

Startup Screen User Guide

Issue 07

Date 2019-09-12

Copyright © HiSilicon (Shanghai) Technologies Co., Ltd. 2019. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon (Shanghai) Technologies Co., Ltd.

Trademarks and Permissions

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon (Shanghai) Technologies Co., Ltd.

Address: New R&D Center, 49 Wuhe Road,

Bantian, Longgang District, Shenzhen 518129 P. R. China

Website: http://www.hisilicon.com/en/

Email: <u>support@hisilicon.com</u>



About This Document

Purpose

This document provides the basic functions and boot command lines that are used to implement the boot screen function. You can configure specific functions and command lines as required.

NOTE

- Unless otherwise stated, the descriptions of Hi3559C V100 and Hi3559A V100 are the same.
- Unless otherwise stated, the descriptions of Hi3559 V200, Hi3516A V300, and Hi3516D V300 are the same.
- Unless otherwise stated, the descriptions of Hi3516E V300, Hi3518E V300, Hi3516D V200, and Hi3516E V200 are the same.
- Unless otherwise stated, the descriptions of Hi3556 V200 and Hi3559 V200 are the same.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3559A	V100ES
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3516D	V300
Hi3516A	V300
Hi3559	V200
Hi3556	V200
Hi3516E	V200
Hi3516E	V300
Hi3518E	V300



Product Name	Version
Hi3516D	V200

Intended Audience

This document is intended for:

- Technical support engineers
- Software development engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
△ DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
△ CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
□ NOTE	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Changes between document issues are cumulative. The latest document issue contains all changes made in earlier issues.

Issue 07 (2019-09-12)

In section 1.2, Tables 1-1 and 1-2 are added.

Section 1.3 is modified.

Issue 06 (2019-07-25)

Sections 1.3, 1.5, and 1.7 are modified.

Issue 05 (2019-06-20)

This issue is the fifth official release, which incorporates the following changes:

In section 1.3, the description of MIPI TX driver APIs is added.

Issue 04 (2019-05-15)

This issue is the fourth official release, which incorporates the following changes:

Section 1.2 is updated.

Issue 03 (2019-03-30)

This issue is the third official release, which incorporates the following changes:

The Hi3516D V200 description is added.

In sections 1.1, 1.2, 1.3, 1.5, and 1.7, the Hi3516E V200 description is updated.

Section 1.8 is updated.

Issue 02 (2019-03-15)

This issue is the second official release, which incorporates the following changes:

The descriptions of Hi3516E V300, Hi3518E V300, and Hi3516E V200 are added.

Section 1.1 is updated.

In section 1.2, the typical values of Hi3516E V200 are updated.

Issue 01 (2018-12-10)

This issue is the first official release, which incorporates the following changes:

The Hi3559 V200 description is added.

Sections 1.2, 1.4, and 1.8 are modified.

Section 1.7 is added.

Issue 00B06 (2018-11-20)

This issue is the sixth draft release, which incorporates the following changes:

The Hi3516D V300 description is added.

Issue 00B05 (2018-10-30)

This issue is the fifth draft release, which incorporates the following changes:

Section 1.1 to Section 1.3 are modified.

Issue 00B04 (2018-08-06)

This issue is the fourth draft release, which incorporates the following changes:

The Hi3519A V100 description is added.

Issue 00B03 (2018-03-15)

This issue is the third draft release, which incorporates the following changes:

The Hi3559A/C V100 description is added.

Issue 00B02 (2017-06-30)

This issue is the second draft release, which incorporates the following changes:

In section 1.2, the parameters of **startgx** are modified.

Issue 00B01 (2017-05-25)

This issue is the first draft release.



Contents

About This Document	j
1 Introduction.	
1.1 Overview	
1.2 Boot Command Lines	1
1.3 Boot Functions	6
1.4 Related Code	
1.5 Command Line Sample	8
1.6 Hardware Decoding	9
1.7 Smooth Transition.	9
1.8 Precautions	9



Tables

Table 1-1 Maximum resolution supported by the video layer	. 3
Table 1-2 Maximum resolution supported by the graphics layer.	3
Table 1-3 Chip differences	5



1 Introduction

1.1 Overview

The U-Boot code provides the following functions:

- Enables or disables the video output (VO) device in the boot environment, covering typical interfaces and timings of the VO device.
- Enables or disables the VO graphics layer in the boot environment.
- Enables or disables the VO video layer in the boot environment.
- Supports the output of red-green-blue (RGB) images (applies only to Hi3559A V100) after Joint Photographic Experts Group (JPEG) hardware decoding in the boot environment and display of these images at the VO graphics layer.
- Supports the output of YVU semi-planar 420 images after the JPEG hardware decoding in the boot environment and display of these images at the VO graphics layer.
- Sets the default display format of the VO graphics layer to **ARGB1555**, and that of the video layer to **YVU semi-planar 420**.

NOTE

Hi3516E V200 does not support hardware decoding. The input format of the video layer is YVU semi-planar 420, while that of the graphics layer is ARGB1555.

1.2 Boot Command Lines

• startvo: Starts the VO device.

Parameters: device ID, interface type, and timing

```
hisilicon # help startvo
startvo - open vo device with a certain output interface.
- startvo [dev intftype sync]
```

- **<dev>**: device ID. See **Table 1-3**.
- < intftype >: interface type. See Table 1-3.
- <sync>: timing type

Hi3559AV100:

```
0(PAL), 1(NTSC), 2(1080P24), 3(1080P25)
4(1080P30), 5(720P50), 6(720P60), 7(1080I50)
```



```
8(1080160),
                  9(1080P50),
                                    10(1080P60),
                                                       11 (576P50)
12(480P60),
                  13(800x600),
                                     14(1024x768),
                                                       15 (1280x1024)
                                    18(1280x800),
16(1366x768),
                  17(1440x900),
                                                       19(1600x1200)
20(1680x1050), 21(1920x1200),
                                     22(640x480),
                                                       23(960H PAL)
24(960H NTSC),
                                   26(2560x1440_30),
                  25(1920x2160),
27(2560x1440 60)
28(2560x1600 60),29(3840x2160 24),30(3840x2160 25),
31(3840x2160 30)
32(3840x2160 50),33(3840x2160 60),34(4096x2160 24),
35(4096x2160 25)
36(4096x2160_30),37(4096x2160_50),38(4096x2160_60),39(320x240_60)
40(320x240_50), 41(240x320_50), 42(240x320_60), 43(800x600_50)
44(720x1280_60), 45(1080x1920_60), 46(7680x4320_30)
```

The typical values for Hi3519A V100 are as follows:

```
0(PAL), 1(NTSC), 4(1080P30), 6(720P60)

10(1080P60), 21(1920x1200), 26(2560x1440_30),

31(3840x2160_30)

33(3840x2160_60), 45(1080x1920_60)
```

The typical values for Hi3516D V300 are as follows:

```
2(1080P24),
                    3(1080P25),
                                          4(1080P30),
5(720P50),
6(720P60),
                    7(1080I50),
                                          8(1080160),
9(1080P50),
10(1080P60),
                    11(576P50),
                                          12(480P60),
13 (800x600),
14(1024x768),
                    15 (1280x1024),
                                         16(1366x768),
17 (1440x900),
18 (1280x800),
                    19(1600x1200),
                                         20 (1680x1050),
21 (1920x1200),
                    45 (1080x1920)
22(640x480),
```

The typical values for Hi3516E V200 are as follows:

```
For BT656: 0(PAL), 1(NTSC)

For BT1120: 2(1080P24), 3(1080P25), 4(1080P30), 5(720P50)
6(720P60), 7(1080I50), 8(1080I60), 11(576P50)
12(480P60), 13(800x600), 14(1024x768),

22(640x480)

For LCD: 39(320x240P60), 40(320x240P50), 41(240x320P50),
42(240x320P60)
```

The timings above must be used with the specific device and interface type. For details, see chapter 4 "VO" in the *HiMPP V4.0 Media Processing Software Development Reference*.

• **stopvo**: Stops the VO device.

Parameter: device ID

```
hisilicon # help stopvo
stopvo - stopvo - close interface of vo device.
- stopvo [dev]
```

- <dev>: device ID. See Table 1-3.
- **startvl**: Starts the video layer.

Parameters: video layer ID, address of the decoded image, stride, and display position and size (x, y, w, h)

```
hisilicon # help startvl
startvl - startvl - open video layer.
- startvl [layer addr stride x y w h]
```

- <layer>: video layer ID. See Table 1-3.



- <addr>: image address
- **<stride>**: stride for image storage
- <x, y, w, h>: display position and size

Table 1-1 Maximum resolution supported by the video layer

Chip	Maximum Resolution		
Hi3559A V100	VHD0: 4096 x 4096 VHD1: 1920 x 1920		
Hi3519A V100	VHD0: 4096 x 4096 VHD1: 1920 x 1920		
Hi3516D V300/ Hi3516A V300	VHD0: 1920 x 1080		
Hi3559 V200	VHD0: 3840 x 2160		
Hi3556 V200	VHD0: 1920 x 1080		
Hi3516E V200	VHD0: 1920 x 1080		

• **stopvl**: Stops the video layer.

Parameter: video layer ID

```
hisilicon # help stopvl
stopvl - stopvl - close video layer.
- stopvl [layer]
```

- <layer>: video layer ID. See Table 1-3.
- **startgx**: Starts the graphics layer.

Parameters: graphics layer ID, address of the decoded image, stride, and display position and size (x, y, w, h)

```
hisilicon # help startgx
startgx - open graphics layer.
- startgx [layer addr stride x y w h]
```

- <layer>: graphics layer ID. See Table 1-3.
- **<addr>**: image address
- **<stride>**: stride for image storage
- <x,y,w,h >: display position, width, and height.

Table 1-2 Maximum resolution supported by the graphics layer

Chip	Maximum Resolution		
Hi3559A V100	VHD0: 3840 x 2160 VHD1: 1920 x 1920		
Hi3519A V100	VHD0: 4096 x 4096 VHD1: 1920 x 1920		



Chip	Maximum Resolution	
Hi3516D V300/ Hi3516A V300	VHD0: 1920 x 1080	
Hi3559 V200	VHD0: 3840 x 2160	
Hi3556 V200	VHD0: 1920 x 1080	
Hi3516E V200	VHD0: 1920 x 1080	

• stopgx: Stops the graphics layer.

Parameter: graphics layer ID

- <layer>: graphics layer ID. See Table 1-3.
- **setvobg**: Sets the background color of the device.

Parameter: graphics layer ID

```
hisilicon # help setvobg
setvobg - setvobg - set vo backgroud color.
- setvobg [dev color]
```

- <dev>: device ID. See Table 1-3.
- **<color>**: RGB color space
- **decjpg**: Enables JPEG decoding.

Parameter: Sets the output format after decoding.

```
hisilicon # help decjpg
decjpg - jpgd - decode jpeg picture.
- decjpg [format]
```

- **<format>**: 0: YVU SEMI-PLANAR 420, 1: ARGB1555, 2: ARGB8888

To use this command line, you need to set environment variables, including jpeg_addr, jpeg_size, jpeg_emar_buf, and vobuf.

jpeg addr: address for storing the raw streams of JPEG images

jpeg size: size of the raw streams of JPEG images

jpeg_emar_buf: buffer address used in JPEG image decoding. Its size is **256 KB**. **vobuf**: address for storing output RGB images after JPEG images are decoded

For example:

```
hisilicon #setenv jpeg_addr 0x90000000
hisilicon #setenv jpeg_size 0xb85f9
hisilicon #setenv jpeg_emar_buf 0x96000000
hisilicon #setenv vobuf 0xa0000000
```

NOTICE

Hi3519A V100, Hi3516D V300, Hi3516A V300, and Hi3559 V200 support the YVU semi-planar 420 output format only. Hi3516E V200 does not support this command.



Table 1-3 Chip differences

Table 1-3 Chip differences				
Chip	Device	Graphi c layer	Video layer	Interface type
Hi3559AV100ES	[0,1]	{0,1}	O,1} Display at the picture-in-picture (PiP) layer is not supported.	16(BT.1120), 32(HDMI), 1024(LCD_8BIT), 16384(MIPI_Tx)
Hi3559AV100/ Hi3519AV100	[0,1]	{0,1}	{0,1} Display at the PiP layer is not supported.	16(BT.1120), 32(HDMI), 16384(MIPI_Tx)
Hi3516DV300/ Hi3516AV300/ Hi3559V200	[0]	{0}	{0} Display at the PiP layer is not supported.	32(HDMI), 16384(MIPI_Tx), 512(LCD_6BIT), 1024(LCD_8BIT), 2048(LCD_16BIT), 4096(LCD_18BIT), 8192(LCD_24BIT)
Hi3516EV200	[0]	{0}	{0} Display at the PiP layer is not supported.	16(BT.1120), 512(LCD_6BIT), 1024(LCD_8BIT), 2048(LCD_16BIT)



NOTE

- Hi3559A V100ES, Hi3559A V100, Hi3519A V100, Hi3516D V300, Hi3516A V300, and Hi3559 V200 support boot screen output over the built-in HDMI and MIPI_TX interfaces. When the MIPI_TX interface is used for output, you need to modify the driver code of the MIPI_TX to adapt to the connected MIPI panel. When the BT.1120 interface is used for output, you need to compile a driver
- If you adopt user timings, modify the **vou_drv.c** code under U-Boot to change a timing in the supported range of **g_stSyncTiming** to the timing of the connected panel? For Hi3516E V200, you need to modify the user timing parameters in **g_stUSER_INTFSYNC_INFO** and **g_stUSER_SYNC_TIMING**.
- For the Hi3516D V300/Hi3516A V300/Hi3559 V200 LCD timing, you need to develop an LCD driver. In addition, you need to modify the VO_OUTPUT_USER part of the VOU_DRV_SetDevClk function in the vo_drv.c file to configure the VO clock.
- For the BT.1120 and LCD timings of Hi3516E V200, you need to develop BT.1120 and LCD drivers.
- The boot screen function of Hi3559 V200/Hi3516E V200 is disabled by default. You need to manually enable it by modifying the include\configs\hi35xx.h file (hi35xx is subject to the actual chip) in U-Boot and defining the CONFIG_OSD_ENABLE macro. If you want to use the HDMI, you need to cancel the definition of the CONFIG_OSD_HDMI_DISABLE macro.
- Hi3516E V200 does not support JPEG decoding. The video layer supports only YVU semi-planar 420 images. The graphics layer supports only BMP images, the pixel formats of which can only be ARGB1555. In addition, to ensure normal display, the width of an image must be 8-pixel aligned and flipped vertically first.

1.3 Boot Functions

The following functions are provided for encoding in the boot:

startvo

int start_vo(unsigned int dev, unsigned int type, unsigned int
sync);

Note: Only the devices with the device IDs listed in Table 1-3 are supported.

stopvo

int stop vo (unsigned int dev);

strartvl

int start_videolayer(unsigned int layer, unsigned addr, unsigned int strd, unsigned int x, unsigned int y, unsigned int w, unsigned int h);

Note:

- JPEG decoding is hardware-based. For Hi3559A V100, the output format can be **YVU semi-planar 420**, **ARGB1555**, or **ARGB8888**. For Hi3519A V100/Hi3516D V300, the output format can be **YVU semi-planar 420**.
- The value of **strd** (stride) can be obtained from a decoded JPEG image. The stride is displayed in the printed information when the **decipg** command is executed.
- For Hi3516E V200, **strd** is the image width.
- The stride must be 16-byte aligned. Otherwise, image display errors may occur.
- addr indicates the address of a decoded image and can be obtained from vobuf.
 When the video layer is displayed, the address of the Y component is specified by addr by default, and the address of the C component is calculated as follows:
 c_addr = addr + stride x h. For Hi3516E V200, addr is the address of the memory where the image is stored. The addr value must be 16-byte aligned.



- Only baseline decoding is supported.
- The display area of the video layer cannot exceed that of the device. Otherwise, image display is abnormal.

stopvl

int stop videolayer (unsigned int layer);

strartgx

int start_gx(unsigned int layer, unsigned long addr, unsigned int strd, unsigned int x, unsigned int y, unsigned int w, unsigned int h);

Note:

- The graphics layer displays data in ARGB1555 format by default. To enable the graphics layer to display data in ARGB8888 format, modifications are required.
- The value of **strd** can be obtained from a decoded JPEG image. The stride is displayed in the printed information when the **decipg** command is executed.
- For Hi3516E V200, **strd** is the image width.
- The stride must be 16-byte aligned. Otherwise, image display errors may occur.
- addr indicates the address of a decoded image and can be obtained from vobuf. For Hi3516E V200, addr is the address of the memory where the image is stored. The addr value must be 16-byte aligned. Otherwise, image display errors may occur.
- Only baseline decoding is supported.
- The display area of the graphics layer cannot exceed that of the device. Otherwise, image display is abnormal.

stopgx

int stop_gx(unsigned int layer);

setvobg

int set vobg (unsigned int dev, unsigned int rgb);

Note:

- Calling this function takes effect only before **startvo** is called. If this function is called after **startvo**, the calling takes effect only when **startvo** is called next time.
- It is recommended that the RGB color format be set to **0xRRGGBB**, which makes images clearer.

decipg

int jpeg decode (unsigned int format);

Note:

- jpeg_decode is used to decode images and store the decoded images in the memory.
- When jpeg_decode is called, four parameters are required. jpeg_addr indicates the memory address for storing the source image. jpeg_size indicates the image size (in bytes). jpeg_emar_buf indicates the buffer address used during decoding. vobuf indicates the address for storing the decoded image, that is, the start display position of the graphics layer.
- Hi3516E V200 does not support this function.

MIPI TX driver APIs

The MIPI TX driver APIs mipi_tx_module_init, mipi_tx_module_exit, and mipi_tx_ioctl are provided to interconnect with the MIPI display device. The mipi_tx_ioctl API has the following features: HI_MIPI_TX_SET_DEV_CFG, HI_MIPI_TX_SET_CMD, HI_MIPI_TX_ENABLE, HI_MIPI_TX_GET_CMD,



and **HI_MIPI_TX_DISABLE**. For details about how to use the APIs, see the *MIPI User Guide*. The APIs in the boot are the same as those described in the document. However, for the APIs in the boot, the MIPI TX device nodes do not need to be enabled and the file descriptor does not need to be input during calling.

Note:

- The mipi tx ioctl API must be called after mipi tx module init.
- Hi3516E V200 does not support the MIPI TX driver APIs.

1.4 Related Code

Only the basic functions relevant to the boot screen are provided. You can configure these functions based on the actual application, particularly in the decoding part which can be more flexible.

```
Makefile
include/hi35xx vo.h
include/configs/hi35xx.h
cmd/cmd_vo_hi35xx.c
cmd/cmd dec.c
cmd/Makefile
product/hiosd/vo/hi35xx (only one level of directory listed)
Makfile vou.c vou_coef.h vou_coef_org.c vou coef org.h vou def.h
vou drv.c vou drv.h
                    vou hal.c vou hal.h vou reg.h hi type.h
product/hiosd/hdmi/hdmi 2 0 (only one level of directory listed)
boot hdmi intf.c boot hdmi intf.h drv hi hdmi.h hi type.h
product/hiosd/mipi tx/hi35xx (only one level of directory listed)
type.h Makefile mipi tx.c mipi tx hal.c hi mipi tx.h mipi tx hal.h
mipi tx reg.h product/hiosd/dec (only one level of directory listed)
jpegd.c jpegd_drv.h jpegd_error.h jpegd_image.c mjpeg_func.h
mjpeg_image.c hi_type.h jpegd_drv.c jpegd_entry.c jpegd.h
jpegd_image.h jpegd_reg.h Makefile mjpeg_idct.c mjpeg_mcu.c
```

MOTE

- For Hi3559A V100ES, replace hi35xx with hi3559av100es.
- For Hi3559A V100, replace hi35xx with hi3559av100.
- For Hi3519A V100, replace hi35xx with hi3519av100.
- For Hi3516D V300, replace hi35xx with hi3516dv300.
- For Hi3516E V200, replace hi35xx with hi3516ev200, and the hdmi, mipi_tx, and dec configurations in the product/hiosd directory are not supported.

1.5 Command Line Sample

The following uses the HDMI 1080p@60 fps output from the Hi3559A V100 configuration device DHD0 as an example.

Note that the address for the Double Data Rate Synchronous Dynamic Random Access Memory (DDR SDRAM) download varies according to chips. The DDR SDRAM address is determined by the specific chip.

• Set environment variables and configure JPEG decoding parameters.

```
setenv jpeg_addr 0x92000000;
setenv jpeg_size 0x8f0b8;
setenv jpeg_emar_buf 0x960000000;
setenv vobuf 0xa00000000;
saveenv
```



Decode JPEG images and store the decoded images to the memory.

decjpg 0

Start DHD0.

startvo 0 32 10

Start V0

startvl 0 0xa0000000 1920 0 0 1920 1080

• Stop V0. stopvl 0

• Stop DHD0. stopvo 0

NOTE

Hi3516E V200 does not support the first two operations.

1.6 Hardware Decoding

The boot screen under U-Boot supports hardware decoding. The hardware-decoded output in RGB format can be displayed at the graphics layer. The hardware-decoded output in YUV format can be displayed at the video layer. Hi3519A V100 and Hi3516D V300 support the hardware-decoded output only in YUV format. Hi3516E V200 does not support hardware decoding.

1.7 Smooth Transition

Smooth transition refers to that the boot screen is smoothly switched to a service screen without disabling the display output. This requires that the boot screen and service screen use the same interface and timing.

NOTICE

- The HDMI smooth transition requires that after the system is entered, the HDMI attributes be the same as those configured for the boot screen under U-Boot.
- The HDMI supports only smooth transition at a timing specified by Consumer Electronics Association (CEA).

1.8 Precautions

- If the .ko drivers are loaded after the boot screen is displayed and the system starts running, the loading of the .ko drivers may be affected by the script **crgctrl_hi35xx.sh** that configures the clock and reset generator (CRG). In this case, you only need to comment out the configurations of the VO and HDMI in the script.
- If the boot screen is displayed over the BT.1120 interface, you need to port the peripheral chip driver that implements BT.1120-to-HDMI adaptation.
- If the boot screen is displayed over the BT.656 interface, you need to port the peripheral chip driver that implements BT.656.
- If the boot screen is displayed over the LCD interface, you need to port the peripheral chip driver that implements LCD.



- When the boot screen is displayed over the HDMI and the system is entered, to display
 the video in the same format as the boot screen, you need to delete the part for setting
 HDMI attributes from the process for properly running the HDMI. For details, see the
 VO sample program in the SDK of the chip.
- If the boot screen is displayed over the MIPI_TX interface, you need to modify the MIPI_TX driver code in U-Boot based on the connected MIPI_TX. The file path of the code is **product/hiosd/mipi_tx/hi35xx/**. In the **mipi_tx_display** function of the **mipi_tx.c** file under this path, configure **MipiTxConfig** based on the screen features in step 1. Then, configure the screen initialization sequence command based on the connected screen in step 2.