Introduction to PetaLinux

Abigail Butka, Muhammed Kawser Ahmed, Wade Fortney, Christophe Bobda

January 21, 2025

1 Objective

The learning objective of this module is to build and install PetaLinux, a Linux OS for embedded systems, onto an FPGA. PetaLinux is embedded Linux development environment that streamlines the process of creating and deploying Linux-based applications on FPGAs. By integrating PetaLinux with FPGA hardware you can have the benefits of working in hardware, through parallel processing and hardware acceleration, as well as the capabilities of the higher-level Linux operating system like a user interface and scheduling.

In this module you will create a Vivado project for your FPGA, install PetaLinux into Windows Subsystem for Linux, build a custom PetaLinux image, install this onto an SD card using a Linux Virtual Machine, and then modulate an LED over a serial port using PuTTY.

For more conceptual details of this module, please refer to the accompanying PowerPoint presentation and video.

NOTE: In the unfortunate case that the following tutorial no longer generates PetaLinux properly, the files in the accompanying ZIP file will allow you to begin from Task 4 Step 12 with the PYNQ-Z2.

2 Equipment and Software Needed for this Module

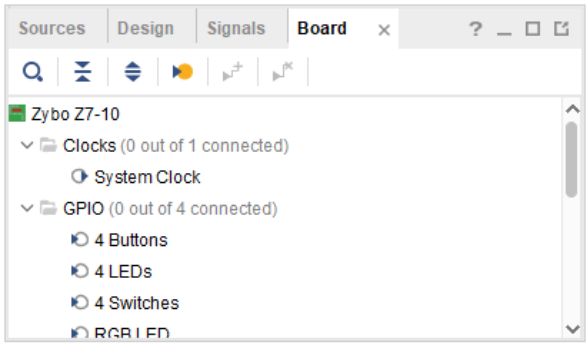
1. Windows 10/11 or Ubuntu 20.04.
   (a) If you are using Windows 10/11, you must also download the following:
   (b) VirtualBox
   (c) Download the Ubuntu 20.04 Desktop Image (ubuntu-20.04.5-desktop-amd64.iso)
2. Xilinx Vivado 2021.1.
3. PetaLinux Version 2021.1. Download PetaLinux from Xilinx's website.
   (a) For more information, refer to the specific hardware/software requirements in the documentation PetaLinux Tools Documentation: Reference Guide (UG1144).
4. Xilinx FPGA with a minimum 32MB RAM.
   (a) This lab can be performed using either a ZYNQ ARM Cortex-A9 processor or a MicroBlaze soft processor. This lab was tested using the PYNQ-Z2.
5. A USB-UART cable to program the board and display the output in the terminal.
6. A 16GB microSD card and microSD card reader.

3 Module Description

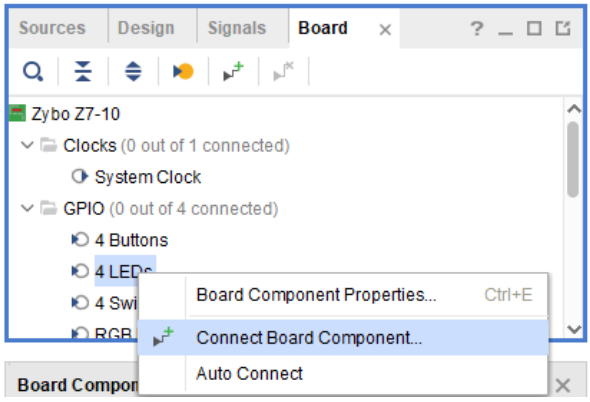
3.1 Task 1: Create Vivado Project for PetaLinux

1. Open Vivado and create a new Vivado project. Make sure you choose the FPGA that you have. This tutorial uses the PYNQ-Z2.

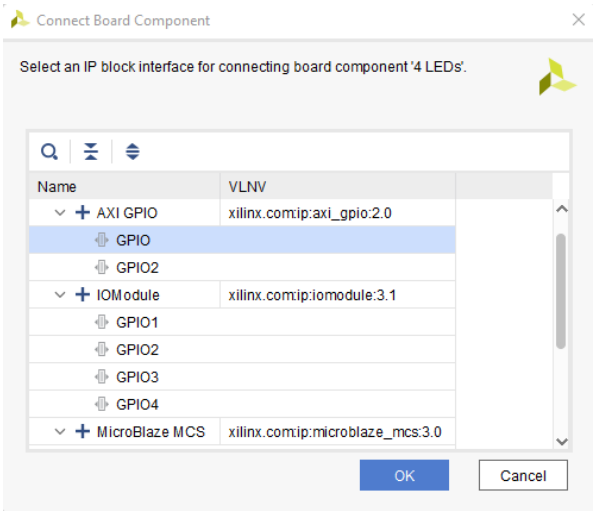
2. Create Block Design.
3. Add IP ZYNQ7 Processing System to the Block Diagram.
4. Run Block Automation, select all boxes, and click OK.
5. Next, we will begin adding the AXI_GPIO IP cores.
6. Navigate to the Board tab.



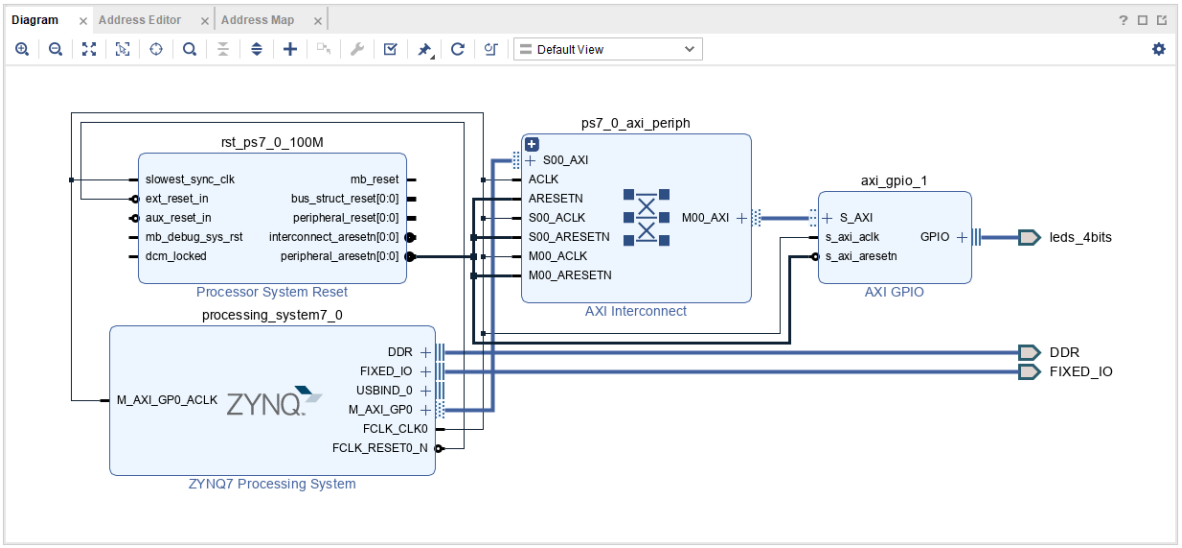
7. Right-click on 4 LEDs under GPIO and select Connect Board Component.



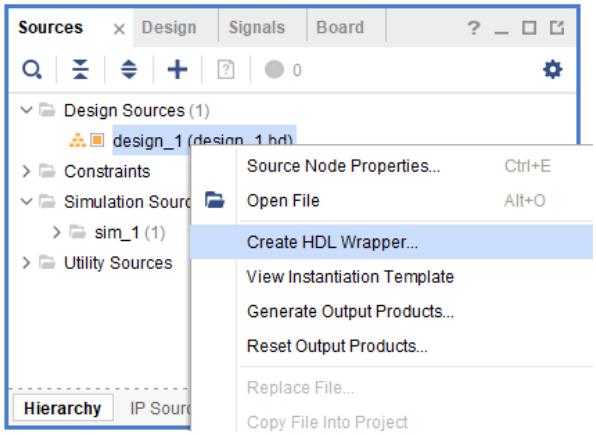
8. This will open a new window titled Connect Board Component. Select AXI GPIO and click OK.



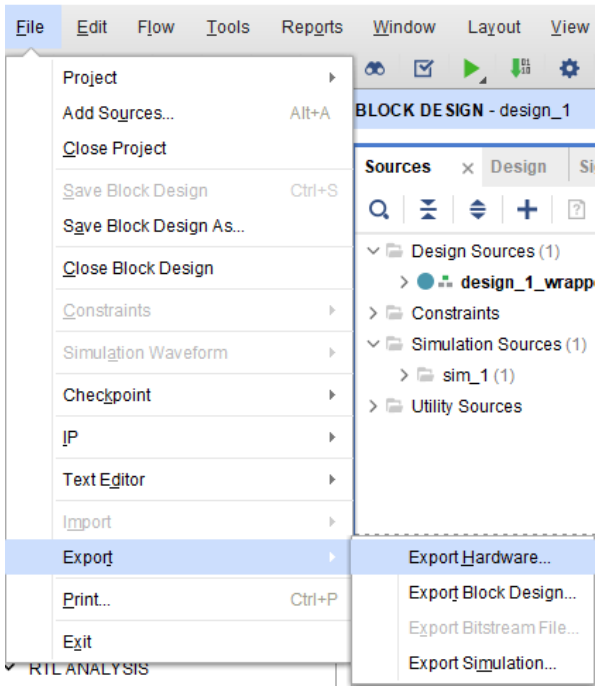
9. Next, click Run Connection Automation
10. Click Regenerate Layout. Your block diagram should look like the image below.



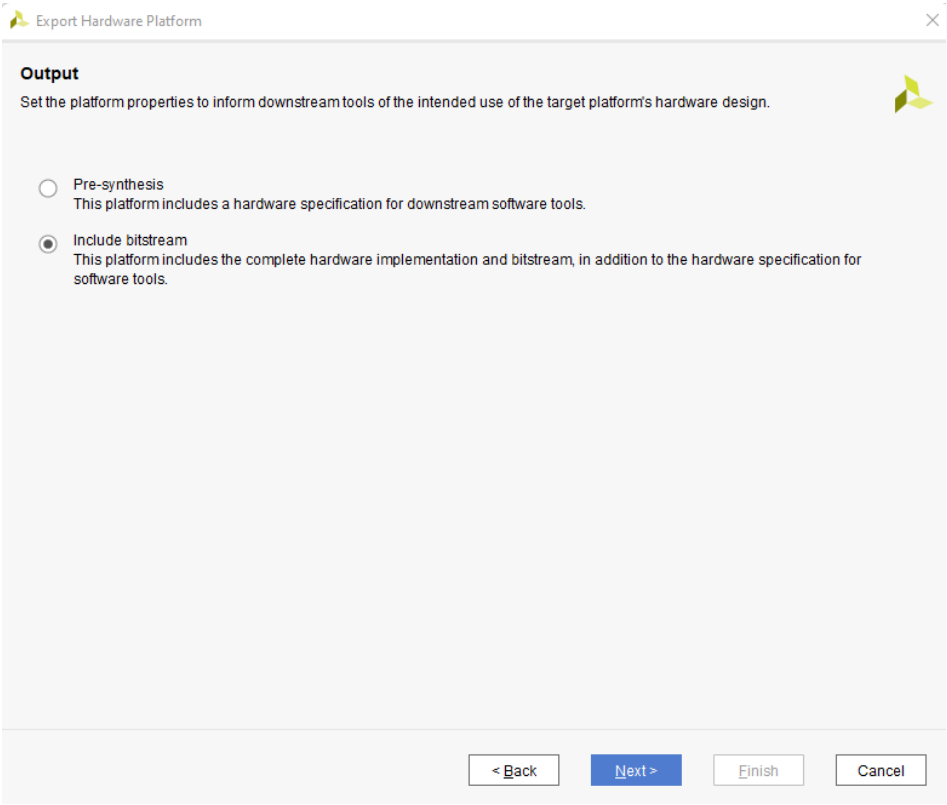
11. Finally, generate the bitstream by clicking on Generate Bitstream in the Flow Navigator.
12. When the window "Bitstream Generation successfully completed" pops up, click cancel.
    (a) If you do not click cancel this will open the implemented design showing the FPGA pin placements.
    (b) We do not need to see this information now.
13. Next, navigate to the Sources tab and expand Design Sources.



14. Right-click design_1.bd and click Create HDL Wrapper. Then select Let Vivado Manage Wrapper and Auto-Update and click OK.
15. This will create a design_1_wrapper.v file that will now be the top file of the design.
16. Next, click on File/Export/Export Hardware.



17. Click Next, ensure that 'Include Bitstream' is selected, and click Next again.



18. Leave the XSA file named design_1_wrapper, note the location of this file for use in Task 4, then click Next and Finish.

19. You have successfully created the hardware description for your FPGA.

3.2 Task 2: Creating a PetaLinux Image

In this task, you will create a simple PetaLinux image to run on the hardware description you created in the previous Task.

1. To begin, ensure you have Windows Subsystem for Linux Version 2 (WSL2) installed on your computer by running the below command in Windows PowerShell. If your operating system is already Linux-based and version 20.04, continue to Step 7.

   ```
   wsl --version
   ```

   (a) If your computer reports a WSL version below 2, please refer to the official WSL2 requirements to confirm your computer can run WSL2.

   (b) If no version of WSL is installed, please refer to the WSL2 installation guide

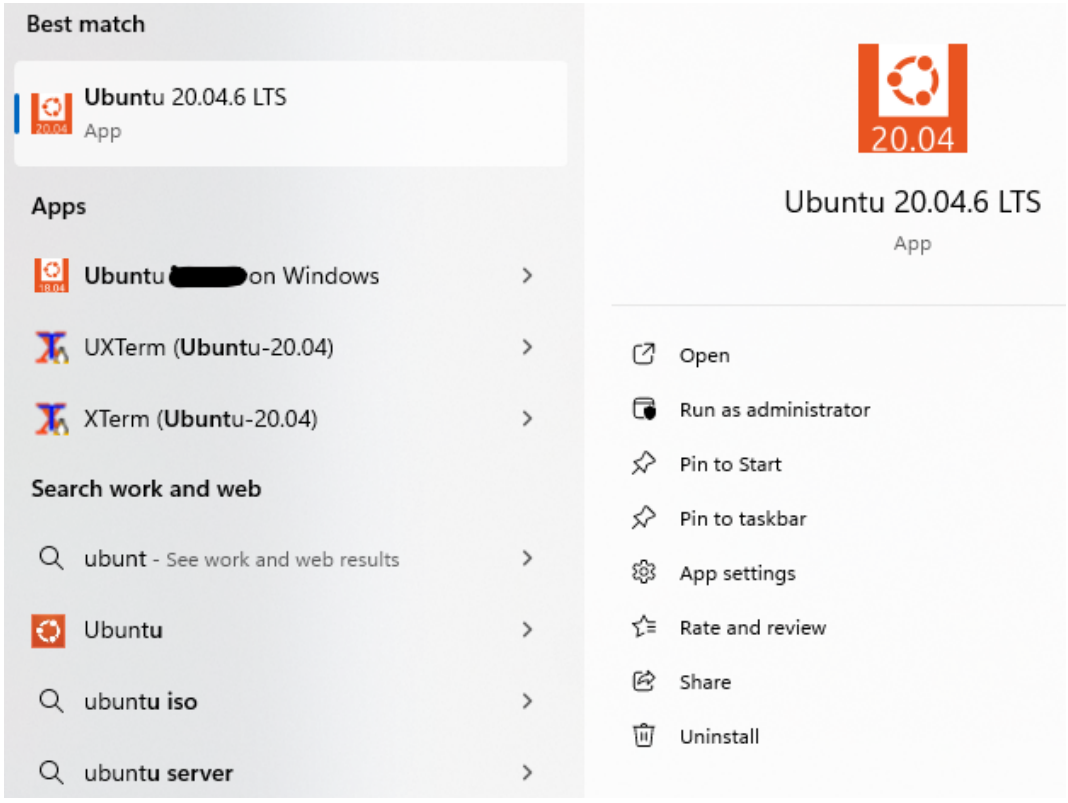
2. Next, in Windows Powershell, run the following commands to confirm you have the Ubuntu-20.04 distribution and then install it.

   ```
   wsl -l -o
   wsl --install -d Ubuntu-20.04
   ```

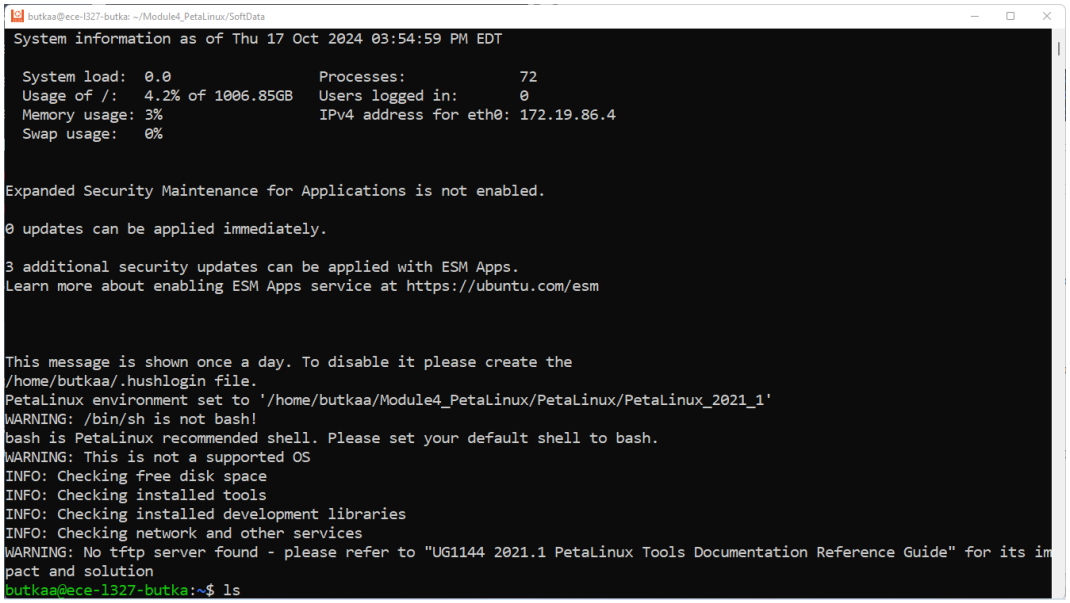
3. Eventually, this will open Ubuntu within Powershell and prompt you to create a username and password. Choose a username and password you will remember.

4. Close Windows Powershell

5. Under the Windows Search Bar, search for Ubuntu, and you should see the following pop-up.



6. Click on this to open Ubuntu. This will open a new window that will look like the image below.



7. You are currently in the home directory of Ubuntu. Run the following commands to create three directories. When you run ls, the result should look like the image below.

```
mkdir DesktopWSL
mkdir SoftData
mkdir PetaLinux
ls
```

8. Next, if you have not done so already, download Petalinux Version 2021.1 as listed in Equipment and Software, bullet point three.

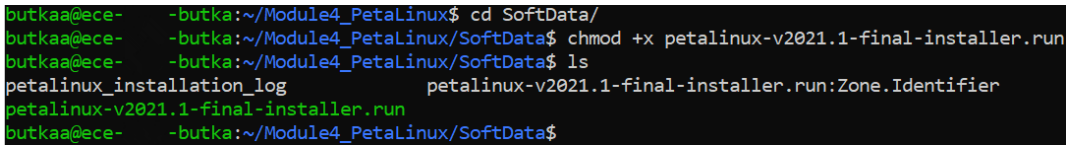
9. In the Ubuntu window, run the following command. This will open the home folder of Ubuntu.

```
explorer.exe .
```

10. Now, move the PetaLinux-v2021.1-final-installer.run that you downloaded from Xilinx to the SoftData folder.

11. Back in Ubuntu, run the following commands.

```
cd SoftData
chmod +x petalinux-v2021.1-final-installer.run
```



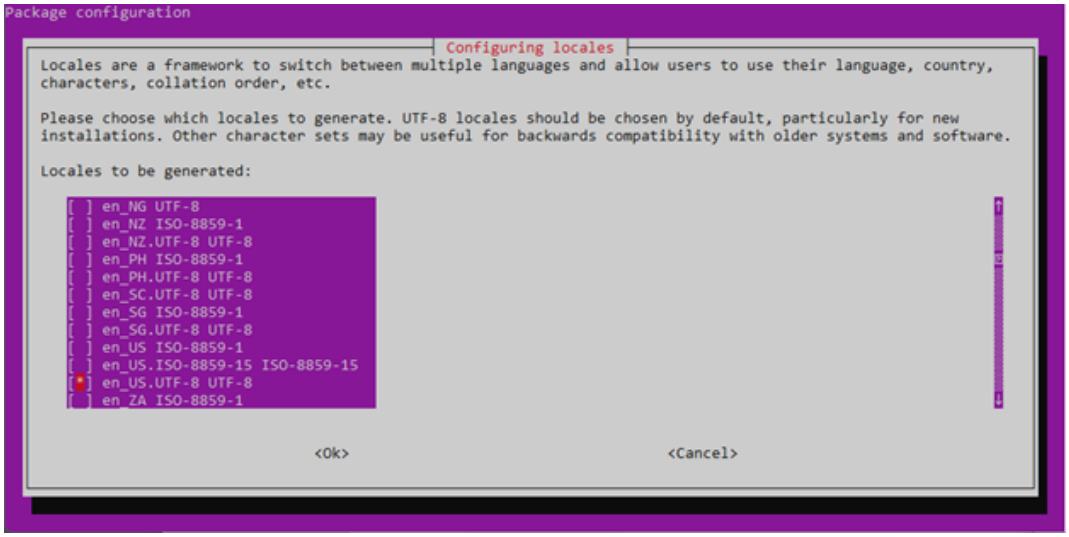
12. Next, you will need to install several packages. Run the following commands in order. It is recommended that these commands be run one by one.

```
sudo apt update
sudo apt full-upgrade

sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install iproute2 gawk python3 build-essential gcc git make net-
    tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1
    gnupg wget git-core diffstat chrpath socat xterm autoconf libtool tar
    unzip texinfo zlib1g-dev gcc-multilib automake zlib1g:i386 screen pax
    gzip cpio python3-pip python3-expect xz-utils debianutils iputils-ping
    python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev

sudo apt-get install rsync
sudo apt-get install locales
```

13. Next, run the following command. This will open a new window.

```
sudo dpkg-reconfigure locales
```

14. In this window, scroll down to en_US.UTF-8 and press enter. On the subsequent screen, select en_US.UTF-8 and press enter again.



15. When this screen exits, run the following commands.

```
sudo locale-gen en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
sudo apt-get install libgtk2.0-0
sudo apt install libtinfo5
sudo apt-get install x11-xkb-utils
```

16. You have now installed all of the required packages and settings for the WSL2 execution of PetaLinux.

17. Next, ensure you are still in the SoftData folder and run the following command to proceed with the PetaLinux Installation.

```
./petalinux-v2021.1-final-installer.run --dir ../PetaLinux/2021_1 --platform "arm"
```

18. This will initiate a set of EULA agreements. After reading each agreement, press 'q' to exit, then enter 'y' to accept the agreement.

19. Once done, this will initiate the PetaLinux build. This should not require any user interaction and should not produce any errors. Please ensure your computer does not shut off during the build sequence.

3.3 Task 3: Enable Ubuntu to Run PetaLinux Commands

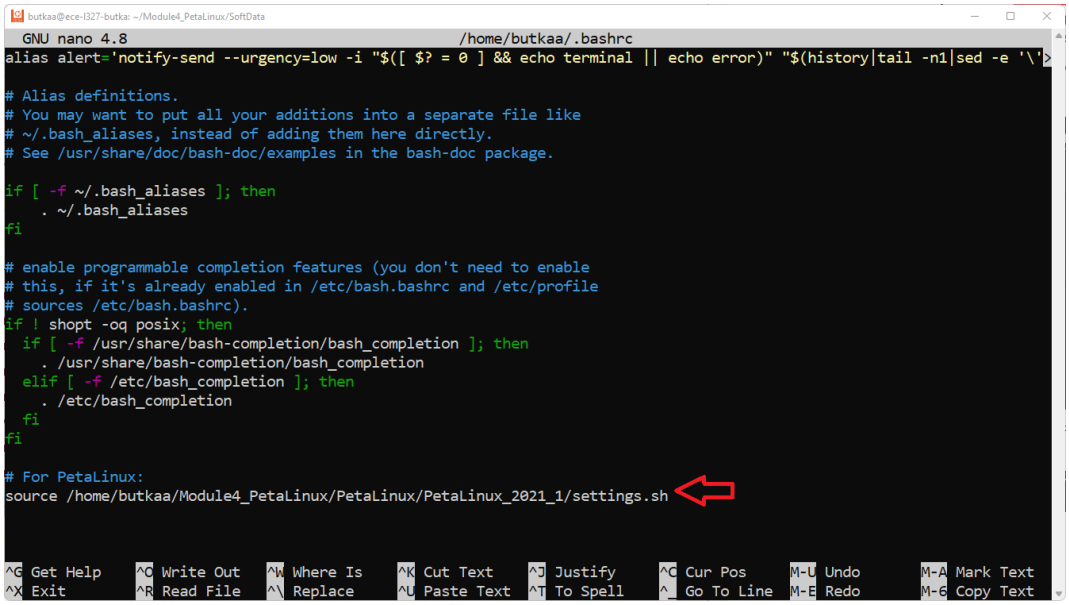
The following instructions will allow you to run PetaLinux from any location of your Ubuntu installation.

1. Run the following command to open the source file for Bash.

```
nano ~/.bashrc
```

2. Scroll to the bottom of the file using the down-arrow key and type the following lines of code. The final product should look similar to the image below.

```
# Bash for PetaLinux
source ~/PetaLinux/2021_1/settings.sh
```



3. Next, press CTRL-S to save the document, then CTRL-X to exit.

4. To confirm these steps were performed properly, run the following commands. These commands should have the same output as the image below.

```
echo $PETALINUX
bash
```

```
butkaa@ece-l327-butka:~/Module4_PetaLinux/SoftData$ bash
PetaLinux environment set to '/home/butkaa/Module4_PetaLinux/PetaLinux/PetaLinux_2021_1'
WARNING: /bin/sh is not bash!
bash is PetaLinux recommended shell. Please set your default shell to bash.
WARNING: This is not a supported OS
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "UG1144 2021.1 PetaLinux Tools Documentation Reference Guide" for its im
pact and solution
butkaa@ece-l327-butka:~/Module4_PetaLinux/SoftData$
```

5. You can now continue to Task 4.

3.4 Task 4: Creating and Running a PetaLinux Image

Finally, this task will create a Petalinux image, upload this to the SD card, and then modulate one of the LEDs.

1. To begin, run the following command to create a new PetaLinux project for the ZYNQ processor.

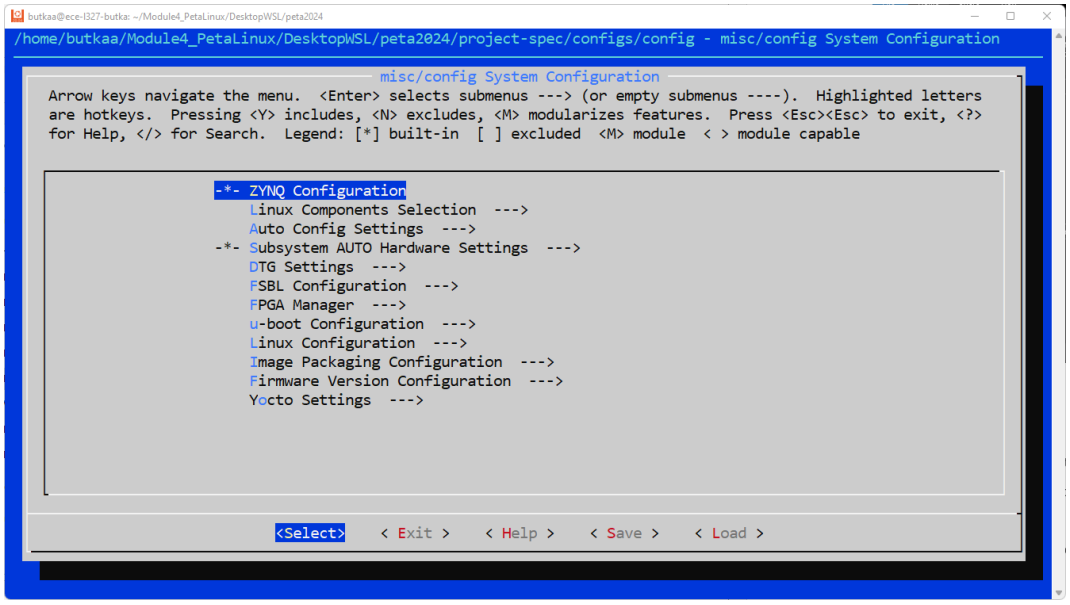
```
cd ~/DesktopWSL
petalinux-create --type project --template zynq --name peta2021
```

2. Next, run the following command to open File Explorer to your current location and copy the .xsa file produced in Task 1 to the peta2021 directory.

```
explorer.exe .
```

3. Next, run the following command to configure your PetaLinux setup. The window should look like the image below.

```
cd ~/DesktopWSL/peta2021
petalinux-config --get-hw-description
```



4. Use the up/down arrow keys and press enter to select Image Packaging Configuration → Root filesystem type → EXT4 (SD/eMMC/SATA/USB)

5. Use the left/right arrow keys and enter to click "Exit" to close the configuration window and go back to the main menu.
6. Use the left/right arrow keys to click "Save," press enter to select OK, then enter again to exit.
7. Now, select Exit to exit the configuration menu.
8. Next, run the following command to build the PetaLinux image. This will take a while; please ensure your computer does not shut off during the build sequence.

```
petalinux-build
```

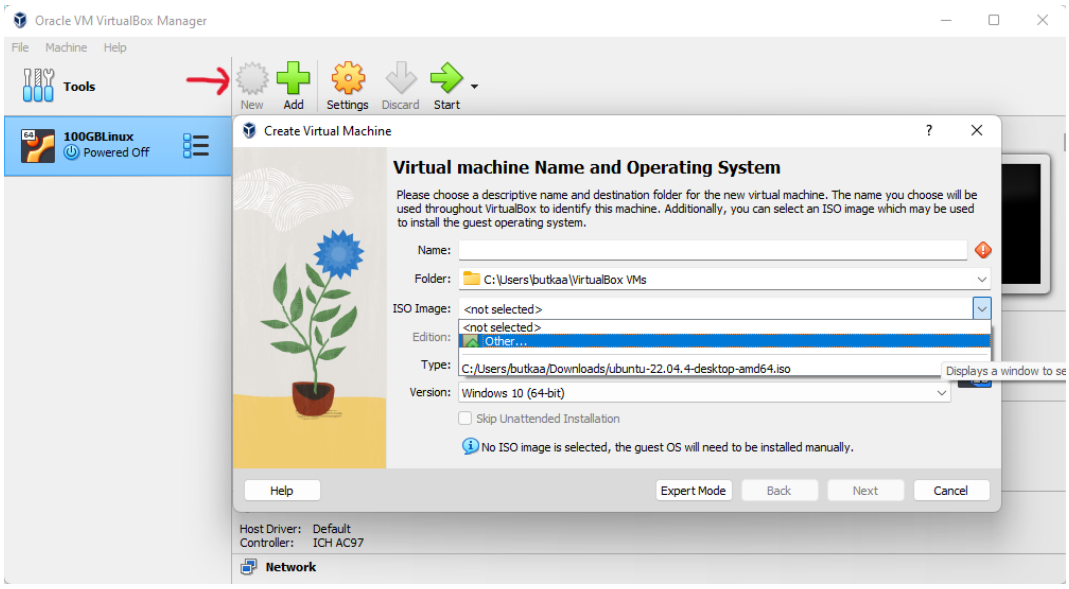
9. After the project build sequence, you will see the following WARNING and INFO. These are fine; please ignore them.
   (a) WARNING: You are running bitbake under WSLv2, this works properly but you should optimize your VHDX file eventually to avoid running out of storage space.
   (b) INFO: Failed to copy built images to tftp dir: /tftpboot
10. Next, run the following command to generate the PetaLinux image.

```
petalinux-package --boot --fpga images/linux/system.bit --fsbl images/
linux/zynq_fsbl.elf --u-boot
```

11. You will again receive warnings. These are fine. Please ignore them.

```
WARNING: Unable to access the TFTPBOOT folder /tftpboot!!!
WARNING: Skip file copy to TFTPBOOT folder!!!
```

12. Now that you have created the PetaLinux image, you will need to format your microSD card and upload the appropriate files.
   (a) If you are using **Windows**, follow the next steps.
   (b) If you are using **Linux**, proceed to Step 24.
13. Ensure you have downloaded VirtualBox and an Ubuntu Desktop image from the Equipment and Software section.
14. Install and open VirtualBox. Click on the blue 'New' button, highlighted with a red arrow.



15. Give your virtual machine a name and drop down the 'ISO Image (not selected)' menu. Select other and navigate to the downloaded ubuntu-20.04.5-desktop-amd64.iso.

16. Click Next and set your username and password.

17. Click Next. You do not need to adjust any hardware settings on this screen. You may ignore this page and click next again.

18. On the virtual hard disk page, allocate more than the minimum disk size if you expect to use Linux again. If you do not expect to use Linux again, click Next.

19. Click Finish and allow VirtualBox to create your Linux Virtual Machine.

20. Once your Ubuntu Virtual Machine has loaded, insert the microSD card into its adapter and plug it into your computer.

21. In VirtualBox, click on the nine dots in a cube on the sidebar and type "Terminal." Select the application that loads.

22. Once here, enter the following commands to allow yourself to use the Sudo command, then log out of VirtualBox by clicking the power button in the top right corner and selecting "Log Out."

```
su -
*Enter your password*
usermod -a -G sudo vboxuser
```

23. Log back into Ubuntu and mount your SD card to Ubuntu by selecting Devices→USB→*Name of your SD card*.

24. Re-open the Terminal, and enter the following commands to remove all partitions from the SD card.

```
fdisk /dev/sdb
d
p
w
```

25. If 'p' returns the following, continue to the next step.

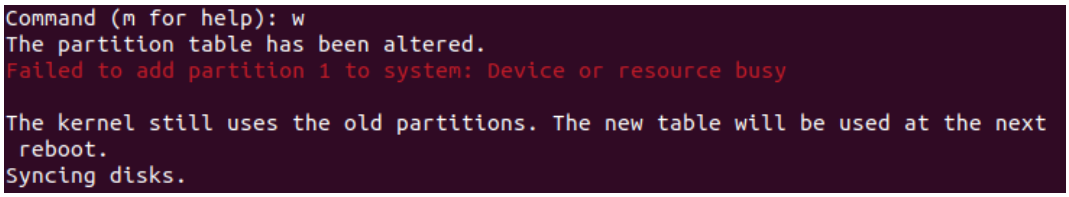
26. Enter the following commands one by one to create a 500MB partition for the BOOT data of PetaLinux

```
n       (New)
p       (Partition)
1       (Partition #)
*Enter* (Select default highest bound for partition)
+500M   (Create 500MB partition)
n       (Do not delete)
t       (Set partition type. No need to select partition number)
c       (Set partition type to W95 FAT32 (LBA))
```

27. Enter the following commands one by one to create a 14.3GB partition for the ROOT data of PetaLinux

```
n       (New)
p       (Partition)
2       (Partition #)
*Enter* (Select default lower bound for partition)
*Enter  (Select default highest bound for partition)
```

28. When done performing these tasks, enter 'w' to write the changes to the SD card. You may receive the warning and failures below. These are fine. To check that the partitions are saved, re-run 'fdisk /dev/sdb', followed by 'p'.



29. You should now be back at the root of Linux. Enter the following commands to write the filesystem types to each partition.

```
mkfs -t vfat -F 32 /dev/sdb1
mkfs -t ext4 /dev/sdb2
```

30. Now, insert your USB drive with the 'peta2024' image files and in VirtualBox, select Devices→USB→*Name of your USB*. My USB drive was listed as USB 3.0.

31. Next, run the command 'lsblk' and identify the corresponding locations of sdb1 and sdb2. For me, these were as seen below.

```
sdb1 /media/butkaa/NEW VOLUME
sdb2 /media/butkaa/df6bbe... (Series of numbers and letters)
```

32. Open your USB Drive in the file manager, open the peta2024/images/linux folder, right-click, and select "Open in Terminal."

33. Ensure you are in the peta2024/images/linux folder. If not, cd to this location.

34. Run the following commands to move the PetaLinux files to their correct locations:

```
su -
cp BOOT.BIN boot.scr image.ub /media/*Your path to sdb1*
sudo cp rootfs.tar.gz /media/Your path to sdb2*
```

35. Next, run the following commands to unzip the rootfs.tar.gz file that you moved to the sdb2, the ROOT drive.

```
cd /media/*Your path to sdb2*
sudo tar xfv rootfs.tar.gz
sudo rm rootfs.tar.gz
```

36. The SD card is now prepared for use. You may close VirtualBox.

37. Back in Windows, eject the SD card and USB drive, then unplug them.

38. Remove the SD card from the adapter and insert it into the PYNQ-Z2 board.

39. Ensure that the PYNQ-Z2's BOOT jumper is correctly placed on the SD option, the left two pins on the top rail. See the yellow boxes on the image below to identify the jumper's correct location.

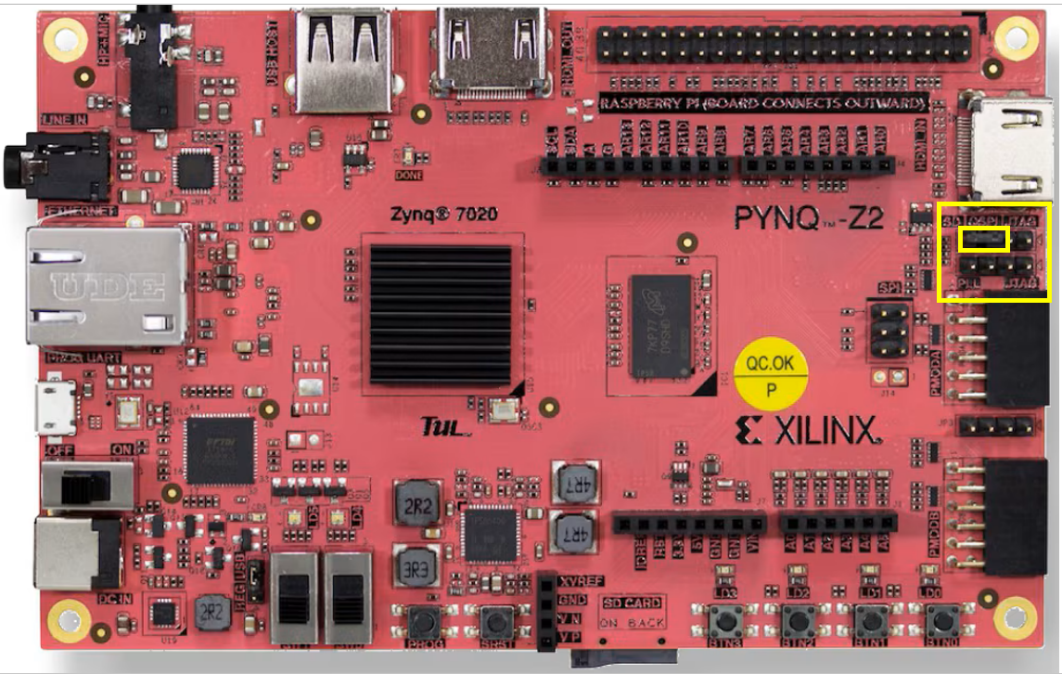
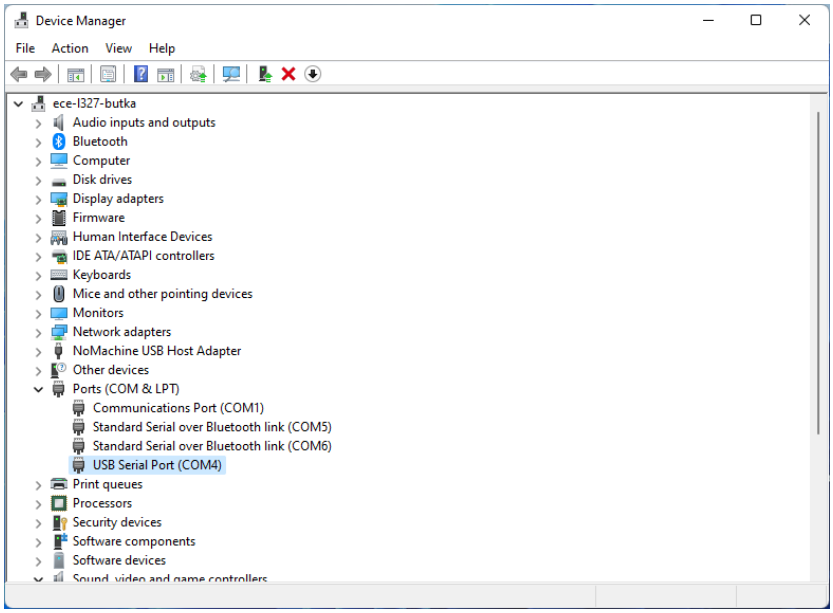


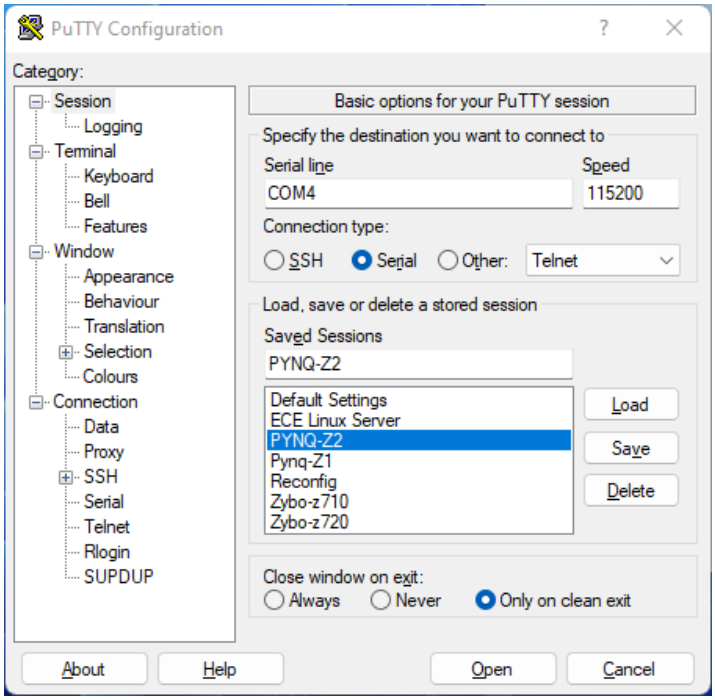
Image from AMD's Website Page on AUP PYNQ-Z2. Modified to Show How to Set Jumpers to Boot From SD Card.

40. Plug the PYNQ-Z2 board into your computer with the USB-to-MicroUSB cable and turn the PYNQ-Z2 board on.

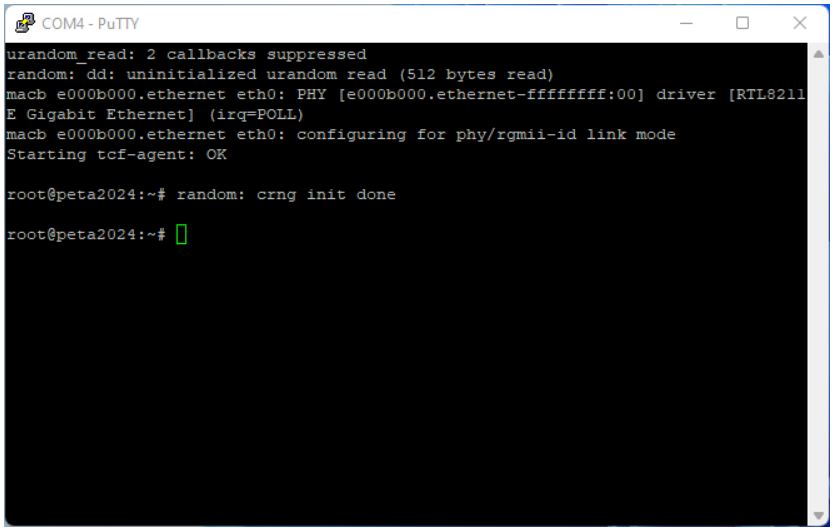
41. Open the Windows Device Manager to identify the COM port of your board. Mine is on COM4. This step only works if the board is turned on!



42. Open PuTTY, select Serial, set the COM port to your identified COM port, and set the speed to 115200, as seen in the image below.



43. Select Open. If all steps were performed correctly, the following will be displayed. If it does not, turn the PYNQ-Z2 board off and back on, then reconnect with PuTTY. There may have been a time-out. The loaded screen should look like the image below.

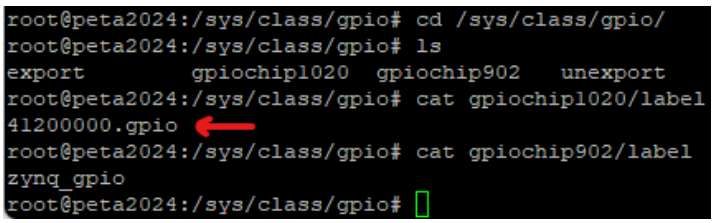


44. Next, we must locate the identifier of the AXI GPIO LEDs inside of PetaLinux by running the following commands.

```
cd /sys/class/gpio/
ls
```

45. You will see the following output, although the numbers may differ. We need to identify which of these are the AXI GPIO for LEDs by running the following command for each GPIO.

```
cat gpiochip*YourGPIONum*/label
```



46. In my case, gpiochip1020 was labeled as a GPIO with the same AXI address as the LEDs, meaning this is the correct GPIO.

47. Next, turn on an LED by running the following commands with the number of your GPIO inserted. *YourGPIONum* is equal to the last digits of the gpiochip for AXI GPIO to LEDs; my GPIO number was 1020.

```
echo *YourGPIONum* > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio*YourGPIONum*/direction (initialises GPIO)
echo 1 > /sys/class/gpio/gpio*YourGPIONum*/value (LED Turns ON)
```

48. The LED has turned on! To turn it off and exit PetaLinux, run the following commands

```
echo 0 > /sys/class/gpio/gpio*YourGPIONum*/value (LED Turns OFF)
echo *YourGPIONum* > /sys/class/gpio/unexport
```

Congratulations! You have installed, built, packaged, and run PetaLinux to modulate an LED!