# **PYNQ SD Card**

The PYNQ image for the PYNQ-Z1 and PYNQ-Z2 boards are provided precompiled as downloadable SD card images, so you do not need to rerun this flow for these boards unless you want to make changes to the image flow.

This flow can also be used as a starting point to build a PYNQ image for another Zynq board.

The image flow will create the Zynq BOOT.bin, the u-boot bootloader, the Linux Device tree blob, and the Linux kernel.

The source files for the PYNQ image flow build can be found here:

<PYNQ repository>/sdbuild

More details on configuring the root filesystem can be found in the README file in the folder above.

## **Building the Image**

It is recommended to use a Virtual machine to run the image build flow. A clean and recent VM image is recommended. The flow provided has been tested on Ubuntu 16.04.

To build the image follow the steps below:

1. Install the correct version of PetaLinux and optionally Vivado and SDK
2. Install dependencies using the following script

The correct version of Xilinx tools for each PYNQ release is shown below:

| **Release version** | **Xilinx Tool Version** |
| --- | --- |
| v1.4 | 2015.4 |
| v2.0 | 2016.1 |
| v2.1 | 2017.4 |
| v2.2 | 2017.4 |
| v2.3 | 2018.2 |

<PYNQ repository>/sdbuild/scripts/setup\_host.sh

1. Source the appropriate settings files from Vivado and Xilinx SDK
2. Navigate to the following directory and run make

cd <PYNQ repository>/sdbuild/

make

The build flow can take several hours. By default images for all of the supported boards will be built. To build for specific boards then pass a BOARDS variable to make

make BOARDS=Pynq-Z1

## **Retargeting to a Different Board**

Additional boards are supported through external *board repositories*. A board repository consists of a directory for each board consisting of a spec file and any other files. The board repository is treated the same way as the <PYNQ repository>/boards directory.

### **Elements of the specification file**

The specification file should be name <BOARD>.spec where BOARD is the name of the board directory. A minimal spec file contains the following information

ARCH\_${BOARD} := arm

BSP\_${BOARD} := mybsp.bsp

BITSTREAM\_${BOARD} := mybitstream.bsp

where ${BOARD} is also the name of the board. The ARCH should be arm for Zynq-7000 or aarch64 for Zynq UltraScale+. If no bitstream is provided then the one included in the BSP will be used by default. All paths should be relative to the board directory.

To customise the BSP a petalinux\_bsp folder can be included in the board directory the contents of which will be added to the provided BSP before the project is created. See the ZCU104 for an example of this in action. This is designed to allow for additional drivers, kernel or boot-file patches and device tree configuration that are helpful to support elements of PYNQ to be added to a pre-existing BSP.

If a suitable PetaLinux BSP is unavailable for the board then this can be left blank and an HDF file provided in the board directory. The system.hdf file should be placed in the petalinux\_bsp/hardware\_project folder and a new generic BSP will be created as part of the build flow.

### **Board-specific packages**

A packages directory can be included in board directory with the same layout as the <PYNQ repository>/sdbuild/packages directory. Each subdirectory is a package that can optionally be installed as part of image creation. See <PYNQ repository>/sdbuild/packages/README.md for a description of the format of a PYNQ sdbuild package.

To add a package to the image you must also define a STAGE4\_PACKAGE\_${BOARD} variable in your spec file. These can either packages in the standard sdbuild library or ones contained within the board package. It is often useful to add the pynq package to this list which will ensure that a customised PYNQ installation is included in your final image.

### **Using the PYNQ package**

The pynq package will treat your board directory the same as any of the officially supported boards. This means, in particular, that:

1. A notebooks folder, if it exists, will be copied into the jupyter\_notebooks folder in the image. Notebooks here will overwrite any of the default ones.
2. Any directory containing a bitstream will be treated as an overlay and copied into the overlays folder of the PYNQ installation. Any notebooks will likewise by installed in an overlay-specific subdirectory.

## **Building from a board repository**

To build from a third-party board repository pass the BOARDDIR variable to the sdbuild makefile.

cd <PYNQ repository>/sdbuild/

make BOARDDIR=${BOARD\_REPO}

The board repo should be provided as an absolute path. The BOARDDIR variable can be combined with the BOARD variable if the repository contains multiple boards and only a subset should be built.

[Next](https://pynq.readthedocs.io/en/v2.3/pynq_package.html)  [Previous](https://pynq.readthedocs.io/en/v2.3/overlay_design_methodology/overlay_tutorial.html)