストールされていると仮定。

Vivado 2019.1 を必要とする PYNQ v2.5 以降は SDx は不要になっている (というか なくなった)。

### VirtualBox のインストール

ドキュメント通り Vagrant 使うのがいいのかもしれないけど、手動でセットアップ。

VirtualBox をインストールして、Ubuntu 20.04.6 環境をセットアップ。

ドキュメントのサポート OS には 20.04.3 までしか記載されてないけど 20.04.6。

なお、好みで server 版を選択。

ホスト OS の /tools を /tools にマウントするなど、開発環境を共有できるようにしておく。

### PYNQ のビルド準備

リポジトリクローンして、セットアップスクリプトを実行。

```
git clone https://github.com/Xilinx/PYNQ.git
cd PYNQ/sdbuild
./scripts/setup_host.sh
```

### 途中で

```
E: Unable to locate package zlib1g:i386
```

### とか言われたら

```
sudo dpkg --add-architecture i386
sudo apt update
```

してやりなおし。他にも途中でエラーがでたらインストールしてやりなおし。

自分の環境では、

```
sudo apt install libncurses-dev
```

が必要だった。

PYNQ の Linux イメージには事前ビルド済みのアーカイブっを利用する。

[pynq.io](https://pynq.io) から [PYNQ rootfs arm v3.0.1](#) と [Prebuilt PYNQ source distribution binary v3.0.1](#) をダウンロード

ドして、  
それぞれ

```
cp jammy.arm.3.0.1.gz ./prebuilt/pynq_rootfs.arm.tar.gz  
cp pynq-3.0.1.tar.gz ./prebuilt/pynq_sdist.tar.gz
```

としてコピーする。  
最後に PetaLinux 2022.1 をインストール。たとえば、

```
chmod 755 petalinux-v2022.1-04191534-installer.run  
./petalinux-v2022.1-04191534-installer.run -d /opt/xilinx/petalinux-v2022.1-04191534
```

とか。  
最後に、Vivado の実行に必要なライブラリで不足してるものをインストール。手元では、

```
sudo apt install libtinfo-dev libtinfo5
```

が必要だった。

ビルド

環境変数を読み込んで

```
source /tools/Xilinx/Vivado/2022.1/settings64.sh  
source /opt/xilinx/petalinux-v2022.1-04191534/settings.sh
```

試しに PYNQ-Z1 だけビルドしてみる。Makefile に

```
ALLBOARDS ?= $(basename $(notdir $(wildcard $(BOARDDIR)/*/*.spec)))  
BOARDS ?= $(ALLBOARDS)
```

とあるので

```
BOARDS=Pynq-Z1 make
```

として Pynq-Z1 向けだけのビルドを実行。  
途中で sudo でパスワード求められるので注意が必要。