



HiFive Unmatched Software Reference Manual

Version 1.0

Copyright © by SiFive, Inc. All rights reserved.

HiFive Unmatched Software Reference Manual

Proprietary Notice

Copyright © 2021, SiFive Inc. All rights reserved.

HiFive Unmatched Software Reference Manual by SiFive, Inc. is licensed under Attribution-Non-CommercialNoDerivatives 4.0 International. To view a copy of this license, visit: <http://creativecommons.org/licenses/by-nc-nd/4.0>

Information in this document is provided “as is,” with all faults.

SiFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

SiFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

SiFive reserves the right to make changes without further notice to any products herein.

Revision History

Version	Date	Changes
1.0	May 17, 2021	<ul style="list-style-type: none">Initial version

Contents

1	Introduction	4
2	Freedom U SDK Overview	5
2.1	Toolchains Included with FUSDK	7
2.2	Bootflow	7
2.3	Console Output from Boot Flow Stages	9
2.3.1	ZSBL Boot ROM	9
2.3.2	U-Boot SPL	10
2.3.3	OpenSBI	10
2.3.4	U-Boot	10
2.4	LED Indicators for Bootflow and Status	12
2.5	Shutdown of the System	12
2.6	Image Organization	13
2.6.1	Using NVMe (M.2 types: 2230, 2260 and 2280)	16
2.7	Boot Options including NVMe	17
2.8	FUSDK Source File Organization	19
2.9	FUSDK Graphical User Interface	20
2.10	FUSDK Features	20
2.11	System Time	26
2.12	Identification of FUSDK version	26
2.13	BitBakeBuild System Organization	26
2.14	BitBakeRecipe Flow	28
2.15	Build History and Logs	29
2.16	BitBakeDebug	29
3	HiFive Unmatched Connectivity	30
3.1	Serial Console and Debugger Pod	31
3.2	FTDI Linux plugdev Configuration	32
3.3	FTDI Device Driver Management	32

3.4	Determining the Network Configuration	35
3.5	Secure Shell (SSH) Connection	36
4	Benchmarks	37
4.1	Performance Debug	38
4.2	Determining the Core and DDR Frequency	39
5	RISC-V Linux Software Structure	42
6	Crosstool-NG	43
6.1	Creating a Location for Cross Compiler and Cloning crosstool-ng.....	43
6.2	Build and install crosstool-ng.....	44
6.3	Build the rv64gc linux-gnu cross-toolchain.....	44
6.4	Option 1. C Language Hello World.....	44
6.5	Option 2. C++ Language Hello World.....	45
6.6	Generate an Assembly Listing from the Binary File	45
6.7	Execute the Binary from RISC-V Linux Environment	45
7	Additional Information	46
8	Known Issues and Limitations	48
8.1	ZSBL Boot ROM Fail Messages are Sometimes Illegible	48
8.2	ASMedia USB 3.0 Controller Incompatible with Some Devices.....	49
8.3	Micro-USB Port Fails to be Recognized by Host Machine	49
8.4	Reboot Command not Supported	49
8.5	SanDisk A2 SD Card Incompatibility	49
8.6	NTFS Compatibility	49
8.7	Default Configuration.....	49
8.8	U-Boot Save Environment	50
8.9	Onboard Flash Memory Bootflow	50
8.10	Xfce File View Issue	50
8.11	Mouse Lockup	50
8.12	Real Time Clock (RTC) Not Enabled	50

Appendix A	Sample Log Files	51
A.1	uBootBoot.log	51
A.2	u-boot_help_and_printenv.log	52
A.3	LinuxBoot.log	56
A.4	99-hifive-unmatched.rules	74

1

Introduction

The HiFive Unmatched Software Reference Manual is designed to help software developers create projects in a RISC-V Linux environment using the HiFive Unmatched board.

This document describes the process of booting Linux on the Unmatched board and points to some useful tools which can ease software development on HiFive Unmatched.

This document does not provide an overview of the initial steps necessary to get started using the HiFive Unmatched. For this information, consult the HiFive Unmatched Getting Started Guide.

This document also is not intended to provide a Linux tutorial. This document targets users who are somewhat familiar with working in a Linux environment and, instead, are looking for tips and tricks on how to make the most out of working on the HiFive Unmatched board.

This document is a software reference on the HiFive Unmatched board, and it should be used to complement other sources of information such as Freedom U SDK and RISC-V specifications.

For information about all of the HiFive Unmatched-related resources available, please refer to Chapter 7 below.

2

Freedom U SDK Overview

Freedom U SDK™ (FUSDK) is SiFive's reference Linux distribution, based on OpenEmbedded. It can be downloaded from a GitHub repository at <https://github.com/sifive/freedom-u-sdk>. Regular software updates are posted there. It is recommended to update the board software at your earliest convenience.

SiFive provides FUSDK as an example Linux distribution that demonstrates Linux running on SiFive's U-series cores. Other Linux distributions and bare metal software environments are also available for the HiFive Unmatched.

Freedom U SDK runs on multiple RISC-V platforms, including HiFive Unleashed and the HiFive Unmatched.

The FUSDK is based on the OpenEmbedded framework. Included in FUSDK is the meta-sifive layer. The SiFive meta-sifive layer can be found at <https://github.com/sifive/meta-sifive>.

meta-sifive sits on top of the open source meta-riscv layer. The meta-riscv layer (not maintained by SiFive) can be found at <https://github.com/riscv/meta-riscv/>.

The FUSDK project builds bootloaders, Machine-mode software, toolchains, the Linux kernel, and a root file system. FUSDK uses BitBake recipes to build the Linux distribution. BitBake is the build task executor and scheduler which uses metadata such as configurations, classes, and recipes to build a project. The structure that OpenEmbedded provides can facilitate creating a custom Linux distribution tailored to embedded systems and moving a Linux project from other processor architectures to RISC-V with the concepts of BSP layers and configurations. Further information on OpenEmbedded and BitBake is beyond the scope of this document; many resources are available on the Internet.

It is important to note that building Freedom U SDK for the HiFive Unmatched is resource intensive. Consider the following requirements to build Freedom U SDK:

- 10 GByte of data will be downloaded
- Approximately 200 GByte of disk space will be consumed

- 4+ hours to build on typical desktop PC (32GB RAM + 8 core 3 GHz processor)
- A minimum of 12GB memory is recommended
- Linux environment is required

There are multiple Linux packages that are required to build the FUSDK. These can be a function of the Linux distribution of the host machine although these packages are shared between machines.

GCC and Clang (LLVM) are large software projects, and they are built as part of building FUSDK. They are an intensive use of host system memory during a build. Out of memory based errors are possible during the memory intensive linking stages if enough resources are not available. You may find that reducing jobs and threads is a more reliable way to build the FUSDK on a host machine with fewer resources. One technique to increase reliability is to compile the more intensive builds with limited jobs/threads prior to building everything else with higher numbers of jobs/threads. Building with low numbers of jobs and threads will be slower. To compile LLVM only with minimal jobs/threads to help prevent out of memory issues see an example below that reduces jobs to 1, threads to 1 and builds only LLVM (which is one of the larger components of FUSDK):

```
PARALLEL_MAKE="-j 1" BB_NUMBER_THREADS=1 MACHINE=unmatched bitbake llvm
```

Another build technique is to use a reactive approach that changes jobs/threads dynamically if problems are encountered. If your system is running out of memory (Segfaults are an indication of out of memory) you can retry the build with lower numbers of jobs and threads.

If you want to only download files required for a build you can use the `--runall fetch` BitBake option:

```
MACHINE=unmatched bitbake --runall fetch
```

There are both graphical and command line interface (CLI) BitBake build targets:

- Command line: `demo-coreip-cli`
- Graphical which includes Mesa support: `demo-coreip-xfce4`

These build targets will construct images suitable for writing to micro SD cards, which can be used to boot the HiFive Unmatched board. Software such as balenaEtcher, Rufus, or dd can be used to write the SD card images. The graphical build image (also known as the xfce4 image) is about 50% larger than the CLI build. The xfce4 image is about 7.9GBytes uncompressed.

When building FUSDK from source, the machine type must be specified by the "MACHINE" parameter:

- `freedom-u540` (HiFive Unleashed)
- `unmatched`

After building FUSDK, or downloading a binary release from <https://github.com/sifive/freedom-u-sdk/releases>, you can write the image to a micro SD card. Block devices such as micro SD cards, NVMe drives, and USB drives can be written with the `dd` command or using applications such as Etcher or Rufus.

Warning

Writing images to the wrong disk can be destructive. Please carefully check that you write the intended media device.

2.1 Toolchains Included with FUSDK

The Freedom-U-SDK build includes both GCC and Clang (LLVM) toolchains.

2.2 Bootflow

The HiFive Unmatched bootflow utilizes multiple software and hardware components to boot the system. This bootflow allows for flexibility in the boot process by allowing the user to target different boot sources using the MSEL[3:0] settings which are configured by DIP switches on the HiFive Unmatched. For information about how the values of MSEL in Table 1 apply to the Boot Mode Select Switches on the HiFive Unmatched board, consult the HiFive Unmatched Getting Started Guide.

Table 1: Boot Mode Select Configuration Settings

MSEL[3:0]	FSBL Location	Access Method	SPI Width
4b'0101	QSPI0 Flash	Memory mapped	X1
4b'0110	QSPI0 Flash	Memory mapped	X4
4b'0111	QSPI1 Flash	Memory mapped	X4
4b'1000	QSPI1 SD Card	Bit-banged	X1
4b'1001	QSPI2 Flash	Bit-banged	X1
4b'1010	QSPI0 Flash	Memory mapped	X4
4b'1011 (Default)	QSPI2 SD Card	Bit-banged	X1
4b'1100	QSPI1 Flash	Bit-banged	X1
4b'1101	QSPI1 Flash	Memory mapped	X4
4b'1110	QSPI0 Flash	Bit-banged	X1

Table 1: Boot Mode Select Configuration Settings

4b'1111	QSPI0 Flash	Memory mapped	X4
---------	-------------	---------------	----

The first software boot stage is the First Stage Boot Loader (FSBL), which is loaded by the SoC boot ROM into on-chip SRAM when the SoC exits reset. Since, after reset, the SoC runs at a low clock rate, and main system memory is not yet available, the size of the FSBL is kept very small (below 64KiB) to speed loading and execution.

FUSDK uses U-Boot SPL as its First Stage Boot Loader. U-Boot SPL increases the CPU clock rate to full speed, and configures the DDR DRAM controller, enabling subsequent software to run at full speed and use all of the DRAM available on the board. U-Boot SPL then loads and starts the next boot stage, consisting of OpenSBI and U-Boot, from the micro SD card into DRAM.

OpenSBI is the M-mode Supervisor Binary Interface software implementation used by FUSDK. Further information on OpenSBI can be found at <https://github.com/riscv/opensbi>.

OpenSBI implements the RISC-V Supervisor Binary Interface (SBI) specification, used by S-mode software, such as the Linux kernel, to implement software environment calls for common low-level system operations requiring a higher privilege level. More information on the RISC-V SBI specification is available from the RISC-V International association at <https://github.com/riscv/riscv-sbi-doc>.

U-Boot is the FUSDK bootloader. U-Boot is responsible for determining which hardware device — micro SD, NVMe, or Ethernet — should be used to load the Linux kernel. It also exposes a command line interface on the board's console serial port. This command line interface can be used to test and configure the board, change the Linux kernel command line parameters, and boot alternate operating systems. Networking and PCIe (gen1) are available in the U-Boot environment.

More detail on the FU740 SoC bootflow can be found in the FU740-C000 Manual.

Figure 1 below shows software components of the boot flow, the type of memory they use, and the RISC-V privilege levels the software runs at.

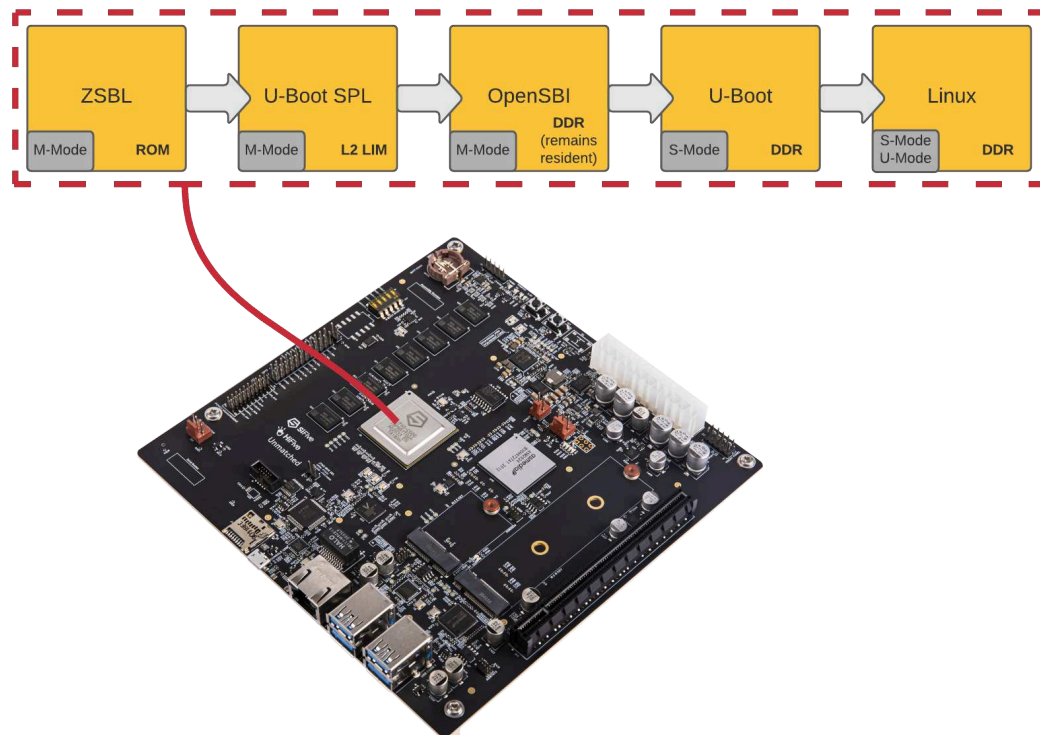


Figure 1: HiFive Unmatched Boot Flow

The HiFive Unmatched bootflow uses as much open source software as possible. The following HiFive Unmatched software components are open source:

- U-Boot SPL
- OpenSBI
- U-Boot
- Linux kernel and applications

2.3 Console Output from Boot Flow Stages

2.3.1 ZSBL Boot ROM

Typically, there is no output from the ZSBL stage of the boot. In this stage of the bootflow, only errors are output to the console.

There is an issue with the ZSBL boot fail message baud rate not being 115200 baud. See Chapter 8 for more information about this issue and a workaround.

2.3.2 U-Boot SPL

In FUSDK, U-Boot SPL is the first piece of software loaded on the chip.

The example console output below includes version and build information as well as boot information.

```
U-Boot SPL 2021.01 (Apr 09 2021 - 08:56:15 +0000)
Trying to boot from MMC1
```

2.3.3 OpenSBI

OpenSBI does not typically print to the console. However, Linux early boot messages can be printed using the SBI console mechanism. Note the `earlycon=sbi` parameter in U-Boot `extlinux.conf`. The contents of this file are shown in Figure 3. The Linux kernel boot process will report OpenSBI version and standard information and below is an example of what's reported from Linux kernel for OpenSBI:

```
[ 0.000000] SBI specification v0.2 detected
[ 0.000000] SBI implementation ID=0x1 Version=0x9
[ 0.000000] SBI v0.2 TIME extension detected
[ 0.000000] SBI v0.2 IPI extension detected
[ 0.000000] SBI v0.2 RFENCE extension detected
[ 0.000000] SBI v0.2 HSM extension detected
```

2.3.4 U-Boot

U-Boot prints information about itself, the system and the boot flow. This is the first bootloader that accepts keyboard input to interrupt the boot flow.

Below is sample output for a system configured with Ethernet connection, PCIe video card, USB data drive, NVMe data drive, and wireless M.2E adapter.

```
U-Boot 2021.01 (Apr 09 2021 - 08:56:15 +0000)

CPU:   rv64imafdc
Model: SiFive HiFive Unmatched A00
DRAM:  16 GiB
MMC:   spi@10050000:mmc@0: 0
EEPROM: SiFive PCB EEPROM format v1
Serial number: SF105SZ210400068
PCB revision: 3
Ethernet MAC address: 70:b3:d5:92:f4:73
CRC:   cc7cb7a1
EEPROM dump: (0x25 bytes)
00: F1 5E 50 45 01 02 00 03 41 00 53 46 31 30 35 53
10: 5A 32 31 30 34 30 30 30 36 38 01 70 B3 D5 92 F4
20: 73 A1 B7 7C CC
found SiFive v1
In:    serial@10010000
Out:   serial@10010000
Err:   serial@10010000
```

```
Model: SiFive HiFive Unmatched A00
Net: eth0: ethernet@10090000
Hit any key to stop autoboot: 0
PCIe Link up, Gen1

Device 0: unknown device
starting USB...
Bus xhci_pci: Register 4000840 NbrPorts 4
Starting the controller
USB XHCI 1.00
scanning bus xhci_pci for devices... 7 USB Device(s) found
    scanning usb for storage devices... 1 Storage Device(s) found

Device 0: Vendor: Kingston Rev: 1.00 Prod: DataTraveler G3
        Type: Removable Hard Disk
        Capacity: 15295.0 MB = 14.9 GB (31324160 x 512)
... is now current device
Scanning usb 0:3...
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
205 bytes read in 2 ms (99.6 KiB/s)
1:      OpenEmbedded-SiFive-HiFive-Unmatched
Retrieving file: /Image.gz
7249366 bytes read in 274 ms (25.2 MiB/s)
append: root=/dev/mmcblk0p4 rootfstype=ext4 rootwait console=ttySIF0,115200
earlycon=sbi
Retrieving file: /hifive-unmatched-a00.dtb
9806 bytes read in 2 ms (4.7 MiB/s)
    Uncompressing Kernel Image
Moving Image from 0x84000000 to 0x80200000, end=810b2000
## Flattened Device Tree blob at 88000000
    Booting using the fdt blob at 0x88000000
    Using Device Tree in place at 0000000088000000, end 000000008800564d

Starting kernel ...
```

An example raw boot log file can be viewed in Appendix A.

There is a default 3 second timeout to interrupt the default boot process. During this time you can press a key on the serial console to get a U-Boot prompt. Note that you will not see this prompt with Ethernet or Graphical connections. From the U-Boot prompt you can inspect the system, control boot flow, and have limited access network and file systems. Using U-Boot is beyond the scope of this document. For U-Boot documentation please see: <https://www.denx.de/wiki/U-Boot/Documentation>

U-Boot has environment variables and functions which are visible with the `printenv` U-Boot command. These variables are used to control boot flow such as which devices to boot Linux from and what addresses will be used in the Linux boot process. As an example, the supported U-Boot commands, the results from the U-Boot `printenv` command and a log of the Linux kernel boot messages can be found in Appendix A.

2.4 LED Indicators for Bootflow and Status

There are two user programmable LEDs present on the board, which provide bootflow and status information by signaling the boot stage with a Red-Green-Blue (RGB) LED and by blinking a green heartbeat function LED.

Table 2: Board D2 reference designator RGB LED function

Status	Meaning
Undetermined	Default state
Yellow	U-Boot SPL (very short run time).
Purple	U-Boot proper was entered.
Blue	extlinux is about to start sysboot.
White	systemd-udev will set it during kernel boot (past init process start).
Green	Will be set 3 minutes after system is booted into basic.target and console is available.

Table 3: Board D12 reference designator Green LED function

Status	Meaning
Blinking	The kernel booted and running. This is driven by the Linux kernel's "heartbeat" function, and only starts blinking after the kernel starts
Not-Blinking	The kernel is hanged/crashed.
Blinking fast	The workload on the system is very high.

2.5 Shutdown of the System

File system corruption can cause a variety of temporary and permanent system errors. A graceful shutdown can help prevent file system corruption.

The commands below should be used to start the Linux shutdown process. After the user has been notified that this process has finished, the board main power will be turned off.

- halt
- poweroff
- shutdown -h now

Note that you should receive a notification that the graceful shutdown commands have completed successfully. As a confirmation, a message stating “System halted” or “Power down” will be printed to the console.

If you find that you have a corrupted file system, you may have to reimage the micro SD card. Micro SD card images can be found the link below. It is recommended to use the latest release that is available:

<https://github.com/sifive/freedom-u-sdk/releases>

2.6 Image Organization

Writing an micro SD card (or another block device) as described in the instructions at <https://github.com/sifive/freedom-u-sdk/tree/2020.03#readme> will result in a disk that has four partitions. This process decompresses the output file from a build in the .wic.xz or .wic.gz format and images this decompressed image to a device. After writing the device, it will look something like this when inspected with: `sudo parted -l`

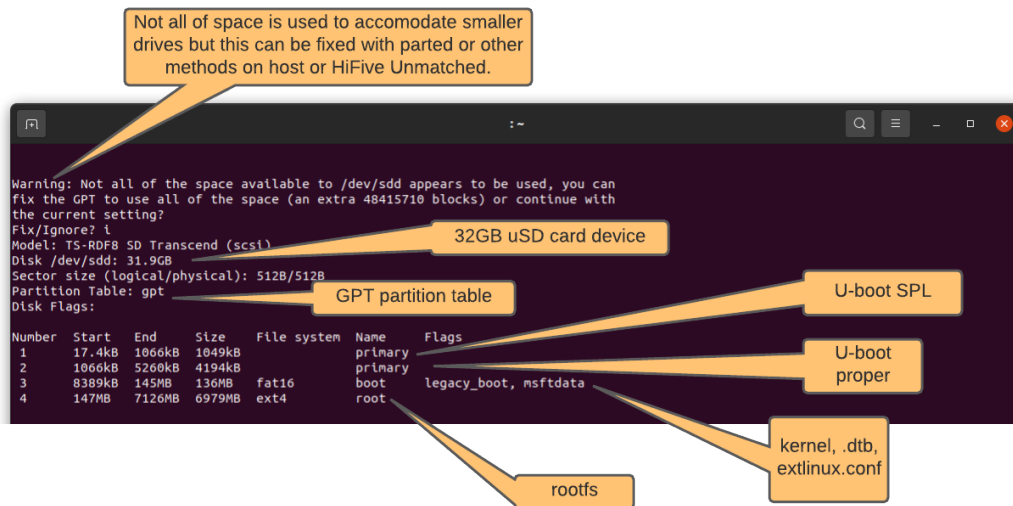


Figure 2: Freedom U SDK Disk Partition

Partition 1 is in a RAW format, and it contains a binary U-Boot SPL. The fixed attributes specified below are used to locate it in the bootflow by the ZSBL:

- type: 5b193300-fc78-40cd-8002-e86c45580b47
- guid: cbcbef44-e627-42bc-b134-93b6f3784b8c

Partition 2 is in a RAW format, and it contains a binary U-Boot ITB (DTB with U-Boot overlay, OpenSBI generic FW_DYNAMIC, U-Boot proper). The fixed attributes specified below are used to locate it in the bootflow by U-Boot SPL:

- type: 2e54b353-1271-4842-806f-e436d6af6985
- guid: f54eba28-d8de-4852-978d-1a673777e2ae

Partition 1 and 2 have been labeled by the community and they have a legacy “Unleashed” type:

<https://kernel.googlesource.com/pub/scm/utils/util-linux/util-linux/+master/include/pt-gpt-part-names.h>

Partition 3 is a FAT16 format partition with the name “boot”. It contains EXTLINUX configuration information, kernel image, and the device tree blob. Note that an optional uEnv.txt file can be added for additional boot control. For more information on EXTLINUX please reference:

<https://wiki.syslinux.org/wiki/index.php?title=EXTLINUX>

```

partition3
├── extlinux
│   └── extlinux.conf
├── hifive-unmatched-a00.dtb
└── Image.gz
  
```

```

SiFive:/partition3/extlinux$ cat /partition3/extlinux/extlinux.conf
default OpenEmbedded
label OpenEmbedded-SiFive-HiFive-Unmatched
kernel /Image.gz
fdt /hifive-unmatched-a00.dtb
append root=/dev/mmcblk0p4 rootfstype=ext4 rootwait console=ttySIF0,115200 earlycon=sbi
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
SiFive:/partition3/extlinux$
  
```

Figure 3: *extlinux.conf* Contents

Partition 4 is a ext4 format partition with the name “root”. This contains the root file system that is built by FUSDK.

The wic program is a BitBake tool is used to create the images and is guided by a file in the meta-sifive layer which describes the partitions, data, format, sizes, etc of an image:

risc-sifive/meta-sifive/scripts/lib/wic/canned-wks/unmatched-spl-opensib.wks

This unmatched-spl-opensib.wks file contains the following information:


```
# short-description: Create SD card image for HiFive Unmatched development board with
U-Boot SPL and OpenSBI (FW_DYNAMIC)

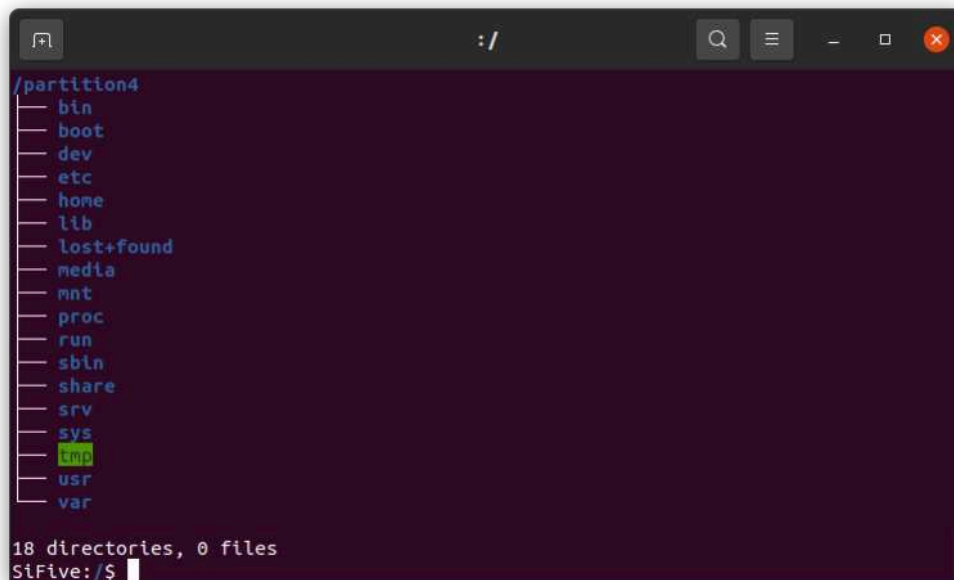
part --source rawcopy --sourceparams="file=u-boot-spl.bin" --ondisk mmcblk0
--fixed-size 1M --align 17 --part-type 5b193300-fc78-40cd-8002-e86c45580b47

part --source rawcopy --sourceparams="file=u-boot.itb" --ondisk mmcblk0 --fixed-size
4M --align 1 --part-type 2e54b353-1271-4842-806f-e436d6af6985

part /boot --source bootimg-partition --sourceparams="loader=u-boot" --ondisk mmcblk0
--fstype=vfat --label boot --active --size=100M --align 4096

part / --source rootfs --ondisk mmcblk0 --fstype=ext4 --label root --align 4096
--size 5G

bootloader --ptable gpt --configfile="unmatched-extlinux.conf"
```



SiFive recommends the use of 32GB (or larger) micro SD cards to enable enough space for Linux as well as applications on the micro SD card. The fourth partition with the root file system is typically expanded to consume all of the unused space on the device. This expansion can be done on a running system, but it's more reliable to expand this rootfs partition offline using a host machine instead of when the micro SD card is being actively used as a root file system on the HiFive Unmatched.

From a host machine, you can issue the following commands where sdX is the device on your system. Note that the commands (except the sync command) need to be run as either root or with sudo privilege:

```
sgdisk -v /dev/sdX
sgdisk -e /dev/sdX
parted /dev/sdX resizepart 4 100%
resize2fs /dev/sdX
sync
```

Other options for resizing partitions on a host machine with a GUI environment include the following applications:

- GNOME disks
- GParted

From the HiFive Unmatched, you can issue the following commands to resize the fourth partition to the maximum size on a running system. These commands are specific to a micro SD card but similar commands would work with other device types with adjustments for the device parameters:

```
sgdisk -v /dev/mmcblk0
sgdisk -e /dev/mmcblk0
parted /dev/mmcblk0 resizepart 4 100%
resize2fs /dev/mmcblk0p4
sync
```

The `-v` option of `sgdisk` verifies the disk and this step is optional but potentially informative. The `-e` option of `sgdisk` fixes the backup GPT structures by moving them to the end of the disk. The `parted` command increases the size of partition 4 to 100% of the (remaining) capacity of the drive. The `resize2fs` command updates the size of the file system to the kernel and `sync` command commits any data resident in the volatile areas of the kernel to the non-volatile buffers which ensures the integrity of non-volatile buffer operations.

2.6.1 Using NVMe (M.2 types: 2230, 2260 and 2280)

Typically, a new NVMe SSD will not be partitioned or formatted when first installed into the system. The user must first partition and format the SSD drive before it may be used.

There are many options for partitioning and formatting the NVMe SSD. The NVMe SSD can be partitioned and formatted before installing on the HiFive Unmatched board, or these steps can be done on the HiFive Unmatched board using FUSDK. The FUSDK includes programs `parted` and `mkfs` which can be used to manage an SSD.

```

root@unmatched:~# parted /dev/nvme0n1
GNU Parted 3.4
Using /dev/nvme0n1
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/nvme0n1: unrecognised disk label
Model: Samsung SSD 970 EVO Plus 500GB (nvme)
Disk /dev/nvme0n1: 500GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted) mklabel gpt
(parted) mkpart
Partition name? []? fastStorage
File system type? [ext2]? ext4
Start? 0%
End? 100%
(parted) print
Model: Samsung SSD 970 EVO Plus 500GB (nvme)
Disk /dev/nvme0n1: 500GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
Number  Start  End    Size  File system  Name  Flags
  1      1049kB 500GB 500GB ext4         fastStorage

(parted) quit
Information: You may need to update /etc[19253.434735] nvme0n1: p1
/fstab.
root@unmatched:~# mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.46.1 (9-Feb-2021)
Discarding device blocks: 4096/122096384/113250304/122096384 done
Creating filesystem with 122096384 4k blocks and 30531584 inodes
Filesystem UUID: b32f24ec-3bd8-4e0a-8f31-0f53f29332d7
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962024, 11239424, 20480000, 23887872, 71663616, 78675968,
102400000
Allocating group tables: 0/3727 done
Writing inode tables: 0/3727 done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: 0/3727 done
root@unmatched:~#

```

An SSD (or USB memory drive) that has been partitioned and formatted will be detected during boot by FUSDK and auto mounted at `/run/media/`.

For many applications, using an NVMe SSD for storage instead of a micro SD card will dramatically increase performance of the HiFive Unmatched system.

2.7 Boot Options including NVMe

With current versions of FUSDK, U-Boot, OpenSBI, and U-Boot SPL must reside in the micro SD card. However, the kernel and rootfs may reside in faster storage such as an NVMe storage device, USB mass storage device, or an Ethernet-connected device. Because of its high performance, an NVMe file system is recommended over micro SD file systems for doing file system intensive work.

The simplest option to enable kernel and rootfs to be used from the NVMe device is to use a two step process:

1. Write the NVMe device using the same `xzcat/dd` process used to write a micro SD card. This can be done from a host machine or from HiFive Unmatched. From a host machine

you'll need an NVMe connection such as a motherboard with NVMe support or a USB to NVMe adapter. Note that the NVMe devices and their connectors are not as rugged as typical USB drives and they can get hot during intensive operations.

- a. `xzcat demo-coreip-xfce4-unmatched-2021.03.01.rootfs.wic.xz | sudo dd of=/dev/nvmeX bs=512K iflag=fullblock oflag=direct conv=fsync status=progress # where nvmeX is the NVMe device you want to write`
- b. `sync`
- c. Resize partition and expand the 4th partition to maximum capacity. Where `nvmeX` is the NVMe device you want to expand the partition of:

- i. `sgdisk -v /dev/nvmeX`
- ii. `sgdisk -e /dev/nvmeX`
- iii. `parted /dev/nvmeX resizepart 4 100%`
- iv. `resize2fs /dev/nvmeX`
- v. `sync`

2. Modify files on the device to configure U-Boot to use that device for the root file system.
 - a. Mount the NVMe drive to Linux (or even use Windows machine to access because the boot partition is a fat file system). It's the third partition with the "boot" label.
 - b. Edit `extlinux/extlinux.conf` file of the boot labeled partition to change the "root=" boot parameter to the fourth partition of the NVMe device which is the root partition of the FUSDK build.

FUSDK U-Boot has an established boot priority with environment settings:

- `bootcmd=run distro_bootcmd`
- `distro_bootcmd=setenv nvme_need_init; for target in ${boot_targets}; do run bootcmd_${target}; done`
- `boot_targets=nvme0 usb0 mmc0 pxe dhcp`

NVMe is the first device in the list and U-Boot will attempt to source the `extlinux.conf` file from the NVMe drive. If an NVMe device is not present or expected or `extlinux.conf` file isn't present on the NVMe device, the next device in `boot_targets` list is queried. U-Boot will run the `bootcmd` script automatically on boot, but there is a 3 second window where you can interrupt the boot flow to get a U-Boot prompt. From a U-Boot prompt, you can manually select the boot device you want to run and override the default priority. For example, you could interrupt the boot flow and then execute the U-Boot command: `run bootcmd_mmc0` to boot from the micro SD card.

U-Boot is configured to source the `extlinux/extlinux.conf` file from the boot partition of a boot target device. This `extlinux.conf` configuration file specifies label(s) with the kernel, device tree, and kernel boot parameters including the root file system partition. You can change the `root=/dev/mmcblk0p4` parameter to `root=/dev/nvme0p4` to configure the rootfs for the kernel to be on the NVMe drive instead of the micro SD card.

A similar technique described for an NVMe drive above could be used to create a USB based device for kernel and rootfs.

Another option to customize the boot process includes setting up a `uEnv.txt` boot script for U-Boot. Creating custom `uEnv.txt` scripts is beyond the scope of this document. Please see U-Boot documentation for more information on this subject.

2.8 FUSDK Source File Organization

The following paths are provided as reference for users looking for the location of certain files or directories in the FUSDK file hierarchy. This section assumes that you've built the FUSDK on a build machine and provides some insight into the build output.

Kernel DTS/DTSI files for HiFive Unmatched:

- `riscv-sifive/build/tmp-glibc/work-shared/unmatched/kernel-source/arch/riscv/boot/dts/sifive/hifive-unmatched-a00.dts`
- `riscv-sifive/build/tmp-glibc/work-shared/unmatched/kernel-source/arch/riscv/boot/dts/sifive/fu740-c000.dtsi`

U-Boot DTS/DTSI files for HiFive Unmatched:

- `riscv-sifive/build/tmp-glibc/work/unmatched-oe-linux/u-boot/1_2021.01-r0/git/arch/riscv/dts/`

Rootfs location in the build (demo-coreip-xfce4 MACHINE example):

- `/riscv-sifive/build/tmp-glibc/work/unmatched-oe-linux/demo-coreip-xfce4/1.0-r0/rootfs`

U-Boot:

- `/riscv-sifive/build/tmp-glibc/work/unmatched-oe-linux/u-boot/1_2021.01-r0`

OpenSBI:

- `/riscv-sifive/build/tmp-glibc/work/unmatched-oe-linux/opensbi/0.9-r0`

Linux kernel:

- `/riscv-sifive/build/tmp-glibc/work/unmatched-oe-linux/linux-mainline/5.11.x+gitAUTOINC+f8df82da75-r0`

GCC:

- `/riscv-sifive/build/tmp-glibc/work-shared/gcc-10.2.0-r0`

A more general solution for identifying the location of a build output may be to use the Bit-Bakedevshell option. With this option you can use the build target as a parameter, run the Bit-Bakecommand, and the build tool will leave you in an interactive shell at the path of the build target. It may also be helpful to use the tmux application in this environment to work in this path.

The FUSDK build uses GCC as its compiler. LLVM is built and installed but not used to build the kernel or file system.

Incremental compiles are possible in the BitBaketool such that the entire project is not recompiled when smaller changes are made. There are shared packages between machine targets and build targets also that prevent some unnecessary duplication of builds.

2.9 FUSDK Graphical User Interface

A GUI initial boot screenshot is shown in Figure 4 below for reference. The user will only see this view of the desktop if the system is configured with a graphics card and the system is using the xfce4 build/image. Note that this system also has Bluetooth adapter and Ethernet connection, so there are icons for these features in the system tray.

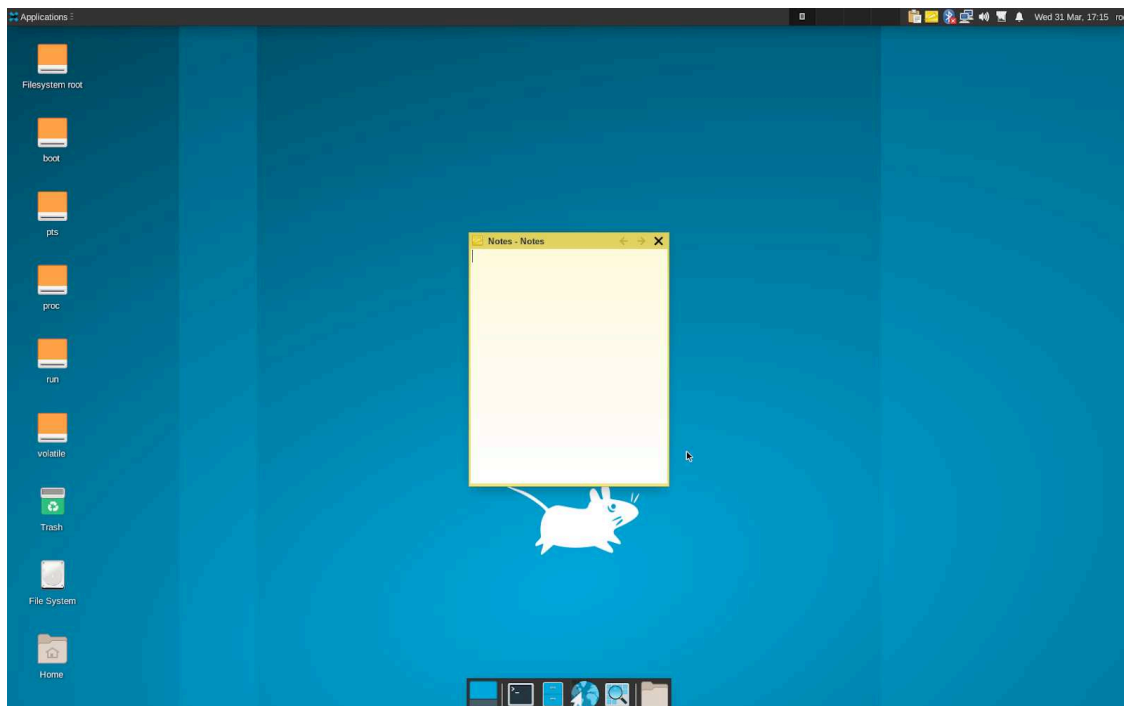


Figure 4: FUSDK Graphical User Interface

2.10 FUSDK Features

Managing storage can be a key component to using the HiFive Unmatched. Figure 5 shows the disk command line tools that are available by default in FUSDK:

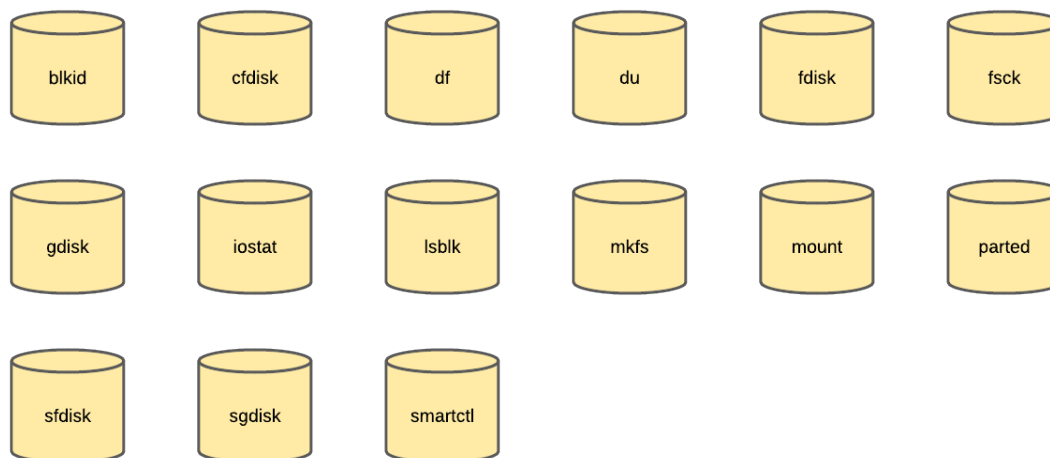


Figure 5: FUSDK Disk Tools

Figure 6 shows the network command line tools that are available by default in FUSDK:

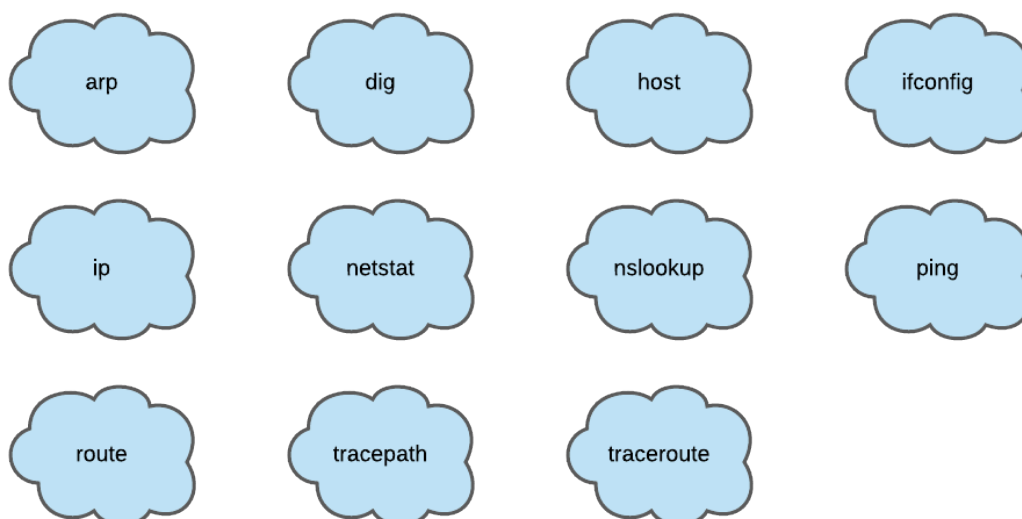


Figure 6: FUSDK Network Tools

There are 3552 packages installed in FUSDK 2021.03. To see a list of these packages, you can execute the following command from HiFive Unmatched prompt: `opkg list-installed`. The FUSDK does not have a live, hosted package repository. You can build and host one yourself; otherwise, you will need to add the recipe and rebuild and flash the disc image to the storage media to access additional packages. Alternatively, you could build and install these on device following the package instructions.

As there is active development in the FUSDK, new features will be developed and issues will be fixed. Therefore, it is highly recommended that the user updates the factory image that is

shipped with the HiFive Unmatched with the latest FUSDK release from GitHub. To find the latest release of FUSDK, see:

<https://github.com/sifive/freedom-u-sdk/releases>

SiFive's strategy is to upstream versus provide patches. It takes time to get patches accepted by the open source community, and it takes time to get these patches included into releases that use them. Patches that haven't been accepted upstream may appear in the FUSDK, but should eventually be upstreamed.

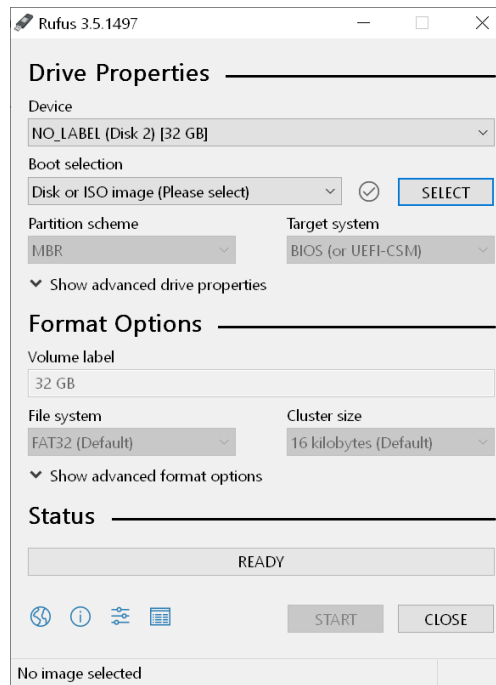
The FUSDK expects a Linux environment to operate, but it's also possible to use another OS to update and write an image from a GitHub release. In Linux, this imaging of media, such as a micro SD card, can be done on the command line. Alternatively, a GUI tool can be used to write a storage device.

Writing a device from Windows may require the use of a utility program such as rufus. Below is a description of how to write a FUSDK image to a micro SD card using the rufus tool:

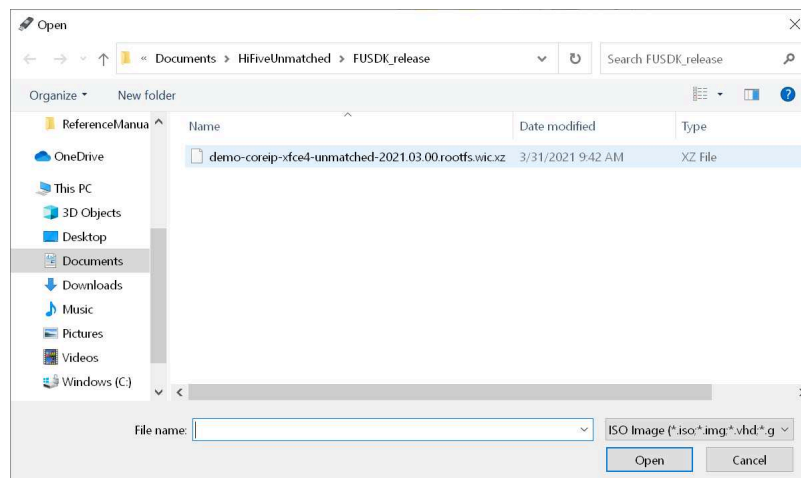
Caution

Writing images to the wrong disk can be destructive. Please carefully check that you write the intended media device.

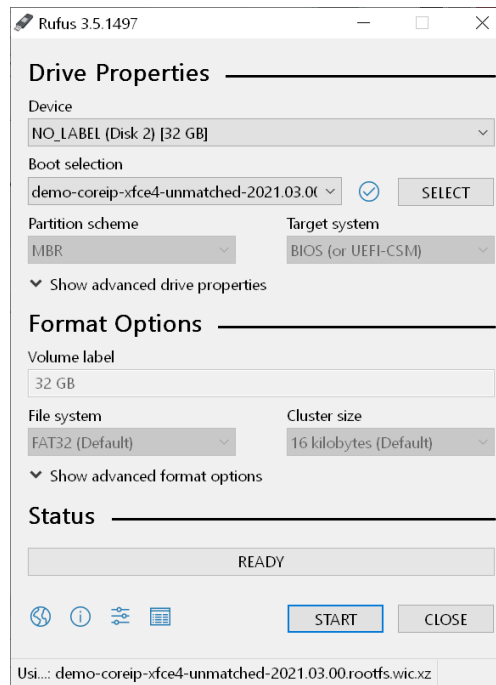
1. Download, install and run rufus (<https://rufus.ie/>) or another disk imaging utility.
2. Download or build the image for the HiFive Unmatched. This image is typically a file in the .wic.xz format such as: demo-coreip-xfce4-unmatched-2021.03.00.rootfs.wic
3. Connect the device to be imaged to the system. For example, use a USB to micro SD card adapter.
4. Open the rufus utility program and select the device you want to write the image to. Be careful not to select the wrong device. Note that sometimes Windows will assign each partition a drive letter so for one device you may see multiple drive letters associated with it.



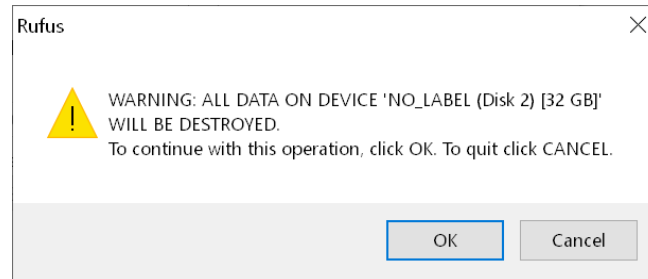
5. Select the image file using the SELECT button. For boot selection, choose the Disk or ISO image options. With this option, other configurations such as “Partition scheme” are not selectable so will be greyed out.



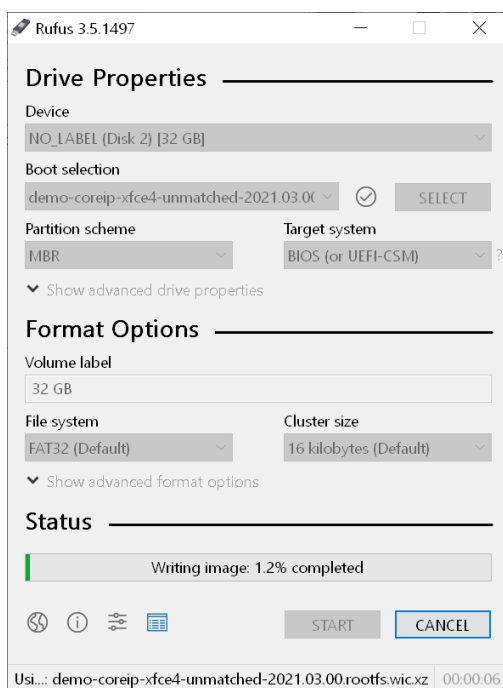
6. After selecting the file and the device, use the START button to write the device.



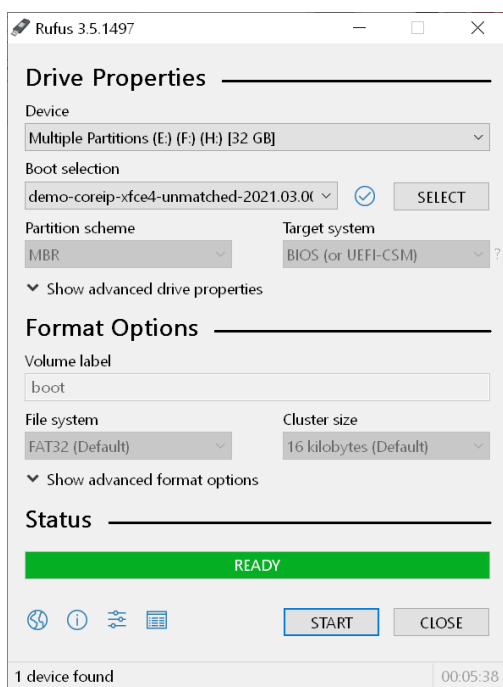
7. You will be prompted to approve the procedure. Verify that you've configured the image writing correctly and then press OK.



8. The image writing process will take about 5 minutes, and then you should see updates as the device is imaged.



9. When imaging is complete you should see a green “READY” bar.



10. Eject the media from the operating system to ensure file buffers are flushed.

11. You can now physically transfer the media to use in a HiFive Unmatched.

2.11 System Time

The FUSDK uses the chrony daemon to automatically synchronize system time with time information from a network time protocol server if there is network access to this server. The configuration for cronyn, which includes the time server and other details about how time is updated on the system, is in:

```
/etc/systemd/timesyncd.conf
```

If there is no network access for the HiFive Unleashed then day and time can be set using the date command. The below command sets the day and time to January 1, 2021 and the hours, minutes, and seconds to 11, 12, and 13 respectively.

```
sudo date -s "1 january 2021 11:12:13"
```

Some operations require time to be set accurately before they will work, such as ssh.

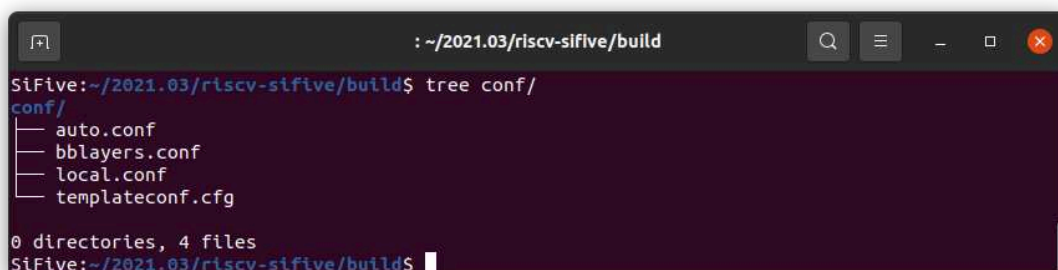
2.12 Identification of FUSDK version

From the prompt of a HiFive Unmatched system you can determine which build you have by executing the following command to view the release info for that build:

```
root@unmatched:~# cat /etc/os-release
ID=nodistro
NAME="FreedomUSDK"
VERSION="2021.03.01 (2021March)"
VERSION_ID=2021.03.01
PRETTY_NAME="FreedomUSDK 2021.03.01 (2021March)"
```

2.13 BitBakeBuild System Organization

Configuration files are located in the build/conf directory:



```
SiFive:~/2021.03/riscv-sifive/build$ tree conf/
conf/
├── auto.conf
├── bblayers.conf
├── local.conf
└── templateconf.cfg

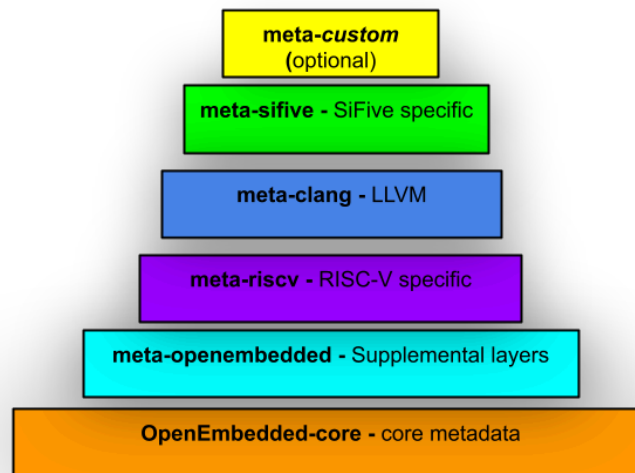
0 directories, 4 files
SiFive:~/2021.03/riscv-sifive/build$
```

The hierarchy of the OpenEmbedded software is shown below. Note that you can add custom layer and that layers are built on top of other layers. The bitbake-layers show-layers command can be used to show the layers, paths, and priorities:

Table 4: *OpenEmbedded Software Hierarchy*

Layer	Path	Priority
meta	riscv-sifive/openembedded-core/meta	5
meta-oe	riscv-sifive/meta-openembedded/meta-oe	6
meta-python	riscv-sifive/meta-openembedded/meta-python	7
meta-multimedia	riscv-sifive/meta-openembedded/meta-multimedia	6
meta-networking	riscv-sifive/meta-openembedded/meta-networking	5
meta-gnome	riscv-sifive/meta-openembedded/meta-gnome	7
meta-xfce	riscv-sifive/meta-openembedded/meta-xfce	7
meta-riscv	riscv-sifive/meta-riscv	6
meta-clang	riscv-sifive/meta-clang	7
meta-sifive	riscv-sifive/meta-sifive	8

The meta-sifive layer contains recipes for bootloaders, patch files, unique configurations for the Unmatched board, kernel, and setup script for the build environment. The meta-riscv is a RISC-V general layer that contains architecture specific metadata. The meta-openembedded contains many extra components for GUI, networking, file system, and additional meta layers. The openembedded-core layer uses bitbake, provides development tools, and enables support for QEMU.



2.14 BitBakeRecipe Flow

A typical BitBake process consists of executing the following steps. This is repeated for all of the recipes in a build. BitBake is able to manage dependencies when processing recipes. There are options to customize the build for each package such as source locations, versions, and additional tasks. There are about 3813 entries and over 10,000 tasks that are run for a FUSDK build.

1. Fetch
2. Unpack
3. Patch
4. Configure
5. Compile
6. Install
7. Package

2.15 Build History and Logs

To analyze build history, status, or to debug a failed build, you can inspect the build logs located in:

```
riscv-sifive/build/buildhistory/
```

In this `riscv-sifive/build/buildhistory/` directory you can execute `git log` command to show commit hashes and build information.

Each build task has execution scripts and output logs. These scripts and logs are located in a temp directory. For example, after FUSDK builds the C++ Boost library, the scripts and logs are available in the directory below and they have a file prefix of `run.` or `log.` respectively:

```
riscv-sifive/build/tmp-glibc/work/riscv64-oe-linux/boost/1.75.0-r0/temp
```

2.16 BitBakeDebug

When making changes or debugging build issues, it can be helpful to reproduce the BitBake environment. You can do this with the `devshell` command for example to setup the U-Boot build environment:

```
MACHINE=unmatched bitbake -c devshell u-boot
```

3

HiFive Unmatched Connectivity

Figure 7 below shows typical and optional connectivity of a HiFive Unmatched system. The supported connectivity options allows the user to monitor and control of the system.

Often there are multiple connectivity options used simultaneously with the HiFive Unmatched with each option having advantages and disadvantages.

The serial console has a low bandwidth, but it's simple and fundamental. The serial console shows early boot messages and provides the ability to interact with U-Boot. If the serial console is connected to the build machine, it can be convenient to cross reference code to information and error messages observed on the console.

An Ethernet connection with high bandwidth can be used for large file transfers such as `scp` or `sftp`. Ethernet is also essential for connecting the HiFive Unmatched to a local network (host build machine) and the internet.

The Graphical User Interface (GUI) option uses video (graphics card and display), USB keyboard, and USB mouse typical of a Personal Computer configuration. A GUI may be preferred for some applications such as vision processing or AI.

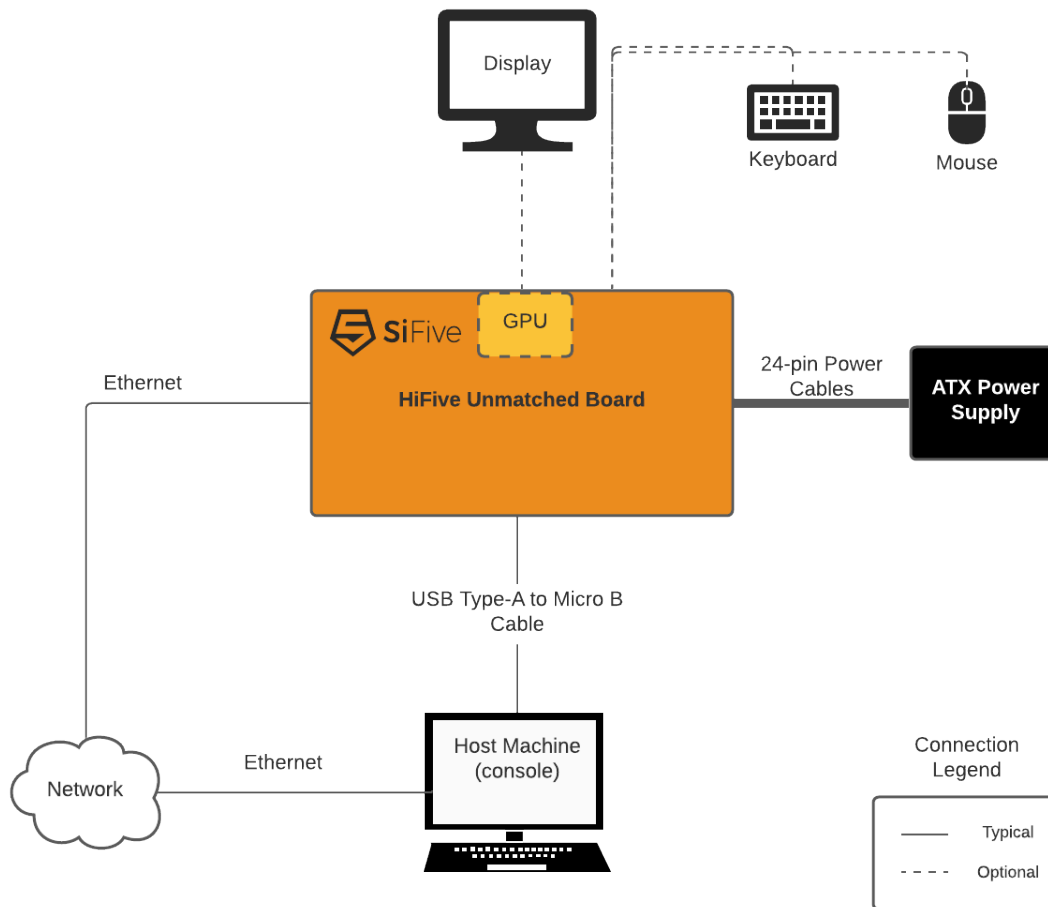


Figure 7: *HiFive Unmatched Connectivity*

3.1 Serial Console and Debugger Pod

There is USB to serial and USB to integrated debugger pod function on the HiFive Unmatched. The serial console as well as the debugger can be connected to a host machine with one USB cable. The connector on the HiFive Unmatched is a Micro-B USB connector, so a USB Type-A to Micro-B cable is used to connect a HiFive Unmatched to a host machine. The HiFive Unmatched console and debugger pod receives power from the host machine such that they are discoverable from the host machine without the main or standby power provided from ATX power supply.

There are two serial ports exposed to the host machine by the HiFive Unmatched board through the USB connection.

3.2 FTDI Linux plugdev Configuration

A Linux host machine is convenient because a Linux host machine can be used to build the FUSDK, use OpenOCD with the FU740-C000, function as a serial console for the HiFive Unmatched, transfer files to/from the HiFive Unmatched, and also develop applications for the HiFive Unmatched platform.

The FTDI FT2232 is the micro USB to serial converter on the HiFive Unmatched board. Typically, FTDI defaults to the `dialout` group which non-root users sometimes can't access. This can be a function of the Host machine's Linux distribution. To access the serial console from a Linux host machine as a non-root user in Linux, it may be necessary to change the group for the FT2232 to `plugdev` or alternatively to add a user to the `dialout` group. Below is an example of how to add a udev rule that assigns the FT2232 device to `plugdev`.

Note that the FTDI FT2232 is not only used for the console. It's also used for OpenOCD debugger.

To configure the group for the FT2232 device to `plugdev`, create a file at `/etc/udev/rules.d/99-hifive-unmatched.d`. `root/sudo` privileges are required to write this file to the directory. In this file, add the VID and PID for the FTDI device which can be found with the `lsusb` command while the HiFive Unmatched is connected to the host with USB:

```
/etc/udev/rules.d$ lsusb |grep FT2232
Bus 003 Device 012: ID 0403:6010 Future Technology Devices International, Ltd FT2232C/D/H Dual UART/FIFO IC
ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6010", MODE="664", GROUP="plugdev"
```

An example `.rules` file can be viewed in Appendix A.

3.3 FTDI Device Driver Management

To access the serial console from a Windows host machine, the FTDI FT2232 device on the HiFive Unmatched needs to be properly recognized by the OS. After the FTDI FT2232 device is recognized then Windows Applications such as Putty can function as a serial console.

Inspect Windows device manager for proper recognition of FT2232 virtual com port. Below are examples of correct and incorrect Windows configurations.

Figure 8 shows an example of incorrect Windows 10 Device Manager FTDI serial configuration:

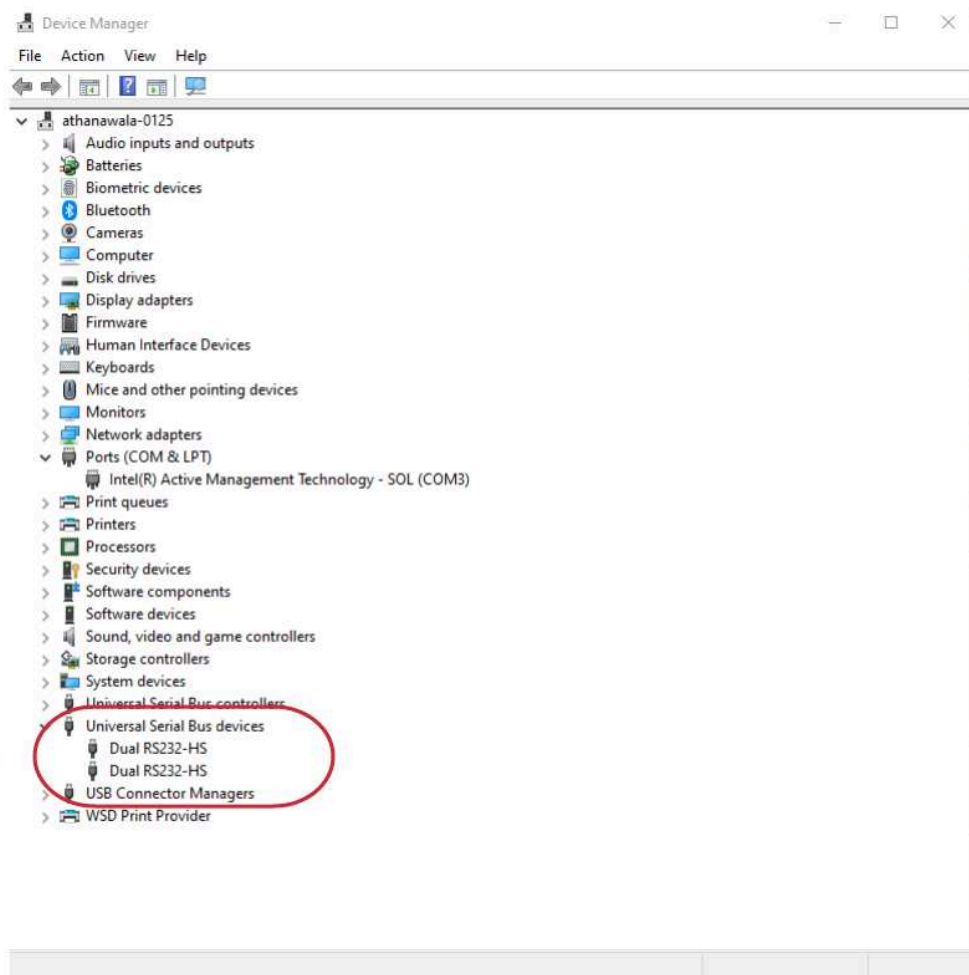


Figure 8: Example of Incorrect Windows 10 FTDI Serial Configuration

Figure 9 shows an example of correct Windows 10 Device Manager FTDI serial configuration:

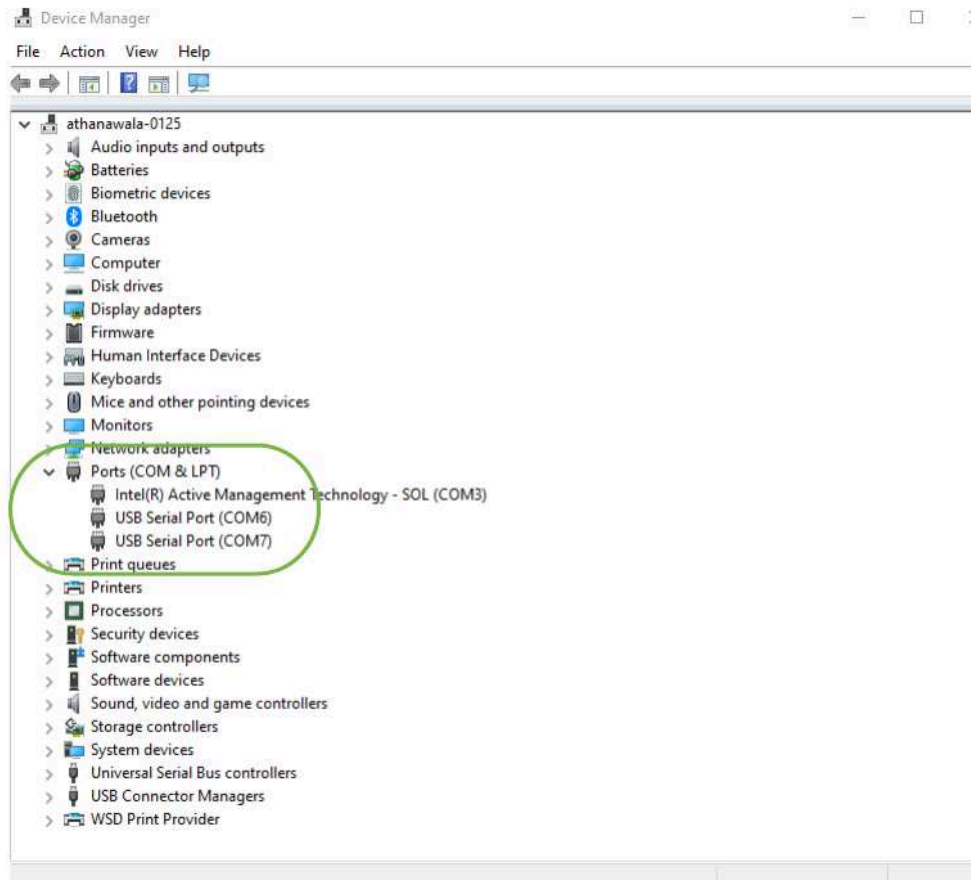


Figure 9: Example of Correct Windows 10 FTDI Serial Configuration

Link to Windows FTDI device driver. Install the latest driver (version 2.12.28 or later):

<https://www.ftdichip.com/Drivers/VCP.htm>

Note

To fix an incorrect configuration, the user may need to manually remove the device. To do this, make sure that all of the driver software is deleted by clicking on the option provided in the Windows Device Manager. Note that it may take multiple attempts before every node of the device driver is removed. Then, disconnect and reconnect the cable. After this, the 2 COM Ports should now show up in the Device Manager.

3.4 Determining the Network Configuration

The IP address for the Unleashed board may be determined externally by inspecting the router or switch that the board is connected. This networking device may have provided the HiFive Unmatched with an IP address. IP address of a HiFive Unmatched board can also be identified by using the `IP ARP` command along with the MAC address which is printed on boot by U-boot. Alternatively, the IP address can be determined from the board using a query of the Linux running on the HiFive Unmatched using the `ip addr` command.

```
root@unmatched:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 70:b3:d5:92:f4:73 brd ff:ff:ff:ff:ff:ff
    inet 10.0.4.25/24 brd 10.0.4.255 scope global dynamic noprefixroute eth0
        valid_lft 79648sec preferred_lft 79648sec
    inet6 fe80::914e:fa69:b6ee:bb6d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
4: wlp5s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether ac:12:03:9e:f2:eb brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.66/24 brd 192.168.0.255 scope global dynamic noprefixroute
wlp5s0
        valid_lft 79820sec preferred_lft 79820sec
    inet6 fe80::93b9:239b:a844:e920/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@unmatched:~#
```

The example above of the `ip addr` command displays the IP addresses of the system as well as other interface configuration information. The `nmccli` command can also be used to display network information for a system. Notice that this system has both wired (`eth0`) and wireless (`wlp5s0`) network connections. The Ethernet connection is available on the base HiFive Unmatched board and the wireless feature was added as an M.2 E expansion slot adapter.

The MAC address for the wired Ethernet connection is stored in the EEPROM on the board. Additionally, the MAC address is printed on a sticker placed on the HiFive Unmatched board for reference. You can use this information to assign a static IP address or to configure network boot for the HiFive Unmatched.

The hostname of HiFive Unmatched board can be determined by printing the hostname parameter to the console:

```
root@unmatched:~# cat /etc/hostname
unmatched
```

3.5 Secure Shell (SSH) Connection

For security reasons SiFive recommends changing the root password from the default `sifive`. This is especially critical if you enable network access to the board.

The SSH daemon is disabled by default for security reasons. SSH can be enabled through the serial console or the GUI interface. You can enable SSH connections to the HiFive Unmatched using `systemctl` utility with the following commands:

```
systemctl enable sshd.socket  
systemctl start sshd.socket
```

The `systemctl enable` command will enable SSH daemon on boot and the `systemctl start` will start the SSH daemon in the current boot session.

4

Benchmarks

Benchmarks can be used for debug and performance analysis. The FUSDK includes some benchmarks in the rootfs by default.

Coremark can be run in Linux by executing the following procedure:

1. Clone master branch of <https://github.com/sifive/freedom-e-sdk>
2. `cd freedom-e-sdk/software/coremark`
3. This command will make and run the benchmark:

```
make XCFLAGS="-O2 -fno-common -funroll-loops -finline-functions  
-funroll-all-loops -falign-functions=8 -falign-jumps=8 -falign-loops=8  
-finline-limit=1000 -mtune=sifive-7-series -ffast-math  
-fno-tree-loop-distribute-patterns --param fsm-scale-path-stmts=3"
```

4. The final step is normalizing the coremark score the coremark iterations to (typically) Iterations/MHz. You need to know the frequency of the core to do this normalization. See Section 4.2 to determine the core frequency. For example, if the core is running at 1GHz divide the reported Iterations/s by 1000.
5. This is expected output from coremark run. Note that you'll need the toolchain delivered from SiFive to reproduce these results:

```
2K performance run parameters for coremark.  
CoreMark Size      : 666  
Total ticks       : 21441  
Total time (secs): 21.441000  
Iterations/Sec    : 5130.357726  
Iterations        : 110000  
Compiler version  : GCC10.1.0  
Compiler flags    : -O2 -fno-common -funroll-loops -finline-functions -funroll-t  
Memory location   : Please put data memory location here  
                   (e.g. code in flash, data on heap etc)  
seedcrc          : 0xe9f5  
[0]crclist       : 0xe714  
[0]crcmatrix     : 0x1fd7  
[0]crcstate      : 0x8e3a
```

```
[0]crcfinal      : 0x33ff
Correct operation validated. See README.md for run and reporting rules.
CoreMark 1.0 : 5130.357726 / GCC10.1.0 -O2 -fno-common -funroll-loops -finlinep
```

Lmbench is a set of microbenchmarks that can be run by executing the `lmbench-run` script. This script will ask about ten questions regarding test configuration and then run the test as configured by the user. Note that some of the test configurations can take a long time to run. The results of this script are output to a file and the path to the file is indicated in the output results. The results of the questions used for configuration are also captured to a file so you will only see the questions the first time you run the `lmbench-run` script.

Sysbench is included in the FUSDK build, and it can be run from the command line: `sysbench`. There are many command line options for running the `sysbench` benchmark so choose the options of interest when you run the test.

Whetstone is included in the FUSDK rootsfs. The benchmark can be run by passing a loop number to the program such as: `whetstone 1000000`.

Dhrystone package is included in the FUSDK. The command to run the benchmark is: `dhry`. As part of the run procedure, you must provide a number of runs for the benchmark such as `10000000`.

Tinymembench is a memory benchmark included in the FUSDK. It can be executed with command: `tinymembench`.

Iperf is a network performance analysis tool that is included in the FUSDK and it's typically used with another networked device as part of a client/server test configuration. It can be executed with command: `iperf`.

4.1 Performance Debug

The FUSDK and SiFive hardware include tools to enable performance debugging. Some of these tools are:

1. Gprof GNU code profiler can be used to identify bottlenecks in code execution and it can be used in User-mode. The process involves compiling a program with this debug option, running the program to generate data, then inspecting the data generated.
2. Hardware Performance Monitors can be configured and sampled to evaluate performance. These counters provide metrics for performance of hardware and software and can be used to provide insight for performance debug. These counters can be read using the debugger.
3. Sampling the core cycle counter can be used to identify performance issues in User mode applications. Linux context switching can interfere with this measurement technique so anomalous data could be discarded and the `taskset` command can be used to bond a process to a single core to help with mitigate the context switching effects:

```
unsigned long read_cycles(void)
```



```
{
    unsigned long cycles;
    asm volatile ("rdcycle %0" : "=r" (cycles));
    return cycles;
}
```

4.2 Determining the Core and DDR Frequency

The initial boot frequency is 26MHz but U-Boot SPL configures the PLL's and switches the core clock to use the PLL. Currently, the PLL frequency is static, but it's expected to be configurable to setpoints from Linux over time.

The U74 core PLL frequency is configured in U-Boot SPL from the value in `fu740-c000-U-Boot.dtsi`, and this value for the 2021.03.01 FUSDK is 1.2GHz. The DDR controller is configured to run at 1846MT/s.

The core frequency and the DDR frequency can be confirmed by reading the PLL configuration registers and decoding the values. The `devmem2` command can be used to read memory mapped registers for the PLL configurations.

Table 5: Determining Core Frequency

Core			
Read PLL config for core	devmem2 0x10000004 w		
	0x820544C2		
PLL config bitfields	ref clock	26000000	
[5:0] pll_r PLL R Value	R	2	
[14:6] pll_f PLL F Value	F	275	
[17:15] pll_q PLL Q Value	Q	2	
[20:18] pll_range PLL Range Value	Range	1	
[23:21] Reserved			
24 pll_bypass RW PLL Bypass		1196000000	Hz
25 pll_fse_bypass PLL FSE Bypass			
[30:26] Reserved			
31 pll_lock X PLL Lock			

Table 6: Determining DDR Frequency

DDR			
Read PLL config for DDR	devmem2 0x1000000c w		
	0x820D1180		
PLL config bitfields	ref clock	26000000	
[5:0] pll_r PLL R Value	R	0	
[14:6] pll_f PLL F Value	F	70	
[17:15] pll_q PLL Q Value	Q	2	
[20:18] pll_range PLL Range Value	Range	3	
[23:21] Reserved			
24 pll_bypass RW PLL Bypass		923000000	Hz
25 pll_fse_bypass PLL FSE Bypass		1846000000	MT/s
[30:26] Reserved			
31 pll_lock X PLL Lock			

5

RISC-V Linux Software Structure

Software for Linux runtime environment is structured into layers with each layer having its own privilege level and responsibilities. Applications are responsible for the unique functions of the system. Application software runs in User mode privilege which is the least privileged mode. Applications have a limited view of memory and access to system resources such as peripherals. Applications must make system ecall (environment call) to the operating system to access additional resources. Applications make ecalls (environment calls) to the operating system using the Application Binary Interface layer. The operating system (OS - Linux) can manage multiple applications and is responsible for allocating resources to the applications such as managing memory or file systems and allocating processor time to applications. The operating system is running in Supervisor mode. Supervisor mode has more privilege than User mode but less than Machine-mode. The OS must make an ecall (environment call) to OpenSBI using the System Binary Interface for functions that require Machine-mode privilege. OpenSBI is a runtime service provider and not a bootloader. OpenSBI runs in Machine-mode and has the highest privilege of all software in this software stack.

Please see Chapter 7 for references to the RISC-V ABI and SBI specifications. The RISC-V privileged spec contains additional information about software stacks and privilege levels.

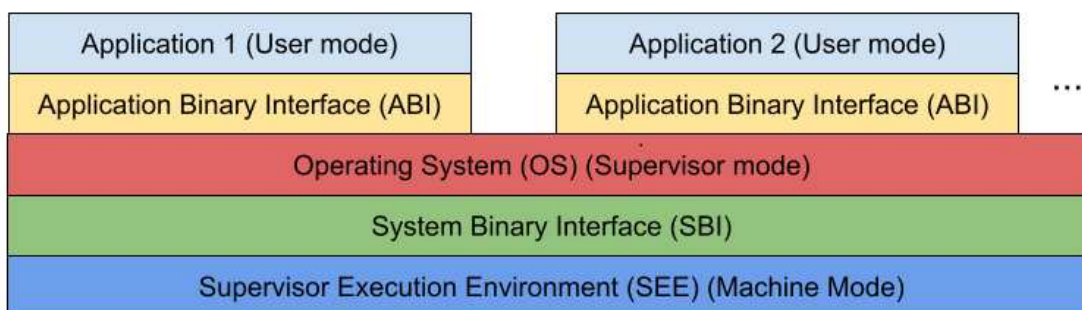


Figure 10: Privileged Software Stack

6

Crosstool-NG

The following section demonstrates a way to create RISC-V Linux binary executables on an x86 development host machine. Because this method doesn't use the FUSDK toolchain, there can be compatibility issues and, to avoid some of these issues, a statically linked program can be created with the `-static` parameter.

Following this example, the user will be able to install a cross compiler for RISC-V, compile a `hello.cpp` or `hello.c` and create an assembly listing of the binary file for inspection. Lastly, the user will be able to transfer the binary file to the Unmatched file system and execute the program in a RISC-V environment.

Note

The following instructions assume an x86 cross compile host machine running Linux and a user which has internet access and sudo privilege.

This while this section describes how to build a cross compiler toolchain for RISC-V 64-bit Linux but a prebuilt toolchain can also be downloaded from:

https://toolchains.bootlin.com/releases_riscv64.html

6.1 Creating a Location for Cross Compiler and Cloning `crosstool-ng`

First, create a directory for the cross compiler, and, within this directory, clone `crosstool-ng`:

```
mkdir x-tools
cd x-tools
git clone https://github.com/crosstool-ng/crosstool-ng
cd crosstool-ng
```

6.2 Build and install crosstool-ng

Next, after cloning the crosstool-ng repository, build and install the program:

```
./bootstrap
./configure
make -j$(getconf _NPROCESSORS_ONLN)
sudo make install
```

6.3 Build the rv64gc linux-gnu cross-toolchain

```
mkdir rv64gc-linux-gnu
cd rv64gc-linux-gnu
ct-ng riscv64-unknown-linux-gnu
ct-ng build
cd ..
```

A bare metal cross-toolchain can be created with:

```
mkdir rv64gc-elf
cd rv64gc-elf
ct-ng riscv64-unknown-elf
ct-ng build
cd ..
```

6.4 Option 1. C Language Hello World

Then, create a hello.c source code file to test the functionality of the cross compilation tool.

```
cat > hello.c
#include <stdio.h>
int main() {
printf("Hello World!");
return 0;
}
```

Use `ctrl-d` to close the file.

Then, use the following command to compile the source code. You can use the `file` command to inspect the binary and it should report: `...ELF... 64-bit... RISC-V...`

```
~/x-tools/riscv64-unknown-linux-gnu/bin/riscv64-unknown-linux-gnu-g++ ./hello.c -o
hello
file hello
```

To statically link the program the `-static` parameter can be used:

```
~/x-tools/riscv64-unknown-linux-gnu/bin/riscv64-unknown-linux-gnu-g++ -static
./hello.c -o hello
```

6.5 Option 2. C++ Language Hello World

Alternatively, create a `hello.cpp` source code file to test the functionality of the cross compilation tool.

```
cat > hello.cpp
#include <iostream>
int main() {
    std::cout << "Hello World!";
    return 0;
}
```

Use `ctrl-d` to close the file.

Then, use the following command to compile the source code. You can use the `file` command to inspect the binary and it should report: `...64-bit... RISC-V...`

6.6 Generate an Assembly Listing from the Binary File

```
~/x-tools/riscv64-unknown-linux-gnu/bin/riscv64-unknown-linux-gnu-objdump -DS ./hello
>& hello.lst
```

`hello.lst` will contain an assembly listing of the `hello` binary.

6.7 Execute the Binary from RISC-V Linux Environment

Lastly, transfer the `hello` executable file to the HiFive Unmatched file system (via `scp` or other methods), make the transferred file executable, then execute the file from the command line of a RISC-V host machine.

7

Additional Information

For more information about the HiFive Unmatched board, access the SiFive HiFive Unmatched board page:

<https://www.sifive.com/boards/hifive-unmatched>

This page provides access to the HiFive Unmatched Getting Started guide, the board schematics, bill of materials and a 3D model of the board itself.

To discuss the HiFive Unmatched board with the wider SiFive community, access the HiFive Unmatched category on the SiFive forums:

<https://forums.sifive.com/c/hifive-unmatched/16>

For information about the Freedom Unleashed SDK, access the GitHub page below. The README on this page outlines both required and optional software packages compatible with HiFive Unmatched. This page also describes some additional resources that can assist advanced users working with freedom-u-sdk.

<https://github.com/sifive/freedom-u-sdk>

To find projects and tools that support RISC-V, access the riscv-software-list GitHub repository. This page documents many of the projects that are contributing to the ever-growing RISC-V software ecosystem:

<https://github.com/riscv/riscv-software-list>

Ubuntu on HiFive Unmatched:

- <https://wiki.ubuntu.com/RISC-V> (official releases)
- <http://cdimage.ubuntu.com/ubuntu-server/daily-preinstalled/current/> (daily builds)
- <https://github.com/carlosedp/riscv-bringup> (roll your own tutorials)

Reference the Yocto, Poky, BitBake reference manual for useful information and context on the FUSDK build system:

<https://www.yoctoproject.org/docs/latest/ref-manual/ref-manual.html>

OpenOCD, GNU Debugger, and HiFive Unmatched OpenOCD config file:

- OpenOCD, GNU toolchain with gdb: <https://www.sifive.com/software>
- <https://github.com/sifive/freedom-e-sdk/blob/master/bsp/sifive-hifive-unleashed/openocd.cfg>

Mailing lists are one way to stay informed of developments to RISC-V software projects.

- RISC-V Linux Kernel: <http://lists.infradead.org/mailman/listinfo/linux-riscv>
- U-Boot: <https://lists.denx.de/listinfo/u-boot>
- OpenSBI: <https://lists.infradead.org/mailman/listinfo/opensbi>
- RISC-V: <https://riscv.org/mailling-lists/>

For direct access to the Linux RISC-V kernel please reference:

- Kernel: <https://git.kernel.org/pub/scm/linux/kernel/git/riscv/linux.git/>
- Patches: <https://patchwork.kernel.org/project/linux-riscv/list/>

U-Boot is a general boot loader used with a variety of architectures. Information about U-Boot can be found here:

<https://www.denx.de/wiki/U-Boot>

Supervisor Binary Interface Specification is part of the RISC-V Linux architecture. Information on this specification can be found here:

- OpenSBI spec: <https://github.com/riscv/riscv-sbi-doc/blob/master/riscv-sbi.adoc>

For the more information about the RISC-V Instruction Set Architecture (ISA):

<https://riscv.org/technical/specifications/>

RISC-V Processor Specific Application Binary Interface, calling convention:

<https://github.com/riscv/riscv-elf-psabi-doc/blob/master/riscv-elf.md>

8

Known Issues and Limitations

8.1 ZSBL Boot ROM Fail Messages are Sometimes Illegible

In some cases, the boot fail messages from ZSBL on the HiFive Unmatched board are illegible. This is because boot fail messages from ZSBL are output to the console at 89856 baud instead of the default 115200 baud.

These informative messages from ZSBL can be used to identify issues with booting in the first stage of boot. If your HiFive Unmatched is not booting and you're seeing garbled or unprintable characters on a console configured for the default 115200 8-n-1 console you can change the console baud rate to 89856 to attempt to get more debug information.

Below is a list of ZSBL Boot ROM error codes:

```
// Bit fields of error codes
#define ERROR_CODE_BOOTSTAGE (0xfUL << 60)
#define ERROR_CODE_TRAP (0xfUL << 56)
#define ERROR_CODE_ERRORCODE ((0x1UL << 56) - 1)
// Bit fields of mcause fields when compressed to fit into the errorcode field
#define ERROR_CODE_ERRORCODE_MCAUSE_INT (0x1UL << 55)
#define ERROR_CODE_ERRORCODE_MCAUSE_CAUSE ((0x1UL << 55) - 1)

#define ERROR_CODE_UNHANDLED_SPI_DEVICE 0x1
#define ERROR_CODE_UNHANDLED_BOOT_ROUTINE 0x2
#define ERROR_CODE_GPT_PARTITION_NOT_FOUND 0x3
#define ERROR_CODE_SPI_COPY_FAILED 0x4
#define ERROR_CODE_SD_CARD_CMD0 0x5
#define ERROR_CODE_SD_CARD_CMD8 0x6
#define ERROR_CODE_SD_CARD_ACMD41 0x7
#define ERROR_CODE_SD_CARD_CMD58 0x8
#define ERROR_CODE_SD_CARD_CMD16 0x9
#define ERROR_CODE_SD_CARD_CMD18 0xa
#define ERROR_CODE_SD_CARD_CMD18_CRC 0xb
#define ERROR_CODE_SD_CARD_UNEXPECTED_ERROR 0xc
#define ERROR_CODE_DMA_MEMCPY_FAILED 0xd
```

8.2 ASMedia USB 3.0 Controller Incompatible with Some Devices

Users have reported that some USB devices are non-functional when plugged into the USB ports on the HiFive Unmatched. Particularly, a Logitech BRIO Webcam was not recognized by the HiFive Unmatched board.

8.3 Micro-USB Port Fails to be Recognized by Host Machine

Users have reported that sometimes the micro-USB port for the serial console fails to be recognized by the host machine. This error can manifest as the HiFive Unmatched not appearing in the Device Manager on Windows machines.

To work around this issue on Windows machines, consider Section 3.3 above. This section provides tips for ensuring that the HiFive Unmatched board shows up in the Device Manager and that the HiFive Unmatched is recognized by the host machine.

8.4 Reboot Command not Supported

In the 2021.03.01 release, the `reboot` command does not work. This command is supposed to gracefully shutdown the HiFive Unmatched and reset the machine for a reboot. Until Freedom U SDK is updated to support this command, this command will not work.

8.5 SanDisk A2 SD Card Incompatibility

SanDisk™; micro SD cards that are marked with "A2" have been identified as causing errors and crashes when connected to the HiFive Unmatched board. Please avoid using the micro SD cards marked with "A2" to load the HiFive Unmatched firmware.

Other SanDisk micro SD cards that do not have the "A2" labeling have been tested and proved to function without problems.

8.6 NTFS Compatibility

NTFS device partitions won't work within the FUSDK Linux build. There are several other options to choose from including `ext2`, `ext4`, and `fat32`. There are some possible workarounds but in general it's recommended to avoid using NTFS formatted devices with Linux.

8.7 Default Configuration

Note that not all functionality is enabled in the default Open Embedded configuration. This is in contrast to a typical Linux distribution to keep it simple. One example of a feature that's not

implemented is SATA. If you need SATA functionality, you may be required to change the kernel configuration and rebuild.

8.8 U-Boot Save Environment

U-Boot for HiFive Unmatched board is a work in progress and the saveenv function isn't working yet. This will affect a user's ability to store non-volatile data relating to the U-Boot environment. This function may be enabled in a future FUSDK release.

8.9 Onboard Flash Memory Bootflow

Onboard flash is a potential boot source for U-Boot SPL and U-Boot. This option for boot flow hasn't been enabled in the 2021.03.01 FUSDK. This feature is expected to be implemented in the future.

8.10 Xfce File View Issue

An xfce issue has been reported relating to file viewer: https://bugzilla.xfce.org/show_bug.cgi?id=16465

Possible workaround: use list view or compact view when viewing files in GUI.

8.11 Mouse Lockup

Under stress testing intermittent mouse lockups have been reported. A USB re-plug of the mouse device is the most effective recovery method. A less reliable mouse recovery method is to: Ctrl + Alt + F<NR> console toggle.

8.12 Real Time Clock (RTC) Not Enabled

There is no support for the RTC function on the HiFive Unmatched yet. Time information can be acquired from an Ethernet or WiFi network. If a user isn't connected to a network where time information is available then some functions that depend on correct time may not work such as ssh.

Appendix A

Sample Log Files

This appendix provides log file samples for users to reference.

A.1 uBootBoot.log

```
U-Boot 2021.01 (Apr 09 2021 - 08:56:15 +0000)
```

```
CPU:   rv64imafdc
Model: SiFive HiFive Unmatched A00
DRAM:  16 GiB
MMC:   spi@10050000:mmc@0: 0
EEPROM: SiFive PCB EEPROM format v1
Serial number: SF105SZ210400068
PCB revision: 3
Ethernet MAC address: 70:b3:d5:92:f4:73
CRC:   cc7cb7a1
EEPROM dump: (0x25 bytes)
00: F1 5E 50 45 01 02 00 03 41 00 53 46 31 30 35 53
10: 5A 32 31 30 34 30 30 30 36 38 01 70 B3 D5 92 F4
20: 73 A1 B7 7C CC
found SiFive v1
In:    serial@10010000
Out:   serial@10010000
Err:   serial@10010000
Model: SiFive HiFive Unmatched A00
Net:   eth0: ethernet@10090000
Hit any key to stop autoboot:  0
PCIe Link up, Gen1

Device 0: unknown device
starting USB...
Bus xhci_pci: Register 4000840 NbrPorts 4
Starting the controller
USB XHCI 1.00
scanning bus xhci_pci for devices... 7 USB Device(s) found
       scanning usb for storage devices... 1 Storage Device(s) found

Device 0: Vendor: Kingston Rev: 1.00 Prod: DataTraveler G3
```

```

        Type: Removable Hard Disk
        Capacity: 15295.0 MB = 14.9 GB (31324160 x 512)
... is now current device
Scanning usb 0:3...
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
205 bytes read in 2 ms (99.6 KiB/s)
1:      OpenEmbedded-SiFive-HiFive-Unmatched
Retrieving file: /Image.gz
7249366 bytes read in 274 ms (25.2 MiB/s)
append: root=/dev/mmcblk0p4 rootfstype=ext4 rootwait console=ttySIF0,115200
earlycon=sbi
Retrieving file: /hifive-unmatched-a00.dtb
9806 bytes read in 2 ms (4.7 MiB/s)
  Uncompressing Kernel Image
Moving Image from 0x84000000 to 0x80200000, end=810b2000
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Using Device Tree in place at 0000000088000000, end 000000008800564d

Starting kernel ...

```

A.2 u-boot_help_and_printenv.log

```

U-Boot SPL 2021.01 (Apr 09 2021 - 08:56:15 +0000)
Trying to boot from MMC1

```

```

U-Boot 2021.01 (Apr 09 2021 - 08:56:15 +0000)

```

```

CPU:      rv64imafdc
Model: SiFive HiFive Unmatched A00
DRAM:     16 GiB
MMC:      spi@10050000:mmc@0: 0
EEPROM:   SiFive PCB EEPROM format v1
Serial number: SF105SZ210400068
PCB revision: 3
Ethernet MAC address: 70:b3:d5:92:f4:73
CRC:      cc7cb7a1
EEPROM dump: (0x25 bytes)
00: F1 5E 50 45 01 02 00 03 41 00 53 46 31 30 35 53
10: 5A 32 31 30 34 30 30 30 36 38 01 70 B3 D5 92 F4
20: 73 A1 B7 7C CC
found SiFive v1
In:       serial@10010000
Out:      serial@10010000
Err:      serial@10010000
Model: SiFive HiFive Unmatched A00
Net:      eth0: ethernet@10090000
Hit any key to stop autoboot:  2      0
=> help
?         - alias for 'help'
base      - print or set address offset
bdinfo    - print Board Info structure
blkcache  - block cache diagnostics and control
boot      - boot default, i.e., run 'bootcmd'

```

```
bootd      - boot default, i.e., run 'bootcmd'
bootefi    - Boots an EFI payload from memory
bootelf    - Boot from an ELF image in memory
booti      - boot Linux kernel 'Image' format from memory
bootm      - boot application image from memory
bootp      - boot image via network using BOOTP/TFTP protocol
bootvx     - Boot vxWorks from an ELF image
cmp        - memory compare
coninfo    - print console devices and information
cp         - memory copy
cpu        - display information about CPUs
crc32      - checksum calculation
dhcp       - boot image via network using DHCP/TFTP protocol
dm         - Driver model low level access
echo       - echo args to console
editenv    - edit environment variable
eeprom     - EEPROM sub-system
env        - environment handling commands
erase      - erase FLASH memory
exit       - exit script
ext2load   - load binary file from a Ext2 filesystem
ext2ls     - list files in a directory (default /)
ext4load   - load binary file from a Ext4 filesystem
ext4ls     - list files in a directory (default /)
ext4size   - determine a file's size
false     - do nothing, unsuccessfully
fatinfo    - print information about filesystem
fatload    - load binary file from a dos filesystem
fatls     - list files in a directory (default /)
fatmkdir   - create a directory
fatrm      - delete a file
fatsize    - determine a file's size
fatwrite   - write file into a dos filesystem
fdt        - flattened device tree utility commands
flinfo     - print FLASH memory information
fstype     - Look up a filesystem type
fstypes    - List supported filesystem types
go         - start application at address 'addr'
gpio       - query and control gpio pins
gpt        - GUID Partition Table
gzwrite    - unzip and write memory to block device
help       - print command description/usage
i2c       - I2C sub-system
iminfo     - print header information for application image
imxtract   - extract a part of a multi-image
itest      - return true/false on integer compare
ln         - Create a symbolic link
load       - load binary file from a filesystem
loadb     - load binary file over serial line (kermit mode)
loads     - load S-Record file over serial line
loadx     - load binary file over serial line (xmodem mode)
loady     - load binary file over serial line (ymodem mode)
loop       - infinite loop on address range
ls         - list files in a directory (default /)
lzmdec     - lzma uncompress a memory region
mac        - display and program the system ID and MAC addresses in EEPROM
md         - memory display
mdio      - MDIO utility commands
```

```

meminfo - display memory information
mii - MII utility commands
mm - memory modify (auto-incrementing address)
mmc - MMC sub system
mmcinfo - display MMC info
mw - memory write (fill)
nfs - boot image via network using NFS protocol
nm - memory modify (constant address)
nvme - NVM Express sub-system
panic - Panic with optional message
part - disk partition related commands
pci - list and access PCI Configuration Space
ping - send ICMP ECHO_REQUEST to network host
printenv - print environment variables
protect - enable or disable FLASH write protection
pwm - control pwm channels
pxe - commands to get and boot from pxe files
random - fill memory with random pattern
reset - Perform RESET of the CPU
run - run commands in an environment variable
save - save file to a filesystem
setenv - set environment variables
setexpr - set environment variable as the result of eval expression
sf - SPI flash sub-system
showvar - print local hushshell variables
size - determine a file's size
sleep - delay execution for some time
source - run script from memory
sysboot - command to get and boot from syslinux files
test - minimal test like /bin/sh
tftpboot - boot image via network using TFTP protocol
true - do nothing, successfully
unlz4 - lz4 uncompress a memory region
unzip - unzip a memory region
usb - USB sub-system
usbboot - boot from USB device
version - print monitor, compiler and linker version
=> printenv
arch=riscv
baudrate=115200
board=hifive_unmatched_fu740
board_name=hifive_unmatched_fu740
boot_a_script=load ${devtype} ${devnum}:${distro_bootpart} ${scriptaddr}
${prefix}${script}; source ${scriptaddr}
boot_efi_binary=load ${devtype} ${devnum}:${distro_bootpart} ${kernel_addr_r} efi/
boot/bootriscv64.efi; if fdt addr ${fdt_addr_r}; then bootefi ${kernel_addr_r}
${fdt_addr_r};else bootefi ${kernel_addr_r} ${fdtcontroladdr};fi
boot_efi_bootmgr=if fdt addr ${fdt_addr_r}; then bootefi bootmgr ${fdt_addr_r};else
bootefi bootmgr;fi
boot_extlinux=run settled_blue; sysboot ${devtype} ${devnum}:${distro_bootpart} any
${scriptaddr} ${prefix}${boot_syslinux_conf};
boot_net_usb_start=usb start
boot_pci_enum=pci enum
boot_prefixes=/ /boot/
boot_script_dhcp=boot.scr.uimg
boot_scripts=boot.scr.uimg boot.scr
boot_syslinux_conf=extlinux/extlinux.conf
boot_targets=nvme0 usb0 mmc0 pxe dhcp

```



```

bootcmd=run distro_bootcmd
bootcmd_dhcp=setenv devtype dhcp; run boot_net_usb_start; run boot_pci_enum; if dhcp
${scriptaddr} ${boot_script_dhcp}; then source ${scriptaddr}; fi;setenv efi_fdtfile
${fdtfile}; setenv efi_old_vci ${bootp_vci};setenv efi_old_arch ${bootp_arch};setenv
bootp_vci PXEClient:Arch:00027:UNDI:003000;setenv bootp_arch 0x1b;if dhcp
${kernel_addr_r}; then tftpboot ${fdt_addr_r} dtb/${efi_fdtfile};if fdt addr
${fdt_addr_r}; then bootefi ${kernel_addr_r} ${fdt_addr_r}; else bootefi
${kernel_addr_r} ${fdtcontroladdr};fi;fi;setenv bootp_vci ${efi_old_vci};setenv
bootp_arch ${efi_old_arch};setenv efi_fdtfile;setenv efi_old_arch;setenv efi_old_vci;
bootcmd_mmc0=devnum=0; run mmc_boot
bootcmd_nvme0=devnum=0; run nvme_boot
bootcmd_pxe=run boot_net_usb_start; run boot_pci_enum; dhcp; if pxe get; then pxe
boot; fi
bootcmd_usb0=devnum=0; run usb_boot
bootdelay=2
cpu=fu740
distro_bootcmd=setenv nvme_need_init; for target in ${boot_targets}; do run
bootcmd_${target}; done
efi_dtb_prefixes=/ /dtb/ /dtb/current/
ethaddr=70:b3:d5:92:f4:73
fdt_addr_r=0x88000000
fdt_high=0xffffffffffffffff
fdtcontroladdr=ff73d0e0
initrd_high=0xffffffffffffffff
kernel_addr_r=0x84000000
kernel_comp_addr_r=0x90000000
kernel_comp_size=0x4000000
load_efi_dtb=load ${devtype} ${devnum}:${distro_bootpart} ${fdt_addr_r}
${prefix}${efi_fdtfile}
mmc_boot=if mmc dev ${devnum}; then devtype=mmc; run scan_dev_for_boot_part; fi
nvme_boot=run boot_pci_enum; run nvme_init; if nvme dev ${devnum}; then devtype=nvme;
run scan_dev_for_boot_part; fi
nvme_init=if ${nvme_need_init}; then setenv nvme_need_init false; nvme scan; fi
partitions=name=loader1,start=17K,size=1M,type=${type_guid_gpt_loader1};name=loader2,size=4MB,type=
preboot=setenv fdt_addr ${fdtcontroladdr};fdt addr ${fdtcontroladdr};
pxefile_addr_r=0x88200000
ramdisk_addr_r=0x88300000
scan_dev_for_boot=echo Scanning ${devtype} ${devnum}:${distro_bootpart}...; for
prefix in ${boot_prefixes}; do run scan_dev_for_extlinux; run scan_dev_for_scripts;
done;run scan_dev_for_efi;
scan_dev_for_boot_part=part list ${devtype} ${devnum} -bootable devplist; env exists
devplist || setenv devplist 1; for distro_bootpart in ${devplist}; do if fstype
${devtype} ${devnum}:${distro_bootpart} bootfstype; then run scan_dev_for_boot; fi;
done; setenv devplist
scan_dev_for_efi=setenv efi_fdtfile ${fdtfile}; for prefix in ${efi_dtb_prefixes}; do
if test -e ${devtype} ${devnum}:${distro_bootpart} ${prefix}${efi_fdtfile}; then run
load_efi_dtb; fi;done;run boot_efi_bootmgr;if test -e ${devtype}
${devnum}:${distro_bootpart} efi/boot/bootriscv64.efi; then echo Found EFI removable
media binary efi/boot/bootriscv64.efi; run boot_efi_binary; echo EFI LOAD FAILED:
continuing...; fi; setenv efi_fdtfile
scan_dev_for_extlinux=if test -e ${devtype} ${devnum}:${distro_bootpart}
${prefix}${boot_syslinux_conf}; then echo Found ${prefix}${boot_syslinux_conf}; run
boot_extlinux; echo SCRIPT FAILED: continuing...; fi
scan_dev_for_scripts=for script in ${boot_scripts}; do if test -e ${devtype}
${devnum}:${distro_bootpart} ${prefix}${script}; then echo Found U-Boot script
${prefix}${script}; run boot_a_script; echo SCRIPT FAILED: continuing...; fi; done
scriptaddr=0x88100000
serial#=SF105SZ210400068

```

```

setled_blue=mw.l 0x10020024 0x0000ffff; mw.l 0x10020028 0x0000ffff; mw.l 0x1002002c
0x0
stderr=serial@10010000
stdin=serial@10010000
stdout=serial@10010000
type_guid_gpt_loader1=5B193300-FC78-40CD-8002-E86C45580B47
type_guid_gpt_loader2=2E54B353-1271-4842-806F-E436D6AF6985
type_guid_gpt_system=0FC63DAF-8483-4772-8E79-3D69D8477DE4
usb_boot=usb start; if usb dev ${devnum}; then devtype=usb; run
scan_dev_for_boot_part; fi
vendor=sifive

```

Environment size: 4645/126972 bytes

=>

A.3 LinuxBoot.log

U-Boot 2021.01 (Apr 09 2021 - 08:56:15 +0000)

```

CPU:   rv64imafdc
Model: SiFive HiFive Unmatched A00
DRAM:  16 GiB
MMC:   spi@10050000:mmc@0: 0
EEPROM: SiFive PCB EEPROM format v1
Serial number: SF105SZ210400068
PCB revision: 3
Ethernet MAC address: 70:b3:d5:92:f4:73
CRC:   cc7cb7a1
EEPROM dump: (0x25 bytes)
00: F1 5E 50 45 01 02 00 03 41 00 53 46 31 30 35 53
10: 5A 32 31 30 34 30 30 30 36 38 01 70 B3 D5 92 F4
20: 73 A1 B7 7C CC
found SiFive v1
In:    serial@10010000
Out:   serial@10010000
Err:   serial@10010000
Model: SiFive HiFive Unmatched A00
Net:   eth0: ethernet@10090000
Hit any key to stop autoboot:  2      1      0
PCIe Link up, Gen1

```

```

Device 0: Vendor: 0x144d Rev: 2B2QEXM7 Prod: S58SNS0N725267K
          Type: Hard Disk
          Capacity: 476940.0 MB = 465.7 GB (976773168 x 512)

```

... is now current device

Scanning nvme 0:3...

** Unable to read file / **

Failed to load '/'

libfdt fdt_check_header(): FDT_ERR_BADMAGIC

7 [r [999;999H [6n 8Scanning disk mmc@0.blk...

** Unrecognized filesystem type **

** Unrecognized filesystem type **

Scanning disk nvme#0.blk#0...

** Unrecognized filesystem type **

** Unrecognized filesystem type **

Found 10 disks

```

No EFI system partition
BootOrder not defined
EFI boot manager: Cannot load any image
starting USB...
Bus xhci_pci: Register 4000840 NbrPorts 4
Starting the controller
USB XHCI 1.00
scanning bus xhci_pci for devices... 7 USB Device(s) found
    scanning usb for storage devices... 1 Storage Device(s) found

Device 0: Vendor: Kingston Rev: 1.00 Prod: DataTraveler G3
        Type: Removable Hard Disk
        Capacity: 15295.0 MB = 14.9 GB (31324160 x 512)
... is now current device
Scanning usb 0:3...
** Unable to read file / **
Failed to load '/'
libfdt fdt_check_header(): FDT_ERR_BADMAGIC
BootOrder not defined
EFI boot manager: Cannot load any image
switch to partitions #0, OK
mmc0 is current device
Scanning mmc 0:3...
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
204 bytes read in 9 ms (21.5 KiB/s)
1:      OpenEmbedded-SiFive-HiFive-Unmatched
Retrieving file: /Image.gz
7247924 bytes read in 4732 ms (1.5 MiB/s)
append: root=/dev/mmcblk0p4 rootfstype=ext4 rootwait console=ttySIF0,115200 earlycon
Retrieving file: /hifive-unmatched-a00.dtb
10473 bytes read in 14 ms (730.5 KiB/s)
    Uncompressing Kernel Image
Moving Image from 0x84000000 to 0x80200000, end=810b2000
## Flattened Device Tree blob at 88000000
    Booting using the fdt blob at 0x88000000
    Using Device Tree in place at 0000000088000000, end 00000000880058e8

Starting kernel ...

[ 0.000000] Linux version 5.11.10 (oe-user@oe-host) (riscv64-oe-linux-gcc (GCC)
10.2.0, GNU ld (GNU Binutils) 2.36.1.20210209) #1 SMP Fri Apr 9 08:35:18 UTC 2021
[ 0.000000] OF: fdt: Ignoring memory range 0x80000000 - 0x80200000
[ 0.000000] earlycon: sifive0 at MMIO 0x0000000010010000 (options '')
[ 0.000000] printk: bootconsole [sifive0] enabled
[ 0.000000] efi: UEFI not found.
[ 0.000000] Zone ranges:
[ 0.000000]   DMA32    [mem 0x0000000080200000-0x00000000ffffffff]
[ 0.000000]   Normal  [mem 0x00000000100000000-0x0000000047ffffffff]
[ 0.000000] Movable zone start for each node
[ 0.000000] Early memory node ranges
[ 0.000000]   node    0: [mem 0x0000000080200000-0x0000000047ffffffff]
[ 0.000000] Initmem setup node 0 [mem 0x0000000080200000-0x0000000047ffffffff]
[ 0.000000]   DMA32 zone: 512 pages in unavailable ranges
[ 0.000000] SBI specification v0.2 detected
[ 0.000000] SBI implementation ID=0x1 Version=0x9
[ 0.000000] SBI v0.2 TIME extension detected
[ 0.000000] SBI v0.2 IPI extension detected

```

```
[ 0.000000] SBI v0.2 RFENCE extension detected
[ 0.000000] software IO TLB: mapped [mem 0x00000000fbfff000-0x00000000fffff000]
(64MB)
[ 0.000000] SBI v0.2 HSM extension detected
[ 0.000000] CPU with hartid=0 is not available
[ 0.000000] CPU with hartid=0 is not available
[ 0.000000] riscv: ISA extensions acdfim
[ 0.000000] riscv: ELF capabilities acdfim
[ 0.000000] percpu: Embedded 26 pages/cpu s67032 r8192 d31272 u106496
[ 0.000000] Built 1 zonelists, mobility grouping on. Total pages: 4136455
[ 0.000000] Kernel command line: root=/dev/mmcblk0p4 rootfstype=ext4 rootwait
console=ttySIF0,115200 earlycon
[ 0.000000] Dentry cache hash table entries: 2097152 (order: 12, 16777216 bytes,
linear)
[ 0.000000] Inode-cache hash table entries: 1048576 (order: 11, 8388608 bytes,
linear)
[ 0.000000] Sorting __ex_table...
[ 0.000000] mem auto-init: stack:off, heap alloc:off, heap free:off
[ 0.000000] Memory: 16422848K/16775168K available (8526K kernel code, 1834K
rwdata, 3927K rodata, 395K init, 350K bss, 352320K reserved, 0K cma-reserved)
[ 0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
[ 0.000000] rcu: Hierarchical RCU implementation.
[ 0.000000] rcu: RCU restricting CPUs from NR_CPUS=8 to nr_cpu_ids=4.
[ 0.000000] Tracing variant of Tasks RCU enabled.
[ 0.000000] rcu: RCU calculated value of scheduler-enlistment delay is 25 jiffies.
[ 0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=4
[ 0.000000] NR_IRQS: 64, nr_irqs: 64, preallocated irq: 0
[ 0.000000] CPU with hartid=0 is not available
[ 0.000000] riscv-intc: unable to find hart id for /cpus/cpu@0/interrupt-controller
[ 0.000000] riscv-intc: 64 local interrupts mapped
[ 0.000000] plic: interrupt-controller@c000000: mapped 69 interrupts with 4
handlers for 9 contexts.
[ 0.000000] random: get_random_bytes called from start_kernel+0x360/0x4ba with
crng_init=0
[ 0.000000] riscv_timer_init_dt: Registering clocksource cpuid [0] hartid [1]
[ 0.000000] clocksource: riscv_clocksource: mask: 0xffffffffffffffff max_cycles:
0x1d854df40, max_idle_ns: 3526361616960 ns
[ 0.000002] sched_clock: 64 bits at 1000kHz, resolution 1000ns, wraps every
2199023255500ns
[ 0.008501] Console: colour dummy device 80x25
[ 0.012846] Calibrating delay loop (skipped), value calculated using timer
frequency.. 2.00 BogoMIPS (lpj=4000)
[ 0.022849] pid_max: default: 32768 minimum: 301
[ 0.028270] Mount-cache hash table entries: 32768 (order: 6, 262144 bytes, linear)
[ 0.035840] Mountpoint-cache hash table entries: 32768 (order: 6, 262144 bytes,
linear)
[ 0.045158] rcu: Hierarchical SRCU implementation.
[ 0.049470] EFI services will not be available.
[ 0.054103] smp: Bringing up secondary CPUs ...
[ 0.060401] smp: Brought up 1 node, 4 CPUs
[ 0.066919] devtmpfs: initialized
[ 0.071867] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff,
max_idle_ns: 7645041785100000 ns
[ 0.080891] futex hash table entries: 1024 (order: 4, 65536 bytes, linear)
[ 0.088600] NET: Registered protocol family 16
[ 0.092706] thermal_sys: Registered thermal governor 'step_wise'
[ 0.104621] iommu: Default domain type: Translated
[ 0.114912] vgaarb: loaded
```

```

[ 0.117693] SCSI subsystem initialized
[ 0.121512] usbcore: registered new interface driver usbfs
[ 0.126716] usbcore: registered new interface driver hub
[ 0.131956] usbcore: registered new device driver usb
[ 0.137082] EDAC MC: Ver: 3.0.0
[ 0.140420] Advanced Linux Sound Architecture Driver Initialized.
[ 0.146702] clocksource: Switched to clocksource riscv_clocksource
[ 0.216391] VFS: Disk quotas dquot_6.6.0
[ 0.219666] VFS: Dquot-cache hash table entries: 512 (order 0, 4096 bytes)
[ 0.231791] NET: Registered protocol family 2
[ 0.235962] tcp_listen_portaddr_hash hash table entries: 8192 (order: 5, 131072
bytes, linear)
[ 0.244576] TCP established hash table entries: 131072 (order: 8, 1048576 bytes,
linear)
[ 0.256393] TCP bind hash table entries: 65536 (order: 8, 1048576 bytes, linear)
[ 0.267280] TCP: Hash tables configured (established 131072 bind 65536)
[ 0.273753] UDP hash table entries: 8192 (order: 6, 262144 bytes, linear)
[ 0.281049] UDP-Lite hash table entries: 8192 (order: 6, 262144 bytes, linear)
[ 0.288896] NET: Registered protocol family 1
[ 0.295462] RPC: Registered named UNIX socket transport module.
[ 0.300632] RPC: Registered udp transport module.
[ 0.305316] RPC: Registered tcp transport module.
[ 0.310004] RPC: Registered tcp NFSv4.1 backchannel transport module.
[ 0.316440] PCI: CLS 0 bytes, default 64
[ 0.321255] Initialise system trusted keyrings
[ 0.325138] workingset: timestamp_bits=62 max_order=22 bucket_order=0
[ 0.337168] NFS: Registering the id_resolver key type
[ 0.341481] Key type id_resolver registered
[ 0.345625] Key type id_legacy registered
[ 0.349705] nfs4filelayout_init: NFSv4 File Layout Driver Registering...
[ 0.356310] Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
[ 0.363664] 9p: Installing v9fs 9p2000 file system support
[ 0.369134] NET: Registered protocol family 38
[ 0.372827] Key type asymmetric registered
[ 0.376898] Asymmetric key parser 'x509' registered
[ 0.381801] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 252)
[ 0.389148] io scheduler mq-deadline registered
[ 0.393662] io scheduler kyber registered
[ 1.487668] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[ 1.532341] fu740-pcie e00000000.pcie: host bridge /soc/pcie@e00000000 ranges:
[ 1.538922] fu740-pcie e00000000.pcie:      IO 0x0060080000..0x006008ffff ->
0x0060080000
[ 1.547132] fu740-pcie e00000000.pcie:      MEM 0x0060090000..0x0070ffffff ->
0x0060090000
[ 1.555362] fu740-pcie e00000000.pcie:      MEM 0x2000000000..0x3fffffff ->
0x2000000000
[ 1.671529] fu740-pcie e00000000.pcie: invalid resource
[ 1.676006] fu740-pcie e00000000.pcie: iATU unroll: enabled
[ 1.681556] fu740-pcie e00000000.pcie: Detected iATU regions: 8 outbound, 8 inbound
[ 1.789232] fu740-pcie e00000000.pcie: Link up
[ 1.793050] fu740-pcie e00000000.pcie: PCI host bridge to bus 0000:00
[ 1.799347] pci_bus 0000:00: root bus resource [bus 00-ff]
[ 1.804809] pci_bus 0000:00: root bus resource [io 0x0000-0xffff] (bus address
[0x60080000-0x6008ffff])
[ 1.814272] pci_bus 0000:00: root bus resource [mem 0x60090000-0x70ffffff]
[ 1.821134] pci_bus 0000:00: root bus resource [mem 0x2000000000-0x3fffffff pref]
[ 1.828816] pci 0000:00:00.0: [f15e:0000] type 01 class 0x060400
[ 1.834781] pci 0000:00:00.0: reg 0x10: [mem 0x00000000-0x000fffff]

```

```

[ 1.841029] pci 0000:00:00.0: reg 0x38: [mem 0x00000000-0x0000ffff pref]
[ 1.847761] pci 0000:00:00.0: supports D1
[ 1.851709] pci 0000:00:00.0: PME# supported from D0 D1 D3hot
[ 1.858297] pci 0000:01:00.0: [1b21:2824] type 01 class 0x060400
[ 1.863608] pci 0000:01:00.0: enabling Extended Tags
[ 1.868598] pci 0000:01:00.0: PME# supported from D0 D3hot D3cold
[ 1.887247] pci 0000:01:00.0: bridge configuration invalid ([bus 00-00]),
reconfiguring
[ 1.894643] pci 0000:02:00.0: [1b21:2824] type 01 class 0x060400
[ 1.900555] pci 0000:02:00.0: enabling Extended Tags
[ 1.905546] pci 0000:02:00.0: PME# supported from D0 D3hot D3cold
[ 1.911836] pci 0000:02:02.0: [1b21:2824] type 01 class 0x060400
[ 1.917577] pci 0000:02:02.0: enabling Extended Tags
[ 1.922569] pci 0000:02:02.0: PME# supported from D0 D3hot D3cold
[ 1.928827] pci 0000:02:03.0: [1b21:2824] type 01 class 0x060400
[ 1.934600] pci 0000:02:03.0: enabling Extended Tags
[ 1.939594] pci 0000:02:03.0: PME# supported from D0 D3hot D3cold
[ 1.945851] pci 0000:02:04.0: [1b21:2824] type 01 class 0x060400
[ 1.951625] pci 0000:02:04.0: enabling Extended Tags
[ 1.956618] pci 0000:02:04.0: PME# supported from D0 D3hot D3cold
[ 1.962940] pci 0000:02:08.0: [1b21:2824] type 01 class 0x060400
[ 1.968649] pci 0000:02:08.0: enabling Extended Tags
[ 1.973642] pci 0000:02:08.0: PME# supported from D0 D3hot D3cold
[ 1.980284] pci 0000:02:00.0: bridge configuration invalid ([bus 00-00]),
reconfiguring
[ 1.987619] pci 0000:02:02.0: bridge configuration invalid ([bus 00-00]),
reconfiguring
[ 1.995604] pci 0000:02:03.0: bridge configuration invalid ([bus 00-00]),
reconfiguring
[ 2.003594] pci 0000:02:04.0: bridge configuration invalid ([bus 00-00]),
reconfiguring
[ 2.011593] pci 0000:02:08.0: bridge configuration invalid ([bus 00-00]),
reconfiguring
[ 2.020170] pci_bus 0000:03: busn_res: [bus 03-ff] end is updated to 03
[ 2.026315] pci 0000:04:00.0: [1b21:1142] type 00 class 0x0c0330
[ 2.032203] pci 0000:04:00.0: reg 0x10: [mem 0x00000000-0x00007fff 64bit]
[ 2.039122] pci 0000:04:00.0: PME# supported from D3cold
[ 2.052246] pci_bus 0000:04: busn_res: [bus 04-ff] end is updated to 04
[ 2.061252] pci 0000:05:00.0: [8086:2723] type 00 class 0x028000
[ 2.067135] pci 0000:05:00.0: reg 0x10: [mem 0x00000000-0x00003fff 64bit]
[ 2.074067] pci 0000:05:00.0: PME# supported from D0 D3hot D3cold
[ 2.091250] pci_bus 0000:05: busn_res: [bus 05-ff] end is updated to 05
[ 2.097247] pci 0000:06:00.0: [144d:a808] type 00 class 0x010802
[ 2.103135] pci 0000:06:00.0: reg 0x10: [mem 0x70400000-0x70403fff 64bit]
[ 2.119250] pci_bus 0000:06: busn_res: [bus 06-ff] end is updated to 06
[ 2.125250] pci 0000:07:00.0: [1002:6779] type 00 class 0x030000
[ 2.131140] pci 0000:07:00.0: reg 0x10: [mem 0x00000000-0xffffffff 64bit pref]
[ 2.138327] pci 0000:07:00.0: reg 0x18: [mem 0x00000000-0x0001ffff 64bit]
[ 2.145093] pci 0000:07:00.0: reg 0x20: initial BAR value 0x00000000 invalid
[ 2.152116] pci 0000:07:00.0: reg 0x20: [io size 0x0100]
[ 2.157524] pci 0000:07:00.0: reg 0x30: [mem 0x00000000-0x0001ffff pref]
[ 2.164202] pci 0000:07:00.0: enabling Extended Tags
[ 2.169250] pci 0000:07:00.0: supports D1 D2
[ 2.173489] pci 0000:07:00.0: 16.000 Gb/s available PCIe bandwidth, limited by 2.5
GT/s PCIe x8 link at 0000:02:08.0 (capable of 32.000 Gb/s with 2.5 GT/s PCIe x16 link)
[ 2.188605] pci 0000:07:00.0: vgaarb: VGA device added:
decodes=io+mem,owns=none,locks=none
[ 2.196923] pci 0000:07:00.1: [1002:aa98] type 00 class 0x040300

```

```

[ 2.202878] pci 0000:07:00.1: reg 0x10: [mem 0x00000000-0x00003fff 64bit]
[ 2.209685] pci 0000:07:00.1: enabling Extended Tags
[ 2.214671] pci 0000:07:00.1: supports D1 D2
[ 2.231250] pci_bus 0000:07: busn_res: [bus 07-ff] end is updated to 07
[ 2.237115] pci_bus 0000:02: busn_res: [bus 02-ff] end is updated to 07
[ 2.243713] pci_bus 0000:01: busn_res: [bus 01-ff] end is updated to 07
[ 2.250352] pci 0000:00:00.0: BAR 15: assigned [mem 0x200000000-0x200ffffff
64bit pref]
[ 2.258473] pci 0000:00:00.0: BAR 0: assigned [mem 0x60100000-0x601fffff]
[ 2.265249] pci 0000:00:00.0: BAR 14: assigned [mem 0x60200000-0x605fffff]
[ 2.272103] pci 0000:00:00.0: BAR 6: assigned [mem 0x60090000-0x6009ffff pref]
[ 2.279316] pci 0000:00:00.0: BAR 13: assigned [io 0x0000-0x0fff]
[ 2.285486] pci 0000:01:00.0: BAR 15: assigned [mem 0x200000000-0x200ffffff
64bit pref]
[ 2.293648] pci 0000:01:00.0: BAR 14: assigned [mem 0x60200000-0x605fffff]
[ 2.300509] pci 0000:01:00.0: BAR 13: assigned [io 0x0000-0x0fff]
[ 2.306682] pci 0000:02:08.0: BAR 15: assigned [mem 0x200000000-0x200ffffff
64bit pref]
[ 2.314841] pci 0000:02:02.0: BAR 14: assigned [mem 0x60200000-0x602fffff]
[ 2.321703] pci 0000:02:03.0: BAR 14: assigned [mem 0x60300000-0x603fffff]
[ 2.328565] pci 0000:02:04.0: BAR 14: assigned [mem 0x60400000-0x604fffff]
[ 2.335427] pci 0000:02:08.0: BAR 14: assigned [mem 0x60500000-0x605fffff]
[ 2.342288] pci 0000:02:08.0: BAR 13: assigned [io 0x0000-0x0fff]
[ 2.348457] pci 0000:02:00.0: PCI bridge to [bus 03]
[ 2.353424] pci 0000:04:00.0: BAR 0: assigned [mem 0x60200000-0x60207fff 64bit]
[ 2.360727] pci 0000:02:02.0: PCI bridge to [bus 04]
[ 2.365659] pci 0000:02:02.0: bridge window [mem 0x60200000-0x602fffff]
[ 2.372441] pci 0000:05:00.0: BAR 0: assigned [mem 0x60300000-0x60303fff 64bit]
[ 2.379742] pci 0000:02:03.0: PCI bridge to [bus 05]
[ 2.384679] pci 0000:02:03.0: bridge window [mem 0x60300000-0x603fffff]
[ 2.391461] pci 0000:06:00.0: BAR 0: assigned [mem 0x60400000-0x60403fff 64bit]
[ 2.398761] pci 0000:02:04.0: PCI bridge to [bus 06]
[ 2.403701] pci 0000:02:04.0: bridge window [mem 0x60400000-0x604fffff]
[ 2.410485] pci 0000:07:00.0: BAR 0: assigned [mem 0x200000000-0x200ffffff 64bit
pref]
[ 2.418567] pci 0000:07:00.0: BAR 2: assigned [mem 0x60500000-0x6051ffff 64bit]
[ 2.425862] pci 0000:07:00.0: BAR 6: assigned [mem 0x60520000-0x6053ffff pref]
[ 2.433056] pci 0000:07:00.1: BAR 0: assigned [mem 0x60540000-0x60543fff 64bit]
[ 2.440367] pci 0000:07:00.0: BAR 4: assigned [io 0x0000-0x00ff]
[ 2.446437] pci 0000:02:08.0: PCI bridge to [bus 07]
[ 2.451384] pci 0000:02:08.0: bridge window [io 0x0000-0x0fff]
[ 2.457466] pci 0000:02:08.0: bridge window [mem 0x60500000-0x605fffff]
[ 2.464239] pci 0000:02:08.0: bridge window [mem 0x200000000-0x200ffffff 64bit
pref]
[ 2.472320] pci 0000:01:00.0: PCI bridge to [bus 02-07]
[ 2.477528] pci 0000:01:00.0: bridge window [io 0x0000-0x0fff]
[ 2.483616] pci 0000:01:00.0: bridge window [mem 0x60200000-0x605fffff]
[ 2.490385] pci 0000:01:00.0: bridge window [mem 0x200000000-0x200ffffff 64bit
pref]
[ 2.498464] pci 0000:00:00.0: PCI bridge to [bus 01-07]
[ 2.503671] pci 0000:00:00.0: bridge window [io 0x0000-0x0fff]
[ 2.509750] pci 0000:00:00.0: bridge window [mem 0x60200000-0x605fffff]
[ 2.516525] pci 0000:00:00.0: bridge window [mem 0x200000000-0x200ffffff 64bit
pref]
[ 2.525163] pcieport 0000:00:00.0: AER: enabled with IRQ 46
[ 2.530309] pcieport 0000:01:00.0: enabling device (0000 -> 0003)
[ 2.536883] pcieport 0000:02:02.0: enabling device (0000 -> 0002)
[ 2.542620] pcieport 0000:02:03.0: enabling device (0000 -> 0002)

```

```

[ 2.548698] pcieport 0000:02:04.0: enabling device (0000 -> 0002)
[ 2.554809] pcieport 0000:02:08.0: enabling device (0000 -> 0003)
[ 2.560826] pci 0000:04:00.0: enabling device (0000 -> 0002)
[ 2.566371] pci 0000:07:00.1: D0 power state depends on 0000:07:00.0
[ 2.572783] switchtec: loaded.
[ 2.575854] L2CACHE: DataError @ 0x00000001.05040170
[ 2.575905] L2CACHE: No. of Banks in the cache: 4
[ 2.585233] L2CACHE: No. of ways per bank: 16
[ 2.589575] L2CACHE: Sets per bank: 512
[ 2.593396] L2CACHE: Bytes per cache block: 64
[ 2.597825] L2CACHE: Index of the largest way enabled: 15
[ 2.636159] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[ 2.642785] 10010000.serial: ttySIF0 at MMIO 0x10010000 (irq = 1, base_baud =
115200) is a SiFive UART v0
[ 2.651632] printk: console [ttySIF0] enabled
[ 2.651632] printk: console [ttySIF0] enabled
[ 2.660293] printk: bootconsole [sifive0] disabled
[ 2.660293] printk: bootconsole [sifive0] disabled
[ 2.670130] 10011000.serial: ttySIF1 at MMIO 0x10011000 (irq = 2, base_baud =
115200) is a SiFive UART v0
[ 2.693527] loop: module loaded
[ 2.723062] nvme nvme0: pci function 0000:06:00.0
[ 2.727075] nvme 0000:06:00.0: enabling device (0000 -> 0002)
[ 2.734379] Rounding down aligned max_sectors from 4294967295 to 4294967288
[ 2.735819] nvme nvme0: missing or invalid SUBNQN field.
[ 2.740820] db_root: cannot open: /etc/target
[ 2.745953] nvme nvme0: Shutdown timeout set to 8 seconds
[ 2.750821] sifive_spi 10040000.spi: mapped; irq=4, cs=1
[ 2.761626] spi-nor spi0.0: is25wp256 (32768 Kbytes)
[ 2.764156] nvme nvme0: 4/0/0 default/read/poll queues
[ 2.767941] sifive_spi 10050000.spi: mapped; irq=6, cs=1
[ 2.777405] libphy: Fixed MDIO Bus: probed
[ 2.780169] GPT:Primary header thinks Alt. header is not at the end of the disk.
[ 2.781416] macb 10090000.ethernet: Registered clk switch 'sifive-gemgxl-mgmt'
[ 2.788136] GPT:13918241 != 976773167
[ 2.788145] GPT:Alternate GPT header not at the end of the disk.
[ 2.788154] GPT:13918241 != 976773167
[ 2.788161] GPT: Use GNU Parted to correct GPT errors.
[ 2.795795] libphy: MACB_mii_bus: probed
[ 2.799026] nvme0n1: p1 p2 p3 p4
[ 2.806370] macb 10090000.ethernet eth0: Cadence GEM rev 0x10070109 at 0x10090000
irq 7 (70:b3:d5:92:f4:73)
[ 2.830847] e1000e: Intel(R) PRO/1000 Network Driver
[ 2.835645] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[ 2.841822] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[ 2.848077] ehci-pci: EHCI PCI platform driver
[ 2.852542] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[ 2.858675] ohci-pci: OHCI PCI platform driver
[ 2.863126] uhci_hcd: USB Universal Host Controller Interface driver
[ 2.869678] xhci_hcd 0000:04:00.0: xHCI Host Controller
[ 2.874771] xhci_hcd 0000:04:00.0: new USB bus registered, assigned bus number 1
[ 3.022812] xhci_hcd 0000:04:00.0: hcc params 0x0200e080 hci version 0x100 quirks
0x0000000010800410
[ 3.031933] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002,
bcdDevice= 5.11
[ 3.039452] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 3.046652] usb usb1: Product: xHCI Host Controller
[ 3.051514] usb usb1: Manufacturer: Linux 5.11.10 xhci-hcd

```



```
[ 3.056985] usb usb1: SerialNumber: 0000:04:00.0
[ 3.061936] hub 1-0:1.0: USB hub found
[ 3.065371] hub 1-0:1.0: 2 ports detected
[ 3.069610] xhci_hcd 0000:04:00.0: xHCI Host Controller
[ 3.074639] xhci_hcd 0000:04:00.0: new USB bus registered, assigned bus number 2
[ 3.081932] xhci_hcd 0000:04:00.0: Host supports USB 3.0 SuperSpeed
[ 3.088968] usb usb2: We don't know the algorithms for LPM for this host,
disabling LPM.
[ 3.096399] usb usb2: New USB device found, idVendor=1d6b, idProduct=0003,
bcdDevice= 5.11
[ 3.104552] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 3.111756] usb usb2: Product: xHCI Host Controller
[ 3.116617] usb usb2: Manufacturer: Linux 5.11.10 xhci-hcd
[ 3.122088] usb usb2: SerialNumber: 0000:04:00.0
[ 3.126955] hub 2-0:1.0: USB hub found
[ 3.130445] hub 2-0:1.0: 2 ports detected
[ 3.134785] usbcore: registered new interface driver usb-storage
[ 3.140540] usbcore: registered new interface driver usbserial_generic
[ 3.146955] usbserial: USB Serial support registered for generic
[ 3.153034] mousedev: PS/2 mouse device common for all mice
[ 3.158535] usbcore: registered new interface driver usbtouchscreen
[ 3.164785] i2c /dev entries driver
[ 3.168905] poweroff-gpio gpio-poweroff: gpio_poweroff_probe: pm_power_off
function already registered
[ 3.177524] poweroff-gpio: probe of gpio-poweroff failed with error -16
[ 3.184346] lm90 0-004c: supply vcc not found, using dummy regulator
[ 3.192188] EDAC DEVICE0: Giving out device to module Sifive ECC Manager
controller sifive_edac.0: DEV sifive_edac.0 (INTERRUPT)
[ 3.228400] mmc_spi spi1.0: SD/MMC host mmc0, no DMA, no WP, no poweroff, cd
polling
[ 3.236028] ledtrig-cpu: registered to indicate activity on CPUs
[ 3.241537] usbcore: registered new interface driver usbhid
[ 3.246937] usbhid: USB HID core driver
[ 3.251041] usbcore: registered new interface driver snd-usb-audio
[ 3.257626] NET: Registered protocol family 10
[ 3.262366] Segment Routing with IPv6
[ 3.265341] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 3.271707] NET: Registered protocol family 17
[ 3.275799] 9pnet: Installing 9P2000 support
[ 3.279891] Key type dns_resolver registered
[ 3.284287] Loading compiled-in X.509 certificates
[ 3.290278] ALSA device list:
[ 3.292502]   No soundcards found.
[ 3.296614] Waiting for root device /dev/mmcblk0p4...
[ 3.335615] mmc0: host does not support reading read-only switch, assuming
write-enable
[ 3.342885] mmc0: new SDHC card on SPI
[ 3.347997] mmcblk0: mmc0:0000 SM32G 29.7 GiB
[ 3.386659] GPT:Primary header thinks Alt. header is not at the end of the disk.
[ 3.393312] GPT:13918241 != 62333951
[ 3.396864] GPT:Alternate GPT header not at the end of the disk.
[ 3.402857] GPT:13918241 != 62333951
[ 3.406408] GPT: Use GNU Parted to correct GPT errors.
[ 3.411583] mmcblk0: p1 p2 p3 p4
[ 3.494720] usb 1-1: new high-speed USB device number 2 using xhci_hcd
[ 3.529945] EXT4-fs (mmcblk0p4): mounted filesystem with ordered data mode. Opts:
(null). Quota mode: none.
[ 3.539011] VFS: Mounted root (ext4 filesystem) readonly on device 179:4.
```

```
[ 3.548903] devtmpfs: mounted
[ 3.552380] Freeing unused kernel memory: 392K
[ 3.556087] Kernel memory protection not selected by kernel config.
[ 3.562330] Run /sbin/init as init process
[ 3.741691] usb 1-1: New USB device found, idVendor=0951, idProduct=1643,
bcdDevice= 1.00
[ 3.749118] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 3.756234] usb 1-1: Product: DataTraveler G3
[ 3.760576] usb 1-1: Manufacturer: Kingston
[ 3.764745] usb 1-1: SerialNumber: 001CC0EC350DFBA0970B2543
[ 3.885281] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 3.891205] scsi host0: usb-storage 1-1:1.0
[ 3.899192] usb 2-2: new SuperSpeed Gen 1 USB device number 2 using xhci_hcd
[ 3.930373] usb 2-2: New USB device found, idVendor=174c, idProduct=3074,
bcdDevice= 0.01
[ 3.937792] usb 2-2: New USB device strings: Mfr=2, Product=3, SerialNumber=1
[ 3.944913] usb 2-2: Product: AS2107
[ 3.948471] usb 2-2: Manufacturer: ASMedia
[ 3.952553] usb 2-2: SerialNumber: USB2.0 Hub
[ 4.025972] hub 2-2:1.0: USB hub found
[ 4.029665] hub 2-2:1.0: 4 ports detected
[ 4.118722] usb 1-2: new high-speed USB device number 3 using xhci_hcd
[ 4.572667] usb 1-2: New USB device found, idVendor=174c, idProduct=2074,
bcdDevice= 0.01
[ 4.580094] usb 1-2: New USB device strings: Mfr=2, Product=3, SerialNumber=1
[ 4.587210] usb 1-2: Product: AS2107
[ 4.590771] usb 1-2: Manufacturer: ASMedia
[ 4.594852] usb 1-2: SerialNumber: USB2.0 Hub
[ 4.668107] hub 1-2:1.0: USB hub found
[ 4.671278] hub 1-2:1.0: 4 ports detected
[ 5.051011] random: fast init done
[ 5.053779] scsi 0:0:0:0: Direct-Access Kingston DataTraveler G3 1.00 PQ: 0
ANSI: 2
[ 5.062434] sd 0:0:0:0: [sda] 31324160 512-byte logical blocks: (16.0 GB/14.9 GiB)
[ 5.069490] sd 0:0:0:0: [sda] Write Protect is off
[ 5.074280] sd 0:0:0:0: [sda] No Caching mode page found
[ 5.074706] usb 1-2.1: new low-speed USB device number 4 using xhci_hcd
[ 5.079369] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 5.126400] GPT:Primary header thinks Alt. header is not at the end of the disk.
[ 5.133045] GPT:13918241 != 31324159
[ 5.136599] GPT:Alternate GPT header not at the end of the disk.
[ 5.142591] GPT:13918241 != 31324159
[ 5.146152] GPT: Use GNU Parted to correct GPT errors.
[ 5.151321] sda: sda1 sda2 sda3 sda4
[ 5.156704] sd 0:0:0:0: [sda] Attached SCSI removable disk
[ 5.277599] usb 1-2.1: New USB device found, idVendor=04f2, idProduct=0939,
bcdDevice= 1.00
[ 5.285204] usb 1-2.1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 5.292493] usb 1-2.1: Product: USB Optical Mouse
[ 5.297182] usb 1-2.1: Manufacturer: PixArt
[ 5.374508] input: PixArt USB Optical Mouse as /devices/platform/soc/
e00000000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/0000:02:02.0/0000:04:00.0/usb1/
1-2/1-2.1/1-2.1:1.0/0003:04F2:0939.0001/input/input0
[ 5.392157] hid-generic 0003:04F2:0939.0001: input: USB HID v1.11 Mouse [PixArt
USB Optical Mouse] on usb-0000:04:00.0-2.1/input0
[ 5.502713] usb 1-2.2: new low-speed USB device number 5 using xhci_hcd
[ 5.686610] usb 1-2.2: New USB device found, idVendor=1a2c, idProduct=2124,
bcdDevice= 1.10
```

```

[ 5.694213] usb 1-2.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 5.701505] usb 1-2.2: Product: USB Keyboard
[ 5.705759] usb 1-2.2: Manufacturer: SEM
[ 5.830414] input: SEM USB Keyboard as /devices/platform/soc/e00000000.pcie/
pci0000:00/0000:00:00.0/0000:01:00.0/0000:02:02.0/0000:04:00.0/usb1/1-2/1-2.2/
1-2.2:1.0/0003:1A2C:2124.0002/input/input1
[ 5.907011] hid-generic 0003:1A2C:2124.0002: input: USB HID v1.10 Keyboard [SEM
USB Keyboard] on usb-0000:04:00.0-2.2/input0
[ 5.920929] input: SEM USB Keyboard Consumer Control as /devices/platform/soc/
e00000000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/0000:02:02.0/0000:04:00.0/usb1/
1-2/1-2.2/1-2.2:1.1/0003:1A2C:2124.0003/input/input2
[ 5.998926] input: SEM USB Keyboard System Control as /devices/platform/soc/
e00000000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/0000:02:02.0/0000:04:00.0/usb1/
1-2/1-2.2/1-2.2:1.1/0003:1A2C:2124.0003/input/input3
[ 6.017090] hid-generic 0003:1A2C:2124.0003: input: USB HID v1.10 Device [SEM USB
Keyboard] on usb-0000:04:00.0-2.2/input1
[ 6.126720] usb 1-2.4: new full-speed USB device number 6 using xhci_hcd
[ 6.410454] usb 1-2.4: New USB device found, idVendor=8087, idProduct=0029,
bcdDevice= 0.01
[ 6.418066] usb 1-2.4: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[ 7.972689] systemd[1]: System time before build time, advancing clock.
[ 8.534585] systemd[1]: systemd 247.3+ running in system mode. (+PAM -AUDIT
-SELINUX +IMA -APPARMOR -SMACK +SYSVINIT +UTMP -LIBCRYPTSETUP -GCRYPT -GNUTLS +ACL
+XZ -LZ4 -ZSTD -SECCOMP +BLKID -ELFUTILS +KMOD -IDN2 -IDN -PCRE2
default-hierarchy=hybrid)
[ 8.556175] systemd[1]: Detected architecture riscv64.

```

Welcome to [1mFreedomUSDK 2021.03.01 (2021March) [0m!

```

[ 8.664100] systemd[1]: Set hostname to <unmatched>.
[ 10.694420] systemd[1]: Queued start job for default target Graphical Interface.
[ 10.706897] random: systemd: uninitialized urandom read (16 bytes read)
[ 10.727564] systemd[1]: Created slice system-getty.slice.
[ [0;32m OK [0m] Created slice [0;1;39msystem-getty.slice [0m.
[ 10.738592] random: systemd: uninitialized urandom read (16 bytes read)
[ 10.745285] systemd[1]: Created slice system-modprobe.slice.
[ [0;32m OK [0m] Created slice [0;1;39msystem-modprobe.slice [0m.
[ 10.756715] random: systemd: uninitialized urandom read (16 bytes read)
[ 10.763598] systemd[1]: Created slice system-serial\x2dgetty.slice.
[ [0;32m OK [0m] Created slice [0;1;39msystem-serial\x2dgetty.slice [0m.
[ 10.776751] systemd[1]: Created slice User and Session Slice.
[ [0;32m OK [0m] Created slice [0;1;39mUser and Session Slice [0m.
[ 10.788424] systemd[1]: Started Dispatch Password Requests to Console Directory
Watch.
[ [0;32m OK [0m] Started [0;1;39mDispatch Password ...ts to Console Directory Watch
[0m.
[ 10.804601] systemd[1]: Started Forward Password Requests to Wall Directory Watch.
[ [0;32m OK [0m] Started [0;1;39mForward Password R...uests to Wall Directory Watch
[0m.
[ 10.820654] systemd[1]: Reached target Paths.
[ [0;32m OK [0m] Reached target [0;1;39mPaths [0m.
[ 10.829295] systemd[1]: Reached target Remote File Systems.
[ [0;32m OK [0m] Reached target [0;1;39mRemote File Systems [0m.
[ 10.841075] systemd[1]: Reached target Slices.
[ [0;32m OK [0m] Reached target [0;1;39mSlices [0m.
[ 10.850964] systemd[1]: Reached target Swap.
[ [0;32m OK [0m] Reached target [0;1;39mSwap [0m.
[ 10.859460] systemd[1]: Reached target System Time Set.

```

```

[ [0;32m OK [0m] Reached target [0;1;39mSystem Time Set [0m.
[ 10.870832] systemd[1]: Reached target System Time Synchronized.
[ [0;32m OK [0m] Reached target [0;1;39mSystem Time Synchronized [0m.
[ 10.982231] systemd[1]: Listening on RPCbind Server Activation Socket.
[ [0;32m OK [0m] Listening on [0;1;39mRPCbind Server Activation Socket [0m.
[ 10.995850] systemd[1]: Reached target RPC Port Mapper.
[ [0;32m OK [0m] Reached target [0;1;39mRPC Port Mapper [0m.
[ 11.006902] systemd[1]: Listening on initctl Compatibility Named Pipe.
[ [0;32m OK [0m] Listening on [0;1;39minitctl Compatibility Named Pipe [0m.
[ 11.023107] systemd[1]: Condition check resulted in Journal Audit Socket being
skipped.
[ 11.030923] systemd[1]: Listening on Journal Socket (/dev/log).
[ [0;32m OK [0m] Listening on [0;1;39mJournal Socket (/dev/log) [0m.
[ 11.043719] systemd[1]: Listening on Journal Socket.
[ [0;32m OK [0m] Listening on [0;1;39mJournal Socket [0m.
[ 11.054054] systemd[1]: Listening on udev Control Socket.
[ [0;32m OK [0m] Listening on [0;1;39mudev Control Socket [0m.
[ 11.065416] systemd[1]: Listening on udev Kernel Socket.
[ [0;32m OK [0m] Listening on [0;1;39mudev Kernel Socket [0m.
[ 11.076745] systemd[1]: Listening on User Database Manager Socket.
[ [0;32m OK [0m] Listening on [0;1;39mUser Database Manager Socket [0m.
[ 11.089724] systemd[1]: Condition check resulted in Huge Pages File System being
skipped.
[ 11.099424] systemd[1]: Mounting POSIX Message Queue File System...
Mounting [0;1;39mPOSIX Message Queue File System [0m...
[ 11.113637] systemd[1]: Mounting Kernel Debug File System...
Mounting [0;1;39mKernel Debug File System [0m...
[ 11.126158] systemd[1]: Mounting Kernel Trace File System...
Mounting [0;1;39mKernel Trace File System [0m...
[ 11.142159] systemd[1]: Mounting Temporary Directory (/tmp)...
Mounting [0;1;39mTemporary Directory (/tmp) [0m...
[ 11.156010] systemd[1]: Starting Create list of static device nodes for the
current kernel...
Starting [0;1;39mCreate list of st...odes for the current kernel [0m...
[ 11.173865] systemd[1]: Starting Load Kernel Module configs...
Starting [0;1;39mLoad Kernel Module configs [0m...
[ 11.186899] systemd[1]: Starting Load Kernel Module drm...
Starting [0;1;39mLoad Kernel Module drm [0m...
[ 11.199490] systemd[1]: Starting Load Kernel Module fuse...
Starting [0;1;39mLoad Kernel Module fuse [0m...
[ 11.212246] systemd[1]: Starting RPC Bind...
Starting [0;1;39mRPC Bind [0m...
[ 11.222591] systemd[1]: Starting File System Check on Root Device...
Starting [0;1;39mFile System Check on Root Device [0m...
[ 11.440275] systemd[1]: Condition check resulted in Load Kernel Modules being
skipped.
[ 11.449621] systemd[1]: Mounting NFS configuration filesystem...
Mounting [0;1;39mNFS [ 11.457704] systemd[1]: Starting Apply Kernel
Variables...
configuration filesystem [0m...
Starting [0;1;39mApply Kernel[ 11.469721] systemd[1]: Starting Coldplug
All udev Devices...
Variables [0m...
Starting [0;1;39mColdplug All udev Devices [0m...
[ 11.485000] systemd[1]: Mounted POSIX Message Queue File System.
[ 11.490836] systemd[1]: Mounted Kernel Debug File System.
[ [0;32m[ 11.496376] systemd[1]: Mounted Kernel Trace File System.
[ 11.502107] systemd[1]: Mounted Temporary Directory (/tmp).

```

```

OK [[ 11.508474] systemd[1]: Finished Create list of static device nodes for
the current kernel.
0m] Moun[ 11.517175] systemd[1]: modprobe@configfs.service: Succeeded.
ted [0;[ 11.523414] systemd[1]: Finished Load Kernel Module configfs.
1;39mPOS[ 11.529965] systemd[1]: modprobe@drm.service: Succeeded.
IX Messa[ 11.535829] systemd[1]: Finished Load Kernel Module drm.
[ 11.541516] systemd[1]: Mounted NFSD configuration filesystem.
ge Queue File System [0m.
[ [0;[ 11.549851] systemd[1]: Mounting Kernel Configuration File System...
32m OK [0m] Mounted [ 11.558799] systemd[1]: Mounted Kernel Configuration File
System.
[0;1;39mKernel Debug File System [0m.
[ [0;32m OK [0m] Mounted [0;1;39mKernel Trace File System [0m.
[ [0;32m OK [0m] Mounted [0;1;39mTemporary Directory (/tmp) [0m.
[ [0;32m OK [0m] Finished [0;1;39mCreate list of st... nodes for the current kernel
[0m.
[ [0;32m OK [0m] Finished [0;1;39mLoad Kernel Module configfs [0m.
[ [0;32m OK [0m] Finished [0;1;39mLoad Kernel Module drm [0m.
[ [0;32m OK [0m] Mounted [0;1;39mNFSD configuration filesystem [0m.
Mounting [0;1;39mKernel Configuration File System [0m...
[ [0;32m OK [0m] Mounted [0;1;39mKernel Configuration File System [0m.
[ 12.901841] systemd[1]: Finished Apply Kernel Variables.
[ [0;32m OK [0m] Finished [0;1;39mApply Kernel Variables [0m.
[ 13.196120] fuse: init (API version 7.33)
[ 13.201905] systemd[1]: modprobe@fuse.service: Succeeded.
[ 13.207388] systemd[1]: Finished Load Kernel Module fuse.
[ [0;32m OK [0m] Finished [0;1;39mLo[ 13.215616] systemd[1]: Mounting FUSE
Control File System...
ad Kernel Module fuse [0m.
[ 13.224485] systemd[1]: Mounted FUSE Control File System.
Mounting [0;1;39mFUSE Control File System [0m...
[ [0;32m OK [0m] Mounted [0;1;39mFUSE Control File System [0m.
[ 13.605803] systemd[1]: Started RPC Bind.
[ [0;32m OK [0m] Started [0;1;39mRPC Bind [0m.
[ 14.120142] systemd[1]: Finished Coldplug All udev Devices.
[ [0;32m OK [0m] Finished [0;1;39mColdplug All udev Devices [0m.
[ 14.809939] systemd[1]: Finished File System Check on Root Device.
[ [0;32m OK [0m] Finished [0;1;39mFile System Check on Root Device [0m.
[ 14.825149] systemd[1]: Starting Remount Root and Kernel File Systems...
Starting [0;1;39mRemount Root and Kernel File Systems [0m...
[ 14.857513] EXT4-fs (mmcblk0p4): re-mounted. Opts: (null). Quota mode: none.
[ 14.867065] systemd[1]: Finished Remount Root and Kernel File Systems.
[ [0;32m OK [0m] Finished [0;1;39mRemount Root and Kernel File Systems [0m.
[ 14.887759] systemd[1]: Condition check resulted in Rebuild Hardware Database
being skipped.
[ 14.895709] systemd[1]: Condition check resulted in Platform Persistent Storage
Archival being skipped.
[ 14.905029] systemd[1]: Condition check resulted in Create System Users being
skipped.
[ 14.915298] systemd[1]: Starting Create Static Device Nodes in /dev...
Starting [0;1;39mCreate Static Device Nodes in /dev [0m...
[ 15.068057] systemd[1]: Finished Create Static Device Nodes in /dev.
[ [0;32m OK [0m] Finished [0;1;39mCreate Static Device Nodes in /dev [0m.
[ 15.080957] systemd[1]: Reached target Local File Systems (Pre).
[ [0;32m OK [0m] Reached target [0;1;39mLocal File Systems (Pre) [0m.
[ 15.099132] systemd[1]: Mounting /var/volatile...
Mounting [0;1;39m/var/volatile [0m...
[ 15.110989] systemd[1]: Started Entropy Daemon based on the HAVEGE algorithm.

```

```

[ 0;32m OK [0m] Started [0;1;39mEntropy Daemon based on the HAVEGE algorithm [0m.
[ 15.130291] systemd[1]: Starting Journal Service...
    Starting [0;1;39mJournal Service [0m...
[ 15.144065] systemd[1]: Starting Rule-based Manager for Device Events and Files...
    Starting [0;1;39mRule-based Manage...for Device Events and Files [0m...
[ 15.161833] systemd[1]: Mounted /var/volatile.
[ 0;32m OK [0m] Mounted [0;1;39m/var/volatile [0m.
[ 15.171748] systemd[1]: Condition check resulted in Bind mount volatile /var/cache
being skipped.
[ 15.180023] systemd[1]: Condition check resulted in Bind mount volatile /var/lib
being skipped.
[ 15.191234] systemd[1]: Starting Load/Save Random Seed...
    Starting [0;1;39mLoad/Save Random Seed [0m...
[ 15.201712] systemd[1]: Condition check resulted in Bind mount volatile /var/spool
being skipped.
[ 15.210116] systemd[1]: Condition check resulted in Bind mount volatile /srv being
skipped.
[ 15.588373] systemd[1]: Started Journal Service.
[ 0;32m OK [0m] Started [0;1;39mJournal Service [0m.
    Starting [0;1;39mFlush Journal to Persistent Storage [0m...
[ 15.660422] systemd-journal[139]: Received client request to flush runtime
journal.
[ 0;32m OK [0m] Finished [0;1;39mFlush Journal to Persistent Storage [0m.
[ 0;32m OK [0m] Started [0;1;39mRule-based Manager for Device Events and Files
[0m.
[ 16.859729] random: crng init done
[ 16.862374] random: 7 urandom warning(s) missed due to ratelimiting
[ 0;32m OK [0m] Finished [0;1;39mLoad/Save Random Seed [0m.
[ 0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3 (9s / 1min 30s)
M
[K [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3 (9s /
1min 30s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3
(10s / 1min 30s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3
(10s / 1min 30s)
[ 21.238764] usbcore: registered new interface driver uas
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3
(10s / 1min 30s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3
(11s / 1min 30s)
M
[K [0;31m* [0;1;31m* [0m] A start job is running for /dev/mmcblk0p3 (12s / 1min
30s)
M
[K [0;31m* [0m] A start job is running for /dev/mmcblk0p3 (12s / 1min 30s)
M
[K [0;31m* [0;1;31m* [0m] A start job is running for /dev/mmcblk0p3 (13s / 1min
30s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3
(13s / 1min 30s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /dev/mmcblk0p3

```

```

(14s / 1min 30s)
M
[K] [0;32m OK [0m] Created slice [0;1;39msystem-systemd\x2dfsck.slice [0m.
[K] [0;32m OK [0m] Found device [0;1;39mSamsung SSD 970 EVO Plus 500GB root [0m.
[ [0;32m OK [0m] Found device [0;1;39mDataTraveler_G3 root [0m.
    Starting [0;1;39mFile System Check on /dev/nvme0n1p4 [0m...
    Starting [0;1;39mFile System Check on /dev/sda4 [0m...
[ [0;32m OK [0m] Finished [0;1;39mFile System Check on /dev/nvme0n1p4 [0m.
    Mounting [0;1;39m/run/media/nvme0n1p4 [0m...
[ 26.803665] EXT4-fs (nvme0n1p4): mounted filesystem with ordered data mode. Opts:
(null). Quota mode: none.
[ [0;32m OK [0m] Mounted [0;1;39m/run/media/nvme0n1p4 [0m.
[ [0;32m OK [0m] Finished [0;1;39mFile System Check on /dev/sda4 [0m.
    Mounting [0;1;39m/run/media/sda4 [0m...
[ 26.943177] EXT4-fs (sda4): mounted filesystem with ordered data mode. Opts:
(null). Quota mode: none.
[ [0;32m OK [0m] Mounted [0;1;39m/run/media/sda4 [0m.
[ 28.082211] snd_hda_intel 0000:07:00.1: enabling device (0000 -> 0002)
[ 28.088034] snd_hda_intel 0000:07:00.1: Force to snoop mode by module option
    Starting [0;1;39mFile System Check on /dev/nvme0n1p3 [0m...
    Starting [0;1;39mFile System Check on /dev/sda3 [0m...
[ [0;32m OK [0m] Finished [0;1;39mFile System Check on /dev/nvme0n1p3 [0m.
    Mounting [0;1;39m/run/media/nvme0n1p3 [0m...
[ [0;32m OK [0m] Mounted [0;1;39m/run/media/nvme0n1p3 [0m.
[ [0;32m OK [0m] Finished [0;1;39mFile System Check on /dev/sda3 [0m.
    Mounting [0;1;39m/run/media/sda3 [0m...
[ [0;32m OK [0m] Mounted [0;1;39m/run/media/sda3 [0m.
[ 30.612155] input: HDA ATI HDMI HDMI/DP,pcm=3 as /devices/platform/soc/
e000000000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/0000:02:08.0/0000:07:00.1/sound/
card0/input4
[ [0;32m OK [0m] Reached target [0;1;39mSound Card [0m.
[ [0;32m OK [0m] Found device [0;1;39m/dev/mmcblk0p3 [0m.
    Mounting [0;1;39m/boot [0m...
[ 34.379793] cfg80211: Loading compiled-in X.509 certificates for regulatory
database
[ 34.392793] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[ [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (24s / 1min
52s)
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (24s /
1min 52s)
M
[K] [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (25s / 1min 52s)
M
[K] [0m [0;31m* [0m] A start job is running for /boot (26s / 1min 52s)
[ 37.109300] Bluetooth: Core ver 2.22
[ 37.112273] NET: Registered protocol family 31
[ 37.116578] Bluetooth: HCI device and connection manager initialized
[ 37.122947] Bluetooth: HCI socket layer initialized
[ 37.127779] Bluetooth: L2CAP socket layer initialized
[ 37.132826] Bluetooth: SCO socket layer initialized
M
[K] [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (26s / 1min 52s)
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (27s /
1min 52s)
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (27s /

```

```

lmin 52s)
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (28s /
lmin 52s)
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (29s /
lmin 52s)
[ 40.333244] Intel(R) Wireless WiFi driver for Linux
[ 40.337634] iwlfwifi 0000:05:00.0: enabling device (0000 -> 0002)
M
[K] [0;31m* [0;1;31m* [0m] A start job is running for /boot (29s / lmin 52s)
[ 40.591883] usbcore: registered new interface driver btusb
[ 40.599845] Bluetooth: hci0: Bootloader revision 0.3 build 0 week 24 2017
[ 40.607359] Bluetooth: hci0: Device revision is 1
[ 40.611313] Bluetooth: hci0: Secure boot is enabled
[ 40.616175] Bluetooth: hci0: OTP lock is enabled
[ 40.620772] Bluetooth: hci0: API lock is enabled
[ 40.625378] Bluetooth: hci0: Debug lock is disabled
[ 40.630240] Bluetooth: hci0: Minimum firmware build 1 week 10 2014
M
[K] [0;31m* [0m] A start job is running for /boot (182us / lmin 21s)
[ 42.701907] Bluetooth: hci0: Found device firmware: intel/ibt-20-1-3.sfi
[ 43.076086] iwlfwifi 0000:05:00.0: api flags index 2 larger than supported by driver
[ 43.083132] iwlfwifi 0000:05:00.0: TLV_FW_FSEQ_VERSION: FSEQ Version: 89.3.35.22
[ 43.100455] iwlfwifi 0000:05:00.0: loaded firmware version 59.601f3a66.0
cc-a0-59.ucode op_mode iwlmvm
M
[K] [0;31m* [0;1;31m* [0m] A start job is running for /boot (2s / lmin 21s)
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for /boot (2s /
lmin 21s)
[ 43.934628] FAT-fs (mmcblk0p3): Volume was not properly unmounted. Some data may
be corrupt. Please run fsck.
M
[K] [0;32m OK [0m] Mounted [0;1;39m/boot [0m.
[K] [0;32m OK [0m] Reached target [0;1;39mLocal File Systems [0m.
Starting [0;1;39mCreate Volatile Files and Directories [0m...
[ 44.654061] Bluetooth: hci0: Waiting for firmware download to complete
[ 44.659951] Bluetooth: hci0: Firmware loaded in 3965814 usecs
[ 44.666063] Bluetooth: hci0: Waiting for device to boot
[ 44.684614] Bluetooth: hci0: Device booted in 18574 usecs
[ 44.993358] Bluetooth: hci0: Found Intel DDC parameters: intel/ibt-20-1-3.ddc
[ 45.001696] Bluetooth: hci0: Applying Intel DDC parameters completed
[ 45.010681] Bluetooth: hci0: Firmware revision 0.0 build 26 week 3 2021
[ [0;32m OK [0m] Finished [0;1;39mCreate Volatile Files and Directories [0m.
Starting [0;1;39mNetwork Name Resolution [0m...
Starting [0;1;39mNetwork Time Synchronization [0m...
Starting [0;1;39mUpdate UTMP about System Boot/Shutdown [0m...
[ [0;32m OK [0m] Finished [0;1;39mUpdate UTMP about System Boot/Shutdown [0m.
[ [0;31m* [0;1;31m* [0m [0;31m* [0m] (1 of 2) A start job is running for...me
Synchronization (6s / lmin 34s)
M
[K] [0;32m OK [0m] Started [0;1;39mNetwork Time Synchronization [0m.
[K] [0;32m OK [0m] Reached target [0;1;39mSystem Initialization [0m.
[ [0;32m OK [0m] Started [0;1;39mDaily rotation of log files [0m.
[ [0;32m OK [0m] Started [0;1;39mDaily Cleanup of Temporary Directories [0m.
[ [0;32m OK [0m] Listening on [0;1;39mAvahi mDNS/DNS-SD Stack Activation Socket
[0m.

```



```

[ [0;32m OK [0m] Listening on [0;1;39mD-Bus System Message Bus Socket [0m.
[ [0;32m OK [0m] Reached target [0;1;39mSockets [0m.
[ [0;32m OK [0m] Reached target [0;1;39mBasic System [0m.
[ [0;32m OK [0m] Started [0;1;39mJob spooling tools [0m.
    Starting [0;1;39mBluetooth service [0m...
    Starting [0;1;39mNTP client/server [0m...
    Starting [0;1;39mConsole System Startup Logging [0m...
[ [0;32m OK [0m] Started [0;1;39mPeriodic Command Scheduler [0m.
[ [0;32m OK [0m] Started [0;1;39mD-Bus System Message Bus [0m.
    Starting [0;1;39mIPv6 Packet Filtering Framework [0m...
    Starting [0;1;39mIPv4 Packet Filtering Framework [0m...
    Starting [0;1;39mTelephony service [0m...
    Starting [0;1;39mUser Login Management [0m...
[ [0;32m OK [0m] Started [0;1;39mXserver startup without a display manager [0m.
[ [0;32m OK [0m] Started [0;1;39mNetwork Name Resolution [0m.
[ [0;32m OK [0m] Reached target [0;1;39mHost and Network Name Lookups [0m.
    Starting [0;1;39mAvahi mDNS/DNS-SD Stack [0m...
[ [0;32m OK [0m] Started [0;1;39mNTP client/server [0m.
[ [0;32m OK [0m] Finished [0;1;39mConsole System Startup Logging [0m.
[ [0;31m* [0;1;31m* [0m [0;31m* [0m] (3 of 6) A start job is running for...ltering
Framework (44s / no limit)
M
[K [0;32m OK [0m] Finished [0;1;39mIPv6 Packet Filtering Framework [0m.
[K [0;32m OK [0m] Finished [0;1;39mIPv4 Packet Filtering Framework [0m.
[ [0;32m OK [0m] Reached target [0;1;39mNetwork (Pre) [0m.
    Starting [0;1;39mNetwork Manager [0m...
[ [0;31m* [0;1;31m* [0m [0;31m* [0m] (4 of 5) A start job is running for Bluetooth
service (16s / 1min 36s)
M
[K [0;1;31m* [0m [0;31m* [0m] (5 of 5) A start job is running for...r Login
Management (47s / 2min 8s)
M
[K [0m [0;31m* [0m] (5 of 5) A start job is running for...r Login Management
(48s / 2min 8s)
M
[K [0;1;31m* [0m [0;31m* [0m] (5 of 5) A start job is running for...r Login
Management (48s / 2min 8s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] (1 of 5) A start job is running for... mDNS/
DNS-SD Stack (49s / 2min 9s)
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] (1 of 5) A start job is running for... mDNS/
DNS-SD Stack (49s / 2min 9s)
[ 60.546833] [drm] radeon kernel modesetting enabled.
[ 60.551502] radeon 0000:07:00.0: enabling device (0000 -> 0003)
[ 60.557980] [drm] initializing kernel modesetting (CAICOS 0x1002:0x6779
0x1002:0x0000 0x00).
[ 60.565715] [drm:radeon_device_init [radeon]] *ERROR* Unable to find PCI I/O BAR
M
[K [0;31m* [0;1;31m* [0m [0;31m* [0m] (1 of 5) A start job is running for... mDNS/
DNS-SD Stack (50s / 2min 9s)
[ 60.782235] [drm:radeon_atombios_init [radeon]] *ERROR* Unable to find PCI I/O
BAR; using MMIO for ATOM IIO
[ 60.885115] ATOM BIOS: C26401
[ 60.888201] [drm] GPU not posted. posting now...
[ 60.898265] radeon 0000:07:00.0: VRAM: 1024M 0x0000000000000000 -
0x000000003FFFFFFF (1024M used)
[ 60.906388] radeon 0000:07:00.0: GTT: 1024M 0x0000000040000000 - 0x000000007FFFFFFF

```

```

[ 60.914033] [drm] Detected VRAM RAM=1024M, BAR=256M
[ 60.918895] [drm] RAM width 64bits DDR
[ 60.922918] [TTM] Zone kernel: Available graphics memory: 8211620 KiB
[ 60.929147] [TTM] Zone dma32: Available graphics memory: 2097152 KiB
[ 60.935751] [drm] radeon: 1024M of VRAM memory ready
[ 60.940611] [drm] radeon: 1024M of GTT memory ready.
[ 60.945594] [drm] Loading CAICOS Microcode
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] (2 of 5) A start job is running for
Telephony service (50s / 2min 8s)
M
[K] [0;31m* [0;1;31m* [0m] (2 of 5) A start job is running for Telephony service
(51s / 2min 8s)
M
[K] [0;32m OK [0m] Started [0;1;39mUser Login Management [0m.
[K] [0;32m OK [0m] Started [0;1;39mBluetooth service [0m.
[ [0;32m OK [0m] Started [0;1;39mTelephony service [0m.
[ [0;32m OK [0m] Started [0;1;39mAvahi mDNS/DNS-SD Stack [0m.
[ [0;32m OK [0m] Reached target [0;1;39mBluetooth [0m.
[ 63.109903] [drm] Internal thermal controller with fan control
[ 63.119463] iwlfwifi 0000:05:00.0: Detected Intel(R) Wi-Fi 6 AX200 160MHz, REV=0x340
[ 63.119828] [drm] radeon: dpm initialized
[ 63.321891] iwlfwifi 0000:05:00.0: base HW address: ac:12:03:9e:f2:eb
[ 63.343515] thermal thermal_zone0: failed to read out thermal zone (-61)
[ 63.663019] iwlfwifi 0000:05:00.0 wlp5s0: renamed from wlan0
[ [0;31m* [0m] A start job is running for Network Manager (53s / 2min 15s)
M
[K] [0;31m* [0;1;31m* [0m] A start job is running for Network Manager (53s /
2min 15s)
[ 64.209272] [drm] GART: num cpu pages 262144, num gpu pages 262144
[ 64.230743] [drm] enabling PCIE gen 2 link speeds, disable with radeon.pcie_gen2=0
[ 64.262285] [drm] PCIE GART of 1024M enabled (table at 0x0000000000162000).
[ 64.268713] radeon 0000:07:00.0: WB enabled
[ 64.272698] radeon 0000:07:00.0: fence driver on ring 0 use gpu addr
0x0000000040000c00
[ 64.280679] radeon 0000:07:00.0: fence driver on ring 3 use gpu addr
0x0000000040000c0c
[ 64.284530] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[ 64.292144] radeon 0000:07:00.0: fence driver on ring 5 use gpu addr
0x0000000000072118
[ 64.293956] Bluetooth: BNEP filters: protocol multicast
[ 64.307180] Bluetooth: BNEP socket layer initialized
[ 64.315731] radeon 0000:07:00.0: radeon: MSI limited to 32-bit
[ 64.320960] radeon 0000:07:00.0: radeon: using MSI.
[ 64.325796] [drm] radeon: irq initialized.
[ 64.346493] [drm] ring test on 0 succeeded in 1 usecs
[ 64.350811] [drm] ring test on 3 succeeded in 10 usecs
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for Network Manager
(53s / 2min 15s)
M
[K] Starting [0;1;39mHostname Service [0m...
[K] [64.533075] [drm] ring test on 5 succeeded in 2 usecs
[ 64.537389] [drm] UVD initialized successfully.
[ 64.542254] [drm] ib test on ring 0 succeeded in 0 usecs
[ 64.547339] [drm] ib test on ring 3 succeeded in 0 usecs
[ 64.703788] [drm] ib test on ring 5 succeeded
[ 64.711675] [drm] Radeon Display Connectors

```

```

[ 64.715107] [drm] Connector 0:
[ 64.718125] [drm]   HDMI-A-1
[ 64.721018] [drm]   HPD2
[ 64.723525] [drm]   DDC: 0x6460 0x6460 0x6464 0x6464 0x6468 0x6468 0x646c 0x646c
[ 64.730906] [drm]   Encoders:
[ 64.733847] [drm]     DFP1: INTERNAL_UNIPHY1
[ 64.738119] [drm] Connector 1:
[ 64.741156] [drm]   DVI-I-1
[ 64.743934] [drm]   HPD4
[ 64.746441] [drm]   DDC: 0x6450 0x6450 0x6454 0x6454 0x6458 0x6458 0x645c 0x645c
[ 64.753842] [drm]   Encoders:
[ 64.756791] [drm]     DFP2: INTERNAL_UNIPHY
[ 64.760958] [drm] Connector 2:
[ 64.763999] [drm]   VGA-1
[ 64.766592] [drm]   DDC: 0x6430 0x6430 0x6434 0x6434 0x6438 0x6438 0x643c 0x643c
[ 64.773988] [drm]   Encoders:
[ 64.776941] [drm]     CRT1: INTERNAL_KLDSCP_DAC1
[ 64.865937] [drm] fb mappable at 0x2000363000
[ 64.869672] [drm] vram apper at 0x2000000000
[ 64.873935] [drm] size 7299072
[ 64.876983] [drm] fb depth is 24
[ 64.880185] [drm]   pitch is 6912
1m* [0;1;31m* [0m [0;31m* [0m] (1 of 2) A start job is running for Network Manager
(54s / 2min 15s)
[ 64.935821] Console: switching to colour frame buffer device 210x65
[ 64.979461] radeon 0000:07:00.0: [drm] fb0: radeondrmfb frame buffer device
[ 64.994999] [drm] Initialized radeon 2.50.0 20080528 for 0000:07:00.0 on minor 0
M
[K] [0;31m* [0;1;31m* [0m [0;31m* [0m] (2 of 2) A start job is running
for...ostname Service (873ms / 1min 30s)
M
[K] [0;32m OK [0m] Started [0;1;39mHostname Service [0m.
[K] [66.805967] Bluetooth: RFCOMM TTY layer initialized
[ 66.810131] Bluetooth: RFCOMM socket layer initialized
[ 66.815271] Bluetooth: RFCOMM ver 1.11
[ [0;31m* [0;1;31m* [0m [0;31m* [0m] A start job is running for Network Manager
(56s / 2min 15s)
M
[K] [0;1;31m* [0m [0;31m* [0m] A start job is running for Network Manager (57s /
2min 15s)
M
[K] [0m [0;31m* [0m] A start job is running for Network Manager (57s / 2min 15s)
M
[K] [0;1;31m* [0m [0;31m* [0m] A start job is running for Network Manager (58s /
2min 15s)
M
[K] [0;32m OK [0m] Started [0;1;39mNetwork Manager [0m.
[K] [0;32m OK [0m] Reached target [0;1;39mNetwork [0m.
Starting [0;1;39mDNS forwarder and DHCP server [0m...
[ [0;32m OK [0m] Started [0;1;39mRespond to IPv6 Node Information Queries [0m.
[ [0;32m OK [0m] Started [0;1;39mNetwork Router Discovery Daemon [0m.
Starting [0;1;39mPermit User Sessions [0m...
[ [0;32m OK [0m] Started [0;1;39mXinetd A Powerful Replacement For Inetd [0m.
[ [0;32m OK [0m] Finished [0;1;39mPermit User Sessions [0m.
[ [0;32m OK [0m] Started [0;1;39mGetty on tty1 [0m.
[ [0;32m OK [0m] Started [0;1;39mSerial Getty on ttySIF0 [0m.
[ [0;32m OK [0m] Reached target [0;1;39mLogin Prompts [0m.
[ [0;32m OK [0m] Started [0;1;39mtimer to trun-on green LED [0m.

```

```
[ [0;32m OK [0m] Reached target [0;1;39mTimers [0m.  
[ [0;32m OK [0m] Started [0;1;39mDNS forwarder and DHCP server [0m.  
[ [0;32m OK [0m] Reached target [0;1;39mMulti-User System [0m.  
[ [0;32m OK [0m] Reached target [0;1;39mGraphical Interface [0m.  
Starting [0;1;39mUpdate UTMP about System Runlevel Changes [0m...  
[ [0;32m OK [0m] Finished [0;1;39mUpdate UTMP about System Runlevel Changes [0m.
```

```
FreedomUSDK 2021.03.01 unmatched[ 71.174618] macb 10090000.ethernet eth0: PHY  
[10090000.ethernet-ffffffff:00] driver [Microsemi VSC8541 SyncE] (irq=POLL)  
[ 71.185432] macb 10090000.ethernet eth0: configuring for phy/gmii link mode  
ttySIF0
```

```
unmatched login: [ 75.284544] macb 10090000.ethernet eth0: Link is Up - 1Gbps/Full  
- flow control tx  
[ 75.291409] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready  
root
```

```
Password:  
7 [r [999;999H [6n 8root@unmatched:~#
```

A.4 99-hifive-unmatched.rules

```
ACTION!="add|change", GOTO="unmatched_rules_end"  
SUBSYSTEM!="usb|tty", GOTO="unmatched_rules_end"  
  
# FT2232 VID:PID derrived from: lsusb |grep FT2232  
ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6010", MODE="664", GROUP="plugdev"  
  
LABEL="unleashed_rules_end"
```