StarFive 赛昉科技

JH-7110 DevKit eDP Developing and Porting Guide

Version: 1.0

Date: 2023/06/09

Doc ID: JH7110-DGEN-003

Legal Statements

Important legal notice before reading this documentation.

PROPRIETARY NOTICE

Copyright © Shanghai StarFive Technology Co., Ltd., 2024. All rights reserved.

Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to product development. Shanghai StarFive Technology Co., Ltd. (hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive 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.

StarFive does not assume any liability arising 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.

All material appearing in this document is protected by copyright and is the property of StarFive. You may not reproduce the information contained herein, in whole or in part, without the written permission of StarFive.

Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: http://www.starfivetech.com

Email:

- Sales: sales@starfivetech.com
- Support: support@starfivetech.com

Contents

© 2018-2024 StarFive Technology
All rights reserved

List of Tables

© 2018-2024 StarFive Technology
All rights reserved

List of Figures

© 2018-2024 StarFive Technology
All rights reserved

© 2018-2024 StarFive Technology
All rights reserved

Preface

About this guide and technical support information.

About this document

This document mainly provides the SDK developers with the programing basics and debugging know-how for the eDP of the StarFive next generation SoC platform - JH-7110.

Audience

This document mainly serves the eDP relevant driver developers. If you are developing other modules, place a request to your sales or support consultant for our complete documentation set on JH-7110.

Revision History

Table 0-1 Revision History

Version	Released	Revision
1.0	2023/06/09	The First Official Release.

Notes and notices

The following notes and notices might appear in this guide:

- **Tip:**
  Suggests how to apply the information in a topic or step.
- **Note:**
  Explains a special case or expands on an important point.
- **Important:**
  Points out critical information concerning a topic or step.
- **CAUTION:**
  Indicates that an action or step can cause loss of data, security problems, or performance issues.
- **Warning:**
  Indicates that an action or step can result in physical harm or cause damage to hardware.

1. Introduction

eDP (Embedded DisplayPort) is a digital interface based on display port architecture and protocol. The interface is developed, maintained and promoted by VESA (Video Electronics Standards Association). At present, the latest protocol standard is version 1.4. The most common protocol supported by the screen in the market is version 1.2 or 1.3.

© 2018-2024 StarFive Technology
All rights reserved

2. LT8911EXB DSI/CSI Bridge to eDP

2.1. Overview

JH-7110 DevKit uses Lontium LT8911EXB chip, a DSI/CSI Bridge to eDP. It is a MIPI®DSI/CSI to eDP converter with a single-port MIPI receiver, which has 1 clock lane and 4 data lanes operating at maximum 2.0 Gbps per data lane and a maximum input bandwidth of 8.0 Gbps. The converter decodes the input MIPI RGB16/18/24/30/36bpp, YUV422 16/20/24bpp, YUV420 12bpp packets and converts the formatted video data stream to a single-link VESA eDP1.4 compliant output with 1/2/4 configurable data lanes, supporting RBR (1.62 Gbps) and HBR (2.7 Gbps) link data rate.

2.2. Features

Lontium LT8911EXB has the following features.

- **Single-Port MIPI® DSI/CSI Receiver**
  - Compliant with D-PHY1.2, DSI1.3 and CSI1.3
  - 1 clock lane and 1 to 4 configurable data lanes
  - 80 Mb/s to 2.0 Gb/s per data lane
  - Data lane input de-skew
  - Internal Rterm calibration with less than 5% error
  - Programmable equalization
  - Support Burst and Non-Burst Mode
  - Support RGB565, RGB666, Loosely RGB666, RGB888, RGB10bpc, RGB12bpc, YUV422 8bpc, YUV422 10bpc, YUV422 12bpc, YUV420 12bpp input
- **eDP1.4 Transmitter**
  - Compliant to VESA eDP1.4 standard
  - Support 1/2/4 data lanes with 1.62 Gbps(RBR) or 2.7 Gbps(HBR)
  - Optional SSC 0.5% down-spreading output
  - Configurable output swing for optimized EMI
  - MCCS over AUX channel
- **Miscellaneous**

© 2018-2024 StarFive Technology
All rights reserved

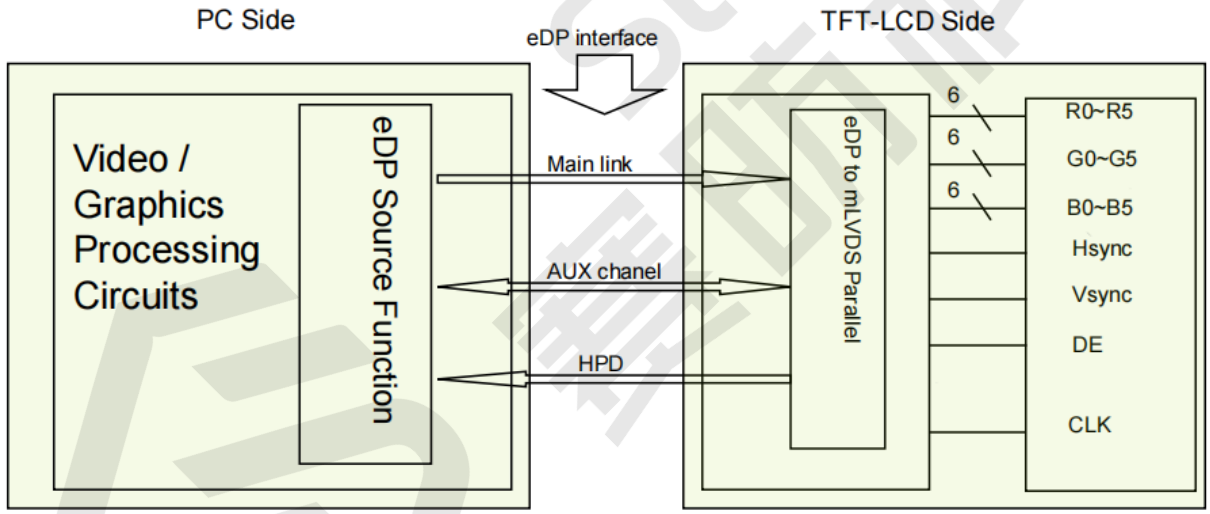
- Single 1.8V supply power
- Temperature range: −40°C to +85°C
- Packaged in 6 mm × 6 mm QFN48

2.3. Diagram

The following provides two diagrams about eDP application and line interface.

Figure 2-1 Application Diagram

Figure 2-2 eDP Interface



© 2018-2024 StarFive Technology
All rights reserved

3. eDP Pin Description

The following table displays the pin description of eDP.

Table 3-1 Pin Description

Terminal	Symbol	Function
Pin No.	Symbol	Description
1	CABC_ENABLE	CABC_ENABLE
2	H_GND	Ground
3	LANE1_N	eDP RX channel 1 negative
4	LANE1_P	eDP RX channel 1 positive
5	H_GND	Ground
6	LANE0_N	eDP RX channel 0 negative
7	LANE0_P	eDP RX channel 0 positive
8	H_GND	Ground
9	AUX_CH_P	eDP AUX CH positive
10	AUX_CH_N	eDP AUX CH negative
11	H_GND	Ground
12	LCD_VCC	Power Supply, 3.3V (typ.)
13	LCD_VCC	Power Supply, 3.3V (typ.)
14	Bist	Panel self test enable
15	H_GND	Ground
16	H_GND	Ground
17	HPD	Hot plug detect output
18	BL_GND	LED Ground
19	BL_GND	LED Ground
20	BL_GND	LED Ground
21	BL_GND	LED Ground
22	BL_ENABLE	LED enable pin (+3.3V Input)
23	BL_PWM	System enable pin (+3.3V Input)

© 2018-2024 StarFive Technology
All rights reserved

Table 3-1 Pin Description (continued)

Terminal	Symbol	Function
24	HSYNC	Reserve HSYNC Function
25	NC	No Connection
26	BL_POWER	LED Power Supply 5V-21V
27	BL_POWER	LED Power Supply 5V-21V
28	BL_POWER	LED Power Supply 5V-21V
29	BL_POWER	LED Power Supply 5V-21V
30	NC	No Connection

© 2018-2024 StarFive Technology
All rights reserved

4. Module Introduction

This chapter displays the following two parts:

- Kernel Display Module *(on page 13)*
- U-Boot Dispaly Module *(on page 23)*

4.1. Kernel Display Module

4.1.1. Kernel Display Driver

4.1.1.1. Device Tree Source Code

The following lists show the address of the display drivers.

- Linux kernel display driver:

```
linux-5.15/linux/drivers/gpu/drm/verisilicon
```

- Device tree of JH-7110 DevKit:

```
linux-5.15/arch/riscv/boot/dts/starfive/jh7110.dtsi
linux-5.15/arch/riscv/boot/dts/starfive/jh7110-devkits.dts
```

4.1.1.2. Device Tree Configuration

A DTS/DTSI file is used to store all the device tree configuration.

MIPI DSI

In the file `jh7110.dtsi`, you can find the device tree configuration of MIPI DSI as the following code block:

```
linux/arch/riscv/boot/dts/starfive/jh7110.dtsi:
   mipi_dsi: mipi@295d0000 {
                        compatible = "starfive,jh7110-mipi_dsi","cdns,dsi";
                        reg = <0x0 0x295d0000 0x0 0x10000>;
                        interrupts = <98>;
                        reg-names = "dsi";
                        clocks = <&clkvout JH7110_U0_CDNS_DSITX_CLK_SYS>,
                                 <&clkvout JH7110_U0_CDNS_DSITX_CLK_APB>,
                                 <&clkvout JH7110_U0_CDNS_DSITX_CLK_TXESC>,
```


© 2018-2024 StarFive Technology
All rights reserved

```
                                <&clkvout JH7110_U0_CDNS_DSITX_CLK_DPI>;
                        clock-names = "sys", "apb", "txesc", "dpi";
                        resets = <&rstgen RSTN_U0_CDNS_DSITX_DPI>,
                                 <&rstgen RSTN_U0_CDNS_DSITX_APB>,
                                 <&rstgen RSTN_U0_CDNS_DSITX_RXESC>,
                                 <&rstgen RSTN_U0_CDNS_DSITX_SYS>,
                                 <&rstgen RSTN_U0_CDNS_DSITX_TXBYTEHS>,
                                 <&rstgen RSTN_U0_CDNS_DSITX_TXESC>;
                        reset-names = "dsi_dpi", "dsi_apb", "dsi_rxesc",

  "dsi_sys", "dsi_txbytehs", "dsi_txesc";
                        phys = <&mipi_dphy>;
                        phy-names = "dphy";
                        status = "disabled";

                        mipi_panel: panel@0 {
                                /*compatible = "";*/
                                status = "okay";
                        };
                };
```

The following list provides explanations for the parameters included in the above code block.

- **compatible**: Compatibility information, used to associate the driver and its target device.
- **reg**: Register base address "`0x295d0000`" and range "`0x10000`".
- **interrupts**: Hardware interrupt ID.
- **reg-name**: The name of the register.
- **clocks**: The clocks used by the eDP module.
- **clock-names**: The names of the above clocks.
- **resets**: The reset signals used by the eDP module.
- **reset-names**: The names of the above reset signals.
- **status**: The work status of the eDP module. To enable the module, set this bit as "`okay`" or to disable the module, set this bit as "`disabled`".

MIPI DPHY

In the file `jh7110.dts`, you can find the device tree configuration of MIPI DPHY as the following code block:

```
linux/arch/riscv/boot/dts/starfive/jh7110.dts:
  mipi_dphy: mipi-dphy@295e0000{
                        compatible
 = "starfive,jh7110-mipi-dphy-tx","m31,mipi-dphy-tx";
                        reg = <0x0 0x295e0000 0x0 0x10000>;
                        clocks = <&clkvout JH7110_U0_MIPITX_DPHY_CLK_TXESC>;
```


© 2018-2024 StarFive Technology
All rights reserved

```
                        clock-names = "dphy_txesc";
                        resets = <&rstgen RSTN_U0_MIPITX_DPHY_SYS>,
                                 <&rstgen RSTN_U0_MIPITX_DPHY_TXBYTEHS>;
                        reset-names = "dphy_sys", "dphy_txbytehs";
                        #phy-cells = <0>;
                        status = "disabled";
                };
```

The following list provides explanations for the parameters included in the above code block.

- **compatible**: Compatibility information, used to associate the driver and its target device.
- **reg**: Register base address "`0x295e0000`" and range "`0x10000`".
- **clocks**: The clocks used by the eDP module.
- **clock-names**: The names of the above clocks.
- **resets**: The reset signals used by the eDP module.
- **reset-names**: The names of the above reset signals.
- **status**: The work status of the eDP module. To enable the module, set this bit as "`okay`" or to disable the module, set this bit as "`disabled`".

I2C2

In the file `jh7110-devkits.dts`, to configure `lt8911exb`, the `lt8911exb` dts port should be added into `i2c2`. You can find the device tree configuration of `i2c2` as the following code block:

```
linux/arch/riscv/boot/dts/starfive/jh7110-devkits.dts:
&i2c2 {
        clock-frequency = <100000>;
        i2c-sda-hold-time-ns = <300>;
        i2c-sda-falling-time-ns = <510>;
        i2c-scl-falling-time-ns = <510>;
        auto_calc_scl_lhcnt;
        pinctrl-names = "default";
        pinctrl-0 = <&i2c2_pins>;
        status = "okay";

        lt8911exb_i2c@29 {
                compatible = "lontium,lt8911exb";
                reg = <0x29>;
                reset-gpio = <&gpio 41 1>;
                pwm-gpio = <&gpio 33 1>;
                bl-gpio = <&ext_gpio 6 GPIO_ACTIVE_LOW>;

                port {
                                lt8911exb_in: endpoint {
                                        remote-endpoint = <&dsi0_output>;
                                };
```


© 2018-2024 StarFive Technology
All rights reserved

```
                        };
        };
};
```

In the above code block, the parameters of **pinctrl-names** and **pinctrl-0** are used to configure the `i2c2` pin configuration settings.

&MIPI DST

In the file `jh7110-devkits.dts`, configure remote-end point for connecting DSI and panel. You can find the device tree configuration of &MIPI DST as the following code block:

```
linux/arch/riscv/boot/dts/starfive/jh7110-devkits.dts:
&mipi_dsi {
        status = "okay";

        ports {
                #address-cells = <1>;
                #size-cells = <0>;

                port@0 {
                        reg = <0>;
                        #address-cells = <1>;
                        #size-cells = <0>;

                        dsi0_output: endpoint@0 {
                                reg = <0>;
                                remote-endpoint = <&lt8911exb_in>;
                        };

                        dsi1_output: endpoint@1 {
                                reg = <1>;
                                remote-endpoint = <&panel_dsi_port>;
                        };

                        dsi2_output: endpoint@2 {
                                reg = <2>;
                                remote-endpoint = <&radxa_in>;
                        };
                };

                port@1{
                        reg = <1>;
                        dsi_in_port: endpoint {
                                remote-endpoint = <&mipi_out>;
                        };
                };
```


© 2018-2024 StarFive Technology
All rights reserved

```
        };
};
```

4.1.1.3. Driver Configuration

The following code block shows the driver configuration.

```
CONFIG_DRM_VERISILICON=y
```

4.1.1.4. Kernel Menu Configuration

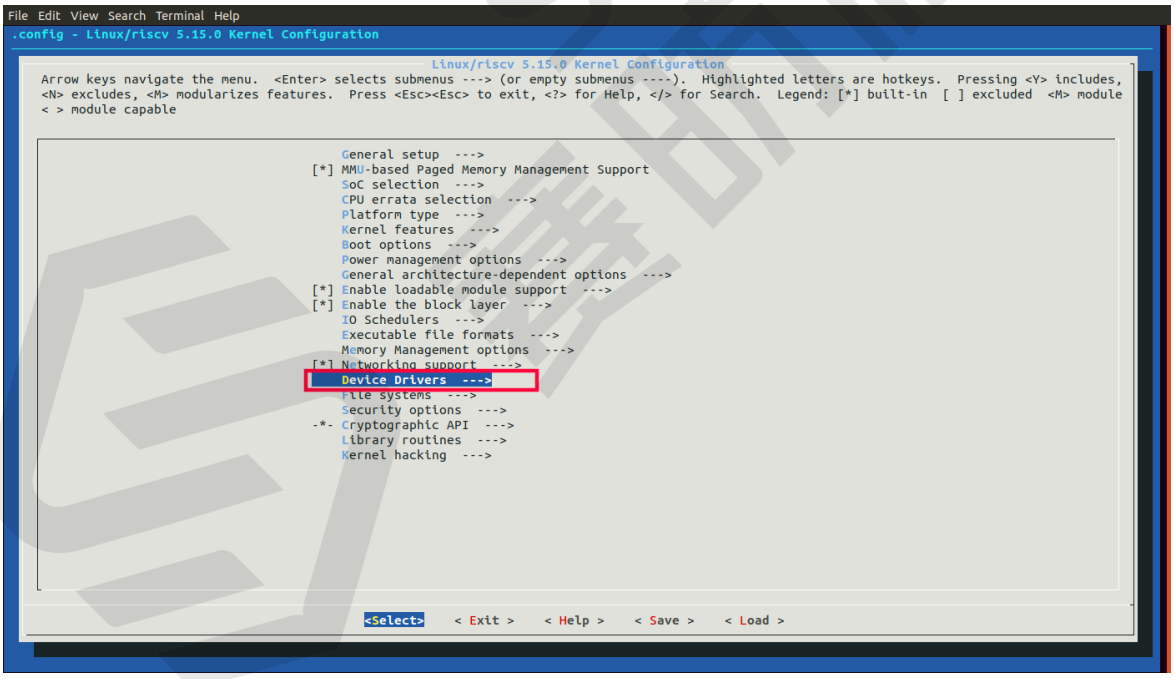
Follow the steps below to enable the kernel configuration for eDP.

1. Under the root directory of `freelight-u-sdk`, type the following command to enter the kernel menu configuration GUI.

   ```
   make linux-menuconfig
   ```

2. Enter the **Device Drivers** menu.

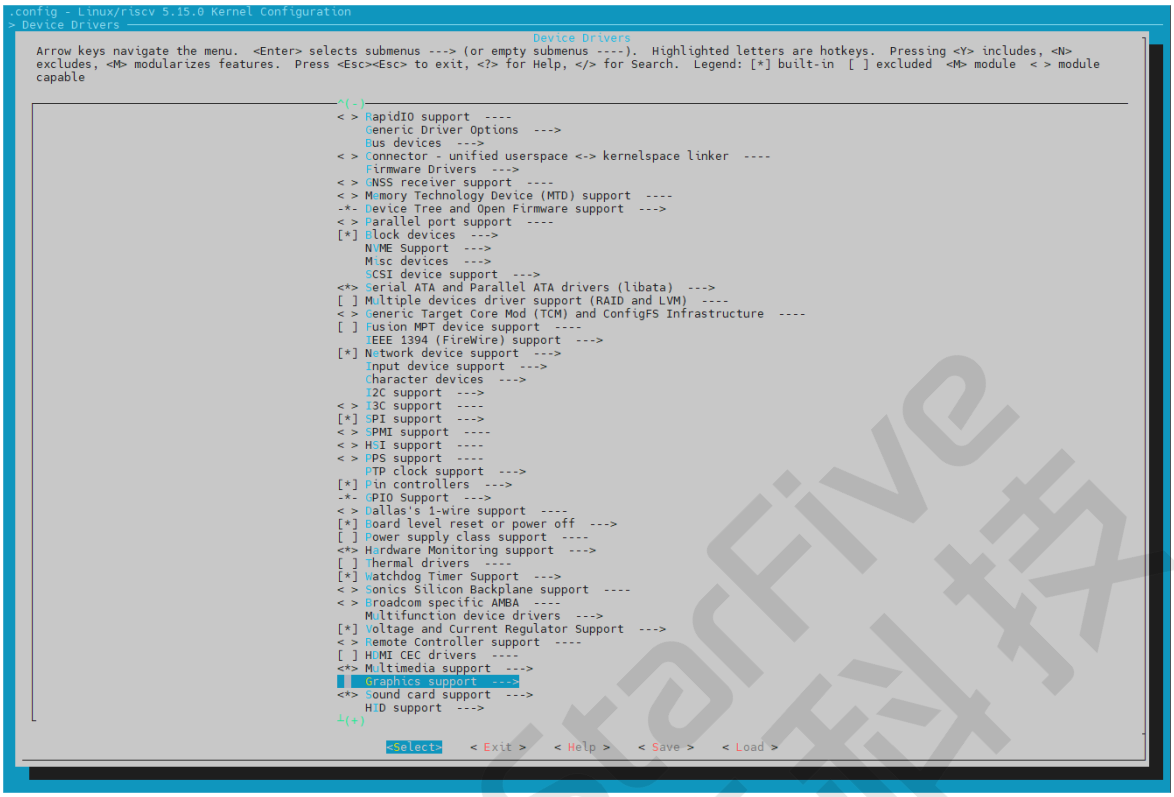
   Figure 4-1 Device Drivers



3. Enter the **Graphics support** menu.

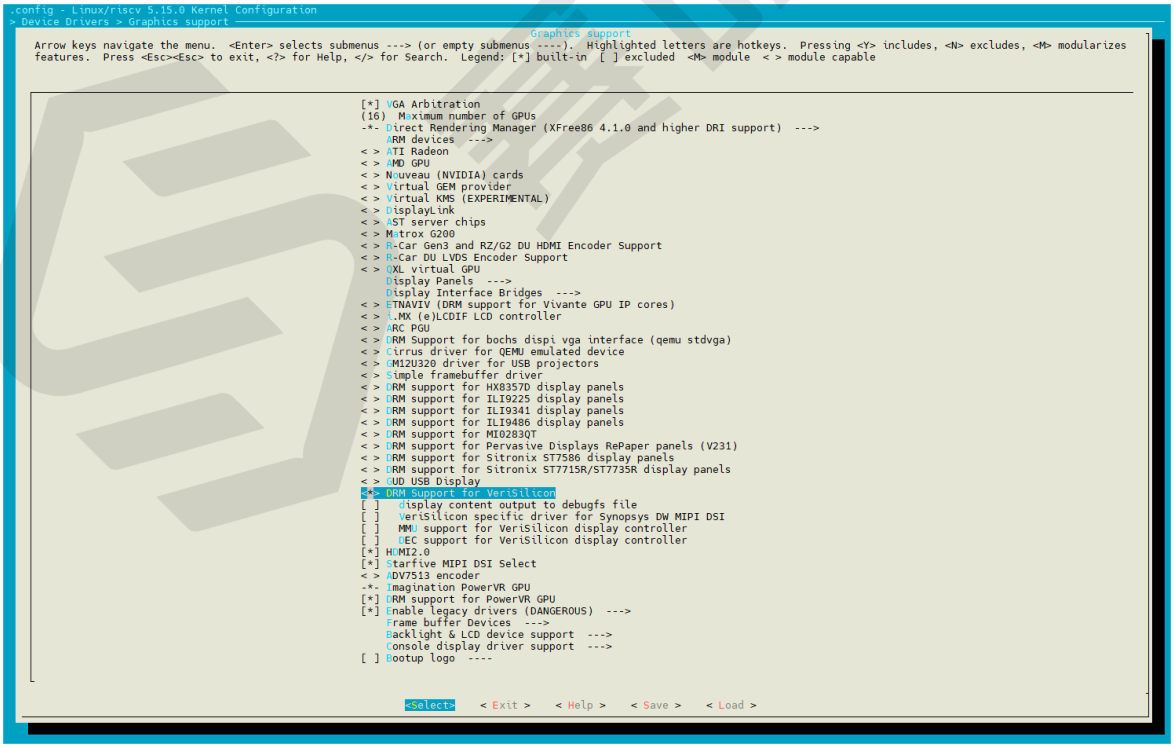
© 2018-2024 StarFive Technology
All rights reserved

Figure 4-2 Graphics support



4. In the **Graphics support** menu, select the **DRM Support for VeriSilicon** option to enable video output.

Figure 4-3 DRM Support for VeriSilicon

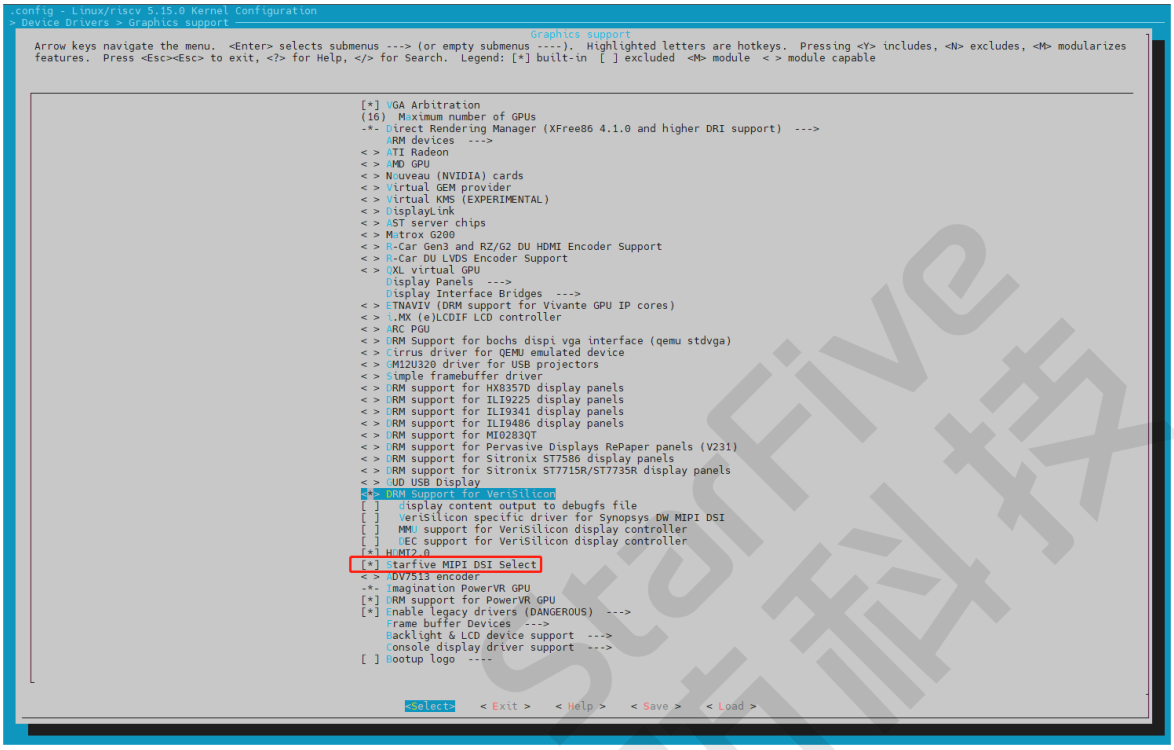


© 2018-2024 StarFive Technology
All rights reserved

For MIPI Output

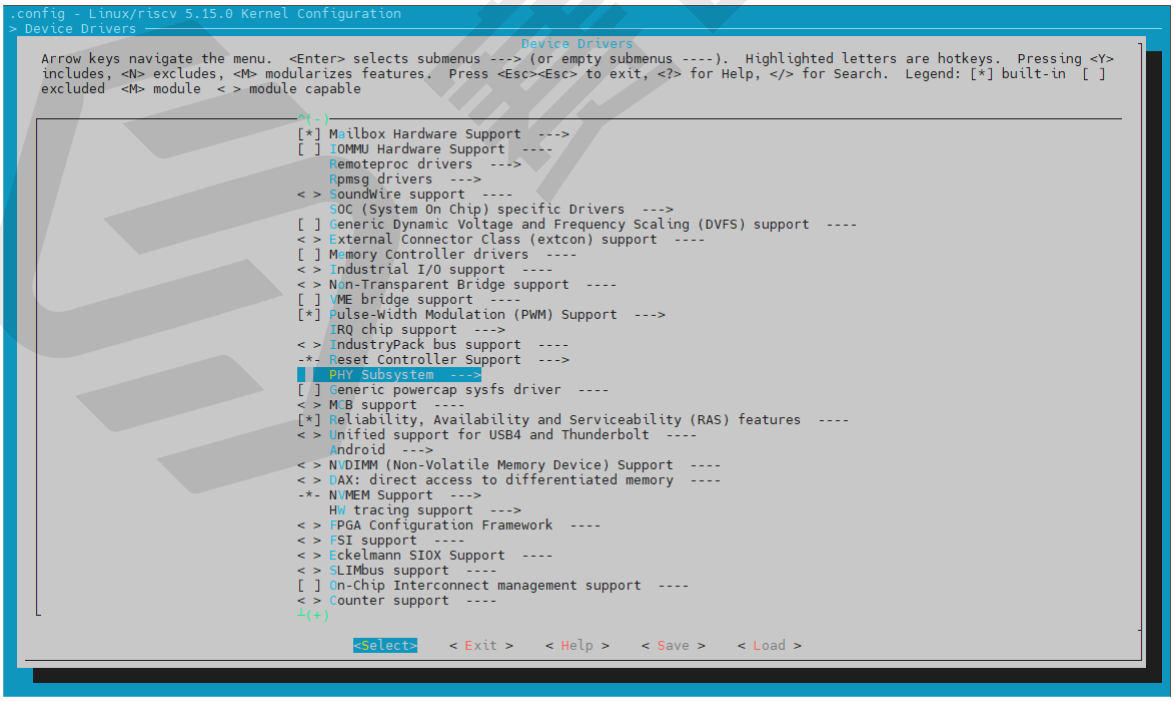
1. For MIPI output, select **StarFive MIPI DSI Select** under **Graphics support**.

Figure 4-4 Starfive MIPI DSI Select



2. Back to **Device Driver** menu and select **PHY Subsystem** to configure M31 DPHY.

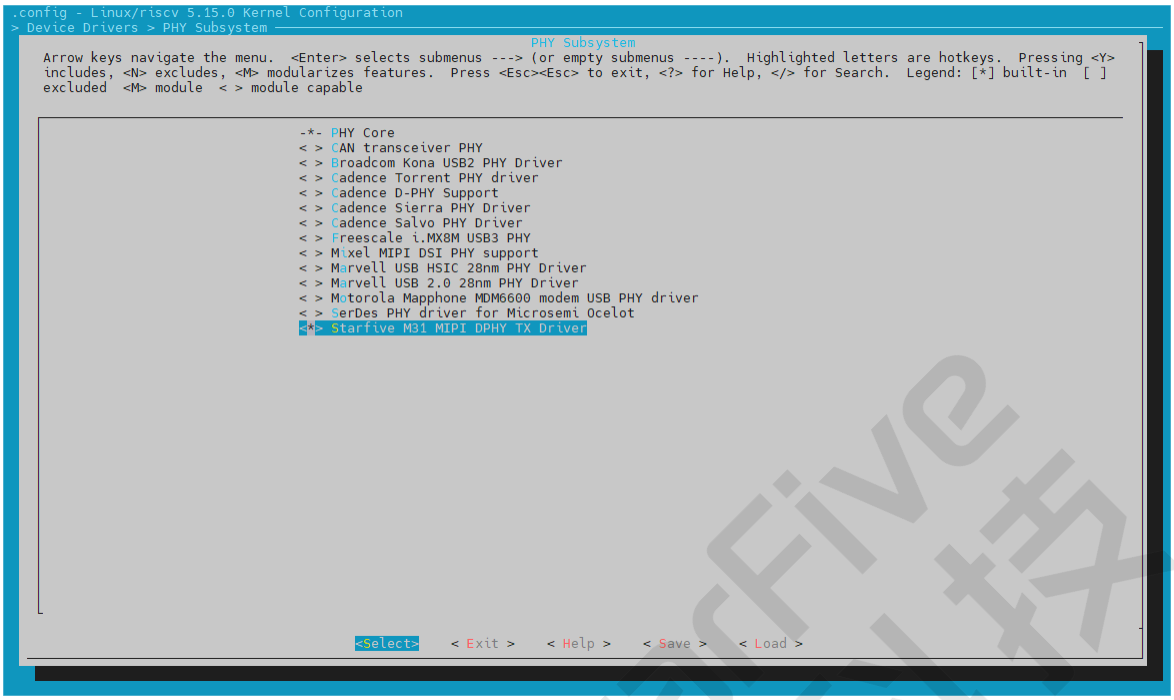
Figure 4-5 PHY Subsystem



3. Enter the **PHY Subsystem** menu and select **Starfive M31 MIPI DPHY TX Driver** option.

© 2018-2024 StarFive Technology
All rights reserved

Figure 4-6 Starfive M31 MIPI DPHY TX Driver

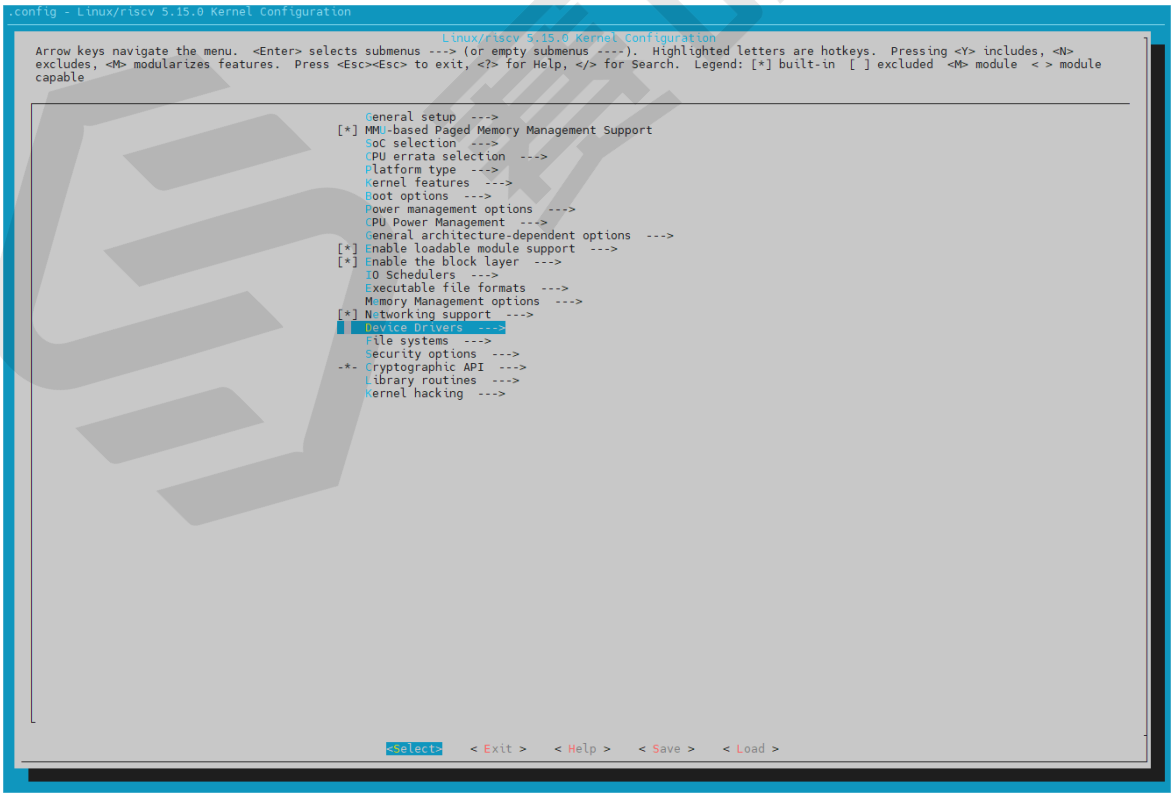


4. Save your change before you exit the kernel configuration dialog.

For eDP Output

1. For eDP output, configure the `lt8911exb`. Enter the **Device Drivers** menu.

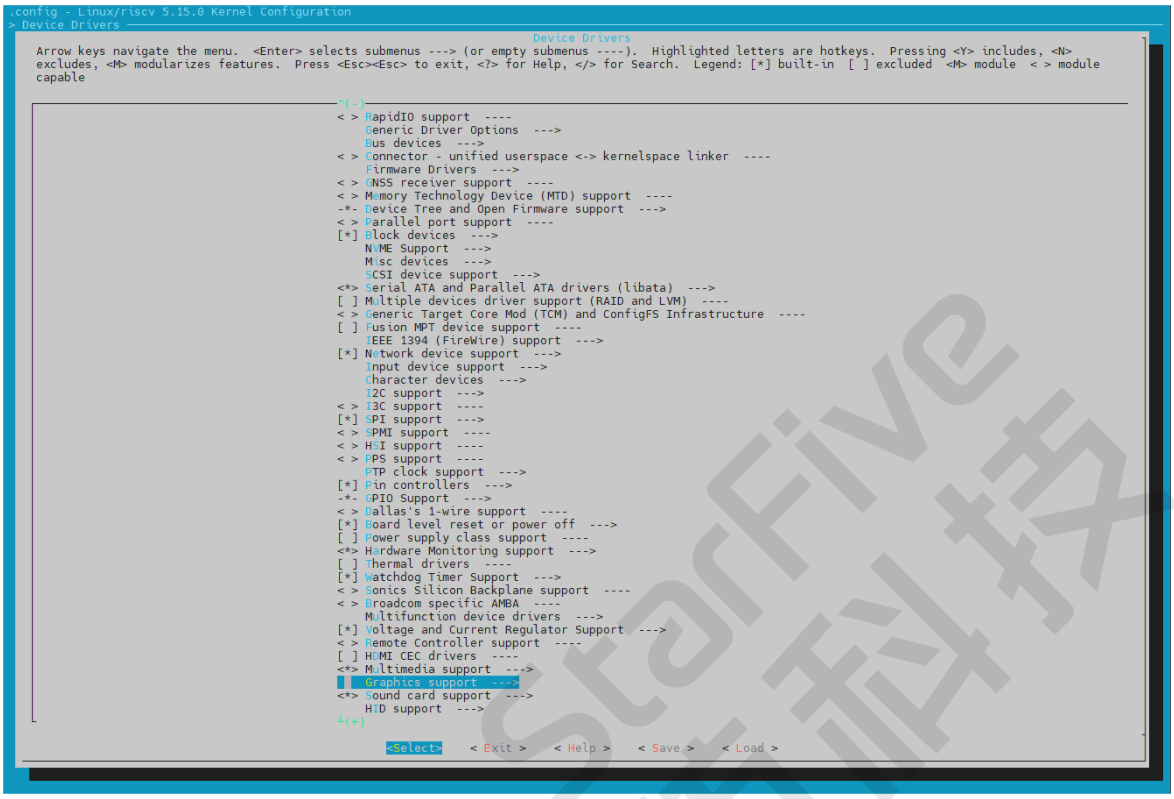
Figure 4-7 Device Driver



© 2018-2024 StarFive Technology
All rights reserved

2. Enter the **Graphics support** menu.

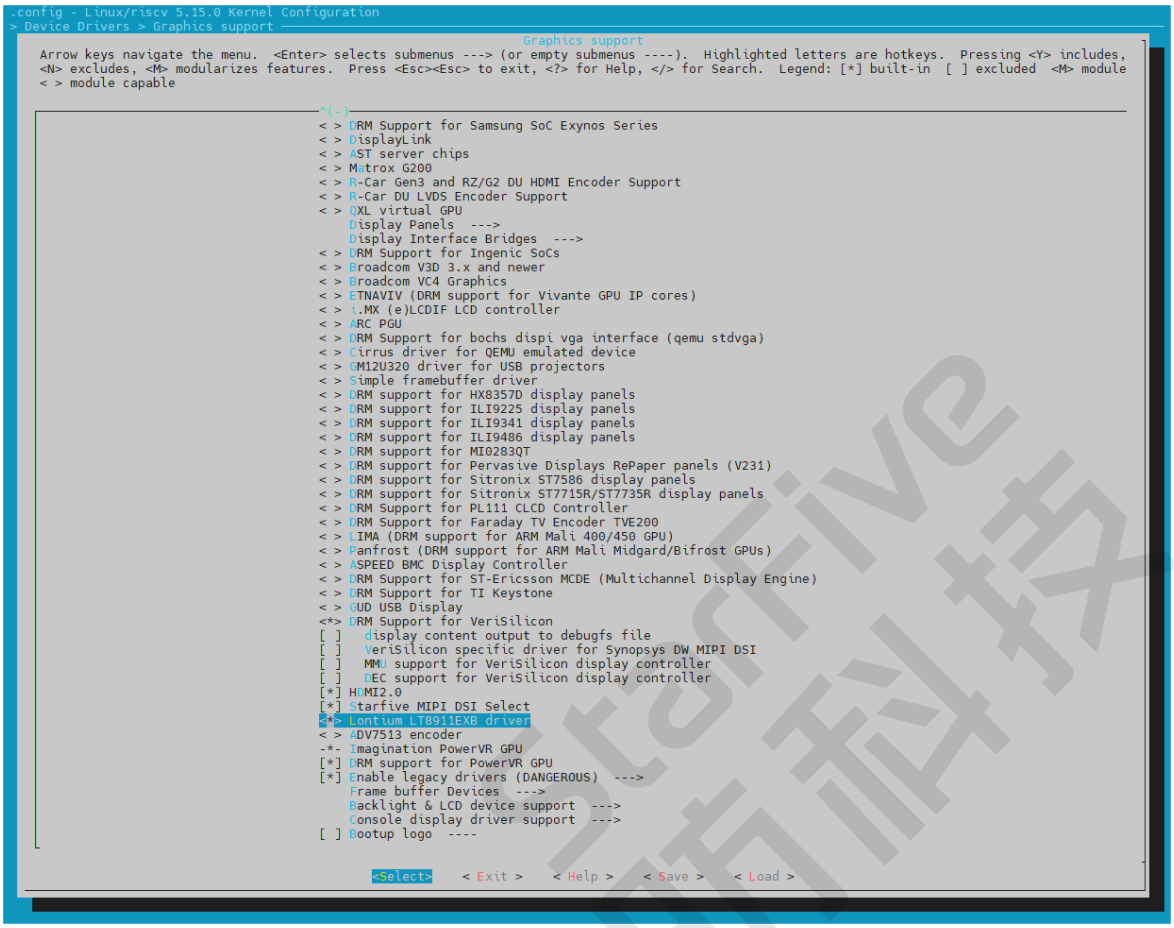
Figure 4-8 Graphics Support



3. Select **Lontium LT8911EXB driver**.

© 2018-2024 StarFive Technology
All rights reserved

Figure 4-9 Lontium LT8911EXB



4. Save your change before you exit the kernel configuration dialog.

4.1.2. Display Interface Description

4.1.2.1. Enable eDP

The function has the following parameters.

- **Function**: `lt8911exb_panel_enable`.
- **Description**: The function is used to enable the display of eDP, initialize lane configuration and DSI configuration, then turn on backlight and power of eDP.
- **Prototype**: `static int lt8911exb_panel_enable(struct drm_panel *panel)`.

4.1.2.2. Disable eDP

The function has the following parameters.

© 2018-2024 StarFive Technology
All rights reserved

- **Function**: `lt8911exb_panel_disable`.
- **Description**: The function is used to turn down backlight and power of eDP.
- **Prototype**: `static int lt8911exb_panel_disable(struct drm_panel *panel)`.

4.1.2.3. Obtain eDP Information

The function has the following parameters.

- **Function**: `lt8911exb_panel_get_modes`.
- **Description**: The function is used to get registered information of panel.
- **Prototype**: `static int lt8911exb_panel_get_modes(struct drm_panel *panel, struct drm_connector *connector)`

4.2. U-Boot Dispaly Module

4.2.1. U-Boot Display Driver

4.2.1.1. Device Tree Source Code

The following lists show the address of the display drivers.

- U-Boot display driver:

```
/u-boot/drivers/video/starfive
/u-boot/drivers/video/starfive_edp.c
```

- Device tree of JH-7110 DevKit:

```
/u-boot/arch/riscv/dts/starfive_devkits.dts
/u-boot/arch/riscv/dts/starfive_devkits-u-boot.dtsi
```

4.2.1.2. Device Tree Configuration

A DTS/DTSI file is used to store all the device tree configuration.

vout DTS Port

In the file `starfive_devkits.dts`, to configure `vout` dts port, you can find the device tree configuration of `vout` DSI as the following code block:

© 2018-2024 StarFive Technology
All rights reserved

```
/u-boot/arch/riscv/dts/starfive_devkits.dts
    &dc8200 {
        status = "okay";

        vopb_out: port {
                #address-cells = <1>;
                #size-cells = <0>;
                vopb_out_mipi: endpoint@0 {
                        reg = <0>;
                        remote-endpoint = <&mipi_in_vopb>;
                };

                vopb_out_hdmi: endpoint@1 {
                        reg = <1>;
                        remote-endpoint = <&hdmi_in_vopb>;
                };

        };
};
    &mipi_dsi0 {
        status = "okay";
        starfive,panel = <&seeed_panel>;
        data-lanes-num = <1>;
        status = "okay";

        ports {
                mipi_in: port {
                        #address-cells = <1>;
                        #size-cells = <0>;
                        mipi_in_vopb: endpoint@0 {
                                reg = <0>;
                                remote-endpoint = <&vopb_out_mipi>;
                        };
                };
        };

};

    &hdmi{
        pinctrl-names = "default";
        pinctrl-0 = <&hdmi_pins>;
        status = "okay";

        ports {
                hdmi_in: port {
                        #address-cells = <1>;
                        #size-cells = <0>;
                        hdmi_in_vopb: endpoint@0 {
                                reg = <0>;
```


© 2018-2024 StarFive Technology
All rights reserved

```
                                remote-endpoint = <&vopb_out_hdmi>;
                        };
                };
        };
};
```

The following list provides explanations for the parameters included in the above code block.

- **status**: The work status of the LCD module. To enable the module, set this bit as "`okay`" or to disable the module, set this bit as "`disabled`".
- **reg**: Register base address "`0x295e0000`" and range "`0x10000`".
- **port**: The port(s) used by the LCD driver.
- **data-lanes-num**: The number of data lanes.

I2C2

In the file `jh7110-devkits.dts`, to configure `lt8911exb`, the `lt8911exb` dts port should be added into `i2c2`. You can find the device tree configuration of `i2c2` as the following code block:

```
linux/arch/riscv/boot/dts/starfive/jh7110-devkits.dts:
&i2c2 {
        clock-frequency = <100000>;
        i2c-sda-hold-time-ns = <300>;
        i2c-sda-falling-time-ns = <510>;
        i2c-scl-falling-time-ns = <510>;
        auto_calc_scl_lhcnt;
        pinctrl-names = "default";
        pinctrl-0 = <&i2c2_pins>;
        status = "okay";

        lt8911exb_i2c@29 {
                compatible = "lontium,lt8911exb";
                reg = <0x29>;
                reset-gpio = <&gpio 41 1>;
                pwm-gpio = <&gpio 33 1>;
                bl-gpio = <&ext_gpio 6 GPIO_ACTIVE_LOW>;

                port {
                        lt8911exb_in: endpoint {
                                remote-endpoint = <&dsi0_output>;
                        };
                };
        };
};
```

In the above code block, the parameters of **pinctrl-names** and **pinctrl-0** are used to configure the `i2c2` pin configuration settings.

© 2018-2024 StarFive Technology
All rights reserved

&MIPI DST

In the file `jh7110-devkits.dts`, configure remote-end point for connecting DSI and panel. You can find the device tree configuration of &MIPI DST as the following code block:

```
linux/arch/riscv/boot/dts/starfive/jh7110-devkits.dts:
&mipi_dsi {
        status = "okay";

        ports {
                #address-cells = <1>;
                #size-cells = <0>;

                port@0 {
                        reg = <0>;
                        #address-cells = <1>;
                        #size-cells = <0>;

                        dsi0_output: endpoint@0 {
                                reg = <0>;
                                remote-endpoint = <&lt8911exb_in>;
                        };

                        dsi1_output: endpoint@1 {
                                reg = <1>;
                                remote-endpoint = <&panel_dsi_port>;
                        };

                        dsi2_output: endpoint@2 {
                                reg = <2>;
                                remote-endpoint = <&radxa_in>;
                        };
                };

                port@1{
                        reg = <1>;
                        dsi_in_port: endpoint {
                                remote-endpoint = <&mipi_out>;
                        };
                };

        };
};
```

4.2.1.3. U-Boot Menu Configuration

Follow the steps below to enable the configuration for U-Boot.

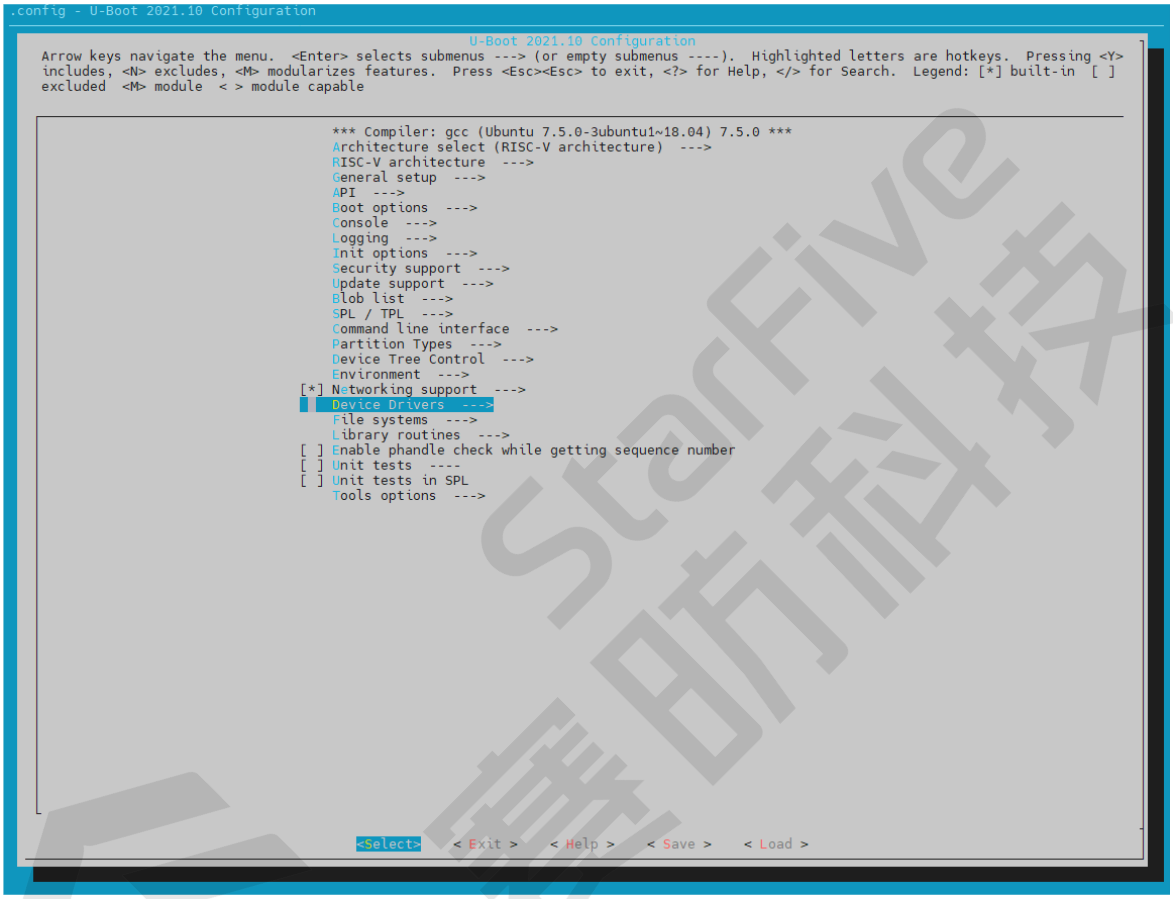
© 2018-2024 StarFive Technology
All rights reserved

1. Under the root directory of `freelight-u-sdk`, type the following command to enter the u-boot menu configuration GUI.

```
make uboot-menuconfig
```

2. Enter the **Device Drivers** menu.

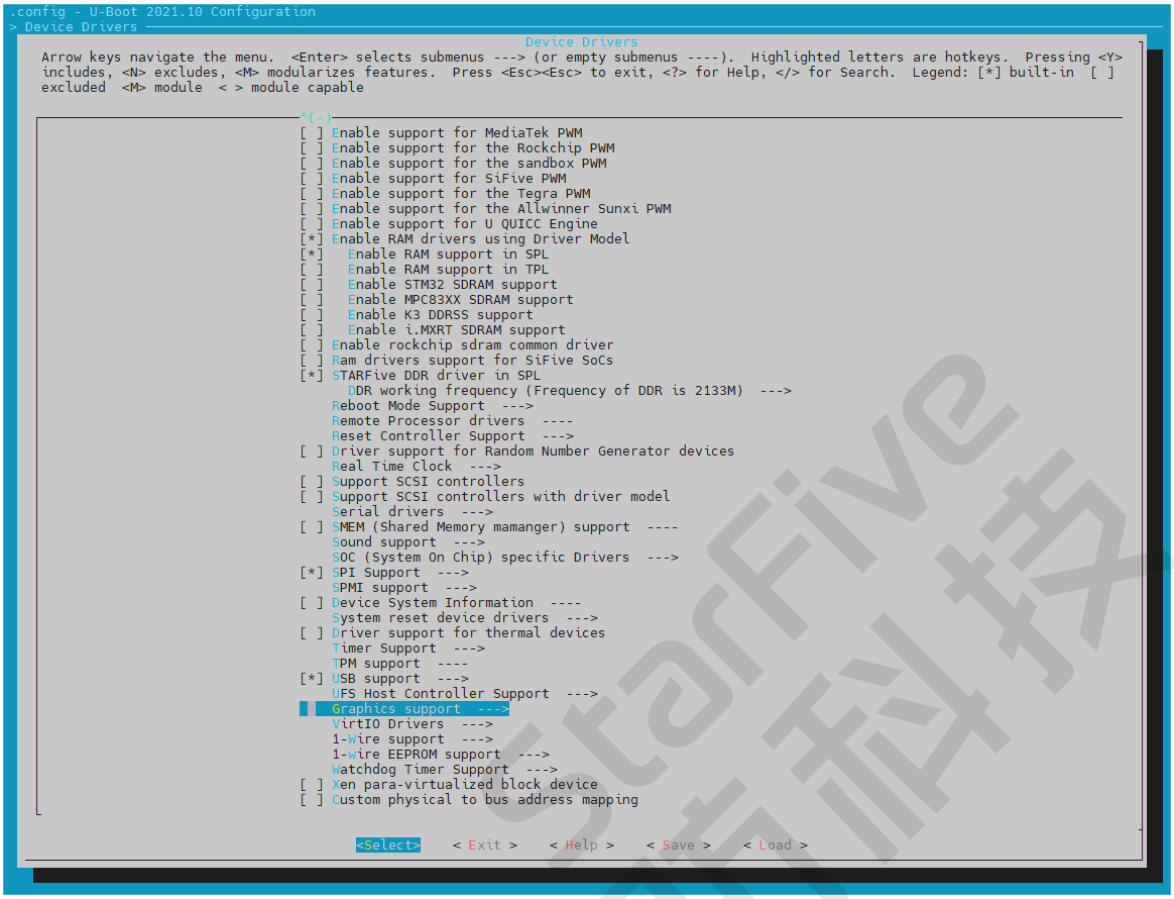
Figure 4-10 Device Drivers



3. Enter the **Graphics support** menu.

© 2018-2024 StarFive Technology
All rights reserved

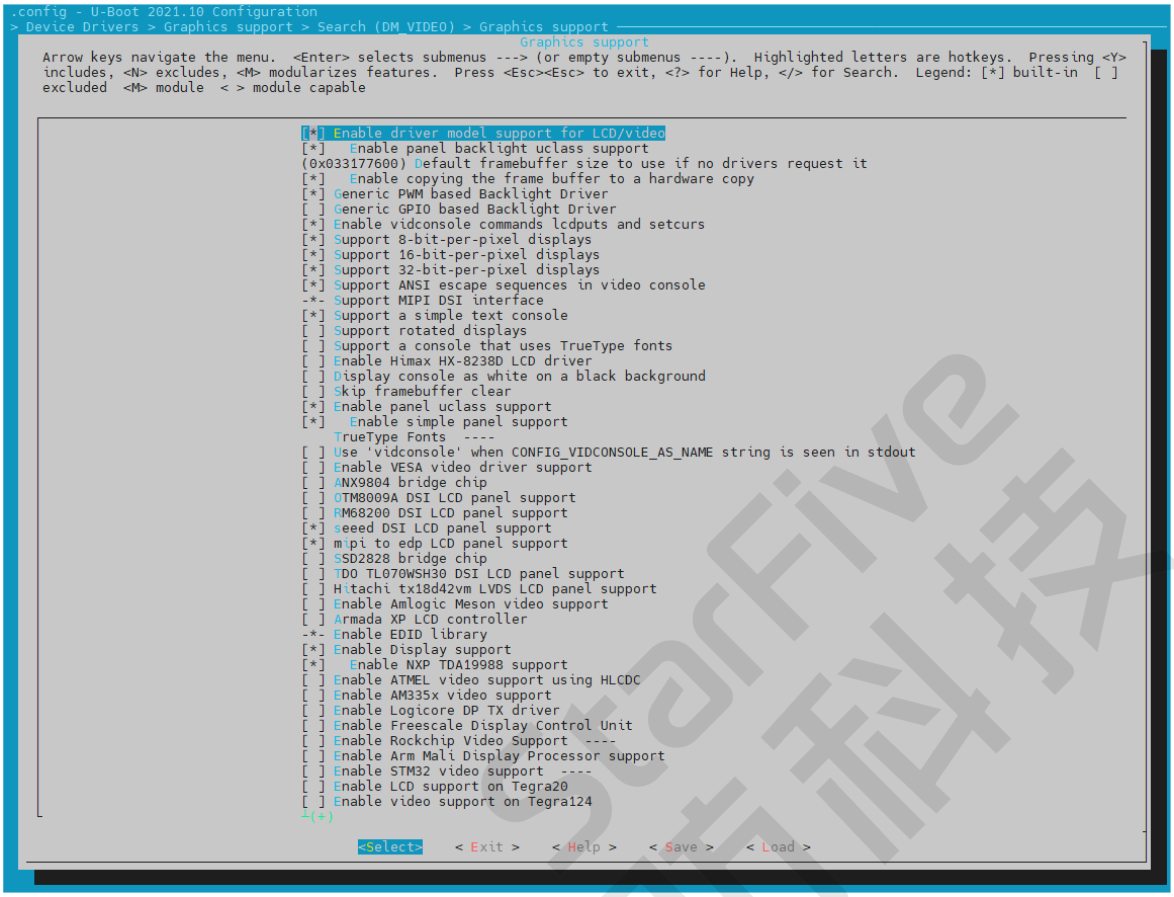
Figure 4-11 Graphics support



4. Select **Enable driver model support for LCD/video**.

© 2018-2024 StarFive Technology
All rights reserved

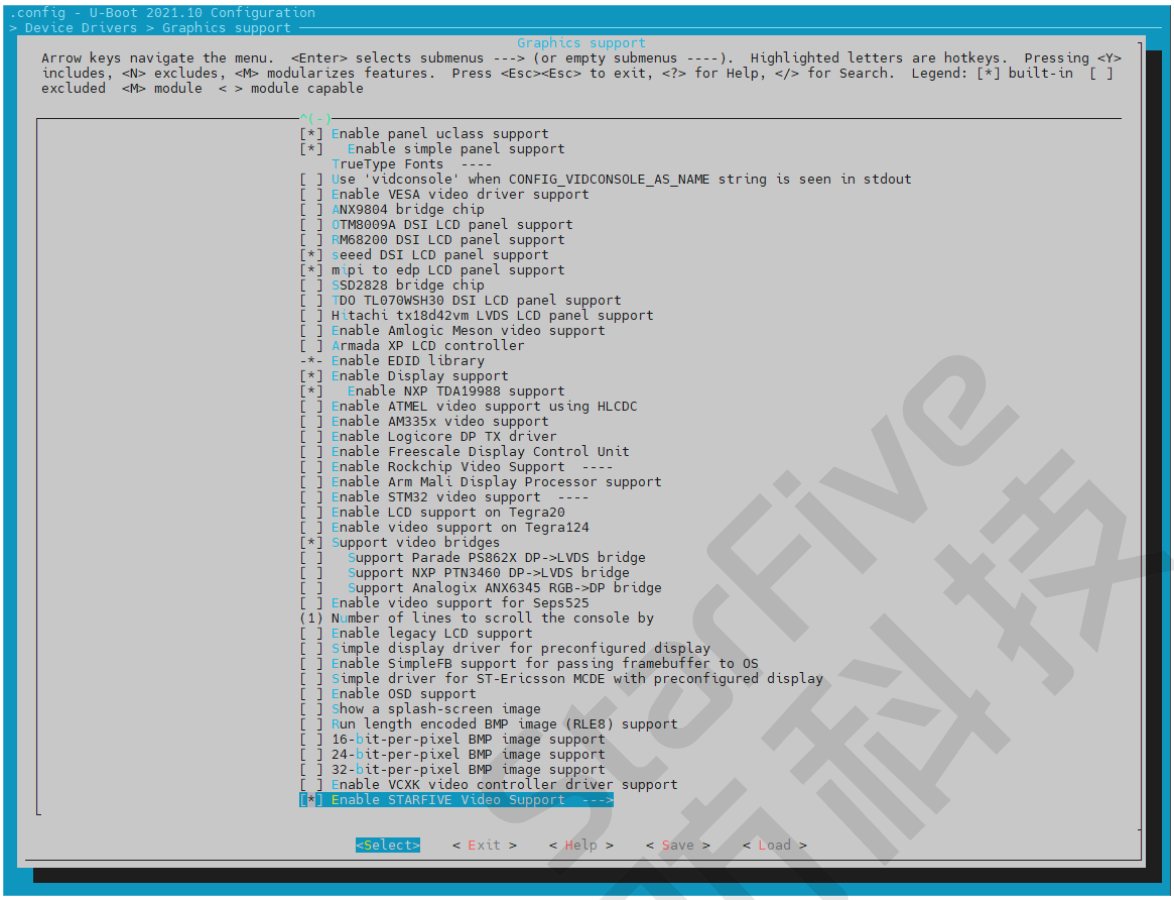
Figure 4-12 Enable driver model support for LCD/video



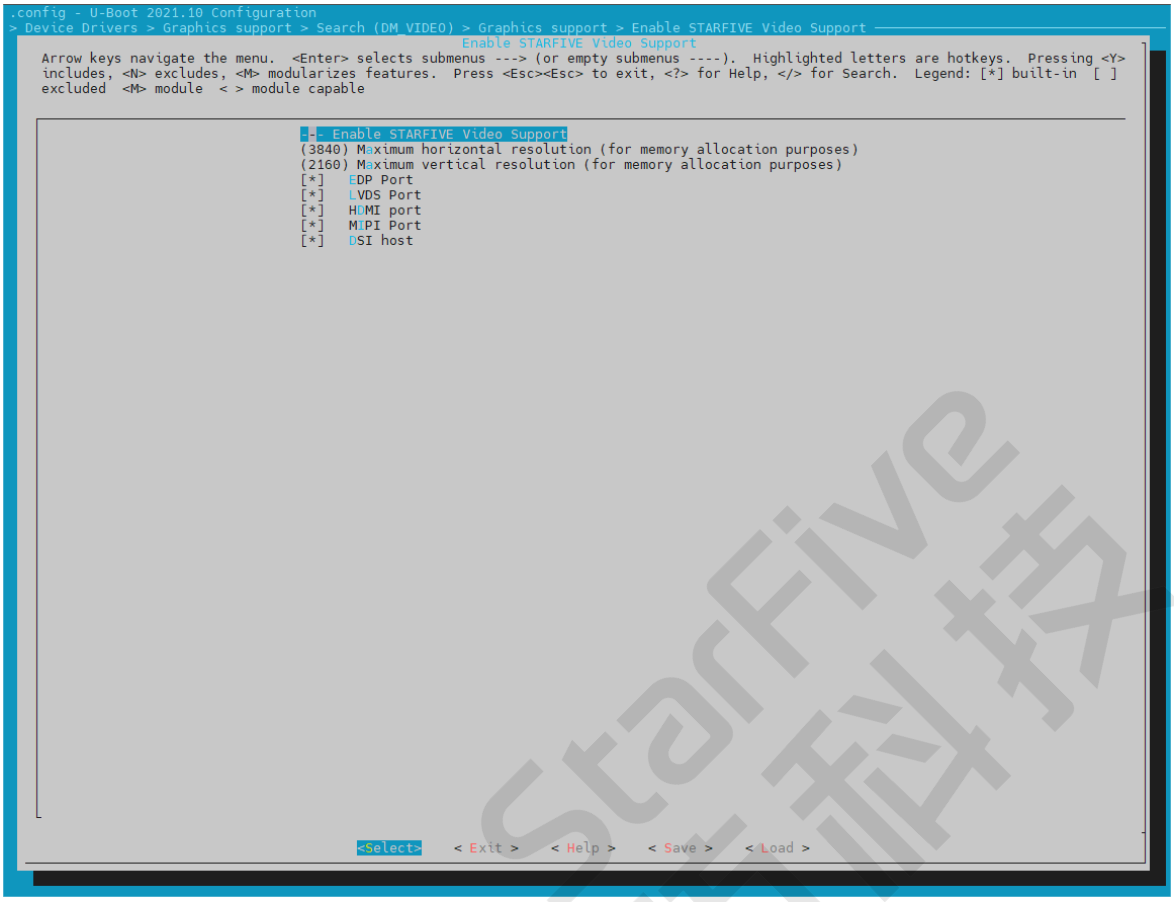
5. Select **Enable STARFIVE Video Support**.

© 2018-2024 StarFive Technology
All rights reserved

Figure 4-13 Enable STARFIVE Video Support



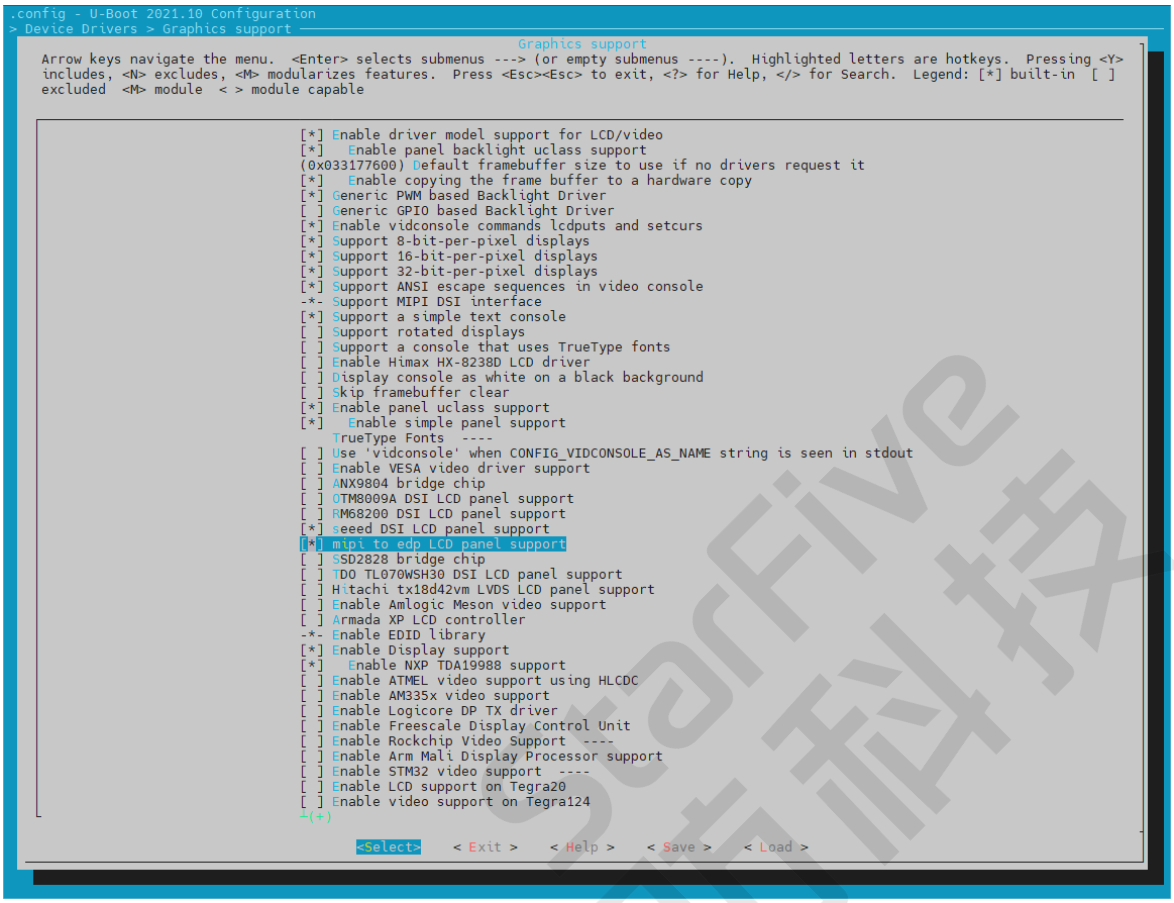
© 2018-2024 StarFive Technology
All rights reserved



6. Select **mipi to edp LCD panel support**.

© 2018-2024 StarFive Technology
All rights reserved

Figure 4-15 mipi to edp LCD panel support



7. Save your change before you exit the kernel configuration dialog.

4.2.2. Display Interface Description

4.2.2.1. Enable eDP

The function has the following parameters.

- **Function**: `edp_panel_enable_backlight`
- **Description**: The function is used to enable the display of eDP, initialize lane configuration and DSI configuration, then turn on backlight and power of eDP.
- **Prototype**: `static int edp_panel_enable_backlight(struct udevice *dev)`.

Note:
This function will be called by system automatically while the eDP and display related configuration is opened.

© 2018-2024 StarFive Technology
All rights reserved

4.2.2.2. Obtain eDP Information

The function has the following parameters.

- **Function**: `edp_panel_get_display_timing`
- **Description**: The function is used to get registered information of panel.
- **Prototype**: `static int edp_panel_get_display_timing(struct udevice *dev, struct display_timing *timings)`.

Note:
This function will be called by system automatically while the eDP and display related configuration is opened.

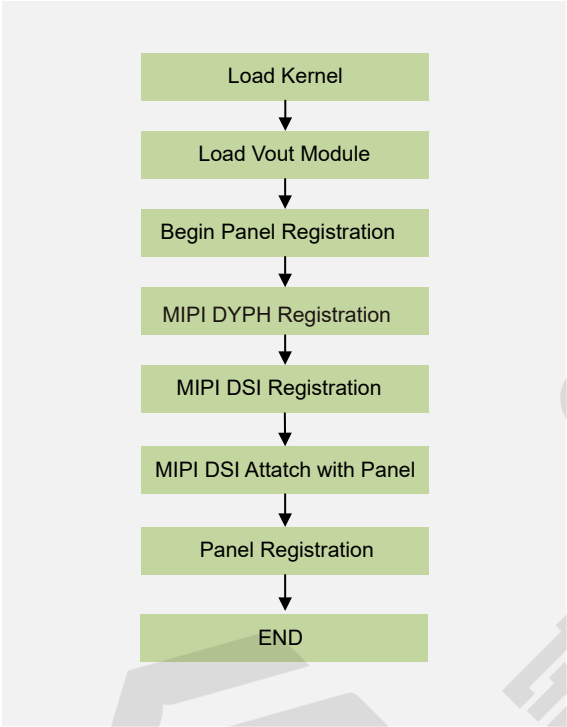
© 2018-2024 StarFive Technology
All rights reserved

5. Work Process

5.1. Initialization Process of Kernel

The following diagram shows the eDP initialization process for JH-7110 DevKit.

Figure 5-1 Initialization Process

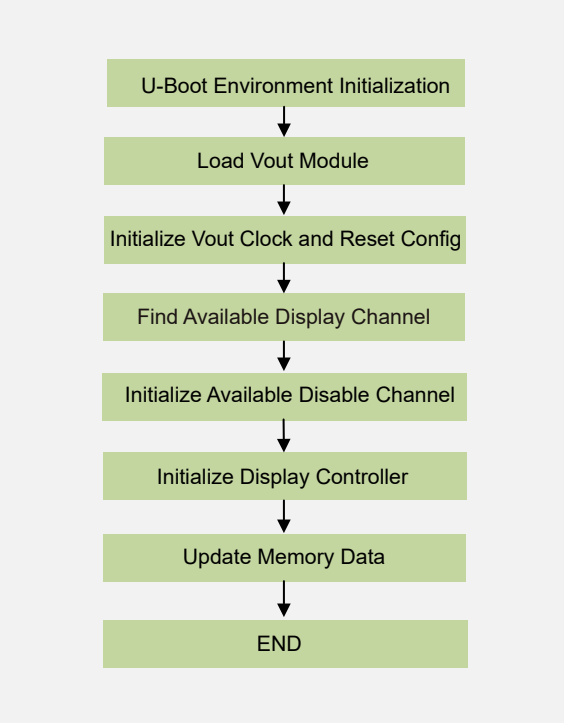


5.2. Initialization Process of U-Boot

The following diagram shows the U-Boot initialization process for JH-7110 DevKit.

© 2018-2024 StarFive Technology
All rights reserved

Figure 5-2 Initialization Process



© 2018-2024 StarFive Technology
All rights reserved

6. Kernel Parameter Configuration

This chapter describes how to configure the parameters of eDP to JH-7110 DevKit, the eDP model is BOE-NV140FHM-N46.

Note:
U-Boot does not support parameter configuration.

6.1. DSI Tool

This chapter mainly provides a tool for users to calculate JH7110 DSI timing and gives an instruction on how to set the calculated timing into the Linux driver to light screen. The tool package (`dsi_tool_v2.0`) includes the following parts:

- `StarFive_DSI_Tool_v2.0.exe`
- `input.md`
- `output.txt`
- `./run.sh`: Script to be used to generate `output.txt`

6.1.1. The `input.md` File

This file provides example parameters for MIPI DSI timing calculate, which can be modified and configured by users based on actual parameters.

Figure 6-1 The `input.md` File

input_gen_dsi.md

```
;bpp pixelclock hactive hfront_porch hback_porch hsync_len vactive vfront_porch vback_porch vsync_len dlanes dsi_hblank_ratio:r_hsa r_hbp r_hfp pixclk_source
24 148500000 1920 88 148 44 1080 4 36 5 4 2 2 2 1188000000
```

The description of the parameters in `input.md` file are as follows:

- **bpp**: Bit per pixel
- **pixelclock***: The pixel clock
- **hactive***: Horizontal effective pixels
- **hfront_porch***: Horizontal front porch, delay before horizontal synchronization signal
- **hback_porch***: Horizontal back porch, delay after horizontal synchronization signal
- **hsync_len***: Horizontal pulse width, the length of the horizontal synchronization signal
- **vactive***: Vertical effective pixels
- **vfront_porch***: Vertical front porch, delay before vertical synchronization signal

© 2018-2024 StarFive Technology
All rights reserved

- **vback_porch***: Vertical back porch, delay after vertical synchronization signal
- **vsync_len***: Vertical pulse width, the length of the vertical synchronization signal
- **dlanes**: MIPI DSI lane number
- **dsi_hblank_ratio**: The ratio of **hsa**, **hbp**, and **hfp** of DSI timing. It is used to allocate the horizontal blanking of the final calculated DSI timing, which can be `0`. If any of the three values are `0`, it means you will use the default ratio.
- **r_hsa**: The ratio of **hsa** of DSI timing, which can be `0`. If the value is `0`, the DSI **hsa**, **hbp**, **hfp** will use the default ratio.
- **r_hbp**: The ratio of **hbp** of DSI timing, which can be `0`. If the value is `0`, the DSI **hsa**, **hbp**, **hfp** will use the default ratio.
- **r_hfp**: The ratio of **hfp** of DSI timing, which can be `0`. If the value is `0`, the DSI **hsa**, **hbp**, **hfp** will use the default ratio.
- **pixclk_source**: Pixel clock source, corresponding to PLL2 of JH-7110 (default is 1188M). To make changes to PLL2, you will have to modify this parameter.

Note:
"*" means you can get the value from the screen manual or manufacturer.

6.1.2. The `output.txt` File

This file is the timing result generated by the tool, which is calculated based on `input.md` after running the `./run.sh` script. The following is an example `output.txt`.

© 2018-2024 StarFive Technology
All rights reserved

Figure 6-2 Example Output

dsi_output.txt

```
Welcome to use StarFive DSI timing generation tool v2.0

{
.dpi_timing = { //1920x1080, 24 bits, 60.00 Hz
    .pixelclock    = 148500000,
    .hactive       = 1920      ,
    .hfront_porch  = 88        ,
    .hback_porch   = 148       ,
    .hsync_len     = 44        ,
    .vactive       = 1080      ,
    .vfront_porch  = 4         ,
    .vback_porch   = 36        ,
    .vsync_len     = 5         ,
},
.dsi_timing = {
    .dlanes        = 4         ,
    .bitrate       = 900000000,
    .hsa           = 306       ,
    .hbp           = 304       ,
    .hfp           = 297       ,
    .hact          = 5760      ,
},
},
```

Due to the fact that the frequency of `pixelclock` is integer divided by JH-7110 PLL2, the `pixelclock` in `output.txt` here may change accordingly.

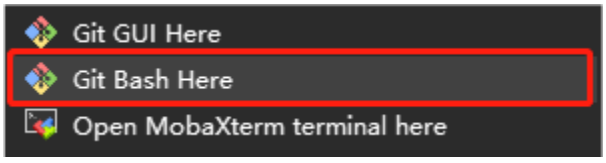
To apply this timing in Linux, refer to Configuration *(on page 38)*.

6.2. Configuration

6.2.1. Use DSI Tool

1. Install Git and open the `dsi_tool_v2.0` folder.
2. Right-click under the `dsi_tool_v2.0` folder to show options.
3. Choose **Git Bash Here** option, and open it.

Figure 6-3 Git Bash Here

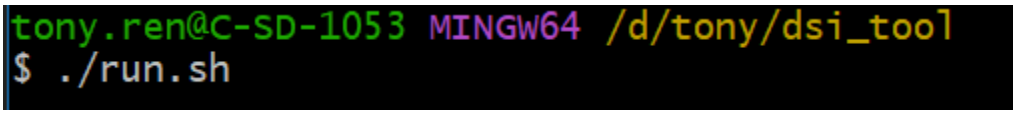


© 2018-2024 StarFive Technology
All rights reserved

4. Run the following to generate `output.txt`:

```
./run.sh
```

Figure 6-4 ./run.sh



Note:

If the screen manufacturer does not provide the ratio of **dsi_hblank_ratio**, you can use the default ratio in the following two methods:

- Set the value of **r_hsa**, **r_hbp**, or **r_hfp** in the `input.md` file to 0.
- Generally use **r_hsa (2)**, **r_ hbp(2)**, **r_ hfp(2)**.

6.2.2. Configuration Example

This section provides an example to configure the parameters in kernel. The following code block is an example of the output file:

```
{
.dpi_timing = {
        .pixelclock   = 148500000, //div 10, 60 fps
        .hactive      = 1920, //w
        .hfront_porch = 88,
        .hback_porch  = 148,
        .hsync_len    = 44,
        .vactive      = 1080, //h
        .vfront_porch = 34,
        .vback_porch  = 6,
        .vsync_len    = 5,
    },
    .dsi_timing = {
        .dlanes       = 4,
        .bitrate      = 900000000,
        .hsa          = 147,//132,
        .hbp          = 480,//508,
        .hfp          = 280,//268,
        .hact         = 5760,
    },
},
```


© 2018-2024 StarFive Technology
All rights reserved

To Configure eDP

`dpi_timing` is the timing of the eDP, which needs to be added in the panel driver. It corresponds to `jh7110_lt8911exb.c` under the path of `\linux\drivers\gpu\drm\verisilicon`, which maps `lt8911exb_panel_modes`.

1. Enter `jh7110_lt8911exb.c` file.
2. Modify the parameters of PLL 1188M under this function according to the output file.

Figure 6-5 Modify the Parameters

```
static const struct drm_display_mode lt8911exb_panel_modes[] = {
        {// pll 1188M   60fps
                .clock = 148500000 / 1000,
                .hdisplay = 1920,
                .hsync_start = 1920 + 88,
                .hsync_end = 1920 + 88 + 148,
                .htotal = 1920 + 88 + 148 + 44,
                .vdisplay = 1080,
                .vsync_start = 1080 + 34,
                .vsync_end = 1080 + 34 + 6,
                .vtotal = 1080 + 34 + 6 + 5,
        },
};
```

Tip:

- clock = pixelclock/1000
- hdisplay = hactive
- hsync_start = hactive + hfp
- hsync_end = hactive + hfp + hbp
- htotal = hactive + hfp + hbp+ hsync_len
- vdisplay = vactive
- vsync_start = vactive + vfp
- vsync_end = vactive + vfp + vbp
- vtotal = vactive + vfp + vbp + vsync_len

3. Perform the following to update the MIPI_Timing[] parameters:

```
static int MIPI_Timing[] =
// hfp, hs,     hbp,    hact,   htotal, vfp,    vs,     vbp,    vact,
 vtotal, pixel_CLK/10000
//-----|---|------|-------|--------|-----|-----|-----|--------|--------
|---------------
{88, 44, 148, 1920, 2200, 34, 5, 6, 1080, 1125, 15000};   // boe config
 for linux
```

Figure 6-6 MIPI_Timing[]

```
static int MIPI_Timing[] =
// hfp, hs,     hbp,    hact,   htotal, vfp,    vs,     vbp,    vact,   vtotal, pixel_CLK/10000
//-----|---|------|-------|--------|-----|-----|-----|--------|--------|---------------
{88, 44, 148, 1920, 2200, 34, 5, 6, 1080, 1125, 15000};   // boe config for linux
```


© 2018-2024 StarFive Technology
All rights reserved

4. After configuration, it will be synchronized to the DC controller driver and eDP DSI driver. For example, to update DSI_Timing lanes to **4**, you can change the corresponding parameters in the input file.

   The corresponding driver code is shown in the following screen:

Figure 6-7 driver_code

```
static int lt8911exb_dsi_probe(struct mipi_dsi_device *dsi)
{
        int ret;

        dsi->lanes = 4;

        dsi->mode_flags = (MIPI_DSI_MODE_VIDEO |
                           MIPI_DSI_MODE_VIDEO_SYNC_PULSE |
                           MIPI_DSI_MODE_LPM);
        dsi->format = MIPI_DSI_FMT_RGB888;
        dsi->hs_rate = 900000000;
        ret = mipi_dsi_attach(dsi);
        if (ret < 0) {
                dev_err(&dsi->dev, "failed to attach dsi to host\n");
                return ret;
        }

        return ret;
}
```

To Configure MIPI DSI

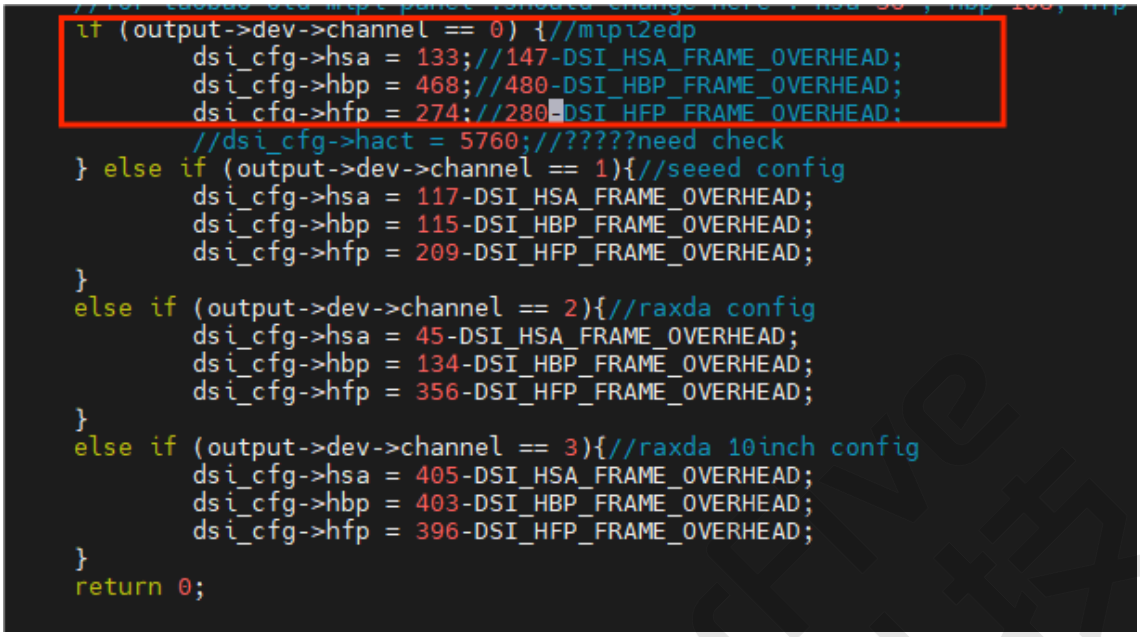
The following is the DSI timing in output.txt we generated before. Follow the steps below to configure MIPI DSI.

```
.dsi_timing = {
        .dlanes       = 4,
        .bitrate      = 900000000,
        .hsa          = 147,//132,
        .hbp          = 480,//508,
        .hfp          = 280,//268,
        .hact         = 5760,
    },
```

1. Open `jh7110_lt8911exb.c` file under the path of `\linux\drivers\gpu\drm\verisilicon` in Linux. In JH-7110 DevKit, the channel number of eDP is channel 0.
2. Locate `cdns_dsi_mode2cfg` function to modify the **hsa**, **hbp** and **hfp** parameter according to the value in the **output.txt** generated previously.

© 2018-2024 StarFive Technology
All rights reserved

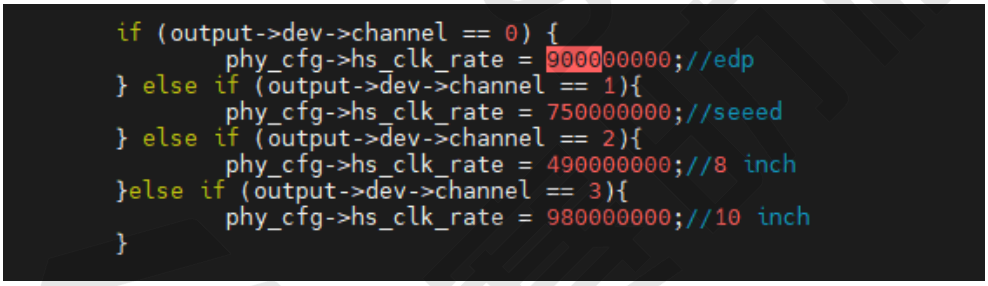
Figure 6-8 hsa hbp hfp



```
if (output->dev->channel == 0) {//mipi2edp
        dsi_cfg->hsa = 133;//147-DSI_HSA_FRAME_OVERHEAD;
        dsi_cfg->hbp = 468;//480-DSI_HBP_FRAME_OVERHEAD;
        dsi_cfg->hfp = 274;//280-DSI_HFP_FRAME_OVERHEAD;
        //dsi_cfg->hact = 5760;//?????need check
} else if (output->dev->channel == 1){//seeed config
        dsi_cfg->hsa = 117-DSI_HSA_FRAME_OVERHEAD;
        dsi_cfg->hbp = 115-DSI_HBP_FRAME_OVERHEAD;
        dsi_cfg->hfp = 209-DSI_HFP_FRAME_OVERHEAD;
}
else if (output->dev->channel == 2){//raxda config
        dsi_cfg->hsa = 45-DSI_HSA_FRAME_OVERHEAD;
        dsi_cfg->hbp = 134-DSI_HBP_FRAME_OVERHEAD;
        dsi_cfg->hfp = 356-DSI_HFP_FRAME_OVERHEAD;
}
else if (output->dev->channel == 3){//raxda 10inch config
        dsi_cfg->hsa = 405-DSI_HSA_FRAME_OVERHEAD;
        dsi_cfg->hbp = 403-DSI_HBP_FRAME_OVERHEAD;
        dsi_cfg->hfp = 396-DSI_HFP_FRAME_OVERHEAD;
}
return 0;
```

3. Then locate `cdns_dsi_adjust_phy_config` function to modify the **bitrate** parameter according to the value in the **output.txt** generated previously.

Figure 6-9 bitrate



```
if (output->dev->channel == 0) {
        phy_cfg->hs_clk_rate = 900000000;//edp
} else if (output->dev->channel == 1){
        phy_cfg->hs_clk_rate = 750000000;//seeed
} else if (output->dev->channel == 2){
        phy_cfg->hs_clk_rate = 490000000;//8 inch
}else if (output->dev->channel == 3){
        phy_cfg->hs_clk_rate = 980000000;//10 inch
}
```

Tip:

The **hs_clk_rate** in the figure means **bitrate**.

© 2018-2024 StarFive Technology
All rights reserved

7. Debug eDP

7.1. Test Case Configuration

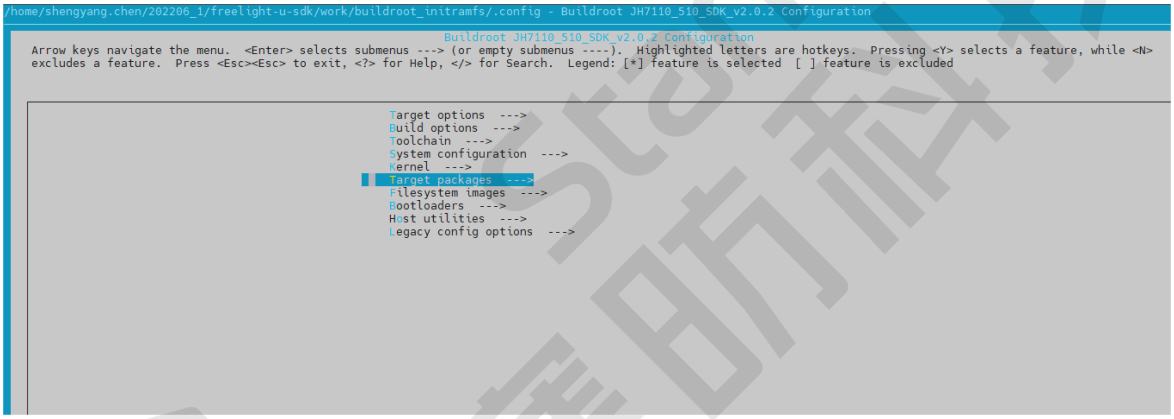
Follow the steps below to enable the kernel configuration.

1. Under the root directory of `freelight-u-sdk`, type the following command to enter the kernel menu configuration GUI.

```
make buildroot_initramfs-menuconfig
```

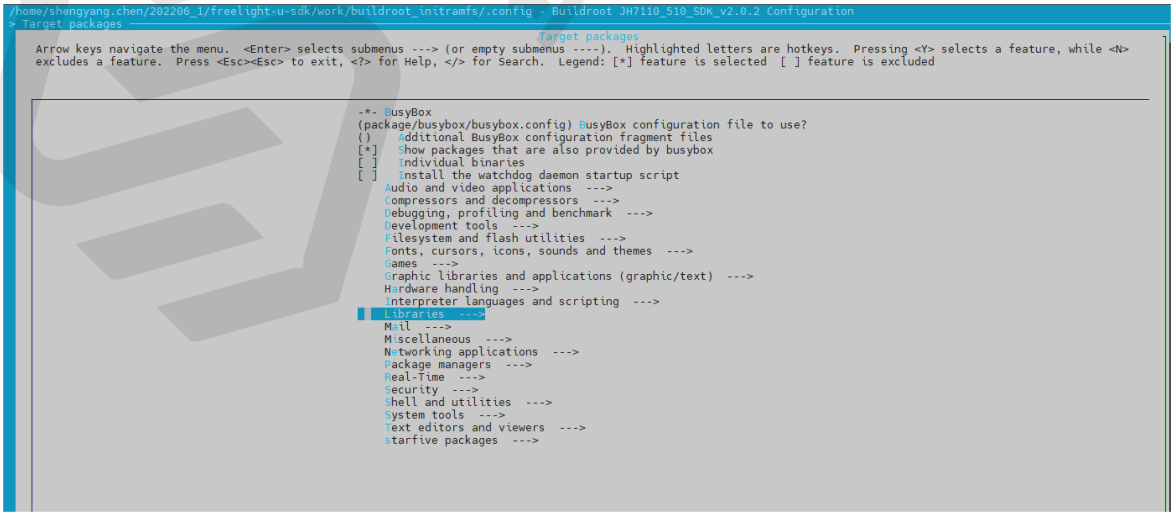
2. Enter the **Target packages** menu.

Figure 7-1 Target Packages



3. Enter the **Libraries** menu.

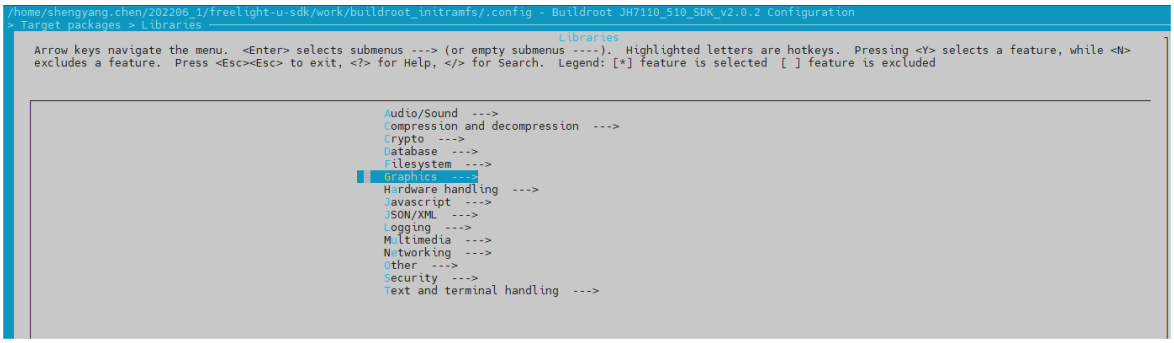
Figure 7-2 Libraries



4. Enter the **Graphics** menu.

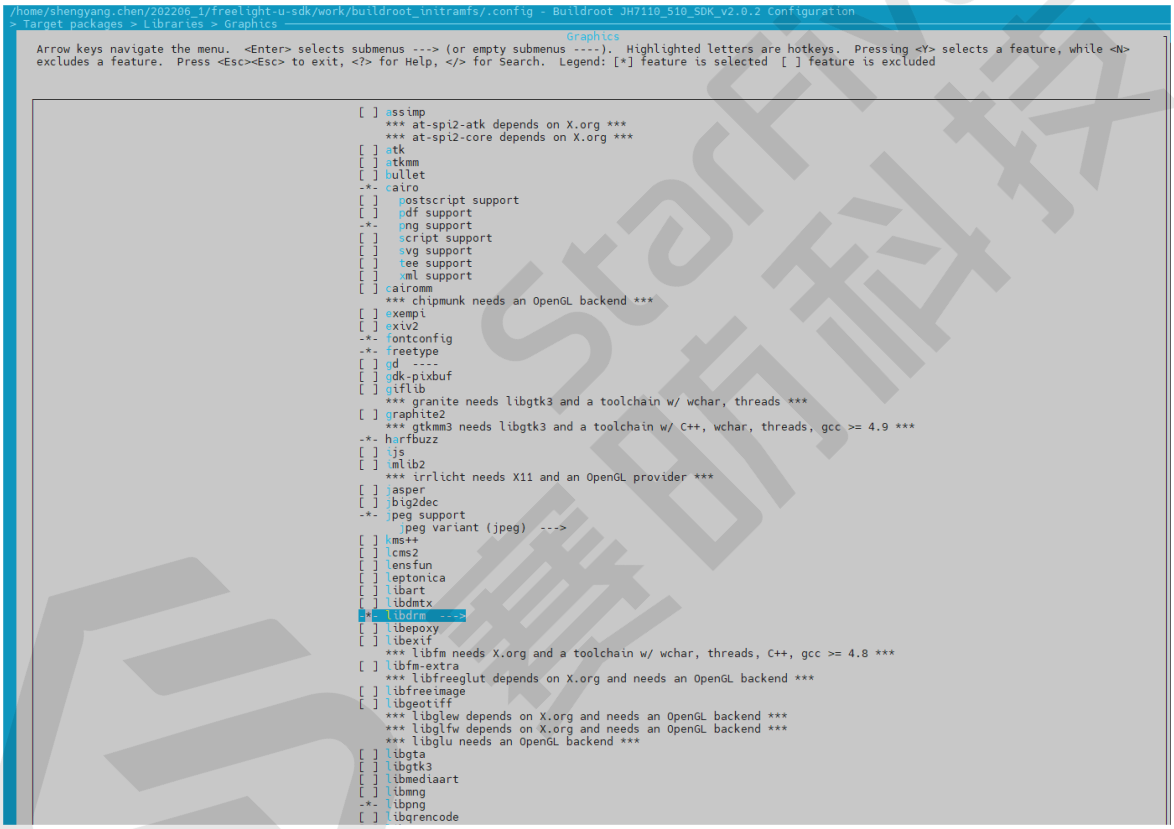
© 2018-2024 StarFive Technology
All rights reserved

Figure 7-3 Graphics



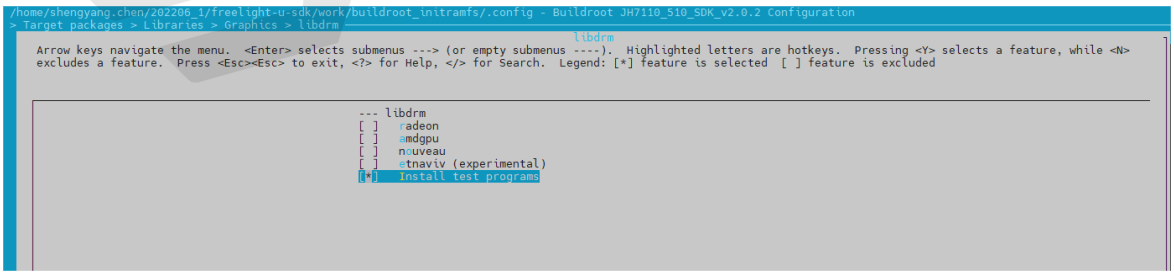
5. Enter the **libdrm** menu.

Figure 7-4 libdrm



6. Select the **Install test programs** option, or you may select ALL options under this menu.

Figure 7-5 Install Test Programs



© 2018-2024 StarFive Technology
All rights reserved

Result: After you have completed all the above configuration, you can use the `modetest` tool in kernel for testing.

7. Save your change before you exit the kernel configuration dialog.

7.2. Before Debug

Before debugging the MIPI controller, make sure you see the following screen in the start-up logs.

Figure 7-6 Start-up Logs

```
[   12.190735] of_cfs_init
[   12.193217] of_cfs_init: OK
[   12.196524] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[   12.314845] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[   12.326364] starfive soc:display-subsystem: bound 29400000.dc8200 (ops 0xffffffff80e76720)
[   12.334695] innohdmi-starfive 29590000.hdmi: inno hdmi bind begin
[   12.341911] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
[   12.350596] cfg80211: failed to load regulatory.db
[   12.351087] innohdmi-starfive 29590000.hdmi: [drm:inno_hdmi_bind] registered Inno HDMI I2C bus driver success
[   12.365560] innohdmi-starfive 29590000.hdmi: HDMI&AUDIO register done.
[   12.372175] innohdmi-starfive 29590000.hdmi: inno hdmi bind end
[   12.378113] starfive soc:display-subsystem: bound 29590000.hdmi (ops 0xffffffff80e774d0)
[   12.386232] vs-simple-encoder soc:rgb-output: encoder_bind begin
[   12.392340] no panel, -517
[   12.395057] vs-simple-encoder soc:rgb-output: encoder_bind error
[   12.401086] starfive soc:display-subsystem: bound soc:rgb-output (ops 0xffffffff80e77118)
[   12.409298] vs-simple-encoder soc:dsi-output: encoder_bind begin
[   12.415383] cdns-dsi 295d0000.mipi: ===>cdns_dsi_bridge_attach begin
[   12.421758] cdns-dsi 295d0000.mipi: ===>cdns_dsi_bridge_attach end
[   12.427968] vs-simple-encoder soc:dsi-output: encoder_bind end
[   12.433828] starfive soc:display-subsystem: bound soc:dsi-output (ops 0xffffffff80e77118)
[   12.442874] [drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1
[   14.488355] ALSA device list:
[   14.491337]   #0: Starfive-HDMI-Sound-Card
[   14.498788] Freeing unused kernel image (initmem) memory: 2196K
[   14.504844] Run /init as init process
[   14.508536]   with arguments:
[   14.511509]     /init
[   14.513785]   with environment:
[   14.516934]     HOME=/
[   14.519311]     TERM=linux
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Starting mdev... OK
```

The content in the red box indicates the MIPI work status. Make sure the log lines showing in the red box are printed before debug.

7.3. Debug eDP

1. Follow the steps in Test Case Configuration *(on page 43)* to configure the test environment.

   Note:
   Make sure you have configured **libdrm** and **modetest** before compiling and burning an image.

2. After you have completed the kernel start-up, use the following command to verify the display functions and connection status.

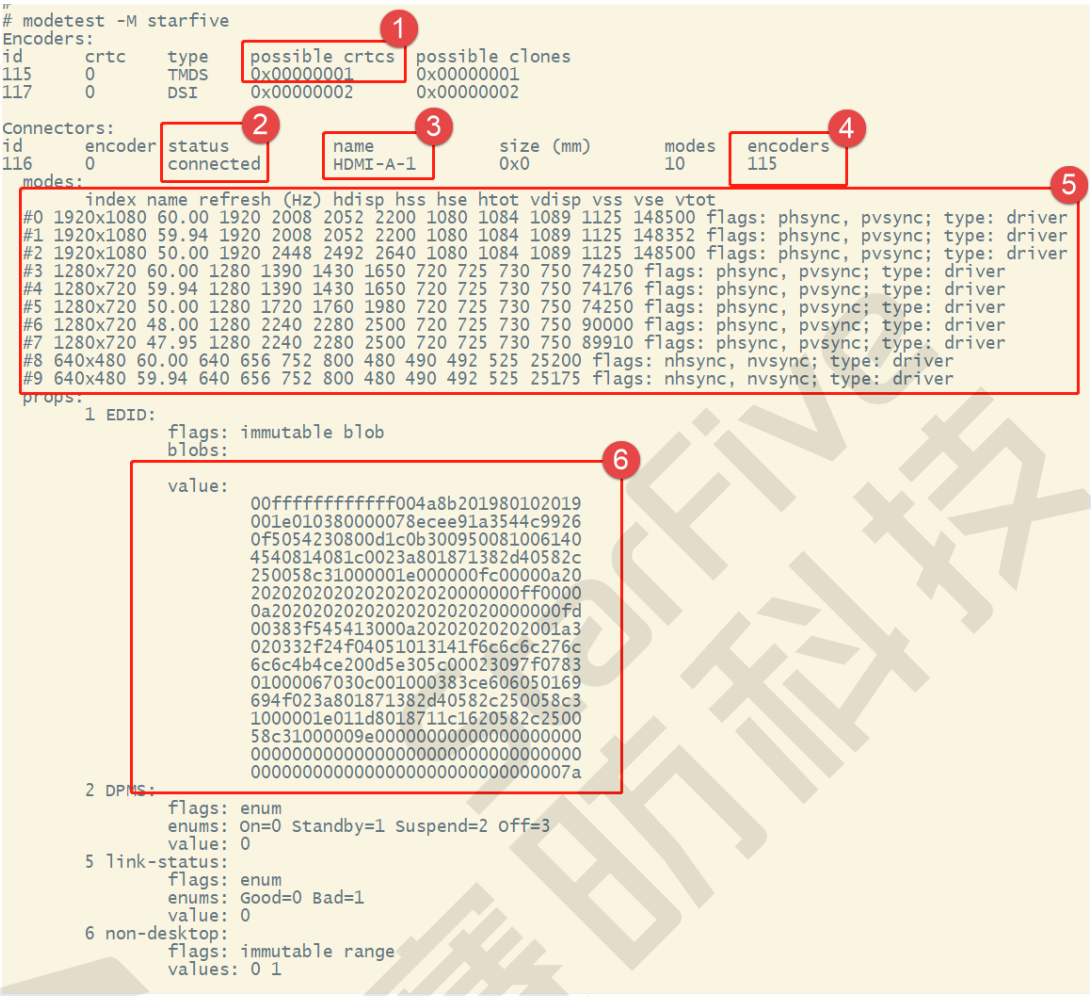
   ```
   modetest -M starfive
   ```

   The following legends and tables display an example output and descriptions.

© 2018-2024 StarFive Technology
All rights reserved

- Debug output 1:

Figure 7-7 Debug Display 1




```
# modetest -M starfive
Encoders:
id      crtc    type    possible crtcs  possible clones
115     0       TMDS    0x00000001      0x00000001
117     0       DSI     0x00000002      0x00000002

Connectors:
id      encoder status          name            size (mm)       modes   encoders
116     0       connected       HDMI-A-1        0x0             10      115
  modes:
        index name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot
  #0 1920x1080 60.00 1920 2008 2052 2200 1080 1084 1089 1125 148500 flags: phsync, pvsync; type: driver
  #1 1920x1080 59.94 1920 2008 2052 2200 1080 1084 1089 1125 148352 flags: phsync, pvsync; type: driver
  #2 1920x1080 50.00 1920 2448 2492 2640 1080 1084 1089 1125 148500 flags: phsync, pvsync; type: driver
  #3 1280x720 60.00 1280 1390 1430 1650 720 725 730 750 74250 flags: phsync, pvsync; type: driver
  #4 1280x720 59.94 1280 1390 1430 1650 720 725 730 750 74176 flags: phsync, pvsync; type: driver
  #5 1280x720 50.00 1280 1720 1760 1980 720 725 730 750 74250 flags: phsync, pvsync; type: driver
  #6 1280x720 48.00 1280 2240 2280 2500 720 725 730 750 90000 flags: phsync, pvsync; type: driver
  #7 1280x720 47.95 1280 2240 2280 2500 720 725 730 750 89910 flags: phsync, pvsync; type: driver
  #8 640x480 60.00 640 656 752 800 480 490 492 525 25200 flags: nhsync, nvsync; type: driver
  #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver
  props:
        1 EDID:
                flags: immutable blob
                blobs:

                value:
                        00ffffffffffff004a8b201980102019
                        001e010380000078ecee91a3544c9926
                        0f5054230800d1c0b300950081006140
                        4540814081c0023a801871382d40582c
                        250058c31000001e000000fc00000a20
                        2020202020202020202020000000ff0000
                        0a2020202020202020202020000000fd
                        00383f545413000a202020202020001a3
                        020332f24f04051013141f6c6c6c276c
                        6c6c4b4ce200d5e305c00023097f0783
                        01000067030c001000383ce606050169
                        694f023a801871382d40582c250058c3
                        1000001e011d8018711c1620582c2500
                        58c31000009e000000000000000000000
                        000000000000000000000000000000000
                        000000000000000000000000000007a
        2 DPMS:
                flags: enum
                enums: On=0 Standby=1 Suspend=2 Off=3
                value: 0
        5 link-status:
                flags: enum
                enums: Good=0 Bad=1
                value: 0
        6 non-desktop:
                flags: immutable range
                values: 0 1
```

Table 7-1 Debug Display 1

Legend	Label	Description
①	**possible crtcs**	Available *Cathode Ray Tube Controller (CRTC)* devices
②	**status**	Whether the display connector is connected or not
③	**name**	The name (type) of the display connector
④	**encoders**	The connected encoders
⑤	**modes**	The supported display modes
⑥	**value**	The *Extended Display Identification Data (EDID)* of the screen

- Debug output 2:

© 2018-2024 StarFive Technology
All rights reserved

Figure 7-8 Debug Display 2

© 2018-2024 StarFive Technology
All rights reserved

Table 7-2 Debug Display 2

Legend	Label	Description
①	**id**	The CRTC 0x00000001 mentioned in row ① of table Table 7-1 : Debug Display 1 *(on page 46)*, which means the CRTC is available for use.
②	**id**	The CRTC 0x00000002 mentioned in row ① of table Table 7-1 : Debug Display 1 *(on page 46)*, which means the CRTC is available for use.

Note:
If the displayed CRTC is 0x00000003, both of the CRTCs are available for use.

- Debug output 3:

Figure 7-9 Debug Display 3

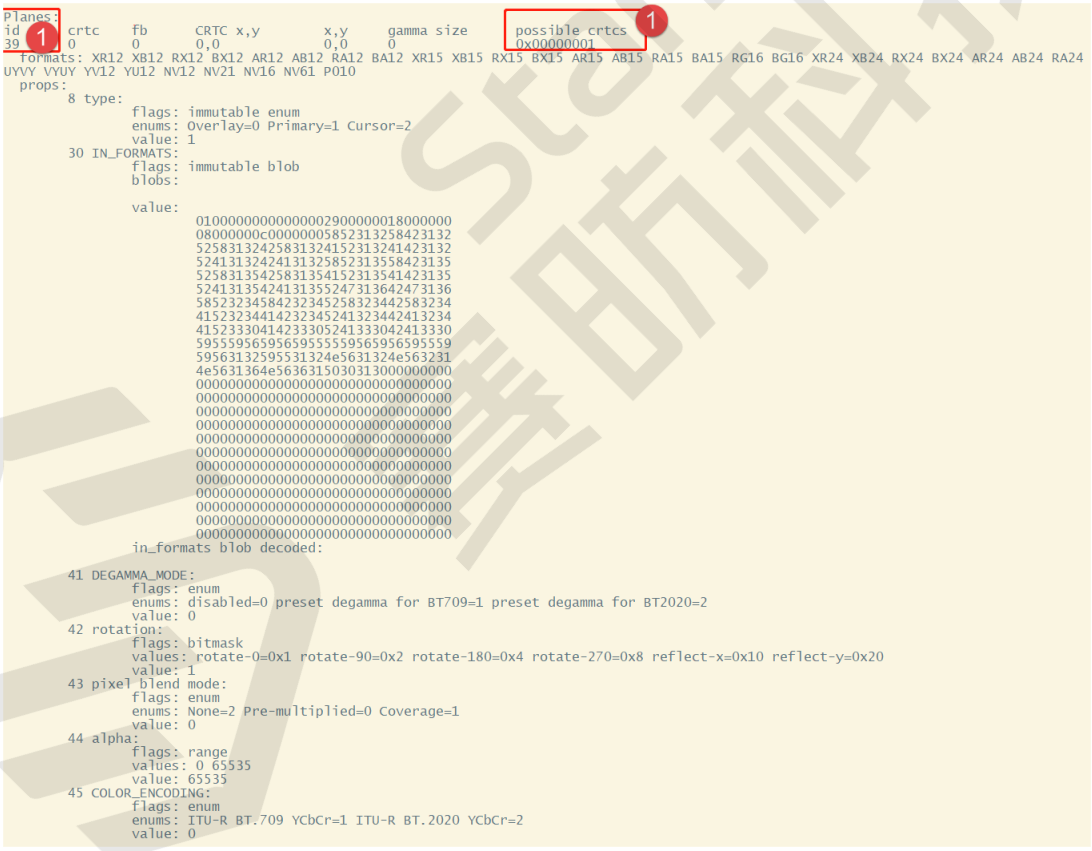


Table 7-3 Debug Display 3

Legend	Description
①	The CRTC and its connected plane

© 2018-2024 StarFive Technology
All rights reserved

7.4. Test Example

For eDP Output

The following command shows an example for testing the LCD output.

```
modetest -M starfive -D 0 -a -s 118@35:1920x1080 -P 74@35:1920x1080@RG16
 -Ftiles
```

The following list provides explanations for the parameters in the above example command.

- **118@35:1920x1080** - <Connector ID>@<CRTC ID>: <Resolution>
- **74@35:1920x1080@RG16** - <Plane ID>@<CRTC ID>: <Resolution>@<Format>

Output Result

The following photo shows the output generated from the above example command.

Figure 7-10 Test Example

© 2018-2024 StarFive Technology
All rights reserved