

1. Overview

Based on Linux on LiteX with a 64-bit RocketChip CPU on Github, build an SoC using liteX and RocketChip on Xilinx FPGA board Arty A7-100T and run 64-bit Linux.

1.1. Environment

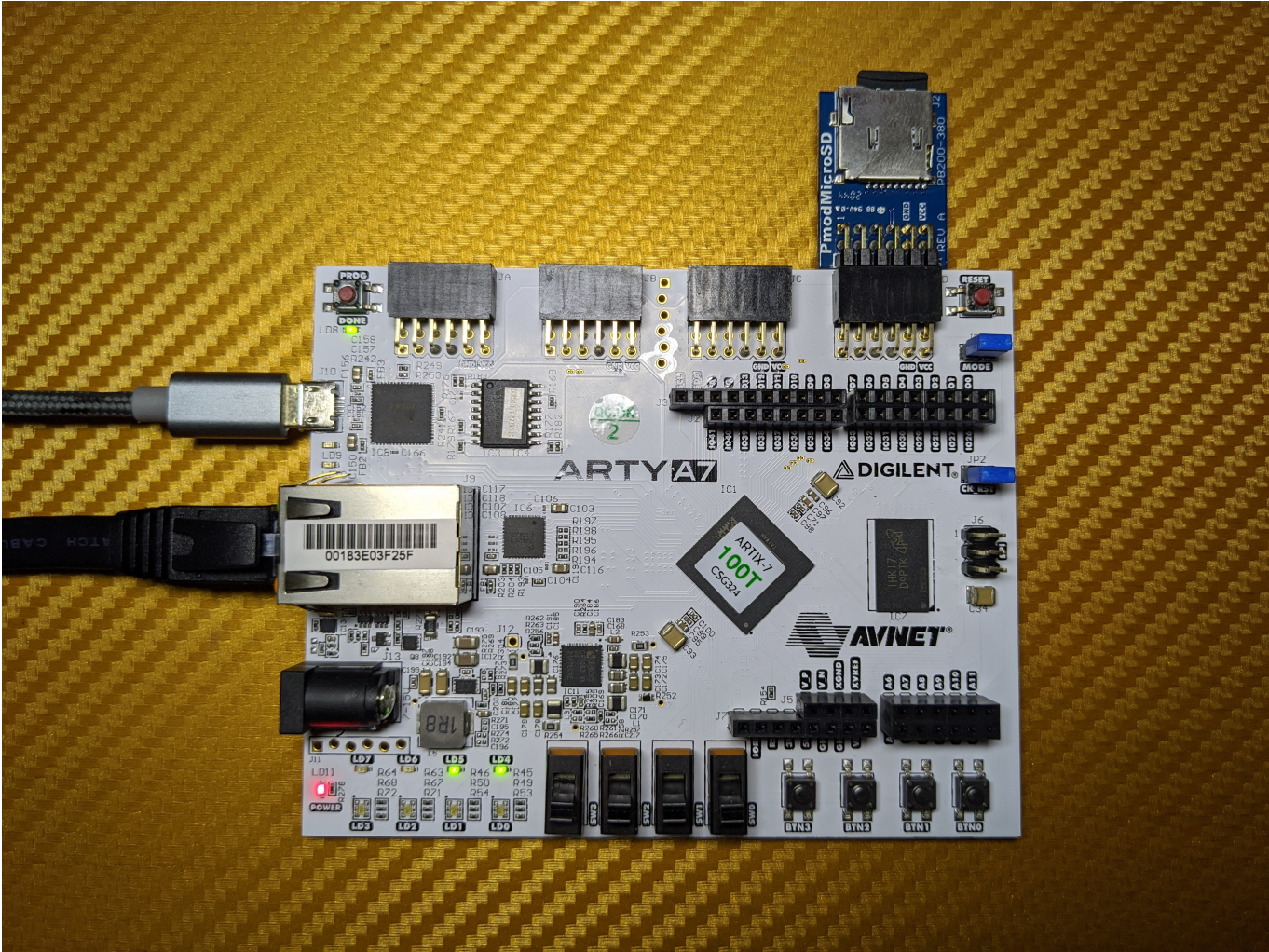
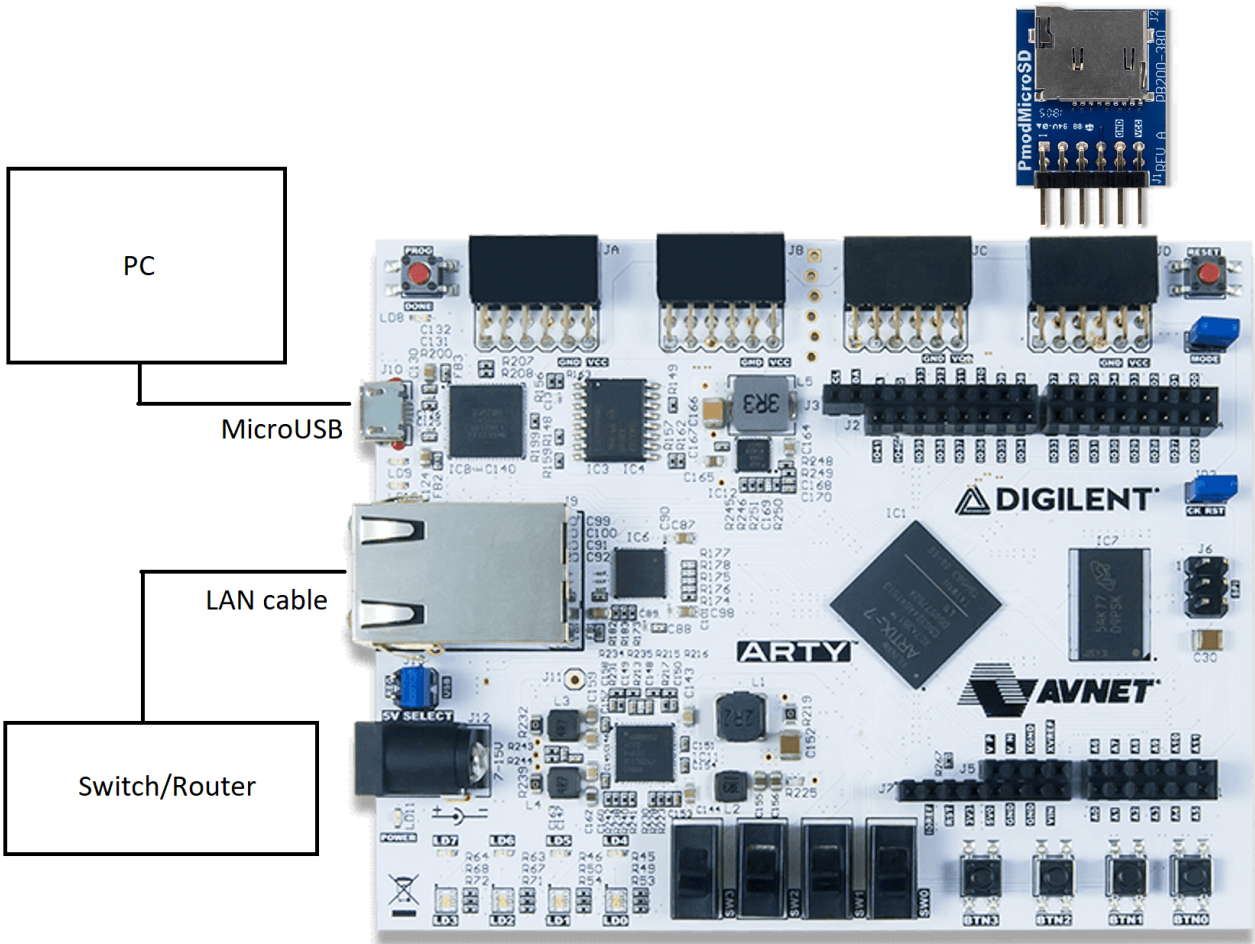
- The OS uses Ubuntu 20.04.5
- The FPGA board uses Arty A7-100T made by Xilinx
- Uses a Micro SD card and PMOD Micro SD Card Reader 410-380

1.2. Procedure

- Step 1: [Connect hardware](#).
- Step 2: [Install prerequisites](#)
- Step 3: [Build Gateware](#)
- Step 4: [Build Software](#)
- Step 5: [Starting Linux on Litex+Rocket](#)

2. Hardware connection

- Insert PmodMicroSD module into PMOD connector **JD** of Arty A7-100T. Make sure VCC and GND pin are connected correctly.
- Connect Arty A7-100T (J9) to Switch/Router using a LAN cable.
- Connect Arty A7-100T (J10) to PC using a MicroUSB cable (power, console and programming).



3. Install prerequisites

3.1. Install support packages

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install openocd fakeroot verilator python3 meson gtkterm gawk texinfo
git python3-pip bison device-tree-compiler autoconf automake autotools-dev curl
python3 libmpc-dev libmpfr-dev libgmp-dev build-essential flex gperf libtool
patchutils bc zlib1g-dev libexpat-dev ninja-build
```

3.2. Install the LiteX development environment

```
$ cd ~
$ mkdir litex
$ cd litex
$ wget https://raw.githubusercontent.com/enjoy-digital/litex/master/litex_setup.py
$ chmod 777 litex_setup.py
$ python3 ./litex_setup.py --config=full --init --install --user
```

Add to PATH

```
$ echo 'export PATH=$PATH:~/.local/bin' >> ~/.bashrc
```

3.3. Installation of cross-compiler toolchain for 64bit RISC-V

```
$ cd ~
$ git clone --recursive https://github.com/riscv/riscv-gnu-toolchain
$ pushd riscv-gnu-toolchain
$ ./configure --prefix=$HOME/RISCV --enable-multilib
$ make newlib linux -j8
$ popd
```

Add to PATH

```
$ export PATH=$PATH:$HOME/RISCV/bin
$ echo 'export PATH=$PATH:$HOME/RISCV/bin' >> ~/.bashrc
```

3.4. Install Vivado

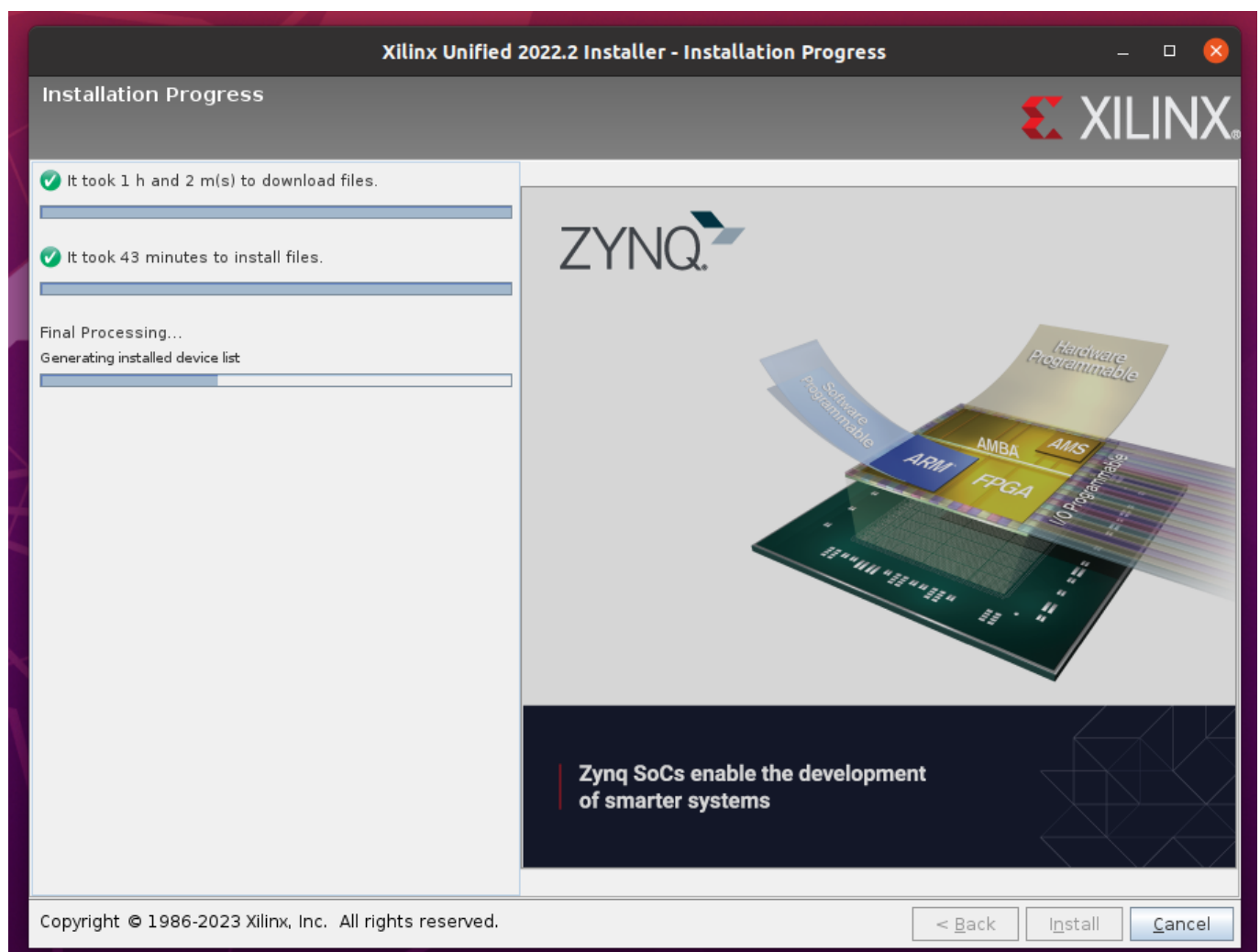
Install support package

```
$ sudo apt install libtinfo-dev
$ sudo ln -s /lib/x86_64-linux-gnu/libtinfo.so.6 /lib/x86_64-linux-
gnu/libtinfo.so.5
```

Download [Xilinx_Unified_2022.2_1014_8888_Lin64.bin](#), and run it:

```
$ sudo chmod +x ./Xilinx_Unified_2022.2_1014_8888_Lin64.bin
$ ./Xilinx_Unified_2022.2_1014_8888_Lin64.bin
```

Install Vivado in the `/tools` directory with the installer.



Add Vivado to PATH

```
$ echo 'export PATH=$PATH:/tools/Xilinx/Vivado/2022.2/bin' >> ~/.bashrc
```

4. Build Gateware

```
$ cd ~/lites
$ lites-boards/lites_boards/targets/digilent_arty.py --build --cpu-type rocket --
cpu-variant linuxd --sys-clk-freq 50e6 --with-ethernet --with-sdcard --variant=a7-
100
```

The build creates a bitstream file in:

```
$HOME/lites/build/digilent_arty/gateware/digilent_arty.bit
```

Each time a board is programmed (directly from the builder if the `--load` option is given, or directly using `openocd`), and each time a programmed board is reset, the bare-metal firmware (BIOS) included in the bitstream will initialize the CPU, RAM, and peripherals, and attempt to load `boot.bin`, first from the microSD card (first partition, expected to be formatted as msdos/fat), then via TFTP over the network (expecting a 192.168.1.0/24 network, with the LiteX+Rocket system using IP address 192.168.1.50, and attempting to download `boot.bin` via TFTP from a server at 192.168.1.100).

Notes

We could also build the gateware that can load the LiteX boot image from microSD card (no netboot):

```
$ cd ~/lites
$ lites-boards/lites_boards/targets/digilent_arty.py --build --cpu-type rocket --
cpu-variant linuxd --sys-clk-freq 50e6 --with-sdcard --variant=a7-100
```

or from TFTP server only (no sdcardboot):

```
$ cd ~/lites
$ lites-boards/lites_boards/targets/digilent_arty.py --build --cpu-type rocket --
cpu-variant linuxd --sys-clk-freq 50e6 --with-ethernet --variant=a7-100
```

5. Build Software

5.1. Clone the linux-on-lites-rocket repository

```
$ cd ~/lites
$ git clone https://github.com/lites-hub/linux-on-lites-rocket
```

5.2. Building a BusyBox

```
$ cd ~/litex/linux-on-litex-rocket
$ curl https://busybox.net/downloads/busybox-1.33.2.tar.bz2 | tar xvj -
$ sudo chmod 777 -R busybox-1.33.2
$ cp conf/busybox-1.33.2-rv64gc.config busybox-1.33.2/.config
$ cd busybox-1.33.2
$ make CROSS_COMPILE=riscv64-unknown-linux-gnu-
$ cd ..
```

5.3. Creating a Kernel Root RAM Filesystem

```
$ cd ~/litex/linux-on-litex-rocket
$ mkdir initramfs
$ pushd initramfs
$ mkdir -p bin sbin lib etc dev home proc sys tmp mnt nfs root usr/bin usr/sbin
usr/lib
$ cp ../busybox-1.33.2/busybox bin/
$ ln -s bin/busybox ./init
$ cat > etc/inittab <<- "EOT"
::sysinit:/bin/busybox mount -t proc proc /proc
::sysinit:/bin/busybox mount -t devtmpfs devtmpfs /dev
::sysinit:/bin/busybox mount -t tmpfs tmpfs /tmp
::sysinit:/bin/busybox mount -t sysfs sysfs /sys
::sysinit:/bin/busybox --install -s
/dev/console::sysinit:-/bin/ash
EOT
$ fakeroot <<- "EOT"
find . | cpio -H newc -o > ../initramfs.cpio
EOT
$ popd
```

5.4. Building the Kernel

```
$ cd ~/litex/linux-on-litex-rocket
$ git clone https://github.com/litex-hub/linux.git
$ cp initramfs.cpio linux/
$ pushd linux
$ git checkout litex-rebase
```

Enable CONFIG_RISCV_SBI_V01 in kernel config

```
$ echo "CONFIG_RISCV_SBI_V01=y" >> ./arch/riscv/configs/litex_rocket_defconfig
```

Build the kernel

```
$ make clean
$ make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- litex_rocket_defconfig
litex_rocket_initramfs.config
$ make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu-
$ popd
```

5.5. Update DTS file

Find UART base address:

```
$ grep "CSR_UART_BASE"
"$HOME/litex/build/digilent_arty/software/include/generated/csr.h"
```

e.g. The UART base address is `0x12006800` as below result (CSR_BASE is `0x12000000`):

```
#define CSR_UART_BASE (CSR_BASE + 0x6800L)
```

Change the UART base address in the DTS file (if needed):

(Current UART base address in DTS file is `0x12004000`. It will be changed to `0x12006800`)

```
$ cd ~/litex/linux-on-litex-rocket
$ sed -i 's/12004000/12006800/g' conf/arty.dts
```

Then, `arty.dts` should look something like this:

```
...
    bootargs = "console=liteuart earlycon=liteuart,0x12006800
swiotlb=noforce";
...
    liteuart0: serial@0x12006800 {
        compatible = "litex,liteuart";
        reg = <0x12006800 0x100>;
        interrupt-parent = <&L1>;
        interrupts = <1>;
    };
...
```

For reference:

Option in building gateway	UART base address
<code>--with-ethernet</code>	<code>0x12004000</code>

Option in building gateway	UART base address
--with-sdcard	0x12005800
--with-sdcard --with-pmod-gpio	0x12006000
--with-ethernet --with-sdcard	0x12006800

5.6. Building a LiteX boot image

```
$ cd ~/lites/lite-on-lites-rocket
$ git clone https://github.com/riscv/riscv-pk
$ mkdir riscv-pk/build
$ pushd riscv-pk/build
$ make clean
$ rm -f *
$ ../configure --host=riscv64-unknown-linux-gnu \
               --with-arch=rv64imac \
               --with-payload=../../linux/vmlinux \
               --with-dts=../../conf/artty.dts \
               --enable-logo
$ make bbl
$ riscv64-unknown-linux-gnu-objcopy -O binary bbl ../../boot.bin
$ popd
```

Output file (LiteX boot image):

```
~/lites/lite-on-lites-rocket/boot.bin
```

6. Starting Linux on LiteX+Rocket

6.1 Prepare the LiteX boot image

6.1.1. OPTION 1: Load the LiteX boot image using MicroSD Card

1. Create a partition table for the microSD card (assuming `/dev/sdX` is your microSD card):

```
$ sudo fdisk /dev/sdX
: o // Create a new empty DOS partition table
: n // Create a new partition
: p // Create a primary partition
: 1 // Specify partition number
: (Enter) // Specify first sector (default)
: +1GiB // Specify last sector
: t // Specify the system ID of the region
: 6 // Set system ID to FAT16
```

```
: p // Show partition table
: w // Write partition table
```

which should look something like this:

```
$ sudo fdisk /dev/sdX

Welcome to fdisk (util-linux 2.35.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Command (m for help): p
Disk /dev/sdX: 29.74 GiB, 31914983424 bytes, 62333952 sectors
Disk model: SD/MMC
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x67f480f9

Device      Boot   Start      End  Sectors  Size Id Type
/dev/sdX1                2048   2099199   2097152    1G  6 FAT16
...
```

2. Format the partition, mount it, and copy `boot.bin` to it:

```
$ cd ~/litex/linux-on-litex-rocket
$ sudo su
# mkdosfs /dev/sdX1
# mount /dev/sdX1 /mnt
# cp boot.bin /mnt
# umount /mnt
# exit
```

6.1.2. OPTION 2: Load the LiteX boot image using TFTP server

Connect the board's Ethernet port to a switch/router port that places it into the same Layer-2 broadcast domain (i.e., on the same LAN) as the machine acting as your TFTP server. Ensure that `boot.bin` is available in the TFTP directory, and that the TFTP server (or socket, if using systemd) is started:

```
$ cd ~/litex/linux-on-litex-rocket
$ sudo cp boot.bin /var/lib/tftpboot/
$ sudo systemctl start tftp.socket
```

Also make sure your TFTP server responds to requests sent to the IP address `192.168.1.100`.

Ensure the microSD slot is empty (as it takes precedence over Ethernet in the hardcoded BIOS boot order).

6.2. Connecting to the system's console

To connect to the system's console, use the `screen` utility (assuming `/dev/ttyUSB1` is used, below):

```
$ screen /dev/ttyUSB1 115200
```

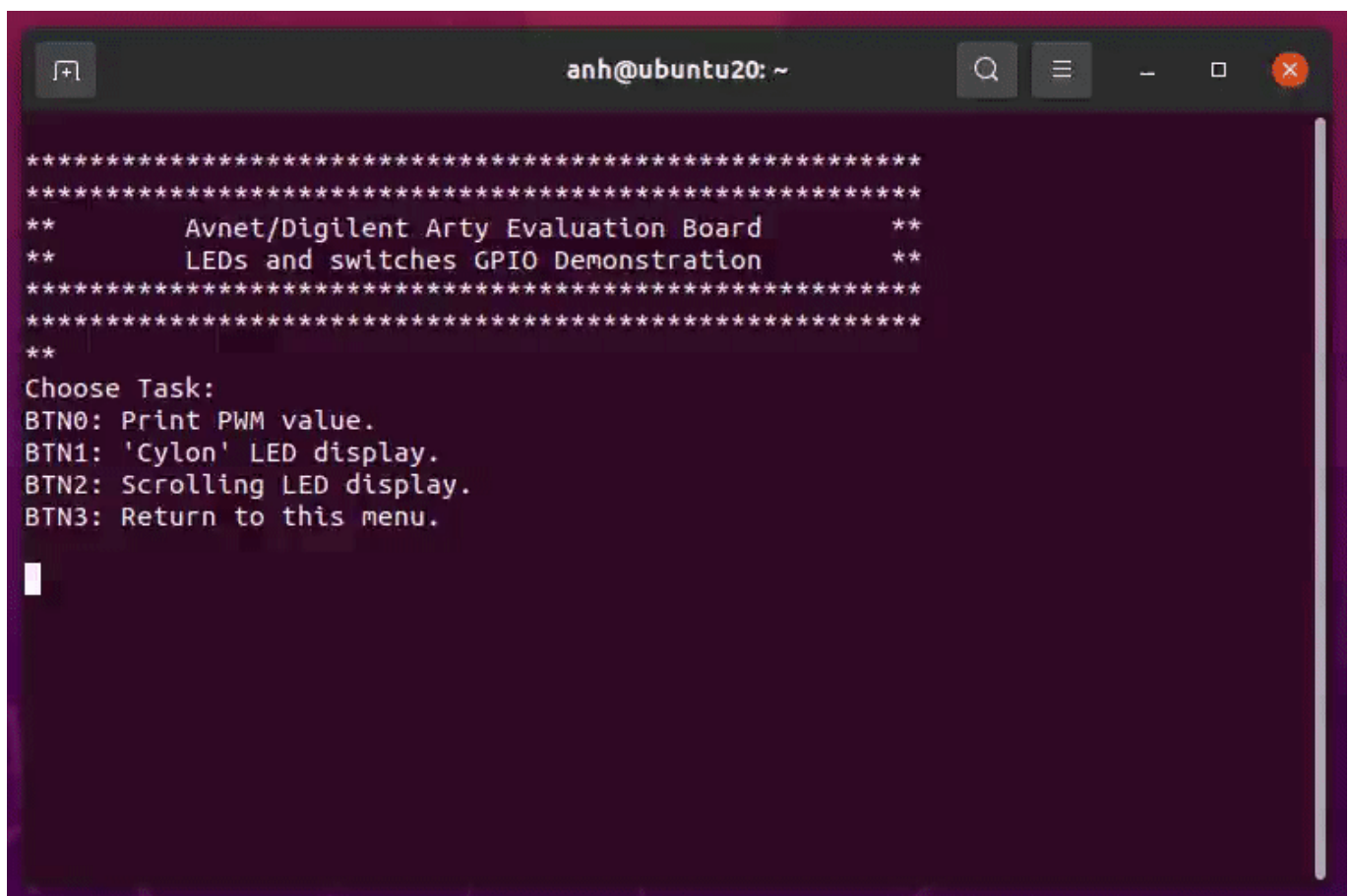
6.3. FPGA programming and booting

Program the board (with the Gateware built in [Build Gateware](#)):

```
$ cd ~/lites
$ litex-boards/lites_boards/targets/digilent_arty.py --load
```

The `boot.bin` blob will be copied in from the microSD card or over TFTP, and BBL, Linux, and BusyBox's shell will be started on the console, in that order.

6.3.1. SD Card:



```

[1m      _ _ _ _ _ _ _ _ _ _ [0m
[1m      / /  ( ) / _ _ | | / _ [0m
[1m      / / _ / / _ / - _ > < [0m
[1m      / _ _ / _ \ _ \ _ / | _ [0m
  
```

```
□[1m   Build your hardware, easily!□[0m
```

```
(c) Copyright 2012-2022 Enjoy-Digital
```

```
(c) Copyright 2007-2015 M-Labs
```

```
BIOS built on Mar 11 2023 11:49:46
```

```
BIOS CRC passed (c7f0f0fc)
```

```
LiteX git sha1: e1f847ef
```

```
----- □[1mSoC□[0m -----
```

```
□[1mCPU□[0m:      RocketRV64[imac] @ 50MHz
```

```
□[1mBUS□[0m:      WISHBONE 32-bit @ 4GiB
```

```
□[1mCSR□[0m:      32-bit data
```

```
□[1mROM□[0m:      128.0KiB
```

```
□[1mSRAM□[0m:      8.0KiB
```

```
□[1mSDRAM□[0m:    256.0MiB 16-bit @ 400MT/s (CL-7 CWL-5)
```

```
□[1mMAIN-RAM□[0m: 256.0MiB
```

```
----- □[1mInitialization□[0m -----
```

```
Ethernet init...
```

```
Initializing SDRAM @0x80000000...
```

```
Switching SDRAM to software control.
```

```
Read leveling:
```

```
m0, b00: |00000000000000000000000000000000| delays: -
```

```
m0, b01: |01111111111111111111111111111110| delays: 15+-14
```

```
m0, b02: |00000000000000000000000000000000| delays: -
```

```
m0, b03: |00000000000000000000000000000000| delays: -
```

```
m0, b04: |00000000000000000000000000000000| delays: -
```

```
m0, b05: |00000000000000000000000000000000| delays: -
```

```
m0, b06: |00000000000000000000000000000000| delays: -
```

```
m0, b07: |00000000000000000000000000000000| delays: -
```

```
best: m0, b01 delays: 15+-14
```

```
m1, b00: |00000000000000000000000000000000| delays: -
```

```
m1, b01: |01111111111111111111111111111110| delays: 15+-14
```

```
m1, b02: |00000000000000000000000000000000| delays: -
```

```
m1, b03: |00000000000000000000000000000000| delays: -
```

```
m1, b04: |00000000000000000000000000000000| delays: -
```

```
m1, b05: |00000000000000000000000000000000| delays: -
```

```
m1, b06: |00000000000000000000000000000000| delays: -
```

```
m1, b07: |00000000000000000000000000000000| delays: -
```

```
best: m1, b01 delays: 15+-14
```

```
Switching SDRAM to hardware control.
```

```
Memtest at 0x80000000 (2.0MiB)...
```

```
Write: 0x80000000-0x80000000 0B
```

```
Write: 0x80000000-0x80020000 128.0KiB
```

```
Write: 0x80000000-0x80040000 256.0KiB
```

```
Write: 0x80000000-0x80060000 384.0KiB
```

```
Write: 0x80000000-0x80080000 512.0KiB
```

```
Write: 0x80000000-0x800a0000 640.0KiB
```

```
Write: 0x80000000-0x800c0000 768.0KiB
```

```
Write: 0x80000000-0x800e0000 896.0KiB
```

```
Write: 0x80000000-0x80100000 1.0MiB
```

```
Write: 0x80000000-0x80120000 1.1MiB
```

```
Read: 0x80000000-0x80000000 0B
Read: 0x80000000-0x80020000 128.0KiB
Read: 0x80000000-0x80040000 256.0KiB
Read: 0x80000000-0x80060000 384.0KiB
Read: 0x80000000-0x80080000 512.0KiB
Read: 0x80000000-0x800a0000 640.0KiB
Read: 0x80000000-0x800c0000 768.0KiB
Read: 0x80000000-0x800e0000 896.0KiB
Read: 0x80000000-0x80100000 1.0MiB
Read: 0x80000000-0x80120000 1.1MiB
Read: 0x80000000-0x80140000 1.2MiB
Read: 0x80000000-0x80160000 1.3MiB
Read: 0x80000000-0x80180000 1.5MiB
Read: 0x80000000-0x801a0000 1.6MiB
Read: 0x80000000-0x801c0000 1.7MiB
Read: 0x80000000-0x801e0000 1.8MiB
Read: 0x80000000-0x80200000 2.0MiB
```

```
Write speed: 37.9MiB/s
Read speed: 54.3MiB/s
```

```

Booting from serial...
Press Q or ESC to abort boot completely.
sL5DdSmmkekro
Timeout
Booting from SDCard in SD-Mode...
Booting from boot.json...
boot.json file not found.
Booting from boot.bin...
Copying boot.bin to 0x80000000 (17922920 bytes)...
[
[#####
[#####
Executing booted program at 0x80000000

```

[illegible]

INSTRUCTION SETS WANT TO BE FREE

13 / 22

```

[ 0.000000] rcu: RCU calculated value of scheduler-enlistment delay is 10
jiffies.
[ 0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=1
[ 0.000000] NR_IRQS: 64, nr_irqs: 64, preallocated irqs: 0
[ 0.000000] riscv-intc: 64 local interrupts mapped
[ 0.000000] plic: interrupt-controller@c000000: mapped 8 interrupts with 1
handlers for 2 contexts.
[ 0.000000] rcu: srcu_init: Setting srcu_struct sizes based on contention.
[ 0.000000] riscv-timer: riscv_timer_init_dt: Registering clocksource cpuid [0]
hartid [0]
[ 0.000000] clocksource: riscv_clocksource: mask: 0xffffffffffffffff
max_cycles: 0x1d854df40, max_idle_ns: 7052723233920 ns
[ 0.000044] sched_clock: 64 bits at 500kHz, resolution 2000ns, wraps every
4398046511000ns
[ 0.014086] Console: colour dummy device 128x32
[ 0.018384] Calibrating delay loop (skipped), value calculated using timer
frequency.. 1.00 BogoMIPS (lpj=5000)
[ 0.028422] pid_max: default: 32768 minimum: 301
[ 0.036322] LSM: initializing lsm=capability,integrity
[ 0.046174] Mount-cache hash table entries: 512 (order: 0, 4096 bytes, linear)
[ 0.053262] Mountpoint-cache hash table entries: 512 (order: 0, 4096 bytes,
linear)
[ 0.150992] cblist_init_generic: Setting adjustable number of callback queues.
[ 0.157854] cblist_init_generic: Setting shift to 0 and lim to 1.
[ 0.169374] riscv: ELF compat mode unsupported
[ 0.169752] ASID allocator disabled (0 bits)
[ 0.184590] rcu: Hierarchical SRCU implementation.
[ 0.188386] rcu: Max phase no-delay instances is 1000.
[ 0.207128] EFI services will not be available.
[ 0.217412] smp: Bringing up secondary CPUs ...
[ 0.220950] smp: Brought up 1 node, 1 CPU
[ 0.241748] devtmpfs: initialized
[ 0.301642] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff,
max_idle_ns: 19112604462750000 ns
[ 0.311542] futex hash table entries: 256 (order: 2, 16384 bytes, linear)
[ 0.375634] NET: Registered PF_NETLINK/PF_ROUTE protocol family
[ 0.387194] DMA: preallocated 128 KiB GFP_KERNEL pool for atomic allocations
[ 0.394154] DMA: preallocated 128 KiB GFP_KERNEL|GFP_DMA32 pool for atomic
allocations
[ 0.524034] pps_core: LinuxPPS API ver. 1 registered
[ 0.528598] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo
Giometti <giometti@linux.it>
[ 0.538040] PTP clock support registered
[ 0.575772] clocksource: Switched to clocksource riscv_clocksource
[ 0.996518] NET: Registered PF_INET protocol family
[ 1.009064] IP idents hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 1.046762] tcp_listen_portaddr_hash hash table entries: 256 (order: 0, 4096
bytes, linear)
[ 1.055374] Table-perturb hash table entries: 65536 (order: 6, 262144 bytes,
linear)
[ 1.062640] TCP established hash table entries: 2048 (order: 2, 16384 bytes,
linear)
[ 1.071804] TCP bind hash table entries: 2048 (order: 4, 65536 bytes, linear)
[ 1.084704] TCP: Hash tables configured (established 2048 bind 2048)

```

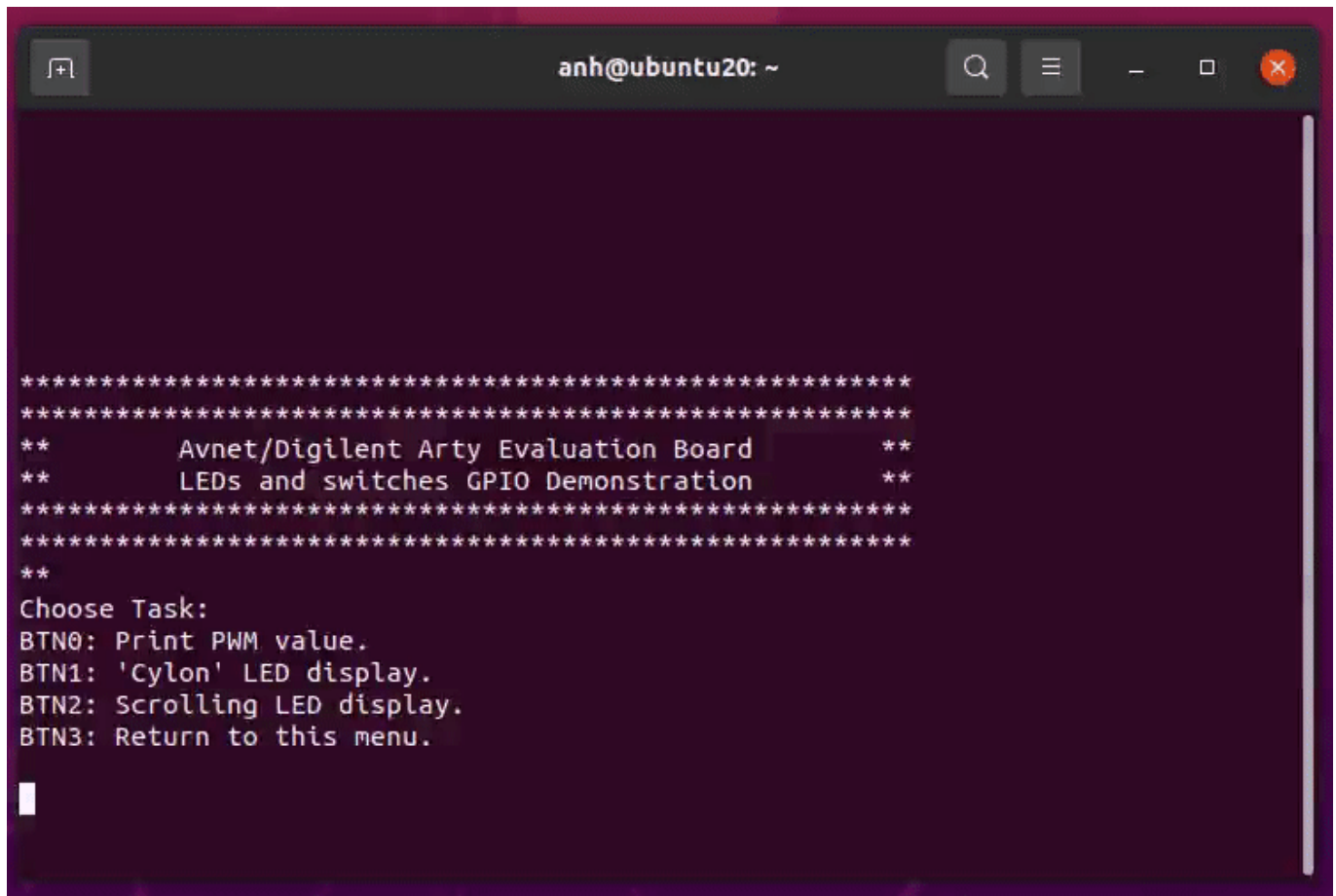
```

[ 1.093312] UDP hash table entries: 256 (order: 1, 8192 bytes, linear)
[ 1.100190] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes, linear)
[ 1.111400] NET: Registered PF_UNIX/PF_LOCAL protocol family
[ 1.315482] workingset: timestamp_bits=46 max_order=16 bucket_order=0
[ 1.441506] LiteX SoC Controller driver initialized
[ 5.033542] 12006800.serial: ttyLXU0 at MMIO 0x0 (irq = 0, base_baud = 0) is a
liteuart
[ 5.046656] printk: console [liteuart0] enabled
[ 5.046656] printk: console [liteuart0] enabled
[ 5.055354] printk: bootconsole [liteuart0] disabled
[ 5.055354] printk: bootconsole [liteuart0] disabled
[ 5.305766] loop: module loaded
[ 5.890194] liteeth 12001000.mac eth0: irq 1 slots: tx 2 rx 2 size 2048
[ 5.916542] NET: Registered PF_INET6 protocol family
[ 5.970540] Segment Routing with IPv6
[ 5.977042] In-situ OAM (IOAM) with IPv6
[ 5.982838] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 6.016568] NET: Registered PF_PACKET protocol family
[ 6.776552] Warning: unable to open an initial console.
[ 6.887680] Freeing unused kernel image (initmem) memory: 2876K
[ 6.897770] Run /init as init process
[ 6.900452]   with arguments:
[ 6.903404]     /init
[ 6.906180]   with environment:
[ 6.908790]     HOME=/
[ 6.911126]     TERM=linux
# uname -a
Linux litex 6.2.0-rc6-314526-g4368aad2af2b-dirty #34 SMP Sun Mar  5 18:53:58 JST
2023 riscv64 GNU/Linux
# df -h
Filesystem                Size      Used Available Use% Mounted on
rootfs                    116.8M    1.1M    115.7M   1% /
devtmpfs                  116.8M         0    116.8M   0% /dev
tmpfs                     118.2M         0    118.2M   0% /tmp
# ls -l
drwxr-xr-x  2 0          0          1120 Jan  1 00:00 bin
drwxr-xr-x  2 0          0         12400 Jan  1 00:00 dev
drwxr-xr-x  2 0          0           60 Mar  5  2023 etc
drwxr-xr-x  2 0          0           40 Mar  5  2023 home
lrwxrwxrwx  1 0          0           11 Mar  5  2023 init -> bin/busybox
drwxr-xr-x  2 0          0           40 Mar  5  2023 lib
lrwxrwxrwx  1 0          0           12 Jan  1 00:00 linuxrc -> /bin/busybox
drwxr-xr-x  2 0          0           40 Mar  5  2023 mnt
drwxr-xr-x  2 0          0           40 Mar  5  2023 nfs
dr-xr-xr-x 91 0          0           0 Jan  1 00:00 proc
drwxr-xr-x  2 0          0           40 Mar  5  2023 root
drwxr-xr-x  2 0          0          620 Jan  1 00:00 sbin
dr-xr-xr-x 11 0          0           0 Jan  1 00:00 sys
drwxrwxrwt  2 0          0           40 Jan  1 00:00 tmp
drwxr-xr-x  5 0          0          100 Mar  5  2023 usr
#

```

6.3.1. Netboot (TFTP):

It will take a bit long to load the LiteX boot image from network. (about 1h30m)



```

anh@ubuntu20: ~
*****
*****
**      Avnet/Digilent Arty Evaluation Board      **
**      LEDs and switches GPIO Demonstration      **
*****
*****
**
Choose Task:
BTN0: Print PWM value.
BTN1: 'Cylon' LED display.
BTN2: Scrolling LED display.
BTN3: Return to this menu.

```

```

[1m      _ _ _ _ _ [0m
[1m      / /  ( ) /___ | |/_/[0m
[1m      / /_ / / _/ -_> <[0m
[1m      /___/_/_/_/_/_/_|_|[0m
[1m      Build your hardware, easily![0m

```

(c) Copyright 2012-2023 Enjoy-Digital

(c) Copyright 2007-2015 M-Labs

BIOS built on Mar 12 2023 15:26:52

BIOS CRC passed (5a795c2d)

LiteX git sha1: 67e8d774

```

----- [1mSoC[0m -----
[1mCPU[0m:      RocketRV64[imac] @ 50MHz
[1mBUS[0m:      WISHBONE 32-bit @ 4GiB
[1mCSR[0m:      32-bit data
[1mROM[0m:      128.0KiB
[1mSRAM[0m:     8.0KiB
[1mSDRAM[0m:    256.0MiB 16-bit @ 400MT/s (CL-7 CWL-5)
[1mMAIN-RAM[0m: 256.0MiB

```

```

--===== [1mInitialization[0m =====--
Ethernet init...
Initializing SDRAM @0x80000000...
Switching SDRAM to software control.
Read leveling:
  m0, b00: |00000000000000000000000000000000| delays: -
  m0, b01: |01111111111111111111111111111100| delays: 15+-14
  m0, b02: |00000000000000000000000000000000| delays: -
  m0, b03: |00000000000000000000000000000000| delays: -
  m0, b04: |00000000000000000000000000000000| delays: -
  m0, b05: |00000000000000000000000000000000| delays: -
  m0, b06: |00000000000000000000000000000000| delays: -
  m0, b07: |00000000000000000000000000000000| delays: -
  best: m0, b01 delays: 15+-14
  m1, b00: |00000000000000000000000000000000| delays: -
  m1, b01: |01111111111111111111111111111110| delays: 15+-14
  m1, b02: |00000000000000000000000000000000| delays: -
  m1, b03: |00000000000000000000000000000000| delays: -
  m1, b04: |00000000000000000000000000000000| delays: -
  m1, b05: |00000000000000000000000000000000| delays: -
  m1, b06: |00000000000000000000000000000000| delays: -
  m1, b07: |00000000000000000000000000000000| delays: -
  best: m1, b01 delays: 15+-14
Switching SDRAM to hardware control.
Memtest at 0x80000000 (2.0MiB)...
  Write: 0x80000000-0x80000000 0B
  Write: 0x80000000-0x80020000 128.0KiB
  Write: 0x80000000-0x80040000 256.0KiB
  Write: 0x80000000-0x80060000 384.0KiB
  Write: 0x80000000-0x80080000 512.0KiB
  Write: 0x80000000-0x800a0000 640.0KiB
  Write: 0x80000000-0x800c0000 768.0KiB
  Write: 0x80000000-0x800e0000 896.0KiB
  Write: 0x80000000-0x80100000 1.0MiB
  Write: 0x80000000-0x80120000 1.1MiB
  Write: 0x80000000-0x80140000 1.2MiB
  Write: 0x80000000-0x80160000 1.3MiB
  Write: 0x80000000-0x80180000 1.5MiB
  Write: 0x80000000-0x801a0000 1.6MiB
  Write: 0x80000000-0x801c0000 1.7MiB
  Write: 0x80000000-0x801e0000 1.8MiB
  Write: 0x80000000-0x80200000 2.0MiB

  Read: 0x80000000-0x80000000 0B
  Read: 0x80000000-0x80020000 128.0KiB
  Read: 0x80000000-0x80040000 256.0KiB
  Read: 0x80000000-0x80060000 384.0KiB
  Read: 0x80000000-0x80080000 512.0KiB
  Read: 0x80000000-0x800a0000 640.0KiB
  Read: 0x80000000-0x800c0000 768.0KiB
  Read: 0x80000000-0x800e0000 896.0KiB
  Read: 0x80000000-0x80100000 1.0MiB
  Read: 0x80000000-0x80120000 1.1MiB

```

```
Memtest OK
Memspeed at 0x80000000 (Sequential, 2.0MiB)...
  Write speed: 37.9MiB/s
  Read speed: 54.3MiB/s
```

```

Booting from serial...
Press Q or ESC to abort boot completely.
sL5DdSMmkekro
Timeout
Booting from SDCard in SD-Mode...
Booting from boot.json...
Booting from boot.bin...
SDCard boot failed.
Booting from network...
Local IP: 192.168.1.50
Remote IP: 192.168.1.100
Booting from boot.json...
Booting from boot.bin...
Copying boot.bin to 0x80000000...

```

```
Executing booted program at 0x80000000
```

[illegible]

INSTRUCTION SETS WANT TO BE FREE

19 / 22

```

[ 0.000000] rcu: RCU restricting CPUs from NR_CPUS=64 to nr_cpu_ids=1.
[ 0.000000] Tracing variant of Tasks RCU enabled.
[ 0.000000] rcu: RCU calculated value of scheduler-enlistment delay is 10
jiffies.
[ 0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=1
[ 0.000000] NR_IRQS: 64, nr_irqs: 64, preallocated irqs: 0
[ 0.000000] riscv-intc: 64 local interrupts mapped
[ 0.000000] plic: interrupt-controller@c000000: mapped 8 interrupts with 1
handlers for 2 contexts.
[ 0.000000] rcu: srcu_init: Setting srcu_struct sizes based on contention.
[ 0.000000] riscv-timer: riscv_timer_init_dt: Registering clocksource cpuid [0]
hartid [0]
[ 0.000000] clocksource: riscv_clocksource: mask: 0xffffffffffffffff
max_cycles: 0x1d854df40, max_idle_ns: 7052723233920 ns
[ 0.000044] sched_clock: 64 bits at 500kHz, resolution 2000ns, wraps every
4398046511000ns
[ 0.014088] Console: colour dummy device 128x32
[ 0.018384] Calibrating delay loop (skipped), value calculated using timer
frequency.. 1.00 BogoMIPS (lpj=5000)
[ 0.028424] pid_max: default: 32768 minimum: 301
[ 0.036322] LSM: initializing lsm=capability,integrity
[ 0.046174] Mount-cache hash table entries: 512 (order: 0, 4096 bytes, linear)
[ 0.053264] Mountpoint-cache hash table entries: 512 (order: 0, 4096 bytes,
linear)
[ 0.150898] cblist_init_generic: Setting adjustable number of callback queues.
[ 0.157760] cblist_init_generic: Setting shift to 0 and lim to 1.
[ 0.169280] riscv: ELF compat mode unsupported
[ 0.169658] ASID allocator disabled (0 bits)
[ 0.184496] rcu: Hierarchical SRCU implementation.
[ 0.188292] rcu: Max phase no-delay instances is 1000.
[ 0.207034] EFI services will not be available.
[ 0.217320] smp: Bringing up secondary CPUs ...
[ 0.220856] smp: Brought up 1 node, 1 CPU
[ 0.241718] devtmpfs: initialized
[ 0.301632] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff,
max_idle_ns: 19112604462750000 ns
[ 0.311528] futex hash table entries: 256 (order: 2, 16384 bytes, linear)
[ 0.375610] NET: Registered PF_NETLINK/PF_ROUTE protocol family
[ 0.387162] DMA: preallocated 128 KiB GFP_KERNEL pool for atomic allocations
[ 0.394120] DMA: preallocated 128 KiB GFP_KERNEL|GFP_DMA32 pool for atomic
allocations
[ 0.523988] pps_core: LinuxPPS API ver. 1 registered
[ 0.528552] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo
Giometti <giometti@linux.it>
[ 0.537992] PTP clock support registered
[ 0.575732] clocksource: Switched to clocksource riscv_clocksource
[ 0.996586] NET: Registered PF_INET protocol family
[ 1.009126] IP idents hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 1.046806] tcp_listen_portaddr_hash hash table entries: 256 (order: 0, 4096
bytes, linear)
[ 1.055400] Table-perturb hash table entries: 65536 (order: 6, 262144 bytes,
linear)
[ 1.062670] TCP established hash table entries: 2048 (order: 2, 16384 bytes,
linear)

```

```

[ 1.071834] TCP bind hash table entries: 2048 (order: 4, 65536 bytes, linear)
[ 1.085306] TCP: Hash tables configured (established 2048 bind 2048)
[ 1.093336] UDP hash table entries: 256 (order: 1, 8192 bytes, linear)
[ 1.100214] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes, linear)
[ 1.111424] NET: Registered PF_UNIX/PF_LOCAL protocol family
[ 1.315472] workingset: timestamp_bits=46 max_order=16 bucket_order=0
[ 1.441522] LiteX SoC Controller driver initialized
[ 5.033496] 12006800.serial: ttyLXU0 at MMIO 0x0 (irq = 0, base_baud = 0) is a
liteuart
[ 5.046610] printk: console [liteuart0] enabled
[ 5.046610] printk: console [liteuart0] enabled
[ 5.055304] printk: bootconsole [liteuart0] disabled
[ 5.055304] printk: bootconsole [liteuart0] disabled
[ 5.305662] loop: module loaded
[ 5.890132] liteeth 12001000.mac eth0: irq 1 slots: tx 2 rx 2 size 2048
[ 5.916484] NET: Registered PF_INET6 protocol family
[ 5.970470] Segment Routing with IPv6
[ 5.976972] In-situ OAM (IOAM) with IPv6
[ 5.982768] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 6.016506] NET: Registered PF_PACKET protocol family
[ 6.776514] Warning: unable to open an initial console.
[ 6.887650] Freeing unused kernel image (initmem) memory: 2876K
[ 6.897740] Run /init as init process
[ 6.900422]   with arguments:
[ 6.903374]       /init
[ 6.906142]   with environment:
[ 6.908760]       HOME=/
[ 6.911096]       TERM=linux
# uname -a
-/bin/ash: uname: not found
# uname -a
Linux litex 6.2.0-rc6-314526-g4368aad2af2b-dirty #34 SMP Sun Mar  5 18:53:58 JST
2023 riscv64 GNU/Linux
# df -h
Filesystem                Size      Used Available Use% Mounted on
rootfs                    116.8M    1.1M    115.7M   1% /
devtmpfs                  116.8M         0    116.8M   0% /dev
tmpfs                     118.2M         0    118.2M   0% /tmp
# ls -l
drwxr-xr-x  2 0          0           1120 Jan  1 00:00 bin
drwxr-xr-x  2 0          0          12400 Jan  1 00:00 dev
drwxr-xr-x  2 0          0           60 Mar  5  2023 etc
drwxr-xr-x  2 0          0           40 Mar  5  2023 home
lrwxrwxrwx  1 0          0           11 Mar  5  2023 init -> bin/busybox
drwxr-xr-x  2 0          0           40 Mar  5  2023 lib
lrwxrwxrwx  1 0          0           12 Jan  1 00:00 linuxrc -> /bin/busybox
drwxr-xr-x  2 0          0           40 Mar  5  2023 mnt
drwxr-xr-x  2 0          0           40 Mar  5  2023 nfs
dr-xr-xr-x 90 0          0           0 Jan  1 00:00 proc
drwxr-xr-x  2 0          0           40 Mar  5  2023 root
drwxr-xr-x  2 0          0           620 Jan  1 00:00 sbin
dr-xr-xr-x 11 0          0           0 Jan  1 00:00 sys
drwxrwxrwt  2 0          0           40 Jan  1 00:00 tmp

```

```
drwxr-xr-x  5 0      0      100 Mar  5  2023 usr
#
```