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

**Author(s):** R. Payne

**Release:** 1.0

**Release Date:** 21/08/2020

**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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 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://edev-group.triumf.ca/fw/exp/nuprism/xu1/rev0-2020.1-petalinux>
3. Open the project using Vivado 2020.1
4. Run synthesis
5. If you want to set up the Integrated Logic Analyzer (ILA)
   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 border
   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

Insert pic of setup

# Petalinux Setup

1. Install Ubuntu 18.04.04.
   1. If using a VM allocate 100GB of memory and 8GB of RAM.
2. 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', consult your system administrator to 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
5. Alter ~/.bashrc to append: source ~/pkg/petalinux/pl2020-1/settings.sh
6. Close your terminal and re-open it to see if the tools have installed properly.
7. 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>
      1. 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
8. Copy BOOT.BIN, image.ub, and boot.scr to the root partition of your SD card
   1. To format your SD card, 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>
9. Extract rootfs.tar.gz to the boot partition of you SD card using: tar xvf rootfs.tar.gz
10. Place the SD card into the NuPRISM board, open up a serial connection, and power on the board

# 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 cmake ../
   8. make install
   9. cd $HOME/packages
   10. git clone https://bitbucket.org/tmidas/rootana
   11. cd rootana
   12. git checkout tags/rootana-2020-03-a
   13. make
   14. cd $HOME/packages
   15. git clone <https://github.com/linev/jsroot.git>
   16. mkdir $HOME/online
   17. cd $HOME/online
2. Alter ~/.bashrc to append:

ROOTANASYS=$HOME/packages/rootana

JSROOTSYS=$HOME/packages/jsroot

1. Create a file in ~/online/ called exptab with the following contents
   1. e777 /home/<account>/online <account>
2. Run the following 3 programs:
   1. odbedit
   2. mhttpd
   3. mlogger
3. 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. In the Petalinux project run:
   1. petalinux-create -t apps --template c++ --name midas --enable
   2. petalinux-create -t apps --template c++ --name mfe –enable
2. Copy the contents of the ds\_midas folder into the relative projects generated in project-spec/meta-user/recipe-apps/
3. Run:
   1. petalinux-build -c midas
   2. petalinux-build -c mfe
4. Build the Petalinux project and copy the files over to your SD card as described in the Petalinux setup section (steps 10f,g – 12)
5. On you host run
   1. odbedit
   2. mhttpd
   3. mlogger
   4. mserver
6. 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
7. Start a run by pressing the Start button on the frontend panel start page
8. Stop the run and check to see that data has been acquired