BB-View Cape

Portable LCD Solution for the BeagleBone Family
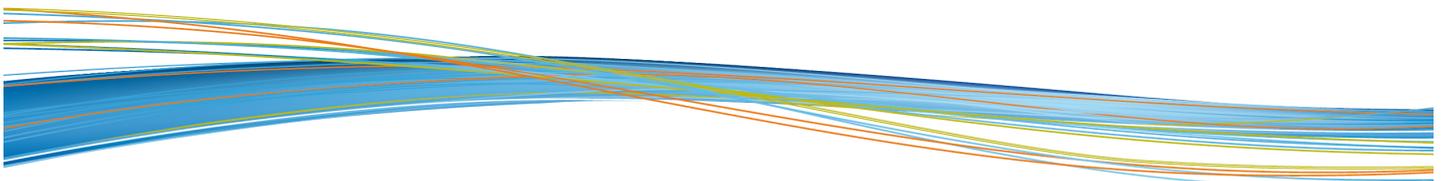
BY

element14

User Manual

Version 2

Dated: 21st May 2014

Revision History:

Version	Date	Description
1.0	26/11/2013	Original Version
1.1	27/03/2014	Correction

DISCLAIMER

This product is intended to be used for **ENGINEERING DEVELOPMENT, DEMONSTRATION OR EVALUATION PURPOSES ONLY** and is not considered by element14 to be a finished end product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards.

The goods being provided are not intended to be complete in terms of required design and/or manufacturing related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

Table of Contents

1 Product Overview

1.1 Introduction

The BB-View is a portable LCD expansion cape with touchscreen capability for BeagleBone boards, a credit-card-sized expandable Linux computer to evaluate the TI's Sitara™ AM335x ARM® Cortex™-A8 processors.

The BB-View is 24-bit LCD expansion cape supplied with a TFT LCD(18-bit) module, available in two size options: 4.3" & 7", which can display up to a resolution of 480x272 (4.3" LCD) and 800x480 (7" LCD). Both have a 4-wire resistive touchscreen interface.

BB-View has been designed with convenience in mind and extends the I/O interfaces of the BeagleBone & BeagleBone Black allowing users to utilise a touchscreen LCD module without sacrificing I/O interface access. BB-View fits on top of the BeagleBone OR BeagleBone Black and still has full access to all the GPIOs via two 46-pin connectors. It's equipped with five switches (four for GPIOs & one for Boot) and two user defined LEDs.

The BB-View draws power directly from the board (BeagleBone or BeagleBone Black) eliminating the need for any kind of external power supply. The BB-View is also supplied with a pre-compiled image with Linux QT demos to help set up your BeagleBone & BeagleBone Black board quickly and easily.

1.2 Kit Contents

- ✓ BB-View Cape Only [BeagleBone OR BeagleBone Black not included]
- ✓ 4.3" OR 7" Touchscreen LCD Module
- ✓ Flexible FPC Cable
- ✓ Quick Start Guide

1.3 Board Interfaces
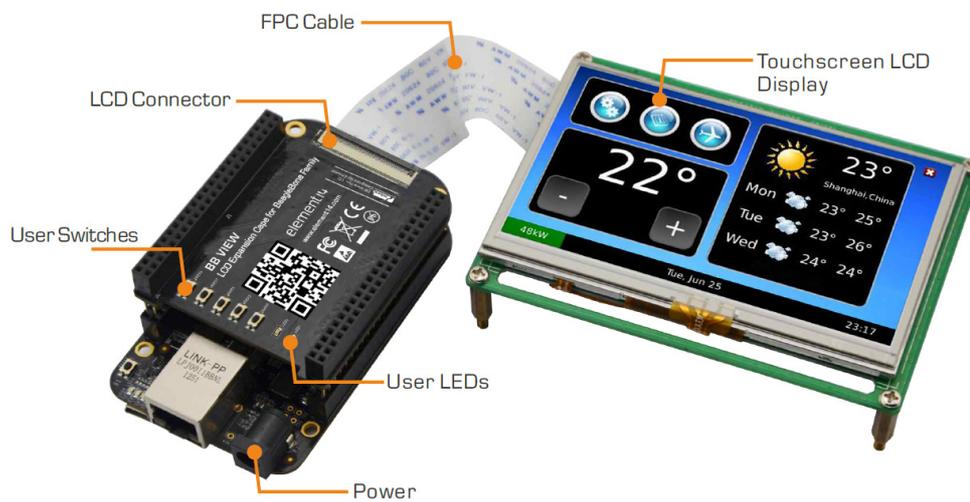
1.4 ystem Block Diagram
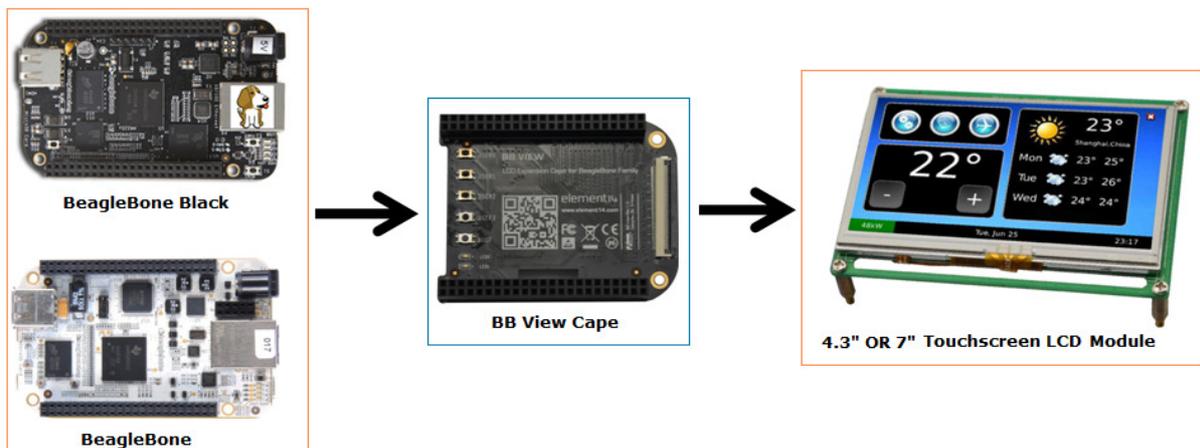
1.5 Physical Dimensions (mm)
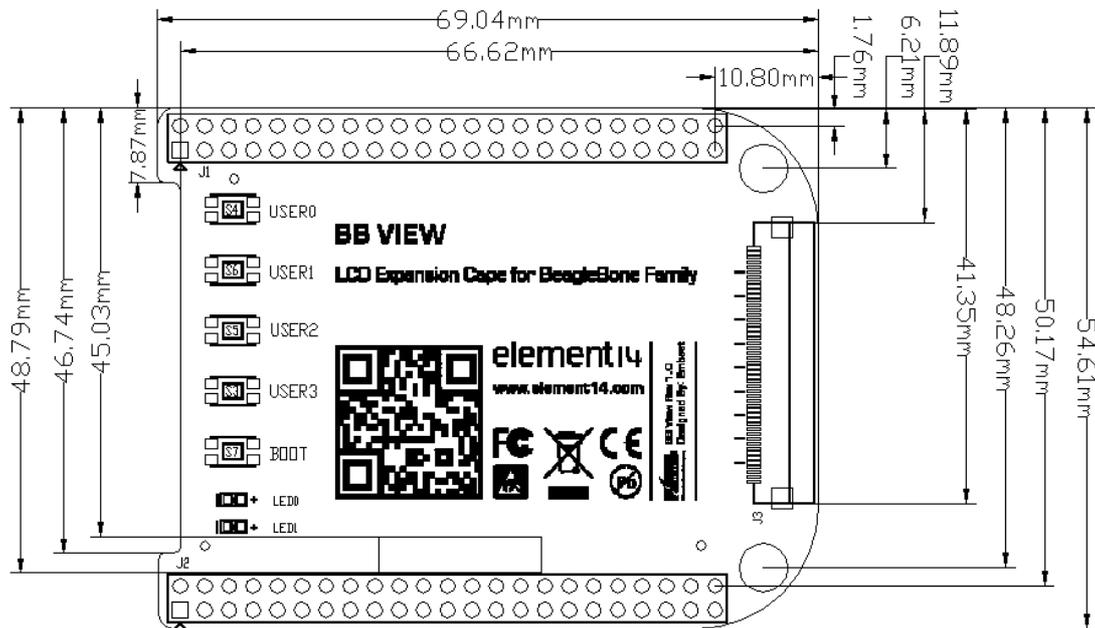
2 BB-View Features

2.1 Hardware and Software Features:

- A 24-bit LCD expansion cape for the BeagleBone board family
- 4-wire resistive touchscreen LCD modules (18-bit):
  - 4.3" LCD: 480x272 resolution
  - 7" LCD: 800x480 resolution
- Five switches (four for GPIOs &   one for BOOT)
- Two user defined LEDs
- Full access to all the GPIOs via two 46-pin connectors (besides those already used by the BB-View)
- Powered directly from BeagleBone boards, no external power supply required.
- Provided with pre-compiled BSP image with QT Demo to help setup the BeagleBone board quickly and easily.
- Works with BeagleBone & BeagleBone Black
- Supporting TISDK and Angstrom image
- Drivers and driver source code are provided along with BB-View

2.2 Operational Parameters:

- Dimensions: 69.04mm x 54.61mm
- Operating Humidity: 20% ~ 90%
- Power Supply: +5V (provided by BeagleBone board)
- PCB Layers: 4

3 Hardware Description

BB-View expansion cape has an LCD connector (LCD) and two 46-position dual-row connectors (J1 and J2) that can be used as extended I/O interfaces for the BeagleBone & BeagleBone Black. This chapter contains pin definitions for these connectors.
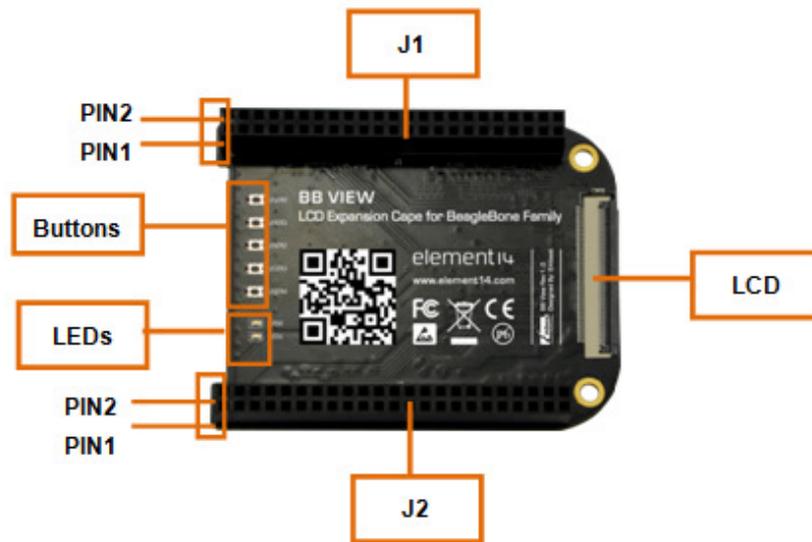
Figure 1: Pin Locations on the BB-View

3.1 LCD Interface (LCD)

Pins	Definitions	Descriptions
1	B0	LCD Pixel data bit 23
2	B1	LCD Pixel data bit 20
3	B2	LCD Pixel data bit 17
4	B3	LCD Pixel data bit 11
5	B4	LCD Pixel data bit 12
6	B5	LCD Pixel data bit 13
7	B6	LCD Pixel data bit 14
8	B7	LCD Pixel data bit 15
9	GND	GND

Pins	Definitions	Descriptions
10	G0	LCD Pixel data bit 22
11	G1	LCD Pixel data bit 19
12	G2	LCD Pixel data bit 5
13	G3	LCD Pixel data bit 6
14	G4	LCD Pixel data bit 7
15	G5	LCD Pixel data bit 8
16	G6	LCD Pixel data bit 9
17	G7	LCD Pixel data bit 10
18	GND	GND
19	R0	LCD Pixel data bit 21
20	R1	LCD Pixel data bit 18
21	R2	LCD Pixel data bit 16
22	R3	LCD Pixel data bit 0
23	R4	LCD Pixel data bit 1
24	R5	LCD Pixel data bit 2
25	R6	LCD Pixel data bit 3
26	R7	LCD Pixel data bit 4
27	GND	GND
28	DEN	AC bias control (STN) or pixel data enable (TFT)
29	HSYNC	LCD Horizontal Synchronization
30	VSYNC	LCD Vertical Synchronization
31	GND	GND
32	CLK	LCD Pixel Clock
33	GND	GND
34	X+	X+ Position Input
35	X-	X- Position Input
36	Y+	Y+ Position Input
37	Y-	Y- Position Input
38	SPI_CLK	SPI clock
39	SPI_MOSI	Slave data in, master data out

Pins	Definitions	Descriptions
40	SPI_MISO	Slave data out, master data in
41	SPI_CS	SPI enable
42	IIC_CLK	IIC master serial clock
43	IIC_DAT	IIC serial bidirectional data
44	GND	GND
45	VDD1	3.3V power for LCD buffer
46	VDD2	3.3V
47	VDD3	5V
48	VDD4	5V
49	RESET	No connection
50	PWREN	GPIO

3.2 Extended I/O Interface (J1)

Pins	Definitions	Descriptions
1	GND	GND
2	GND	GND
3	NC	NC
4	NC	NC
5	NC	NC
6	NC	NC
7	NC	NC
8	NC	NC
9	NC	NC
10	NC	NC
11	LCD_DATA18	LCD Pixel data bit 18
12	LCD_DATA19	LCD Pixel data bit 19
13	LCD_DATA22	LCD Pixel data bit 22
14	LCD_DATA21	LCD Pixel data bit 21
15	LCD_DATA16	LCD Pixel data bit 16

Pins	Definitions	Descriptions
16	LCD_DATA17	LCD Pixel data bit 17
17	LCD_DATA20	LCD Pixel data bit 20
18	NC	NC
19	LCD_DATA23	LCD Pixel data bit 23
20	NC	NC
21	NC	NC
22	NC	NC
23	NC	NC
24	NC	NC
25	NC	NC
26	NC	NC
27	LCD_VSYNC	LCD Vertical Synchronization
28	LCD_PCLK	LCD Pixel Clock
29	LCD_HSYNC	LCD Horizontal Synchronization
30	LCD_DE	AC bias control (STN) or pixel data enable (TFT)
31	LCD_DATA14	LCD Pixel data bit 14
32	LCD_DATA15	LCD Pixel data bit 15
33	LCD_DATA13	LCD Pixel data bit13
34	LCD_DATA11	LCD Pixel data bit 11
35	LCD_DATA12	LCD Pixel data bit 12
36	LCD_DATA10	LCD Pixel data bit 10
37	LCD_DATA8	LCD Pixel data bit 8
38	LCD_DATA9	LCD Pixel data bit 9
39	LCD_DATA6	LCD Pixel data bit 6
40	LCD_DATA7	LCD Pixel data bit 7
41	LCD_DATA4	LCD Pixel data bit 4
42	LCD_DATA5	LCD Pixel data bit 5
43	LCD_DATA2	LCD Pixel data bit 2
44	LCD_DATA3	LCD Pixel data bit 3
45	LCD_DATA0	LCD Pixel data bit 0

Pins	Definitions	Descriptions
46	LCD_DATA1	LCD Pixel data bit 1

3.3 Extended I/O Interface (J2)

Pins	Definitions	Descriptions
1	GND	GND
2	GND	GND
3	VDD_3V3B	3.3V
4	VDD_3V3B	3.3V
5	VDD5V	5V
6	VDD5V	5V
7	SYS5V	5V
8	SYS5V	5V
9	NC	NC
10	SYS_RESETn	Reset
11	USER3	GPIO
12	LED0	GPIO
13	NC	NC
14	PWM	Power on enable
15	NC	NC
16	USER0	GPIO
17	I2C1_SCL	IIC3 master serial clock
18	I2C1_SDA	IIC serial bidirectional data
19	NC	NC
20	LED1	GPIO
21	NC	NC
22	NC	NC
23	USER2	GPIO
24	USER1	GPIO
25	NC	NC

Pins	Definitions	Descriptions
26	NC	NC
27	NC	NC
28	SPI1_CS0	SPI enable 0
29	SPI1_DO	SPI data 0
30	SPI1_D1	SPI data 1
31	SPI1_SCLK	SPI Clock
32	VDD_ADC	ADC power
33	NC	NC
34	NC	NC
35	NC	NC
36	NC	NC
37	Y+	Y+ Position Input
38	Y-	Y- Position Input
39	X+	X+ Position Input
40	X-	X- Position Input
41	NC	NC
42	NC	NC
43	GND	GND
44	GND	GND
45	GND	GND
46	GND	GND

3.4 Keypad Interfaces

Pins	Definitions	Descriptions
1	USER0	User defined
2	USER1	User defined
3	USER2	User defined
4	USER3	User defined
5	BOOT	Boot switch

3.5 LED Interfaces

Pins	Definitions	Descriptions
1	USER0	User defined
2	USER1	User defined

4 How to Setup BB-View

A pre-compiled image for BB-View can be downloaded from the element14 website which can be used for demonstration of the board using BeagleBone & BeagleBone Black. However, there are some preparations such as installing USB and Ethernet drivers and setting up PuTTY that need to be done prior to starting the demonstration.

Note:
📖 To download the pre-compiled image for BB-view please visit:
www.element14.com/BeagleBone

4.1 Installing Windows Drivers for the USB Interface

1) BeagleBone boards are supplied with a pre-programmed eMMC with Angstrom system. Connect the BeagleBone or BeagleBone Black to the USB interface of your PC using an OTG cable and power on the board, after the Angstrom system start-up is complete, you can find a drive named "**BEAGLEBONE**" under "**My Computer**".

2) The New Hardware Wizard should automatically detect the new hardware and launch a pop-up window as shown below. Select **Install from a list or specific location (Advanced)** and click **Next**.

Figure 2: *Serial Driver Installation Prompt*

3) Click **Browse** as highlighted in the following image and specify the location of CDCAM folder in the drive BEAGLEBONE and then click **Next** to install the driver.
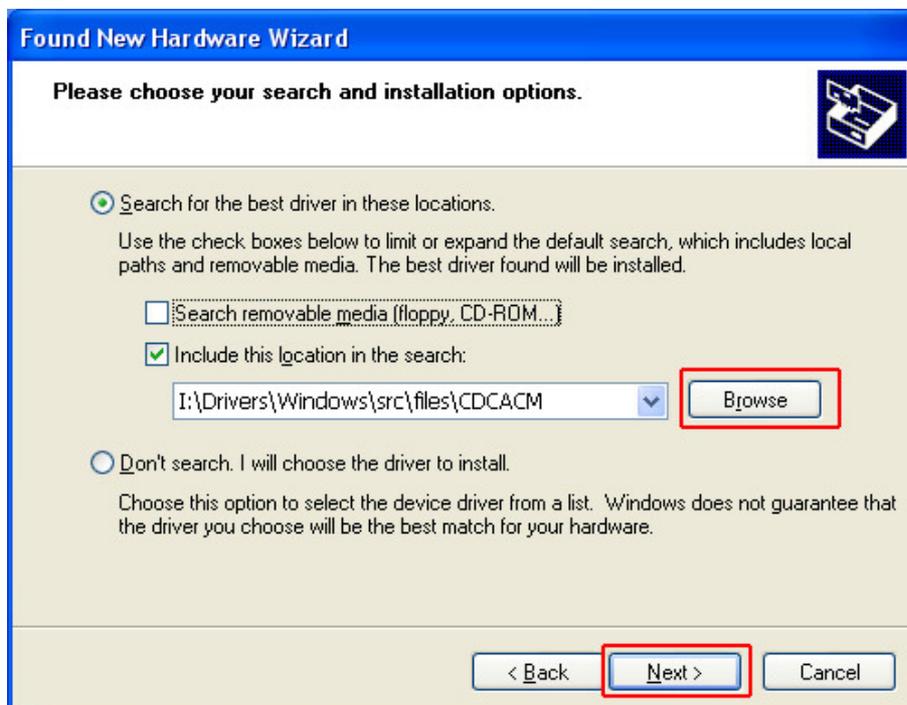

Figure 3: *Setting Driver Location*

4) The New Hardware Wizard should once again pop up having detected another aspect of the hardware. As before, select **Install from a list or specific location (Advanced)** and click **Next** when the following window shows up to install the USB Ethernet driver.
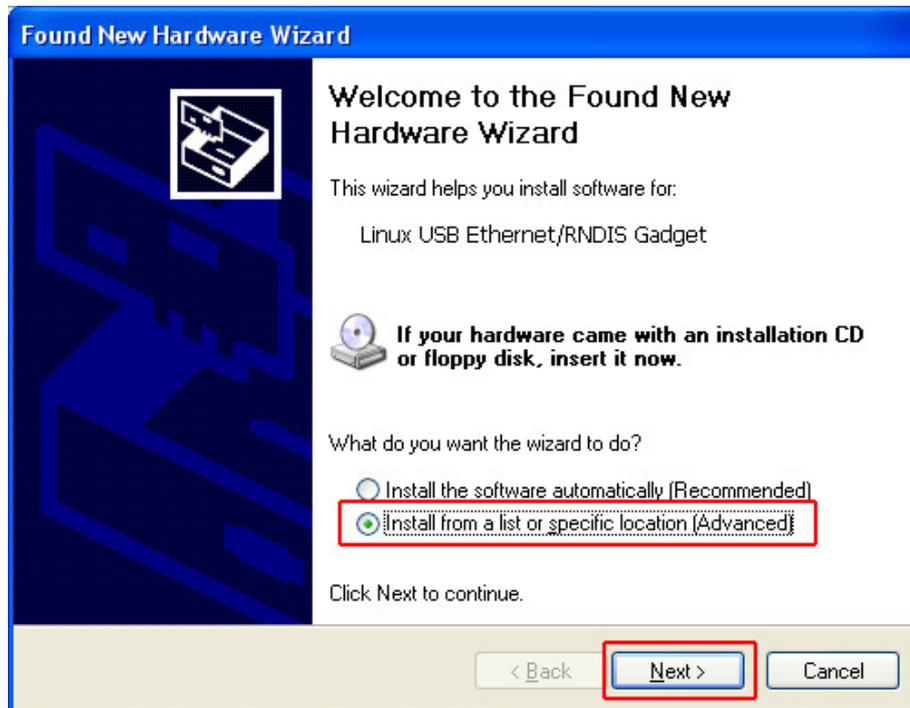
Figure 4: USB Ethernet Driver Installation Prompt

5) Click **Browse** in the following figure and specify the location of the **RNDIS** folder in the drive **BEAGLEBONE** and then click **Next** to install the driver.
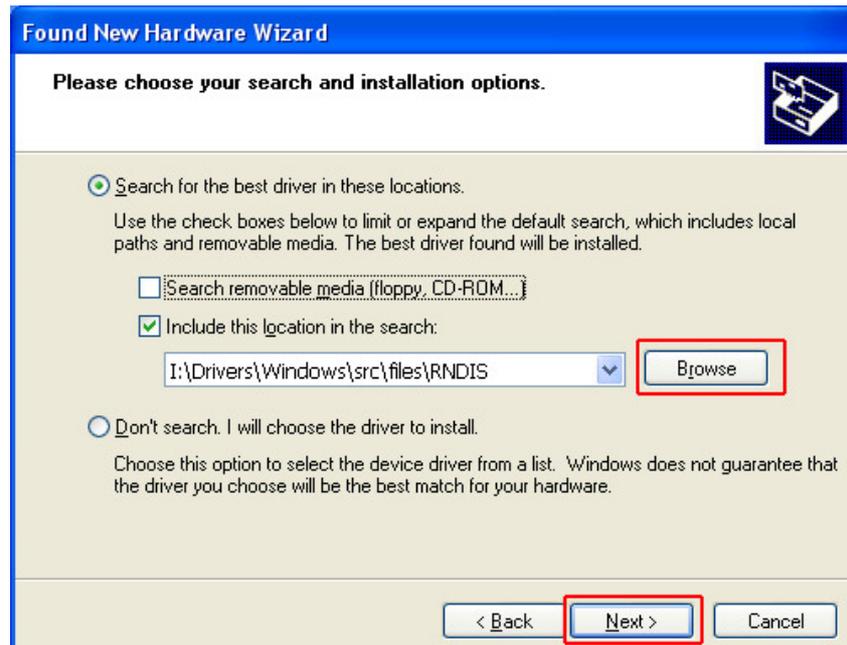
Figure 5: Driver Installation Prompt

4.2 Setting up PuTTY

PuTTY is a virtual terminal which can receive and display the working information of the BeagleBone or BeagleBone Black. It needs to be configured on your PC before it can communicate with BeagleBone or BeagleBone Black. Follow the steps listed below to ensure correct configuration.

1) Download "***putty.exe***" (can be downloaded from www.element14.com/BeagleBone) to your PC and run it to open the **PuTTY Configuration** window then configure it as shown below;
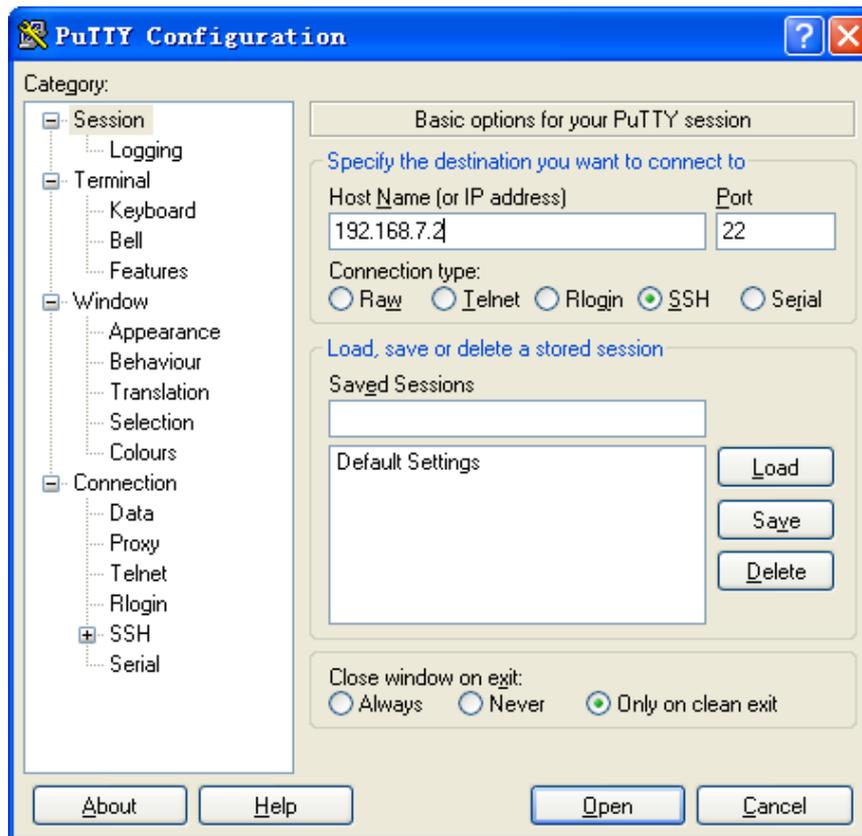
Figure 6: *PuTTY Configuration Settings*

Note:

The Host Name (or IP address) in the above window may need to be changed according to the IP configuration of the board when power is on. The value in Figure 6 is the default. In the event the default value is incorrect please refer to your network documentation for assistance in discovering the IP address of the device.

Click **Open** when you finish the configuration

2) Enter your login details as in the following window and press **enter** on your keyboard. As default the username is "***root***" and the password field should be left empty.
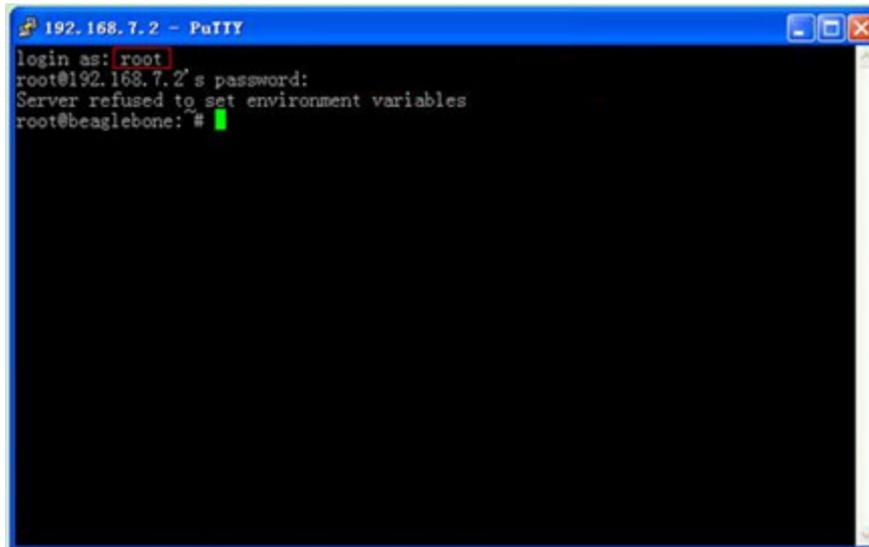
Figure 7: *Login to the BeagleBone Black*

After logging in successfully, a bash shell environment is activated. All the shell instructions hereafter are typed and executed under this environment.

5 Demonstration and Compilation of the TI-SDK

This chapter will introduce how to carry out demonstrations of the display function of the BB-View using BeagleBone & BeagleBone Black with provided pre-compiled TI-SDK, as well as how to create Linux development environment and customise the Linux system.

> **Note:**
> 📖 Each instruction has been proceeded by a pencil "🖉" to prevent confusion caused by the long instructions that occupy more than one line in the document.
> 📖 Please note that there are SPACES in the following instructions; Missing any SPACE will lead to failure when running an application.

5.1 BB-View Demonstration using a Pre-Compiled Image

Here we will use a pre-compiled & patched TI-SDK image provided for the BB-View for demonstration purposes. First we need to program the image into the BeagleBone or BeagleBone Black.

5.1.1 Image Programming

5.1.1.1 For BeagleBone Black

1) Download the "**BB-View Pre-Compiled TI-SDK Image.zip**" to get the following image from:

> **URL: www.element14.com/BeagleBone**
>
> ***Image Name:** "bb-view-emmc-flasher-2013-09-09.img.xz"
>
> **Note: Please use the latest image available for download.*

2) Uncompress the TI-SDK image by using 7-zip compression tool

(available from www.element14.com/BeagleBone)

3) Program the uncompressed image into a MicroSD card by using the "**Win32DiskImager Tool"** (from www.element14.com/BeagleBone) and then insert the card into the BeagleBone Black TF card slot.

4) Press and hold the **uSD BOOT** button and then power on the BeagleBone Black (without connecting BB-View & LCD module).

5) When all 4 user LEDs stay constantly ON, this indicates that the image has been programmed into the BeagleBone Black (*it may take up to 30-40min*). Please power OFF the board and remove the MicroSD card, and then connect BeagleBone Black, BB-View and an LCD module as shown below and power it ON again;
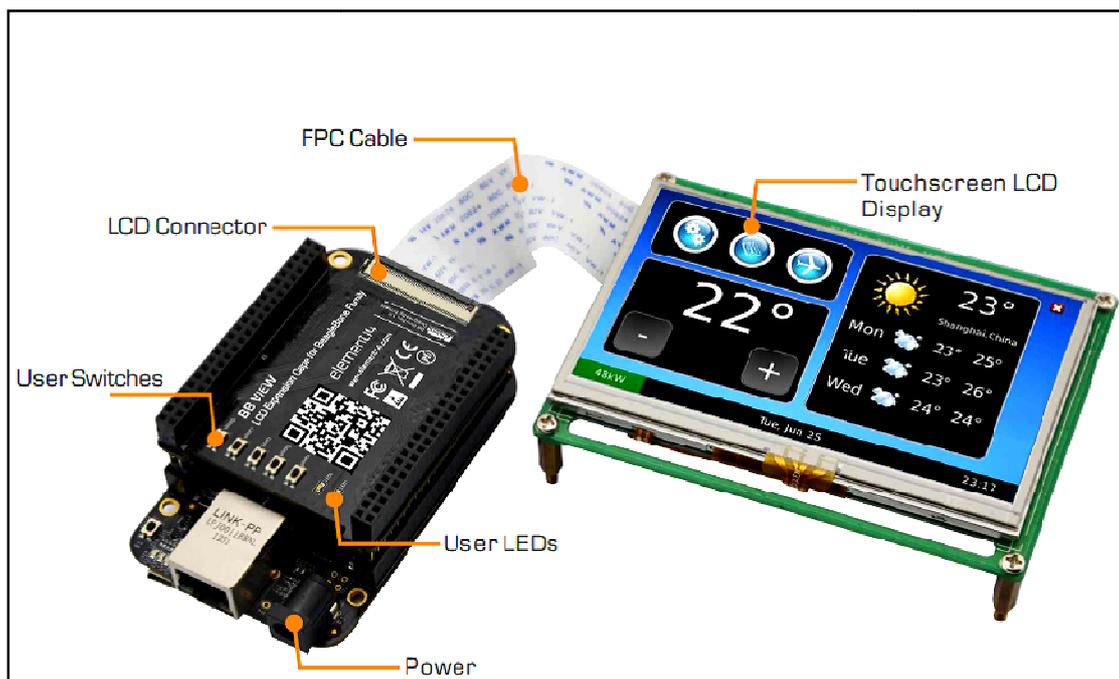

Figure 8: Connecting the BB-View, BeagleBone Black and the LCD Module

Note:
📖 Please ensure the blue tape on the LCD FPC cable is facing upwards.

6) Follow the instructions shown on the LCD module to complete the screen calibration.

5.1.1.2 For BeagleBone

1) Download the "**BB-View Pre-Compiled TI-SDK Image.zip**" to get the following image from:

URL: **www.element14.com/BeagleBone**

*Image Name: "bb-view-mmc-beaglebone-2013-10-16.img.xz"

*Note: Please use the latest image available for download.

---

2) Uncompress the TI-SDK image by using 7-zip compression tool (available from www.element14.com/BeagleBone)

3) Program the uncompressed image into a MicroSD card by using the "*Win32DiskImager Tool"* (from www.element14.com/BeagleBone) and then insert the card into the BeagleBone TF card slot.

4) Connect the BeagleBone, BB-View and an LCD module as shown above and power it ON

5) Follow the instructions shown on the LCD module to complete the screen calibration.

5.1.2 Setting up a 4.3" LCD Module

The following instructions are executed (in PuTTY) for use of a 4.3-inch LCD module;

- ✎ `root@am335x-evm:~# echo "optargs=dispmode=4.3inch_LCD" > /media/mmcblk0p1/uEnv.txt`
- ✎ `root@am335x-evm:~# sync`

Note:

🕮 The system supports 4.3-inch LCD modules by default, and therefore there is no need to

---

execute these instructions if the default settings haven't been changed.

Now restart the board to finish the setup for a 4.3-inch LCD module.

5.1.3 Setting up a 7" LCD Module

The following instructions are executed (in PuTTY) for use of a 7-inch LCD module;

- ✎ `root@am335x-evm:~# echo "optargs=dispmode=7inch_LCD" >`

    `/media/mmcblk0p1/uEnv.txt`

- ✎ `root@am335x-evm:~# sync`

Now restart the board to finish the setup for a 7-inch LCD module.

Note:

📖 The system uses 4.3-inch as a default display mode, to use 7" display for the first time just touches the screen for 5 times to skip touchscreen calibration and enter Linux shell, then use the above commands to configure the display.

📖 If you have calibrated the touch screen, after you change the display mode, you must execute the following instructions to recalibrate in PuTTY:

`root@am335x-evm:~#rm /media/mmcblk0p1/pointercal`

`root@am335x-evm:~# sync`

Now reboot the board and you will be presented with the calibration interface

---

5.2 Keypad Test

Execute the following instruction then press the user keypad buttons to test

- ✎ `root@am335x-evm:~# hexdump -C -v /dev/input/event1`

5.3 LED Test

Turn off LED0:

- ✎ `root@am335x-evm:~# echo 0>`

    `/sys/class/leds/bb-view\:led0/brightness`

Turn on LED0:

   ✎ `root@am335x-evm:~# echo 1>`

     `/sys/class/leds/bb-view\:led0/brightness`

Turn off LED1:

   ✎ `root@am335x-evm:~# echo 0>`

     `/sys/class/leds/bb-view\:led1/brightness`

Turn on LED1:

   ✎ `root@am335x-evm:~# echo 1>`

     `/sys/class/leds/bb-view\:led1/brightness`
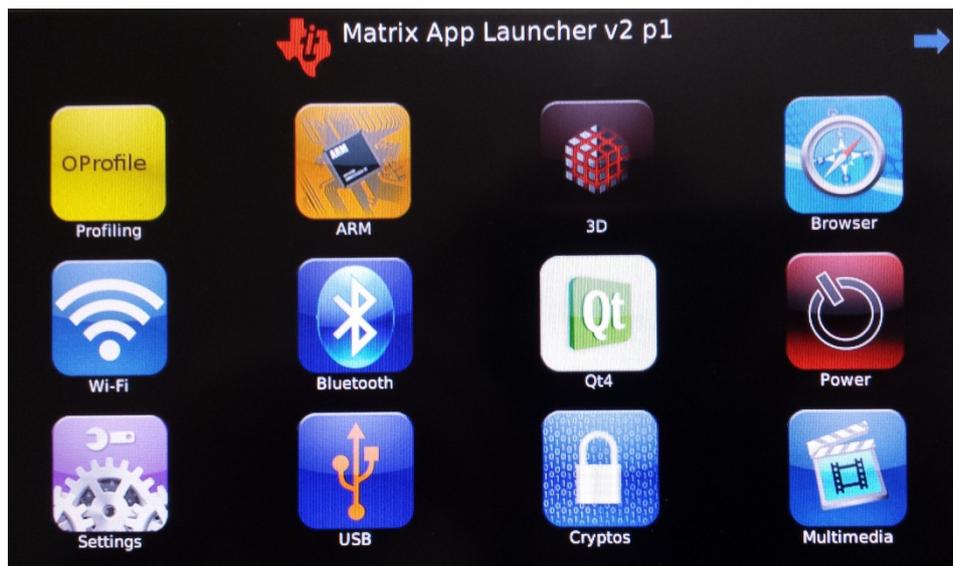
5.4 Pre-Compiled TI-SDK Image Features

Figure 9: TI-SDK Main Menu Screen

The provided pre-compiled TI-SDK image contains a wealth of applications including:

- Benchmarks
- 3D Demo Applications
- Internet Browser
- WiFi Connection (requires separately available WiFi daughter card)
- Bluetooth Connection (requires separately available Bluetooth daughter card)
- Video Decoding & Audio Capture

The TI-SDK also provides several Linux QT based demo applications that showcase the touch screen functionality of the BB-View. These are discussed below:
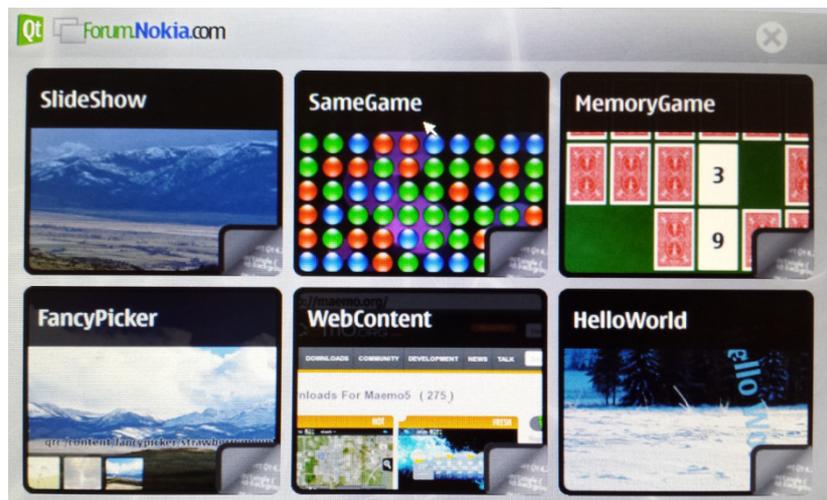
5.4.1 QML Playground

Figure 10: QML Playground Main Menu

This section contains several applications showcasing the features of QT Quick. The demo applications allow the user to view the source code giving an insight into how certain features can be used.
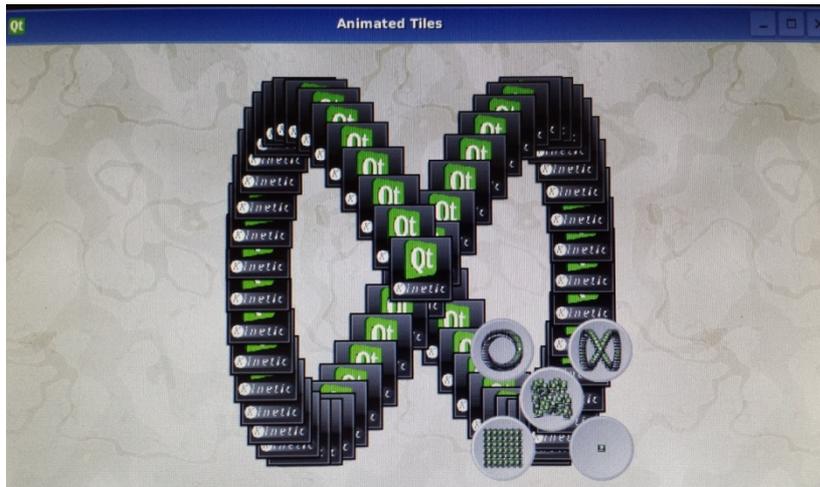
5.4.2 Animated Tiles

Figure 11: *Animated Tiles Demo Application*

The animated tiles demo shows how to animate transitions between five predefined states.
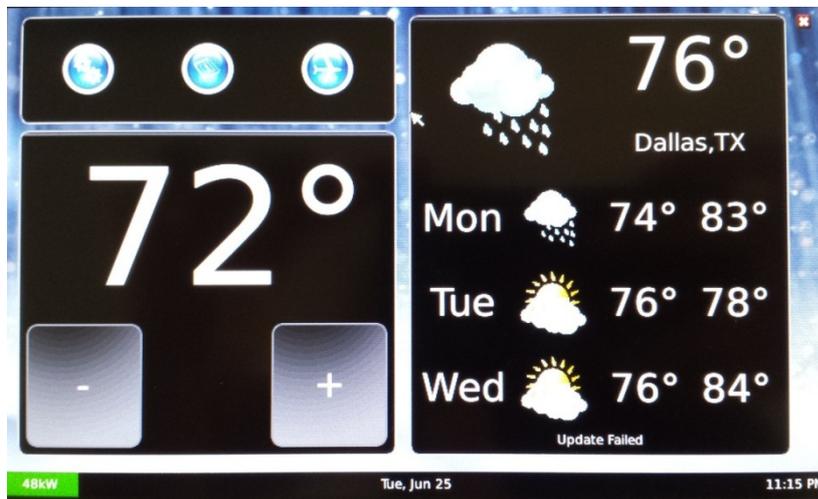
5.4.3 Thermostat Demo

Figure 12: *Thermostat Demo Application*

The thermostat demo demonstrates some of the functionalities that would be available in a home automation system. The demo shows a simple GUI that can be used to control a thermostat. It also draws weather information

from an online service (internet connection required).
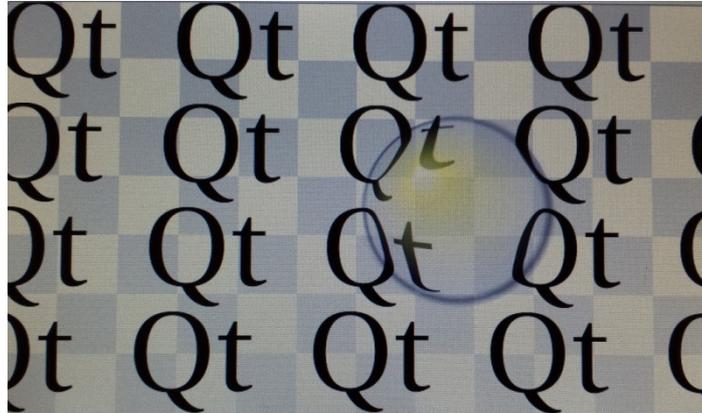
5.4.4 Vector Deformation

Figure 13: *Vector Deformation Demo*

The vector deformation demo shows how advanced vector techniques can be used to draw text using a Qpainter path.

5.5 Working With the TI-SDK Image

5.5.1 Setup for Development Environment

Before getting started with system compilation, a development environment under Linux is required to be built first. (For installation of Linux system, please refer to **Appendix 1 Installing Ubuntu Linux System**).

1) Download the "**BB-View TI-SDK Source Code.zip**" from:
www.element14.com/BeagleBone to the "**$HOME**" directory and unzip. From the directory execute the following instruction to install the development environment;

> ✎ `Ubuntu@Ubuntu:~$ ./ti-sdk-am335x-evm-06.00.00.00-Linux-x8`
> `6-Install --mode console`

2) Follow the instructions to finish the installation, and then you'll find a file "***ti-sdk-am335x-evm-06.00.00.00***" under the *$HOME* directory.

3) Execute the following command to install "***mkimage***".

   ✎ `sudo apt-get install uboot-mkimage`

4) Execute the following instruction to define a temporary environment variable for the compiler in the Ubuntu system.

   ✎ `export `

   `PATH=$HOME/ti-sdk-am335x-evm-06.00.00.00/linux-devkit/sys`

   `roots/i686-arago-linux/usr/bin/:$PATH`

Note:

📖 The instruction used to define environment variables can be added into .bashrc under $HOME so that the system can create the variable automatically when booting up.

📖 To view the path of compiler, please execute **echo $PATH**.

Now a development environment has been installed successfully.

5.5.2 Compiling "Uboot" for BB-View

Users have the two options; either they can use the official TISDK provided by Texas Instruments which needs to be patched first so that it can support BB-View OR they can use the **uboot** developed by element14 which already has been patched to be used with BB-View and ready to use. This section will introduce both the **uboot** compilation processes.

5.5.2.1 Compiling "uboot" from Texas Instruments

1) Copy "**u-boot-2013.01.01-psp06.00.00.00-bb-view.patch**" from "***$HOME***" to:
   "**$HOME/ti-sdk-am335x-evm-06.00.00.00/board-support/**"

2) Execute the following instruction to apply a patch to the **uboot** source code so that it can be running in eMMC;

- ✎ `Ubuntu@Ubuntu:~/ti-sdk-am335x-evm-06.00.00.00$ cd board-support/`

- ✎ `$ cd u-boot-2013.01.01-psp06.00.00.00`

- ✎ `$ patch -p1`

  `< ../u-boot-2013.01.01-psp06.00.00.00-BB-View.patch`

3) Execute the following instruction to compile

- ✎ `$ make CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm distclean`

- ✎ `$ rm -rf am335x`

- ✎ `$ make O=am335x CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm am335x_evm`

After the compilation is done, the files "*MLO*" and "*u-boot.img*" can be found under "*am335x/*" of the current directory.

5.5.2.2 Compiling "uboot" from element14

1) Execute the following commands to uncompress the **uboot** source code

- ✎ `$ cd $HOME`

- ✎ `$ tar xvf u-boot-2013.01.01-psp06.00.00.00-bb-view.tar.bz2`

2) Execute the following instructions to compile

- ✎ `$ cd $HOME/u-boot-2013.01.01-psp06.00.00.00`

- ✎ `$ make CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm distclean`

- ✎ `$ rm -rf am335x`

- ✎ `$ make O=am335x CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm am335x_evm`

After the compilation is done, the files "*MLO*" and "*u-boot.img*" can be found under "*am335x/*" of the current directory.

5.5.3 Compiling "Kernel" for BB-View

There are two options; you can either use the official TISDK provided by Texas Instruments which needs to be patched first so that it can support BB-View OR you can use the Linux kernel developed by element14 which has already been patched for use with the BB-View and is ready to use. This section will introduce both of the **kernel** compilation processes.

5.5.3.1 Compiling the "Kernel" from Texas Instruments

1) Copy "**linux-3.2.0-psp04.06.00.11-BB-View.patch**"

   from "***$HOME***" to

   "$HOME/ti-sdk-am335x-evm-06.00.00.00/board-support/"

2) Execute the following instructions to install the patch and then compile the TI-SDK kernel;

✎ `$ cd $HOME/ti-sdk-am335x-evm-06.00.00.00/board-support/ linux-3.2.0-psp04.06.00.11`

✎ `$ patch -p0 < ../ linux-3.2.0-psp04.06.00.11-BB-View.patch`

✎ `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- distclean`

✎ `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am335x_evm_defconfig`

✎ `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- uImage modules`

A kernel file named "***uImage***" with LCD module support is generated under "***arch/arm/boot/***".

5.5.3.2 Compiling the "Kernel" from element14

Execute the following instructions to unzip the source code and compile the kernel image;

- `$ cd $HOME`
- `$ tar xvf linux-3.2.0-psp04.06.00.11-bb-view.tar.bz2`
- `$ cd linux-3.2.0-psp04.06.00.11`
- `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- distclean`
- `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am335x_evm_defconfig`
- `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- uImage modules`

A kernel file named "*uImage*" with LCD module support is generated under "*arch/arm/boot/*".

5.5.4 System Update with BB-View Support

1) Format a flash drive or a TF card to FAT32 and mount it under the Ubuntu Linux system, and then execute the following instructions to copy the image to the flash drive or TF card. (These instructions assume the flash drive or TF card is mounted under "*/mnt*")

- `$ cd $HOME`
- `$ cp u-boot-2013.01.01-psp06.00.00.00/am335x/MLO /mnt`
- `$ cp u-boot-2013.01.01-psp06.00.00.00/am335x/u-boot.img /mnt`
- `$ cp linux-3.2.0-psp04.06.00.11/arch/arm/uImage /mnt`
- `$ mkdir /mnt/rootfs`
- `$ cd $HOME/linux-3.2.0-psp04.06.00.11`

✎ `$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-`
   `modules_install INSTALL_MOD_PATH=/mnt/rootfs`

✎ `$ cd /mnt/rootfs`

✎ `$ tar -czvf ../kernel_modules.tar.gz ./`

✎ `$ cd /mnt`

✎ `$ rm -rf rootfs`

2) Connect BB-View to the BeagleBone or BeagleBone Black and then
   connect the power supply; Insert the flash drive or TF card into the
   BeagleBone or BeagleBone Black and execute the following instructions in
   PuTTY to update the system image (If a TF card is used instead of a flash
   drive, please change the path in the instructions accordingly)

✎ `$ cp /media/sda1/MLO /media/mmcblk0p1/`

✎ `$ cp /media/sda1/u-boot.img /media/mmcblk0p1/`

✎ `$ cp /media/sda1/uImage /media/mmcblk0p1/`

✎ `$ tar -xvf /media/sda1/kernel_modules.tar.gz -C /`

6 Demonstration and Compilation of Angstrom

This chapter will introduce how to carry out demonstrations of the display functions of BeagleBone Black and BB-View under Angstrom, as well as how to create a Linux development environment and compile the system.

6.1 Demonstration of the Display Function

6.1.1 Image Programming

1) Use the format tool from HP named "**HPUSBFW.exe**" to format a flash drive with the FAT32 file system.

2) Download the **"BB-VIEW Angstrom Image.zip"** to get following files from:

> **URL: www.element14.com/BeagleBone**
>
> **File Names:** am335x-boneblack-lcd4.dtb
>
> am335x-boneblack-lcd7.dtb
>
> am335x-boneblack.dtb
>
> kernel_modules.gz
>
> uImage

   Unzip the files and copy all the image files to the flash drive.

3) Connect the flash drive to the BeagleBone Black and power it ON.

4) Execute the following instructions in a terminal program (such as PuTTY) to mount the flash drive:

   🖉 `$ mkdir /media/udisk`

   🖉 `$ mount /dev/sda1 /media/udisk`

5) Execute the following instructions to install the image and then restart the

system;

- ✎ `$ cp –f /media/udisk/uImage /boot/`

- ✎ `$ cp –f /media/udisk/*.dtb /boot/`

- ✎ `$ tar –xvf /media/udisk/kernel_modules.gz –C /`

- ✎ `$ sync`

6.1.2 Setup for 4.3" LCD Module

The following instructions are executed in PuTTY for use of 4.3" LCD module:

- ✎ `root@beaglebone:~# cd /boot`

- ✎ `root@beaglebone:~# cp am335x-boneblack-lcd4.dtb`
  `am335x-boneblack.dtb`

- ✎ `root@beaglebone:~# sync`

Now restart the board to finish the setup for 4.3" LCD module.

6.1.3 Setup for 7" LCD Module

The following instructions are executed in PuTTY for use of 7" LCD module;

- ✎ `root@beaglebone:~# cd /boot`

- ✎ `root@beaglebone:~# cp am335x-boneblack-lcd7.dtb`
  `am335x-boneblack.dtb`

- ✎ `root@beaglebone:~# sync`

Now restart the board to finish the setup for 7" LCD module.

Note:

📖 If you have calibrated the touch screen, after you change the display mode, you must execute the following instructions to recalibrate in PuTTY:

```
root@beaglebone:~# rm /etc/pointercal.xinput
root@beaglebone:~# sync
```

Then select System > Administration > Calibrate Touchscreen on the desktop of Angstrom

---

6.2 Keypad Test

Execute the following instruction then press the user keypad buttons to test

> ✎ `root@beaglebone:~# hexdump-C-v /dev/input/event2`

6.3 LED Test

Turn off LED0:

> ✎ `root@beaglebone:~# echo 0>`
>
> `/sys/class/leds/bb-view\:led0/brightness`

Turn on LED0:

> ✎ `root@beaglebone:~# echo 1>`
>
> `/sys/class/leds/bb-view\:led0/brightness`

Turn off LED1:

> ✎ `root@beaglebone:~# echo 0>`
>
> `/sys/class/leds/bb-view\:led1/brightness`

Turn on LED1:

> ✎ `root@beaglebone:~# echo 1>`
>
> `/sys/class/leds/bb-view\:led1/brightness`

6.4 Compilation of the Kernel

Users can download the Angstrom kernel source code **"BB-VIEW Angstrom Source Code.zip**" from;

www.element14.com/BeagleBone

File name: "***bb-black-kernel-3.8.13-bb-view.tar.bz2***"

This is enabled with support for BB-View. Please refer to the document relevant to the BeagleBone Black for the details of kernel compilation and updating.

7 Appendix 1: Installing an Ubuntu Linux System

An appropriate development environment is required for software development. The CD included with product contains a development environment which needs to be installed under a Linux environment. If you are working on a PC running Windows, you have to create a Linux system first, and then you can install the environment. An easy method for achieving this is to use virtual machine software such as VirtualBox to install Ubuntu Linux on an emulated/virtual PC. The following sections will introduce the installation processes of VirtualBox and an Ubuntu system.

7.1 Installing VirtualBox

You can access http://www.virtualbox.org/wiki/Downloads to download the latest version of VirtualBox. At the time of writing VirtualBox requires a minimum of 512MB of RAM to run however 1GB is recommended.

1) The installation process is simple. Start VirtualBox from the Start menu of Windows, and then click New in the VirtualBox window. A pop-up window titled "Create New Virtual Machine" will be shown as below:
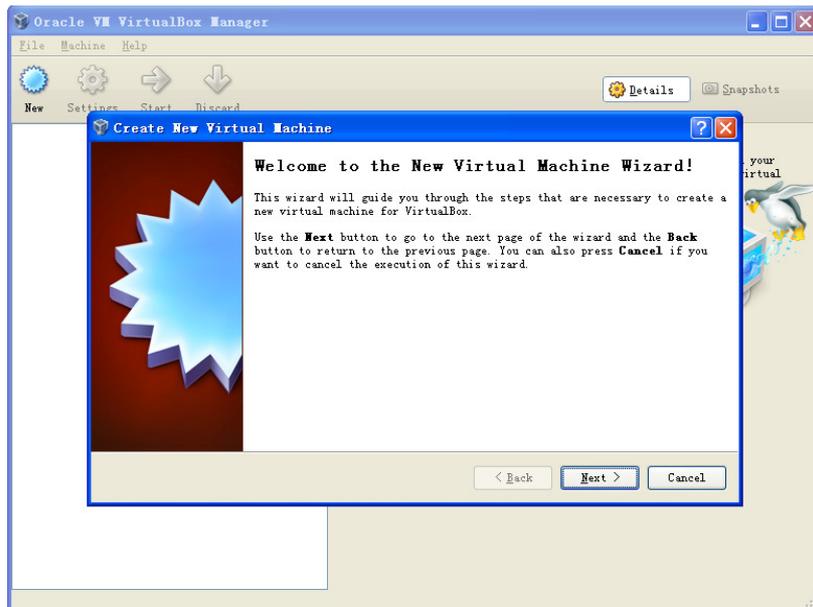
2) Click Next to create a new virtual machine.

3) Enter a name for the new virtual machine and select the operating system type as shown below:
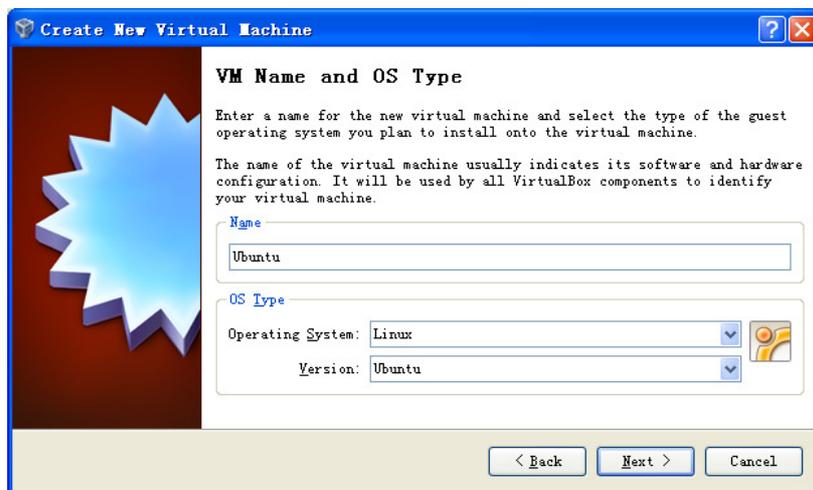
4) Enter a name in the Name field, e.g. Ubuntu, and select Linux in the Operating System drop-down menu, and then click next.

5) Allocate memory to the virtual machine and then click Next
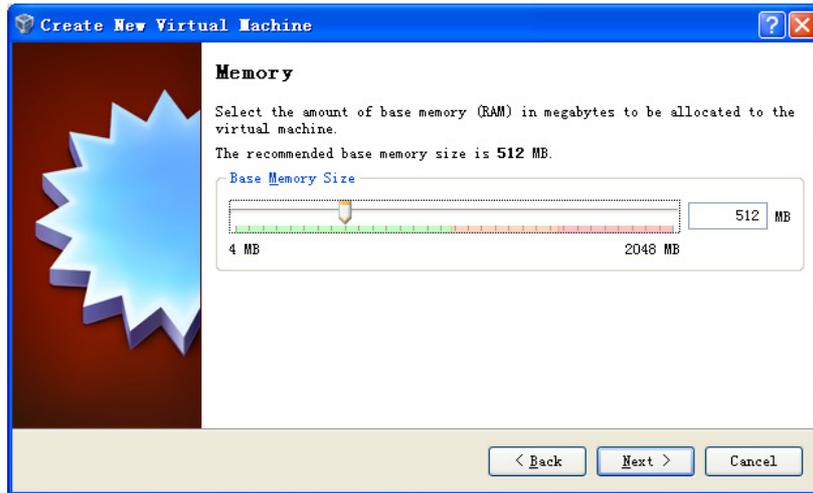
Note:

📖 If your PC has 1GB of RAM or lower, keep the default setting;

📖 If your PC more than 1GB of RAM, you can allocate up to 1/4 to the virtual machine, for example, 512MB out of 2GB memory could be allocated to virtual machine.

6) If this is the first time you have installed VirtualBox, you need to select Create new hard disk in the following window, and then click Next
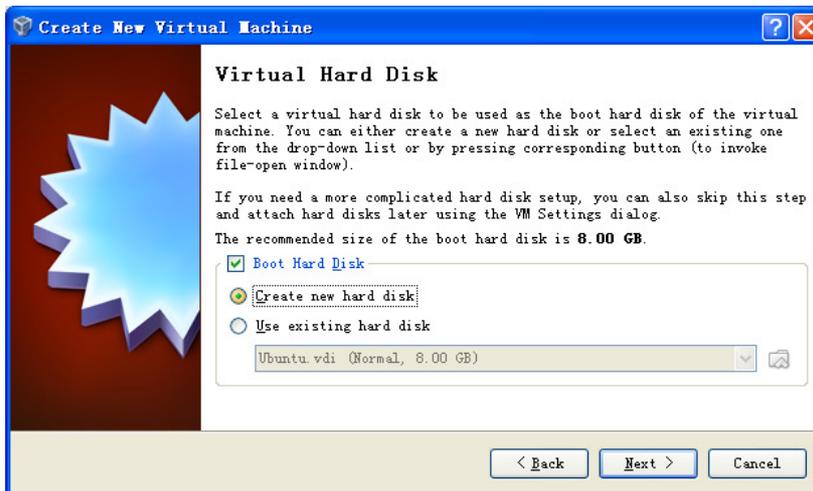
7) Click Next in the following window

8) Select Fixed-size storage in the following window and click Next
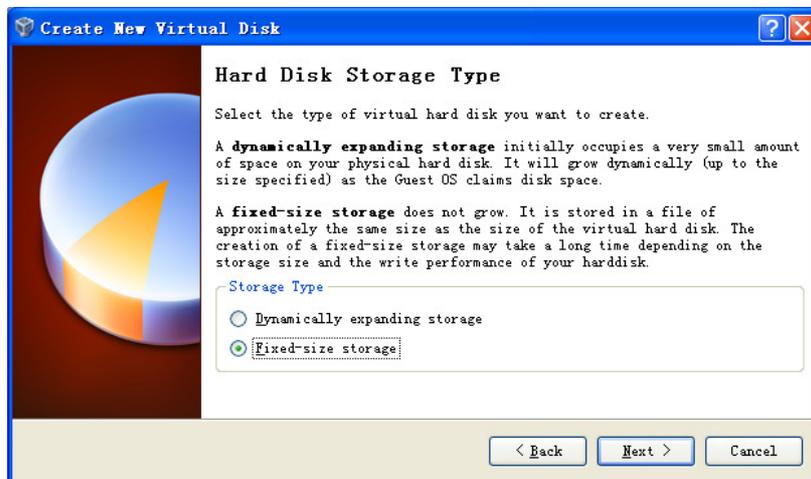
9) Define where the hard disk data is stored and the default space of the virtual disk (8GB at least), and then click Next
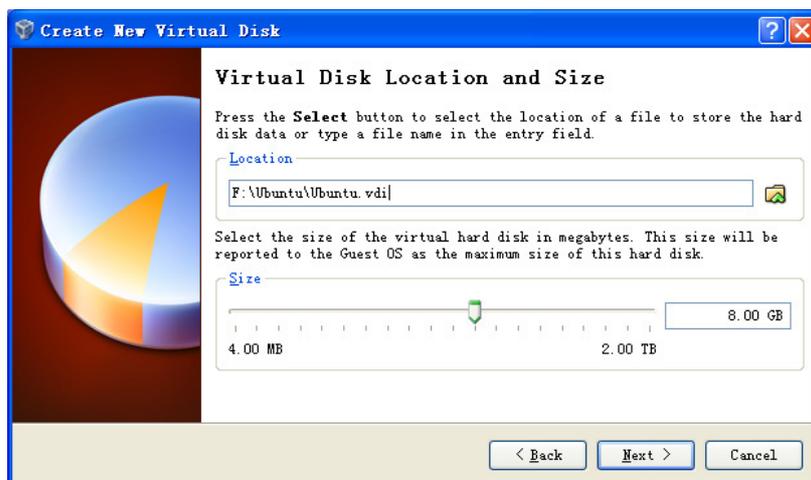
10) Click Finish in the following window
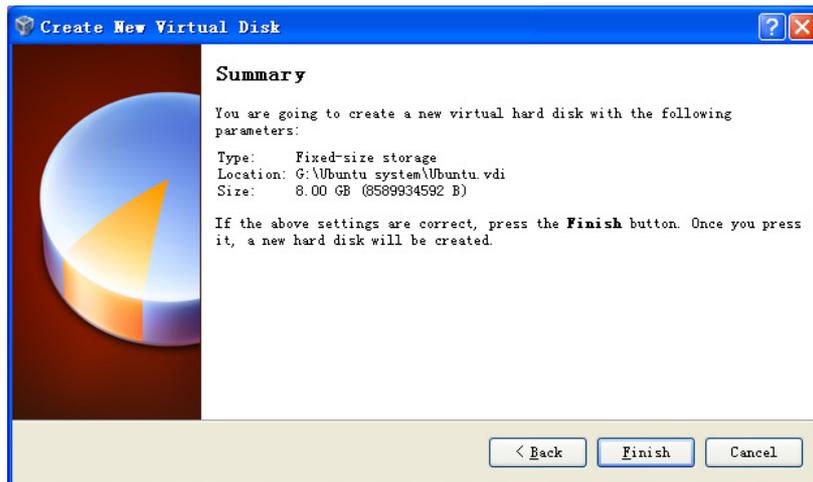
11) Your PC will then create a new virtual disk
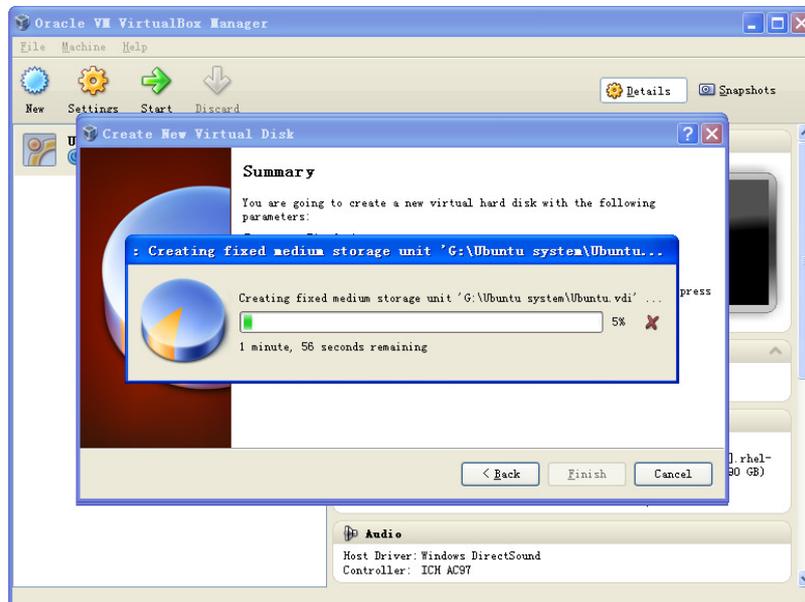
12) A window with summary of the newly created virtual machine will be shown as below when the creation process is done. Please click Finish to complete the whole process.
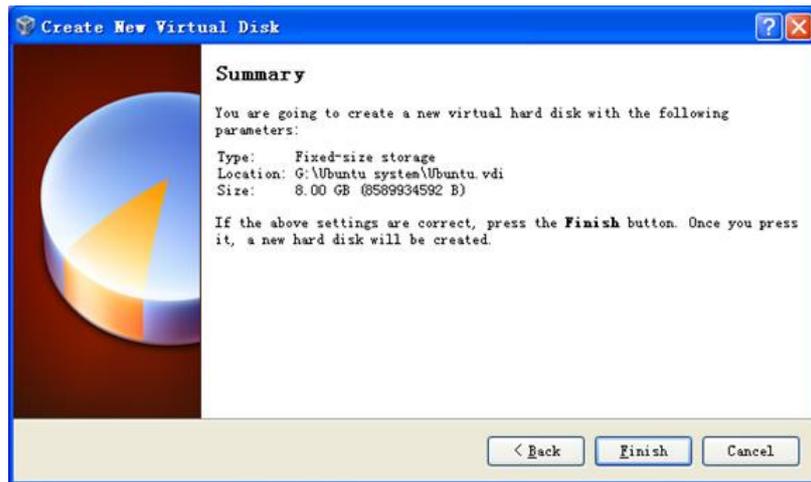
7.2 Installing an Ubuntu Linux System

After VirtualBox is installed, we can install the Ubuntu Linux system. Visit http://www.Ubuntu.com/download/Ubuntu/download to download the ISO image file of Ubuntu, and then follow the steps below:

1) Start VirtualBox from the Start menu and click Settings on the VirtualBox window. A Settings window will be shown as below
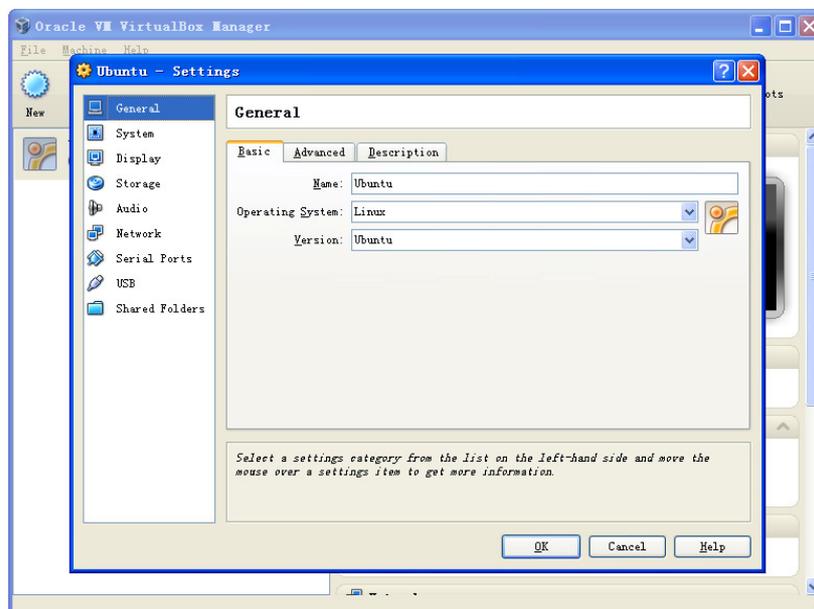
2) Select Storage on the left in the Settings window and click the CD icon next to the option Empty under IDC controller in the right part of the window, and then find the ISO file you downloaded
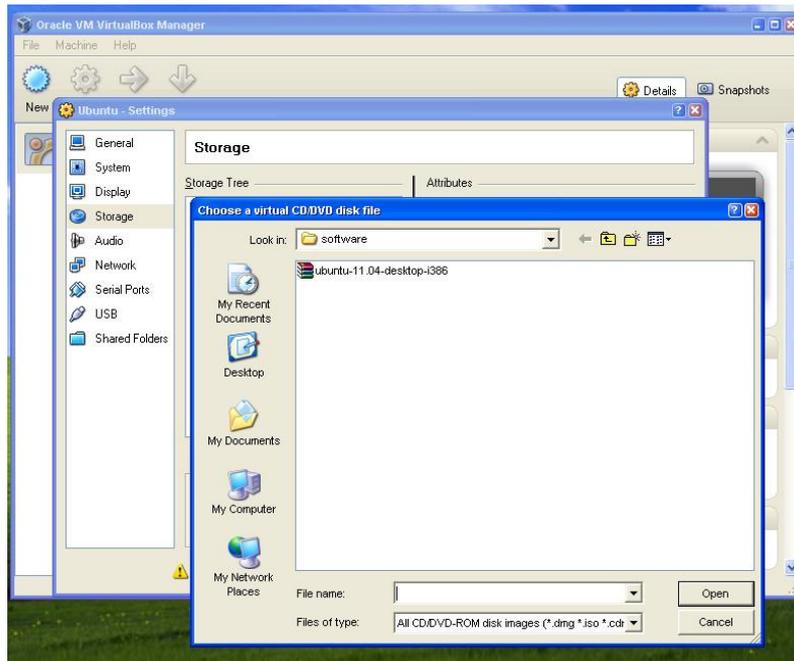
3) Select the ISO file you downloaded and click OK as shown below
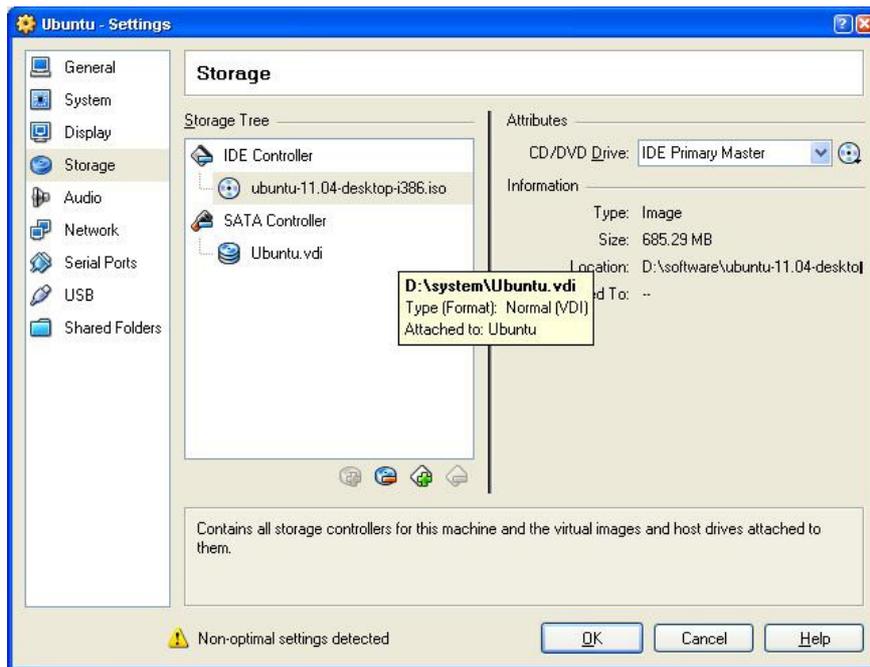
4) Click Start on the VirtualBox window, the Ubuntu installation program will start as shown below:
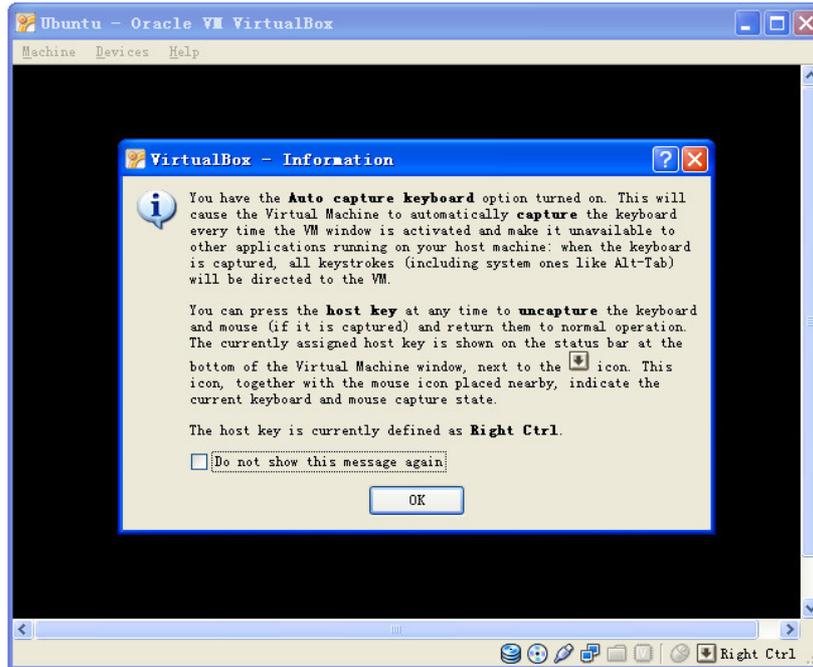
5) Some prompt windows will pop up during the initiation process. You just need only click OK all the way to the end of the process.

6) Click Install Ubuntu to start installation when the following window appears
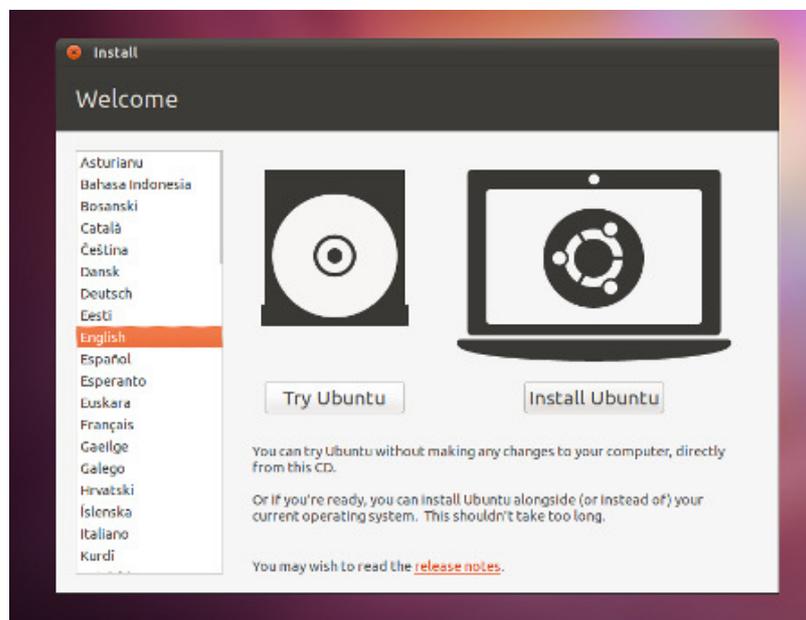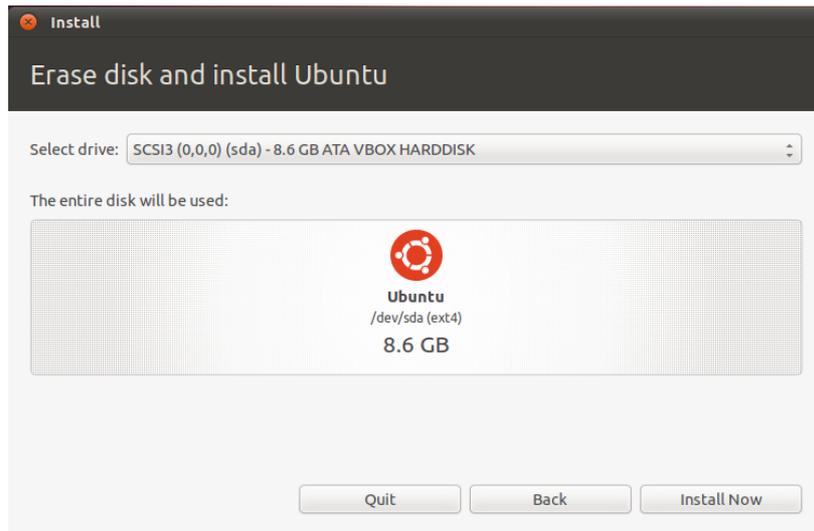

7) Click Forward to continue the process
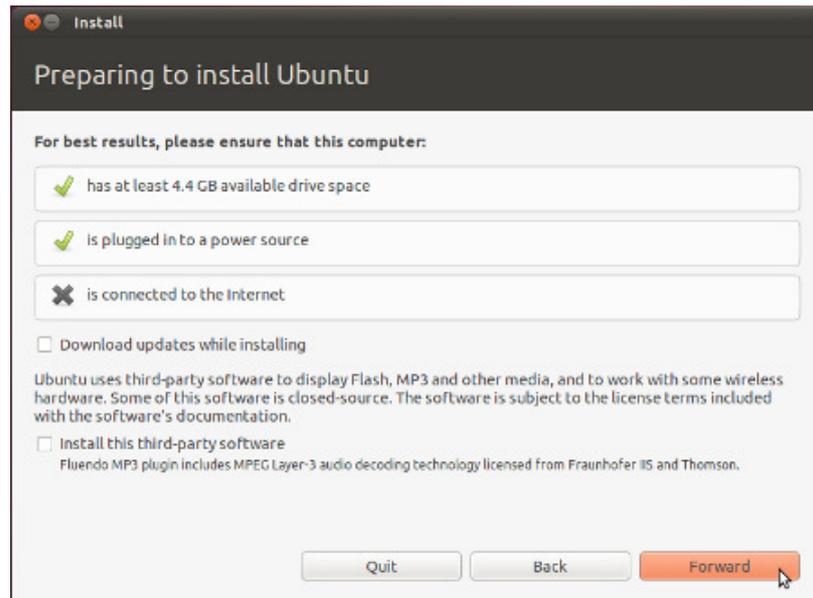
8) Select Erase disk and install Ubuntu and click Forward
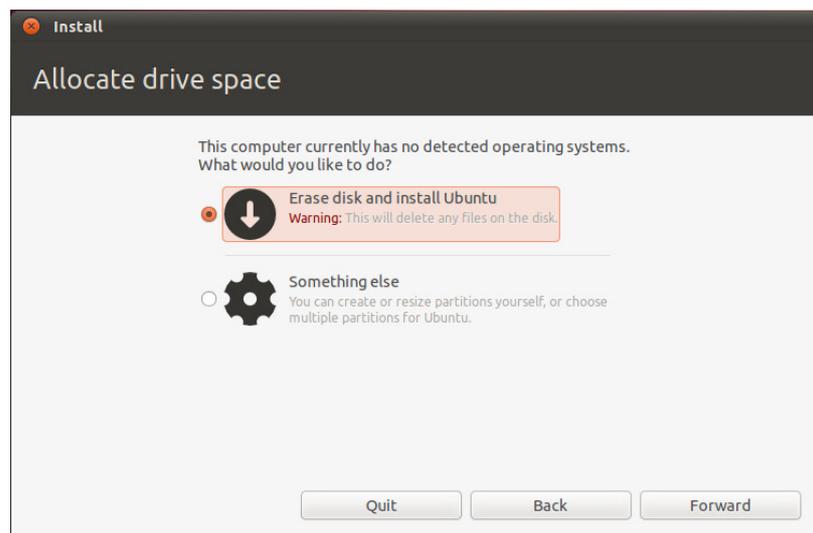
Note:

📖 Selecting this option will only affect the virtual hard drive you created earlier and will not lead to any content loss on your hard drive.

9) Click Install Now in the following window to start installation:

10) Some simple questions need to be answered during the installation process. Please enter appropriate information and click "Forward". The following window is the last question that will appear during the process:
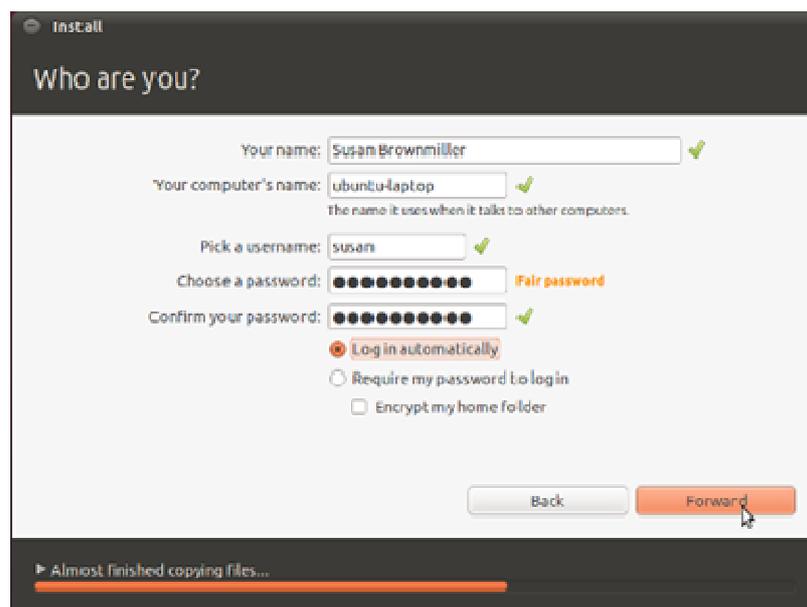
11) After all the required information is properly entered in to the fields, select Log in automatically and click forward.

12) The installation of Ubuntu may take between 15 minutes to an hour depending on your PC's specification. A prompt window will be shown as below after installation is done. Please select Restart Now to restart Ubuntu system.

13) The Ubuntu system is ready for use after restarting. Normally the ISO file shown below will be ejected automatically by VirtualBox after restarting Ubuntu. If it is not, you can eject the ISO file manually in the Settings window of VirtualBox. The following window shows the settings window after the ISO file is ejected.