JH7100 SoC Boot User Guide

01 (2021-6-7)

Shanghai StarFive Technology Co., Ltd.

All Rights Reserved

PROPRIETARY NOTICE

Copyright © Shanghai StarFive Technology Co., Ltd., 2018-2025. All rights reserved.

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

Shanghai StarFive Technology Co., Ltd., 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.

Shanghai StarFive Technology Co., Ltd., 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.

Shanghai StarFive Technology Co., Ltd., reserves the right to make changes without further notice to any products herein.

All material appearing in this document is protected by copyright and is the property of Shanghai StarFive Technology Co., Ltd. You may not copy, reproduce, distribute, publish, display, perform, modify, create derivative works, transmit, or in any way exploit any such content, nor may you distribute any part of this content over any network, including a local area network, sell or offer it for sale, or use such content to construct any kind of database. Copying or storing any content except as provided above is expressly prohibited without prior written permission of the Shanghai StarFive Technology Co, Ltd (hereinafter “StarFive”).

Shanghai StarFive Technology Co., Ltd

Address:	Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China
Website:	www.starfivetech.com
e-Mail:	sales@starfivetech.com (sales)
	support@starfivetech.com (support)

About This Manual

Introduction

This document mainly describes the boot flow, the boot sources available for the JH7100 SoC and the Bare-metal boot examples.

Prerequisite

In order to run the examples presented in this guide, the following are required:

- Ubuntu 18.04
- BeagleV™ - StarLight development board

Revision History

Version	Released	Change Description
01	2021-6-7	The first official release.

Tables of Content

1 Boot Sources

The GPIO is used to select the boot vector and BootLoader source and offer multiple methods to obtain the BootLoader image.

The JH7100 SoC can boot from one of the sources listed in the following table, as selected by the PAD_GPIO [62:60] values.

Table 1-1 PAD_GPIO Values for Boot Source Selection

Processor	SCFG_boot_mode	PAD_GPIO [63]	Boot Vector	PAD_GPIO [62:60]
U74	0x1	-	SCFG_u74_reset_vector	-
	0x0 (default)	0x0	0x00_2000_0000, XIP Flash	• 0x0: 1-bit quad SPI NOR flash memory • 0x1: 4-bit quad SPI NOR flash memory • 0x2: SDIO (Reserved) • 0x3: eMMC (Reserved) • 0x4: UART • 0x5: USB (Reserved) • 0x6: chiplink (Reserved) • 0x7: SPI2AHB (Reserved)
		0x1	0x00_1840_0000, on-chip BootROM (32KB)	

Note:

1) The boot mode and boot source selection (PAD_GPIO [63]) can be read through syscon status registers.
2) Use the GPIO pad to select the vector and loader source by default.
3) PAD_GPIO [63] and PAD_GPIO [62:60] can be configured to 1 or 0 via pull-up/pull-down resister, button or jumper according to board hardware design.

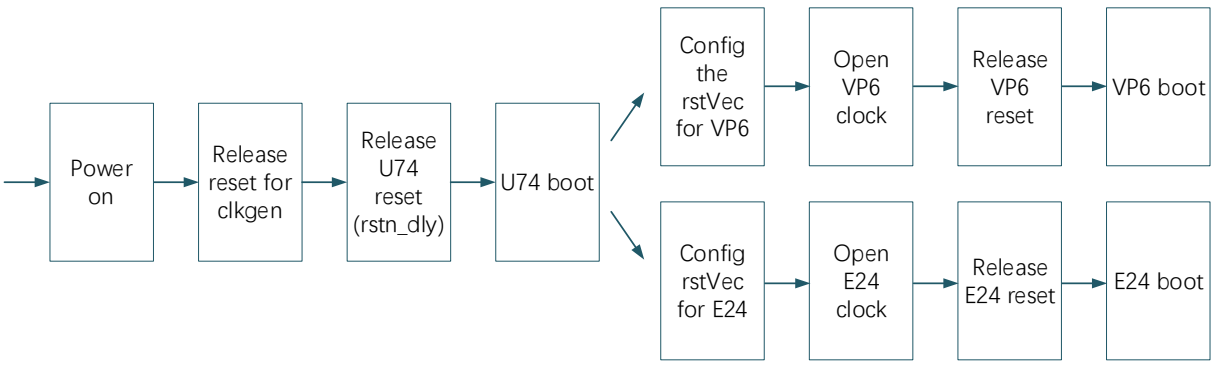

Figure 1-2 Hardware Boot Sequence

2 Boot Flow

The boot process starts when the processor is released from reset, and jumps to the reset vector address (0x1840,0000 by default), located in the BootROM address space.

The boot flow is a multi-stage process. Each stage is responsible for loading the next stage. The typical boot flow is illustrated in the following figure.

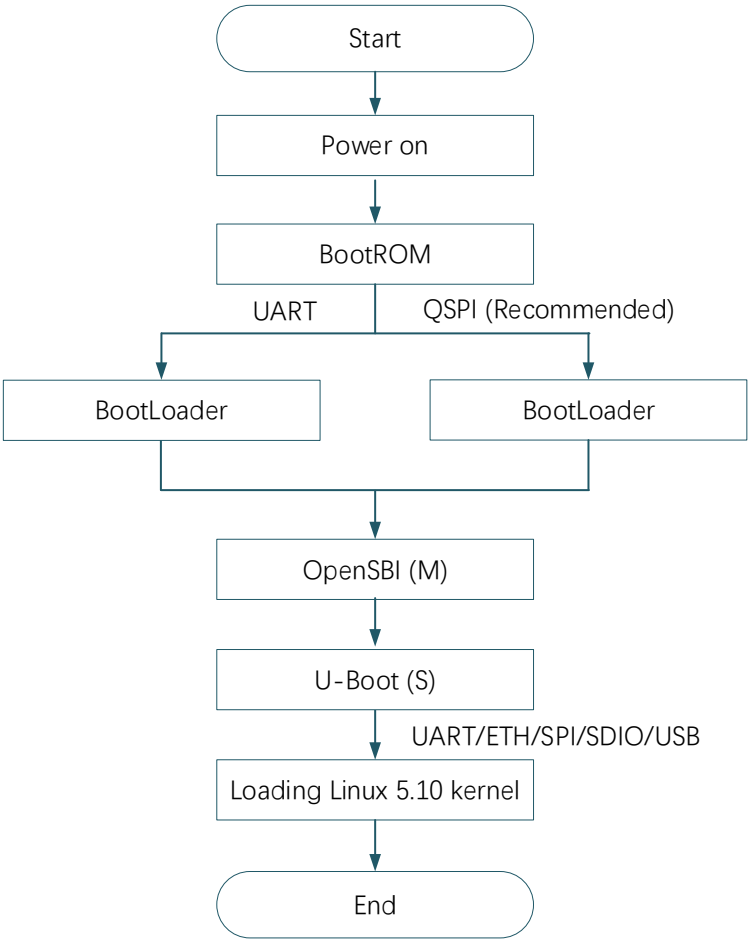

Figure 2-1 Typical Boot Flow

2.1 BootROM

The BootROM is located in on-chip ROM, and the storage address is 0x1840,0000, which cannot be dynamically updated. After power-on, each HART jumps to 0x1800,0000 (located in RAM) by default and starts to execute BootROM.

The main function of the BootROM is to select the boot source and execute it. According to different hardware jumpers on the chip, only UART and QSPI sources are supported currently.

Table 2-1 Boot Source Description

Source	Description
UART	Enter a simple command line. Load a limited size binary into the on-chip RAM and execute it. This mode is mainly used for firmware update.
QSPI	Automatically load the 32K Bootloader to 0x1800,0000 (located in RAM) from address 0 of NOR Flash and jump to it.

LIMITATION

The file loaded from NOR Flash cannot exceed 32KB.

2.2 BootLoader

The BootROM limits the size of data read from NOR Flash. The BootLoader reads DDRInit from 0x10000 in NOR Flash to 0x1808,0000 (located in RAM), and then jump to it for execution.

The DDRInit will initialize the DDR, then read fw_payload.bin (OpenSBI+Uboot, the file header contains file size information) from 0x40000 in NOR Flash to 0x8000,0000 (located in DDR), and then jump to it to execute the OpenSBI.

The normal output information is illustrated in the following figure.

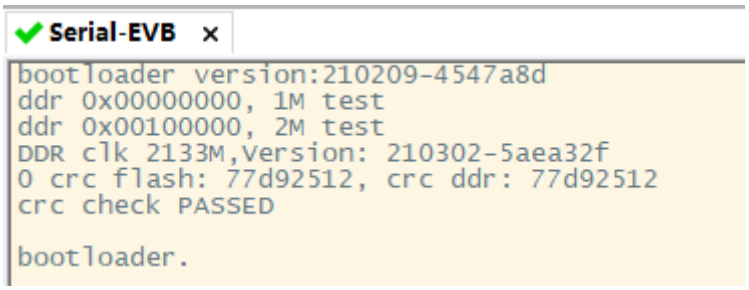

Figure 2-2 BootLoader Output Example

2.3 OpenSBI

The binary of OpenSBI is packaged with the binary compiled by U-Boot in the way of payload to generate the final fw_payload.bin. The main functions of OpenSBI are:

- Provide basic system calls for Linux
- Switch the mode from M mode to S mode
- Jump to 0x8002,0000 (located in DDR) to execute U-Boot.

The normal output information is illustrated in the following figure.

Figure 2-3 OpenSBI Output Example

2.4 U-Boot

U-Boot runs at 0x8002,0000 and works in S mode. It contains basic file system and commonly used peripheral drivers (such as GMAC, UART, QSPI, USB, SDIO etc.). U-Boot can load the kernel image through ETH, UART, QSPI, SDIO or USB.

The following example describes how to load Linux 5.10 kernel image from SDIO.

> The example assumes the installation of Ubuntu 18.04.
> Press Enter to confirm the operation or for the next command.

1. SD card partition.

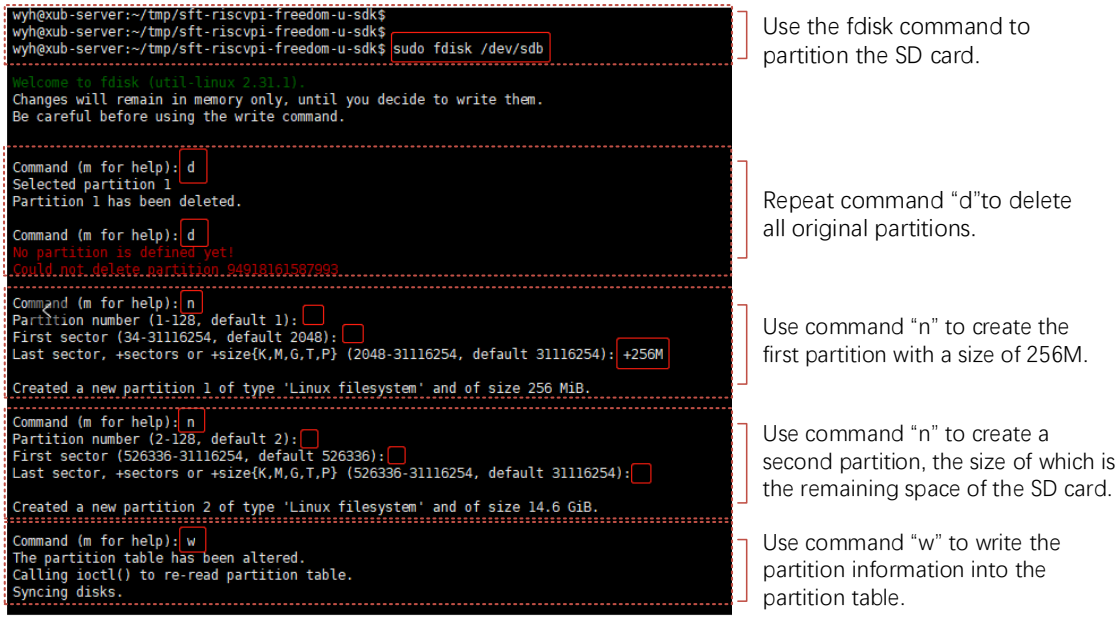

Figure 2-4 SD Card Partition Command Explanation

2. Format the partition.

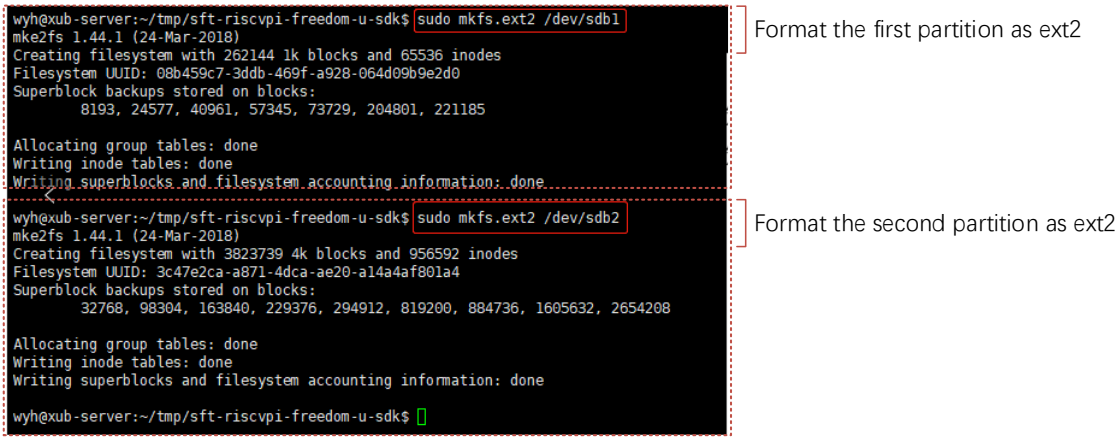

Figure 2-5 Format the Partition

3. Generate image.fit from Freelight U SDK, please refer to the detailed guidelines in the link https://github.com/starfive-tech/freelight-u-sdk.

4. Copy the boot file.

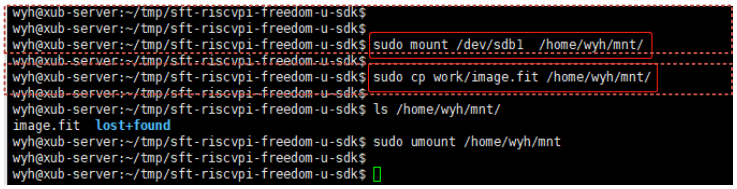

Figure 2-6 Copying the Boot File

5. Load the kernel (Linux 5.10 as an example).

The addresses 0x80200000, 0x86100000 and 0x86000000 have been specified when compiling and generating image.fit and cannot be modified.

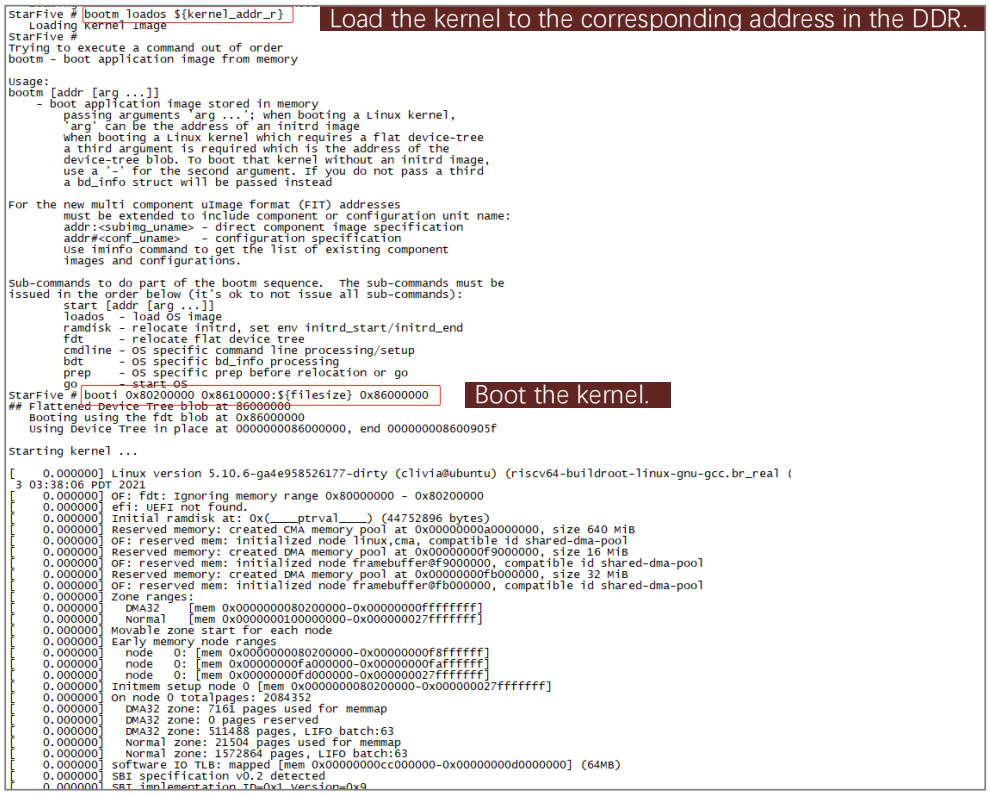

Figure 2-7 Loading the Kernel