Beaglebone green
User Manual

Release date： 2015/9/22

Version： 1.0

Wiki: http://www.seeedstudio.com/wiki/Beaglebone_green

Bazaar: http://www.seeedstudio.com/depot/BeagleBone-Green-p-2504.html?cPath=122_113

Document Revision History

Revision	Date	Author	Description
1.0	Sep 22, 2015	Loovee	Create file

Contents

Disclaimer

For physical injuries and possessions loss caused by those reasons which are not related to product quality, such as operating without following manual guide, natural disasters or force majeure, we take no responsibility for that.

Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

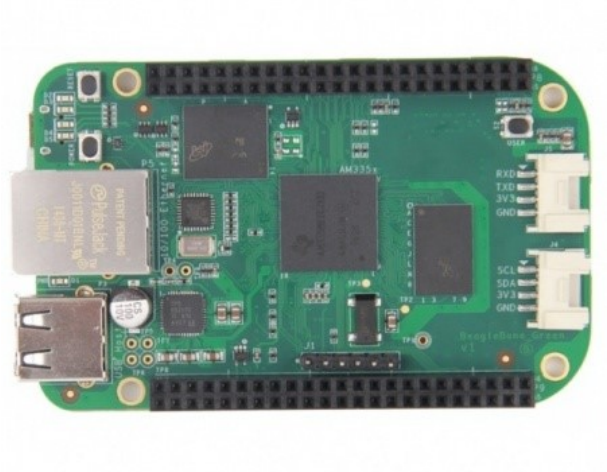
Copyright

1. Introduction

BeagleBone Green (BBG) is based on the classical open-source hardware design of BeagleBone Black (BBB) and added two Grove connectors.

It has removed the HDMI port on the BBB and also updated the 5V barrel to Micro USB host. It is a low-cost, community-supported development platform for developers and hobbyists.

Boot Linux in under 10 seconds and get started on development in less than 5 minutes with just a single USB cable.



2. Technical Details

Processor

- AM335x 1GHz ARM® Cortex-A8
- 512MB DDR3 RAM
- 4GB eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point

Software Compatibility

- Debian
- Android
- Ubuntu
- Cloud9 IDE on Node
- Support much more

Connectivity

- USB client for power & communications
- USB host
- Ethernet
- 2x Grove
- 2x 46 pin headers

3. Hardware Overview

Seeed.cc's BeagleBoneGreen page documents all of the known hardware issues,

as well as the latest available software, hardware documentation and design materials.

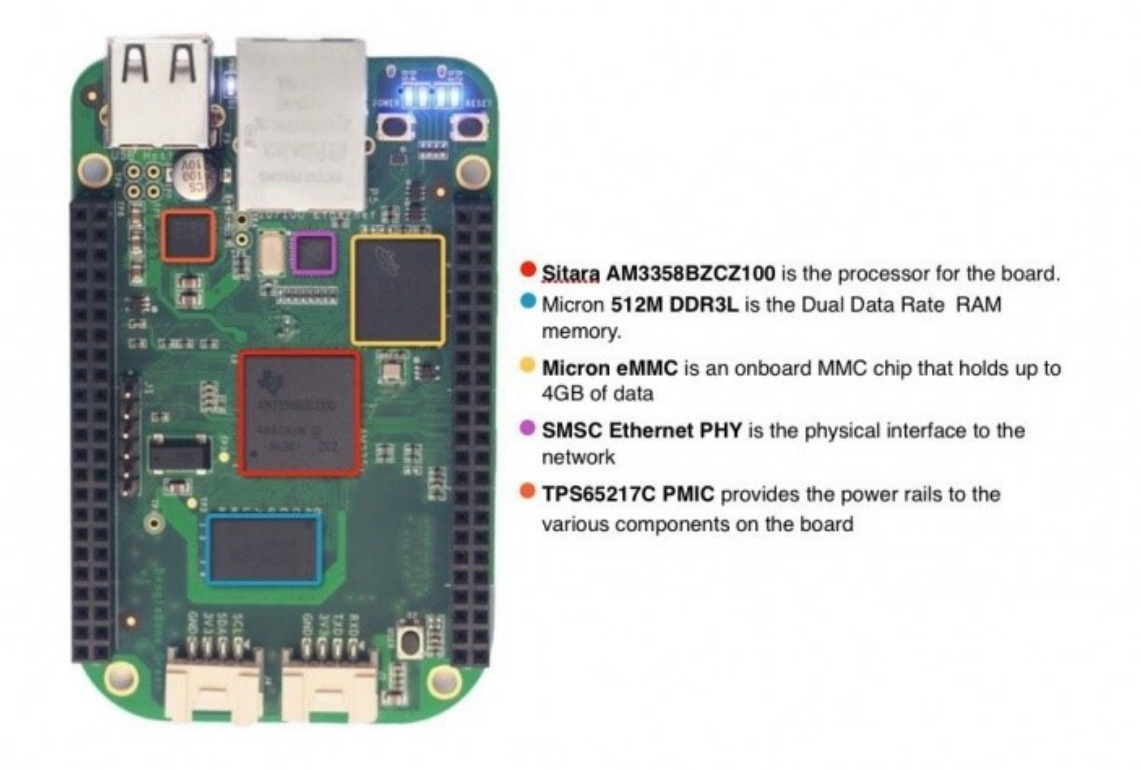
Always read the System Reference Manual!

3.1 Design materials

Design materials for creating your own customized version of the hardware,

or for better understanding the design are also linked from the traditional home of "http://beagleboard.org/hardware/design"

BeagleBone Green hardware details



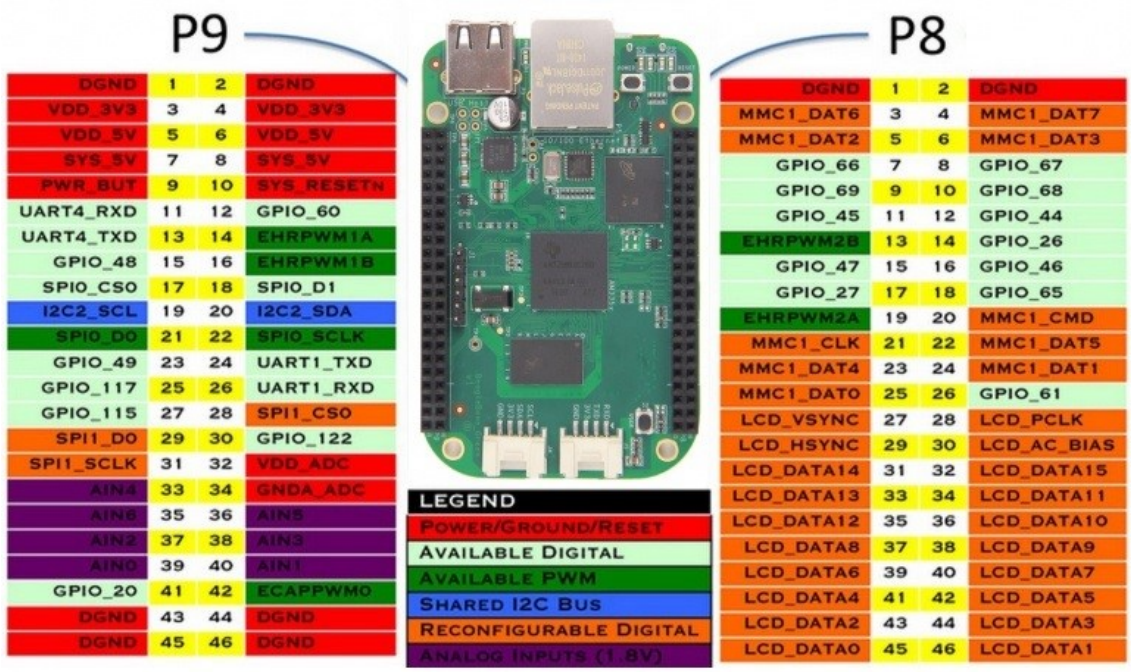
POWER button can be used to enter and exit hibernate modes once that feature is implemented in the software.

3.2 Headers

The expansion headers provide extensive I/O capabilities.

3.2.1 Cape Expansion Headers

Each digital I/O pin has 8 different modes that can be selected, including GPIO.



3.2.2 65 Possible Digital I/Os

In GPIO mode, each digital I/O can produce interrupts.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

3.2.3 PWMs and Timers

Up to 8 digital I/O pins can be configured with pulse-width modulators (PWM) to produce signals to control motors or create pseudo analog voltage levels, without taking up any extra CPU cycles.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
EHRPWM0B	21	22	EHRPWM0A
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	ECAPPWM2
EHRPWM0B	29	30	GPIO_122
EHRPWM0A	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
TIMER4	7	8	TIMER7
TIMER5	9	10	TIMER6
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	EHRPWM1B
GPIO_8	35	36	EHRPWM1A
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
EHRPWM2A	45	46	EHRPWM2B

3.2.4 Analog Inputs

Make sure you don't input more than 1.8V to the analog input pins. This is a single 12-bit analog-to-digital converter with 8 channels, 7 of which are made available on the headers.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

3.2.5 UART

There is a dedicated header for getting to the UART0 pins and connecting a debug cable. Five additional serial ports are brought to the expansion headers, but one of them only has a single direction brought to the headers.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
UART1_RTSN	19	20	UART1_CTSN
UART2_TXD	21	22	UART2_RXD
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	UART3_TXD
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
UART5_CTSN+	31	32	UART5_RTSN
UART4_RTSN	33	34	UART3_RTSN
UART4_CTSN	35	36	UART3_CTSN
UARR5_TXD+	37	38	UART5_RXD+
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

3.2.6 I2C

The first I2C bus is utilized for reading EEPROMS on cape add-on boards and can't be used for other digital I/O operations without interfering with that function, but you can still use it to add other I2C devices at available addresses. The second I2C bus is available for you to configure and use.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
I2C1_SCL	17	18	I2C1_SDA
I2C2_SCL	19	20	I2C2_SDA
I2C2_SCL	21	22	I2C2_SDA
GPIO_49	23	24	I2C1_SCL
GPIO_117	25	26	I2C1_SDA
GPIO_115	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

3.2.7 SPI

For shifting out data fast, you might consider using one of the SPI ports.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
SPI0_CS0	17	18	SPI0_D1
SPI1_CS1	19	20	SPI1_CS0
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	SPI1_D1
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	SPI1_CS1
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

4. Software

4.1 Software Support

Android? Ubuntu? Gentoo? Whatever your software preference, get started on development with your Beagle and the software resources by

click http://beagleboard.org/Support/Software+Support.

4.2 Getting Started

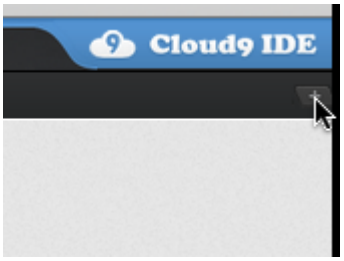
Beaglebone Green is a tiny computer with all the capability of today's desktop machines, without the bulk, expense, or noise. Read the step-by-step getting started tutorial by

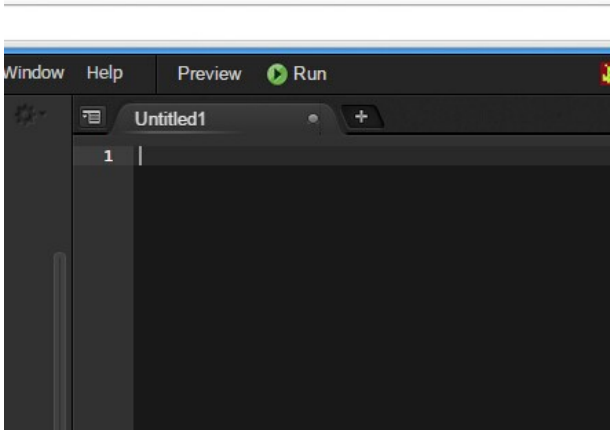
click http://beagleboard.org/getting-started to begin developing with your BeagleBone in minutes.

4.3 Cloud9 IDE

To begin editing programs that live on your board, you can use the Cloud9 IDE. As a simple exercise to become familiar with Cloud9 IDE, creating a simple application to blink one of the 4 user programmable LEDs on the BeagleBone is a good start.

- **Step1:** Close any open file tabs.
- **Step2:** Click the "+" in the top-right to create a new file.





- **Step3:** Copy and paste the following code into the new tab

```
import Adafruit_BBIO.GPIO as GPIO
import time
GPIO.setup("P9_14", GPIO.OUT)
while True:
    GPIO.output("P9_14", GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output("P9_14", GPIO.LOW)
    time.sleep(0.5)
```

- **Step4:** Save the file by clicking the disk icon and giving the file a name with the .py extension.
- **Step5:** Run the code

Select the arrow to the right of "**run**" (or "**debug**") in the toolbar to pull down the list of files to run and select your new file.

- **Step6:** Observe the BeagleBone P9_14 led blinking steadily about once a second.
- **Step7:** Stop the code by clicking "stop" in the toolbar.

4.4 Update image

There are multiple ways to run initial software on your board, but it is likley that the simplest way to get an update is to create an exact replica of a bootable microSD card and boot off of it.

The BeagleBone Green that can be initialized by a program booted off of a microSD card. If you want to update to the latest software image for your board, this is a way to do that.

4.4.1 Step1: Download the latest microSD card image

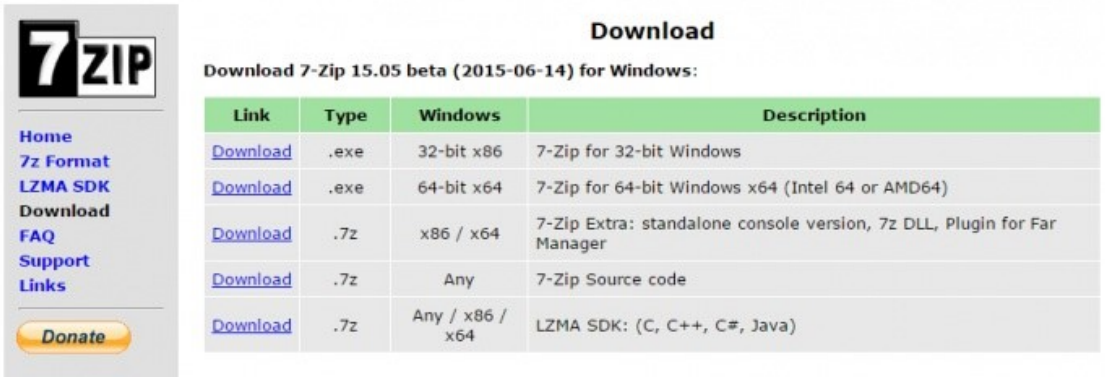
Download the image from

http://statics.seeed.cc/products/platforms/BeagleBoneGreen/bone-debian-7.8-lxde-4gb-armhf-2015-07-13-4gb-for-user.7z .

The file you download will have an .img.xz extention. This is a compressed sector-by-sector image of the SD card.

4.4.2 Step2: Install compression utility

Download and install 7-zip by click http://www.7-zip.org/download.html .



4.4.3 Step3: Decompress the image

Use 7-zip to decompress the SD card .img file

4.4.4 Step4: Install SD card programming utility

Download and install Image Writer for Windows by click

http://sourceforge.net/projects/win32diskimager/files/latest/download .

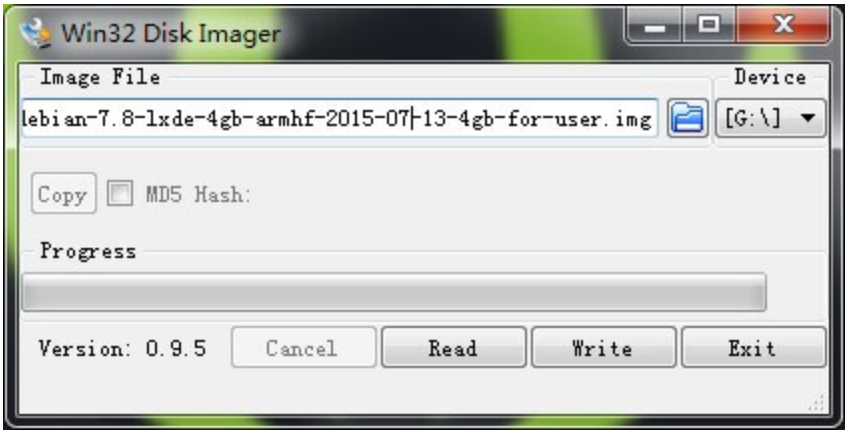
Be sure to download the binary distribution.

4.4.5 Step5: Connect SD card to your computer

Use the provided microSD card to SD adapter or a USB adapter to connect the SD card to your computer.

4.4.6 Step6: Write the image to your SD card

Use either the Ubuntu Image Writer or instructions on its page to write the decompressed image to your SD card.



4.4.7 Step7: Eject the SD card

Eject the newly programmed SD card.

4.4.8 Step8: Boot your board off of the SD card

Insert SD card into your (powered-down) board, hold down the USER/BOOT button (if using Black) and apply power, either by the USB cable or 5V adapter.

If using an original white BeagleBone, you are done.

If using BeagleBone Green and the image is meant to program your on-board eMMC, you'll need to wait while the programming occurs.

When the flashing is complete, all 4 USRx LEDs will be lit solid.

Note: This can take up to 45 minutes. Power-down your board, remove the SD card and apply power again to be complete.