

# HiKey Getting Started Guide

Linaro

July 20, 2015

## **Abstract**

This is the Getting Started Guide for the HiKey consumer edition 96Board. It describes how to use the HiKey board, update the operating systems on the board, and recover the board should things go awry when using it. It also contains notes regarding the hardware, known issues at the time of publishing this document, as well as information on how to build the key software components for the HiKey board.

---

# 96Boards HiKey

## Getting Started

This document describes how to get started with the HiKey ARMv8 community development boards shipped from May 2015.

**The following information is provided in these release notes:**

1. Pre-Installed Debian Linux  
Information on the Debian 8.0 (“jessie”) OS installation software provided with the HiKey board as shipped from the factory.
2. Installing Android Open Source Project  
Information on loading the AOSP version of Android 5.1 as an alternative OS onto the HiKey board
3. Updating the OS  
Information on loading an OS update from 96Boards.org
4. Board Recovery  
Information on board recovery and/or loading bootloader software onto the HiKey board
5. Hardware notes
6. Known Issues
7. Building Software from Source Code  
Information on building software for the HiKey board from source code
8. Appendices  
Information on the partition table used on HiKey and the contents of the boot partition.

## Updating from the Early Access Build

If you already have a HiKey board delivered before May 2015 under the Early Access program you will need to do the following:

- First, follow the instructions in Section 4. Board Recovery - Installing a Bootloader, to update the bootloader software on your board
- Then follow the instructions in Section 3. Updating the OS, to install either the Debian or the Android Open Source Project (AOSP) build

## 1. Pre-Installed Debian Linux

The HiKey board is ready to use “out of the box” with a preinstalled version of the Debian Linux distribution.

To get started you will need a power supply, an HDMI monitor and a USB keyboard and mouse.

### IMPORTANT NOTES

- At present the HDMI EDID display data is not used. A fixed HDMI timing is used at 1280x720p 60Hz. This will work with most but not all monitors/TVs. A future software update is expected to address this issue.
- There are limitations on the usage of the USB ports on the HiKey board. Please refer to the Hardware section in the document for further information.

---

## Power Supply

The HiKey board requires an external power supply providing 12V at 2A. (The board will also work with 9V or 15V power supplies). It is not possible to power the board from a USB power supply because the board can use more power than is available from a standard USB power supply.

The HiKey board uses a standard DC Jack with a 1.7mm barrel, center pin positive. An adapter cable is provided with the board to also enable the use of power supplies with 2.1mm barrel jacks.

## Monitor, Keyboard and Mouse

A standard monitor or TV supporting 720p resolution is required. The keyboard and mouse can be combined or separate.

## Powering up the Board

Link 1-2 causes HiKey to auto-power up when power is applied. The other two links should be not fitted (open). If Link 1-2 is not installed then the back edge push button switch is used to power on the HiKey board.

Please refer to the Hardware User Guide (Chapter 1. Settings Jumper) for more information on board link options.

A few seconds after applying power the right hand green User LED0 will start flashing once per second. The next User LED1 is used as a disk indicator showing access to the on-board eMMC flash memory. The startup console messages will then appear on the connected HDMI screen.

After about 10 seconds the LXDE User Interface will appear and you can start using the HiKey Linux software.

Next we describe how to set up wireless or wired networking and Bluetooth interfaces.

## Wireless Network

The HiKey board includes built in 2.4GHz IEEE802.11g/n WiFi networking. The board does not support the 5GHz band. To use the wireless LAN interface you will need to edit the following file. To do this first open a terminal window by selecting LXTerminal from the System Tools menu option. Then type the following into the terminal window.

```
$ sudo leafpad /etc/network/interfaces.d/wlan0      Using UI editor, or
$ sudo vi /etc/network/interfaces.d/wlan0          Using vi editor
```

Remove the # characters from the start of each line. Add your wireless network name into the ssid line, and network password into the psk line and save the file.

Next reboot the HiKey board:

```
$ sudo reboot
```

After the HiKey board has rebooted the wireless network should be active. The yellow LED between the microUSB and the Type A USB on the front board edge indicates wireless network activity.

## Wired Network

You can connect to a wired network by using a USB Ethernet adapter. To use a USB ethernet adapter you will need to edit the following file. To do this first open a terminal window by selecting LXTerminal from the System Tools menu option. Then type the following into the terminal window.

```
$ sudo vi /etc/network/interfaces.d/eth0
```

Remove the # characters from the start of each line. Next reboot the HiKey board:

```
$ sudo reboot
```

After the HiKey board has rebooted the wired network should be active.

---

## Bluetooth

The HiKey board includes built-in Bluetooth 4.0 LE support.

To setup a Bluetooth device open the Bluetooth Manager from the Preferences menu. If a “Bluetooth Turned Off” popup appears then select “Enable Bluetooth”

Click on Search to search for devices

## Audio Device

- Select the audio device you want to use
- If you have not used this device before then Click Setup
- Click Next (use Random PassKey unless device requires a specific PassKey)
- Click Next (connect to audio sink)
- After a few seconds you should see that the device has been successfully added. Click on the Close button

Now open the LXMusic player from the Sound & Video menu. Create a playlist from your music files. Supported audio formats are .mp3 and .ogg. You should now be able to play from the LXMusic player.

## Keyboard/Mouse

- Select the keyboard device you want to use
- If you have not used this device before then Click Setup
- Click Next (use Random PassKey unless device requires a specific PassKey)
- Click Next (connect to Human Interface Device HID)
- After a few seconds you should see that the device has been successfully added. Click on the Close or Cancel button

After the device has been successfully connected you should be able to use the Bluetooth keyboard/mouse. If you make the device trusted then this should operate over a reboot of the board.

## Other Useful Information

### 1. Updating and Adding Software

Before adding any software to your system you must do an update as follows:

```
$ sudo apt-get update
```

You can now add Debian packages to your system:

```
$ sudo apt-get install [package-name]
```

You can search for available packages here:

<https://www.debian.org/distrib/packages>

Search the stable distribution for packages for the HiKey.

### 2. File Systems

The following is the default file system layout for HiKey running Linux:

---

```

/dev/mmcblk0p4    64M   15M   49M   /boot    copy of boot file system
/dev/mmcblk0p9    3.0G  1.2G  1.8G   /        main user space file system

```

### 3. Logging in

The default user name is “linaro” and the default password for user linaro is “linaro”.

### 4. Clock

The HiKey board does not support a battery powered RTC. System time will be obtained from the network if available. If you are not connecting to a network you will need to manually set the date on each power up

### 5. USB

A utility is provided in /home/linaro/bin to change the configuration of the host (Type A and Expansion) and OTG USB ports. By default these ports operate in low/full speed modes (1.5/12 Mbits/s) to support mouse/keyboard devices. Other USB devices such as network or storage dongles/sticks will be limited to full speed mode. Using the usb\_speed utility it is possible to support high speed devices (480 Mbits/s) as long as they are not mixed with full/low speed devices.

For information on using the utility do the following:

```
$ sudo ~/bin/usb_speed -h
```

Please refer to the Hardware Notes section below for further information on the USB port configuration of the HiKey board.

### 6. System and User LEDS

Each board led has a directory in /sys/class/leds. By default the LEDs use the following triggers:

LED	Trigger
wifi_active	phy0tx
bt_active	hci0tx
user_led1	heartbeat
user_led2	mmc0 disk access
user_led3	mmc1 disk access (microSD card)
user_led4	CPU core 0 active (not used)

To change a user LED you can do the following as a root user:

```

$ su bash
# echo heartbeat > /sys/class/leds/<led_dir>/trigger    make a LED flash
# cat /sys/class/leds/<led_dir>/trigger                  show triggers
# echo none > /sys/class/leds/<led_dir>/trigger          remove triggers
# echo 1 > /sys/class/leds/<led_dir>/brightness          turn LED on
# echo 0 > /sys/class/leds/<led_dir>/brightness          turn LED off
# exit
$

```

## 2. Installing Android Open Source Project

Users may install a version of the Android Open Source Project (AOSP) onto the HiKey board. This will remove the factory installed Debian Linux OS. This section provides instructions on installing the AOSP build which consists of:

- Derived from Linux 3.18 kernel

- 
- AOSP Android Lollipop latest release (5.1)

Download the following files from:

<http://builds.96boards.org/releases/hikey/linaro/aosp/15.06>

- boot\_fat.img.tar.xz
- cache.img.tar.xz
- system.img.tar.xz
- userdata.img.tar.xz
- ptable-aosp.img

Uncompress the .tar.xz files using your operating system file manager, or with the following command for each file:

```
$ xz --decompress [filename].tar.xz; tar -xvf [filename].tar
```

To install updates you will need a Linux PC with fastboot support. For information on installing and setting up Fastboot see Section 4. Board Recovery - Installing a Bootloader below.

After setting up Fastboot on your Linux PC do the following:

Install Link 5-6 on the HiKey board. This tells the bootloader to start up in fastboot mode.

Power on the HiKey board and verify communications from the Linux PC:

```
$ sudo fastboot devices
0123456789abcdef fastboot
```

Then install the update using the downloaded files.

Note that the ptable must be flashed first

Note also that the larger system file will take longer due to its size.

```
$ sudo fastboot flash ptable ptable-aosp.img
$ sudo fastboot flash boot boot_fat.img
$ sudo fastboot flash cache cache.img
$ sudo fastboot flash system system.img
$ sudo fastboot flash userdata userdata.img
```

When flashing is completed power down the HiKey, remove Link 5-6 and power up the HiKey. You may now use the AOSP operating system. Note the first time boot up will take a couple of minutes.

Please read the Hardware notes and the Known Issues later in this document before using the OS.

### 3. Updating the OS

Updates to 96Boards supported operating systems will be made available from time to time at:

<http://builds.96boards.org/releases/hikey>

IMPORTANT NOTE:

The installation process will overwrite all contents of the eMMC memory. This will remove all installed software and all user files. Before updating the OS make sure that you have saved any user files or data that you want to keep onto an SD Card or USB memory stick etc.

To install updates you will need a Linux PC with fastboot support. For information on installing and setting up Fastboot see Section 4: Board Recovery - Installing a Bootloader below.

Once fastboot is installed on the Linux PC proceed as follows:

---

## Debian Linux OS

Download the following files onto your Linux PC from:

<http://builds.96boards.org/releases/hikey/linaro/debian/15.06>

- boot-fat.emmc.img.gz
- hikey-jessie\_alip\_20150701-323.emmc.img.gz
- ptable-linux.img

Note that the jessie image is a large file and may take several minutes (or longer on a slow internet connection) to load. You will need to accept the end user license for the Mali GPU software before you are able to download the OS image.

Unzip the .gz files (using gunzip or equivalent)

Install Link 5-6 on the HiKey board. This tells the bootloader to start up in fastboot mode.

Power on the HiKey board and verify communications from the Linux PC:

```
$ sudo fastboot devices
0123456789abcdef fastboot
```

Then install the update using the downloaded files:

Note: The ptable must be flashed first.

Note: The larger system file will take longer and will be loaded in several chunks due to its size.

```
$ sudo fastboot flash ptable ptable-linux.img
$ sudo fastboot flash boot boot-fat.emmc.img
$ sudo fastboot flash system hikey-jessie_alip_20150701-323.emmc.img
```

When completed, power down the HiKey, remove Link 5-6 and power up the HiKey. If you wish to use a keyboard and mouse in the Type A USB ports remember to remove the microUSB cable.

You may now use the updated OS.

### Using an SD Card

The built-in HiKey eMMC boot software also enables booting a kernel and root file system installed on an SD card. If an SD card is installed at power up the HiKey board will boot the software on the SD Card rather than the software flashed in the eMMC.

This section describes how to prepare a bootable SD card.

Download the following file onto your Linux PC from:

<http://builds.96boards.org/releases/hikey/linaro/debian/15.06>

- hikey-jessie\_alip\_20150701-323.img.gz

Unzip the .gz file. Install an SD card into your Linux PC. Make sure that you know the SD card device node before carrying out the next step.

**Note:** for this example we assume the device node is `/dev/sdb`. Replace with your assigned SD card device.

```
$ sudo dd if=hikey-jessie_alip_20150701-323.img of=/dev/[sdb] bs=4M oflag=sync status=noxfer
```

If your SD card is more than 2GB capacity you may want to change the rootfs to use the rest of the SD card as follows:

---

```
$ sudo fdisk /dev/sdb
```

- use p to list partitions
- note the start cylinder number of rootfs
- use d to delete the root partition info
- use n to create the new primary partition (the start cylinder must be same as before)
- use w to write the partition table (don't worry about error message)
- remove the disk and re-insert

Then the following command will make the file system take up all the space left on the SD card

```
$ sudo resize2fs /dev/sdb2
```

If you power up and boot the HiKey board with the SD card the kernel and software on the SD card will be used and not the eMMC. Your user files will also be created on the SD card. You may still access the eMMC files as follows:

```
$ sudo mount /dev/mmcblk0p9 /mnt
```

Note: Do not mount and access other partitions on the eMMC unless you are an expert. The bootloader and other binary files necessary for correct operation are stored in the eMMC and if they are removed or changed your board may become “bricked”, in which case all your data will be lost and you will need to follow the process in Section 4: Board Recovery to reload the HiKey software.

## Android Open Source Project (AOSP)

Download the following files from:

<http://builds.96boards.org/releases/hikey/linaro/aosp/15.06>

- boot-fat.img.tar.xz
- cache.img.tar.xz
- system.img.tar.xz
- userdata.img.tar.xz
- ptable-aosp.img

Uncompress the .tar.xz files using your operating system file manager, or with the following command for each file:

```
$ xz --decompress [filename].tar.xz; tar -xvf [filename].tar
```

Install Link 5-6 on the HiKey board. This tells the bootloader to start up in fastboot mode.

Power on the HiKey board and verify communications from the Linux PC:

```
$ sudo fastboot devices  
0123456789abcdef fastboot
```

---

Then install the update using the downloaded files:

```
$ sudo fastboot flash ptable ptable-aosp.img
$ sudo fastboot flash boot boot_fat.img
$ sudo fastboot flash cache cache.img
$ sudo fastboot flash system system.img
$ sudo fastboot flash userdata userdata.img
```

When completed power down the HiKey, remove Link 5-6 and power up the HiKey. If you wish to use a keyboard and mouse in the Type A USB ports remember to remove the microUSB cable.

You may now use the updated OS.

Please read the Hardware notes and the Known Issues later in this document before using the OS.

## 4. Board Recovery

### Installing a Bootloader

For most users a board can be “recovered” from a software failure by reloading the operating system using the instructions provided above. However, if the primary bootloader in the eMMC flash memory has been corrupted then the bootloader will need to be re-installed. This section describes how to reinstall the primary bootloader.

#### Preparation

Download the following files onto a Linux PC:

<https://builds.96boards.org/releases/hikey/linaro/binaries/15.06>

- ptable-linux.img
- fastboot1.img
- fastboot2.img
- nvme.img
- mcuimage.bin

You will also need the fastboot application installed on your Linux PC – if this is not installed use the following commands

```
$ sudo apt-get install android-tools-fastboot      On Debian/Ubuntu
$ sudo yum install android-tools                  On Fedora
```

Either create the file: `/etc/udev/rules.d/51-android.rules` with the following content, or append the content to the file if it already exists. You will need to have superuser privileges so use

```
$ sudo vi /etc/udev/rules.d/51-android.rules      or
$ sudo gedit /etc/udev/rules.d/51-android.rules
```

to create and edit the file. Add the following to the file.

```
# fastboot protocol on HiKey
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="d00d", MODE="0660", GROUP="dialout"
# adb protocol on HiKey
SUBSYSTEM=="usb", ATTR{idVendor}=="12d1", ATTR{idProduct}=="1057", MODE="0660", GROUP="dialout"
# rndis for HiKey
SUBSYSTEM=="usb", ATTR{idVendor}=="12d1", ATTR{idProduct}=="1050", MODE="0660", GROUP="dialout"
```

---

You will also need a standard microUSB cable connected between the HiKey microUSB and your Linux PC. Do not power up the HiKey board yet.

### Set Board Link options

For flashing the bootloader (fastboot), the top two links should be installed (closed) and the 3rd link should be removed (open):

Name	Link	State
Auto Power up	Link 1-2	closed
Boot Select	Link 3-4	closed
GPIO3-1	Link 5-6	open

Link 1-2 causes HiKey to auto-power up when power is installed. Link 3-4 causes the HiKey SoC internal ROM to start up in at a special “install bootloader” mode which will install a supplied bootloader from the microUSB OTG port into RAM, and will present itself to a connected PC as a ttyUSB device.

Please refer to the Hardware User Guide (Chapter 1. Settings Jumper) for more information on the HiKey link options.

Connect a standard microUSB to USB connector between the HiKey microUSB port and your Linux PC. Connect the HiKey power supply to the board.

**Note:** USB does NOT power the HiKey board because the power supply requirements in certain use cases can exceed the power supply available on a USB port. You must use an external power supply.

**Note:** The HiKey board will remain in USB load mode for 90 seconds from power up. If you take longer than 90 seconds to start the install then power cycle the board before trying again.

Check that the HiKey board has been recognized by your Linux PC:

```
$ ls /dev/ttyUSB*
```

The following instructions assume that `/dev/ttyUSB0` is the tty port for communication with the HiKey board. Adjust the port for your own tty port.

[hisi-idt.py](#) is the Python download tool for the HiKey. This is used to install the bootloader as follows:

Execute the following commands as a script or individually:

First, get the Python script that is needed to load the initial boot software onto the SoC:

```
$ wget https://raw.githubusercontent.com/96boards/burn-boot/master/hisi-idt.py
```

The script was written for Python 2. Make sure you're not defaulted to Python 3 by typing:

```
$ python --version
```

**Note:** Python 3 currently has a serial library bug, and will fail during data transfer - so if you are using Python 3 then you will need to install and/or change to Python 2.7:

```
$ sudo apt-get install python2.7 python2.7-dev
$ alias python=python2.7
```

Run the script to initially prepare fastboot:

```
$ sudo python hisi-idt.py -d /dev/ttyUSB0 --img1 fastboot1.img --img2 fastboot2.img
```

---

If you get the following error message, while running the hisi-idt.py script:

```
ImportError: No module named serial
```

Then you need to install the python-serial module, on Ubuntu/Debian run:

```
$ sudo apt-get install python-serial
```

or you can use pip install:

```
$ sudo pip install pyserial
```

If you have Python 3 installed, make sure to install with the right version, for instance:

```
$ sudo pip2.7 install pyserial
```

After the python command has been issued you should see the following output:

```
+-----+
Serial:  /dev/ttyUSB0
Image1:  fastboot1.img
Image2:  fastboot2.img
+-----+
```

```
Sending fastboot1.img ...
Done
```

```
Sending fastboot2.img ...
Done
```

Note: You may see the word “failed” before Done. This is under investigation but is not fatal. As long as the “Done” is printed at the end you may proceed.

The bootloader has now been installed into RAM. Wait a few seconds for fastboot to actually load. The following fastboot commands then load the partition table, the bootloaders and other necessary files into the HiKey eMMC flash memory.

```
$ sudo fastboot flash ptable ptable-linux.img
$ sudo fastboot flash fastboot1 fastboot1.img
$ sudo fastboot flash fastboot fastboot2.img
$ sudo fastboot flash nvme nvme.img
$ sudo fastboot flash mcuimage mcuimage.bin
$ sudo fastboot reboot
```

Once this has been completed the bootloader has been installed into eMMC.

Power off the HiKey board by removing the power supply jack.

Next change the link configuration as follows:

1. Remove the 2nd jumper (Boot Select 3-4) so that the HiKey board will boot from the newly installed bootloader in eMMC.
2. Install the 3rd jumper (GPIO3-1 5-6) so that the HiKey board will enter fastboot mode when powered up (if the link is open HiKey will try to boot an OS that is not yet installed).

---

Now power up the HiKey board again.

Check that the HiKey board is detected by your Linux PC:  
You should see the ID of the HiKey board returned

```
$ sudo fastboot devices
0123456789abcdef fastboot
```

Your bootloader has been successfully installed and you are now ready to install the operating system into the eMMC flash memory (see Section 3: Updating the OS, above).

**Note:**

The default installed bootloader is not based on open source code. We expect to make available (and pre-install) a new open source bootloader in the near future. This bootloader will be based on UEFI and include:

- ARM Trusted Firmware
- UEFI with DeviceTree
- Fastboot support
- Optional OPTEE (open source Trusted Execution Environment)

## 5. Hardware Notes

### Schematics and HiKey Board Hardware User Guide

- [Schematics](#)
- [Hardware User Guide](#)

### CPU Load

The supplied Linux 3.18-based kernel supports the thermal protection framework and DVFS. This will cause the HiKey CPU core frequencies to be reduced from the maximum 1.2GHz if the thermal setpoint of the SoC is reached. In an extreme case thermal shutoff will occur if DVFS has not been effective at reducing the SoC temperature to an acceptable level.

Higher performance may be obtained by using forced air (fan) cooling on the HiKey board.

### HDMI Port

At present the HDMI port is fixed to use 1280x720 non-interlaced at 60Hz. We expect a future software update to support EDID and setting of alternate video modes for the display. Note that the fixed settings may not work on all monitors/TVs but have been demonstrated to work on most.

### USB Ports

There are multiple USB ports on the HiKey board:

- One microUSB OTG port on the front edge of the board
- Two Type A USB 2.0 host ports on the front edge of the board

- 
- One USB 2.0 host port on the high-speed expansion bus

Please read the HiKey Board Hardware User Guide for more information on the following hardware restrictions:

1. The microUSB OTG port may be used (in host or slave mode) OR the Type A host ports may be used. They may not both be used simultaneously. If a cable is inserted into the OTG port then the Type A ports and the expansion bus port will be automatically disabled.
2. For the microUSB OTG port a single Low Speed (1.5Mbit/sec), Full Speed (12Mbit/sec) or High Speed (480Mbit/sec) device is supported.
3. For the USB host ports all attached USB devices MUST be one of the following two options:
  - Low Speed (1.5Mbit/sec) and Full Speed (12Mbit/sec) devices, or
  - High Speed devices (480Mbit/sec)

If a mixture of High Speed and Low/Full speed devices are attached the devices will not operate correctly. This also applies if any hubs are attached to the ports.

The reason for this limitation is that USB 2.0 split transfers are not supported by the mobile-targeted SoC hardware USB implementation.

In order to address this limitation the USB ports are by default configured into Low/Full speed operation.

In Debian the `usb_speed` utility (use `-h` option for help) is provided in `/home/linaro/bin` to switch the USB ports between modes (see Other Useful Information in Section 1 above for details on this utility).

In the AOSP build a small application is provided (`usb-speed-switch`) to change between High Speed and Full Speed operation.

## UART Ports

In Debian the two 96Boards expansion IO UART serial ports will appear as `/dev/ttyAMA2` and `/dev/ttyAMA3` and are configured at 9600 baud by default. Note that `/dev/ttyAMA3` requires an updated build from the daily snapshots.

Note that the LS expansion port I/O pins on the 96Boards 2mm header, including the UART signals, are at 1.8V levels.

## 6. Known Issues

The following are known software issues on the current release.

### 1. Not Yet Supported

- HDMI EDID support and video mode switching (see above)
- HDMI and Expansion bus audio. At present only Bluetooth audio is supported (on both Debian and AOSP builds)
- Video playback in Debian. This will be addressed in a future software release
- Some video formats are not decoded in Android, and will not be played with the current release
- Open source bootloader. The current bootloader is not open source and is provided by HiSilicon. An open source implementation of ARM Trusted Firmware, PSCI and UEFI with fastboot support is in development and will be made available in a future software release

- 
- Hardware graphics acceleration (Mali GPU) for OpenGL ES on the Debian build. This will be addressed in a future software release. GPU acceleration is functional in the AOSP build
  - The Bluetooth LED is not functional in the Android build

## 2. USB gives occasional non-fatal kernel trace messages

```
usb usb1: clear tt 1 (9032) error -22
```

This is under current investigation.

## 3. Apple Bluetooth Keyboards/Mice/Trackpads do not work

This is under current investigation.

## 4. Debian build does not handle WiFi misconfiguration

If the WiFi network is misconfigured (for example, you attempt to connect to a non-existent network or a 5GHz network (not supported), or your network password/passphrase is incorrect), then you cannot just fix the problem and proceed with a network down/up operation. The work round is to fix the configuration problem in `/etc/network/interfaces.d` and then to perform a reboot with the changes.

## 5. Thermal Issues

Certain stress tests or heavy CPU load will cause the HiKey to hang due to thermal shutdown. Power and thermal management are still being tuned on the HiKey board, and this will be addressed in the 15.06 releases. In the meantime if you wish to run intensive tasks we recommend using a fan to generate good airflow across the board.

## 6. `/dev/ttyAMA3`

The second expansion bus UART is not implemented. This has been fixed in the daily Snapshot builds and will be implemented in the next release.

## 7. 1.8V Expansion bus power rail

This is not enabled in the 15.05 release. This has been fixed in the daily Snapshot builds and will be implemented in the next release.

## Reporting New Issues

To report new issues with the HiKey please use the [96Boards Bug Tracker](#).

To view the open bugs, click [here](#). Alternatively go to the [site](#), click on the Search menu item, select HiKey in the product dropdown list and click the search button.

To add a new bug, click [here](#) and fill in the form. Alternatively go to the [site](#), click on the New menu item, then click on Consumer Edition Boards, then click on HiKey, and finally fill in the form.

## 7. Building Software from Source Code

To build a kernel using a linux computer use the following instructions. These assume that you have a good level of knowledge in using Linaro tools and building Linux kernels.

The HiKey kernel sources are located at: <https://github.com/96boards/linux>

To build a kernel, make sure you have an AArch64 cross-toolchain installed on your linux computer, and configured to cross compile to ARMv8 code. For example, Linaro GCC 4.9:

```
$ wget http://releases.linaro.org/14.09/components/toolchain/binaries/\
> gcc-linaro-aarch64-linux-gnu-4.9-2014.09_linux.tar.xz
$ mkdir ~/arm64-tc/bin
$ tar --strip-components=1 -C ~/arm64-tc/bin -xf gcc-linaro-aarch64-linux-gnu-4.9-2014.09_linux.tar.xz
$ export PATH=~/arm64-tc/bin:$PATH
```

---

Note: the toolchain binaries are for a 32 bit host system. On Debian/Ubuntu, you should install multiarch-support and enabled i386 architecture. On Fedora, you should install glibc.i686 package.

The following instructions can then be used to build the kernel:

Git clone the source code tree:

```
$ git clone https://github.com/96boards/linux.git
$ git checkout -b working-hikey 96boards-hikey-15.06
```

To build the kernel:

```
$ export LOCALVERSION="-linaro-hikey"

$ make distclean
$ make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- defconfig
$ make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- -j8 Image modules hi6220-hikey.dtb
```

You will need to decide whether you want your kernel to built for internal eMMC usage, or usage on an installed microSD card.

The rootfs included in each hikey release uses a different wifi driver than the one defined in the kernel.config file present in the release page.

<https://builds.96boards.org/snapshots/hikey/linaro/debian/latest>

By default, hikey includes the TI R8.5 wl18 driver (some information below)

[http://processors.wiki.ti.com/index.php/WL18xx\\_System\\_Build\\_Scripts](http://processors.wiki.ti.com/index.php/WL18xx_System_Build_Scripts)

In order to compile and install this driver you will have to do the following:

```
$ git clone https://github.com/96boards/linux linux.git
$ cd linux.git
$ git clone https://github.com/96boards/wilink8-wlan_build-utilites.git build_utilities.git
$ git clone -b hikey https://github.com/96boards/wilink8-wlan_wl18xx.git build_utilities.git/src/driver
$ git clone -b R8.5 https://github.com/96boards/wilink8-wlan_wl18xx_fw.git\
> build_utilities.git/src/fw_download
$ git clone -b hikey https://github.com/96boards/wilink8-wlan_backports.git\
> build_utilities.git/src/backports
$ patch -p1 < build_utilities.git/patches/hikey_patches/\
> 0001-defconfig-hikey-discard-CFG80211-and-MAC80211.patch
```

Then compile the kernel as usual.

Before building the kernel drivers, create a file build\_utilities.git/setup-env using the build\_utilities.git/setup-env.sample as reference.

Please ignore any warnings/errors reported during the following steps

```
$ cd linux.git
$ make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- -j8 modules INSTALL_MOD_PATH=./build_utilities.git/fs\
> modules_install
$ cd build_utilities.git
$ ./build_wl18xx.sh modules
$ ./build_wl18xx.sh firmware
```

Now, all the kernel drivers and kernel firmwares have been installed in build\_utilities.git/fs/lib.

Make sure to remove the following file: lib/firmware/ti-connectivity/wl18xx-conf.bin to avoid the possibility of a format mismatch (the new firmware wl18xx-fw-4.bin uses its internal conf data)

You could now chown root:root the directory, compress it and decompress it in your final target (or an intermediate rootfs if you are generating an image)

---

```
$ cd build_utilities.git/fs/lib
$ rm firmware/ti-connectivity/wl18xx-conf.bin
$ sudo chown -R root:root *
$ sudo tar jcvf fw-modules.tar.bz2 *
```

To include the drivers compiled above in a jessie image you would:

a) install simg2img and make\_ext4fs both from Linaro's modified package 'android-tools-fsutils'

```
$ wget http://repo.linaro.org/ubuntu/linaro-overlay/pool/main/a/android-tools/\
> android-tools-fsutils_4.2.2+git20130218-3ubuntu41+linaro1_amd64.deb
$ sudo dpkg -i --force-all android-tools-fsutils_*.deb
```

b) then do the following

```
$ gzip -d -c hikey_jessie_developer.img.gz > /tmp/jessie.img
$ simg2img /tmp/jessie.img /tmp/raw.img
$ mkdir /tmp/mnt
$ sudo mount /tmp/raw.img /tmp/mnt
$ cd /tmp/mnt/lib/
$ sudo tar xvf fw-modules.tar.bz2
$ cd /tmp/
$ sudo make_ext4fs -o -L rootfs -l 1500M -s jessie.updated.img mnt/
$ sudo umount mnt/
```

Notice that by rebuilding the image file you could also transfer your public ssh keys or private files - like wifi credentials - to the target before booting it.

At this point you would have an image with the required drivers.

## Install onto eMMC

To build the boot image for eMMC:

### Method 1 - Build from scratch

Create a dummy ramdisk for the ramdisk image:

```
$ touch initrd ; echo initrd | cpio -ov > initrd.img
```

Create the boot image:

```
$ echo "console=tty0 console=ttyAMA0,115200n8 root=/dev/disk/by-partlabel/system rootwait rw" > cmdline
$ mkdir boot-fat
$ dd if=/dev/zero of=boot-fat.emmc.img bs=512 count=131072
$ sudo mkfs.fat -n "BOOT IMG" boot-fat.emmc.img
$ sudo mount -o loop,rw,sync boot-fat.emmc.img boot-fat
$ sudo cp -a arch/arm64/boot/Image boot-fat/Image
$ sudo cp arch/arm64/boot/dts/hi6220-hikey.dtb boot-fat/lcb.dtb
$ sudo cp initrd.img boot-fat/ramdisk.img
$ sudo cp cmdline boot-fat/cmdline
$ sudo umount boot-fat
$ rm -rf boot-fat
```

After the above, you can flash the boot-fat.emmc.img to eMMC with the command:

---

```
$ sudo fastboot flash boot boot-fat.emmc.img
$ sudo fastboot reboot
```

## Method 2 - Use an existing boot-fat.emmc.img

```
$ mkdir tmp
$ sudo mount boot-fat.emmc.img tmp
$ sudo cp YOUR-KERNEL-BUILD/arch/arm64/boot/Image tmp/Image
$ sudo cp YOUR-KERNEL-BUILD/arch/arm64/boot/dts/hi6220-hikey.dtb tmp/lcb.dtb
$ sudo umount tmp
$ rm -rf tmp
```

After the above, you can flash the boot-fat.emmc.img to eMMC with the command:

```
$ sudo fastboot flash boot boot-fat.emmc.img
$ sudo fastboot reboot
```

Note: if you make ANY of your own changes to the tagged tree your built kernel will be named 3.18.0-linaro-hikey+ (use `uname -a` to see the kernel name). This means that the installed kernel modules in `/lib/modules` will not work correctly unless you install a new set of kernel modules in `/lib/modules` from your kernel build.

## Install onto the SD card

1. Use the kernel Image and hi6220-hikey.dtb as explained above.
2. Prepare your SD card. See Section 3: Using an SD Card for more information. There will be two partitions on it: `boot` and `rootfs`
3. Insert your SD card into your Linux PC and copy your newly built kernel and device tree blob onto the SD card boot partition - use your own SD card `/dev` device in place of `/dev/sda1`:

```
$ sudo cp arch/arm64/boot/Image /dev/sda1/boot/Image
$ sudo cp arch/arm64/boot/dts/hi6220-hikey.dtb /dev/sda1/boot/lcb.dtb
```

Note: File names must not be changed - Refer to Appendix 1 to see the 4 files that are expected to be in the boot partition. If any of these are missing from the SD card boot partition, HiKey will fall back to the eMMC boot partition and boot from eMMC.

Plug your SD card to HiKey board.

## Source for jessie rootfs build

We pull all the packages from Debian official repository. The only change is the uim package. Sources are available in github at <https://github.com/96boards>

## AOSP Build

AOSP sources are hosted in these repositories:

- [https://github.com/96boards/android\\_hardware\\_ti\\_wpan](https://github.com/96boards/android_hardware_ti_wpan)
- [https://github.com/96boards/android\\_external\\_wpa\\_supplicant\\_8](https://github.com/96boards/android_external_wpa_supplicant_8)

- [https://github.com/96boards/android\\_device\\_linaro\\_hikey](https://github.com/96boards/android_device_linaro_hikey)
- [https://github.com/96boards/android\\_manifest](https://github.com/96boards/android_manifest)

## Build setup

Please setup the host machine by following the instructions here: <http://source.android.com/source/initializing.html>

NOTE: The build tries to mount a loop device as fat partition to create the boot-fat.img filesystem image. Please make sure your user is allowed to run those commands in sudo without password by running “visudo” and appending the following lines (replacing “<USER>” with your username):

```
<USER> ALL= NOPASSWD: /bin/mount
<USER> ALL= NOPASSWD: /bin/umount
<USER> ALL= NOPASSWD: /sbin/mkfs.fat
<USER> ALL= NOPASSWD: /bin/cp
```

Here are the instructions on how to download the code:

```
$ mkdir android/
$ cd android/
```

Download and extract the Mali vendor binaries in the above directory:

<http://builds.96boards.org/snapshots/hikey/linaro/binaries/20150607/vendor.tar.bz2>

```
$ repo init -u https://android.googlesource.com/platform/manifest -b android-5.1.1_r1\
> -g "default,-device,hikey"
$ cd .repo/
$ git clone https://github.com/96boards/android_manifest -b android-5.0 local_manifests
$ cd -
$ repo sync -j8
$ source build/envsetup.sh
$ lunch hikey-userdebug
$ make droidcore -j8
$ cd out/target/product/hikey
```

Install the built files following the instructions on loading the AOSP build in Section 2 above.

## Appendix 1: Partition Information

Table 1 describes the partition layout on the HiKey eMMC.

Name	Partition	Offset	Size
fastboot1	–	0x0000_0000	0x0004_0000 (256KB)
ptable	–	0x0000_0000	0x0010_0000 (1MB)
vr1	1	0x0010_0000	0x0010_0000 (1MB)
vr1_backup	2	0x0020_0000	0x0010_0000 (1MB)
mcuimage	3	0x0030_0000	0x0010_0000 (1MB)
fastboot	4	0x0040_0000	0x0080_0000 (8MB)
nvme	5	0x00C0_0000	0x0020_0000 (2MB)
boot	6	0x00E0_0000	0x0400_0000 (64MB)
Reserved	7	0x04E0_0000	0x1000_0000 (256MB)

---

Name	Partition	Offset	Size
cache	8	0x14E0_0000	0x1000_0000 (256MB)
system	9	0x24E0_0000	0x6000_0000 (1536MB)
userdata	10	0x84E0_0000	0x6000_0000 (1536MB)

Table 1: HiKey Partitions

Note that for the Debian build the system and userdata partitions are merged to create a single system (root file system) partition.

Table 2 describes the binaries located in the boot partition.

File Name	Description	Supported Max. Size
Image	Kernel Image1	16MB
ramdisk.img	Ramdisk Image	8MB
lcb.dtb	Device Tree Binary2	512KB
cmdline	Command line text file	512B

Table 2: boot partition files

Note1: Kernel build image: `arch/arm64/boot/image`

Note2: DTB: `arch/arm64/boot/dts/hi6220-hikey.dtb`