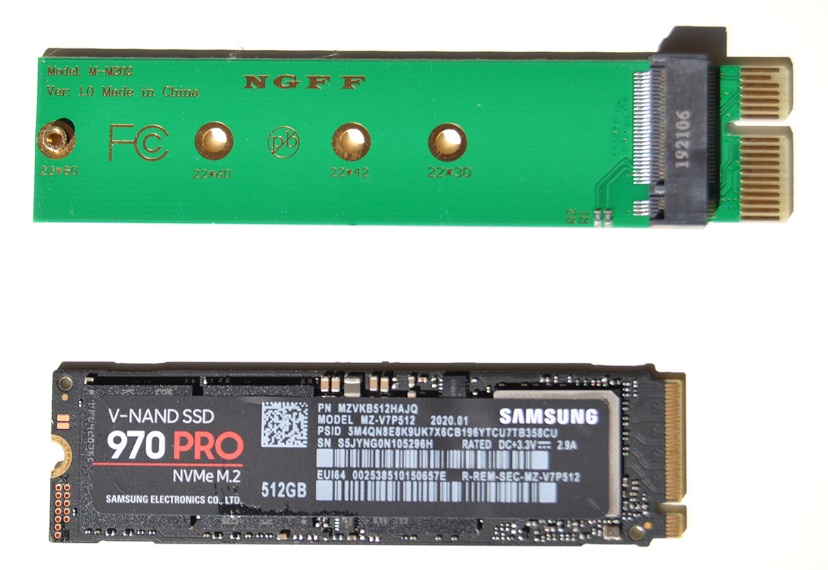
# NVMe SSD for Ultramyir

The UltraMyir board by [MYIR](http://www.myirtech.com/list.asp?id=613) has a PCIe slot that can be used to connect an NVMe SSD drive. Here I will show how to configure the FPGA PS to use a Samsung 970 Pro as an SSD end point.

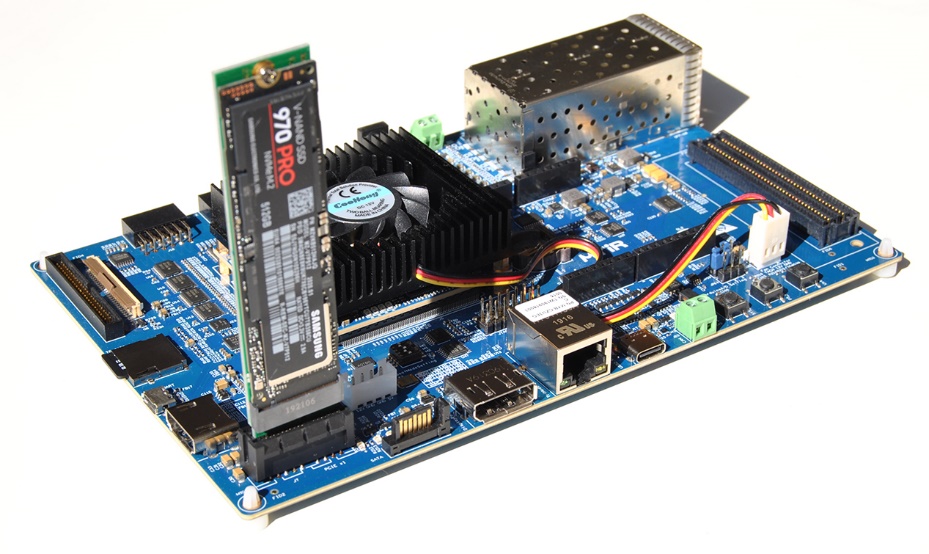
We need an adapter board to interface between the PCIe and the M.2 connector the SSD has. They are readily available, I got the one below on eBay for less than £4.



The SSD is easily assembled into the adapter and secured with the provided screw. Connector polarity prevents inserting in the wrong way:



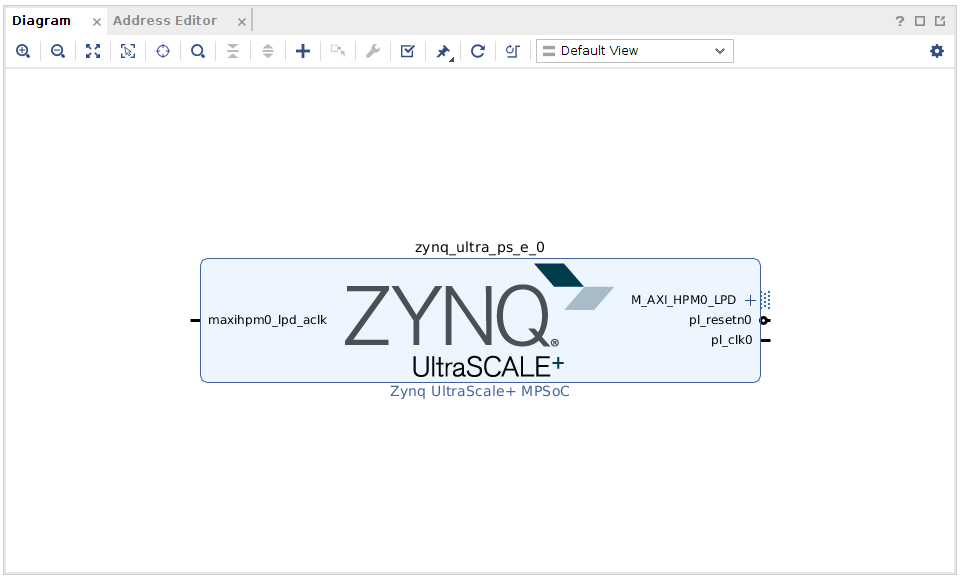
Then it fits into the UltraMyir PCIe slot:



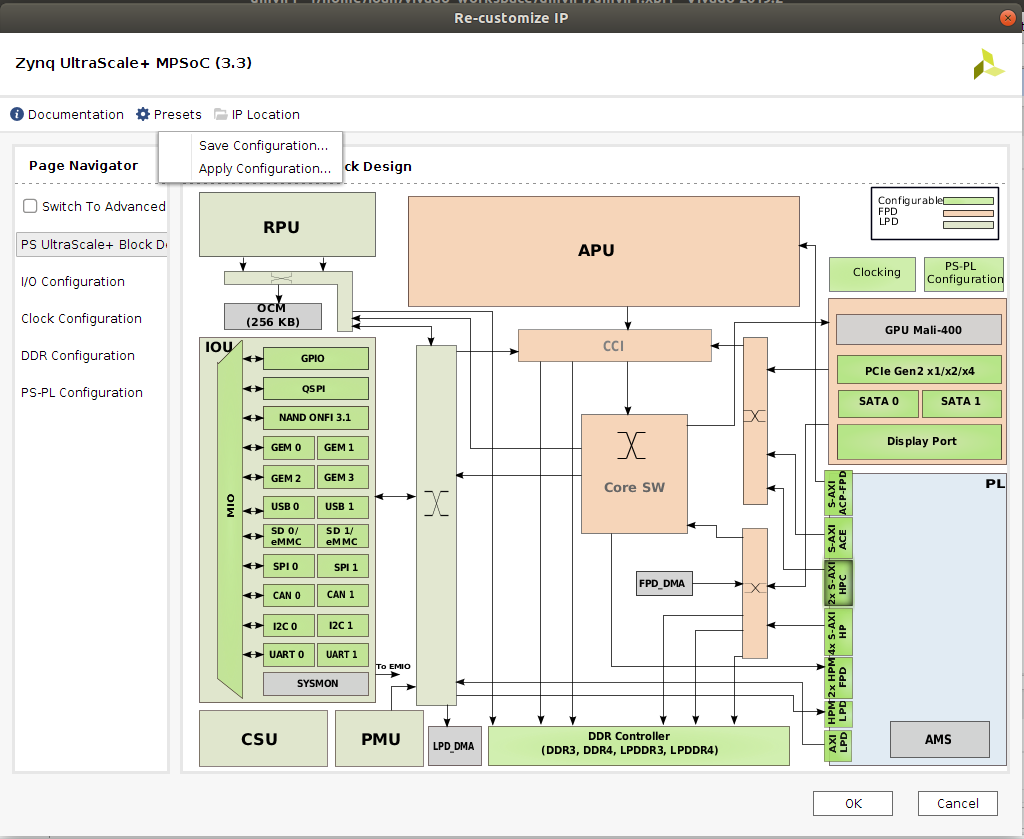
## Hardware design

Open Vivado and create an RTL project. Configure it for the UltraMyir board or select its device (xczu3eg-sfvc784-1-e).

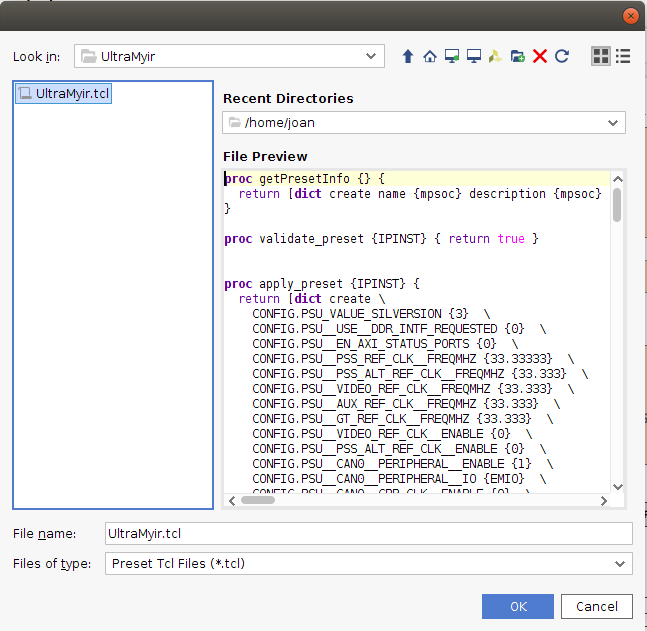
Create a block design and add to it the Zynq Ultrascale+ MPSoC IP:



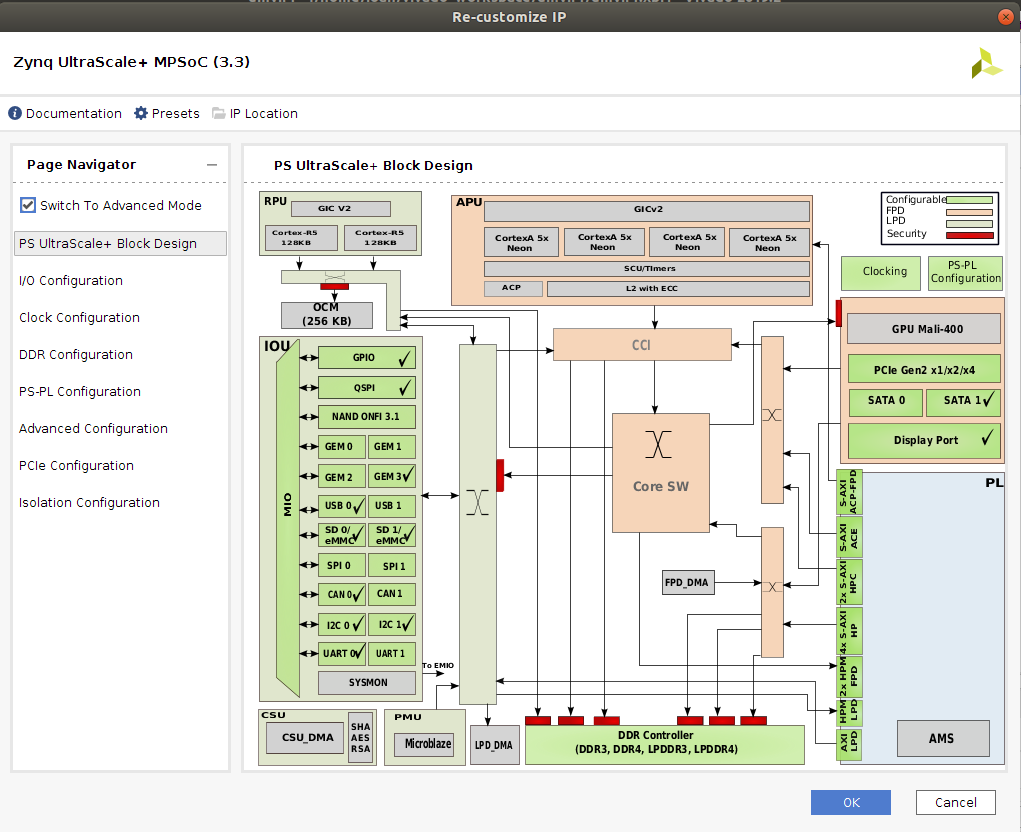
Double click on it to configure it. Click on the Presets button and select “Apply Configuration…”



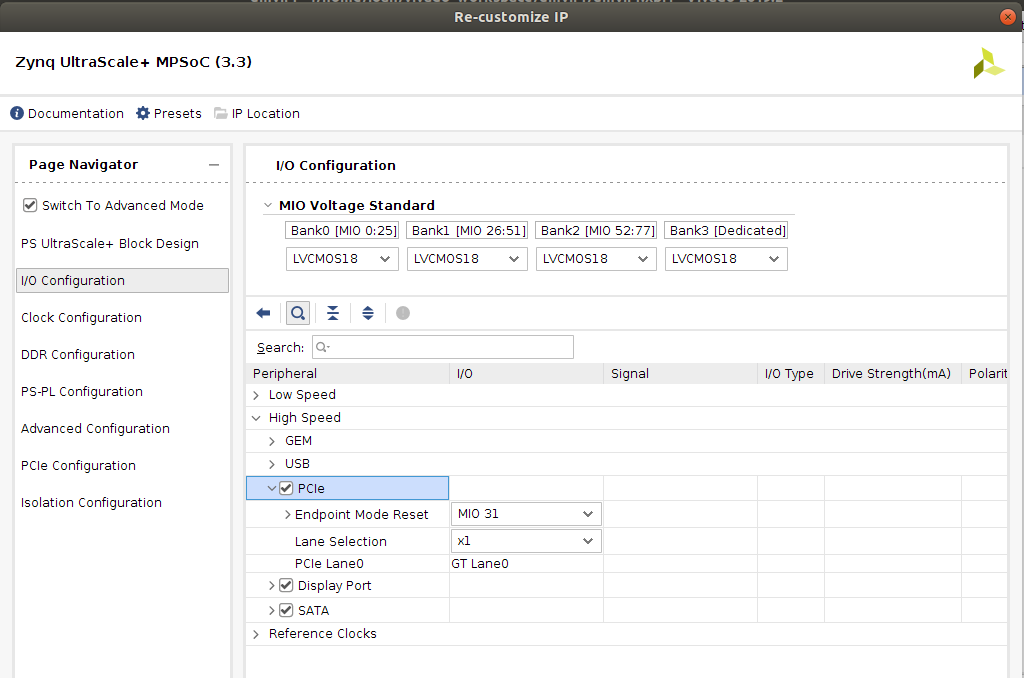
Browse for the tcl configuration file (Ultramyir\_PS\_config.tcl) and click OK.



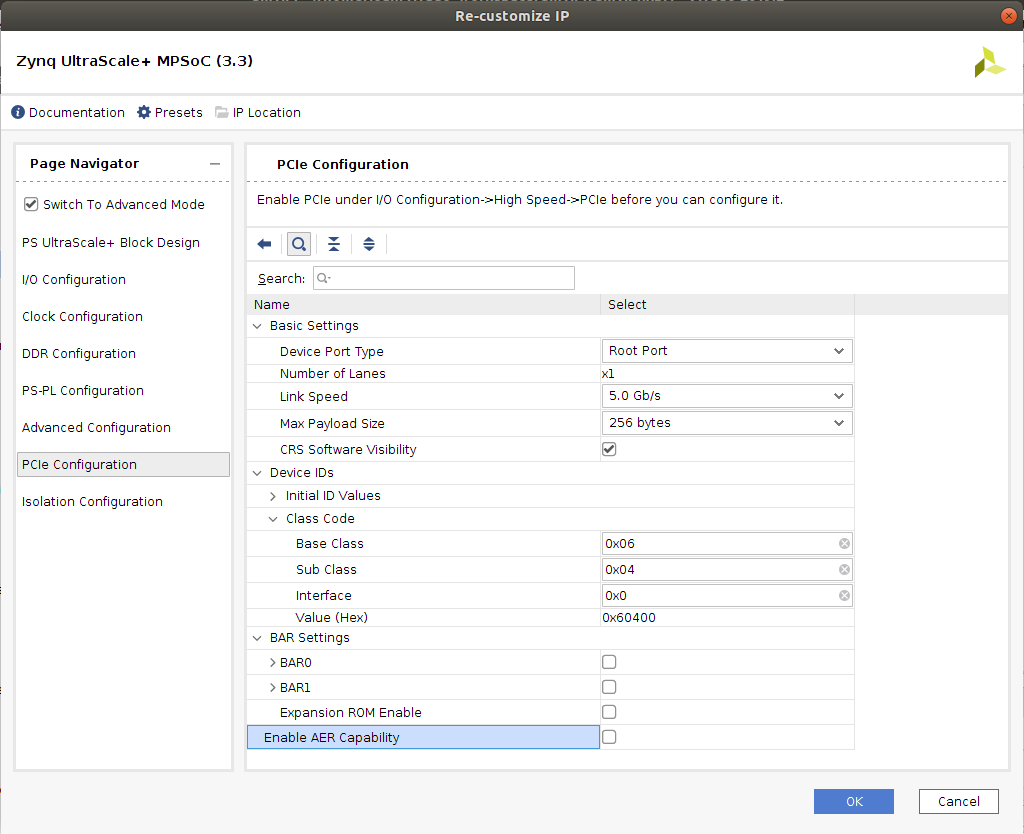
The basic configuration does not include PCIe so we have to do it by hand, first click on the ‘Switch to advanced’ box:



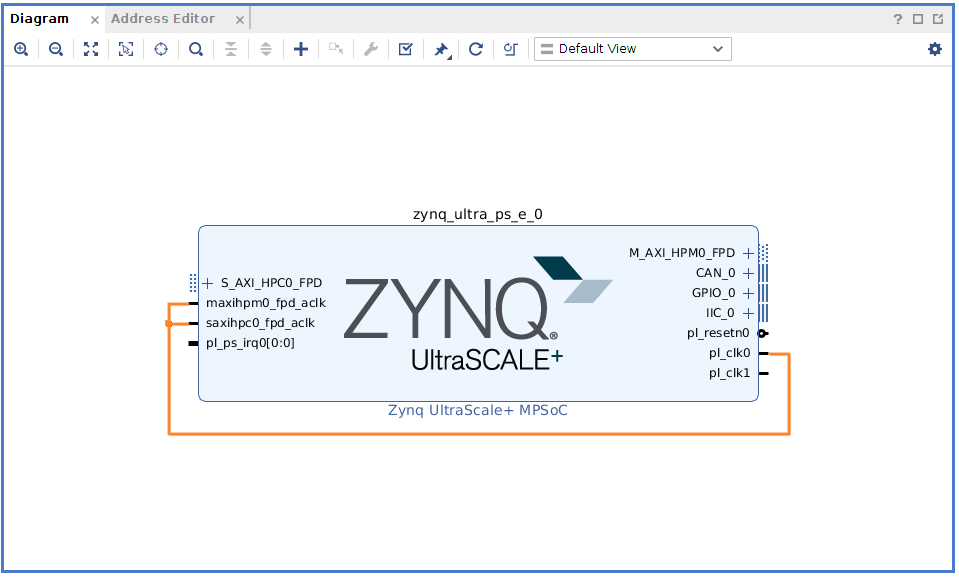
In the I/O configuration tab, activate PCIe and change the Reset pin to MIO31:



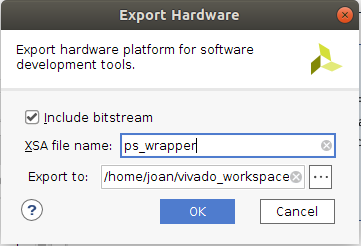
In the PCIe configuration tab, change the Port Type to Root, select one lane, check the CRS Software Visibility and set the Base Class and Sub Class to 0x06 and 0x04 respectively. Then click OK.



Finally, connect clk0 to the AXI clock inputs as below (AXI ports are not used and could as well being turned off).



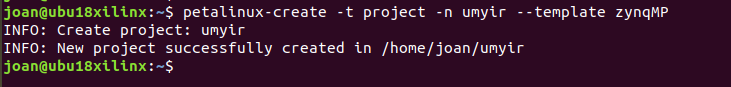
Validate the design and create an HDL wrapper by right clicking on the block diagram in the Design Sources panel. Once this is done, generate the bitstream and export the hardware definition file (xsa) with File > Export > Export Hardware. Check the box ‘Include bitstream’:



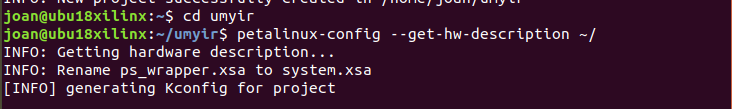
Once the hardware definition and bitstream has been created we can close Vivado.

## Petalinux creation

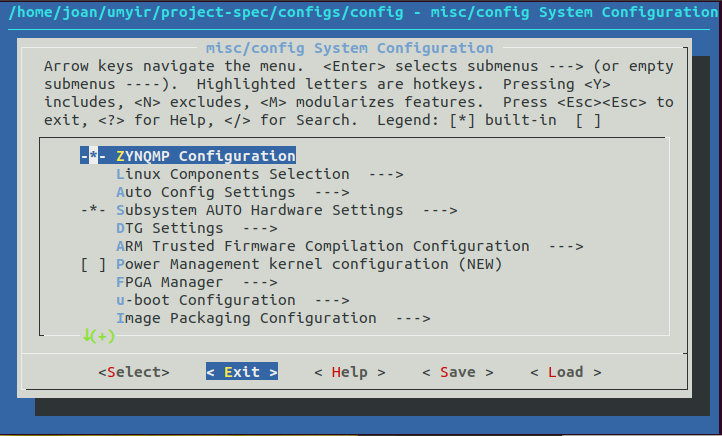
It is assumed here you have a Linux machine (I’m using Ubuntu 18.04 on a Virtual machine) with Petalinux tools installed (I have v2019.2). On a terminal in the root folder create a petalinux project:



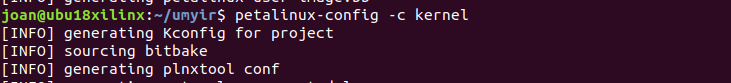
Change to the created directory and configure the project with the exported hardware. Here I previously copied the xsa file generated with Vivado to the root folder. Note that only one hardware description file must exist in the specified folder.



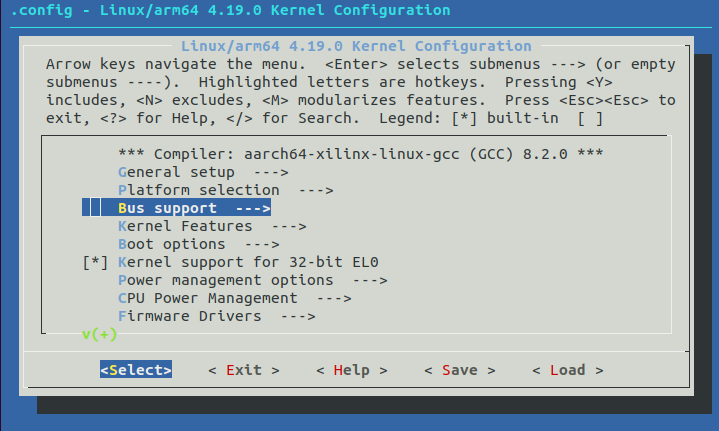
The base configuration does not need any changes so exit and save it:



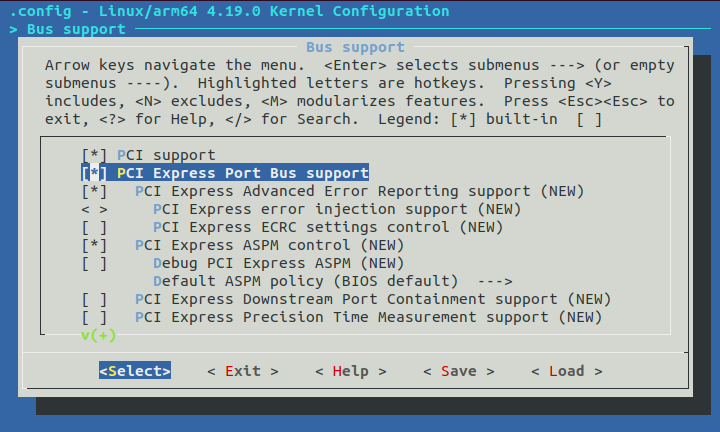
The kernel needs some configuration. run the following command:



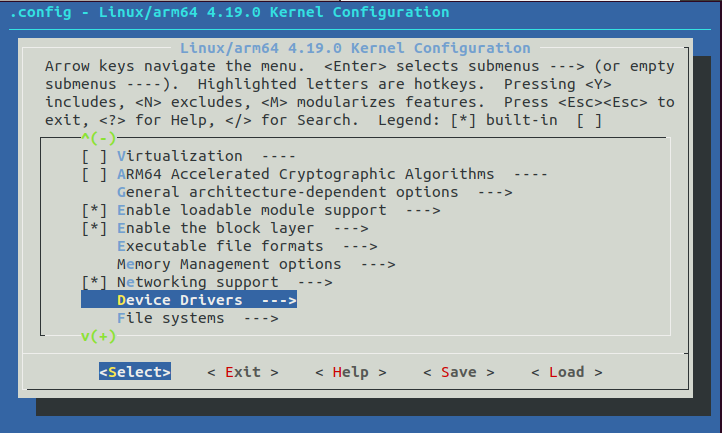
Browse to Bus Support and enter:



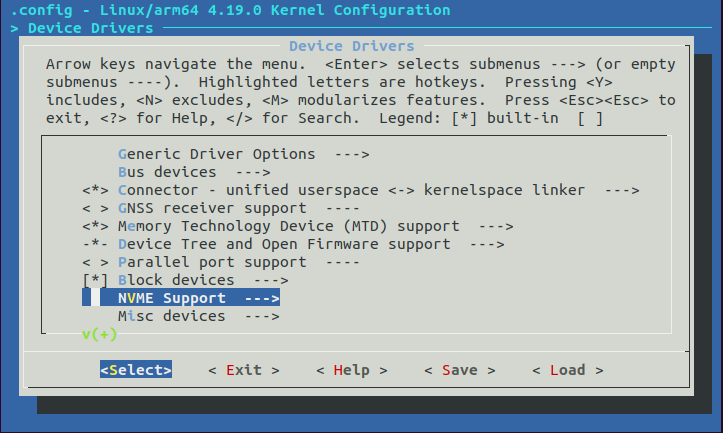
Here activate the PCI Express Port Bus Support option and leave others as they are.



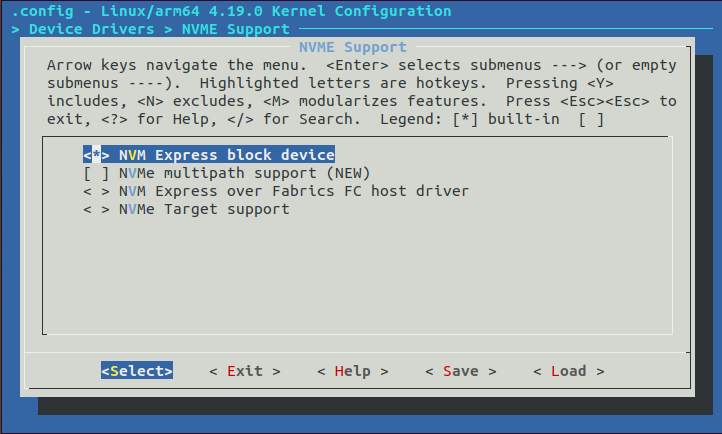
Go one level back, browse to Device Drivers and enter:



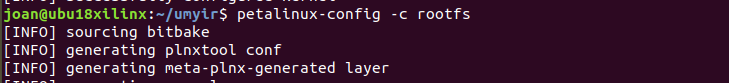
Browse for NVMe Support and enter:



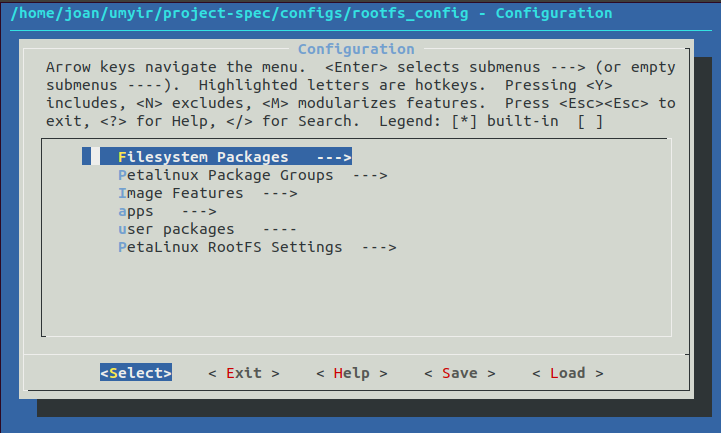
Here activate NVMe Express Block Device:



Save and exit, then configure the root file system with the following command:



Enter Filesystem packages



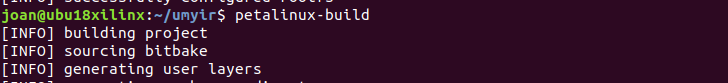
Inside base > util-linux, check util-linux, util-umount, util-mount, util-mkfs, util-fdisk

Inside base > e2fsprogs, check e2fsprogs-mke2fs

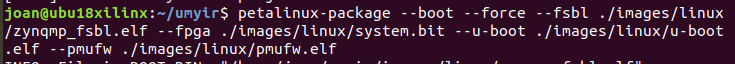
Inside console > utils, check libpci, pciutils

Save and exit.

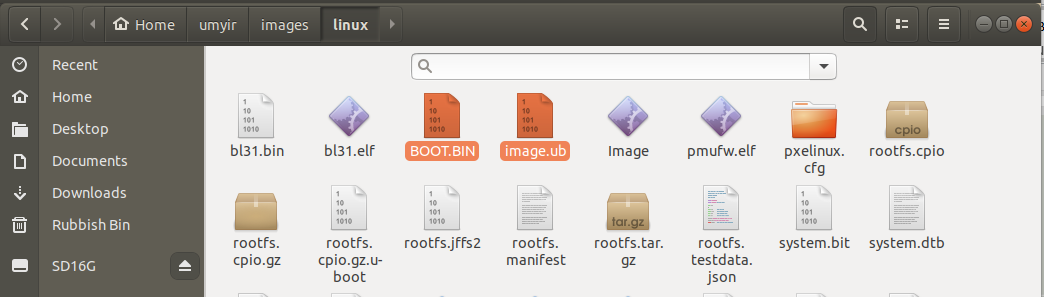
Build with this command:



Last, package to boot from SD card:

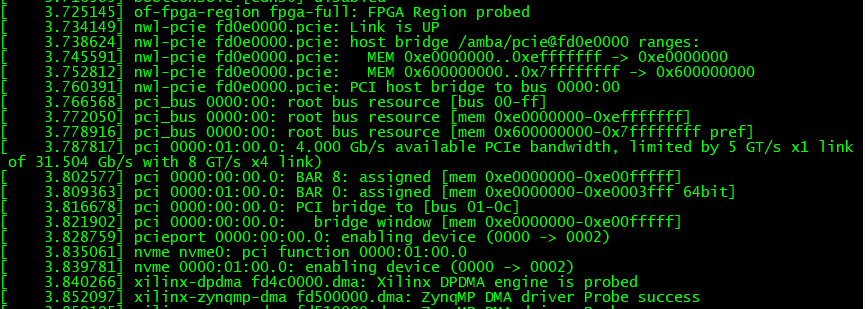


The files boot.bin and image.ub should be in the images/linux folder inside the project folder. Insert an SD card and copy these two files to the card (FAT formatted)

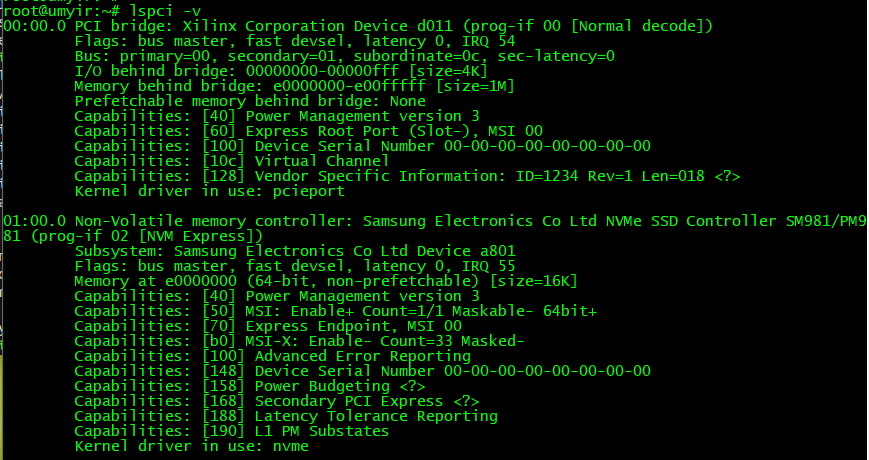


Extract the SD card, insert it into the ultramyir board, set the boot switches to boot from SD card, connect a USB cable to use the serial console and power the board:

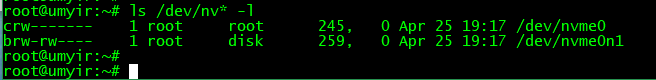
The boot log should show the PCI:



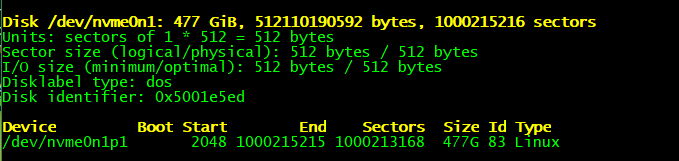
And lspic command should be like:



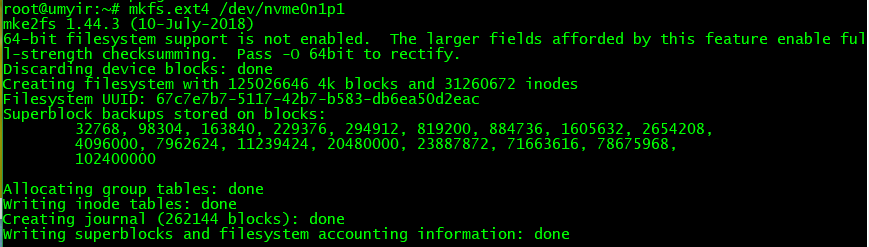
Check it appears in /dev/



Let’s create and format a partition with fdisk:



And create a filesystem with mkfs.ext4:



Reboot. The disk should auto-mount:

