Xilinx XUP Blackboard Linux Configuration Steps

Jason Losh, revised September 18, 2023

This document provides steps for making and testing the bootable Linux v6.1 LTS image for the Xilinx XUP Blackboard that has been customized for use in the SoC course.

The complete process to compile the kernel and copy all the needed packages takes about 6 hours to complete. The provided image includes the tool chain for compiling programs in C and Rust and building Linux kernel modules. It is also pre-configured to support DHCP using the provided ASIX USB-to-Ethernet adapter and with an enabled SSH server "out of the box". Special thanks goes to Rick at Real Digital for all his help in creating these images.

To save time, we will just make a clone of the current CpE golden image on a true 8 GB micro SD card using one of these three methods which will take less than 10 minutes:

Clone the disk image on Ubuntu using the Disk Manager

1. Download the golden image for the board (blackboard_linux_v6.1_cpe_20230918.zip)
2. Extract the image file (blackboard_linux_v6.1_cpe_20230918.img)
3. Go to to the Show Applications Icon "3x3 dots" > Select Utilities > Select Disks
4. Insert a blank true 8 GB or larger micro SD card into a USB or SD adapter
5. Look for the 8 GB SD Card Reader in the Left Pane
6. Select … at the top > Restore Disk Image
7. In Image to Restore, specify the path to blackboard_linux_v6.1_cpe_20230918.img

Clone the disk image on a Linux PC using the command line

1. Download the golden image for the board (blackboard_linux_v6.1_cpe_20230918.zip)
2. Extract the image file (blackboard_linux_v6.1_cpe_20230918.img)
3. Run the following command to list all the block devices (except the loop devices):
   lsblk -e 7
4. Insert a blank true 8 GB or larger microSD card
5. Run the lsblk command again, noting the new device added
   (on a Ubuntu machine, the device will likely be mmcblk0)
6. Issue the dd command to copy the image to the micro SD card:
   (assuming the micro SD care is mmcblk0)
   sudo dd if=*path_to_image*/blackboard_linux_v6.1_cpe_20230918.img of=/dev/mmcblk0 bs=4M status=progress

Clone the disk image on a Windows PC using Win32 Disk Manager

1. Download the golden image for the board (blackboard_linux_v6.1_cpe_20230918.zip)
2. Extract the image file (blackboard_linux_v6.1_cpe_20230918.img)
3. Insert a blank true 8 GB or larger micro SD card into a USB or SD adapter
4. Specify the path to the micro SD card and the path to source image
   blackboard_linux_v6.1_cpe_20230918.img

Boot with the new image and write a test application

1. Connect a USB A to USB microB cable from the PC to the PROG/UART jack
2. Connect a high current USB microB power supply to EXTP jack
3. Set the jumper to use the EXTP power source
   Note: It may be possible to power the board completely throught the PROG/UART jack when writing FPGA-only apps, but the high current supply is required when using the processor system
4. Connect an Ethernet adapter to the USB A jack on the board
5. Connect an Ethernet cable from the adapter to a DHCP server connected to the Internet (this can be to the Ethernet jack of a PC connected to WiFi with connect sharing enabled or one of the smart switches at each desk in the ERB 125 or bench in ERB 126
6. Insert the microSD card
7. Turn on power
8. Issue the following command to see the diagnostic messages:
   (the tty device at the end is the name of the USB device behind the PROG/UART jack)
   sudo dmesg | grep tty
9. Use Putty to connect to the serial device above, selecting a 115200 baud rate
10. Press the PS_SRST button on the SoC board
11. Watch for the boot process to complete in Putty
12. Enter a username of xilinx and a password of xilinx
13. Verify the kernel version with:
    uname -r
14. Create a Documents directory in the home directory and cd to that directory using:
    mkdir Documents
    cd Documents
15. Create a C directory and enter it:
    mkdir C
    cd C
16. Find the DHCP address assigned to the board using:
    ip a
17. Use FileZilla (or similar SFTP client) to connect the the board at the address above.
18. Transfer the demo source files to the C directory or use nano editor to write a C file.
19. Compile the source with the GCC v. 12.2.0 compiler and execute as follows:
    gcc -o *program_name program_name*.c
    ./program_name