**Rev0-2020.1-Petalinux Project Setup Guide**

**Author(s):** L. Bidulka

**Release:** 2.0

**Release Date:** 23/12/2020

**Document Type:** Guide

**Project:** NuPRISM

**Keywords:** Vivado, Petalinux, NuPRISM, Firmware, Guide, 2020.1

**History of Changes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Release No.** | **Date** | **Description** | **Authour(s)** |
| 1.0 | 21/08/2020 | Initial Release | R. Payne |
| 2.0 | 23/12/2020 | December Revision | L. Bidulka |
|  |  |  |  |
|  |  |  |  |

# Introduction

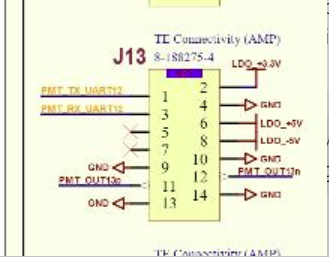
# Vivado and Vivado Project Setup

1. If not installed, download Vivado 2020.1 from <https://www.xilinx.com/support/download.html>
   1. The only chip required is the ZynqMPSoC. To limit the download size, it is recommended that only that one is downloaded
2. Clone the project located [here](https://github.com/nuPRISM/iwcd-xu1-petalinux-rev0-2020-01/tree/canada-develop/documents), be sure to clone the canada-develop branch.
   1. Be sure to clone it as high as possible in your directory to avoid Vivado compilation errors due to length of path
      1. Ex: C:/Repos/
3. Open the project using Vivado 2020.1
4. Run synthesis
   1. If you get some errors, try regenerating output products and synthesizing again.
5. If you want to set up the Integrated Logic Analyzer (ILA) (Do this later, once you know what it is and need to do it for debugging)
   1. Open synthesized design
   2. Navigate to set up a debug and select the signals you would like to view
6. Run implementation
7. Generate Bitstream
8. Go to File -> Export -> Export Hardware
   1. Select Fixed platform type
   2. Select Include Bitstream
   3. Export to default folder
   4. Finish

# Hardware Setup

Required Hardware:

* Mainboard
* Enclustra XU1
* Mainboard Power Cable
* Xilinx JTAG Debugger
* Power Supply
* Signal Generator
* BNC to Mainboard Cable
* USB-to-Serial Converter
* Ethernet Cable

1. Ensure the XU1 is properly in place on the mainboard
2. Plug the JTAG Debugger into the Xilinx JTAG port of the mainboard
3. Connect your USB-to-Serial Converter to the TX, RX, and GND of the J13 connector 
4. Connect the two BNC-to-Mainboard Cables to and of J1-4, J6-8, J14-20
5. Use your DC power supply to deliver 12V to the mainboard

# A circuit board Description automatically generated

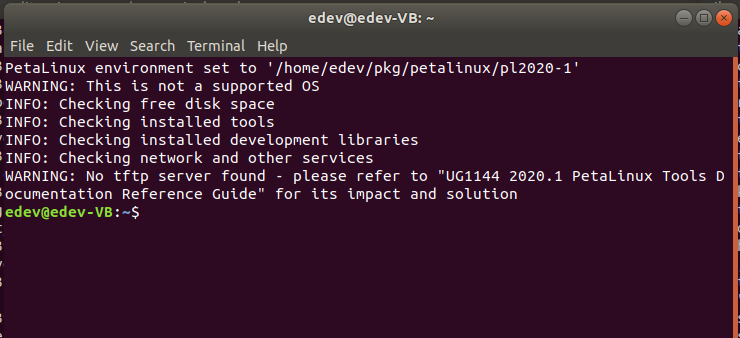
Note: The BCN-to-Mainboard is not necessary for this setup, it is used for feeding test signals to the board.

# Petalinux Setup

**Note:** Please use [the Xilinx Petalinux Tools Guide PDF](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug1144-petalinux-tools-reference-guide.pdf) in addition to this document if you run into issues, it has examples of expected outputs and instructions on how to use commands and how to build Petalinux. Please also reference [the Xilinx QEMU System Simulation Guide](https://www.xilinx.com/support/documentation/sw_manuals/petalinux2013_10/ug982-petalinux-system-simulation.pdf) for information on how to use the QEMU system simulator, where you can test your Petalinux image without hardware. QEMU runs entirely in a simulator through the terminal.

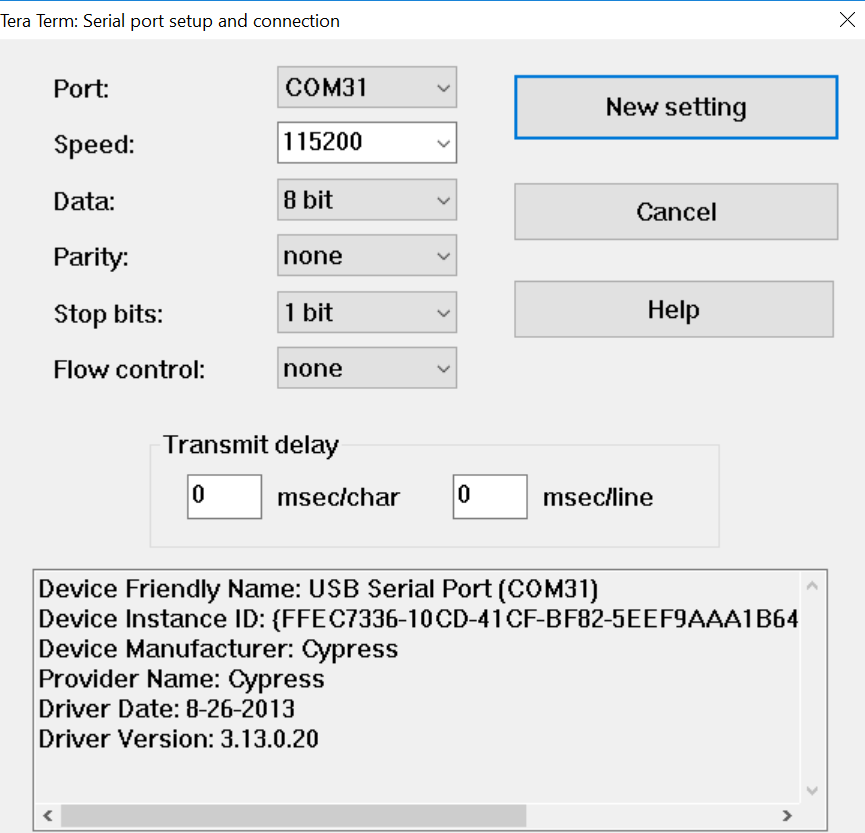
1. Install Ubuntu 18.04.04.
   1. If using a VM allocate 100GB of memory and 8GB of RAM.
2. Add Guest Additions to the VM
   1. https://www.tecmint.com/install-virtualbox-guest-additions-in-ubuntu/
3. To download all necessary packages, run:

sudo apt-get install -y iproute2 gcc g++ net-tools libncurses5-dev zlib1g:i386 libssl-dev flex bison libselinux1 xterm autoconf libtool texinfo zlib1g-dev gcc-multilib build-essential screen pax gawk python3 python3-pexpect python3-pip python3-git python3-jinja2 xz-utils debianutils iputils-ping libegl1-mesa libsdl1.2-dev pylint3 cpio vim

1. PetaLinux tools require that your host system /bin/sh is 'bash'. If you are using Ubuntu distribution and your /bin/sh is 'dash', change your default system shell /bin/sh with the sudo dpkg-reconfigure dash command.
2. Download the PetaLinux Tools - Installer - 2020.1 located here: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>
3. Run: mkdir -p ~/pkg/petalinux/pl2020-1
4. In the download location, run: ./petalinux-v2020.1-final-installer.run --dir ~/pkg/petalinux/pl2020-1
   1. You need to mark the file as executable first, using the chmod command run: chmod + x <filename>.run
5. Alter ~/.bashrc to append: source ~/pkg/petalinux/pl2020-1/settings.sh
   1. One way is to use nano editor by running: nano ~/.bashrc
6. Close your terminal and re-open it to see if the tools have installed properly. You should see the following: 
7. Create a shared folder between the guest and host machines
   1. In the VM menu, go to Devices>Shared Folders>Shared Folders Settings
   2. Add a new shared folder by clicking the small blue folder with a green plus on the far right
   3. Select the folder path and browse to the petalinux git repo folder in your host machine files
   4. Select “Auto-mount” and “Make Permanent”, leave “Mount point” blank
   5. Click OK
   6. Open a terminal in the VM and give your user permissions by running: sudo adduser [username] vboxsf
   7. Log out and log in, you should now have permissions
   8. The folder will be located in the /media directory
8. Run the following commands:
   1. mkdir ~/project
   2. cd ~/project
   3. petalinux-create -t project -n nuprism --template zynqMP
   4. cd ~/project/nuprism
   5. petalinux-config --get-hw-description <path-to-xsa-exported-from-vivado, should be in the /Vivado\_NuPRISM folder of the git repo (shared folder)>
      1. If you used the shared folder, the directory will be something like: /media/sf\_<name of shared folder>/<path-to-xsa-exported-from-vivado>
      2. Exit the menu that opens, to read more about this see the Petalinux 2020.1 guide
   6. petalinux-build
      1. This should generate files inside ~/project/nuprism/images/linux/
   7. petalinux-package --boot --u-boot --fpga images/linux/system.bit --format BIN
9. Be sure you have access to usb in the VM, so that you can access the SD card
10. Copy BOOT.BIN, image.ub, and boot.scr to the boot partition of your SD card
    1. BOOT.BIN, image.ub, and boot.scr will be located under ~/project/nuprism/images/linux/ in the guest machine
    2. To copy use: cp {BOOT.BIN,image.ub,boot.scr} /media/edev/boot
    3. To format your SD card if needed, complete the tutorial in Appendix H of the Petalinux 2020.1 user guide: <https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug1144-petalinux-tools-reference-guide.pdf>
11. Extract rootfs.tar.gz from /project/nuprism/images/linux to the root partition of your SD card using: tar xvf rootfs.tar.gz
    1. **Sudo tar -C /<destination directory, likely /media/<user>/root > -zxvf rootfs.tar.gz**
12. Eject the drives and place the SD card into the NuPRISM board

Cd ~

1. Open up a serial connection using your preferred tool. I used the CYUSBS232 USB-UART LP REFERENCE DESIGN KIT from TRIUMF. Follow the instructions on the inside of the package to setup the dongle.
   1. CYUSBS232 resource download <https://www.cypress.com/documentation/development-kitsboards/cyusbs232-usb-uart-lp-reference-design-kit#res585>
   2. Run Teraterm and open a serial connection with the following settings (change com port to whatever the CYUSB shows up under):



* 1. VIVADO hardware manager can be used to check fpga presence if desired

1. Power on the board, you should observe a series of boot messages ending with a login prompt. The user and password are both “root”. You should now be logged in on the SoC.



# MIDAS Host Setup

1. Alter ~/.bashrc to append:

export GIT\_EDITOR="emacs -nw"

export MIDASSYS=$HOME/packages/midas

export MIDAS\_EXPTAB=$HOME/online/exptab

export MIDAS\_EXPT\_NAME=e777

export PATH=$PATH:$MIDASSYS/bin

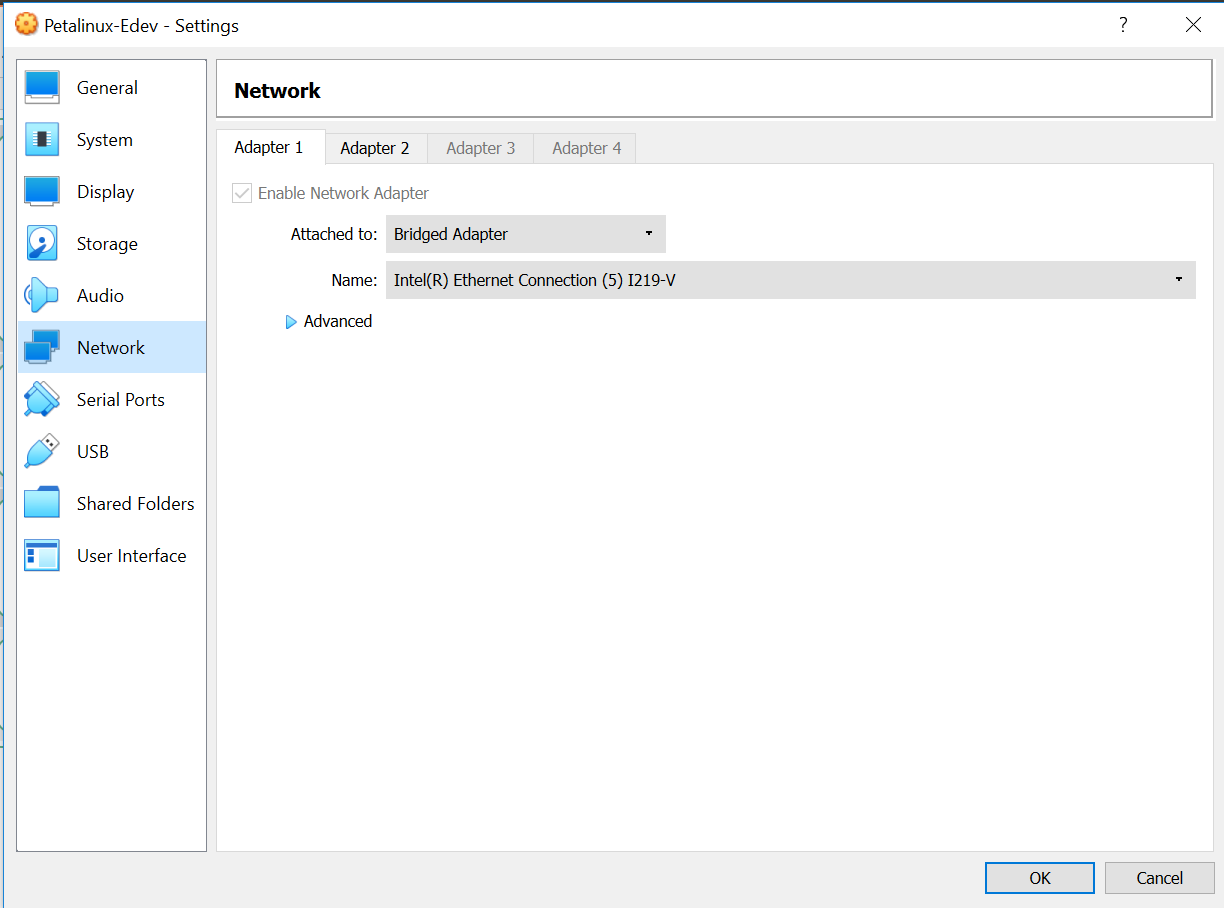
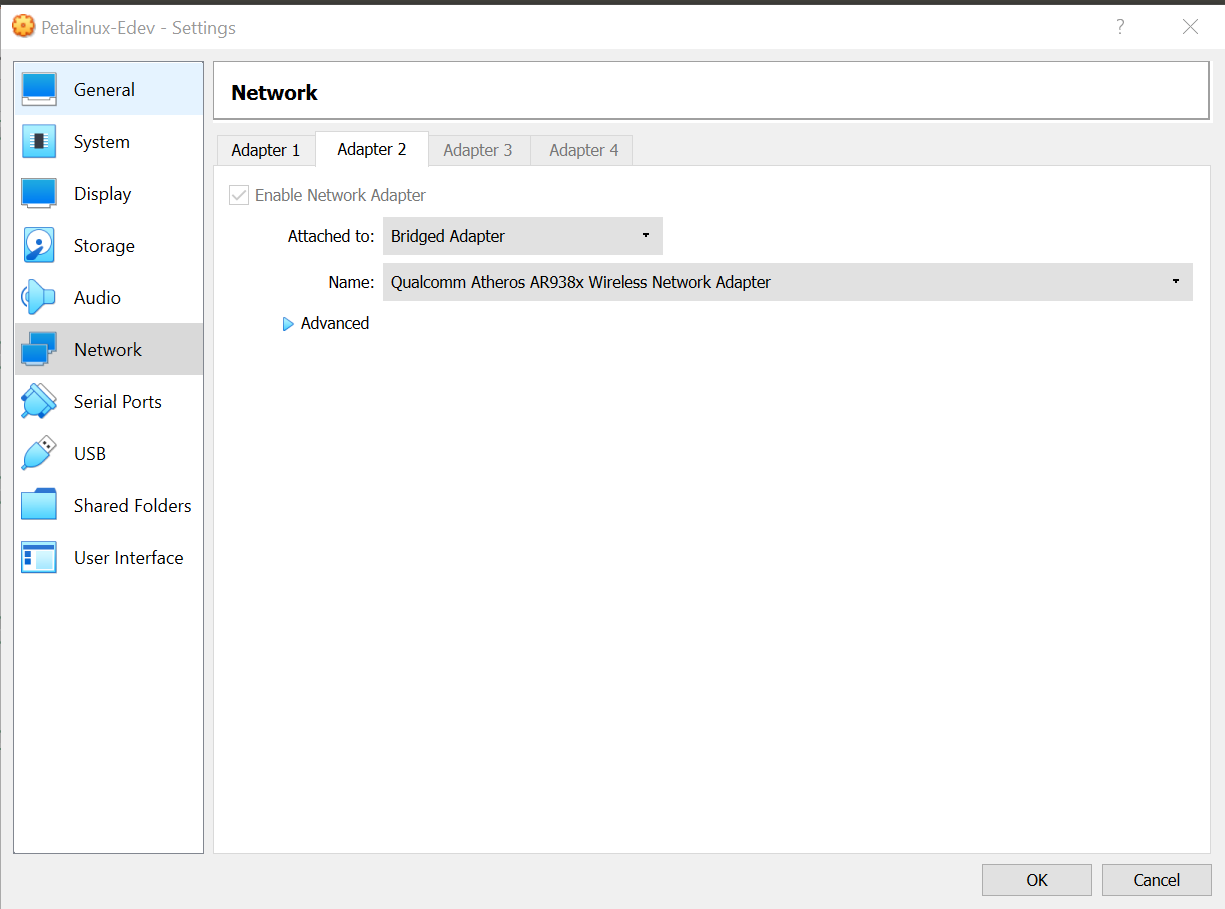
1. Run
   1. mkdir $HOME/packages
   2. cd $HOME/packages
   3. git clone https://bitbucket.org/tmidas/midas --recursive
   4. cd midas
   5. mkdir build
   6. cd build
   7. Sudo apt install libcurl4-openssl-dev
   8. sudo apt-get install build-essential cmake
   9. sudo cmake ../
   10. sudo make install
   11. cd $HOME/packages
   12. git clone https://bitbucket.org/tmidas/rootana
   13. cd rootana
   14. git checkout tags/rootana-2020-03-a
   15. make
   16. cd $HOME/packages
   17. git clone <https://github.com/linev/jsroot.git>
   18. mkdir $HOME/online
   19. cd $HOME/online
2. Alter ~/.bashrc to append:

ROOTANASYS=$HOME/packages/rootana

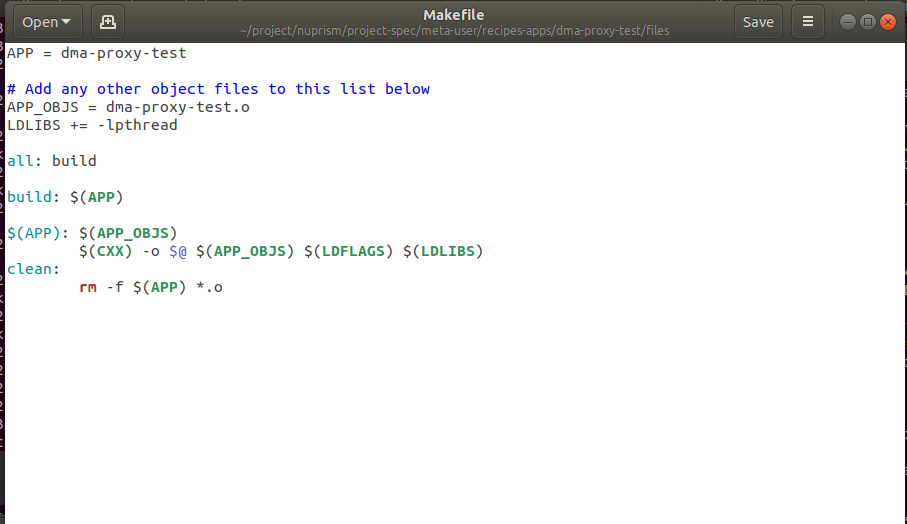
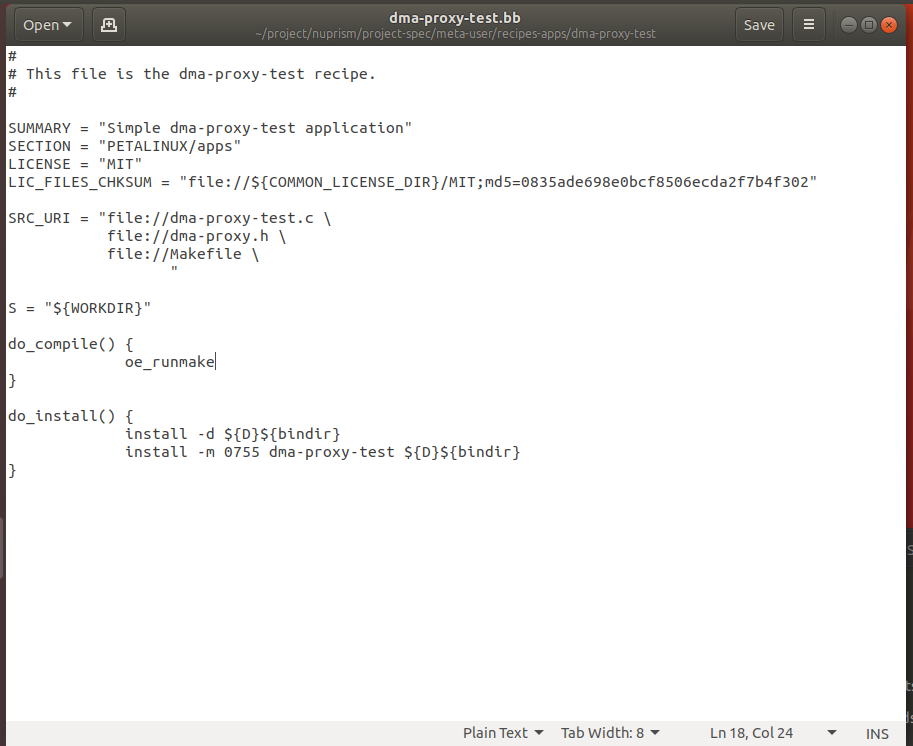
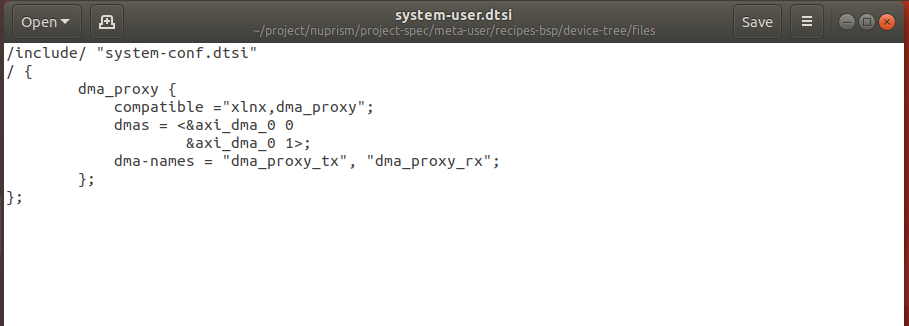
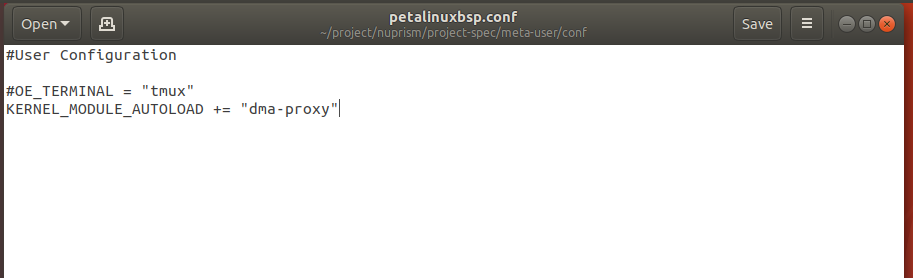
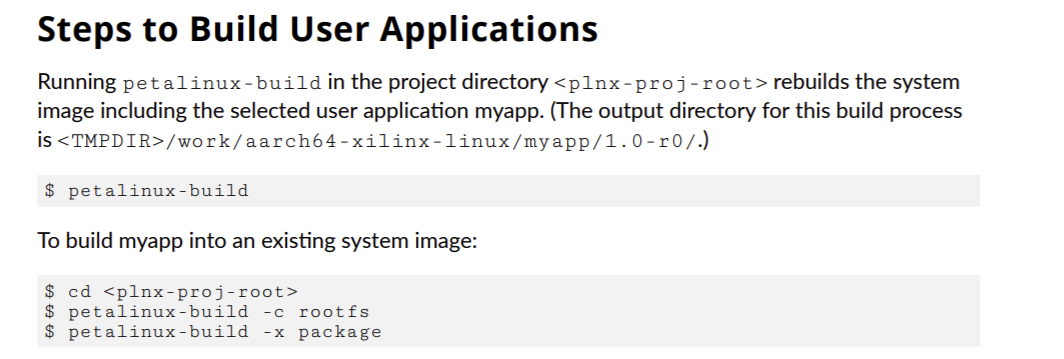
JSROOTSYS=$HOME/packages/jsroot

1. Create a file in ~/online/ called exptab by running: sudo touch exptab
   1. Run: sudo nano exptab to add the following contents
      1. e777 /home/<user>/online <user>
2. Append ~/.bashrc to include the MIDAS environment variables by running: sudo nano ~/.bashrc
   1. Add the lines:
      1. export MIDAS\_EXPTAB=$HOME/online/exptab
      2. export MIDASSYS=$HOME/packages/midas
3. Reboot terminal to have bashrc changes come into effect
4. Run the following 3 programs (3 separate terminals) by navigating into /midas/bin and running: ./<program name>
   1. odbedit
   2. mhttpd
   3. mlogger
5. You should now be able to open up a browser and navigate to <http://localhost:8080> (or whichever port number is provided by mhttpd) and access the MIDAS frontend panel.

# Adding MIDAS to Petalinux and Running the MIDAS Demo

1. For this to work, you must ensure the network adapter for the VM is a Bridged Adapter.
   1. Use [this link](https://www.linuxbabe.com/virtualbox/a-pretty-good-introduction-to-virtualbox-bridged-networking-mode#:~:text=Change%20Networking%20Mode%20to%20Bridged&text=Click%20the%20Network%20tab%20on,button%20to%20save%20your%20settings.) for steps, or…
   2. Shutdown the VM
   3. In the VM settings, find the Network settings
   4. For Adapter 1, attach it to a “Bridged Adapter” and set Name to whatever the LAN Ethernet connection will be on the Host machine. Whichever Host machine adapter connects to the network with the board. 
   5. For adapter 2, attach it to a “Bridged Adapter” and set Name to whichever connection will provide internet, wireless or wired, depending on your host machine setup. 
   6. This will “bridge” your VM and allow it to appear as a separate device on the network, so that we can talk to it with other devices
2. In the Petalinux project root folder on the VM run:
   1. petalinux-create -t apps --template c++ --name midas --enable
   2. petalinux-create -t apps --template c++ --name mfe –enable
   3. This will create two new c++ apps within this petalinux project, “midas” and “mfe”
3. Copy the midas and mfe folders from inside the petalinux\_midas folder from the sf\_rev0-2020.1-petalinux git repo (shared folder) into the nuprism/project-spec/meta-user/recipes-apps folder
   1. This will replace the automatically generated c++ app templates with the proper midas and mfe apps
4. Build the Petalinux project again and copy the files over to your SD card as described in the Petalinux setup section (steps 11f through 14)
5. On your host run in this order:
   1. ./odbedit
   2. ./mhttpd
   3. ./mlogger
   4. ./mserver
6. Add the ip of both the MIDAS host and the petalinux board to the list of allowed RPC hosts for MIDAS
   1. Connect to the localhost:8080 midas server in a browser on the VM
   2. Navigate to the ODB tab
   3. Navigate to /Experiment/Security/RPC hosts
   4. Add the ip of the VM and the petalinux board to the allowed hosts list
7. Once the firmware has booted, run (on the mainboard):
   1. mfe -h <host-ip-address>:<mserver-address> -e e777
      1. example: mfe -h 192.168.1.80:1175 -e e777
      2. you may need to add the mainboards ip address to a list of trusted ip addresses in the MIDAS control panel
8. Start a run by pressing the Start button on the MIDAS frontend panel start page
9. Stop the run and check to see that data has been acquired
10. Profit.

**DMA Setup**

1. Set the .xsa path to the proper vivado project which includes DMA
   1. petalinux-config --get-hw-description <path-to-xsa-exported-from-vivado, should be in the /Vivado\_NuPRISM\_AddingDMA folder of the git repo (shared folder)>
      1. If you used the shared folder, the directory will be something like: /media/sf\_<name of shared folder>/<path-to-xsa-exported-from-vivado>
      2. Exit the menu that opens, to read more about this see the Petalinux 2020.1 guide
2. Create an app called “dma-proxy-test”
   1. run: petalinux-create -t apps --template c –name dma-proxy-test --enable
   2. Copy “dma-proxy-test.c” and “dma-proxy.h” over into the files folder
   3. Add “LDLIBS += –lpthread” to the make file so that the compiler can find the pthread library: 
   4. Add “file://dma-proxy.h \” to the .bb file in the upper application folder: 
3. Create a module called “dma-proxy”
   1. Run: petalinux-create –t modules –name dma-proxy --enable
   2. Copy “dma-proxy.c” and “dma-proxy.h” over into the files folder
   3. Add “dma-proxy.h” to the .bb file in the upper module folder: 
   4. Makefile: 
4. Navigate into /project/nuprsim/project-spec/meta-user/recipes-bsp/device-tree/files and modify the “system-user.dtsi” file to include the new dma-proxy-rx/tx devices in the petalinux device tree: 
5. Navigate to /project/nuprism/project-spec/meta-user/conf and add “KERNEL\_MODULE\_AUTOLOAD += “dma-proxy” to the “petalinuxbsp.conf” file to automatically load the dma-proxy module on boot: 
6. Rebuild the petalinux project with: petalinux-build
   1. Or build the specific app using “petalinux-build –c dma-proxy –x compile” and “petalinux-build –c dma-proxy-test –x compile” as follows: 
   2. petalinux-package --boot --u-boot --fpga images/linux/system.bit --format BIN
7. Copy BOOT.BIN, image.ub, and boot.scr to the boot partition of your SD card
   1. BOOT.BIN, image.ub, and boot.scr will be located under ~/project/nuprism/images/linux/ in the guest machine
   2. To format your SD card if needed, complete the tutorial in Appendix H of the Petalinux 2020.1 user guide: <https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug1144-petalinux-tools-reference-guide.pdf>
8. Extract rootfs.tar.gz from /project/nuprism/images/linux to the root partition of your SD card using: tar xvf rootfs.tar.gz
   1. Sudo tar -C /<destination directory, likely /media/<user>/root > -zxvf rootfs.tar.gz

**Quick Reference:**

1. Build sequence:
   1. petalinux-config --get-hw-description <path-to-xsa-exported-from-vivado, should be in the /Vivado\_NuPRISM folder of the git repo (shared folder)>
      1. If you used the shared folder, the directory will be something like: /media/sf\_<name of shared folder>/<path-to-xsa-exported-from-vivado>
      2. Exit the menu that opens (press e), to read more about this see the Petalinux 2020.1 guide
   2. petalinux-build
      1. This should generate files inside ~/project/nuprism/images/linux/
   3. petalinux-package --boot --u-boot --fpga images/linux/system.bit --format BIN --force
   4. cd into /images/linux
   5. be sure sd card is on usb
   6. Extract file system with: sudo tar -C /<destination directory, likely /media/<user>/root > -zxvf rootfs.tar.gz
   7. Copy boot files with: cp {BOOT.BIN,image.ub,boot.scr} /media/edev/boot
   8. Eject root drive