

# JH7110 SDK Developer Guide for Display Controller

VisionFive 2 Version: 1.1 Date: 2023/8/11 Doc ID: JH7110-DGEN-014

## **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: sales@starfivetech.com
- Support: <a href="mailto:support@starfivetech.com">support@starfivetech.com</a>

## Contents

List of Tables4
List of Figures
Legal Statementsii
Prefacevi
1. Introduction7
1.1. Function Introduction
1.2. Block Diagram
1.3. DC8200 Display Controller
1.4. Video Output Driver Framework
1.5. Device Tree Overview
1.6. Source Code Structure
2. Configuration
2.1. Device Tree Configuration
2.2. Kernel Menu Configuration
2.3. Driver Configuration
3. Debug Method20
3.1. Test Case Configuration
3.2. Before Debug
3.3. Debug Display 22
3.4. Test Example

## List of Tables

Table 0-1 Revision History	vi
Table 1-1 Display Subsystem Data Mapping	8
Table 3-1 Start-up Logs	
Table 3-2 Debug Display 1	23
Table 3-3 Debug Display 2	
Table 3-4 Debug Display 3	

## List of Figures

Figure 1-1 Display Subsystem Block Diagram	7
Figure 1-2 DC8200 Display Controller Block Diagram	
Figure 1-3 Driver Framework	9
Figure 1-4 Device Tree Workflow	
Figure 2-1 Device Drivers	14
Figure 2-2 Graphics Support	
Figure 2-3 HDMI2.0	16
Figure 2-4 MIPI DSI Select	
Figure 2-5 Direct Rendering Manager	
Figure 2-6 I2C Encoder or Other Helper	18
Figure 2-7 NXP Semiconductors TDA998X	
Figure 3-1 Target Packages	20
Figure 3-2 Libraries	
Figure 3-3 Graphics	
Figure 3-4 libdrm	21
Figure 3-5 Install Test Programs	21
Figure 3-6 Start-up Logs	22
Figure 3-7 Debug Display 1	23
Figure 3-8 Debug Display 2	24
Figure 3-9 Debug Display 3	25
Figure 3-10 Test Example	27

## 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 display module of the StarFive next generation SoC platform - JH7110.

#### Audience

This document mainly serves the display module relevant driver developers. If you are developing 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	2023/8/11	Corrected the wrong command in <u>Test Case Configuration (on</u> <u>page 20)</u> Step 1.
1.0		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

The display subsystem, named as **dom\_vout\_top** in the JH7110 system, includes front-end video data capture, display controller and display interface, such as RGB IF, HDMI, and MIPI.

In the display subsystem, DC8200 display controller works as a third party high-performance optimized-area *Display Processor Unit (DPU)* IP that can be used for reading rendered images from the frame buffer to the display.

See <u>Block Diagram (on page 7)</u> for more information.

## **1.1. Function Introduction**

The display controller supports the following features:

- Support 2 display output panels
- Support 1 video/graphic layer per output panel
- Support 2 overlay layers per output panel
- Support 6 video/graphic and overlay layers
- Support 2 cursor layers
- Programmable cursor sizes: 32 × 32, 64 × 64
- Dynamic layer allocation support for video/graphic and overlay layers
- Support 2-screen display
- Support output interfaces: DP (RGB, YUV), DPI (RGB)

### 1.2. Block Diagram

The block diagram of the display subsystem is displayed in following diagram.

#### Figure 1-1 Display Subsystem Block Diagram



#### Data Mapping

The DSI transmitter's pixel data could be from panel 0 or panel 1 interface of DC8200, and could be selected from DP or DPI interface. The RGB PAD and HDMI have similar mechanism.

Destination	Supported Data Mapping	Comment
DPI to PAD	<ul> <li>DP0/DP1 or DPI0/1 is used, default DPI is used.</li> <li>RGB24, RGB666 (CFG1), RGB565 (CFG1) when DPI is used.</li> </ul>	For flexibility
DSI Tx Data from DC8200	<ul> <li>Both DPI and DP are supported.</li> <li>YUV420 8-bit only (CFG3).</li> <li>YUV422 8-bit only (CFG1).</li> </ul>	Default DPI
HDMI Data from DC8200	<ul> <li>Both DP0 and DP1 are used for RGB and YUV.</li> <li>YUV444 and YUV422 8-bit/10-bit (CFG1).</li> <li>YUV420 8-bit/10-bit (CFG3).</li> </ul>	DP by default, and DPI for back- up

Table	1-1	Display	Subsystem	Data	Mapping

## 1.3. DC8200 Display Controller

The following image shows the block diagram of the DC8200 display controller.





The following components are included.

- Host Interface: Allows communication with the system and the DC controller. The host interfaces include the AXI, AHB, and APB. In this block, data crosses clock domain boundaries.
- Memory Control: Contains the AXI interface to manage the access between the system memory and layers of the DC8200.
- Write Back: For debug use only.
- Layers: Include video/graphic, overlay, and cursor layers.

- Video/graphic layers support both video and graphic configurations. Video/graphic and overlay layers support dynamic layer allocation and de-gamma.
- Overlay 1 and overlay 3 do not support scaling, rotation, and line buffers.
- Cursor layers provide hardware cursor functionality.
- Dither: Provides a Lookup Table (LUT).
- Gamma: Performs gamma correction.
- Output Panels: Support two output panels, as shown by Display0 and Display1.
- **Output Interfaces**: Support parallel pixel output with 30-bit Data, Horizontal Sync, Vertical Sync, and Data Enable. Support easy adaptation to external serialization logic, for example, HDMI.
- **Pixel Pipelines**: Reside in the layers and output panels. Two display pipelines support linear and tiled frame buffers for RGB and YUV inputs. Optional enhancements include multiple overlay layers, composition and blending, up/down scaling with multi-tap filtering, and color space conversions.

If you need more information, you may contact StarFive technical support and request documentation from the third-party IP.

### **1.4. Video Output Driver Framework**

The following figure shows the framework of the video output driver and the display controller.

#### Figure 1-3 Driver Framework



The video output driver framework has the following 3 layers.

- Application layer consists of application code and test code and communicate with kernel layer through libdrm.
- Kernel layer consists of DRM core and Vout driver. DRM core receives commands from libdrm and transfer to Vout driver.
- Hardware layer is connected with Vout driver, and it operates the hardware directly.

### **1.5. Device Tree Overview**

Since Linux 3.x, device tree is introduced as a data structure and language to describe hardware configuration. It is a systemreadable description of hardware settings so that the operating system doesn't have to hard code details of the machine. A device tree is primarily represented in the following forms.

- Device Tree Compiler (DTC): The tool used to compile device tree into system-readable binaries.
- Device Tree Source (DTS): The human-readable device tree description file. You can locate the target parameters and modify hardware configuration in this file.
- *Device Tree Source Information (DTSI)*: The human-readable header file which you can include in device tree description. You can locate the target parameters and modify hardware configuration in this file.
- Device Tree Blob (DTB): The system-readable device tree binary blob files which is burned in system for execution.

The following diagram shows the relationship (workflow) of the above forms.

#### Figure 1-4 Device Tree Workflow



## **1.6. Source Code Structure**

Locate the JH7110 Software Development Kit (SDK) with the following information.

- Repository: https://github.com/starfive-tech/VisionFive2
- Branch: JH7110\_VisionFive2\_devel
- Tag: Select the newest tag. For example, VF2\_v2.11.5 is newer than VF2\_v2.10.10.

The following code block shows the source code structure of the display controller.

linux	-5.15.	0		
L dr	ivers			
L   ·	gpu			
			drm	
			ve	erisilicon
				vs_dc.c
				vs_dc.h
				vs_dc_hw.c
				vs_dc_hw.h
				vs_drv.c
				vs_drv.h
				vs_crtc.c
				vs_crtc.h
				vs_plane.c
				vs_plane.h
				vs_simple_enc.c
				vs_simple_enc.h
				vs_gem.c

				vs_gem.h
				vs_virtual.c
				vs_virtual.h
				vs_dc_dec.c
				vs_dc_dec.h

## 2. Configuration

### 2.1. Device Tree Configuration

#### **Overview**

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

The device tree of JH7110 is stored in the following path:

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

The following code block shows the DTS file structure for JH7110.

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

#### **Display Subsystem**

In the file jh7110.dtsi, you can find the device tree configuration of the display subsystem as the following code block:

```
display: display-subsystem {
    compatible = "startive,jh7110-display","verisilicon,display-subsystem";
    ports = <&dc_out_dpi0>;
    status = "disabled";
dssctrl: dssctrl@295B0000 {
    compatible = "starfive,jh7110-dssctrl","verisilicon,dss-ctrl", "syscon";
    reg = <0 0X295B0000 0 0x90>;
};
```

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

- compatible: Compatibility information, used to associate the display controller and its target device.
- ports: The port(s) used by the display controller.
- status: The work status of the display controller module. To enable the module, set this bit as "okay" or to disable the module, set this bit as "disabled".
- reg: Register base address "0x295B0000" and range "0x90".

#### DC8200

In the file jh7110.dtsi, you can find the device tree configuration of DC8200 (host) as the following code block:

```
dc8200: dc8200@29400000 {
    compatible = "verisilicon,dc8200";
    verisilicon,dss-syscon = <&dssctrl>;
    reg = <0x0 0x29400000 0x0 0x100>,
        <0x0 0x29400800 0x0 0x2000>,
        <0x0 0x17030000 0x0 0x1000>;
    interrupts = <95>;
    status = "disabled";
    clocks = <&clkgen JH7110_NOC_BUS_CLK_CPU_AXI>,
        <&clkgen JH7110_NOC_BUS_CLK_GPU_AXI>,
        <&clkgen JH7110_NOC_BUS_CLK_VDEC_AXI>,
        <&clkgen JH7110_NOC_BUS_CLK_VENC_AXI>,
        <&clkgen JH7110_NOC_BUS_C
```

```
<&clkgen JH7110_NOC_BUS_CLK_ISP_AXI>,
                 <&clkgen JH7110_NOC_BUS_CLK_STG_AXI>,
                 <&clkgen JH7110_VOUT_SRC>,
                 <&clkgen JH7110_VOUT_TOP_CLK_VOUT_AXI>,
                 <&clkgen JH7110_AHB1>,
                 <&clkgen JH7110_VOUT_TOP_CLK_VOUT_AHB>,
                 <&clkgen JH7110_VOUT_TOP_CLK_HDMITX0_MCLK>,
                 <&clkgen JH7110_I2STX_4CH0_BCLK_MST>,
                 <&clkvout JH7110_U0_DC8200_CLK_PIX0>,
                 <&clkvout JH7110_U0_DC8200_CLK_PIX1>,
                 <&clkvout JH7110_U0_DC8200_CLK_AXI>,
                 <&clkvout JH7110_U0_DC8200_CLK_CORE>,
                 <&clkvout JH7110_U0_DC8200_CLK_AHB>,
                 <&clkgen JH7110_VOUT_TOP_CLK_VOUT_AXI>,
                 <&clkvout JH7110_DOM_VOUT_TOP_LCD_CLK>,
                 <&hdmitx0_pixelclk>,
                 <&clkvout JH7110_DC8200_PIX0>,
                 <&clkvout JH7110 U0 DC8200 CLK PIX0 OUT>,
                 <&clkvout JH7110_U0_DC8200_CLK_PIX1_OUT>;
        clock-names = "noc_cpu", "noc_cfg0", "noc_gpu", "noc_vdec", "noc_venc",
                      "noc_disp","noc_isp","noc_stg","vout_src",
                       "top_vout_axi","ahb1","top_vout_ahb",
                       "top_vout_hdmiTX0","i2stx","pix_clk","vout_pix1",
                       "axi_clk","core_clk","vout_ahb",
                       "vout_top_axi", "vout_top_lcd", "hdmitx0_pixelclk", "dc8200_pix0",
                      "dc8200_pix0_out","dc8200_pix1_out";
        resets = <&rstgen RSTN_U0_DOM_VOUT_TOP_SRC>,
                 <&rstgen RSTN_U0_DC8200_AXI>,
                 <&rstgen RSTN_U0_DC8200_AHB>,
                 <&rstgen RSTN_U0_DC8200_CORE>,
                 <&rstgen RSTN U0 NOC BUS CPU AXI N>.
                 <&rstgen RSTN_U0_NOC_BUS_AXICFG0_AXI_N>,
                 <&rstgen RSTN_U0_NOC_BUS_APB_BUS_N>,
                 <&rstgen RSTN_U0_NOC_BUS_DISP_AXI_N>,
                 <&rstgen RSTN_U0_NOC_BUS_STG_AXI_N>;
        reset-names = "rst_vout_src", "rst_axi", "rst_ahb", "rst_core",
                      "rst_noc_cpu","rst_noc_axicfg0","rst_noc_apb",
                      "rst_noc_disp","rst_noc_stg";
        power-domains = <&pwrc JH7110_PD_VOUT>;
};
```

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

- compatible: Compatibility information, used to associate the display controller and its target device.
- dss-syscon: The SYSCON register(s) of the display panel.
- reg: Register base address "0x29400000" and range "0x100".
- interrupts: Hardware interrupt ID.
- status: The work status of the display controller module. To enable the module, set this bit as "okay" or to disable the module, set this bit as "disabled".
- clocks: The clocks used by the display controller module.
- clock-names: The names of the above clocks.
- resets: The reset signals used by the display controller module.
- reset-names: The names of the above reset signals.
- power-domains: The power supply domain of the display controller module.

In the file jh7110-common.dtsi, you can find the device tree configuration of DC8200 (endpoint) as the following code block:

```
&dc8200 {
  status = "okay";
  dc_out: port {
    #address-cells = <1>;
    #size-cells = <0>;
}
```

#### | 2 - Configuration

```
dc_out_dpi0: endpoint@0 {
  reg = <0>;
  remote-endpoint = <&hdmi_input0>;
 };
 dc_out_dpi1: endpoint@1 {
  reg = <1>;
  remote-endpoint = <&hdmi_in_lcdc>;
 };
 dc_out_dpi2: endpoint@2 {
  reg = <2>;
  remote-endpoint = <&mipi_in>;
 };
 };
};
```

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

- reg: The register ID of the endpoint device.
- remote-endpoint: The endpoint device type of the display controller output.

### 2.2. Kernel Menu Configuration

Follow the steps below to enable the kernel configuration for display controller.

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 2-1 Device Drivers

e Edit View Search Terminal Help
config - Linux/riscv 5.15.0 Kernel Configuration
<pre>e Edit View Sarch Terminal Help config - Linux/riscy 5.15.0 Kernel Configuration Arrow keys navigate the menu. <enter> selects submenus&gt; (or enpty submenus&gt;). Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <n> modularizes features. Press <esc><esc> to exit, <?> for Help,  for Search. Legend: [*] built-in [] excluded <h> module &lt;&gt; module capable Ceneral setup&gt; (*] MMU-based Paged Memory Management Support SoC selection&gt; Platform type&gt; Nemeral features&gt; Boot options&gt; Power Management options&gt; [*] Inable loadele module support&gt; [*] Inable loadele module support&gt; It look charger&gt; It look the block layer&gt; It look the block layer&gt; It look Stedulers&gt; Lecurity options&gt; Security options&gt; Lecurity options&gt; Kernel hacking&gt; Kernel hacking&gt;</h></esc></esc></n></n></y></enter></pre>
<pre>eSelect&gt; &lt; Exit &gt; &lt; Help &gt; &lt; Save &gt; &lt; Load &gt;</pre>

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

#### Figure 2-2 Graphics Support

Device Drivers
Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus&gt;). Highlighted letters are hotkeys. Pressing &lt;&gt;&gt; includes, <n></n></enter>
escudes, modularizes realures. Press <esc> to exit, for Help, for Search. Legend: [^] built-in [] excluded module &lt;&gt; module</esc>
Capable
^( - )
< > RapidIO support
Generic Driver Options>
Bus devices>
<pre>&lt; &gt; Connector - unified userspace &lt;-&gt; kernelspace linker</pre>
Firmware Drivers>
<> When the part of the par
* Device Tree and Open Firmware support
<pre>&lt; &gt; Parallel port support</pre>
[*] Block devices>
NVME Support>
Misc devices>
SCSI device support>
<pre>&lt;*&gt; Serial ATA and Parallel ATA drivers (lubata)&gt; </pre>
[ ] Multiple devices ariver support (KALD and LUM)
< > Generic Large Core Mod (10M) and Configes Inflastructure
TEFE 1394 (EireWire) support>
[*] Network device support>
Input device support>
Character devices>
I2C support>
<> I3C support
[*] PI support>
< > PMI support
<> PPS support
PTP clock support>
[*] Pin controllers>
-*- CPIO Support>
< > Dallas's 1-wire support
[*] Board Level reset or power off>
[] Power Supply Class Support
[] Thermal drivers
[*] Watchdog Timer Support>
Sonics Šilicon Backplane support
< > Broadcom specific AMBA
Multifunction device drivers>
[*] Voltage and Current Regulator Support>
< > Memote Controller support
() Hitimedia support
<pre></pre>
HID support>
L 1(+)
Sectories < Exit > < Hetp > < Save > < Load >

- 4. Continue your settings per the sections below depending on your target output devices.
  - For HDMI output (on page 15)
  - For MIPI output (on page 16)
  - For RGB2HDMI output (on page 17)

#### For HDMI Output

Continue your settings with the following steps to enable the kernel configuration for HDMI output.

#### | 2 - Configuration

1. In the Graphics support menu, select the HDMI2.0 option.

#### Figure 2-3 HDMI2.0

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

#### For MIPI Output

Continue your settings with the following steps to enable the kernel configuration for MIPI output.

1. In the Graphics support menu, select the Starfive MIPI DSI Select option.

#### Figure 2-4 MIPI DSI Select

Graphics 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 <a> module &lt;&gt; module capable</a></esc></esc></m></n></y></enter>
<pre>(*) (A Arkitration (*) (A Arkitration (*) (A Arkitration *) (F Arkitration *) (</pre>

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

#### For RGB2HDMI Output

Continue your settings with the following steps to enable the kernel configuration for RGB2HDMI output.

1. In the Graphics support menu, select and enter the Direct Rendering Manager menu.

#### Figure 2-5 Direct Rendering Manager

2. In the Direct Rendering Manager menu, select and enter the I2C encoder or helper chips menu.

#### Figure 2-6 I2C Encoder or Other Helper

Device Drivers > Graphics support > Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)
Cirect Rendering Manager (XFreeB8 4.1.0 and higher DKI support) 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>
Diract Rendering Nanager (KFreeB6 4.1.0 and higher DRI support) [] FM DW XX Interfac [] insert extra checks and debug info into the DRM range managers < iself tests for DRM [] trable refcount backtrace history in the DP MST helpers [*] trable legacy fides support for your modesetting driver [100] (verallocation of the fidev physical address (DANGEROUS) [] chamelessiy allow leaking of fidev physical address (DANGEROUS) [] chamelessiy allow leaking of rotes set instead of probing for it [] chambel eigplayPort CecTunneling-over-AUX HOMI support 22c encoder or helper chips
Elect < Exit > < Help > < Save > < Load >

3. In the I2C encoder or helper chips menu, select the NXP Semiconductors TDA998X HDMI encoder option.

Figure 2-7 NXP Semiconductors TDA998X

120 encoder or higher china 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, &gt; for Search. Legend: [*] built-in [] excluded <a> module &lt;&gt; module &lt;&gt; module capable</a></esc></esc></m></n></y></enter>		
<pre>reatures. Press escretes to exit, <pre><pre></pre> tor Heip, <pre><pre></pre> to throatel chr00e TV encoder <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>		
Selects < Exit > < Help > < Save > < Load >		

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

## 2.3. Driver Configuration

The following code block shows the driver configuration.

CONFIG\_DRM\_VERISILICON=y

## 3. Debug Method

### 3.1. 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 3-1 Target Packages

Buildreet 307110_510_50K_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> to exit, <? > for Help,  for Search. Legend: [*] feature is selected [] feature is excluded</esc></n></y></enter>
Target options> Hulld options> ioolchain> isystem configuration> ierael> Tooption podepoise Tooption of the> Hest utilities> Legacy config options>

3. Enter the Libraries menu.

#### **Figure 3-2 Libraries**

Tarrot narkans-		
Target packages  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><to <?="" exit,=""> for Help, &gt; for Search. Legend: [*] feature is selected [] feature is excluded</to></esc></n></y></enter>		
<pre>-** !usyBox (package/busybox/comfig) !usyBox configuration file to use? } / dditional BusyBox configuration fragment files * show packages that are also provided by busybox i not vidual binaries down startur script udition and tides applications&gt; compressors and decompressors&gt; leebugging, profiling and benchmark&gt; neteugging, profiling and benchmark&gt; ineteugging, profiling and benchmark&gt; rebugging, profiling and benchmark&gt; reputer languages&gt; reputer languages and scripting&gt; hitranese&gt; hit torainese&gt; hit torainese&gt; recurring</pre>		

4. Enter the Graphics menu.

Figure 3-3 Graphics		
Intrace packages > Cubraries     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> Hirdware handing> Javascript> Javascript> Logging> Multimedia> Nitworking> Security> Security> Text and terminal handling>		

5. Enter the **libdrm** menu.

#### Figure 3-4 libdrm

Taraet backages > Libraries - Grabits -
Graphics 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> to exit, <? > for Help,  for Search. Legend: [*] feature is selected [] feature is excluded</esc></n></y></enter>
<pre>[] :ssimp  *** at-spi2-atk depends on X.org ***  *** at-spi2-ore depends on X.org ***</pre>
[] gdk-pixbuf [] gdflib
*** granite needs libgtk3 and a toolchain w/ wchar, threads ***
*** gtKmm3 needs libgtk3 and a toolchain w/ C++, wchar, threads, gcc >= 4.9 ***
-*- hirtbuzz [] ijs
[ ] mlib2 **** irrlicht needs X11 and an OpenGL provider ***
[] asper [] bioZder
peg support
peg varlant (jpeg)> []ims+
[] i cms2 [] iensfun
[] leptonica [] libart
[ ] Libdetx
Li thepoxy
**** Libém needs X.org and a toolchain w/ wchar, threads, C++, gcc >= 4.8 ***
[ ] lubtm-extra **** Libtreeglut depends on X.org and needs an OpenGL backend ***
[] libfreeimage [] libbeviff
**** libglew depends on X.org and needs an OpenGL backend *** *** libglfw depends on X.org and needs an OpenGL backend *** *** libglw needs on OpenGL backend ***
[] togta
L J UbgK3 [] UbgK3
[] Libang -* Libong
[]] ibgrencode

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

#### Figure 3-5 Install Test Programs

arget packages > Libraries > Graphics > libdrm —
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 [] :adeon [] :andgpu [] andgpu [] nouveau [] etnaviu (experimental) []*] = notali 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.

## **3.2.** Before Debug

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

#### Figure 3-6 Start-up Logs

<pre>6.8716201 cfg80211: failed to load regulatory.db 6.871962 innohdmi-starfive 29590000.hdmi: [drm:inno_hdmi_bind] registered Inno HDMI I2C bus driver success 6.886475 starfive soc:display-subsystem: bound 29590000.hdmi (ops 0xfffffff80e74af0) 6.894379] vs-simple-encoder soc:rgb-output: encoder_bind begin 6.906456 starfive soc:display-subsystem: bound soc:rgb-output (ops 0xfffffff80e74738) 6.914641 vs-simple-encoder soc:rgb-output: encoder_bind begin 6.927091 cdns-dsi 295d0000.mipi: ===&gt;cdns_dsi_bridge_attach begin 6.927091 cdns-dsi 295d0000.mipi: ==&gt;cdns_dsi_bridge_attach end 6.93294 vs-simple-encoder soc:dsi-output: encoder_bind end 6.93294 vs-simple-encoder soc:dsi-output (ops 0xfffffff80e74738) 6.947967 [drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1. 7.287268 mmc_host mmcO: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.977276 ALSA device list: 8.980246 No soundcards found. 8.98585 Freeing unused kernel image (initmem) memory: 2168K 8.991229 Run /init as init process 8.995322 with arguments: 8.998368 / init 9.000640 with environment: 9.003781 HOME=/ 9.006139 TERM=linux Starting Syslogd: OK Running syscl1: OK Running syscl1: OK</pre>		6.827275] 6.827542] 6.840870] 6.845226] 6.849161] 6.853766] 6.862977]	<pre>mmc_host mmc0: Bus speed (slot 0) = 198000000Hz (slot req 200000Hz, actual 200000Hz div = 495) DC_CURSOR_FOREGROUND + 0 = 0 DC_CURSOR_FOREGROUND + 0 = aaaaaa DC_CURSOR_FOREGROUND + 1 = 0 DC_CURSOR_FOREGROUND + 1 = aaaaaa starfive soc:display-subsystem: bound 29400000.dc8200 (ops 0xfffffff80e73c88) platform regulatory.0: Direct firmware load for regulatory.db failed with error -2</pre>
<pre>[ 6.871962] innohdmi-starfive 29590000.hdmi: [drm:inno_hdmi_bind] registered Inno HOWI I2C bus driver success [ 6.886475] starfive soc:display-subsystem: bound 29590000.hdmi (ops 0xffffff80e74af0) [ 6.894579] vs-simple-encoder soc:rgb-output: encoder_bind begin [ 6.900619] vs-simple-encoder soc:rgb-output: encoder_bind begin [ 6.914641] Vs-simple-encoder soc:dsi-output: encoder_bind begin [ 6.927091] cdns-dsi 295d0000.mipi: ===&gt;cdns_dsi_bridge_attach begin [ 6.927091] cdns-dsi 295d0000.mipi: ===&gt;cdns_dsi_bridge_attach begin [ 6.933294] vs-simple-encoder soc:dsi-output: encoder_bind end [ 6.937091] cdns-dsi 295d0000.mipi: ===&gt;cdns_dsi_bridge_attach end [ 6.933294] vs-simple-encoder soc:dsi-output: encoder_bind end [ 6.937087] [drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1.4 [ 7.287268 mmc_host mmcO: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) [ 8.97276] ALSA device list: [ 8.980246] No soundcards found. [ 8.985685 Freeing unused kernel image (initmem) memory: 2168K [ 8.991729] Run /init as init process [ 8.993302] with arguments: [ 9.003781] HOME=/ [ 9.006139] TERM=linux Starting syslogd: OK Running sysctl: OK Running sysctl: OK </pre>	Ē	6.871620	cfg80211: failed to load regulatory.db
<pre>[ 6.886475] starfive soc:display-subsystem: bound 29590000.hdmi (ops 0xfffffff80e74af0) [ 6.886475] vs-simple-encoder soc:rgb-output: encoder_bind begin [ 6.900619] vs-simple-encoder soc:rgb-output: encoder_bind end [ 6.900619] vs-simple-encoder soc:display-subsystem: bound soc:rgb-output (ops 0xfffffff80e74738) [ 2</pre>	ſ	6.871962]	innohdmi-starfive 29590000.hdmi: [drm:inno_hdmi_bind] registered Inno HDMI I2C bus driver success 者
<pre>6.894579] vs-simple-encoder soc:rgb-output: encoder_bind begin 6.900619] vs-simple-encoder soc:rgb-output: encoder_bind begin 6.9006456 starfive soc:display-subsystem: bound soc:rgb-output (ops 0xfffffff80e74738) 6.914641 vs-simple-encoder soc:dsi-output: encoder_bind begin 6.9207391 cdns-dsi 295d0000.mip1: ===&gt;cdns_dsi_bridge_attach begin 6.932941 vs-simple-encoder soc:dsi-output: encoder_bind end 6.9332941 vs-simple-encoder soc:dsi-output: encoder_bind end 6.9391400 starfive soc:display-subsystem: bound soc:rdisplay-subsystem on minor 1. 6.947967 [drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1. 7.287268 mmc_host mmC0: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.977276] ALSA device list: 8.980246] No soundcards found. 8.985858 Freeing unused kernel image (initmem) memory: 2168K 8.995392] with arguments: [ 8.998388] /init [ 9.0003781] HOME=/ 9.0003781] HOME=/ 9.</pre>	Ē	6.886475]	starfive soc:display-subsystem: bound 29590000.hdmi (ops 0xfffffff80e74af0) 🛛 🦊
<pre>6.900619] vs-simple-encoder soc:rgb-output: encoder_bind end 6.906456 starfive soc:display-subsystem: bound soc:rqb-output (ops 0xfffffff80e74738) 6.914641 vs-simple-encoder soc:ds1-output: encoder_bind begin 6.920739] cdns-dsi 295d0000.mipi: ==&gt;cdns_ds1_bridge_attach begin 6.933294 vs-simple-encoder soc:ds1-output: encoder_bind end 6.933294 vs-simple-encoder soc:ds1-output: encoder_bind end 6.937840 starfive soc:display-subsystem: bound soc:display-subsystem on minor 1.4 6.9479677 [drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1.4 7.287268 mmc_host mmc0: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.97276 ALSA device list: 8.9802461 No soundcards found. 8.991729] Run /init as init process 8.991729] Run /init as init process 8.991729] Run /init as init process 8.9938081 /init 9.0006401 with environment: 9.0037811 HOME=/ 9.006139] TERM=linux Starting syslogd: OK Running sysctl: OK</pre>	L	6.894579	vs-simple-encoder soc:rgb-output: encoder_bind begin
<pre>6.906456 starfive soc:display-subsystem: bound soc:rgb-output (ops 0xfffffff80e74738) 6.914641 vs-simple-encoder soc:dsi-output: encoder_bind begin 6.920739] cdns-dsi 295d0000.mipi: ===&gt;cdns_dsi_bridge_attach begin 6.932941 vs-simple-encoder soc:dsi-output: encoder_bind end 6.9332941 vs-simple-encoder soc:dsi-output: encoder_bind end 6.9312941 vs-simple-encoder soc:dsi-output: encoder_bind end 6.931294 vs-simple-encoder soc:dsi-output: encoder_bind end 6.931295 us speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.97276 ALSA device list: 8.980246 No soundcards found. 8.985685 Freeing unused kernel image (initmem) memory: 2168K 8.991299 Run /init as init process 8.993381 /init 9.000640] with environment: 9.003781 H0ME=/ 9.006139 TERM=linux Starting syslogd: OK Running sysctl: OK Running sysctl: OK </pre>	Γ	6.900619]	vs-simple-encoder_soc:rgb-output: encoder_bind end
<pre>6.914641 vs-simple-encoder socidsi-output: encoder_bind begin 6.927391 cdns-dsi 295d0000.mip1: ===&gt;cdns_dsi_bridge_attach begin 6.932294 vs-simple-encoder socidsi-output: encoder_bind end 6.939294 vs-simple-encoder socidsi-output: encoder_bind end 6.939294 vs-simple-encoder socidsi-output: encoder_bind end 6.947967 [drm] Initialized starfive 1.0.0 20191101 for socidisplay-subsystem on minor 1. 7.287268 mmc_host mmcO: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.977276] ALSA device list: 8.980246] No soundcards found. 8.985685 Freeing unused kernel image (initmem) memory: 2168K 8.9957392] with arguments: 8.998368] /init 9.000640] with environment: 9.003781] HOME=/ 9.006139] TERM=linux Starting syslogd: OK Running sysctl: OK</pre>		6.906456]	starfive soc:display-subsystem: bound soc:rgb-output (ops 0xffffffff80e74738)
<pre>6.920/39 cdns-dsi 295d0000.mip1: ==&gt;cdns_dsi_bridge_attach begin 6.93794 (us-dsi 295d0000.mip1: ==&gt;cdns_dsi_bridge_attach begin 6.933294 vs-simple-encoder soc:dsi-output: encoder_bind end 6.933294 vs-simple-encoder soc:dsi-output: encoder_bind end 6.939140 starfive soc:display-subsystem: bound soc:dsi-output (ops 0xfffffff80e74738) 6.947967 [drm] Initialized starfive 1.0.0 20191101 for soc:display-subsystem on minor 1.4 7.287268 mmc_host mmc0: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.97276 ALSA device list: 8.980246] No soundcards found. 8.9985685] Freeing unused kernel image (initmem) memory: 2168k 8.991729] Run /init as init process 8.993388 / init 9.000640] with environment: 9.003781 HOME=/ 9.006139] TERM=linux Starting syslogd: OK Running sysctl: OK </pre>	Ļ	6.914641	vs-simple-encoder soc:dsi-output: encoder_bind begin
<pre>[ 6.927091] CdnS-dS1 29500000.mip1: ==&gt;CdnS_dS1_DT13Ge_attach end 6.93294 vs-simple-encoder socidsi-output: encoder_bind end 6.939140 startive socidisplay-subsystem: bound socidsi-output (ops 0xffffffff80e74738) [ 6.947967] [drm] Initialized starfive 1.0.0 20191101 for socidisplay-subsystem on minor 1.4 7.287268 mmc_host mmcO: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) [ 8.977276] ALSA device list: 8.980246] No soundcards found. 8.985685] Freeing unused kernel image (initmem) memory: 2168K [ 8.991229] Run /init as init process 8.995392] with arguments: [ 8.998368] /init [ 9.000640] with environment: 9.003781] HOME=/ [ 9.0006139] TERM=linux Starting klogd: OK Running systl: OK</pre>	Ļ	6.920/39]	cdns-dsi 295d0000.mipi: ===>cdns_dsi_pridge_attach begin
<pre>6.932294 Vs-Simple-encoder Socids1-output: encoder_bind end 6.93140 startive socidisplay-subsystem: bound socids1-output (ops 0xfffffff80e74738) 4 6.947967 [drm] Initialized starfive 1.0.0 20191101 for socidisplay-subsystem on minor 1. 7.287268 mmc_host mmcO: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) 8.977276] ALSA device list: 8.980246] No soundcards found. 8.985685 Freeing unused kernel image (initmem) memory: 2168K 8.995729] Run /init as init process 8.995392] with arguments: 8.998368] /init 9.000640] with environment: 9.003781] HOME=/ 9.006139] TERM=linux Starting syslogd: OK Running sysctl: OK</pre>	Ļ	6.92/091]	cdns-ds1 29500000.mp1: ===>cdns_ds1_br1dge_attach end
<pre>6.9591401 Starlive Succonsplay-Subsystem. Double Social Science (Starline Starlive Social Science) (Starlive Science) (St</pre>	Ļ.	6.933294]	vs-simple-encoder socials-output: encoder_bind end
<pre>0.547307 [dimInterlated startive 1:00 2013/11/01 30C:013p1ay/subsystem 01 minor 1100000Hz div = 990) [ 7.287268 mmc_host mmc0: Bus speed (slot 0) = 198000000Hz (slot req 100000Hz, actual 100000Hz div = 990) [ 8.987276] ALSA device list: [ 8.980246] No soundcards found. [ 8.985685] Freeing unused kernel image (initmem) memory: 2168K [ 8.991729] Run /init as init process [ 8.995392] with arguments: [ 8.998368] /init [ 9.000640] with environment: [ 9.003781] HOME=/ [ 9.0006139] TERM=linux Starting klogd: OK Running systl: OK</pre>	F	6 947967]	Idrmit Tritialized starfive 1 0.0.20191101 for socialized startic minor 1
<pre>1.107000 mmc_host mmc bas speed (shot of = 15000000012 (shot req 10000012, actual 10000012 dr = 550) 8.997276] ALSA device list: 8.980246] No soundcards found. 8.985685 Freeing unused kernel image (initmem) memory: 2168K 8.991729] Run /init as init process 8.995392] with arguments: 8.998368] /init 9.000640] with environment: 9.0003781 HOME=/ 9.006139] TERM=linux Starting syslogd: OK Running sysctl: OK</pre>	4	7 287268	$[dim]$ introduced starting i.e. $b_{2}$ (shot no social spray-subsystem of introduced interval $div = 990$ )
<pre>8 .980246] No soundcards found. [ 8 .980246] No soundcards found. [ 8 .981729] Run /init as init process [ 8 .991729] Run /init as init process [ 8 .995392] with arguments: [ 9 .000640] with environment: [ 9 .003781] HOME=/ [ 9 .0006139] TERM=linux Starting klogd: OK Starting klogd: OK Running syscl: OK</pre>	F	8 9772761	Alsa davire list
<pre>8.985685] Freeing unused kernel image (initmem) memory: 2168K 8.991729] Run /init as init process 8.995392] with arguments: [ 8.998368] /init [ 9.000640] with environment: 9.003781] HOME=/ [ 9.0006139] TERM=linux Starting syslogd: OK Starting klogd: OK Running syscl1: OK</pre>	F	8,9802461	No soundcards found.
<pre>[ 8.991729] Run /init as init process [ 8.995392] with arguments: [ 8.998368] /init [ 9.000640] with environment: [ 9.003781] HOME=/ [ 9.006139] TERM=linux Starting syslogd: OK Starting klogd: OK Running sysctl: OK</pre>	È	8,9856851	Freeing unused kernel image (initmem) memory: 2168K
<pre>[ 8.995302] with arguments: [ 8.998368] /init [ 9.000640] with environment: [ 9.003781] HOME=/ [ 9.006139] TEMM=linux Starting syslogd: OK Starting klogd: OK Running sysctl: OK</pre>	Ē	8.991729	Run /init as init process
<pre>[ 8.998368] /init [ 9.000640] with environment: [ 9.003781] HOME=/ [ 9.006139] TERM=linux Starting syslogd: OK Starting klogd: OK Running sysctl: OK</pre>	Ē	8.995392]	with arguments:
<pre>[ 9.000640] with environment: [ 9.003781] HOME=/ [ 9.006139] TERM=linux Starting syslogd: OK Starting klogd: OK Running sysctl: OK</pre>	Ε	8.998368]	/init <sup>®</sup>
[ 9.003781] HOME=/ [ 9.006139] TERM=linux Starting syslogd: OK Starting klogd: OK Running sysctl: OK	E	9.000640]	with environment:
[ 9.006139] TERM=11nux Starting syslogd: OK Starting klogd: OK Running sysctl: OK	Ę	9.003781]	HOME=/
Starting Sysloga: OK Starting klogd: OK Running sysctl: OK	L	9.0061391	TERM=linux
Running sysctl: OK	St	arting sysic	iga: ok
Running System on	SU	arcing klogo	
	RU	initial system	· UN

#### Table 3-1 Start-up Logs

Legend	Description	
1	HDMI work status	
2	RGB2HDMI work status	
3	MIPI work status	
4	Display controller work status	

The log lines showing display controller and the HDMI are required before the debug.



Verify the connection status if you cannot find the above log records.

## 3.3. Debug Display

Follow the steps below to debug the display functions for your JH7110.

1. Follow the steps in Test Case Configuration (on page 20) 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 3-7 Debug Display 1
# modetest -M starfive
Incoders:         Description           id         crtc         type           115         0         TMDS           117         0         DSI           0x00000002         0x00000002
Connectors: id encoder status 116 0 modes connected HDMI-A-1 0x0 10 115 5
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 148500 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 #4 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 90000 flags: phsync, pvsync; type: driver #6 1280x720 47.95 1280 2240 2280 2500 720 725 730 750 89910 flags: phsync, pvsync; type: driver #7 1280x720 47.95 1280 2240 2280 2500 720 725 730 750 89910 flags: phsync, pvsync; type: driver #6 640x480 60.00 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver #9 640x480 59.94 640 656 752 800 480 490 492 525 25175 flags: nhsync, nvsync; type: driver
1 EDID:
blobs:
<pre>value:</pre>
<pre>flags: enum enums: On=0 Standby=1 Suspend=2 Off=3 value: 0 5 link-status:</pre>
Tlags: enum enums: Good=0 Bad=1 value: 0 6 non-dekton:
flags: immutable range values: 0 1

#### Table 3-2 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:

Figure 3-8 Debug Display 2



#### Table 3-3 Debug Display 2

Legend	Label	Description
1	id	The CRTC 0x00000001 mentioned in row (1) of table Table 3-2 : Debug Display 1 (on page 23), which means the CRTC is available for use.

Legend	Label	Description
2	id	The CRTC 0x00000002 mentioned in row (1) of table Table 3-2 : Debug Display 1 (on page 23), which means the CRTC is available for use.



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

#### • Debug output 3:

#### Figure 3-9 Debug Display 3

Planest id 1 formats: crtc 0 formats: xR12 uvvy vvuy vv12 v props: 8 type: 30 IN_FC	fb CRTC x,y x,y gamma size 0 0,0 0,1 0 0,1 0 0,1 0 0,1 0 0,1 0 0,1 0 0,0 0 0,
	value: 0100000000000000000000000000000000000
41 DEGAM 42 rotat	MA_MODE: flags: enum enums: disabled=0 preset degamma for BT709=1 preset degamma for BT2020=2 value: 0 :ion: flags: bitmask
43 pixel	values: rotate-0=0x1 rotate-90=0x2 rotate-180=0x4 rotate-270=0x8 reflect-x=0x10 reflect-y=0x20 value: 1 blend mode: flags: enum enums: None=2 Pre-multiplied=0 Coverage=1 value: 0
45 COLOR	Tags: range values: 0 65535 LENCODING: flags: enum enums: ITU-R BT.709 YCbCr=1 ITU-R BT.2020 YCbCr=2 value: 0

Table 3-4 Debug Display 3

Legend	Description	
1	The CRTC and its connected plane	

### 3.4. Test Example

#### For HDMI Output

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

modetest -M starfive -D 0 -a -s 116@31:1920x1080 -P 39@31:1920x1080@RG16 -Ftiles

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

- 116@31:1920x1080 <Connector ID>@<CRTC ID>: <Resolution>
- 39@31:1920x1080@RG16 <Plane ID>@<CRTC ID>: <Resolution>@<Format>

#### For MIPI Output

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

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

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>

#### For RGB2HDMI Output

The following command shows an example for testing the MIPI 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>

#### For Both MIPI and RGB2HDMI Outputs

If your board is connected with both a MIPI and a RGB2HDMI output devices, the following commands show an example for testing on each of them.

• For MIPI:

modetest -M starfive -D 0 -a -s 120@35:800x480 -P 74@35:800x480@RG16

• 120@35:800x480 - <Connector ID>@<CRTC ID>: <Resolution>

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

#### • For RGB2HDMI:

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

• 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.

