Chipsee BeagleBoneBlack User Manual

V1.0.0

Copyright:

Update record:

Version	Date	Details
v1.0.0	20140401	The initial version

Summary：

This manual is used to provide users with a fast guide of Chipsee BeagleBone Black expansion board. Through this manual, users can quickly understand the hardware resources; and simple declaration of the Systems for Beaglebone blac k: Angstrom, Debian, Android and WinCE 7.

Contects

1. Chipsee BeagleBoneBlack Development Kit

Hardware:

（1）BeagleBoneBlack

（2）Chipsee BeagleBoneBlack expansion (board 4.3 inch)

（3）Micro SD card and card reader

（4）5V power adapter

（5）Micro USB cable

（6）Common serial cable or USB to serial cable

Software:

（1）Android 4.2 prebuilt file(to make a bootable SD card)

（2）Debian prebuilt file(to make a bootable SD card)

（3）WinCE 7 binary file(to make a bootable SD card)

Chipsee BeagleBoneBlack 4.3 inch expansion board is designed for Beaglebone black, as s hown in Figure 1-1: LCD, buttons, COM. You can use the systems from the official webs ite: Angstrom, Android and Debian, and we also provide WinCE 7.

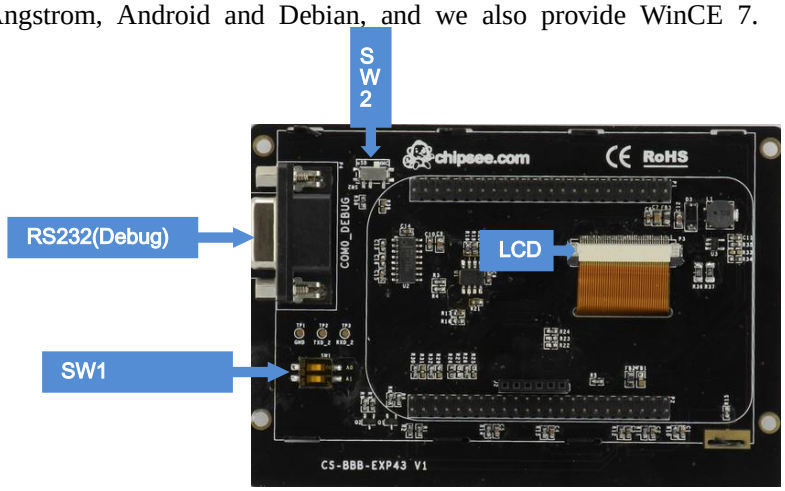

Figure 1-1 Hardware

2. Angstrom OS

In this chapter, you will see the information of the Angstrom system.

2.1. Getting started

When you get the Beaglebone black, the Angstrom system already exists in the eMMC if you want to update the OS, follow the steps here:http://elinux.org/Beagleboard:Updating_The_Software

After testing, we found only version "Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.06.06.img.xz" and "BBB-eMMC-flasher-2013.06.06.img.xz" work prefect for the expansion board, the others may have a little problem of touchscreen.

2.1.1 Start Angstrom OS

1. Make sure the SD card slot of BeagleBoneBlack empty.
2. Connect BeagleBoneBlack with the expansion board, switch SW2 to “eMMC”.
3. Power on the board.

Wait for a moment, Angstrom system will boot successfully:

Figure 2-1 Angstrom Desktop

2.2. Tests

2.2.1 Touchscreen

Click on the screen, the mouse arrow stays in position triggered, the touchscreen works well, After working for some time resistive touch screen may not be accurate, need to be calibrated: delete the file ***/etc/pointercal.xinput,*** choose ***System→Administration→Calibrate Touchscreen*** application to recalibrate.

2.2.2 GPIO test(buttons)

The definition of the buttons on board as shown in Figure 2-2:

Figure 2-2 Buttons

2.2.3 Serial port

Connect COM0_DEBUG on board to PC, use software SecureCRT or putty in PC, se t baud 115200, you can see the information when the system star-up:

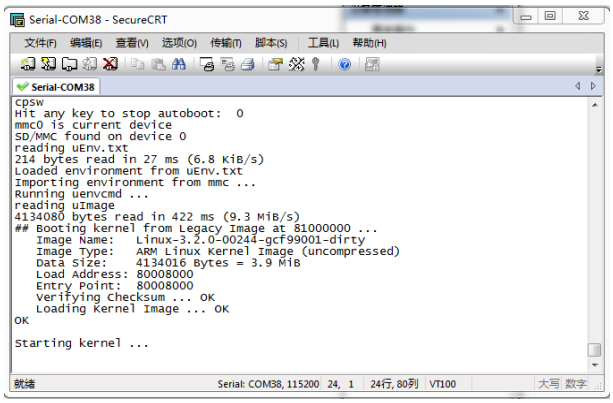

Figure 2-3 System star-up information

Login the system as root, no password.

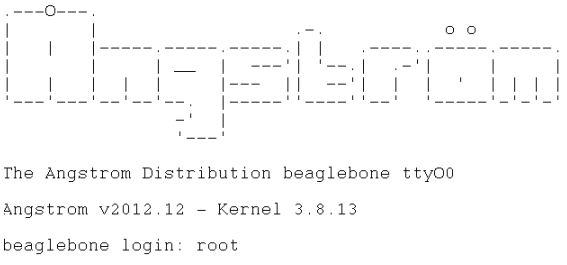

Figure 2-4 Login

2.2.4 Backlight

Change the value of file /sys/class/backlight/backlight.11/brightness.
Range from 0 to 100, 0 means shutdown the backlight, 100 is the MAX

```
# echo 50 > /sys/class/backlight/backlight.11/Brightness
```

3. Debian OS

In this chapter, you will see the information of the Debian system.

3.1. Getting started

3.1.1 Make a bootable SD card

For Linux:
1. Insert the SD card into your computer, if using virtual machines, please make sure the SD card mounted to the Ubuntu(or other Linux) operating system.
2. Confirm the SD card mount point, "**/dev/sdX**" usually it should be "**/dev/sdb".** You can use this command to find out what the "X" is in the Linux system.

```
$ sudo fdisk –l
```

3. Copy the file ***"bone-debian-7.4-2014-03-04-2gb.img.xz"*** somewhere(such as $HOME).
4. Flash the Debian OS to the SD card

```
$ xz –cd bone-debian-7.4-2014-03-04-2gb.img.xz > /dev/sd<?>
```

For Windows:
1. Install "HP USB Disk Storage Format Tool.exe", then format the uSD card as FAT32.
2. Install "7z_win32.exe", then extract the file "bone-debian-7.4-2014-03-04-2gb.img.xz".
3. Extract "win32diskimager-v0.9.zip", write the "bone-debian-7.4-2014-03-04-2gb.img" file to uSD card.

Note: If you want to flash the OS to eMMC, you download the BBB-eMMC-flasher version here:http://beagleboard.org/latest-images

3.1.2 Start Angstrom OS

1. Insert SD card into BeagleBoneBlack
2. Connect BeagleBoneBlack with the expansion board, switch SW2 to "uSD".
3. Power on the board

After a while, you can see the desktop of Debian:

Figure 3-1 Debian Desktop

3.2. Tests

3.2.1 Touchscreen

Click on the screen, the mouse arrow stays in position triggered, indicating that t ouch works properly. After working for some time resistive touch screen may not be accurate, need to be calibrated: choose ***Preferences→Calibrate Touchscreen*** app to rec alibrate, but it only works this time, if you reboot the System, you need to do it aga in. You can do it this way: delete the file ***/etc/pointercal.xinput***, then reboot. You wil l see the calibrate app first before you access to the system. Just calibrate, the result will saved.

3.2.2 GPIO test(buttons)

The definition of the buttons on board as shown in Figure 3-3:

Figure 3-3 Buttons

3.2.3 Serial port

Connect COM0_DEBUG on board to PC, use software SecureCRT or putty in PC, set baud 115200, you can see the information when the system star-up:

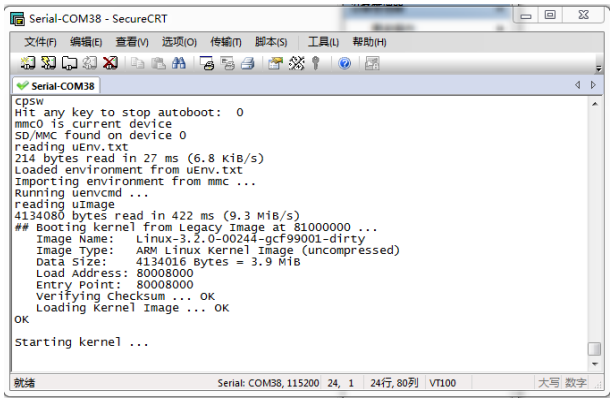

Figure 3-4 System star-up information

Login as debian, password is temppwd。

```
Debian GNU/Linux 7 beaglebone ttyO0

default username:password is [debian:temppwd]

The IP Address for usb0 is: 192.168.7.2
beaglebone login:
```

Figure 3-5 Login

If you want to change the information shown before login, you can edit the file **/etc/issue**.

3.2.4 Backlight

Change the value of file /sys/class/backlight/backlight.11/brightness.

Range from 0 to 100, 0 means shutdown the backlight, 100 is the MAX

```
# echo 50 > /sys/class/backlight/backlight.11/Brightness
```

3.3. Java

This chapter we will setup the environment of Java, and show you how to a simple Java application

1. Install jdk

```
# sudo apt-get install openjdk-6-jdk
```

2. Edit a simple program HelloWorld.java

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Toolkit;
import javax.swing.JFrame;
import javax.swing.JTextField;
```

```
public class HelloWorld extends JFrame{
    public HelloWorld(){
        JTextField text = new JTextField("Hello, world!");
        text.setFont(new Font("Times New Roman",Font.BOLD,60));
        text.setForeground(Color.BLACK);
        this.getContentPane().add(text);
    }
    public static void main(String argv[]){
        HelloWorld win = new HelloWorld();
        Toolkit tk  = Toolkit.getDefaultToolkit();
        int winWidth = 512;
        int winHeight = 300;
        int Width  = tk.getScreenSize().width;
        int Height = tk.getScreenSize().height;
        win.setSize(winWidth, winHeight);
        win.setLocation((Width-winWidth)/2, (Height-winHeight)/2);
        win.setVisible(true);
        win.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

3. Compile the source

```
# javac HelloWorld.java
```

This will be very slow in Debian OS, we suggest do it in your PC, you need install jdk-1.6 first.

4. Run the program

```
# java HelloWorld
```

You will see this:

Figure 3-6 HelloWorld(1)

5. Adding Java program to Quick Start

a) Make a directory

```
# mkdir /usr/lib/java/
```

b) Copy HelloWorld.class to /usr/lib/java/

```
# cp HelloWorld.class /usr/lib/java/
```

c) Edit script **/usr/bin/HelloWorld.sh** like this:

```
#!/bin/bash
cd /usr/lib/java/
java HelloWorld
```

Change the permissions of the script

```
# sudo chmod a+x HelloWorld.sh
```

d) Edit file **/usr/share/applications/javatest.desktop** like this:

```
[Desktop Entry]
Name=HelloWorld
Comment=Simple test for Java
Exec=/usr/bin/HelloWorld.sh
Icon=/usr/share/pixmaps/chipsee.png
Terminal=false
Type=Application
Categories=GTK;Utility;GNOME;
```

This is the result:

Figure 3-7 HelloWorld(2)

6. Auto-Launch Java app

Add script **89javatest** in directory **/etc/X11/Xsession.d/** :

```
#!/bin/bash
cd /usr/lib/java/
java HelloWorld
```

Reboot, the app HelloWorld will automatically launch.

4. Android OS

In this chapter, you will see the information of the Android system.

4.1. Getting started

4.1.1 Make a bootable SD card

1. Insert the SD card into your computer, if using virtual machines, please make sure the SD card mounted to the Ubuntu(or other Linux) operating system.
2. Confirm the SD card mount point, "**/dev/sdX**" usually it should be "**/dev/sdb".** You can use this command to find out what the "X" is in the Linux system.

```
$ sudo fdisk –l
```

3. Copy the file ***"TI_Android_JB_4.2.2_DevKit_4.1.1_beagleboneblack.tar.gz"*** somewhere(such as $HOME).
4. Extract the file ***"TI_Android_JB_4.2.2_DevKit_4.1.1_beagleboneblack.tar.gz***"

```
$ tar xzvf TI_Android_JB_4.2.2_DevKit_4.1.1_beagleboneblack.tar.gz
```

5. Go to the folder "***beagleboneblack***"

```
$ cd beagleboneblack
```

6. Flash the Android OS to the SD card

```
$ sudo ./mkmmc-android.sh /dev/sd<?>
```

Note: more information about Android 4.2 you can find here：http://processors.wiki.ti.com/index.php/TI-Android-JB-4.2.2-DevKit-4.1.1_DeveloperGuide

4.1.2 Start Android OS

1. Insert SD card into BeagleBoneBlack
2. Switch SW2 on expansion board to "uSD", switch SW1 to "ON".
3. Power on the board

This is the desktop of Android 4.2 OS:

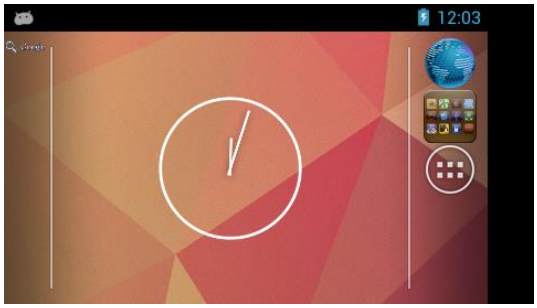

Figure 4-1 Android desktop

4.2. Tests

4.2.1 Touchscreen

Install and run app “MultiTouchTester” :

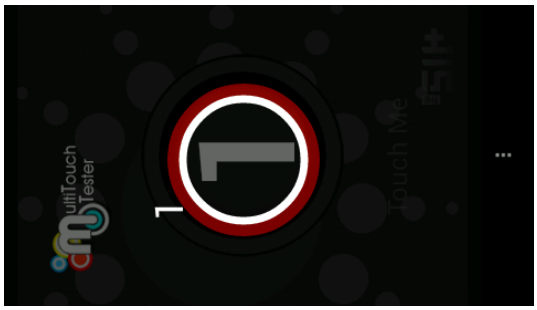

Figure 4-2 MultiTouchTester

4.2.2 GPIO test(buttons)

The buttons on board are designed for Angstrom OS, you can use some of them:
LEFT is used as BACK in Android OS, RIGHT is HOME, UP is MENU, DOWN is SE ARCH, and RESET is used to reboot the system.

Figure 4-3 buttons

4.2.3 Serial port

Connect COM0_DEBUG on board to PC, use software SecureCRT or putty in PC, se t baud 115200, you can see the information when the system star-up:

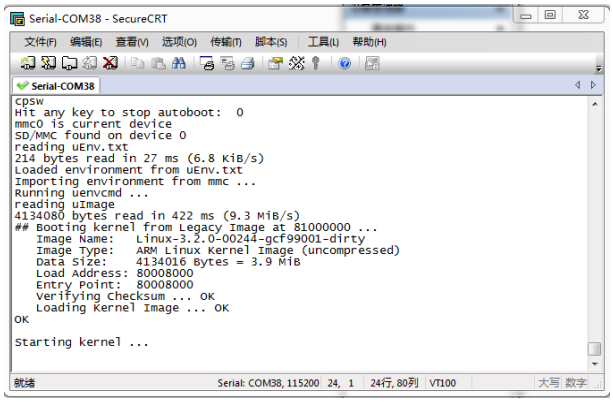

Figure 4-4 System star-up information

After the system bootup successfully, you can communicate with the board via serial, just as shell.

4.3. Development in Windows

In this chapter we will describe how to view Android system via the serial port and how to debug the system via USB OTG. We can also install applications via USB OTG. The following operation under Windows 7 x64 environment, similar to other Windows platfor ms.

4.3.1 Adb connect via USB OTG

1. Install Oracle JDK 6 for Windows.
2. Install ADT. Download the file here: http://developer.android.com/sdk/index.html.

Extract the file somewhere(named ADT). Adb command located**<ADT>\sdk\platform-tools.**

3. Optionally, you may want to add the location of the SDK's primary tools directory to your system PATH. Right-click on My Computer, and select Properties. Under the Advanc ed tab, hit the Environment Variables button, and in the dialog that comes up, double-clic k on Path (under System Variables). Add the full path to the tools\ directory to the path.
4. Install Android USB driver: Copy the folder "**usb_driver"** in CD to **<ADT>** folder

Boot the board as normal and wait until shell prompt is available (micro-B USB cable m ust be disconnected).

- Connect micro-B USB cable between board and Windows PC.
- If it is proceeding as planned, Windows will tell you it found a new hardware asks yo u to install the driver. Install driver that was downloaded as described in step 3 above: Answer "No, not this time" to the question about running Windows Update to search for software.
- Choose "Install the hardware that I manually select from a list (Advanced)" this is the 2nd option, then click "Next"
- Select "Show All Devices", then click "Next"
- You are going to see a grayed-out text box with "(Retrieving a list of all devices)", cli

ck the "Have Disk..." button

• Browse" to your driver folder (<ADT>\usb_driver). It will be looking of a .inf file so s elect "android_winusb.inf" and click "Open" then "OK". It's the only file there so you sho uldn't go wrong.

• Select "Android ADB Interface" then click the "Next" button.

• A warning will appear, answer "Yes" but read the warning anyway.

• Click the "Close" when the wizard is completed.

Now you can see the driver is installed successfully link Figure 4-5.

Figure 4-5 ADB driver

5. Test adb: “Win+r” enter “cmd”, test like below:

```
> cd <ADT>\sdk\platform-tools\
> adb kill-server
> adb start-server
> adb devices
> adb shell
```

When the "#" prompt appears, it means we connect the board with PC successfully.

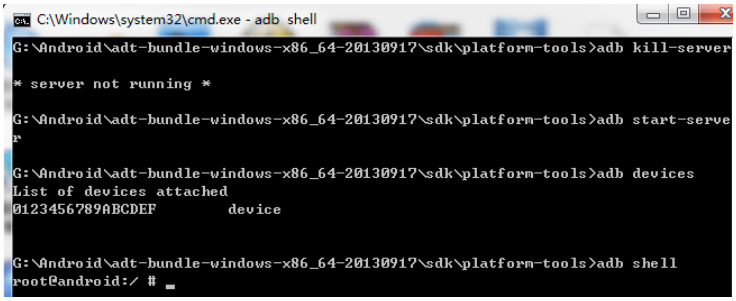

Figure 4-6 ADB Command

Now you can use Linux commands like “ls”, “cd” and so on. Ctrl + C to exit the shell return to Windows system.

6. Use adb command to install Android App: for example SogouInput.apk.

```
> adb install SogouInput.apk
```

If there is a “Success”, the app has already installed in Android.

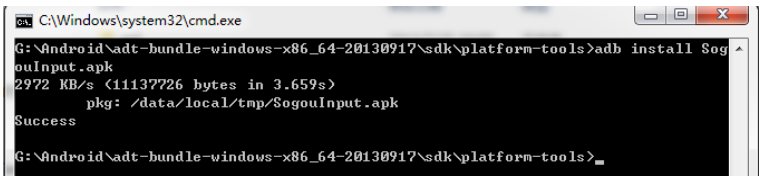

Figure 4-7 Install App

7. Use adb command to uninstall App

(1) Uninstall user app(such as ***SogouInput.apk***): Use command “pm list” to get the full nam

e of the app, like Figure 4-8. Then use command "uninstall" to uninstall the app.

```
> adb shell pm list packages
> adb uninstall com.sohu.inputmethod.sogou
```

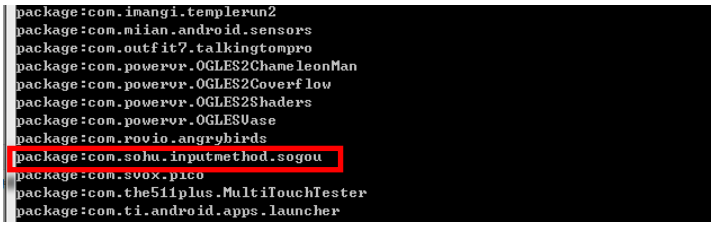

Figure 4-8 Command "pm list" to get app's name

(2) Uninstall default app: Use "adb shell" to log in the board and delete the apk file.

```
> adb shell
# cd /system/app/
# ls
# rm Browser.apk
```

8. Use adb command to transport files between board and PC: "adb pull" and "adb push"

(1) Board to PC: **<remote>** is the file or folder on board, **<local>** is the file or fold er in PC.

```
>adb pull <remote> <local>
```

(2) PC to board:

```
>adb push <local> <remote>
```

For example copy <ADT>\sdk\platform-tools\chipsee.txt to board:

```
>adb push chipsee.txt /chipsee.txt
```

Opposite, board to PC:

```
>adb pull /testFile.txt testFile.txt
```

4.3.2 Adb connect via internet

Make sure Ethernet port on board and host machine are connected to the network. C heck Ethernet configuration for the board

```
# netcfg
lo      UP    127.0.0.1       255.0.0.0       0x00000049
eth0    UP    192.168.1.117/24  255.255.252.0   0x00001043
```

1. If Ethernet was not configured, configure Ethernet of the board using ifconfig/netc fg as shown below.

```
# netcfg eth0 dhcp
```

2. Configure the ADB Daemon to use an ethernet connection using setprop as show n below.

```
# setprop service.adb.tcp.port 5555
```

3. If network is configured successfully (above steps) then Restart service adbd on t

he target

```
# stop adbd
# start adbd
```

4. On the host machine use following commands to establish adb connection

```
$ adb kill-server
$ adb start-server
$ adb connect <target_ip_address>:5555
```

5. Verify for device connectivity, by executing the following commands. If connected, find the device name listed as a "IPADDRESS:PORT"

```
$ adb devices
List of devices attached
emulator-5554       device
192.168.1.117:5555   device
```

6. An example of using adb to install software for Android

Make sure **.apk at the current folder, and export the adb path

```
$ adb –s 192.168.1.117:5555 install **.apk
```

Use the argument **–s** to appoint the device over the internet.

5. WinCE 7

In this chapter, you will see the information of the Android system.

5.1. Getting started

1. Install "HP USB Disk Storage Format Tool.exe", format the uSD card as FAT32.
2. Copy all the files in folder **WinCE7/Bin** in the CD to the uSD card.
3. Insert SD card into BeagleBoneBlack
4. Switch SW2 on expansion board to “uSD”, switch SW1 to “ON”.
5. Power on the board

This is the desktop of WinCE 7:

Figure 5-1 WinCE 7 Desktop

5.2. Tests

5.2.1 Touchscreen

The first time you login, you need to calibrate first. If you want to recalibrate, you can do this:

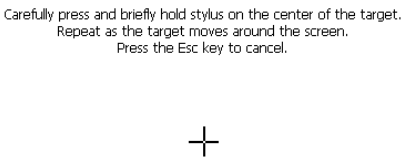

Figure 5-2 Touchscreen calibrate

5.2.2 GPIO test(buttios)

The definition of the buttons like Figure 5-3:

Figure 5-3 Buttons

Contact us:

TEL: 010-62105170
FAX: 010-62561127
E-mail: service@chipsee.com

About us:

ChipSee Information Technology Co., Ltd.
WEB: http://www.chipsee.com
Address: Room1004,Haojing Mansion 3#,Zhichun Rd.108,Haidian District, Beijing