

# JH7110 MIPI LCD Developing and Porting Guide

VisionFive 2 Version: 1.1 Date: 2023/06/09 Doc ID: JH7110-PGEN-004

# Legal Statements

Important legal notice before reading this documentation.

#### **PROPRIETARY NOTICE**

Copyright © Shanghai StarFive Technology Co., Ltd., 2023. 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: <u>sales@starfivetech.com</u>
- Support: <u>support@starfivetech.com</u>

# Contents

List of Tables	4
List of Figures	5
Legal Statements	ii
Preface	vi
1. Introduction	8
2. IC Specification	9
3. Module Introduction	12
3.1. Display Driver	12
3.1.1. Display Driver Locations	
3.1.2. Device Tree configuration	12
3.1.3. Driver Configuration	15
3.1.4. Kernel Menu Configuration	15
3.2. Display Interface Description	19
3.2.1. Enable LCD	19
3.2.2. Disable LCD	19
3.2.3. Obtain LCD Information	20
4. Work Process	21
4.1. Initialization Process	21
5. MIPI Parameter Configuration	22
5.1. DSI Tool	22
5.1.1. The input.md File	22
5.1.2. The output.txt File	23
5.2. Configuration	24
5.2.1. Use DSI Tool	24
5.2.2. Configuration for 1C2L	25
5.2.3. Configuration for 1C4L	
6. Debug LCD	
6.1. Test Case Configuration	32
6.2. Before Debug	34
6.3. Debug LCD	34
6.4. Test Example	

# List of Tables

Table 0-1 Revision History	vi
Table 6-1 Debug Display 1	35
Table 6-2 Debug Display 2	
Table 6-3 Debug Display 3	

# List of Figures

Figure 2-1 Block Digram of DSITX	10
Figure 2-2 Block Digram of M31D-PHY	11
Figure 3-1 Device Drivers	16
Figure 3-2 Graphics support	17
Figure 3-3 DRM Support	17
Figure 3-4 Starfive MIPI DSI Select	
Figure 3-5 Device Drivers	
Figure 3-6 Starfive M31 MIPI DPHY TX Driver	19
Figure 4-1 Initialization Process	
Figure 5-1 The input.md File	
Figure 5-2 Example Output	24
Figure 5-3 Git Bash Here	24
Figure 5-4 ./run.sh	25
Figure 5-5 Modify the Parameters	26
Figure 5-6 Example Input	27
Figure 5-7 Driver Code	27
Figure 5-8 hsa hbp hfp	28
Figure 5-9 Bitrate	
Figure 5-10 Modify the Parameters	29
Figure 5-11 Driver Code	
Figure 5-12 hsa hbp hfp	31
Figure 5-13 Bitrate	
Figure 6-1 Target Packages	32
Figure 6-2 Libraries	32
Figure 6-3 Graphics	33
Figure 6-4 libdrm	33
Figure 6-5 Install Test Programs	33
Figure 6-6 Start-up Logs	34
Figure 6-7 Debug Display 1	35
Figure 6-8 Debug Display 2	
Figure 6-9 Debug Display 3	37
Figure 6-10 MIPI Connect Status	38
Figure 6-11 Test Example	

# Preface

About this guide and technical support information.

#### About this document

This document mainly provides the SDK developers with the developing and porting instructions for the LCD module of the StarFive next generation SoC platform - JH7110.

#### Audience

This document mainly serves the LCD relevant driver developers. If you are developing and porting other modules, place a request to your sales or support consultant for our complete documentation set on JH7110.

#### **Revision History**

#### Table 0-1 Revision History

Version	Released	Revision
1.1	2023/06/09	Added <u>MIPI Parameter Configuration (on page 22)</u> .
1.0	2023/03/10	First official release.

#### Notes and notices

The following notes and notices might appear in this guide:

*i* 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.



Indicates that an action or step can result in physical harm or cause damage to hardware.

# 1. Introduction

This document is intended to:

- Introduce the porting procedures of the StarFive JH7110 Software Development Kit (SDK) to a new platform.
- Provide instructions for porting a new LCD screen to the JH7110 SoC platform.
- Instruct on how to write an LCD driver.
- Provide an example of a typical LCD interface configuration.

The code sources referenced in this document are based on the following conditions:

- SDK version: 3.0
- U-Boot version: 3.0
- Linux Kernel version: 5.15
- For different U-Boot or Linux Kernel versions, these references may be slightly different.

# 2. IC Specification

The StarFive JH7110 LCD module uses Cadence IP. The following is the information about Cadence IP.

#### **MIPI DSITX CADENCE**

The Cadence MIPI DSI v1.3.1 TX Controller IP (DSITX) provides an interface that receives data and control from the host processor display system using either the DPI, DSC or SDI input bus interfaces. The DSITX will translate the incoming pixel information and control signals into an internal packed byte format, in the case of DPI and DSC, or pass in the pre-packed SDI byte format, before the internal byte format data is packeted and sent to the MIPI DSI Compatible display via the D-PHY physical interface. It supports video and command mode displays and can work in dual-display mode using virtual channel identification on the packets.

The supported MIPI display types are (at least) type 1 (command only), type 2 (video plus a partialframe buffer in command mode), type 3 (video display with some programming capability in command mode).

In video mode, all the regular modes are supported (Non-Burst with Sync Pulses, Non-Burst with Sync Events and same for Burst mode).

Command mode is used for any panel that integrates a display controller and frame buffer. Data is passed to the display using a command message followed by data pixels and/or parameter messages. The host side can also perform read and write to the panel registers and frame buffer using the bidirectional lane on the DPHY.

#### Figure 2-1 Block Digram of DSITX



#### **MIPI DPHY M31**

MIPI DPHY M31 has the following features:

- Supports standard 8b PPI interface compliant of MIPI D-PHY Specification.
- Supports 1 Clock Lane and up to 4 Data Lanes scalability in DPHY mode.
- Supports independent (1 Clock Lane & up to 2 Data Lanes) X 2 in DPHY mode.
- Supports HS-Rx Data rate from 80Mbps up to 1.5 Gbps. (DPHY)
- Supports LS-Rx Data rate of 10Mbps & Ultra-low power mode.
- Supports Triggers, ULPS and LPDT.
- Supports on-die terminated and non-terminated operation, integrates switchable termination.
- Supports Fault Detection of Sequence Error. (Error Report)
- Supports clock and data lane swapping function.
- Build-in BISTTX for at-speed testing.

#### **IP Overview**

M31DPHYRX611TL028D\_00151501 integrates 2 Clock Lane and up to 4 Data Lanes MIPI D-PHY compatible high-speed receiver that supports 80 Mbps up to 1.5 GHz data transfer rate, and low-

power receiver that supports data transfer in 10 Mbps. This IP supports D-PHY v1.1 specification. It is optimized with low power design for mobile CSI-2/DSI slave application. The interface of M31D-PHY is designed for standard MIPI D-PHY PPI interface. It can be easily integrated into the lane management of each customer's CSI-2 controller. The following is the block digram of IP.





# 3. Module Introduction

# 3.1. Display Driver

## 3.1.1. Display Driver Locations

The following list shows the address of the display drivers.

• Linux Kernel Display Driver:

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

• Device tree:

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

## 3.1.2. Device Tree configuration

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

The device tree of MIPI LCD is stored in the following path:

linux-5.10/arch/riscv/boot/dts/starfive/

The following code block shows the DTS file structure for MIPI LCD.

```
linux-5.15.0
L-- arch
L-- | -- riscv
| -- | -- | -- boot
| -- | -- | -- dts
| -- | -- | -- | -- starfive
| -- | -- | -- | -- jh7110-common.dtsi
| -- | -- | -- | -- | -- jh7110.dtsi
```

#### **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>;
   req-names = "dsi";
   clocks = <&clkvout JH7110_U0_CDNS_DSITX_CLK_SYS>,
     <&clkvout JH7110 U0 CDNS DSITX CLK APB>,
     <&clkvout JH7110 U0 CDNS DSITX CLK TXESC>,
     <&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";
   port {
    dsi_out_port: endpoint@0 {
     remote-endpoint = <&panel_dsi_port>;
    };
    dsi_in_port: endpoint@1 {
    remote-endpoint = <&mipi_out>;
    };
   };
   mipi_panel: panel@0 {
   /*compatible = "";*/
   status = "okay";
   };
  };
linux/arch/riscv/boot/dts/starfive/jh7110-common.dtsi:
&mipi_dsi {
 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 above register.

- clocks: The clocks used by the LCD module.
- clock-names: The names of the above clocks.
- resets: The reset signals used by the LCD module.
- reset-names: The names of the above reset signals.
- **phys**: The phys used by the LCD module.
- phy-names: The name of the phys.
- 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".
- port: The port(s) used by the LCD driver.

#### **MIPI DPHY**

In the file jh7110.dtsi, 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>;
  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";
  };
linux/arch/riscv/boot/dts/starfive/jh7110-common.dts:
&mipi_dphy {
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 "0x295e0000" and range "0x10000".
- clocks: The clocks used by the LCD module.
- clock-names: The names of the above clocks.
- resets: The reset signals used by the LCD module.

- reset-names: The names of the above reset signals.
- 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".

#### I2C2

In the file jh7110-common.dtsi, in order to configure LCD DTS port, the seeed\_panel 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-common.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";
 seeed_plane_i2c@45 {
  compatible = "seeed_panel";
 req = <0x45>;
 port {
  panel_dsi_port: endpoint {
    remote-endpoint = <&dsi_out_port>;
   };
  };
 };
};
```

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

## 3.1.3. Driver Configuration

The following code block shows the driver configuration.

```
CONFIG_DRM_VERISILICON=y
```

## 3.1.4. Kernel Menu Configuration

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

- | 3 Module Introduction
  - 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 3-1 Device Drivers

<pre>config - Linux/risev 5.15.0 Kernel Configuration Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus&gt;). Highlighted letters are hotkeys. Pressing <y> includes, &lt;+&gt; module capable  (eneral setup&gt; [*] MNU-based Paged Memory Management Support SoC selection&gt; PU errat a selection&gt; Pu errat a selection&gt; Pu errat a selecture-dependent options&gt; Power management options&gt; Pi anble the block layer&gt; Pi anble the block layer&gt; Pi Schedulers&gt; Pi</y></enter></pre>	ile Edit View Search Terminal Help
Arrow keys navigate the menu. <inter> selects submenus) (or enpty submenus). Highlighted letters are hotkeys. Pressing <y> includes,         </y></inter>	config - Linux/riscv 5.15.0 Kernel Configuration
<pre>Ceneral setup&gt; [*] Mil-based Paged Menory Mnangement Support SoC selection&gt; CPU errata selection&gt; Platform type&gt; Rernel features&gt; Boot options&gt; Ceneral architecture-dependent options&gt; (*) Enable the block layer&gt; Is Enable the block layer&gt; Executable fit formats&gt; Device Driveria&gt; Fit Braytens&gt; Fit Braytens&gt; Fit Braytens&gt; Rernel hacking&gt; Kernel hacking&gt; Kernel hacking&gt; Kernel hacking&gt; Kernel hacking&gt; Kernel hacking&gt;</pre>	Linux/riscv 5.15.0 Kernel Configuration Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus&gt;). Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt; module capable</m></esc></esc></m></n></y></enter>
<pre>cselect&gt; &lt; Exit &gt; &lt; Help &gt; &lt; Save &gt; &lt; Load &gt;</pre>	<pre>Ceneral setup&gt; [1] MMU-based Paged Memory Management Support Soc Selection&gt; CPU errata selection&gt; Platform type&gt; Boot options&gt; Ceneral architecture-dependent options&gt; Ceneral architecture-dependent options&gt; [*] Enable the block layer&gt; I Schedulers&gt; Executable file formats&gt; File Systems&gt; Security options&gt; File Systems&gt; Kernel hacking&gt; Kernel hacking&gt;</pre>
CELECED < EXIL > < Mech > < 2000 >	
	<pre>cselect&gt; &lt; Exit &gt; &lt; Help &gt; &lt; Save &gt; &lt; Load &gt;</pre>

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

# Figure 3.2 Graphics support Proceedings Proceedings

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

onfig - Linux/riscv 5.15.0 Kernel Configuration
Device Drivers-> Graphics support- Graphics support
Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus&gt;). Highlighted letters are hotkeys. Pressing &lt;&gt;&gt; includes, <n> excludes, <m> modularizes features. Press description of the second secon</m></n></enter>
reactions. Press addreades to exit, six for heipt six for bearchin, regener. [] excluded six module six module capable
[*] VGA Arbitration
<ul> <li>(10) MIX Multi Hulled TO GOUS</li> <li>-*- Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)&gt;</li> </ul>
IRM devices>
< > MD GPU
<pre>&lt; &gt; Nouveau (NVIDIA) cards <pre></pre></pre>
<pre>&lt; &gt; Virtual kms (experimental)</pre>
< > DisplayLink <> ST server chins
< > Matrox 6200
> ar users and to revise up not encoder Support < > R-Car DU LVDS Encoder Support
<> XL virtual CPU Control av Densel>
Display Interface Bridges>
<pre>&lt;&gt; ETNAUTU (DRM support for Vivante GPU IP cores) &lt;&gt;&gt; i.MX (e) (CDT F (C) controller</pre>
< > ARC PGU
v support for boors ouspilled interface (genu stovga) <> Cirrus driver for QEMU emulated device
<> CM120230 driver for USB projectors
Studyce trained rot to text > New Support of HX8357D display panels
<pre>&lt;&gt; DRM support for IL19225 display panels &lt;&gt;&gt; DRM support for IL19341 display panels</pre>
< > DRM support for ILI9486 display panels
<pre>&lt; &gt; DRM support for MI020300T &lt; &gt; DRM support for Pervasive Displays RePaper panels (V231)</pre>
CPM support for Sitronix ST7586 display panels
C > DNA support for structure structures structures and utspray particle <> CUD USB Display
ENS Support for Versilicon [] J. display content output to debunds file
VeriSilicon specific driver for Synopsys DW_MIPI DSI
[] MMI support for VerSilicon display controller [] DEC support for VerSilicon display controller
[*] HUNI2.0
<pre></pre>
-* : imagination PowerVR GPU
[*] [nable legacy drivers (DANGEROUS)>
Frame Dutter Devices> Backlight & LCD device support>
Console display driver support>
[ ] south togo
<pre><select> &lt; Exit &gt; &lt; Help &gt; &lt; Save &gt; &lt; Load &gt;</select></pre>

5. In the **Graphics support** menu, for MIPI output, select the **Starfive MIPI DSI Select** option.

Figure 3-3 DRM Support

#### Figure 3-4 Starfive MIPI DSI Select

.config - Linux/riscy 5.15.0 Kernel Configuration
Graphies support Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help, &gt; for Search. Legend: [*] built-in [ ] excluded <m> module &lt;&gt; module capable</m></esc></esc></m></n></y></enter>
<pre>[1] WA Arbitration [1] WA Arbitration [1] WA Arbitration [1] WA Arbitration [1] WA Arbitration [2] WA A</pre>
<pre>(*) tarfive MIPI DSI Select &lt; &gt; TROYSI3 encoder -* Tragitation PowerVR GPU [1] EMM support for PowerVR GPU [2] Emable Legacy drivers (DANGEROUS)&gt; Frame buffer Devices&gt; Backlight &amp; LCD device support&gt; Console display driver support&gt; [] Bootup logo&gt; [] Bootup logo</pre>
L Selects < Exit > < Help > < Save > < Load >

6. Then, to configure the M31 DPHY, back to **Device Drivers** menu to enter the **PHY Subsystem** menu.

includes, <n> excludes, <m excluded <m> module &lt; &gt;</m></m </n>	nu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> &gt; modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [ ] module capable</esc></esc></y></enter>
	<pre>(-) (*) (*) M ilbox Hardware Support (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)</pre>

7. Select Starfive M31 MIPI DPHY TX Driver.

#### Figure 3-6 Starfive M31 MIPI DPHY TX Driver

.config - Linux/riscv 5.15.0 Kernel Configuration > Device Drivers > PHY Subsystem —
PHY Subsystem Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt;&gt; module capable</m></esc></esc></m></n></y></enter>
-*- PHY Core < > CAN transceiver PHY < > Groadcom Kona USB2 PHY Driver < > Gadence Torrent PHY driver < > Gadence Serra PHY Driver < > Gadence Salvo PHY Driver < > Gadence Salvo PHY Driver < > Gadence Salvo PHY Driver < > Mixel MIPI DSI PHY Support < > Mixel MIPI DSI PHY Support < > Mirvell USB HSIC 28nm PHY Driver < > Mirvell USB 2.0 28nm PHY Driver < > Mirvell USB 2.0 28nm PHY Driver < > Mirvell USB PHY driver for Microsemi Ocelot 
<pre><select: <="" exit=""> &lt; Help &gt; &lt; Save &gt; &lt; Load &gt;</select:></pre>

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

# **3.2.** Display Interface Description

## 3.2.1. Enable LCD

The function has the following parameters.

- Function: seeed\_panel\_enable.
- **Description**: The function is used to enable the display of LCD, initialize lane configuration and DSI configuration, and then turn on the backlight and power of LCD.
- Prototype: Static int seeed\_panel\_enable(struct drm\_panel\* panel).

## 3.2.2. Disable LCD

The function has the following parameters.

- Function: seeed\_panel\_enable.
- Description: The function is used to turn down the backlight and power of LCD.
- Prototype: Static int seeed\_panel\_disable(struct drm\_panel\* panel).

## 3.2.3. Obtain LCD Information

The function has the following parameters.

- Function: seeed\_panel\_get\_modes.
- **Description**: The function is used to get registration information of panel.
- **Prototype**: Static int seeed\_panel\_get\_modes(struct drm\_panel\* panel, struct drm\_connector\* connector).

# 4. Work Process

# 4.1. Initialization Process

The following diagram shows the LCD initialization process for JH7110.

#### Figure 4-1 Initialization Process



# 5. MIPI Parameter Configuration

VisionFive 2 supports two MIPI display output channel, one is 1C2L (2-lane MIPI DSI), the other is 1C4L (4-lane MIPI DSI). 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 the screen.

# 5.1. 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  $(\underline{dsi\_tool\_v2.0})$  includes the following parts:

- StarFive\_DSI\_Tool\_v2.0.exe
- input.md
- output.txt
- ./run.sh: Script to be used to generate output.txt

## 5.1.1. 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 5-1 The input.md File

cycloand ;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
- 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 JH7110 (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.

## 5.1.2. 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.

5 - MIPI Parameter Configuration

#### Figure 5-2 Example Output

≡ dsi_o	output.txt
1	Welcome to use StarFive DSI timing generation tool v2.0
2	
3	{
4	.dpi_timing = { //1920x1080, 24 bits, 60.00 Hz
5	.pixelclock = 148500000,
6	.hactive = 1920,
7	
8	.hback_porch = 148,
9	.hsync_len = 44,
10	.vactive = 1080,
11	.vfront_porch = 4,
12	.vback_porch = 36 ,
13	.vsync_len = 5 ,
14	}.
15	.dsi_timing = {
16	.dlanes = 4 ,
17	.bitrate = 900000000,
18	.hsa = 306 ,
19	.hbp = 304 ,
20	.hfp = 297 ,
21	.hact = 5760 ,
22	}.
23	},

Due to the fact that the frequency of pixelclock is integer divided by JH7110 PLL2, the pixelclock in output.txt here may change accordingly.

To apply this timing in Linux, refer to Configuration (on page 24).

# 5.2. Configuration

## 5.2.1. Use DSI Tool

- 1. Install <u>Git</u> 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 5-3 Git Bash Here



4. Run the following to generate output.txt:

./run.sh

Figure 5-4 ./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).

## 5.2.2. Configuration for 1C2L

This section provides an example to configure the parameters in 1C2L MIPI driver. The following code block is an example of the output file:

```
{
.dpi_timing = { //800x480, 24 bits, 60.00 Hz
   .pixelclock = 29700000,
                 = 800
    .hactive
   .hfront_porch = 90
    .hback_porch = 5
    .hsync len
               = 5
    .vactive
               = 480
    .vfront_porch = 60
    .vback_porch = 5
   .vsync_len
                 = 5
},
.dsi_timing = {
    .dlanes
                 = 1
    .bitrate
                = 750000000,
    .hsa
                 = 36
    .hbp
                 = 108
    .hfp
                 = 288
    .hact
                 = 2400
},
},
```

#### | 5 - MIPI Parameter Configuration

 For the 1C2L MIPI channel, to support a new screen, it is necessary to modify the driver file starfive\_drm\_seeedpanel.c under /Linux/drivers/gpu/drm/ verisilicon.

This driver file servers as an i2c device by default, and the probe interface will use the i2c command to read the panel ID, and verify whether the current screen is connected properly based on the return value. When debugging a new screen, there are two cases:

- If the i2c command can read the panel ID, the probe function needs to support the i2c interface; In seed\_ panel\_ enable function, the panel should be enabled according to the specific command of i2c.
- If the i2c command cannot read panel ID, you are recommand to remove the i2c related command, then the probe process can be completed, so that the screen is in connected status by default.



Note:

It is necessary to remove all i2c read/write command.

- 2. dpi\_timing is the timing of the panel, which needs to be added in the panel driver. It corresponds to starfive\_drm\_seedpanel.c under the path of \linux\drivers\gpu \drm\verisilicon, which maps seeed\_panel\_modes.
  - a. Enter starfive\_drm\_seedpanel.c file.
  - b. Modify the parameters of PLL 1188M under this function according to the output file.

#### Figure 5-5 Modify the Parameters



#### i Tip:

- clock = pixelclock/1000
- hdisplay = hactive
- hsync\_start = hactive + hfp

- hsync\_end = hactive + hfp + hbp
  - htotal = hactive + hfp + hbp+ hsync\_len
  - vdisplay = vactive

i

- vsync\_start = vactive + vfp
- vsync\_end = vactive + vfp + vbp
- vtotal = vactive + vfp + vbp + vsync\_len
- c. After configuration, it will be synchronized to the DC controller driver and MIPI DSI driver. For example, to set the lanes to **1**, you can change the corresponding parameters in the input file.

```
Figure 5-6 Example Input

: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 pixelk_source

24 29700000 800 90 5 5 480 60 5 5 2 2 2 1188000000
```

The corresponding driver code is shown in the following figure:

- 3. Follow the steps below to configure MIPI DSI.
  - a. Open starfive\_drm\_dsi.c file under the path of \linux\drivers\gpu \drm\verisilicon in Linux.
  - b. Locate cdns\_dsi\_mode2cfg function to modify the value of hsa, hbp and hfp in channel 0 to 36, 108, 288.

#### Note:

Modifying parameters should based on the corresponding channel. 1C2L corresponds to channel 0 while 1C4L corresponds to channel 1.

Figure 5-8 hsa hbp hfp

```
if (output->dev->channel == 0) {//seeed
    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 == 1){//raxda 8 inch 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 == 2){//raxda 10 inch config
    dsi_cfg->hsa = 405-DSI_HSA_FRAME_OVERHEAD;
    dsi_cfg->hsa = 405-DSI_HSA_FRAME_OVERHEAD;
    dsi_cfg->hsa = 405-DSI_HSA_FRAME_OVERHEAD;
    dsi_cfg->hbp = 403-DSI_HBP_FRAME_OVERHEAD;
    dsi_cfg->hfp = 396-DSI_HFP_FRAME_OVERHEAD;
}
}
```

c. Then locate cdns\_dsi\_adjust\_phy\_config function to modify the **bitrate** to **750000000**.

Figure 5-9 Bitrate



*Tip:* The **hs\_clk\_rate** in the figure means **bitrate**.

## 5.2.3. Configuration for 1C4L

This section provides an example to configure the parameters in 1C4L MIPI driver. The following code block is an example of the output file:

```
dpi_timing = { //800x1280, 24 bits, 59.68 Hz
    .pixelclock
                 = 66000000,
    .hactive
                 = 800
    .hfront porch = 44
    .hback_porch = 5
    .hsync_len
                 = 5
    .vactive
                = 1280
    .vfront_porch = 5
    .vback porch = 5
    .vsync_len
                 = 5
},
```

```
.dsi_timing = {
   .dlanes = 4 ,
   .bitrate = 400000000,
   .hsa = 66 ,
   .hbp = 64 ,
   .hfp = 57 ,
   .hact = 2400 ,
},
```

1. For the 1C4L MIPI channel, to support a new screen, it is necessary to modify the driver file panel-jadard-jd9365da-h3.c under /Linux/drivers/gpu/drm/panel.

This driver file servers as an i2c device by default, and the probe interface will use the i2c command to read the panel ID, and verify whether the current screen is connected properly based on the return value. When debugging a new screen, there are two cases:

- If the i2c command can read the panel ID, the probe function needs to support the i2c interface; In seed\_ panel\_ enable function, the panel should be enabled according to the specific command of i2c.
- If the i2c command cannot read panel ID, you are recommand to remove the i2c related command, then the probe process can be completed, so that the screen is in connected status by default.



#### Note:

It is necessary to remove all i2c read/write command.

- 2. dpi\_timing is the timing of the panel, which needs to be added in the panel driver. It corresponds to cz101b4001\_desc under the path of \linux\drivers\gpu\drm \panel.
  - a. Enter starfive\_drm\_seedpanel.c file.
  - b. Modify the parameters under this function according to the output file.

#### Figure 5-10 Modify the Parameters

```
static const struct jadard_panel_desc cz101b4001_desc = {
      .mode = {
           .clock
                           = 79200,
           .hdisplay
                          = 800,
           .hsync_start = 800 + 139,
.hsync_end = 800 + 139 + 5,
.htotal = 800 + 139 + 5 + 5,
           .vdisplay = 1280,

    .vsync_start
    = 1280 + 84,

    .vsync_end
    = 1280 + 84 + 20,

    .vtotal
    = 1280 + 84+ 20 + 7,

           .width mm = 62,
           .height_mm = 110,
                          = DRM_MODE_TYPE_DRIVER | DRM_MODE_TYPE_PREFERRED,
           .type
      .
1anes = 4,
     .format = MIPI_DSI_FMT_RGB888,
     .init cmds = cz101b4001_init_cmds,
     .num_init_cmds = ARRAY_SIZE(cz101b4001_init_cmds),
     .timings = & jadard_timing,
     .num_timings = 1,
};
```

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
- c. After configuration, it will be synchronized to the DC controller driver and MIPI DSI driver. For example, to set the lanes to **4**, you can change the corresponding parameters in the input file. The corresponding driver code is shown in the following figure:

```
Figure 5-11 Driver Code
```

```
static int jadard_dsi_probe(struct mipi_dsi_devic
{
    struct device *dev = &dsi->dev;
    struct jadard *jadard = mipi_dsi_get_drvdata(dsi);
    int ret;
    dsi->mode_flags = MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_'
    dsi->format = MIPI_DSI_FMT_RGB888;
    dsi->lanes = 4;
    dsi->channel = 1;
    dsi->hs_rate = 490000000;
    ret = mipi_dsi_attach(dsi);
    if (ret < 0) {
        return ret;
    }
    return 0;
}
</pre>
```

3. Follow the steps below to configure MIPI DSI.

- a. Open starfive\_drm\_dsi.c file under the path of \linux\drivers\gpu \drm\verisilicon in Linux.
- b. Locate cdns\_dsi\_mode2cfg function to modify the value of hsa, hbp and hfp in channel 1 to 66, 64, 57.

```
Note:
```

Modifying parameters should based on the corresponding channel. 1C2L corresponds to channel 0 while 1C4L corresponds to channel 1.

```
Figure 5-12 hsa hbp hfp
```

```
if (output->dev->channel == 0) {//seeed
    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 == 1){//raxda 8 inch 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 == 2){//raxda 10 inch config
    dsi_cfg->hsa = 405-DSI_HSA_FRAME_OVERHEAD;
    dsi_cfg->hsa = 405-DSI_HSA_FRAME_OVERHEAD;
    dsi_cfg->hsp = 403-DSI_HBP_FRAME_OVERHEAD;
    dsi_cfg->hbp = 403-DSI_HBP_FRAME_OVERHEAD;
    dsi_cfg->hfp = 396-DSI_HFP_FRAME_OVERHEAD;
}
}
```

c. Then locate cdns\_dsi\_adjust\_phy\_config function to modify the **bitrate** to **400000000**.

```
Figure 5-13 Bitrate
```

```
if (output->dev->channel == 0) {
    phy cfg->hs clk rate = 75000000;//seeed
} else if (output->dev->channel == 1){
    phy_cfg->hs_clk_rate = 49000000;//8 inch
} else if (output->dev->channel == 2){
    phy_cfg->hs_clk_rate = 98000000;//10 inch
}

Tip:
```

The **hs\_clk\_rate** in the figure means **bitrate**.

# 6. Debug LCD

# 6.1. Test Case Configuration

Follow the steps below to enable the kernel configuration for MIPI LCD.

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 Target packages menu.

#### Figure 6-1 Target Packages

Buildnest JM/IIG_SID_SUK_V2.0.2 Configuration Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> selects a feature, while <n> excludes a feature. Press <esc><esc><to <?="" exit,=""> for Help,  for Search. Legend: [*] feature is selected [ ] feature is excluded</to></esc></esc></n></y></enter>
iarget options> iuild options> ioolchain> iystem configuration> iernel> filesystem images> iboolloaders> Hist utilities> Legacy config options>

3. Enter the Libraries menu.

#### **Figure 6-2 Libraries**

/home/shengyang.chen/202206_1/freelight-u-sdk/work/buildroot_initramfs/.config < Buildroot JH7110_510_5DK_v2.0.2 Configuration		
Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> selects a feature, while <n> excludes a feature. Press <esc><esc><esc><to <7="" exit,=""> for Help,  for Search. Legend: [+] feature is selected [] feature is excluded</to></esc></esc></esc></n></y></enter>		
<pre> uryGox (package-busybox/busybox.config) PusyBox configuration file to use? ()</pre>		

4. Enter the **Graphics** menu.

igure 6-3 Graphics		
/home/shongyang.chen/202206_1/freelight-u-sdk/work/buildroot_initramfs/.config - Buildroot JH7110_510_5DK_v2.0.2 Configuration		
Libraries Libraries Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> selects a feature, while <n> excludes a feature. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] feature is selected [] feature is excluded</esc></esc></n></y></enter>		
Audio/Sound> Compression and decompression> Crypto> Database> Filesystem> Hurdware handling> Bayascript> Bogging> Logging> Multimedia> Nitworking>		
Security> Text and terminal handling>		

#### 5. Enter the **libdrm** menu.

#### Figure 6-4 libdrm

Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus&gt;). Highlighted letters are hotkeys. Pressing <y> selects a feature, while <n> excludes a feature. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] feature is selected [ ] feature is excluded         [ ] essimp</esc></esc></n></y></enter>
[] ssimp *** at-spi2-atk depends on X.org *** *** at-spi2-core depends on X.org ***
<pre>cl sk w not depute an horg cl sk w not depute an operat cl sk w not depute cl sk w not depute an operat cl sk w not depute cl sk w n</pre>

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

#### Figure 6-5 Install Test Programs

/home/shengyang.chen/202206_1/freelight-u-sdk/work/buildroot_initramfs/.config - Buildroot JH7110_510_SOK_v2.0.2 Configuration > Target Daskages > Liberties > Checkings > Liberties > Lib	
Lindra Arrow keys navigate the menu. «Enter> selects submenus> (or empty submenus). Highlighted letters are hotkeys. Pressing <y> selects a feature, while <n> excludes a feature. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] feature is selected [] feature is excluded</esc></esc></n></y>	
libdrm [] radeon [] andgpu [] n.uveau [] etnaviv (experimental) [#] install test programs	

**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.

# 6.2. Before Debug

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

#### Figure 6-6 Start-up Logs

	12.190735]	of_cfs_init
Ē	12.193217]	of_cfs_init: OK
Ē	12.196524]	cfg80211: Loading compiled-in X.509 certificates for regulatory database
Ē	12.314845	cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
Ē	12.326364	starfive soc:display-subsystem: bound 29400000.dc8200 (ops 0xfffffff80e76720)
Ē	12.334695	innohdmi-starfive 29590000.hdmi: inno hdmi bind begin
Ē	12.341911	platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
Ē	12.350596	cfg80211: failed to load regulatory.db
Ē	12.351087	innohdmi-starfive 29590000.hdmi: [drm:inno hdmi bind] registered Inno HDMI I2C bus driver success
Ē	12.365560	innohdmi-starfive 29590000.hdmi: HDMI&AUDIO register done.
Ē	12.372175	innohdmi-starfive 29590000.hdmi: inno hdmi bind end
Ē	12.378113	starfive soc:display-subsystem: bound 29590000.hdmi (ops 0xffffffff80e774d0)
Ē	12.386232	vs-simple-encoder soc:rdb-output: encoder bind begin
Ē	12.392340	no panel517
Ē	12.395057	vs-simple-encoder soc:rab-output: encoder bind error
Ē	12.401086	starfive soc:display-subsystem: bound soc:rgb-output (ops 0xfffffff80e77118)
È	12,409298	vs-simple-encoder_soc:dsi-output: encoder_bind_begin
ſ	12.415383]	cdns-dsi 295d0000.mipi: ===>cdns_dsi_bridge_attach begin
I	12.421758	cdns-dsi 295d0000.mipi: ===>cdns_dsi_bridge_attach end
ľ	12.427968	vs-simple-encoder soc:dsi-output: encoder_bind end
Ē	12.433828]	starfive soc:display-subsystem: bound soc:dsi-output (ops 0xfffffff80e77118)
Ē	12.442874]	[drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1
Ē	14.488355]	ALSA device list:
Ē	14.491337]	#0: Starfive-HDMI-Sound-Card
Ē	14.498788]	Freeing unused kernel image (initmem) memory: 2196K
Ē	14.504844]	Run /init as init process
Ē	14.508536]	with arguments:
Ē	14.511509]	/init <sup>®</sup>
E	14.513785]	with environment:
E	14.516934]	HOME=/
E	14.519311]	TERM=linux
S	starting syslo	gd: ок
S	tarting klogd	: OK
R	unning syscil	: OK
S	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.

# 6.3. Debug LCD

1. Follow the steps in <u>Test Case Configuration (on page 32)</u> to configure the test environment.



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.

• Debug output 1:

Figure 6-7 Debug Display 1
# modetest -M starfive
id crtc type possible crtcs possible clones 115 0 TMDS 0x00000001 0x00000001 117 0 DSI 0x00000002 0x0000002
Connectors: id encoder status 116 0 modes: A A A A A A A A A A A A A
<pre>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 50.94 1920 2008 2052 2200 1080 1084 1089 1125 148532 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 50.00 1280 1390 1430 1650 720 725 730 750 74250 flags: phsync, pvsync; type: driver #5 1280x720 50.00 1280 1720 1760 1980 720 725 730 750 74250 flags: phsync, pvsync; type: driver #5 1280x720 48.00 1280 2240 2280 2500 720 725 730 750 94250 flags: phsync, pvsync; type: driver #6 1280x720 48.00 1280 2240 2280 2500 720 725 730 750 99910 flags: phsync, pvsync; type: driver #8 640x480 60.00 640 656 752 800 480 490 492 525 25200 flags: nhsync, nvsync; type: driver #8 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver</pre>
1 EDID: flags: immutable blob blobs:
<pre>value:</pre>
flags: enum enums: On=O Standby=1 Suspend=2 Off=3 value: O 5 link-status: flags: enum
enums: Good=0 Bad=1 value: 0 6 non-desktop: flags: immutable range
values: 0 1

#### Table 6-1 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 <i>Extended Display Identification Data (EDID)</i> of the screen

• Debug output 2:

Figure 6-8 Debug Display 2



Table 6-2 Debug Display 2		
Legend	Label	Description
1	id	The CRTC 0x00000001 mentioned in row ① of ta- ble Table 6-1 : Debug Display 1 (on page 35), which means the CRTC is available for use.
2	id	The CRTC 0x0000002 mentioned in row (1) of table Table 6-1 : Debug Display 1 (on page 35), 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:

Planes: id crtc fb CRTC x,y x,y gamma size possible crtcs
39 0 0 0,0 0,0 0 0 0 0 0 0 0 0 0 0 0 0 0
props:
s type: flags: immutable enum
enums: Overlay=0 Primary=1 Cursor=2 value: 1
30 IN_FORMATS: Flags: immutable blob
blobs:
value:
010000000000000000000000000000000000000
52583132425831324152313241423132 52413132424313125852313558423135
525831354258313542131354123135 524313542413155247313552473136
58523234584232345258323442583234 45532344142323452513234
412232414235305241233062412530 4152330614223305241333042413330
59553525553555555955559 59563132595531324e5631324e563231
4e5631364e5635315030313000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
in_formats blob decoded:
41 DEGAMMA_NODE:
enum: disabled=0 preset degamma for BT709=1 preset degamma for BT2020=2
42 rotation:
flags: bitmask values: rotate-0=0x1 rotate-90=0x2 rotate-180=0x4 rotate-270=0x8 reflect-x=0x10 reflect-v=0x20
value: 1 43 nixel bland mode
flags: enum
Contract of the interpreted coverage-1
44 atpirat Flags: range
values: 0 65535 value: 65535
45 COLOR_ENCODING: flags: enum
enums: TIU-R BT.709 YCbCr=1 ITU-R BT.2020 YCbCr=2 value: 0

#### Table 6-3 Debug Display 3

Legend	Description	
1	The CRTC and its connected plane	

Check MIPI connect status:

```
Figure 6-10 MIPI Connect Status
   modetest -M starfive -c
  480.003234] PVR_K: 325: modetest connected - (devID = 0)
  480.008882] PVR_K: 325: modetest disconnected - (devID =
  100.008882]
                                                                                                é 0)
Connectors:
            encoder status
0 disconr
                                                                             size (mm)
                                                                                                       modes
                                                   name
                                                                                                                    encoders
116
                         disconnected HDMI-A-1
                                                                             0x0
                                                                                                                    115
   props:
            1 EDID:
                         flags: immutable blob
blobs:
                         value:
            2 DPMS:
            2 DPMS:
    flags: enum
    enums: On=0 Standby=1 Suspend=2 Off=3
    value: 0
5 link-status:
    flags: enum
    flags: cood=0 Bad=1
                         enums: Good=0 Bad=1
value: 0
            6 non-desktop:
flags: immutable range
values: 0 1
value: 0
             4 TILE:
                         flags: immutable blob
blobs:
                         connected
                                                  DSI-1
                                                                             154x86
     odes:
   modes:
index name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot
#0 800x480 60.00 800 890 895 900 480 540 545 550 29700 flags: ; type: preferred, driver
            1 EDID:
                         flags: immutable blob
blobs:
                         value:
             2 DPMS:
                         flags: enum
                         enums: On=0 Standby=1 Suspend=2 Off=3 value: 0
            5 link-status:
flags: enum
            riags: enum
enums: Good=0 Bad=1
value: 0
6 non-desktop:
flags: immutable range
values: 0 1
value: 0
4 TILE:
            4 TTLE:
                         flags: immutable blob
blobs:
                         value:
#
```

The common connector ID of MIPI is 118. If both rgb2hdmi and MIPI panel are registered, the connector ID of MIPI will be assigned as 120.

# 6.4. Test Example

#### For LCD Output

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

modetest -M starfive -D 0 -a -s 118@35:800x480 -P 74@35:800x480@RG16 -Ftiles

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

- 118@35:800x480 <Connector ID>@<CRTC ID>: <Resolution>
- 74@35:800x480@RG16 <Plane ID>@<CRTC ID>: <Resolution>@<Format>

#### **Output Result**

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

Figure 6-11 Test Example

