

# Akebi96 User's Guide

Issue: 2.0

Date: 2019-12-05

#### COPYRIGHT © 2019 Jiangsu HopeRun Software Co., Ltd. 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 Jiangsu HopeRun Software Co., Ltd.

#### Trademarks and Permissions



and other HopeRun icons are trademarks of Jiangsu HopeRun Software Co., Ltd.

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 HopeRun 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.

Jiangsu HopeRun Software Co., Ltd.

Address: HopeRun Software Park Software road No.168, Yuhuatai

Nanjing 210012

People's Republic of China

- Website: http://www.hoperun.com
- Email: hihope@hoperun.com

# **About This Document**

### Purpose

This document describes how to build and flash images to a akebi96 board and test akebi96's dvb driver.

### **Related Version**

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

Product Name	Version
Akebi96	all

### **Intended Audience**

This document is intended for: Technical support engineers

Software development engineers

1. Introduction	5
2. Boot Board	6
(1) . Download binary images	6
(2) . Flash binary images to board	6
3. Build Images	10
1.1. Install Packages	10
1.2. Build bootload & kernel	10
1.3. Build android	12
1.4. Flash images	12
4. Test Tuner	12
4.1 Establish mini digital TV transmission base station	12
4.2 Akebi96 hardware link	14
4.3 Install DVB Demos On akebi96 devices	17
4.4 Run DVB Demos	18
4.5 Test Videos	19

## Catalog

# 1. Introduction



1	DC Jack	2	Power Switch
3	HDMI TX Connector	4	Ether Connector
5	Coaxial/SPDIF Connector	6	USB 3.0 Port
7	8KDEC PCIE Connector	8	USB 2.0 Port
9	Boot Mode Switch	10	Reset Key
11	USB Boot key	12	MCU ICSP Connector
13	MCU UART Connector	14	USB Micro-B Port / UART1
15	USB Micro-B Port /UARTO/ STM	16	Expansion Connector
17	Tuner Card IF Connector	18	BT/WIFI Module
19	SC1408A Processor		

# 2. Boot Board

(1). Download binary images

#### cd ~

git clone https://github.com/hihope-akebi96/binary-images.git

(2). Flash binary images to board

i. Prepare USB stick

Set up USB memory formatted with FAT32 on PC, and find device file from dmesg. (ex. /dev/sdc1)  $\,$ 

Copy all files under "~/binary-images" into "usb" folder in USB stick.

sudo mount /dev/sdc1 /mnt sudo mkdir -p /mnt/usb sudo cp ~/binary-images/\* /mnt/usb/ sudo umount /mnt

ii. Preparation of target board.Boot mode must be set as "BE mode"\* Set as "BE BOOT" on "SW2002".





- iii. Connect USB stick to board usb 3.0 port.
- iv. Connect PC to board uart1 port with usb cable.

v. Push power switch tp power on board, and stop uboot prompt by any key on PC Serial Terminal(eg, minicom). After that, enter "run update\_from\_usb", then start to flash all rom images. It takes abount 90 second.

U-Boot 2019.01 (Feb 28 2019 - 14:31:04 +0900) SoC: LD20 (model 1, revision 2) Model: Akebi96 Board DRAM: 3 GiB SC: Micro Support Card (CPLD version 15.15) NAND: nand\_base: timeout while waiting for chip to become ready nand base: No NAND device found 0 MiB MMC: sdhc@5a000000: 0 Loading Environment from MMC... OK In: serial@54006800 Out: serial@54006800 serial@54006800 Err: MODE: eMMC Boot (STM: OFF) Net: Warning: ethernet@65000000 (eth0) using random MAC address - 3a:57:2d:50:4e:26 eth0: ethernet@65000000 Hit any key to stop autoboot: 0 => => run update\_from\_usb 5269 bytes read in 26 ms (197.3 KiB/s) ## Executing script at 8c000000 \*\*\*\*\*\*\* \*\*\* \*\*\* set GPT partition ... \*\*\*\*\*\*\* switch to partitions #0, OK mmc0(part 0) is current device Writing GPT: success! \*\*\*\*\*\*\* \*\*\* Writing to boot partition 0 ... \*\*\* \*\*\*\*\*\* switch to partitions #1, OK mmc0(part 1) is current device 39705 bytes read in 31 ms (1.2 MiB/s) MMC write: dev # 0, block # 0, count 256 ... 256 blocks written: OK 425472 bytes read in 23 ms (17.6 MiB/s) MMC write: dev # 0, block # 256, count 2816 ... 2816 blocks written: OK

\*\*\*\* Writing to normal area ... \*\*\*

\*\*\*\*\*\*\*

switch to partitions #0, OK mmc0(part 0) is current device 6610395 bytes read in 95 ms (66.4 MiB/s)

MMC write: dev # 0, block # 80, count 24576 ... 24576 blocks written: OK 23773 bytes read in 26 ms (892.6 KiB/s)

MMC write: dev # 0, block # 24656, count 64 ... 64 blocks written: OK 6899565 bytes read in 87 ms (75.6 MiB/s)

MMC write: dev # 0, block # 24720, count 131072 ... 131072 blocks written: OK 6612992 bytes read in 71 ms (88.8 MiB/s)

MMC write: dev # 0, block # 155792, count 16520 ... 16520 blocks written: OK 12673024 bytes read in 156 ms (77.5 MiB/s)

MMC write: dev # 0, block # 172312, count 40824 ... 40824 blocks written: OK 107151930 bytes read in 725 ms (140.9 MiB/s) Uncompressed size: 268435456 = 0x10000000

MMC write: dev # 0, block # 213136, count 524288 ... 524288 blocks written: OK 126873443 bytes read in 1442 ms (83.9 MiB/s) Uncompressed size: 268435456 = 0x10000000

MMC write: dev # 0, block # 737424, count 524288 ... 524288 blocks written: OK 93456661 bytes read in 1077 ms (82.8 MiB/s) Uncompressed size: 268435456 = 0x10000000

MMC write: dev # 0, block # 1261712, count 524288 ... 524288 blocks written: OK 118157455 bytes read in 1355 ms (83.2 MiB/s) Uncompressed size: 268435456 = 0x10000000

MMC write: dev # 0, block # 1786000, count 524288 ... 524288 blocks written: OK 42349274 bytes read in 309 ms (130.7 MiB/s) Uncompressed size: 268435456 = 0x10000000

MMC write: dev # 0, block # 2310288, count 524288 ... 524288 blocks written: OK 260547 bytes read in 32 ms (7.8 MiB/s) Uncompressed size: 268435456 = 0x10000000

MMC write: dev # 0, block # 2834576, count 524288 ... 524288 blocks written: OK

54509644 bytes read in 639 ms (81.4 MiB/s)

MMC Sparse write: dev # 0, block # 3358864 ... Flashing Sparse Image ...... wrote 54509568 bytes to '0x66812000' 1737068 bytes read in 42 ms (39.4 MiB/s)

MMC Sparse write: dev # 0, block # 3621008 ... Flashing Sparse Image ....... wrote 1736704 bytes to '0x6e812000' 319756 bytes read in 27 ms (11.3 MiB/s)

\*\*\* USB update is finished. \*\*\*

\*\*\*\*\*\*

vi. Reset board.

=> reset

#### U-Boot 2019.01 (Feb 28 2019 - 14:31:04 +0900) SoC: LD20 (model 1, revision 2) Model: Akebi96 Board DRAM: 3 GiB SC: Micro Support Card (CPLD version 15.15) NAND: nand\_base: timeout while waiting for chip to become ready nand base: No NAND device found 0 MiB MMC: sdhc@5a000000:0 Loading Environment from MMC... OK In: serial@54006800 Out: serial@54006800 Err: serial@54006800 MODE: eMMC Boot (STM: OFF) Net: Warning: ethernet@65000000 (eth0) using random MAC address - 3a:57:2d:50:4e:26 eth0: ethernet@65000000 Hit any key to stop autoboot: 0 => => env default -a ## Resetting to default environment => saveenv Saving Environment to MMC... Writing to MMC(0)... OK

After these operations, execute reset or power off -> on, then the board will be wake up by new

ROM.

# 3. Build Images

# 1.1.Install Packages

apt-get install --fix-missing -y git bc cmake ncurses-dev autoconf bison ccache cscope curl flex gdisk libfdt-dev libglib2.0-dev libpixman-1-dev netcat python-crypto python-serial uuid-dev xz-utils zlib1g-dev gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev xterm cpio libssl-dev rsync

apt-get install -y mtd-utils genromfs sudo stgit device-tree-compiler python3 iputils-ping iasl sparse bsdmainutils u-boot-tools img2simg repo openjdk-8-jdk ccache libgl1-mesa-dev libxml2-utils xsltproc lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev zip dosfstools mtools simg2img connect-proxy locales python-mako python-pycryptopp kmod

cd /opt/ wget http://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/arm-linux-gnueabihf/gcclinaro-7.3.1-2018.05-x86\_64\_arm-linux-gnueabihf.tar.xz tar xf gcc-linaro-7.3.1-2018.05-x86\_64\_arm-linux-gnueabihf.tar.xz wget http://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/aarch64-linux-gnu/gcc-lin aro-7.3.1-2018.05-x86\_64\_aarch64-linux-gnu.tar.xz tar xf gcc-linaro-7.3.1-2018.05-x86\_64\_aarch64-linux-gnu.tar.xz export PATH="\$PATH:/opt/gcc-linaro-7.3.1-2018.05-x86\_64\_arm-linux-gnueabihf/bin:/opt/gcc-linaro-7.3 .1-2018.05-x86\_64\_aarch64-linux-gnu/bin"

## 1.2. Build bootload & kernel

(1). Downlaod bootload & kernel source codes

mkdir ~/aosp/bsp cd ~/aosp/bsp git clone https://github.com/buildroot/buildroot -b 2018.02.6 --single-branch git clone https://github.com/96boards-akebi96/buildroot-configs.git -b master --single-branch (2). Build bootload & kernel source codes

cd ~/aosp/bsp/buildroot make BR2\_EXTERNAL=../buildroot-configs akebi96\_defconfig make clean make

Notes :

- i. Do not add "-j" option. It will be optimised by "buildroot".
- ii. Internet connection must be available on PC for using build.
  - \* Packages needed to be build will be download automatically at the first build.
  - \* After the first time, these will be cached under "bsp/build-local/dl" .
- iii. For buildroot, please refer following links.
  - \* https://buildroot.org/
  - \* https://buildroot.org/downloads/manual/manual.html

### 1.3. Build android

(1). Download android source code

cd ~/aosp/android repo init -u https://android.googlesource.com/platform/manifest -b android-9.0.0\_r34 git clone -b sni-release --single-branch https://github.com/96boards-akebi96/akebi96-manifests.git .repo/local\_manifests repo sync -j8

(2). Build android source code

cd ~/aosp/android source build/envsetup.sh lunch akebi96-userdebug make -j8 ./make\_romimage.sh

Notes :

i. Android ROM will be installed in "~/aosp/bsp/buildroot/output/images" after these procedure.

ii. This may take a few hours.

### 1.4. Flash images

All builded images are installed in "~/aosp/bsp/buildroot/output/images". Flash them flow Chapter 2.

# 4. Test Tuner

### 4.1 Establish mini digital TV transmission base station.

In the test state, we use "DekTec output adapters" & "streamxpress player" to establish mini digital TV transmission base station in the laboratory.

(1). You need a "Dektec output adapter".

List of supported devices:

1. DTA-100, 105, 112, 115

2. DTA-116,117,145,160
3. DTA-2136,2137C,2142
4. DTA-2144B,2145,2152
5. DTA-2154,2160,2172
6. DTA-2174,2175,2179
7. DTA-2195
8. DTE -3100

(2). Install "StreamXpress player".

LINK: <a href="https://www.dektec.com/products/applications/StreamXpress/">https://www.dektec.com/products/applications/StreamXpress/</a>

(3). Open StreamXpress player and choose TS file.

🗟 DekTec StreamXpress - Stream	Player		
File Edit Tools Settings Help File C:VDocuments and SettingsVhoperun/Des	ktop/DVBT100.ts	🕞 Open	Adapter 1: MOD (DTA-115)
Transport-Stream 16384     TV Moving Picture (5.45 Mbps)     W Monoscope (5.45 Mbps)     TV Color Bar (5.44 Mbps)     TV Color Bar (5.44 Mbps)     D 17 SDT-actual (1.97 kbps)     D 18 (9.86 kbps)     D 201 Comp. Program 166     D 222 Comp. Program 166     D 221 Comp. Program 166     D 201 Comp. Program		ops) 768 (5.13 Mb) 768 (313 kbp: 832 (5.13 Mb) 832 (5.13 Mb) 832 (5.13 Mb) 960 (75.13 Mb)	File - DVBT100.ts - 166,512,352 bytes - 885,704 packets Packet Size - 188 bytes Estimated Rate - TS: 22,000,000 bps
Rate         Out:         31,668,449         bps         ✓ RMX           TS:         22,000,000         bps         default           Play-Out         0:00         0         0	Channel 474.000 🌍 MHz	Modulation Para DVB-T V 64-QAM V 1:00	ameters 8 MHz V 8K V G=1/32 V 7/8 V more Memory buffers 32 MB; 4 MB
#Errors Time	Wrap #Wraps	Total Time 0:09	TxFIFO16 MB HW 16 MB

(4). Setting Stream Player:

e Edit Tools Settings Help "ile			Adapter
C:∿Documents and Settings\hoperun\Des	sktop <sup>i</sup> DVBT100.ts	🔁 Open	1: MOD (DTA-115)
Image: Transport-Stream 16384       PID Info         Image: Transport-Stream 16384       Image: Transport-Stream 16384         Image: Transport-Stream 16384       Image: Transport-Stream 16384         Image: Transport-Stream 16384       Image: Transport-Stream 16384         Image: Transport-Stream 16384       Image: Transport-Stream 16768         Image: Transport-Stream 16768       Image: Transport-Stream 1676		<ul> <li>s)</li> <li>8 (5.13 Mb)</li> <li>8 (313 kbp:</li> <li>2 (5.13 Mb)</li> <li>2 (313 kbp:</li> <li>0 (5.13 Mb)</li> </ul>	File DVBT100.ts 166,512,352 bytes 885,704 packets Packet Size 188 bytes Estimated Rate TS: 22,000,000 bps
tate Dut: 31,668,449 bps	Channel 474.000 MHz	Modulation Part	ameters 8 MHz • 8K • G=1/32 • 7/8 • more

Ps:The value showed in this picture is the default config in tuner's driver.

(5). Play TS file:

0.ts	😂 Open	Adapter 1: MOD (DTA-115	
<ul> <li>■ Transport-Stream 16384</li> <li>■ Moving Picture (5.45 Mbps)</li> <li>■ 0 PAT (16.8 kbps)</li> <li>■ 0 PAT (197 kbps)</li> <li>■ 16 NIT (1.97 kbps)</li> <li>■ 17 SDT-actual (1.97 kbps)</li> <li>■ 18 (9.86 kbps)</li> <li>■ 201 Comp. Program 16766</li> <li>■ 222 Comp. Program 16833</li> <li>■ 222 Comp. Program 16833</li> <li>■ 221 Comp. Program 16833</li> <li>■ 221 Comp. Program 16833</li> <li>■ 221 Comp. Program 16835</li> <li>■ 281 Comp. Program 16864</li> </ul>		File DVBT100.ts 166,512,352 b 885,704 packe Packet Size 188 bytes Estimated Rate TS: 22,000,000	ytes ts ) bps
MHz #Wraps T	Modulation Para DVB-T 64-QAM 1:00 tal Time	Inneters 8 MHz V 8K G=1/32 V 7/8 Memory buffers 32	• more
	0.ts	0.ts	0.ts

## 4.2 Akebi96 hardware link

There ares three devices need here, there are akebi96 board, tuner board, antenna.





Connect like this:



# 4.3 Install DVB Demos On akebi96 devices

1. download source codes

git clone https://github.com/hihope/akebi96-dvbdemos.git

2. Build

./init-android.sh

cd android/contrib ./compile-ffmpeg.sh clean ./compile-ffmpeg.sh all

cd ..

./compile-ijk.sh all

# Android Studio:

- # Open an existing Android Studio project
- # Select android/ijkplayer/ and import
- # Select Run/Run ijkplayer-examples

	Dun Sikalawar ayamalat	Low DA				
ikalawar \ Do iikalawar ayamala \ Do oro \ Do main \ 40 Androidhtaoifact yal	Apply Changes: No active 'likela	ver-evemple' launch	halana ananala wila a da da m	0. * = 0+ /		÷ 0 I
schalter / main	Debug 'iikplayer-example'	^D	kplayer-example •	, m ta	V O D   M2 14	
👷 Android 👻 😳 🛨 🗢	Run 'iikplayer-example' with Cov	erage				
n likplayer-arm64	Profile 'iikplayer-example'					
manifests	Pup	635 B				
java	P Dahua	0250				
Is generated Java	Debug					
▶ IIi jniLibs	m Profile					
▶ Ingres	Record Espresso Test					
na ijkplayer-armv5	the Attach to Process	<b>℃</b> ①F5				
na ijkplayer-armv7a	Edit Configurations					
n 🖹 ijkplayer-example	Import Test Results	►				
v Emanifests	Stop	36F2				
AndroidManifest.xml	Show Running List					
🕨 🖿 java	Stop Background Processes	Q 26.F2				
v 🔤 cpp						
Includes	<ul> <li>Restart Activity</li> </ul>					
CMakeLists.txt	🔿 Step Over	F8				
dvb_jni.cpp	🚈 Force Step Over	X OF8				
dvb_jni.h	+ Step Into	F7				
DvbDevice.cpp	+ Force Step Into	X Orz				
DvbDevice.h	A Smart Step Into	0.07				
Eg generated Java	* Stop Out	0.07				
In the second	T step Out	U Pa				
lia ijkplayer-exo	T Run to Cursor	V.F9				
ijkplayer-java	* Force Run to Cursor	<b>乙</b> 36F9				
ijkplayer-x86	Force Return					
ma ijkplayer-x86_64	Throw Exception					
W Gradie Scripts	II Pause Program					
w build.gradie (Project: ijkplayer)	I▶ Resume Program	\℃ 將R				
w build.gradie (Module: ijkplayer-arm64)	El Duchuste Ducascales	27.44				
W build.gradie (Module: ijkplayer-armvb)	E Evaluate Expression	CF8				
	Quick Evaluate Expression	× 36F8				
Build: Sync ×	E Show Execution Point					\$
🙀 🔻 🖌 ijkplayer: synced successfully at 2019-10-15 09:45	Toggle Line Breakpoint	96 F.B				15 s 598
Starting Gradle Daemon						851
🖈 🔍 🔻 Run build /Users/andy/workspace/projects/akebi96/psi-ijkplayer/android/ijkplayer	Toggle Temporary Line Breakpoint	大众 Hira				11 s 290
🔻 🖌 Load build						1 s 110
Run init scripts	View Breakpoints	☆ 第68				1 s 48
Evaluate settings	-					55
Configure build	Get thread dump					8 s 608
Calculate task graph	5 Attach Debugger to Android Pro	cess				105
▶ ✓ Run tasks						1 s 320

### 4.4 Run DVB Demos

Run demos on devices Following steps below :

# Select Settings

- # Select Apps
- # Select Brow system apps
- # Select Show all apps
- # Select ijkplaydemos
- # Select Open

App will run like this :



# 4.5 Test Videos

demo videos url: https://pan.baidu.com/s/1Pt\_vJDRFkcpHj71uNldvow password: 24an