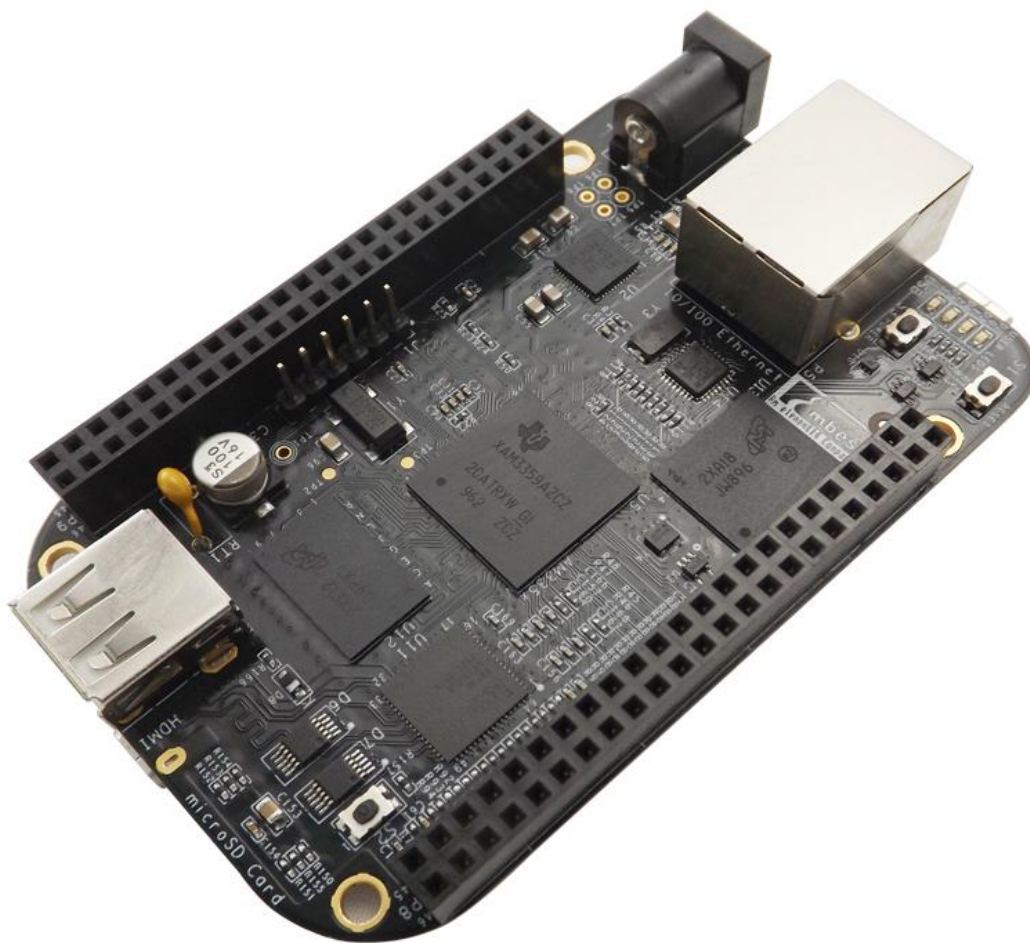


element14

BeagleBone Black



用户手册

版本 2.0– 2014 年 5 月 7 日

版权声明：

- 中国版 **BB-Black** 开发套件及其相关知识产权由深圳市英蓓特科技有限公司所有。
- 本文档由深圳市英蓓特科技有限公司版权所有，并保留一切权利。在未经英蓓特公司书面许可的情况下，不得以任何方式或形式来修改、分发或复制本文档的任何部分。
- Microsoft, MS-DOS, Windows, Windows95, Windows98, Windows2000, Windows xp, Windows Embedded Compact 7 由微软公司授权使用。

版本更新记录：

版本	更新日期	描述
1.0	2013-4-22	初始版本
1.1	2013-5-20	文档修订
1.2	2013-9-27	文档修订
1.3	2014-3-20	文档修订
2.0	2014-5-7	硬件升级为版本 C, 默认映象改为 Debian

目录

第 1 章 产品概述.....	1
1.1 产品介绍.....	1
1.2 包装内容.....	1
1.3 硬件特性.....	2
1.4 系统框图.....	4
1.5 硬件尺寸（毫米）	4
第 2 章 硬件系统介绍	5
2.1 CPU 简介	5
2.1.1 时钟信号	5
2.1.2 复位信号	5
2.1.3 通用接口	6
2.1.4 可编程实时单元和工业通讯子系统	6
2.1.5 3D 图形引擎	6
2.2 CPU 周边芯片	6
2.2.1 TPS65217 电源管理芯片	6
2.2.2 KE4CN2H5A eMMC Flash 存储器	7
2.2.3 MT41K256M16HA-125 DDR 存储器.....	7
2.2.4 LAN8710A-EZC-TR Ethernet 芯片	7
2.2.5 TDA19988 发送器	7
2.3 接口/按钮/LED	8
2.3.1 电源接口（P1）	8
2.3.2 JTAG 接口（P2）	8
2.3.3 USB Host 接口（P3）	9
2.3.4 USB Device 接口（P4）	9
2.3.5 LAN 接口（P5）	10
2.3.6 HDMI D Type 接口（P6）	10
2.3.7 调试串口（J1）	11
2.3.8 右扩展接口（P8）	11
2.3.9 左扩展接口 2（P9）	13
2.3.10 按钮	14
2.3.11 LED 指示灯	14

第 3 章 系统更新.....	15
3.1 软件系统介绍	15
3.2 更新前的准备	15
3.3 开始更新系统	16
第 4 章 Angstrom 系统开发.....	17
4.1 获取源代码.....	17
4.2 编译系统.....	18
4.3 映像更新.....	19
4.3.1 准备映像	19
4.3.2 开始更新	20
第 5 章 Debian 系统开发.....	21
5.1 获取源代码.....	21
5.2 编译系统.....	22
5.3 映像更新.....	23
5.3.3 准备映像	23
5.3.4 开始更新	24
技术支持和保修服务	25

第1章 产品概述

1.1 产品介绍

element BeagleBone Black 是深圳市英蓓特科技有限公司推出的一款基于 AM3358 处理器的开发套件。处理器集成了高达 1GHz 的 ARM Cortex™-A8 内核，并提供了丰富的外设接口。element14 BeagleBone Black 的扩展接口包括网口、USB Host、USB OTG、TF 卡接口、串口、JTAG 接口（默认不焊）、HDMI D Type 接口、eMMC、ADC、I2C、SPI、PWM 和 LCD 接口。

element14 BeagleBone Black 的应用场景非常广泛，能够满足包括游戏外设、家庭和工业自动化、消费类医疗器械、打印机、智能收费系统、智能售货机称重系统、教育终端和高级玩具等在内的各个领域的不同需求。

1.2 包装内容

- element14 BeagleBone Black 评估板×1
- Mini USB 线缆×1

1.3 硬件特性

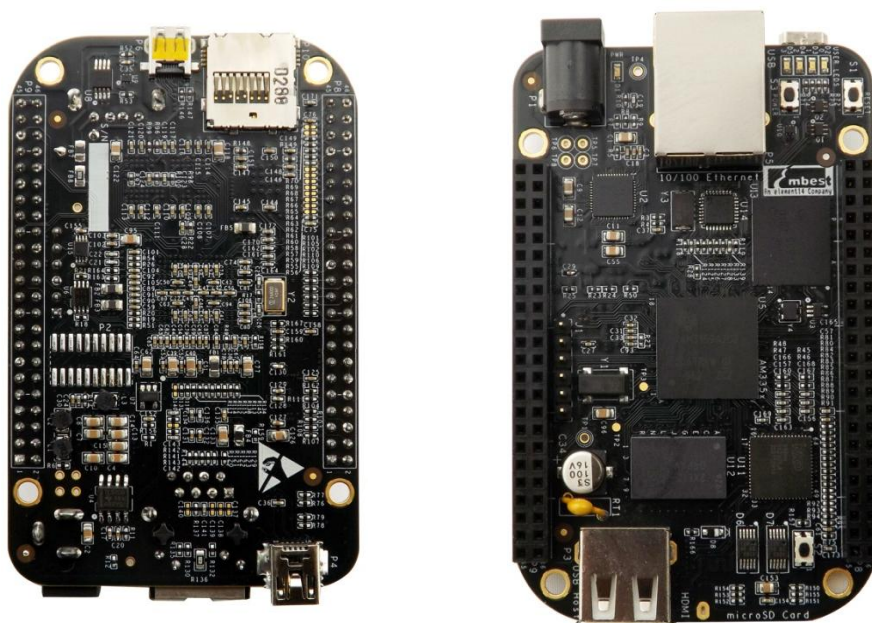


图 1-1 产品背面（左）和正面（右）

- **产品参数:**
 - 产品尺寸: 86.36 mm×54.61mm
 - 工作温度: 0~70℃
 - 环境湿度: 20% ~ 90% (无凝结)
 - 输入电源: 5V
- **AM3358 处理器:**
 - 集成 1GHz ARM Cortex™-A8 内核
 - 集成 NEON™ SIMD 协处理器
 - 集成 SGX530 图形引擎
 - 集成可编程实时单元子系统
 - 集成 32KB 指令缓存和 32KB 数据缓存, 支持奇偶校验
 - 集成 256KB 二级缓存, 支持错误校验码
- **板载存储器:**
 - 4GB eMMC Flash 存储器

- 512MB DDR3 SDRAM 存储器
- **板载接口:**
 - 一个 HDMI D type 接口 (16 位色输出, 支持音频输出)
 - 一个 10/100M 以太网接口 (RJ45 连接器)
 - 一个集成了 PHY 的高速 USB 2.0 OTG 接口 (Mini USB B 型连接器)
 - 一个集成了 PHY 的高速 USB 2.0 HOST 接口 (USB A 型连接器)
 - 一个 TF 卡接口 (兼容 SD/MMC)
 - 一个 3 线调试串口 (6-pin 2.54 间距连接器)
 - 一个 HDMI D type 接口
 - 两个扩展接口, 可扩展 LCD、UART、eMMC、ADC、I2C、SPI 和 PWM 等接口
 - 一个 JTAG 接口 (20-pin 标准接口, 默认未焊接连接器)
- **按钮和 LED:**
 - 一个启动选择按键
 - 一个电源按键
 - 一个复位按键
 - 一个 LED 电源指示灯
 - 四个用户自定 LED 灯

1.4 系统框图

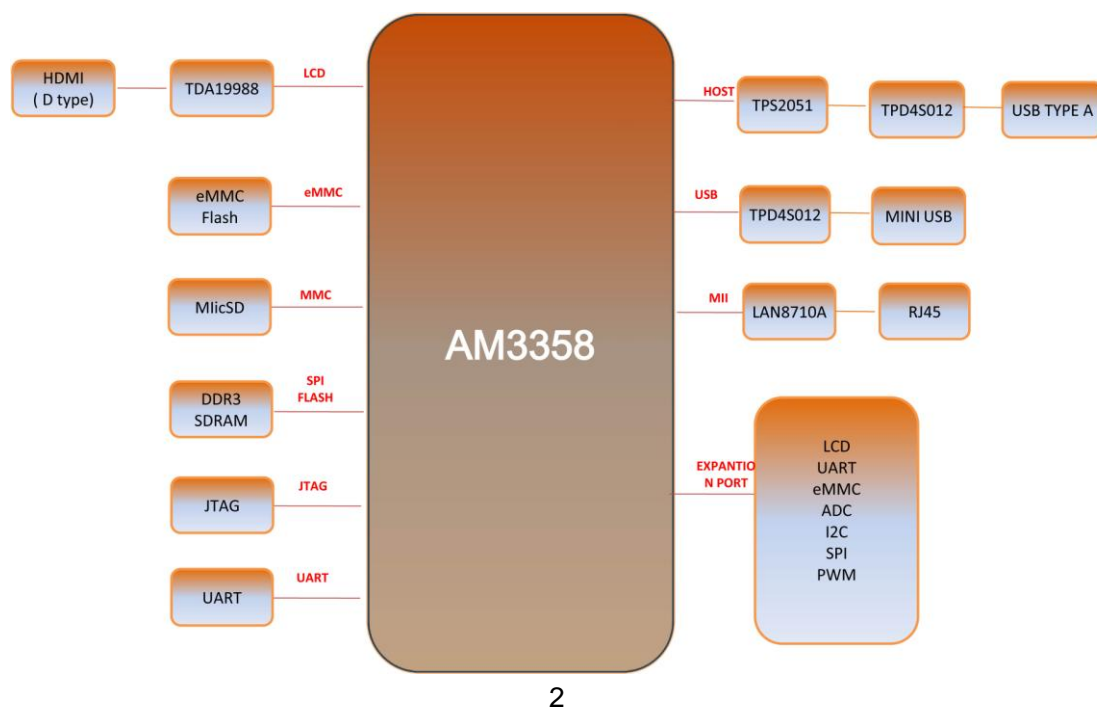


图 1-2 element14 BeagleBone Black 系统框图

1.5 硬件尺寸（毫米）

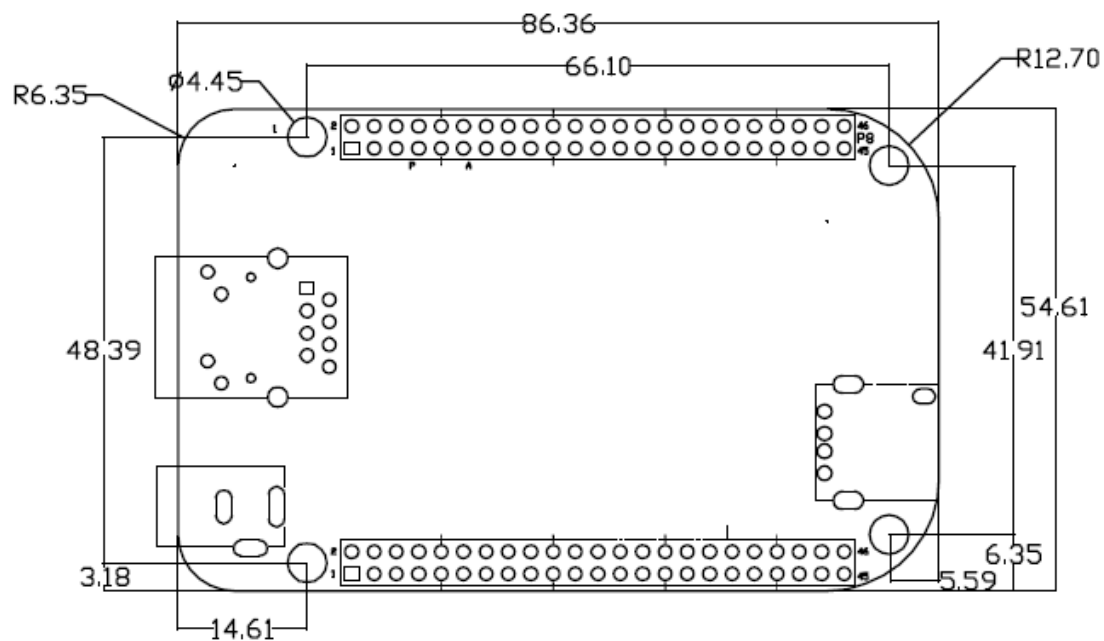


图 1-3 element14 BeagleBone Black 硬件尺寸

第2章 硬件系统介绍

本章节将通过简要介绍 element14 BeagleBone Black 上所采用的 CPU、CPU 周边芯片、以及各种板载接口的引脚定义来让您对该硬件系统有一个大致的了解。

2.1 CPU 简介

AM3358 是基于 ARM Cortex-A8 内核的微处理器，在图像、图形处理、外设和诸如 etherCAT 和 PROFIBUS 的工业接口选项方面进行了增强，并支持 Linux、WinCE 和 Android 等高级操作系统。

该处理器包含了多个子系统。微处理器单元（MPU）子系统基于 ARM Cortex-A8 微处理器；POWERVR SGX 图形加速子系统用于 3D 图形加速以支持显示和游戏效果；可编程实时单元子系统（PRUSS）使用户可以创建各种超越本地外设的数字资源。此外，PRUSS 独立于 ARM 内核，这就允许设备有独立的操作和时钟，从而在复杂系统解决方案中有更大的灵活性。

2.1.1 时钟信号

AM3358 的时钟信号包括两个输入时钟—OSC1 和 OCC0，和两个输出时钟—LCKOUT1 和 LCKOUT2，其中：

- **OSC1**：为 RTC 提供 32.768KHz 参考时钟并用于连接 RTC_XTALIN 和 RTC_XTALOUT 终端。
- **OCC0**：为所有无 RT 功能的时钟提供 19.2MHz、24MHz、25MHz 或 26MHz 参考时钟，并用于连接 XTALIN 和 XTALOUT 终端。

2.1.2 复位信号

复位功能由 CPU 的 PWRONRSTn 信号实现；低电平有效。

2.1.3 通用接口

通用接口包括 4 组通用输入输出接口（GPIO），每一组 GPIO 模组提供 32 个专用的通用接口输入输出管脚，因此通用的 GPIO 可以高达 128 个（4x32）管脚。

2.1.4 可编程实时单元和工业通讯子系统

可编程实时单元和工业通讯子系统（PRU-ICSS）包含了两个 32 位 RISC 内核（可编程实时单元，即 PRUs）、存储器、终端控制器以及能够支持更多周边接口和协议的内部外设。

与 AM1x 和 OMAP-L13x 系列处理器相比，AM3358 拥有的这种子系统是下一代的 PRU 子系统。

2.1.5 3D 图形引擎

POWERVR® SGX 图形加速器子系统用于 3D 图形加速以支持显示和游戏效果，该子系统的主要特性如下：

- Tile-Based 架构，处理能力高达 20Mpoly/秒
- 通用可扩展渲染引擎是一个具有像素和顶点渲染功能的多线程引擎
- 超过 Microsoft VS3.0、PS3.0 和 OGL2.0 的高级渲染功能指令集
- 工业标准 API，支持 Direct3D Mobile、OGL-ES 1.1 和 2.0、OpenVG 1.0 和 OpenMax

2.2 CPU 周边芯片

2.2.1 TPS65217 电源管理芯片

TPS65217 是一个综合的电源管理芯片，它包括 3 个降压转换器、一个 LED 驱动器、四格 LDO 和电池充电单元，并通过 IIC 与 CPU 相连。TPS65217 的主要作用是为 CPU、eMMC 和 DDR 等板载芯片提供 1.1V、1.2V、1.5V、1.8V 或 3.3V 等电压，以保证芯片正常工作。

2.2.2 KE4CN2H5A eMMC Flash 存储器

KE4CN2H5A 是 element14 BeagleBone Black 的 eMMC Flash，大小为 4GB。

2.2.3 MT41K256M16HA-125 DDR 存储器

MT41K256M16HA-125 是 element14 BeagleBone Black 的 DDR3 SDRAM 存储器，大小为 512MB，由 1 片 16bit 的 MT41K256M16HA-125 芯片构成。

2.2.4 LAN8710A-EZC-TR Ethernet 芯片

LAN8710A-EZC-TR 是一个低功耗的 10BASE-T/100BASE-TX 物理层以太网收发器。兼容 IEEE 802.3-2005 标准，支持 1.6V~3.6V 之间的可变 I/O 电压。该芯片支持物理层自动协商机制，因此 element14 BeagleBone Black 可通过直通网线或者交叉网线连接到网络 hub 或者电脑。

如果您希望了解更多关于此芯片的信息，请打开 Disk-BeagleBone Black\HW design\datasheet\LAN\LAN8710A.pdf 文档。

2.2.5 TDA19988 发送器

TDA19988 是 NXP 推出的小尺寸、低功耗的 HDMI 发送器，支持 HDMI1.4a 标准。TDA19988 能够将 RGB 信号转换成 HDMI 信号，并且后向兼容 DVI1.0 标准。

2.3 接口/按钮/LED

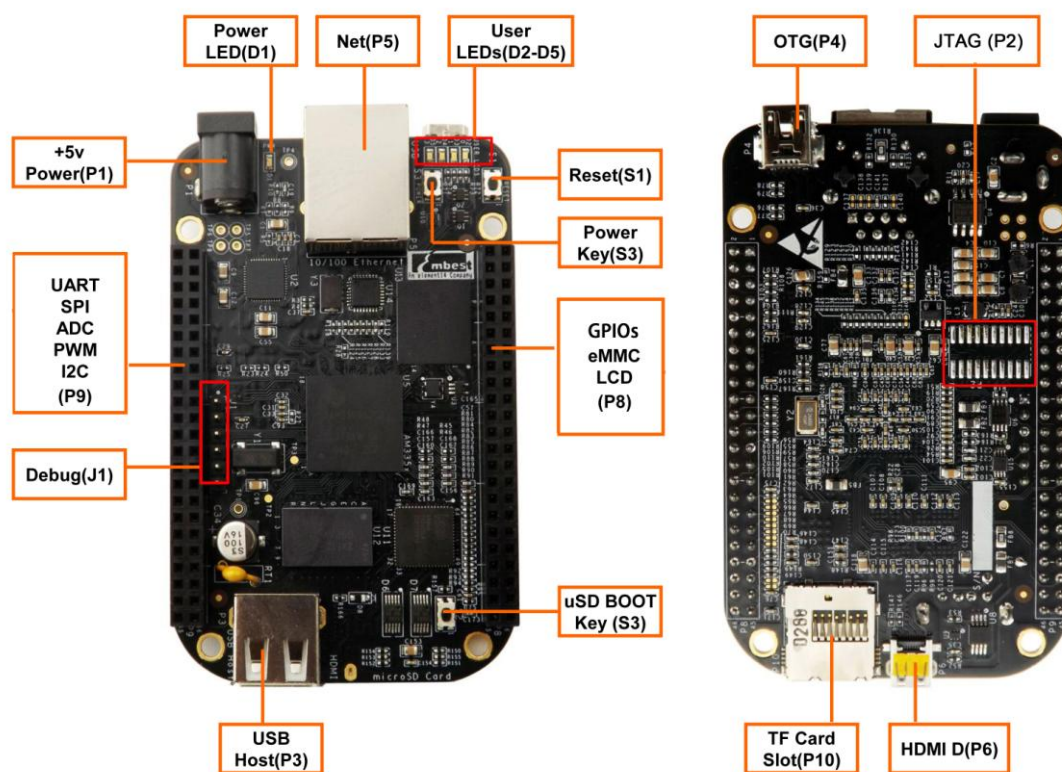


图 2-4 接口/按钮/LED

2.3.1 电源接口（P1）

表 2-1 电源接口

引脚	信号定义	描述
1	GND	GND
2	+5V	Power supply (+5V)
3	NC	NC

2.3.2 JTAG 接口（P2）

表 2-2 JTAG 接口

引脚	信号定义	描述
1	TMS	Test mode select
2	NTRST	Test system reset

引脚	信号定义	描述
3	TDI	Test data input
4	GND	GND
5	VIO	3.3V
6	NC	NC
7	TDO	Test data output
8	GND	GND
9	RTCK	Receive test clock
10	GND	GND
11	TCK	Test clock
12	GND	GND
13	EMU0	Test emulation 0
14	EMU1	Test emulation 1
15	SRST	System reset
16	GND	GND
17	EMU2	Test emulation 2
18	EMU3	Test emulation 3
19	EMU4	Test emulation 4
20	GND	GND

2.3.3 USB Host 接口（P3）

表 2-3 USB Host 接口

引脚	信号定义	描述
1	VB	+5V
2	D-	USB data-
3	D+	USB data+
4	GND	GND

2.3.4 USB Device 接口（P4）

表 2-4 USB Device 接口

引脚	信号定义	描述
1	VB	+5V
2	D-	USB data-
3	D+	USB data+
4	ID	USB ID

引脚	信号定义	描述
5	G1	GND

2.3.5 LAN 接口（P5）

表 2-5 LAN 接口

引脚	信号定义	描述
1	RD+	Receive data+
2	RD-	Receive data-
3	TD+	Transmit data+
4	RCT	Receive common terminal
5	TCT	Transmit common terminal
6	TD-	Transmit data-
7	NC	No connect
8	GND	GND
9	GRNA	Green LED backlight anode
10	GRNC	Green LED backlight catcode
11	YELC	Yellow LED backlight catcode
12	YELA	Yellow LED backlight anode
13	SHD1	Shield
14	SHD2	Shield

2.3.6 HDMI D Type 接口（P6）

表 2-6 HDMI D Type 接口

引脚	信号定义	描述
1	HPLG	Hot plug and play detect
2	NC	No connect
3	HDMI_TX2+	TMDS data 2+
4	DAT2_S	GND
5	HDMI_TX2-	TMDS data 2-
6	HDMI_TX1+	TMDS data 1+
7	DAT1_S	GND
8	HDMI_TX1-	TMDS data 1-
9	HDMI_TX0+	TMDS data 0+
10	DAT0_S	GND
11	HDMI_TX0-	TMDS data 0-

引脚	信号定义	描述
12	HDMI_CLK+	TMDS data clock+
13	CLK_S	GND
14	HDMI_CLK-	TMDS data clock-
15	HDMICONN_CEC	Consumer electronics control
16	DDC/CEC GND	GND
17	HDMI_I2CSCL	IIC master serial clock
18	HDMI_I2CSDA	IIC serial bidirectional data
19	5V	5V

2.3.7 调试串口（J1）

表 2-7 调试串口

引脚	信号定义	描述
1	DGND	GND
2	NC	NC
3	NC	NC
4	UART0_RX	UART0 receive data
5	UART0_TX	UART0 transit data

2.3.8 右扩展接口（P8）

表 2-8 扩展接口 1

引脚	信号定义	描述
1	GND	GND
2	GND	GND
3	mmc1_dat6	mmc1 data6
4	mmc1_dat7	mmc1 data7
5	mmc1_dat2	mmc1 data2
6	mmc1_dat3	mmc1 data3
7	TIMER4	Time interrupt input
8	TIMER7	Time interrupt input
9	TIMER5	Time interrupt input
10	TIMER6	Time interrupt input
11	GPIO1_13	GPIO
12	GPIO1_12	GPIO
13	EHRPWM2B	eHRPWM output
14	GPIO0_26	GPIO

引脚	信号定义	描述
15	GPIO1_15	GPIO
16	GPIO1_14	GPIO
17	GPIO0_27	GPIO
18	GPIO2_1	GPIO
19	EHRPWM2A	eHRPWM output
20	mmc1_dat5	mmc1 data5
21	mmc1_dat4	mmc1 data4
22	mmc1_dat5	mmc1 data5
23	mmc1_dat4	mmc1 data4
24	mmc1_dat1	mmc1 data1
25	mmc1_dat0	mmc1 data0
26	GPIO1_29	GPIO
27	LCD_VSYNC	LCD vertical sync
28	LCD_PCLK	LCD pixel clock
29	LCD_HSYNC	LCD horizontal sync
30	LCD_DE	LCD AC bias enable chip select
31	LCD_DATA14	LCD data14
32	LCD_DATA15	LCD data15
33	LCD_DATA13	LCD data13
34	LCD_DATA17	LCD data17
35	LCD_DATA12	LCD data12
36	LCD_DATA16	LCD data16
37	LCD_DATA8	LCD data8
38	LCD_DATA9	LCD data9
39	LCD_DATA6	LCD data6
40	LCD_DATA7	LCD data7
41	LCD_DATA4	LCD data4
42	LCD_DATA5	LCD data5
43	LCD_DATA2	LCD data2
44	LCD_DATA3	LCD data03
45	LCD_DATA0	LCD data0
46	LCD_DATA1	LCD data1

2.3.9 左扩展接口 2 (P9)

表 2-9 扩展接口 2

引脚	信号定义	描述
1	DGND	GND
2	DGND	GND
3	VDD_3V3B	3.3V power
4	VDD_3V3B	3.3V power
5	VDD_5V	5V power
6	VDD_5V	5V power
7	SYS_5V	5V power
8	SYS_5V	5V power
9	PWR_BUT	Push-button monitor input
10	SYS_RESETn	System reset input
11	UART4_RXD	UART receive data
12	GPIO1_28	GPIO
13	UART4_TXD	UART transit data
14	EHRPWM1A	eHRPWMoutput
15	GPIO1_16	GPIO
16	EHRPWM1B	eHRPWM output
17	I2C1_SCL	I2C clock
18	I2C1_SDA	I2C data
19	I2C2_SCL	I2C clock
20	I2C2_SDA	I2C data
21	UART2_TXD	UART transit data
22	UART2_RXD	UART receive data
23	GPIO1_17	GPIO
24	UART1_TXD	UART transit data
25	GPIO3_21	GPIO
26	UART1_RXD	UART receive data
27	GPIO3_19	GPIO
28	SPI1_CS0	SPI chip select
29	SPI1_D0	SPI data0
30	SPI1_D1	SPI data1
31	SPI1_SCLK	SPI clock
32	VDD_ADC	ADC power
33	AIN4	ADC input
34	AGND	ADC GND
35	AIN6	ADC input
36	AIN5	ADC input

引脚	信号定义	描述
37	AIN2	ADC input
38	AIN3	ADC input
39	AIN0	ADC input
40	AIN1	ADC input
41	CLKOUT2	Audio input master clock
42	GPIO0_7	Interrupt output
43	DGND	GND
44	DGND	GND
45	DGND	GND
46	DGND	GND

2.3.10 按钮

表 2-10 按钮

按钮	信号定义	描述
S1	Reset	重启按键
S2	uSD BOOT	启动方式选择
S3	Power	在系统运行时长按该按钮来关闭系统

2.3.11 LED 指示灯

表 2-11 LED 指示灯

LED	信号定义	描述
S1	Power LED	电源指示灯
S2	User LED	用户自定义 LED 灯
S3	User LED	用户自定义 LED 灯
S4	User LED	用户自定义 LED 灯
S5	User LED	用户自定义 LED 灯

第3章 系统更新

本章节将介绍如何下载最新的映像文件，并通过 TF 卡更新 element14 BeagleBone Black 的操作系统。

3.1 软件系统介绍

以下表格列出了将要使用的 Linux 内核的版本，以及各种设备的驱动程序。

表 3-12 操作系统和驱动程序

	类型	描述
OS	Linux	版本 3.8 或更高
Device Driver	Serial	串行接口驱动
	RTC	硬件时钟驱动
	Net	10/100 IEEE1588 以太网驱动
	Display	DVI 模式驱动
	mmc/sd	SD 2.0 接口和 eMMC 驱动
	USB	高速 USB 接口驱动（一个 OTG 一个 Host）
	LED	用户 LED 驱动

注意：

更多软件资源会不定期的发布在 <http://beagleboard.org/Products/BeagleBone%20Black>；请随时关注该网站的最新内容。

3.2 更新前的准备

在更新系统前，请按照以下步骤下载最新的映像文件、压缩工具和 SD 卡烧写工具。

表 3-13 准备工作

步骤	操作	地址或备注
1	下载映像文件	http://beagleboard.org/latest-images
2	下载和安装压缩工具	http://sourceforge.net/projects/sevenzip
3*	下载和安装 SD 卡烧写工具	http://sourceforge.net/projects/win32diskimager
4	MicroSD 卡	不小于 4G

注意：

📖 element14 BeagleBone Black 出厂默认的系统版本是 BBB-eMMC-flasher-debian-7.4-2014-04-23-2gb.img.xz。如果您在使用最新版本的映像时出现问题，请从[英蓓特官网](#)找到并下载默认的系统版本。

📖 [英蓓特官网](#)提供已经下载好的用于 element14 BeagleBone Black 工具和映像，但不保证是最新版本。

*访问 <https://help.ubuntu.com/community/Installation/FromImgFiles> 可以找到关于 SD 卡烧写的帮助信息。

3.3 开始更新系统

准备工作完成后，请按照以下步骤进行系统更新。

- 1) 使用 7-zip 压缩工具解压刚才下载的映像文件；
- 2) 用一个 MicroSD 卡卡套或者 USB 读卡器将一张 MicroSD 卡连接到 PC；
- 3) 使用刚才安装的 Win32 Disk Imager 工具将解压后映像文件写入 MicroSD 卡；
- 4) 将 MicroSD 卡插入 element14 BeagleBone Black 的卡槽中，然后按住 uSD BOOT 按钮（请参考第 8 页上的图 2-4 接口/按钮/LED）并使用 USB 线缆接通电源；
- 5) 当 4 个 User LED 同时持续亮起时，表示已成功完成系统更新（需耗时 45 分钟左右）。请断开电源并取出 MicroSD 卡，然后再次接通电源即可。

第4章 Angstrom 系统开发

本章将通过获取 bootloader 和内核源代码以及编译系统两部分来简要介绍 Linux 环境下的系统开发大致流程。

注意:

📖 每一条指令前都加上了符号“•”，以便防止由于指令较长占用多行而造成误解。

4.1 获取源代码

Bootloader 和内核源代码保存在 git 中，下载地址分别为 [git://git.denx.de/u-boot.git](https://git.denx.de/u-boot.git) 和 [git://github.com/beagleboard/kernel.git](https://github.com/beagleboard/kernel.git)。u-boot 的 git 代码 clone 下来后需要打补丁，在 <http://www.embest-tech.cn/product/pinggubanxilie/beaglebone-black.html> 下载补丁文件 0001-change-arguments-for-boot-with-beagleboneblack.patch 到用户主目录(即 \$HOME 目录)，请在 Linux 环境下执行以下命令来获取源代码以及打补丁。

1) 获取 bootloader 源代码及打补丁;

- `$ cd ~`
- `$ git clone git://git.denx.de/u-boot.git`
- `$ cd ~/u-boot`
- `$ git checkout v2013.04`
- `$ patch -p1 < ~/0001-change-arguments-for-boot-with-beagleboneblack.patch`

2) 获取内核源代码;

- `$ cd ~`
- `$ git clone git://github.com/beagleboard/kernel.git`
- `$ cd ~/kernel`
- `$ git checkout origin/3.8 -b 3.8`
- `$./patch.sh`
- `wget`
`http://arago-project.org/git/projects/?p=am33x-cm3.git;a=blob_plain;f=bin/am`

```
335x-pm-firmware.bin\;hb=HEAD -O kernel/firmware/am335x-pm-firmware.bin
```

获取源代码的过程用时较长，替代方法为使用[英蓓特科技官网](#)上提供的 Bootloader 和内核源代码（不保证为最新版本），下载源码 `bb-black-u-boot.tar.bz2` 和 `bb-black-kernel-3.8.tar.bz2` 到用户主目录(即\$HOME 目录)再按照以下命令操作。

- 1) 解压 u-boot 源码；
 - `$ cd ~`
 - `$ tar xvf bb-black-u-boot.tar.bz2`
- 2) 解压内核源代码；
 - `$ cd ~`
 - `$ tar xvf bb-black-kernel-3.8.tar.bz2`

4.2 编译系统

编译系统的工作大致包括下载和安装 ARM 交叉编译工具、设置环境变量和编译源代码几个步骤。

- 1) 执行以下命令来下载和安装交叉编译工具；
 - `$ cd ~`
 - `$ wget https://sourcery.mentor.com/sgpp/portal/package7851/public/arm-none-linux-gnueabi/arm-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2`
 - `$ cd ~`
 - `$ tar jxvf arm-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2`
 - `$ export PATH=~:/arm-2010.09/bin:$PATH`

注意：

 如果 `wget` 下载失败，请通过浏览器进行下载。也可以在[英蓓特科技官网](#)下载 `arm-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2` 到用户主目录再继续操作。

- 2) 执行以下命令来编译 u-boot；
 - `$ cd ~/u-boot`
 - `$ [-d ./am335x] && rm -rf ./am335x`

- `$ make O=am335x CROSS_COMPILE=arm-none-linux-gnueabi- ARCH=arm am335x_evm`
- 3) 执行以下命令来为 mkimage 添加环境变量;
- `$ export PATH=~/.u-boot/am335x/tools:$PATH`
- 4) 执行以下命令来编译内核;
- `$ cd ~/kernel/kernel`
 - `$ cp ../configs/beaglebone .config`
 - `$ make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- ulmage dtbs`

注意:

如果编译内核过程中出现/bin/sh: lzop: command not found 的错误, 在 Ubuntu 系统下, 使用命令 `sudo apt-get install lzop` 安装 lzop 包。

4.3 映像更新

u-boot 和内核编译完成后, 按照以下步骤运行新编译的映像。这里以 U 盘为媒介传送新映像到 BB-Black。

4.3.1 准备映像

在 PC 机的 Linux 中运行以下命令。

- `$ mkdir ~/images`
准备 u-boot 映像
- `$ cd ~/.u-boot`
- `$ cp am335x/MLO ~/images`
- `$ cp am335x/u-boot.img ~/images`
准备内核映像和设备描述文件
- `$ cd ~/kernel/kernel`
- `$ cp arch/arm/boot/ulmage ~/images`
- `$ cp arch/arm/boot/dts/am335x-boneblack.dtb ~/images`
生成内核模块压缩包
- `$ mkdir ~/images/rootfs`

- `$ cd ~/kernel/kernel`
- `$ make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- modules`
- `$ make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- modules_install INSTALL_MOD_PATH=$HOME/images/rootfs`
- `$ cd ~/images/rootfs`
- `$ tar -czvf ../kernel_modules.tar.gz .`
- `$ cd ~/images/`
- `$ rm -rf rootfs`
- # \$HOME/images 下共五个文件
- `$ cd ~/images/`
- `$ ls`
- `am335x-boneblack.dtb kernel_modules.tar.gz MLO u-boot.img ulmage`

现在，发送所有~/images/下的映像到 U 盘，常见的单 FAT32 分区的 U 盘即可。

4.3.2 开始更新

- 1) 在 BB-Black 上准备好映像

连接上节中的 U 盘到正在运行的 BB-Black。


- 2) 在 BB-Black 的终端(putty 或串口)中使用如下命令安装映像

- # 挂载 U 盘和 eMMC 第一分区
- `$ mkdir /media/sda1`
- `$ mount /dev/sda1 /media/sda1`
- `$ mkdir /media/mmcblk0p1`
- `$ mount /dev/mmcblk0p1 /media/mmcblk0p1`
- # 拷贝(替换)文件到合适的路径
- `$ cp -f /media/sda1/MLO /media/mmcblk0p1/`
- `$ cp -f /media/sda1/u-boot.img /media/mmcblk0p1/`
- `$ cp -f /media/sda1/ulmage /boot/`
- `$ cp -f /media/sda1/am335x-boneblack.dtb /boot/`
- `$ tar -xvf /media/sda1/kernel_modules.tar.gz -C /`
- # 同步文件，确保写入 eMMC
- `$ sync`
- `$ reboot`

第5章 Debian 系统开发

本章将通过获取 bootloader 和内核源代码以及编译系统两部分来简要介绍 Debian 环境下的系统开发大致流程。

注意：

 每一条指令前都加上了符号“•”，以便防止由于指令较长占用多行而造成误解。

5.1 获取源代码

Bootloader 和内核源代码保存在 git 中，下载地址分别为 [git://git.denx.de/u-boot.git](https://git.denx.de/u-boot.git) 和 [git://github.com/beagleboard/kernel.git](https://github.com/beagleboard/kernel.git)。注意：u-boot 的 git 代码 clone 下来后需要打补丁。

1) 获取 bootloader 源代码及打补丁；

- `$ cd ~`
- `$ git clone git://git.denx.de/u-boot.git`
- `$ cd ~/u-boot`
- `$ git checkout v2014.04-rc3 -b tmp`
- `$ wget`
`https://raw.githubusercontent.com/eewiki/u-boot-patches/master/v2014.04-rc3/0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch`
- `$ patch -p1 < 0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch`

2) 获取内核源代码；

- `$ cd ~`
- `$ git clone git://github.com/beagleboard/kernel.git`
- `$ cd ~/kernel`
- `$ git checkout origin/3.8 -b 3.8`
- `$./patch.sh`
- `wget`
`http://arago-project.org/git/projects/?p=am33x-cm3.git;a=blob_plain;f=bin/am335x-pm-firmware.bin;hb=HEAD -O kernel/firmware/am335x-pm-firmware.bin`

获取源代码的过程用时较长，替代方法为使用[英蓓特科技官网](#)上提供的 Bootloader 和内核源代码（不保证为最新版本），下载源码 `bb-black-debian-u-boot.tar.bz2` 和 `bb-black-debian-kernel-3.8.tar.bz2` 到用户主目录(即\$HOME 目录)再按照以下命令操作。

1) 解压 u-boot 源码；

- `$ cd ~`
- `$ tar xvf bb-black-debian-u-boot.tar.bz2`
- `$ tar xvf bb-black-debian-kernel-3.8.tar.bz2`

5.2 编译系统

编译系统的工作大致包括下载和安装 ARM 交叉编译工具、设置环境变量和编译源代码几个步骤。

1) 执行以下命令来下载和安装交叉编译工具；

- `$ cd ~`
- `$ wget -c https://releases.linaro.org/14.03/components/toolchain/binaries/gcc-linaro-arm-linux-gnueabihf-4.8-2014.03_linux.tar.xz`
- `$ cd ~`
- `$ tar xf gcc-linaro-arm-linux-gnueabihf-4.8-2014.03_linux.tar.xz`
- `$ export CC=`pwd`/gcc-linaro-arm-linux-gnueabihf-4.8-2014.03_linux/bin/arm-linux-gnueabihf-`

注意：

📖 如果 wget 下载失败，请通过浏览器进行下载。也可以在[英蓓特科技官网](#)下载 `gcc-linaro-arm-linux-gnueabihf-4.8-2014.03_linux.tar.xz` 到用户主目录再继续操作。

2) 执行以下命令来编译 u-boot；

- `$ cd ~/u-boot`
- `$ make ARCH=arm CROSS_COMPILE=${CC} distclean`
- `$ make ARCH=arm CROSS_COMPILE=${CC} am335x_evm_config`
- `$ make ARCH=arm CROSS_COMPILE=${CC}`

3) 执行以下命令来编译内核；

- `$ cd ~/kernel/kernel`
- `$ cp ../configs/beaglebone .config`
- `$ make ARCH=arm CROSS_COMPILE=${CC} zImage dtbs`

注意：

如果编译内核过程中出现/bin/sh: lzop: command not found 的错误，在 Ubuntu 系统下，使用命令 `sudo apt-get install lzop` 安装 lzop 包。

5.3 映像更新

u-boot 和内核编译完成后，按照以下步骤运行新编译的映像。这里以 U 盘为媒介传送新映像到 BB-Black。

5.3.3 准备映像

在 PC 机的 Linux 中运行以下命令。

- `$ mkdir ~/images`
准备 u-boot 映像
- `$ cd ~/u-boot`
- `$ cp MLO ~/images`
- `$ cp u-boot.img ~/images`
准备内核映像和设备描述文件
- `$ cd ~/kernel/kernel`
- `$ cp arch/arm/boot/zImage ~/images`
- `$ cp arch/arm/boot/dts/am335x-boneblack.dtb ~/images`
生成内核模块压缩包
- `$ mkdir ~/images/rootfs`
- `$ cd ~/kernel/kernel`
- `$ make ARCH=arm CROSS_COMPILE=${CC} modules`
- `$ make ARCH=arm CROSS_COMPILE=${CC} modules_install`
`INSTALL_MOD_PATH=$HOME/images/rootfs`

- `$ cd ~/images/rootfs`
- `$ tar -czvf ../kernel_modules.tar.gz ./`
- `$ cd ~/images/`
- `$ rm -rf rootfs`
\$HOME/images 下共五个文件
- `$ cd ~/images/`
- `$ ls`
- `am335x-boneblack.dtb kernel_modules.tar.gz MLO u-boot.img zImage`

现在，发送所有~/images/下的映像到 U 盘，常见的单 FAT32 分区的 U 盘即可。

5.3.4 开始更新

- 1) 在 BB-Black 上准备好映像

连接上节中的 U 盘到正在运行的 BB-Black。

- 2) 在 BB-Black 的终端(putty 或串口)中使用如下命令安装映像

挂载 U 盘和 eMMC 第一分区

- `$ mkdir /media/sda1`
- `$ mount /dev/sda1 /media/sda1`
- `$ mkdir /media/mmcblk0p1`
- `$ mount /dev/mmcblk0p1 /media/mmcblk0p1`
- # 拷贝(替换)文件到合适的路径
- `$ cp -f /media/sda1/MLO /media/mmcblk0p1/`
- `$ cp -f /media/sda1/u-boot.img /media/mmcblk0p1/`
- `$ cp -f /media/sda1/zImage /media/mmcblk0p1/`
- `$ cp -f /media/sda1/am335x-boneblack.dtb /boot/uboot/dtbs/`
- `$ tar -xvf /media/sda1/kernel_modules.tar.gz -C /`
- # 同步文件，确保写入 eMMC
- `$ sync`
- `$ reboot`

技术支持和保修服务

技术支持



英蓓特科技对所销售的产品提供一年的免费技术支持服务，技术支持服务范围：

- 提供英蓓特科技嵌入式平台产品的软硬件资源；
- 帮助用户正确地编译和运行我们提供的源代码；
- 用户在按照本公司提供的产品文档操作的情况下，如本公司的嵌入式软硬件产品出现异常问题，我们将提供技术支持；
- 帮助用户判定是否存在产品故障。



以下情况不在我们的免费技术支持服务范围内，但我们将根据情况酌情处理：

- 用户自行开发中遇到的软硬件问题；
- 用户自行修改嵌入式操作系统遇到的问题；
- 用户自己的应用程序遇到的问题；
- 用户自行修改本公司提供的软件代码遇到的问题。

保修服务

- 1) 产品自出售之日起，在正常使用状况下为印刷电路板提供 12 个月的免费保修服务；
- 2) 以下情况不属于免费服务范围，英蓓特科技将酌情收取服务费用：
 - A. 无法提供产品有效购买凭证、产品识别标签撕毁或无法辨认，涂改标签或标签与实际产品不符；
 - B. 未按用户手册操作导致产品损坏的；
 - C. 因天灾 (水灾、火灾、地震、雷击、台风等) 或零件之自然耗损或遇不可抗力导致的产品外观及功能损坏；

- D. 因供电、磕碰、房屋漏水、动物、潮湿、杂 / 异物进入板内等原因导致的产品外观及功能损坏;
 - E. 用户擅自拆焊零件或修改而导致不良或授权非英蓓特科技认可的人员及机构进行产品的拆装、维修, 变更产品出厂规格及配置或扩充非英蓓特科技公司销售或认可的配件及由此引致的产品外观及功能损坏;
 - F. 用户自行安装软件、系统或软件设定不当或由电脑病毒等造成的故障;
 - G. 非经授权渠道购得此产品者。
 - H. 非英蓓特科技对用户做出的超出保修服务范围的承诺 (包括口头及书面等) 由承诺方负责兑现, 英蓓特科技恕不承担任何责任;
- 3) 保修期内由用户发到我们公司的运费由用户承担, 由我们公司发给用户的运费由我们承担; 保修期外的全部运输费用由用户承担。
- 4) 若板卡需要维修, 请联系技术支持服务部。

注意:

 未经本公司许可私自将产品寄回的, 英蓓特科技公司不承担任何责任。

联系方式

技术支持

电话: +86-755-25635626-872/875/897

Email: support@embest-tech.com

销售信息

电话: +86-755-25635626-860/861/862

传真: +86-755-25616057

Email: chinasales@embest-tech.com

公司信息

网站: <http://www.embest-tech.cn>

地址: 深圳市南山区留仙大道 1183 号南山云谷创新产业园山水楼 4 楼 B