Software Development Guide for touch-screen display

terasIC
www.terasic.com

CONTENTS

Chapter 1

Overview

This tutorial describes how to quickly develop a touch-screen GUI software application based on Terasic's Linux Console BSP (Board Support Package). This Linux Console provided by Terasic has a built-in framebuffer driver for LCD display and **QT** GUI library for graphic display, so developers can quickly design a GUI program even without X-Windows support. The Linux Console also has a built-in a touch-screen driver for touch-screen and **tslib** touch-screen library, so developer can easily implement touch function. The BSP supports two Terasic touch-panel boards: VEEK-MT-SoCKit and DE1-SoC-MTL.

In software development, the touch-screen GUI program is designed and cross-compiled on a Host PC, and the cross-compiled binary code will be executed on the target – Altera SoC ARM. To speed up GUI program design, the GUI-based **QT Creator** IDE (Integrated Development Environment) tool is used on Host PC for GUI layout, program coding and project cross-compile. The GCC (GNU Compiler Collection) is used for cross-compile, so a Linux development is required. Throughout this document, we will call the Linux running on PC for cross-compiling as **Host Linux**, and call the Linux running on Altera SoC board as **Altera SoC Linux**.

In the tutorial, we show developers how to set up development environment in Host PC, and how to design and cross-compile a GUI program for the target processor – Altera SoC ARM. The development environment setup steps include installing **VMware** player on Windows, and installing Linux on the VMware. This document also describes the details about how to cross-compile the **tslib** touch-screen library and **QT** GUI library for Altera SoC ARM.

This tutorial is focused on the GUI programing development. For the knowledge about how to create controllers in FPGA and how control FPGA in Linux program, please refer to manual for developing Control Panel in Altera DE1-SoC CD. This CD is available in the web page:

http://cd-de1-soc.terasic.com.

1.1 Linux BSP

Figure 1-1 shows the Terasic's Linux BPS for touch-screen display module. In HPS, the Linux running on ARM supports framebuffer driver for LCD display and touch-screen driver for touch-screen panel. The BPS also includes **QT** GU library and **tslib** touch-screen library. In FPGA site, Altera VIP (Video and Image Processing) suite is used to design LCD displayer controller, and Terasic touch-screen controller is used to control touch-screen.

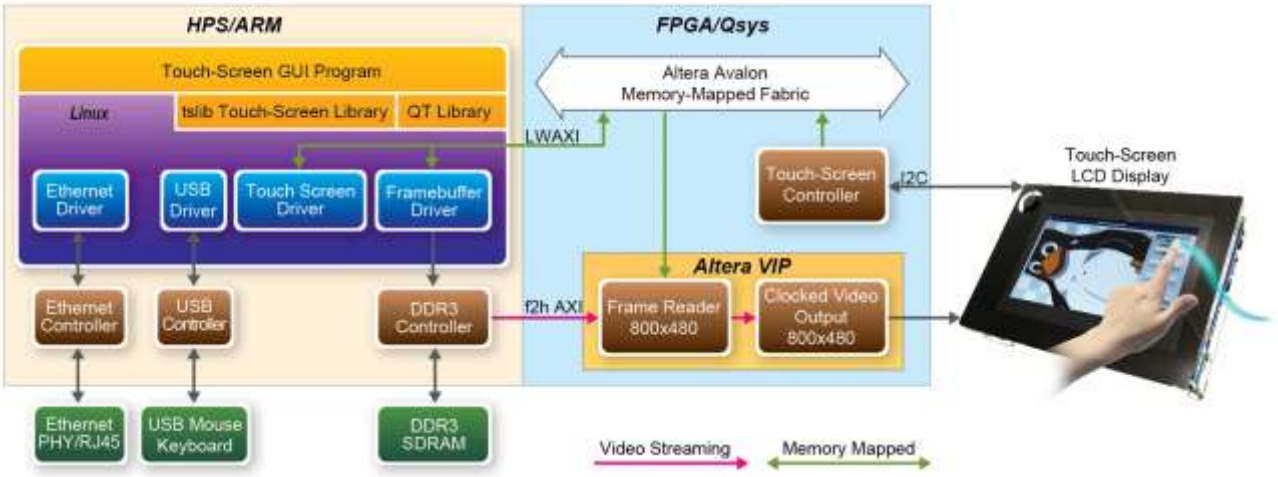

Figure 1-1 TERASIC Linux BPS for Touch-Screen LCD Display Module

Below describes where to download the required Linux BSP for various Altera SoC FPGA boards. In this tutorial, Linux Console microSD card image is used. Actually, LXDE Desktop also build-in the same display and touch functions. If LXDE Desktop is used, developers need to enter Linux terminal (press CTRL+ALT+F1) before running any application software used in this tutorial, and press CTRL+ALT+F7 can return to LXDE Desktop. Note that 8GB microSD card is required for LXDE Desktop.

■ Linux BSP for VEEK-MT-SoCKit

VEEK-MT-SoCKit is composed of SoCKit FPGA main board and HSMC-interfaced MTLC touch-screen display module. Its Linux BSP is available on the web page:

http://cd-sockit-upgrade.terasic.com

In this tutorial, developers need to download the VEEK-MT-SoCKit CD-ROM and the prebuilt

microSD card image for Linux Console. On the web site, the image is entitled as: Linux Console with framebuffer, and its file name is SoCKit_FB__MTLC.zip. **Table 1-1** shows the Linux BSP relative items includes in the CD-ROM.

Table 1-1 SoCKit Linux BSP for touch-screen display module

Description	***Location***
tslib touch-screen library	Demonstrations/SoC/Libraries/ tslib-altera-soc.tar.gz2
QT library	Demonstrations/SoC/Libraries/ qt-4.8.5-tslib-altera-soc.tart.gz2
Quartus HPS-Qsys Project	Demonstrations/SoC/MTLC_HPS

■ Linux BSP for DE1-SoC-MTL

DE1-SoC-MTL is composed of DE1-SoC Main board and GPIO-interfaced MTL touch-screen display module. Its Linux BSP is available on the web page:

http://cd-de1-soc-mtl.terasic.com

In this tutorial, developers need to download the DE1-SoC-MTL CD-ROM and the prebuilt microSD card image for Linux Console. On the web site, the image is entitled as: Linux Console with framebuffer, and its file name is DE1SOC_FB_MTL.zip. **Table 1-2** shows the Linux BSP relative items includes in the CD-ROM.

Table 1-2 DE1-SoC Linux BSP for touch-screen display module

Description	***Location***
tslib touch-screen library	Demonstrations/SoC/Libraries/ tslib-altera-soc.tar.gz2
QT library for Altera	Demonstrations/SoC/Libraries/ qt-4.8.5-tslib-altera-soc.tart.gz2
Quartus HPS-Qsys Project	Demonstrations/SoC/MTL_HPS

1.2 System Requirements

Before starting this tutorial, please note that the following items are required:

- A x86 PC with MS Windows Installed
- microSD card and card reader, 4GB at least
- Altera SoC Board with touch-screen display: VEEK-MT-SoCKit or DE1-SoC-MTL
- VMware Player Installer
- Linux Ubuntu 12.04 x32 Installer

- Option: Source code of **tslib** touch-screen library
- Option: Source code of **QT** 4.8.5 project
- **QT** 5.2.0 Designer Installer

1.3 Steps to Build a Program

This tutorial shows developers how to build a touch-screen GUI program from scratch with the following steps:

1. Install VMware Player on Windows Host
2. Install a Host Linux x86 on VMware Player for cross-compiling
3. Install Altera SoC tool-chain on Linux x86
4. Install prebuilt touch-screen library, or optional cross-compile touch-screen library
5. Install prebuilt QT library, or optional cross-compile QT library
6. Install QT 5.2.0 on Host Linux x86 and design a GUI program
7. On Host Linux x86, cross-compile the GUI program and execute it on Altera SoC Linux

1.4 Boot Linux on Altera SoC FPGA Board

This section describes how to boot Linux on Altera SoC Board by using the prebuild microSD image provided by TERASIC. There are five steps involved. Please carefully follow the instructions below:

1. Download microSD card writer utility – **Win32 Disk Imager.**
2. Download prebuild microSD card image file for Linux Console from TERASIC web site.
3. Decompress the downloaded image file, and write the file into a microSD card with an ImageWriter Utility.
4. Make sure the TERASIC touch-screen LCD display module is connected to an Altera SoC FPGA board.
5. Insert the microSD card intro Altera SoC FPGA Board, connect a USB keyboard and mouse to the USB host port, then power on the board to boot Linux System.
6. Login in Linux with ‘root’ user name.

■ Download Win32 Disk Imager

To write the microSD card image into a microSD card, a writing tool is required. In this tutorial, **Win32 Disk Imager** utility is used. This utility is a shareware, users can download it from the web:
http://sourceforge.net/projects/win32diskimager

On the download page as shown in **Figure 1-2**, click "Download" to start the download process. The downloaded filename is "win32diskimager-v0.9-binary.zip".

Figure 1-2 Download Web Page of Win32 Disk Imager

Decompress the file and execute Win32DiskImage.exe to launch the tool as shown in **Figure 1-3**.

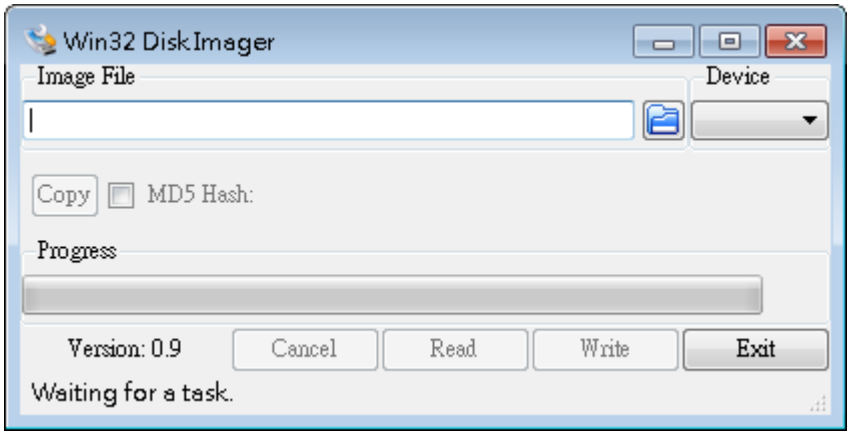

Figure 1-3 Win32 Disk Imager

■ Create Booting microSD card for Linux

First, we need to prepare the microSD card image. The required microSD card image for Linux console can be download from the web location specify the previous section 1.1 Linux BSP. The image file is in compression format, please decompress it first before programming it into a microSD card.

When microSD card image is ready, insert a microSD card into the host Windows. Launch Win32 Disk Imager, select the inserted microSD card in the Device list box and click on the Folder icon to select the image file DE1_SoC_FB_MTL.img for DE1-SoC-MTL board or SoCKit_FB_MTLC.img for VEEK-MT-SoCKit board. Then, click "Write" to start writing the image file into the microSD card as shown in **Figure 1-4.**

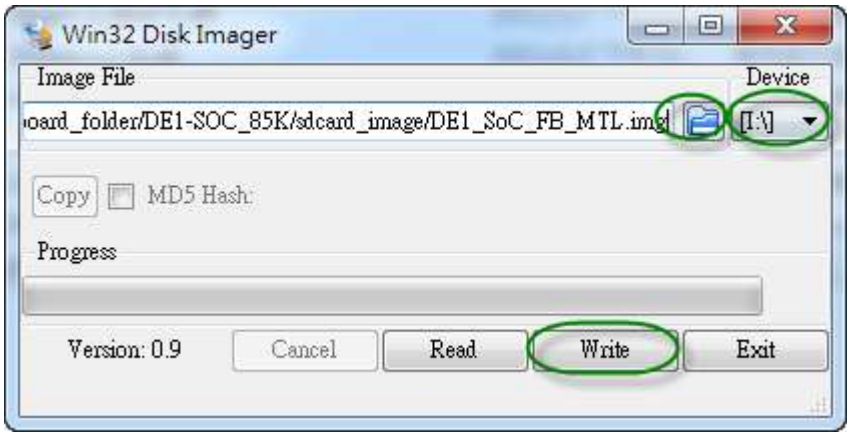

Figure 1-4 Write microSD card with a Given Image File

Before writing, a "Confirm overwrite**"** dialog appears as shown in **Figure 1-5**. Make sure the Target Device represents the microSD card, and click "Yes" to conform for overwriting.

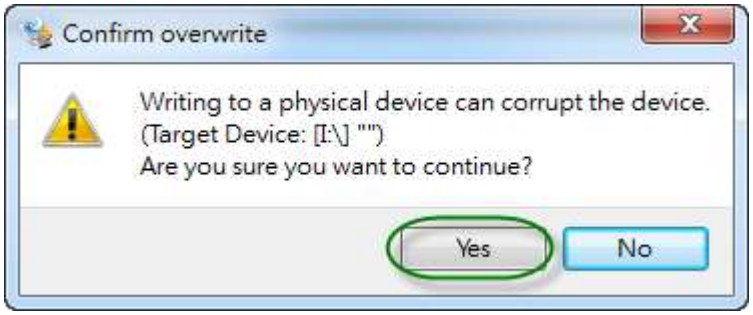

Figure 1-5 Confirm to Continue

When the writing process is complete, a “Complete” dialog appears as shown in **Figure 1-6**. Click “OK” to close the window.

Figure 1-6 Write SD-card Complete

■ Set up Hardware and Boot Linux

After booting microSD card for Linux is ready, connect the following items to the Altera SoC FPGA board.

- Terasic touch-panel LCD display module
- USB mouse and keyboard
- microSD Card for booting Linux
- DC Power

Figure 1-7 shows the connectivity with VEEK-MT-SoCKit and **Figure 1-8** shows the connectivity with DE1-SoC-MTL.

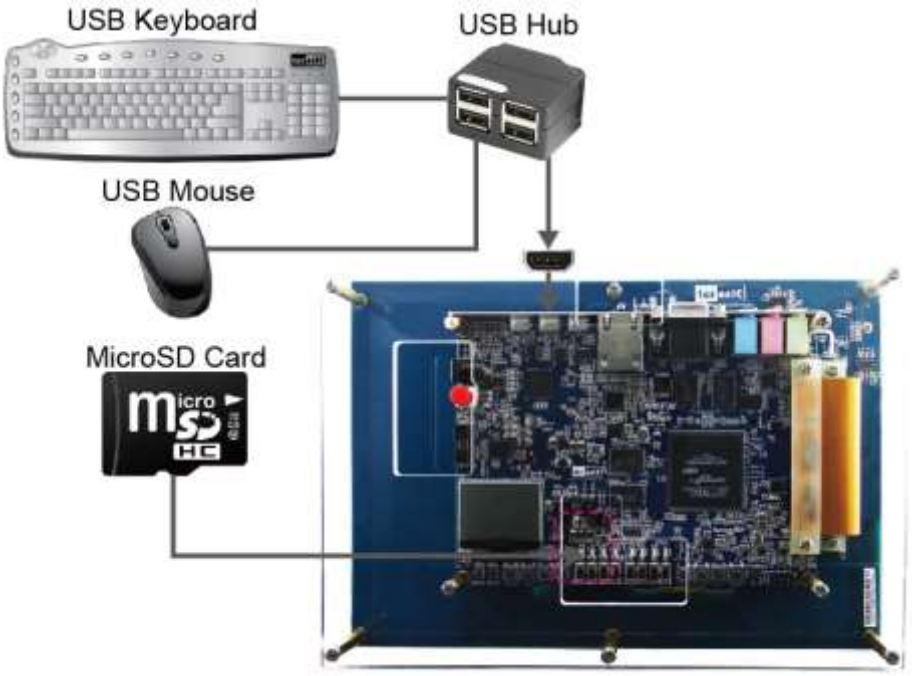

Figure 1-7 VEEK-MT-SoCKit Setup

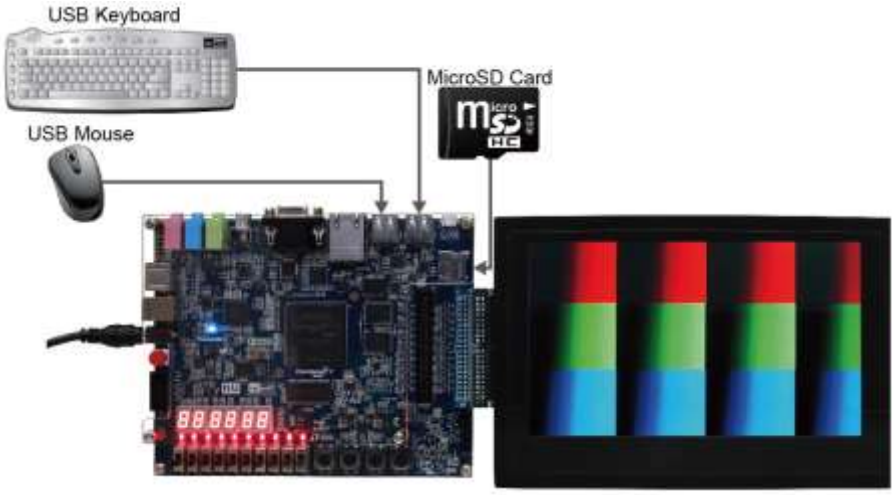

Figure 1-8 DE1-SoC-MTL Setup

Please also make sure that MSEL[4:0]=00000. **Figure 1-9** shows MSEL[4:0] location on VEEK-MT-SoCKit and **Figure 1-10** shows the MSEL[4:0] location on DE1-SoC-MTL.

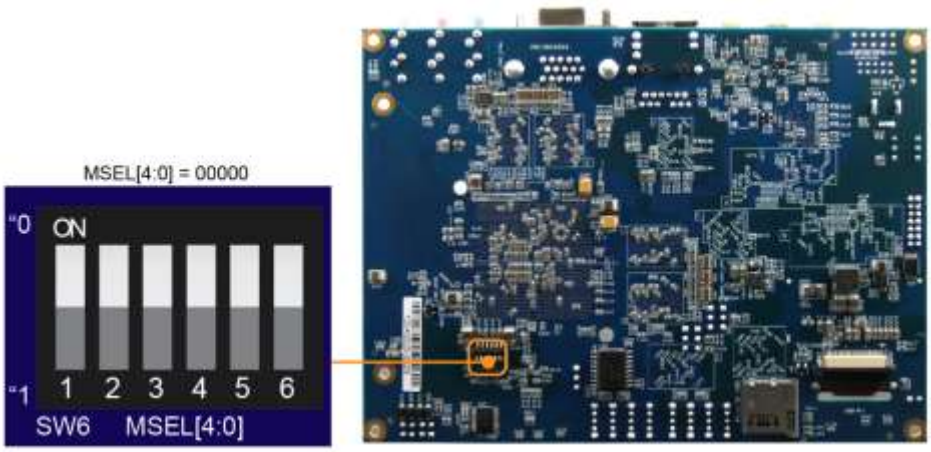

Figure 1-9 MSEL[4:0] on the SoCKit Board

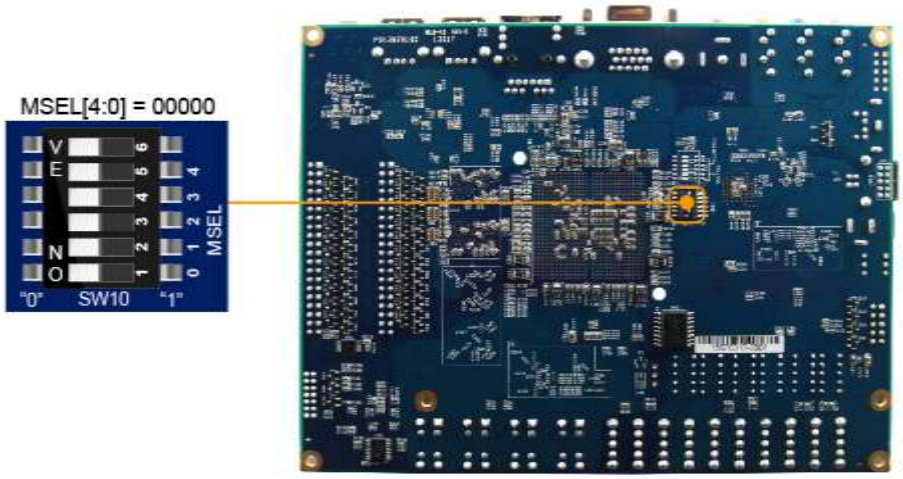

Figure 1-10 MSEL[4:0] on the DE1-SoC Board

After connecting the above peripherals, power on the Altera SoC FPGA board. The LCD display will now show Linux booting message as shown in **Figure 1-11**. When the Linux boot process is complete, a login prompt will appear. Type in "root" and press ENTER to login Linux.

Figure 1-11 Linux Booting Using Root Account

1.5 Execute Touch-Screen Demo

This section describes how to test the 'ts_test' application included in the **tslib** touch-screen library. The Terasic Linux BSP for touch-screen display already includes the **tslib** touch-screen library, so developer can test it without any extra installation effort.

In the Linux terminal, type "cd /usr/local/tslib-altera-soc/bin" to go to the bin folder of **tslib** library. Type "./ts_test" to launch **ts_test** program. Figure 1-12 shows the GUI of **ts_test** program shown in the LCD display. Click the "Draw" button to start drawing and click "Quit" button can terminal the **ts_test** program.

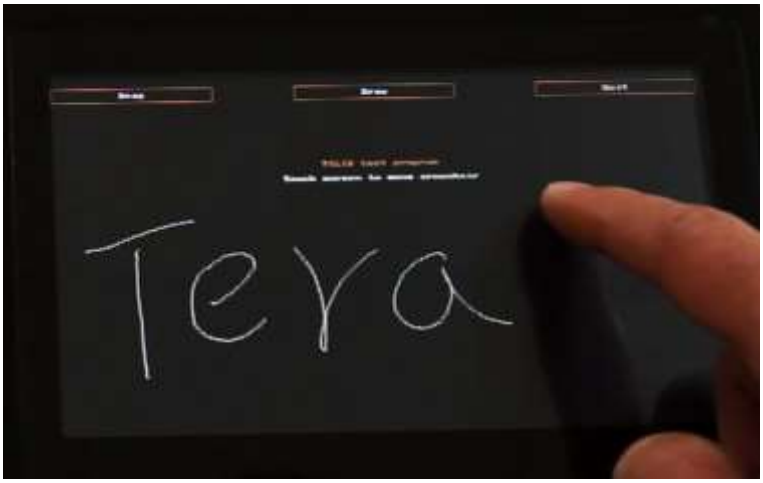

Figure 1-12 GUI of ts_test program

1.6 Execute QT Demo

Here will describe how to launch the **affine** demo in QT library. The Terasic Linux BSP for touch-screen display already includes the **QT** GUI library so developer can test it without any extra installation effort.

In the Linux terminal, type "cd /usr/local/qt-4.8.5-tslib-altera-soc/demos" to go to the demos folder of **QT** library, and type "cd affine" to go to the **affine** demo folder. Then, type "./affine -qws" to launch the affine demo. **Figure 1-13** shows the GUI of **affine** program shown in the LCD display. User can use touch-screen to adjust transformation setting.

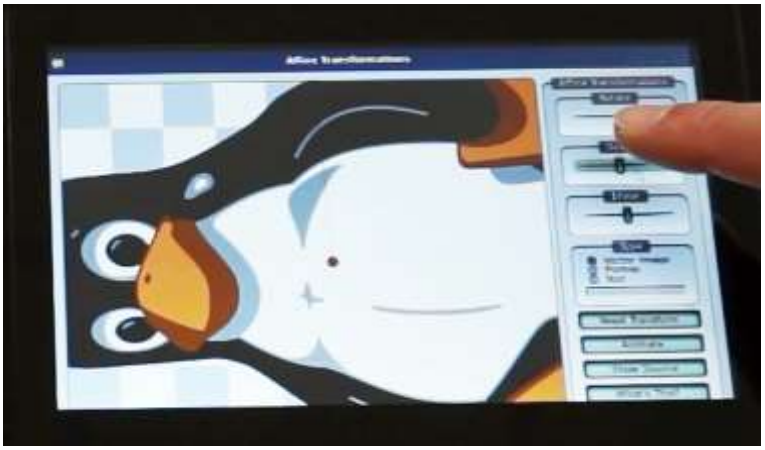

Figure 1-13 GUI of affine Demo

Chapter 2

Host Linux Installation

An Linux system is required for cross-compile GUI program for Altera SoC Linux. The Linux system can be installed on a x86 PC directly or on a virtual machine running on Microsoft Windows. This chapter describes how to install an Ubuntu OS, a Linux system, on a virtual machine VMware Player running on Microsoft Windows. The VMware Player is first installed on a Windows host, and subsequently Ubuntu Linux is installed on the VMware Player with an Easy Install mode. In this document, we call this Linux for performing cross-compiling as **Host Linux**. If developers already have an Ubuntu Desktop 12.04, they can skip this chapter except for the last section in this chapter.

Regarding the Linux installation, users might ask if it is possible to install Cygwin Linux in Microsoft Windows instead of installing Ubuntu on VMware Player running on Microsoft Windows. Note that QT Creator, a software meant for GUI-based programs, can't be successfully installed on Cygwin Linux in Microsoft Windows.

2.1 System Requirements

Before starting the Linux installation on a virtual machine, please make sure and check if these following items are in place:

- a PC with Microsoft Windows installed
- VMware Player installer (shareware)
- Linux Ubuntu 12.04 x32 .iso image file

2.2 Install VMware Player

Now, we proceed to install the virtual machine under Microsoft Windows. This section shows (1)

where to download VMware player and (2) how to install it under Microsoft Windows.

■ Download VMware Player Installer

Go to VMware download web page: https://my.vmware.com/web/vmware/downloads, find the VMware Player item and click "Download Product" as shown in **Figure 2-1**. In the VMware Player download page, click the "Downloads" button to download VMware Player, as shown **Figure 2-2** .

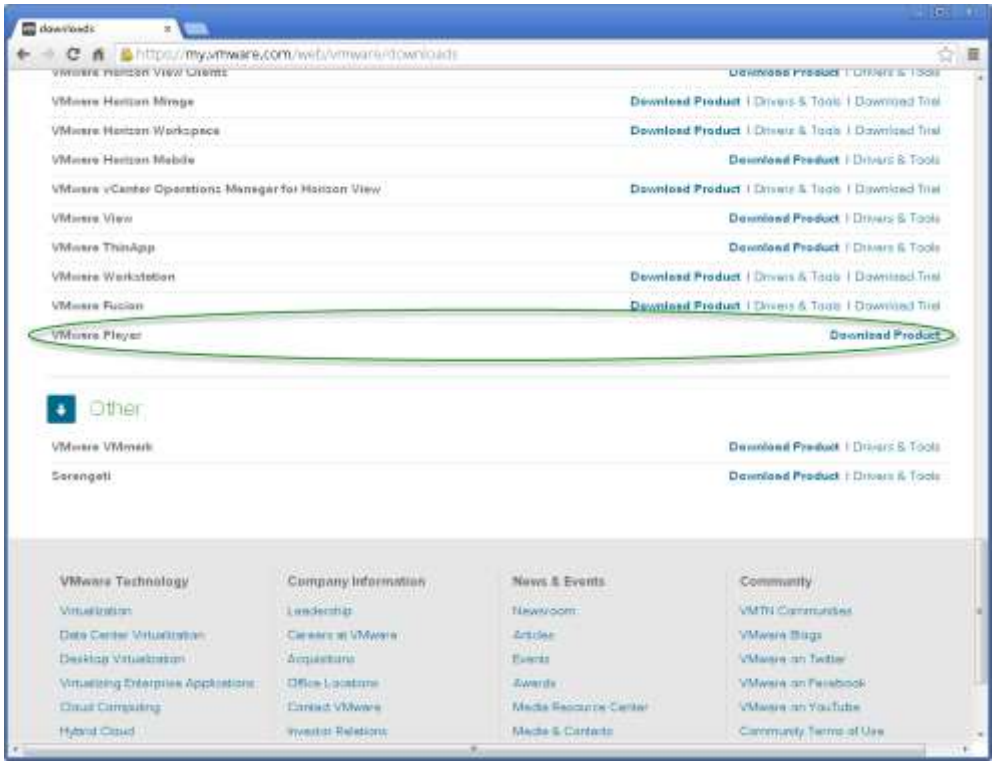

Figure 2-1 Download Web Page of VMware

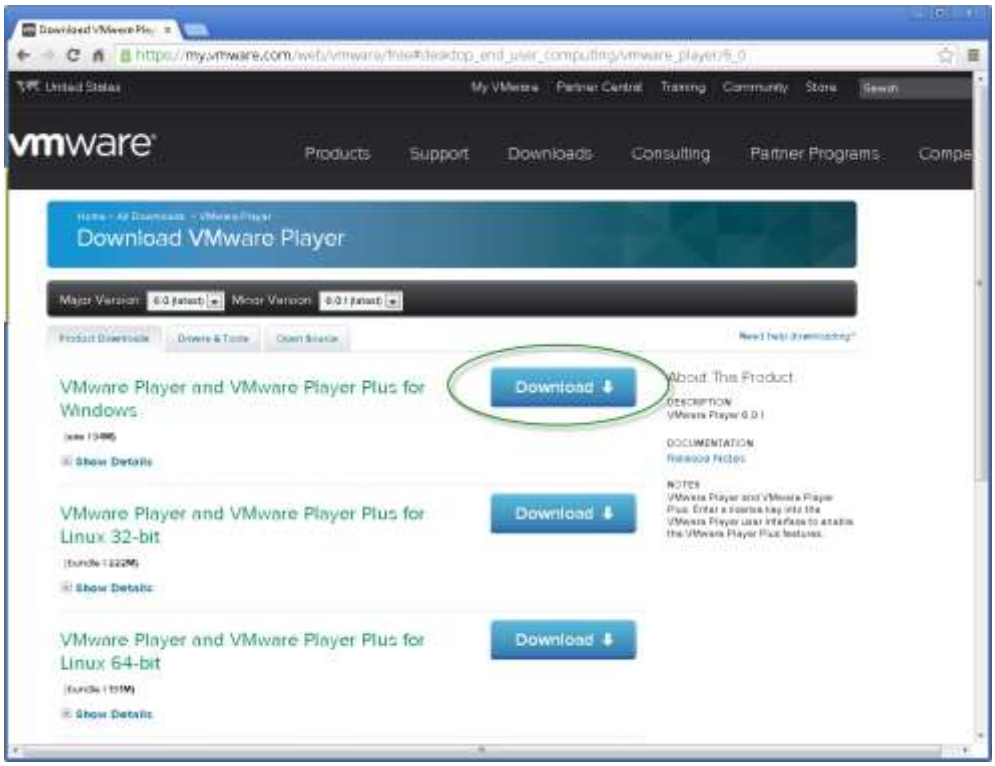

Figure 2-2 VMware Player Download Web Page

■ Install VMware

Under the Windows host, execute the downloaded installer "VMware-player-6.0.1-1379776.exe" to start the setup process. **Figure 2-3** shows the screenshot of the installer. Click “Next >” button to go to the next step.

Figure 2-3 Welcome Dialog

In the **License Agreement** dialog as shown in **Figure 2-4**, check "I accept the terms in the license agreement" and click "Next >" button to go to the next step.

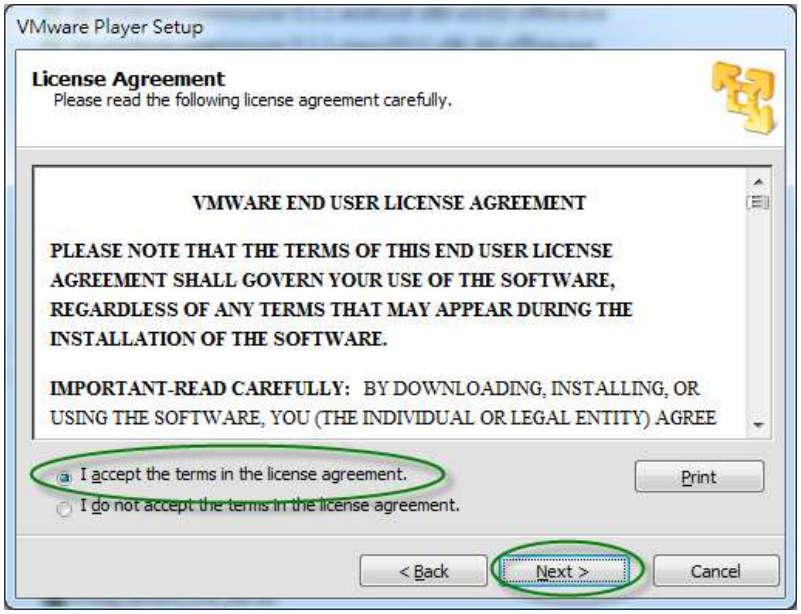

Figure 2-4 License Agreement Dialog

In the **Destination Folder** dialog as shown in **Figure 2-5**, keep the default destination or specify desired installation folder, then click "Next >" button to go to the next step.

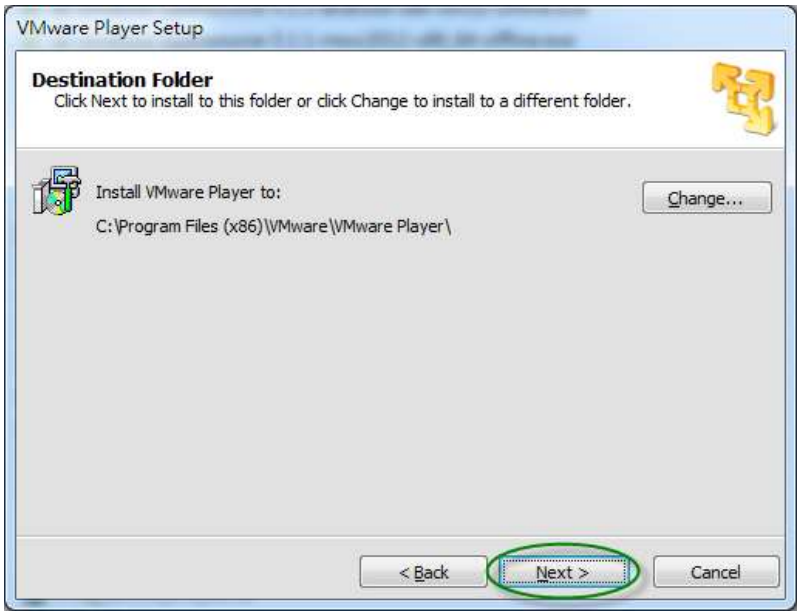

Figure 2-5 Destination Folder Dialog

In the **Software Updates** dialog as shown in **Figure 2-6**, check on the "Check for product updates on startup" and click "Next" to continue.

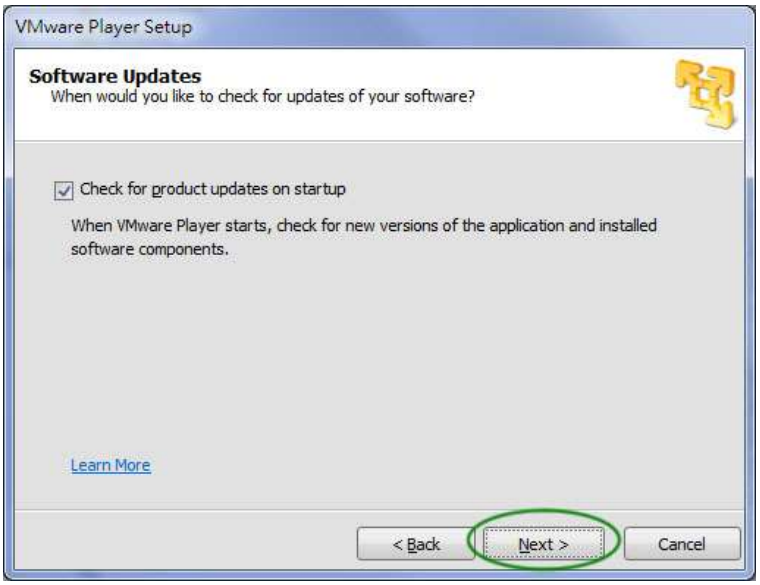

Figure 2-6 Software Updates Dialog

Next, you should see a "**User Experience Improvement Program**" dialog as shown in **Figure 2-7.** Check on "Help improved VMware Player" and click "Next" to continue.

Figure 2-7

In the **Shortcuts** dialog as shown in **Figure 2-8**, keep the default setting, then click "Next >" button

to go to the next step

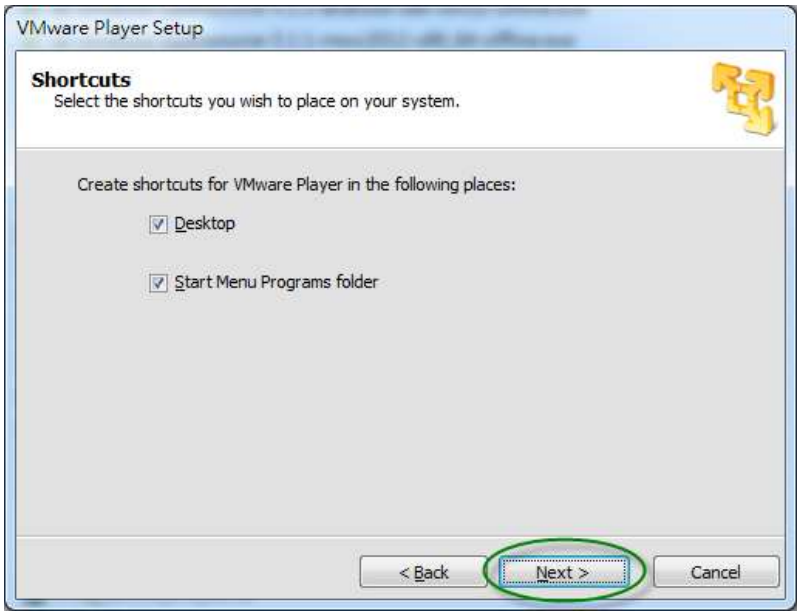

Figure 2-8 Shortcuts Dialog

In the **Ready to Perform the Requested Operations** dialog as shown in **Figure 2-9**, click "Next >" button to go to the next step

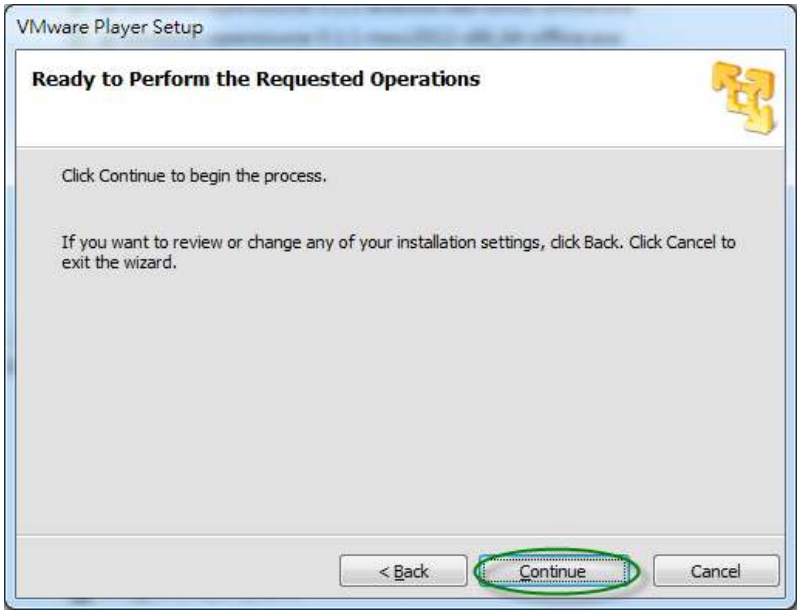

Figure 2-9 Ready to Perform the Requested Operation Dialog

When the installation is complete, **Setup Wizard Complete** dialog appears as shown in **Figure 2-10**. Click "Finished" button to finish the setup process.

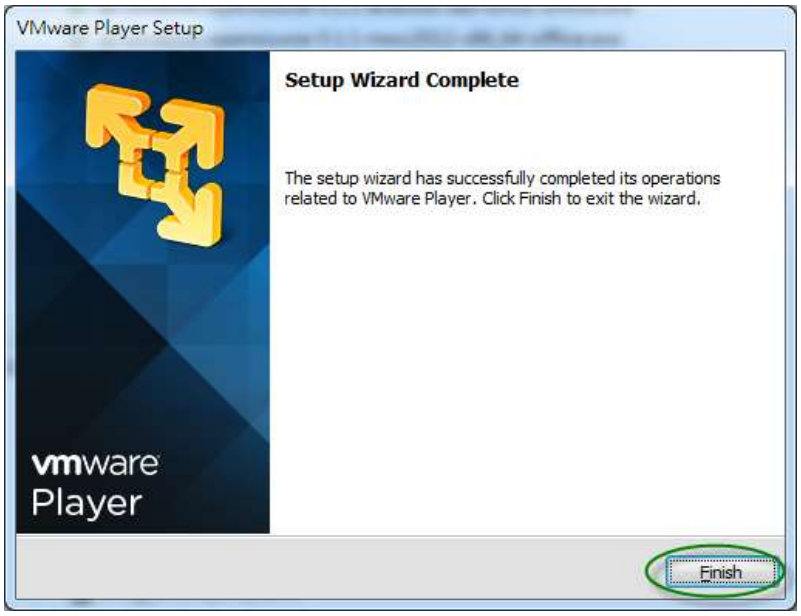

Figure 2-10 Setup Wizard Complete Dialog

2.3 Launch VMware

Once VMware Player is completely installed, a program shortcut icon will appear on the desktop in Windows, as shown in **Figure 2-11**. Double-click the shortcut icon will launch the VMware Player. **Figure 2-12** shows the main window of VMware Player.

Figure 2-11 VMware Player Shortcut on Windows Desktop

Figure 2-12 Main Window of VMware Player

2.4 Install Linux Ubuntu Desktop

Now we have downloaded and installed the VMware player, we are ready to download Ubuntu image file and install it on the VMware Player. Note that the “Easy Install” mode is automatically applied when VMware Player detects if the installed OS is Ubuntu 12.04.

■ Download Ubuntu Linux Image

The Ubuntu Desktop can be downloaded from the weblink provided below, as shown in **Figure 2-13**

http://www.ubuntu.com/download/desktop

On the download page, select “32-bit (for machines with less than 2GB)” and click "Ubuntu 12.04 LTS" button to start the download process.

Figure 2-13 Download Page of Ubuntu Desktop

A **Contribution** page appears as shown in **Figure 2-14**. If you do not wish to contribute now, click "Not now, take me to the download >" to start the download process. The downloaded image filename is "**ubuntu-12.04.3-desktop-i386.iso**".

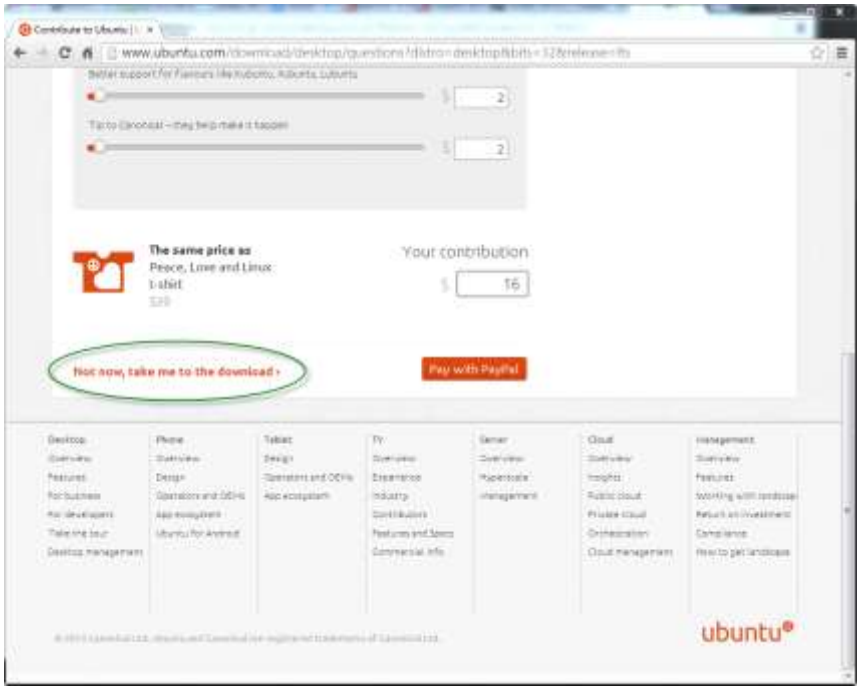

Figure 2-14 Download Page of Ubuntu Desktop

■ Create a Ubuntu Virtual Machine on the VMware Player

Launch the VMware Player as shown in **Figure 2-15**. Click the "Create a New Virtual Machine" Icon to create a virtual machine accordingly.

Figure 2-15 VMware Player Main Window

When the **Welcome to the New Virtual Machine Wizard** dialog appears as shown in **Figure 2-16**, select the “installer disk image file (iso):” radio button, and click “Browse” button to specify the location of the Ubuntu Linux image file “**ubuntu-12.04.3-desktop-i386.iso**” which has been downloaded in the previous step. Note that the “Easy Install” mode is enabled for Ubuntu 12.04. Click “Next >” to go to the next step.

Figure 2-16 Install Dialog

When the **Easy Install Information** dialog appears as shown in **Figure 2-17**, please specify your username and password. In this tutorial, the user name “terasic” and the password “123” are used. Click "Next >" button to go to the next step. **Please remember your username and password, as they will be frequently used throughout the tutorial.**

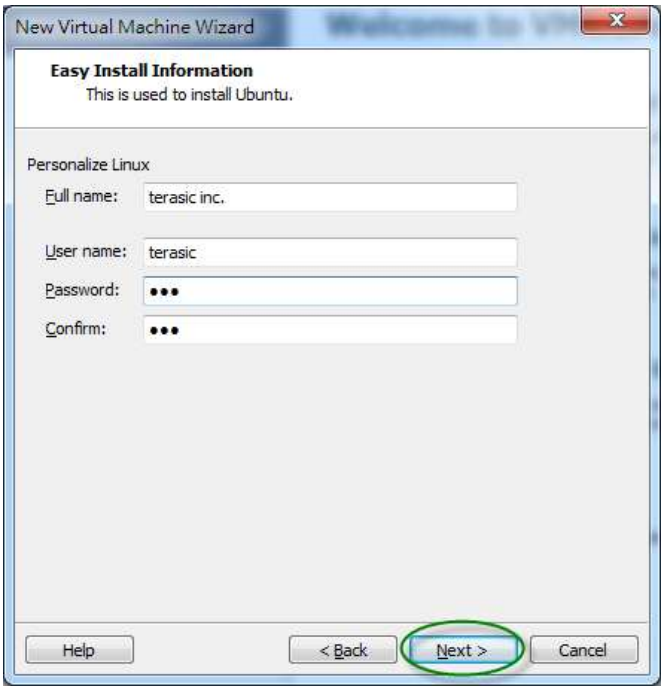

Figure 2-17 Easy Install Information Dialog

When the **Name the Virtual Machine** dialog appears as shown in **Figure 2-18**, you can change the virtual machine name as you wish. In this example, a machine name “Ubuntu 12.04 x32” is used.

Click the “Browse” button to specify a folder location for the virtual disk of the virtual machine. Finally, click "Next > " button to go to the next step.

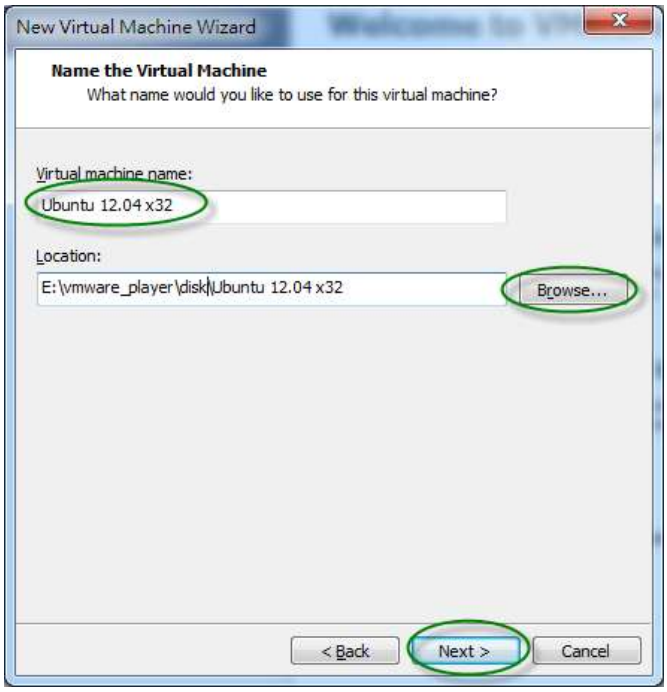

Figure 2-18 Name the Virtual Machine Dialog of New Virtual Machine Wizard

When the **Specify Disk Capacity** dialog appears as shown in **Figure 2-19**, please specify the maximum disk size and click "Next >" button to go to the next step.

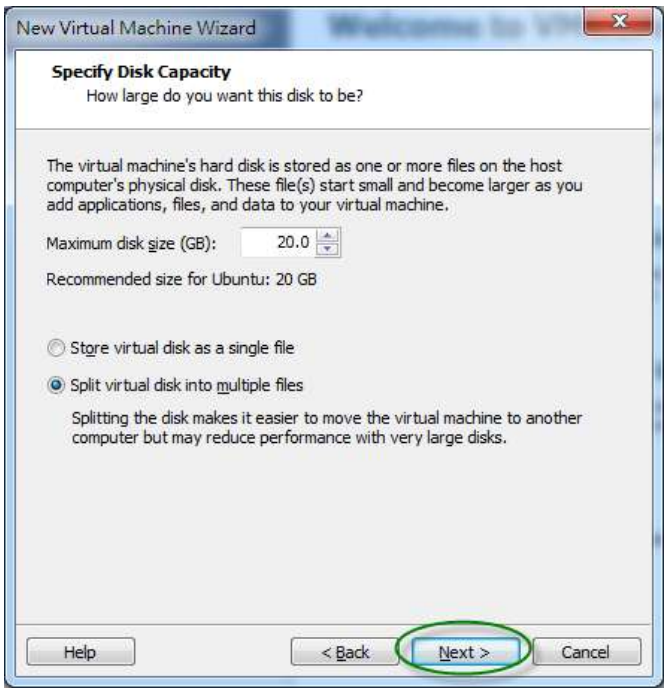

Figure 2-19 Specify Disk Capacity Dialog

Now, the **Ready to Create Virtual Machine** dialog appears as shown in **Figure 2-20**. Keep the default setting and click "Finish >" button to go to the next step.

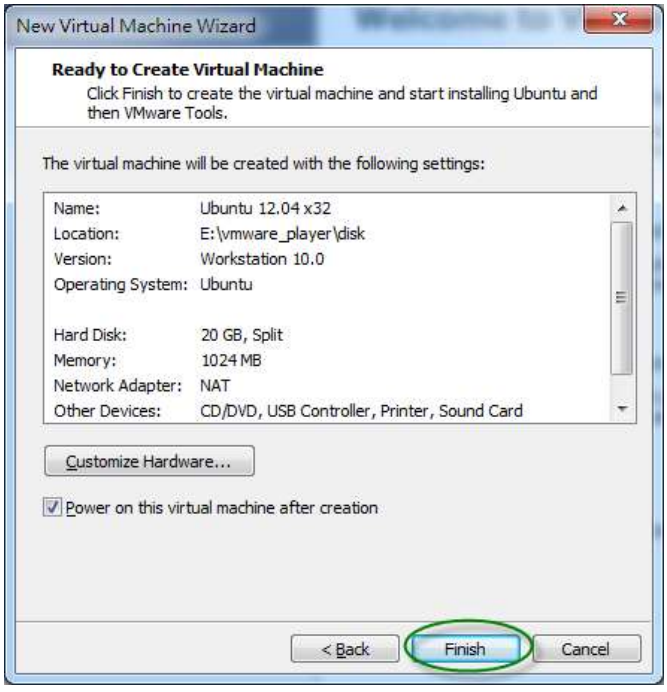

Figure 2-20 Ready to Create Virtual Machine Dialog

While installing, the **Software Updates** dialog appears as shown in **Figure 2-21**. Click the "Download and Install" button to proceed.

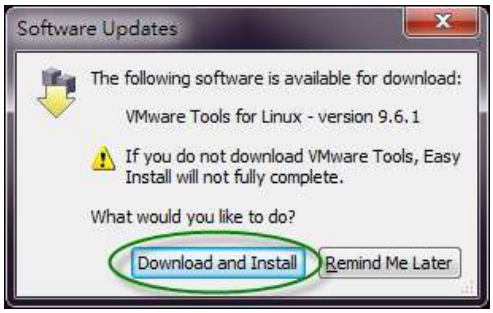

Figure 2-21 Software Updates Dialog

Figure 2-22 shows a screenshot of the installation progress.

Figure 2-22 Installation Progress

When the installation is completed, a Ubuntu login dialog appears as shown in **Figure 2-23**. In the Password edit box, key in the password specified in the previous step (in this tutorial, password is "123"), and press "ENTER" on the keyboard.

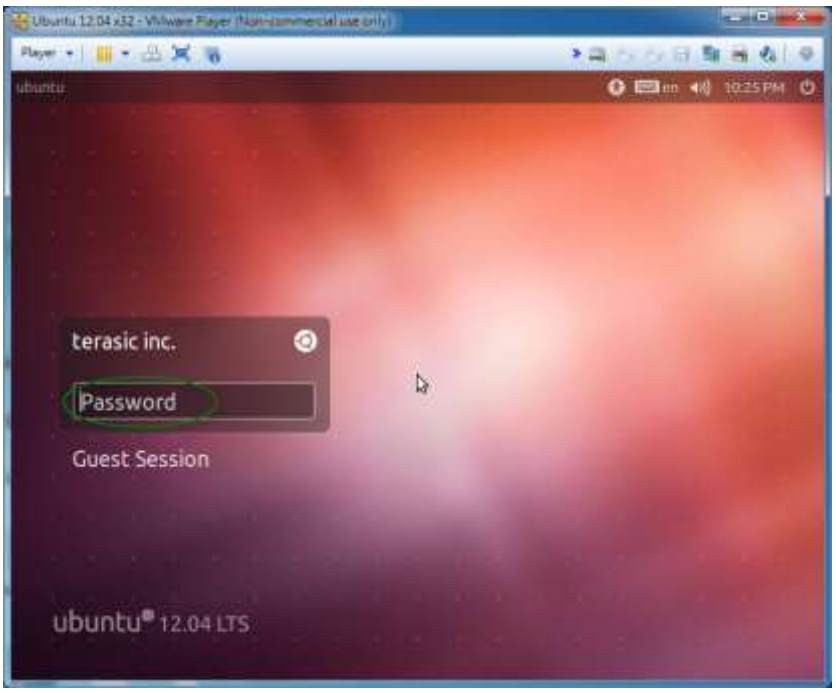

Figure 2-23 Ubuntu Login

After successful login, a Ubuntu Desktop appears as shown in **Figure 2-24**.

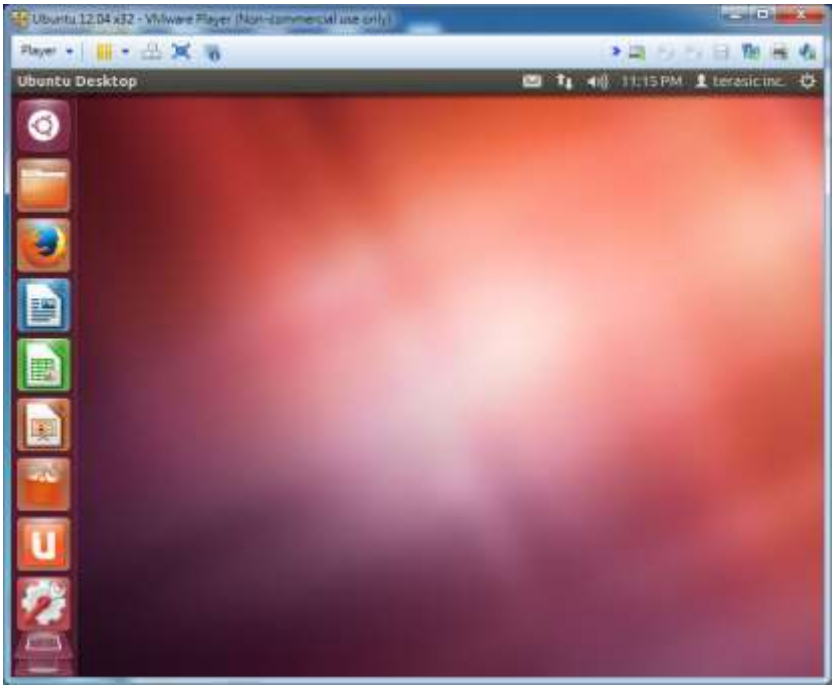

Figure 2-24 Ubuntu Desktop

Click the Power icon and select the "Shut Down…", as shown in **Figure 2-25** to turn off the Linux

system.

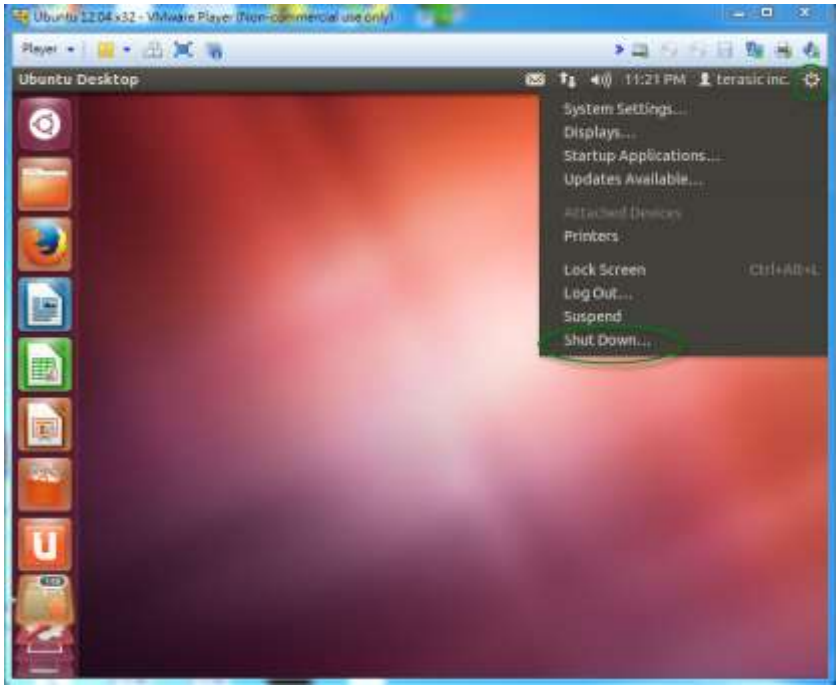

Figure 2-25 Shut Down Ubuntu Desktop

■ Restart Ubuntu

To restart Ubuntu after it has been shut down, here is the step-by-step instruction:

1. Launch VMware Player
2. Select the “Ubuntu 12.04 x32” item and click “Play virtual machine” button to start Ubuntu as shown in **Figure 2-26**.

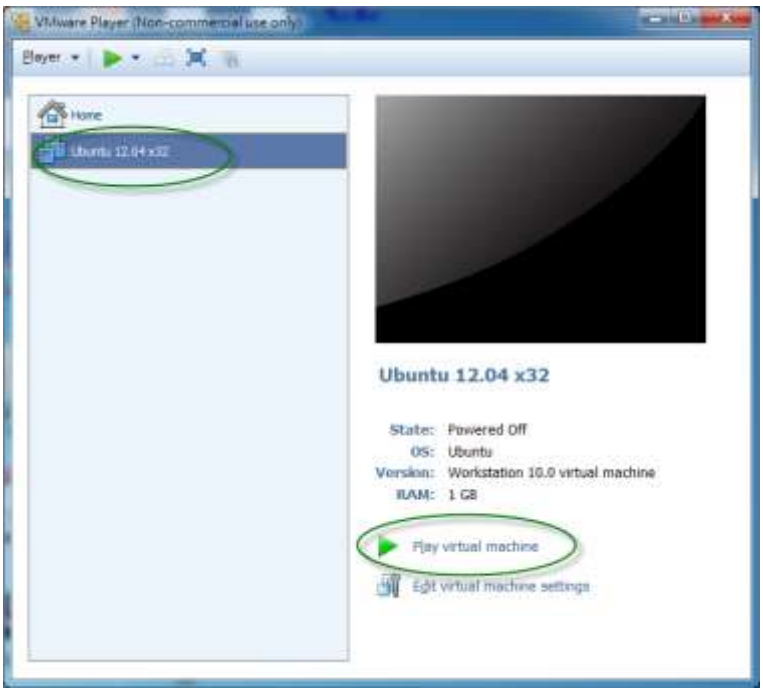

Figure 2-26 Start Ubuntu

2.5 Upgrade Linux Software Package

After Linux has been completely installed, we also need to upgrade the system to make sure the system is the most up-to-date.

Please follow carefully with the installation procedures in this section. First, press "CTRL+ALT+T" on keyboard to launch a terminal as show in **Figure 2-27**.

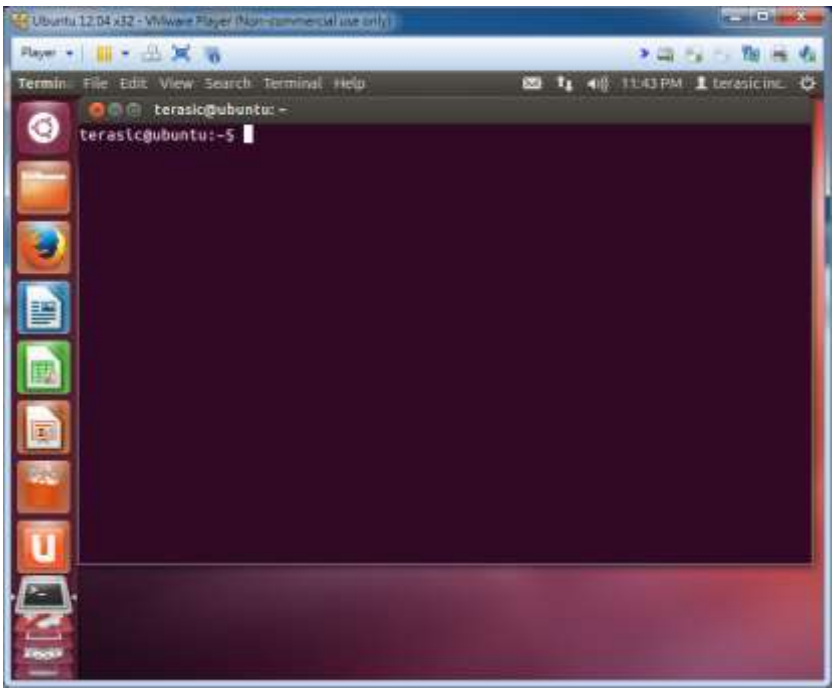

Figure 2-27 Linux Terminal

In the terminal, type the following command and press ENTER.

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

The system will prompt users to input a password. Please input your password (in this tutorial, the password is “123”) and press ENTER as shown in **Figure 2-28**.

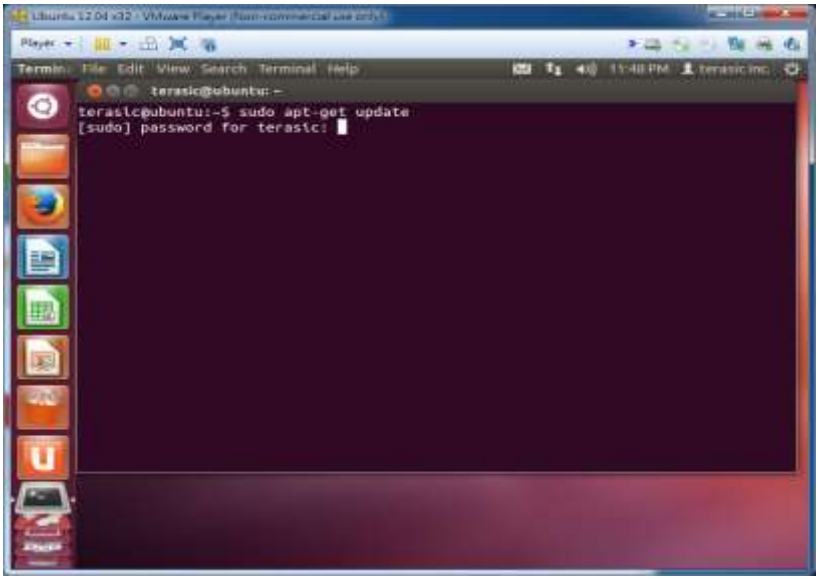

Figure 2-28 Input Password for TERASIC

After finishing typing in "sudo apt-get update' and the password, now type in the following command and press ENTER, as shown in **Figure 2-29**.

```
$sudo apt-get upgrade
```

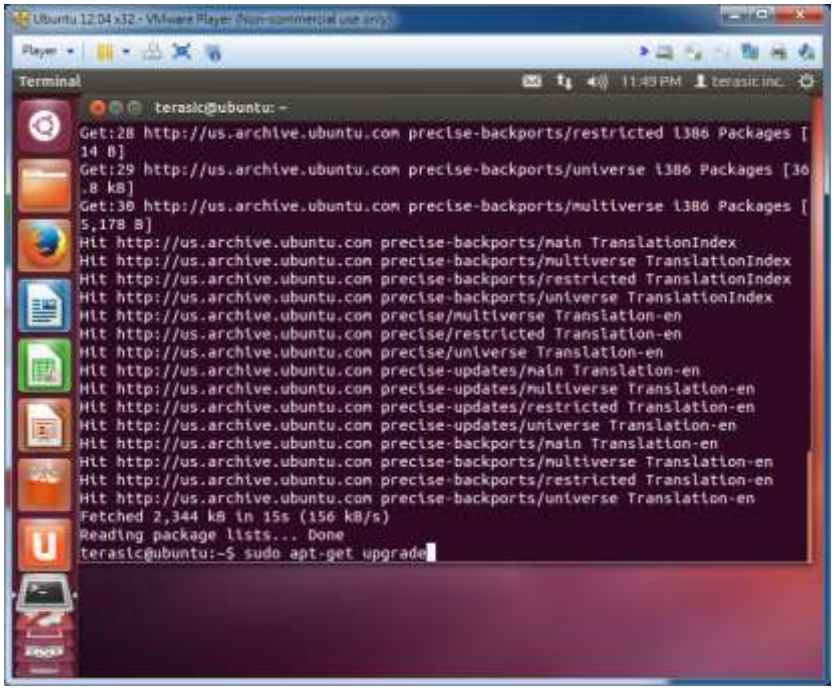

Figure 2-29 Typing in the Command "sudo apt-get upgrade"

System will prompt user to confirm the upgrade process. Please type "y" and press ENTER as shown in **Figure 2-30**. The upgrade process will take about 10~15 minutes.

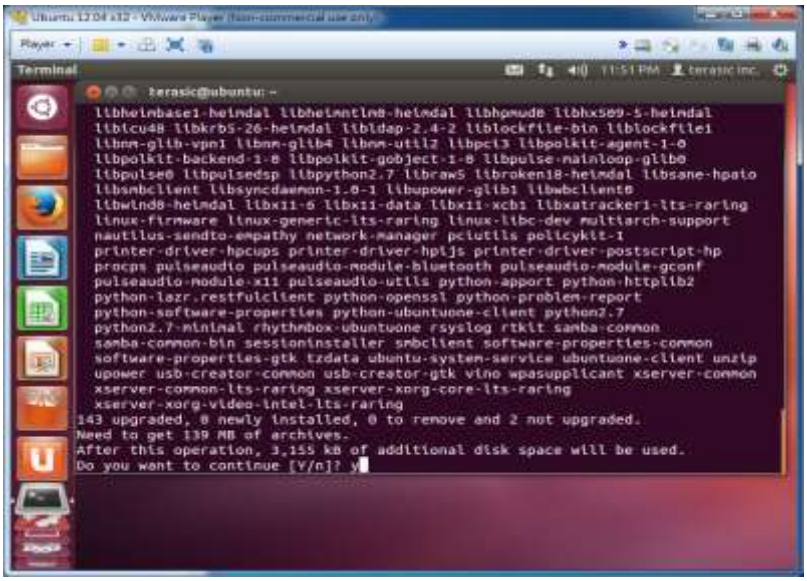

Figure 2-30 Type 'y' to Continue the Upgrade Process

Figure 2-31 shows the screenshot after the upgrade process has been complete.

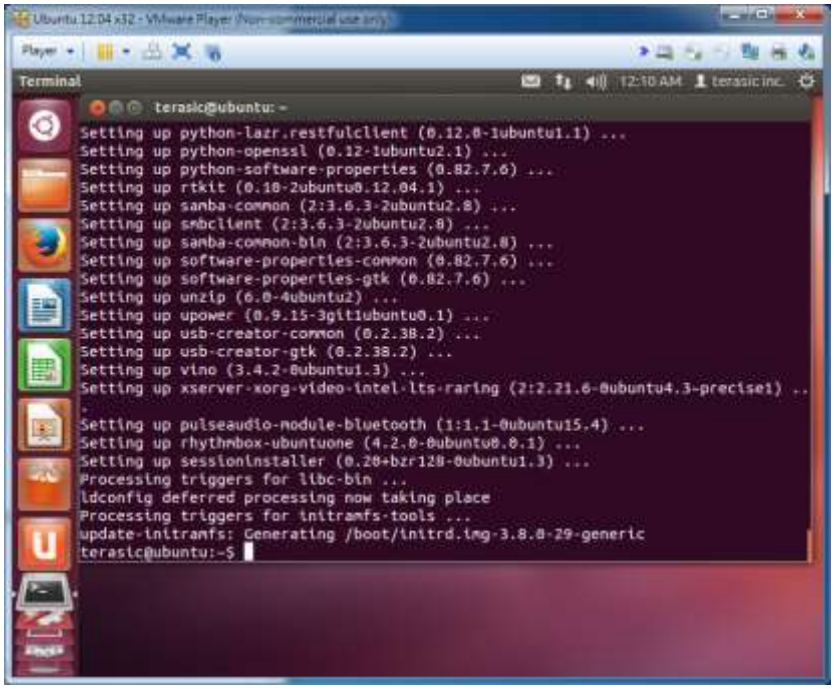

Figure 2-31 Upgrade Process Complete

Type "exit" to close the terminal, as shown in Figure 2-32.

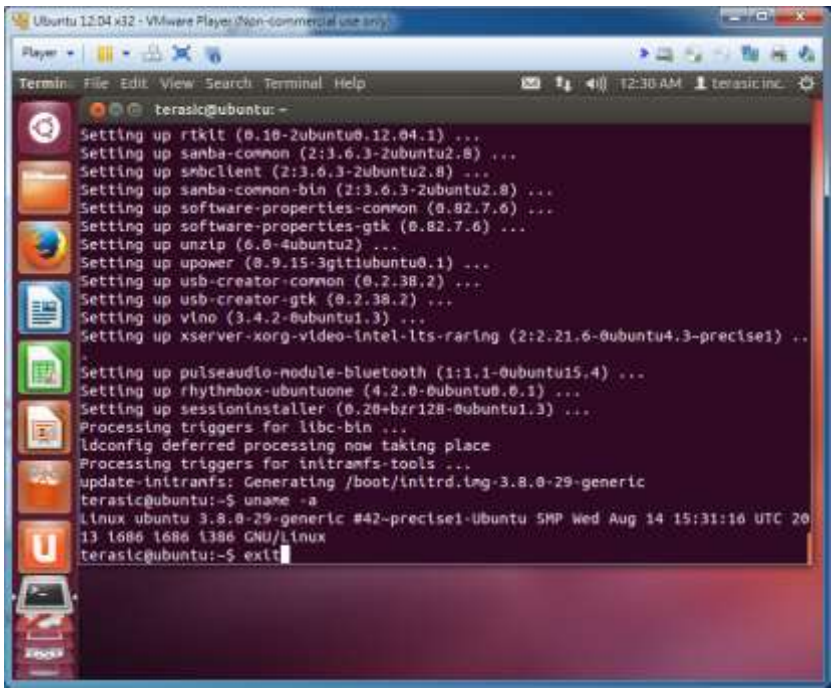

Figure 2-32 Type 'exit' to Close the Terminal

2.6 Install Software Package

To compile project some software should be installed. Please enter Linux terminal (CTRL+ALT+T) and execute the following commands and press ENTER to install required software.

```
$sudo apt-get install git autoconf automake libtool build-essential libgl1-mesa-dev
```

The system will prompt users to input a password. Please input your password (in this tutorial, the password is "123") and press ENTER as shown in **Figure 2-33**.

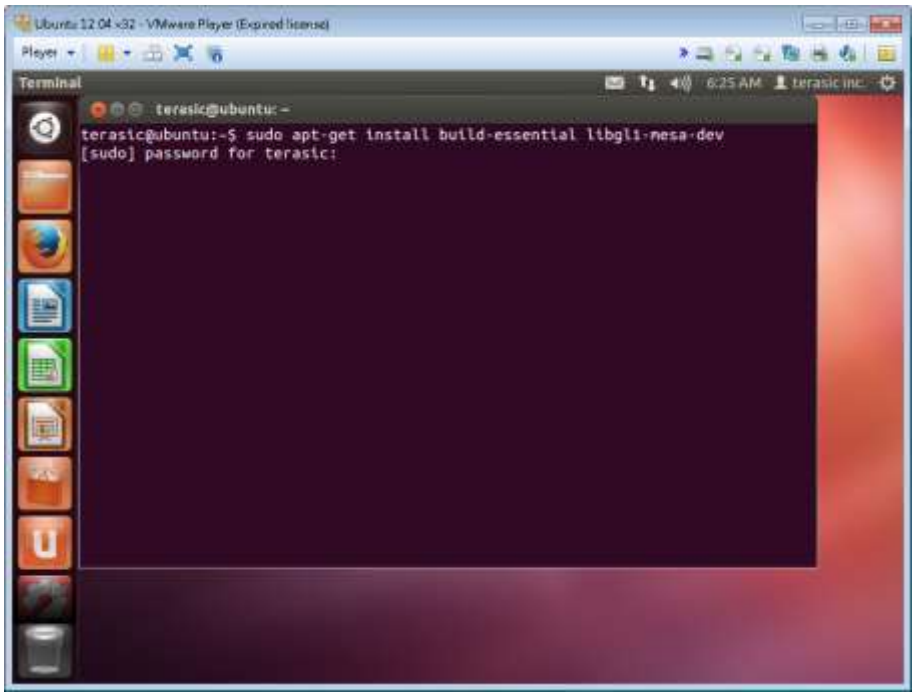

Figure 2-33 Install Software Packages for Building QT Project

Chapter 3

Altera SoC Tool-Chain Installation

Altera SoC tool-chain is required while building (cross-compile) the QT and touch-screen library for the Altera SoC ARM. In this chapter, we show the users steps needed to install the Altera SoC tool-chain. Here is a quick look of the 2 simple steps:

- Download the Altera SoC Tool-Chain and extract the file
- Include Tool chain path into the system environment variable $PATH

3.1 Download and Install Tool-Chain

Launch Linux terminal (CTRL+ALT+T) and execute the following commands to download and extract the Altera SoC tool-chain,

```
$cd ~
$wget https://launchpad.net/linaro-toolchain-binaries/trunk/2012.11/\
+download/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux.tar.bz2
$tar xjf gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux.tar.bz2
```

Figure 3-1 shows the screenshot of tool-chain download progress after typing in the wget command. It will take about 5 minutes to download the file.

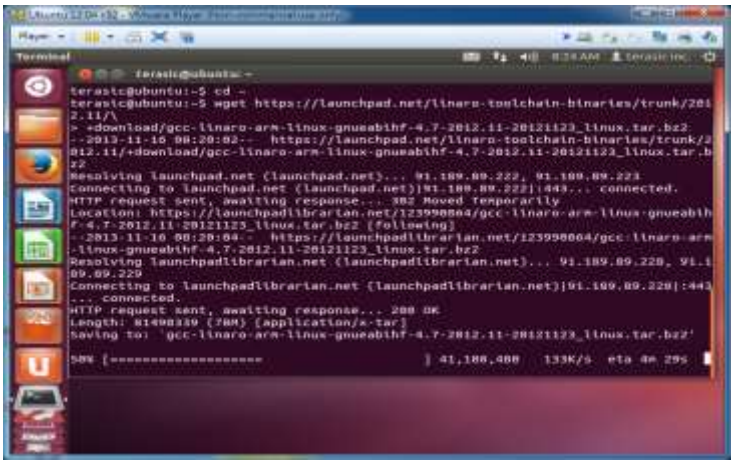

Figure 3-1 Altera SoC Tool-Chain Download Progress

The downloaded file is in the compressed format, so we need to decompress it before we can use it. **Figure 3-2** shows the screenshot after tool-chain download has been finished and has been extracted by the **tar** command. After extraction, the tool-chain is located in the folder "~/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux"

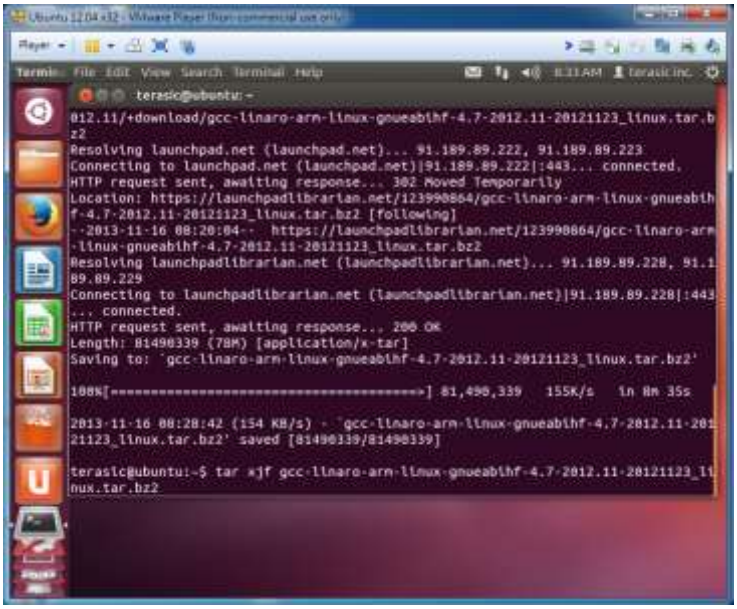

Figure 3-2 Altera SoC Tool-Chain Download Finished and Extracted by the Tar Command

3.2 Set up Tool-Chain Path

Now, the tool-chain path needs to be defined and added into the system variable **$PATH**. We use

the system editor tool **gedit** to add the path. Please refer to the following for more details to set up the tool-chain path.

Launch terminal (CTRL+ALT+T) and type in the following command to open the batch file "~./profile"

```
$gedit ~/.profile
```

Figure 3-3 shows the screenshot that the "~/.proifle" opened by gedit. Please add the following line,

```
export PATH=/home/terasic/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux/bin:$PATH
```

to the **end** of the batch file. Note, in the path string, you should replace "teraisc" with your linux user name. Click "save" icon to save the file followed by clicking "close" icon (red circle with a x on the upper left corner) to terminate gedit tool.

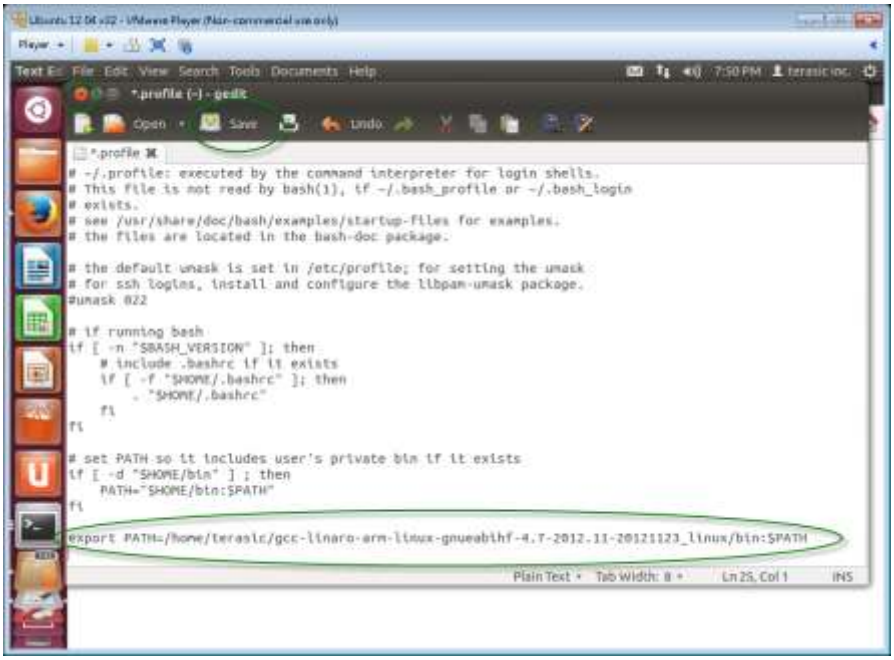

Figure 3-3 ./Profile Opened by gedit

In order for the path setting to take effect immediately, please type in "source ~/.profile" in the terminal or restart the OS.

Chapter 4

Build Touch-Screen Library

Up to now, you should have successfully installed the Altera SoC tool-chain in the Host Linux. Now we start building **tslib** touch-screen library for Altera SoC ARM on the Host Linux, and showing how to install the library on Altera SoC Linux. Developer can skip this chapter if they are not interested in it now.

Here is a quick look of the 3 simple steps:

- Download source code of **tslib** touch-screen library
- Configure, build and install touch-screen library
- Install **tslib** touch-screen library on Altera Soc Linux

4.1 Download Source Code of 'tslib' Library

In Linux terminal, execute the following commands and press ENTER to download the source code of **tslib** touch-screen library by **git** command.

```
$git clone https://github.com/kergoth/tslib
```

Figure 4-1 shows the screenshot after **tslib** source code download has been finished. The source code is saved in the "~/tslib" folder.

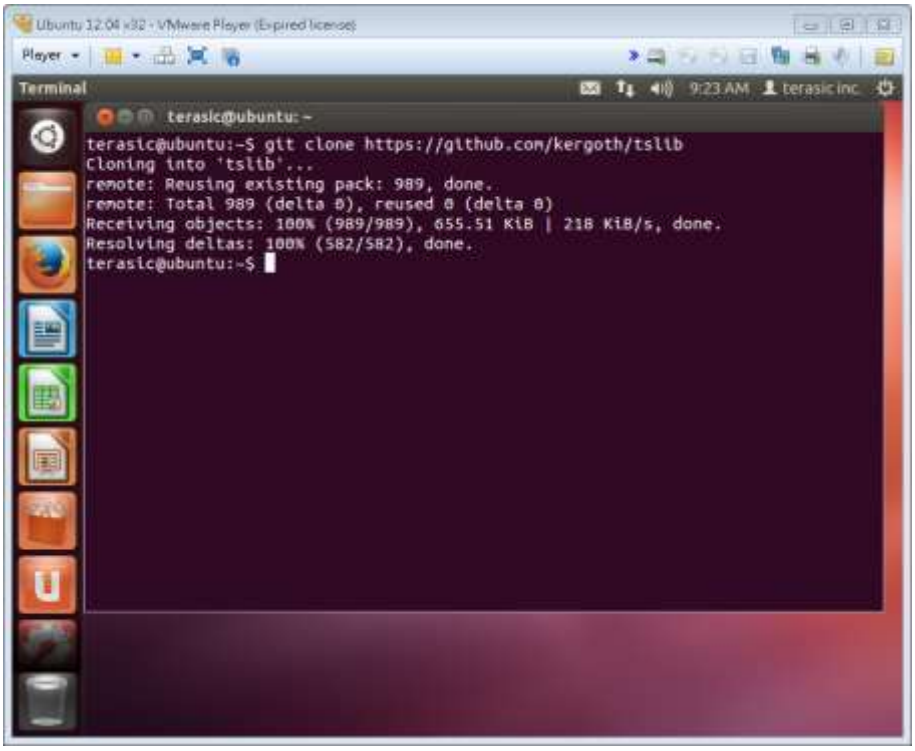

Figure 4-1 Complete Download the Source Code of tslib Project

4.2 Configure, Build and Install touch library

Now we are ready to configure, build, and install the **tslib** touch-screen library. Here is a list of the commands that will be used toward building and installing the **tslib** touch-screen library.

```
$export CC=/home/terasic/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux\
/bin/arm-linux-gnueabihf-gcc
$export CXX=/home/terasic/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux\
/bin/arm-linux-gnueabihf-g++
$cd ~/tslib
$./autogen.sh
$./configure --prefix=/usr/local/tslib-altera-soc --host=arm-linux
$make
$sudo make install
```

■ Configure

The following commands are used to specify the compilers which will be used to build **tslib** library in later.

```
$export CC=/home/terasic/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux\
/bin/arm-linux-gnueabihf-gcc
$export CXX=/home/terasic/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux\
/bin/arm-linux-gnueabihf-g++
```

To go to the source code folder of **tslib** library, type

```
$cd ~/tslib
```

The following command is used to generate files required for configure.

```
$./autogen.sh
```

Figure 4-2 shows the screenshot when this command is executed completely.

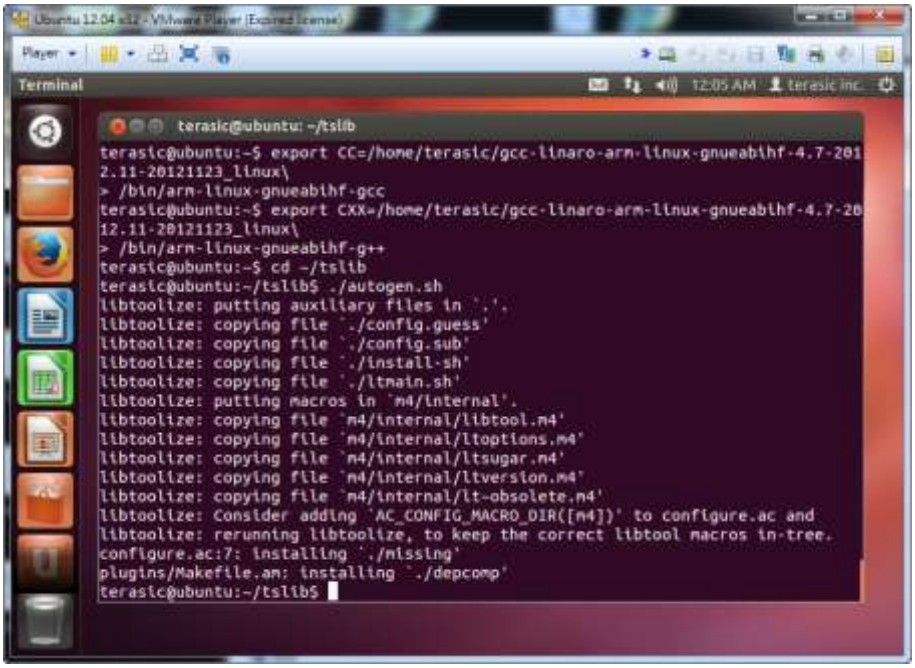

Figure 4-2 ./autogen.sh Executed Completely

The following command is used generate required makefile for compiler.

```
$./configure --prefix=/usr/local/tslib-altera-soc --host=arm-linux
```

Figure 4-3 shows the screenshot when this command is executed completely.

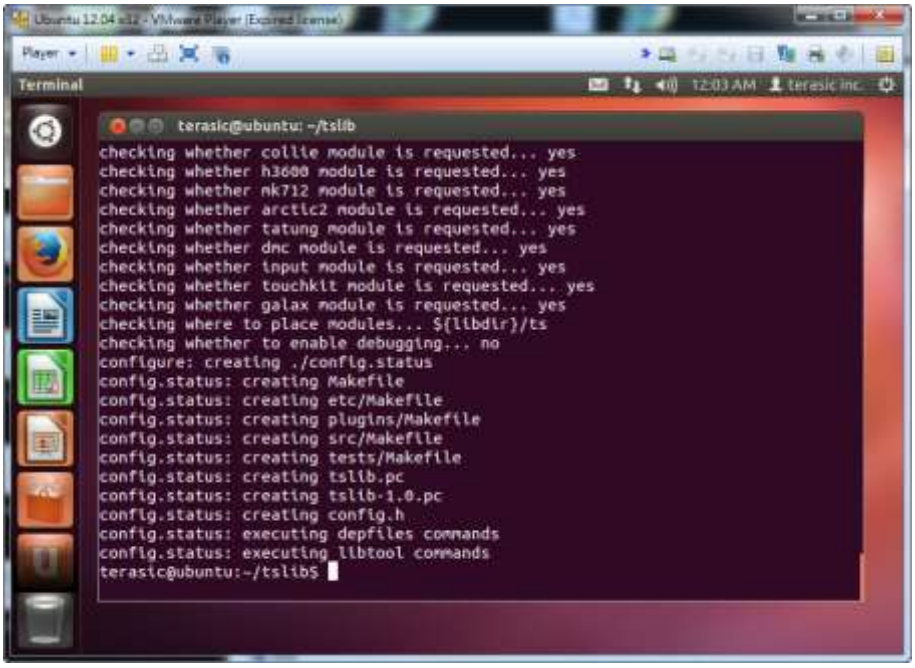

Figure 4-3 Configure Executed Completely

■ Make

In the terminal simply type:

```
$make
```

to build the tslib library as shown in **Figure 4-4**.

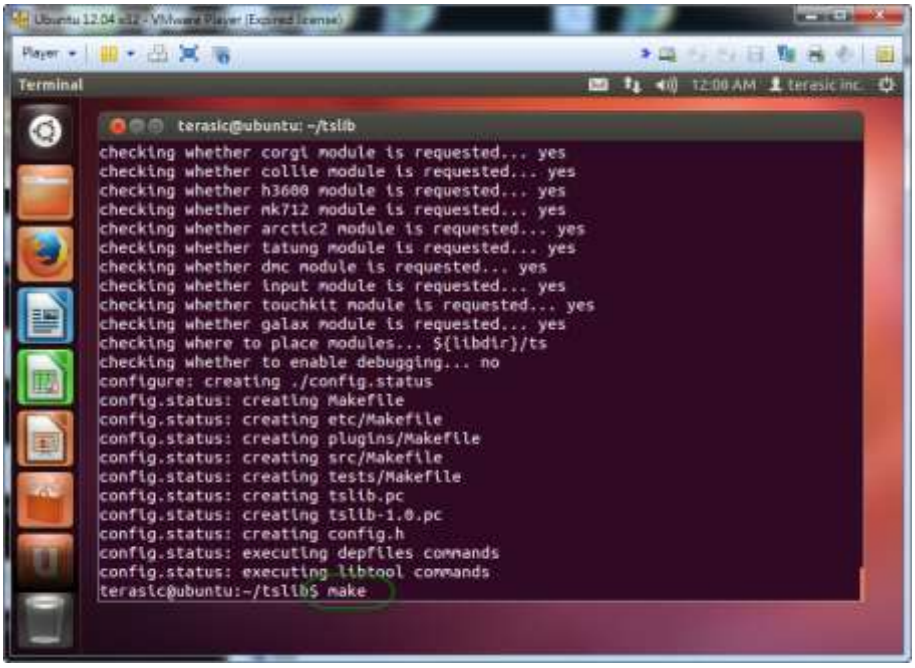

Figure 4-4 Type Make to Start the Build Process

Figure 4-5 shows the screenshot when the make process is completed.

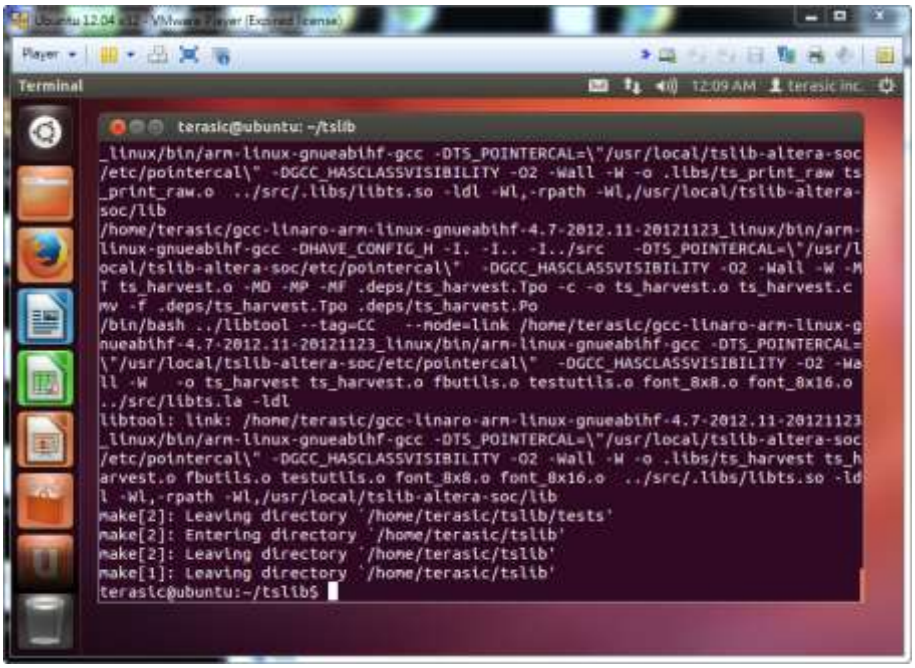

Figure 4-5 Make Process Now Complete

■ Install

Finally, to install the **tslib** library please type in

```
$sudo make install
```

and type in “123” when asked for password for Terasic as shown in Figure 4-6.

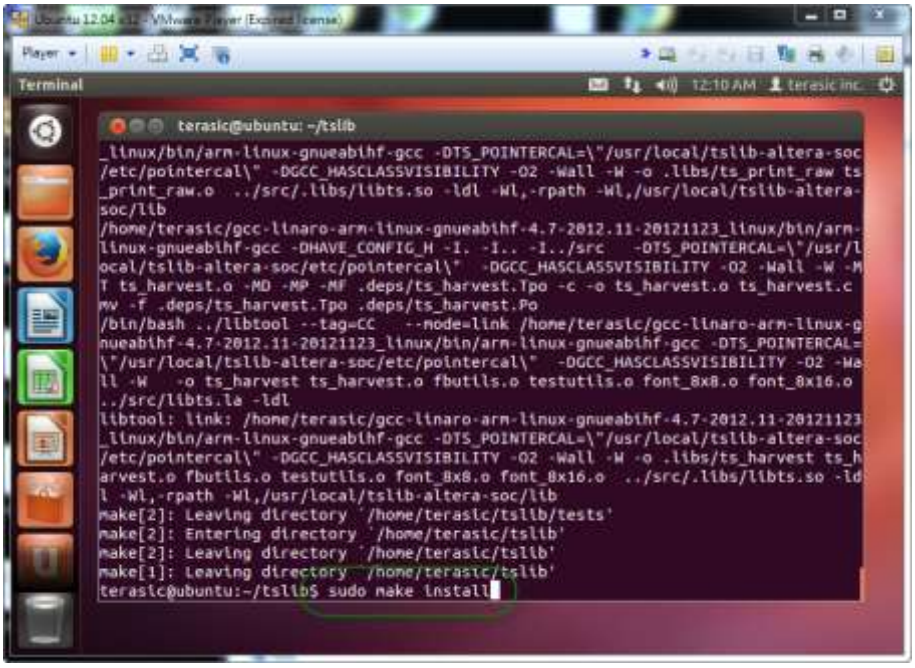

Figure 4-6 Input User Password to Continue the Installation

Figure 4-7 shows the screenshot when the installation is completed.

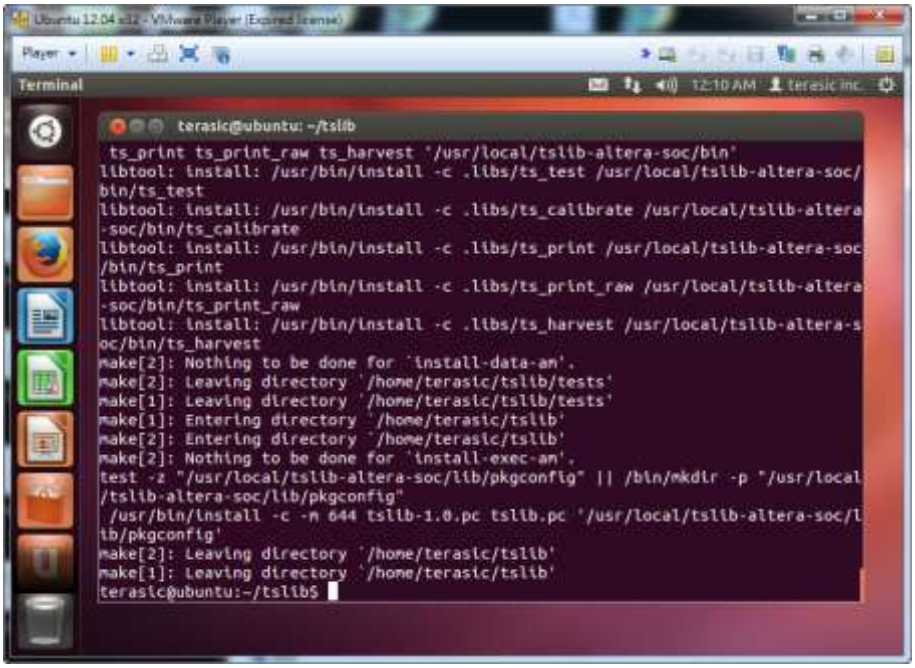

Figure 4-7 Installation is now complete

The **tslib** library will be installed in the folder "/usr/local/tslib-altera-soc" as shown in Figure 4-8.

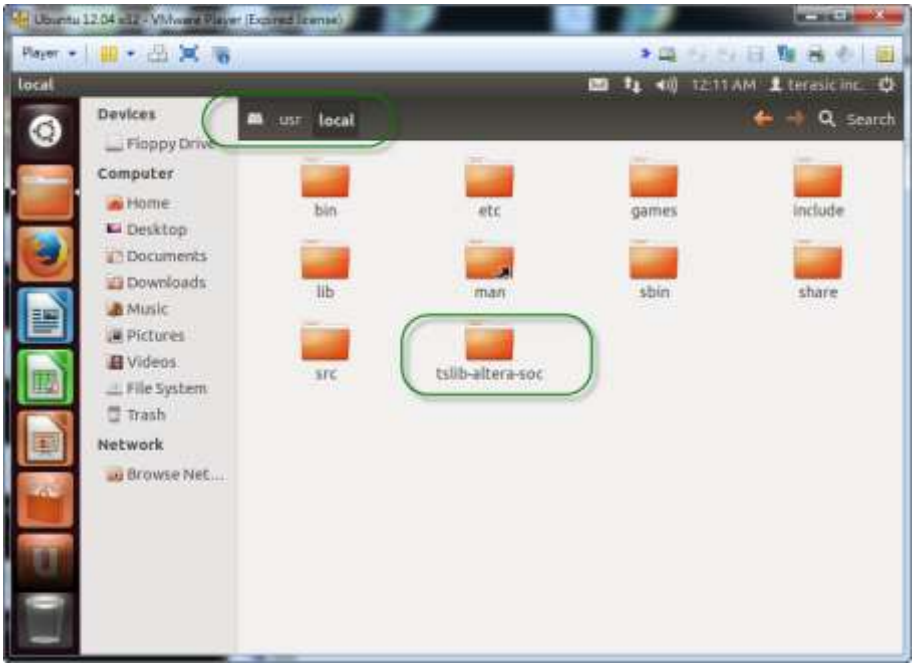

Figure 4-8 Installed Location of Altera SoC tslib Library

4.3 Install tslib Library on Altera SoC Linux

This section describes how to install the generated **tslib** library on the Altera SoC Linux. The installation is very straightforward. Just copy the installed folder on Host Linux to Altera SoC Linux and specify the input device in ts.conf file. To speed up the copying process, we will compress the folder first, copy the compressed file to Altera SoC Linux, and decompress it. In this tutorial, **vi** editor is used to modify ts.conf. Developers also can use other text editors to modify ts.conf file.

Here are the installation procedures:

1. In Host Linux, enter Linux terminal and type "cd /usr/local" to go to the "/usr/local" folder.
2. Type "sudo tar -jcv -f tslib-altera-soc.tar.gz2 tslib-altera-soc" to compress the folder as **tslib-altera-soc.tar.gz2**.
3. From Host Linux, copy the **tslib-altera-soc.tar.gz2** to the folder "/usr/local" on Altera SoC Linux. For detailed copying operation, please refer to the section 8.1 Copy files to Altera SoC Linux.
4. In Altera SoC Linux, enter Linux terminal and type "cd /usr/local" to go to the "/usr/local" folder. Because the Linux has built-in touch-screen library in the folder

"tslib-altera-soc", type "mv tslib-altera-soc tslib-altera-soc-org" to rename the built-in folder.

5. Type " tar -jxv -f tslib-altera-soc.tar.gz2" to decompress the library into the folder "/usr/local/tslib-altera-soc".
6. Type "cd /usr/local/tslib-altera-soc/etc" to go to the folder " /usr/local/tslib-altera-soc/etc"
7. Type "chmod +w ts.conf" to add write-attribute to the ts.conf file.
8. Type "vi ts.conf" to edit the file.
9. In **vi** editor, uncomment the "module_raw input" as shown in **Figure 4-9**. Note, the space before "moudle_raw input" also should be removed. Then, type ":w" to save the file and type ":q" to exit **vi**.
10. Type "vi –w ts.conf" to remove write-attribute from the ts.conf file.

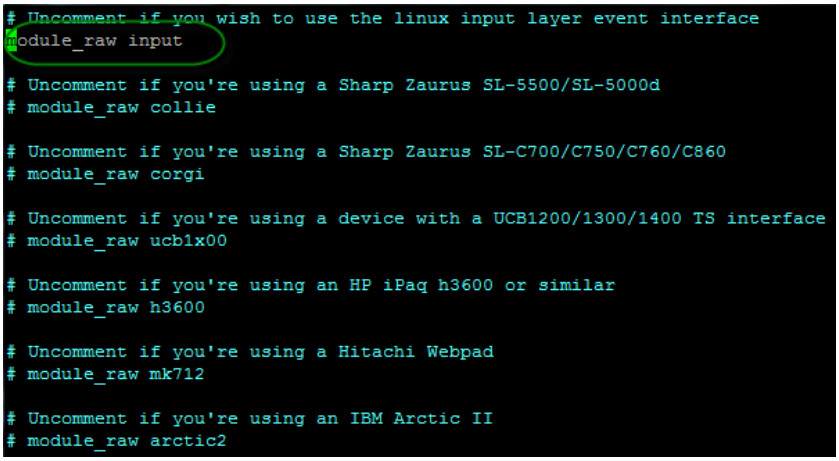

Figure 4-9 Content of ts.conf

Now, you have finished the installation of **tslib** on Altera SoC Linux. You can test the **ts_test** application in the library as described in section **1.5 Execute touch-screen Demo** to make sure the installation is correct.

Chapter 5

Build QT Library

Up to now, you should have successfully installed the Altera SoC tool-chain in the Host Linux. Now we start building QT project for Altera SoC ARM on the Host Linux, and showing how to install the library on Altera SoC Linux. Before building QT project, please make sure you have successfully built the **tslib** touch-screen library. Developers can skip this chapter if they are not interested in it now.

Here is a quick look of the 4 simple steps:

- Download and extract the QT source code
- Create a make configuration folder 'mkspecs' for Altera SoC
- Configure, build and install QT Library
- Install QT library on Altera SoC Linux

5.1 Download the QT source code

■ Download Compressed Source File

Go to the QT libraries download home page http://qt-project.org/downloads as shown in **Figure 5-1**. Then, click "Qt libraries 4.8.5 for embedded Linux" to download the source code.

Figure 5-1 QT Source Code Download Web Page

When an **Opening** dialog appears as shown in **Figure 5-2**, please select "Save File" radio button and click "OK". The file is saved as "~/Downloads/qt-everywhere-opensource-src-4.8.5.tar.gz".

Figure 5-2 Save QT File

■ Extract the Compressed Source File

The downloaded source file is a compressed file. Again we need to extract the file before we can use it. Please launch a terminal (CTRL+ALT+T) and use the following commands to extract the file:

```
$cd ~/Downloads
$ls
$tar -zxvf qt-everywhere-opensource-src-4.8.5.tar.gz
```

Note the 2nd command line 'ls" checks if the source file exists and the 3rd command line is used to extract the source file. The source will be extracted to the folder:

"~\Download\qt-everywhere-opensource-src-4.8.5".

5.2 Create a new 'mkspecs' for Altera SoC

In order to perform cross-compile for the QT library, a new make configuration file should be defined first. What we will do is to first copy existing make configuration file and then to modify the file accordingly.

■ Copy existing make configuration files

Copy the entire make configuration folder "linux-arm-gnueabi-g++" and rename it (simply add "hf" at the end of "gnueabi") as "linux-arm-gnueabihf-g++" with the following command:

```
$cd ~/Downloads/qt-everywhere-opensource-src-4.8.5/mkspecs/qws
$cp -r linux-arm-gnueabi-g++ linux-arm-gnueabihf-g++
```

Where the 2nd command line is to browse the folder where the folder "linux-arm-gnueabi-g++" exists and the 3rd command line is to duplicate the "linux-arm-gnueabi-g++" folder and rename it as "linux-arm-gnueabi**hf**-g++" s shown in **Figure 5-3**

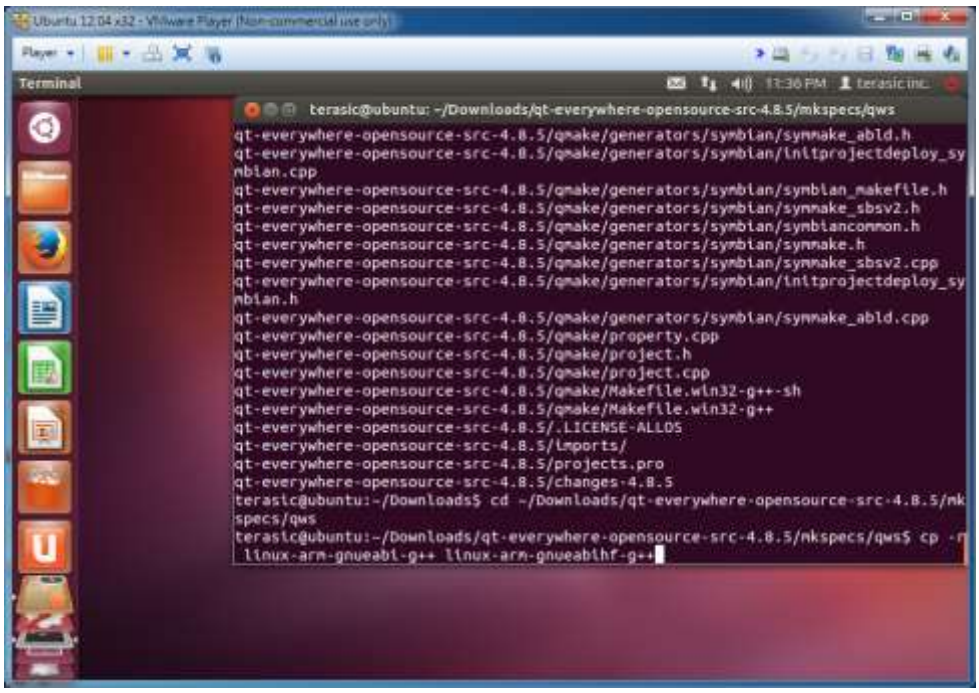

Figure 5-3 Duplicate Configuration Folder

■ Modify the Configuration File

Simply follow the below procedures to modify the configuration file in the terminal:

1. Type “cd ~/Downloads/qt-everywhere-opensource-src-4.8.5/mkspecs/qws”
2. Type “cd linux-arm-gnueabihf-g++”
3. Type “gedit qmake.conf”. The file is opened by gedit as shown in **Figure 5-4**
4. In gedit, add the following three compile option definition statements.

```
QMAKE_INCDIR += /usr/local/tslib-altera-soc/include
QMAKE_LIBDIR += /usr/local/tslib-altera-soc/lib
QMAKE_LFLAGS += -Wl,-rpath-link=/usr/local/tslib-altera-soc/lib
```

5. In gedit, replace all “arm-none-linux-gnueabi” with “arm-linux-gnueabihf” as shown in **Figure 5-5**. Click “Save” icon followed by clicking “Close” icon to terminate gedit.

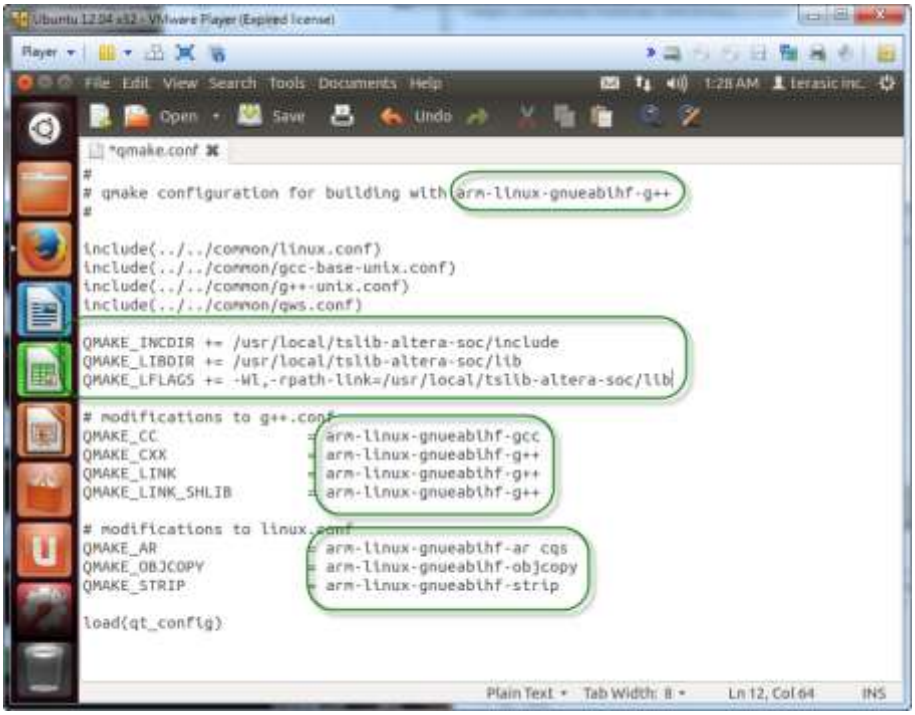

Figure 5-4 qmake.conf File

5.3 Configure, build, and install QT library

After we have modified the make configuration files, now we are ready to configure, build, and install the QT library. Here is a list of the commands that will be used toward building and installing the QT library.

```
$source ~/.profile
$cd ~/Downloads/qt-everywhere-opensource-src-4.8.5
$ ./configure -prefix /usr/local/qt-4.8.5-tslib-altera-soc -release -shared \
-nomake examples -nomake tools -nomake docs -make demos \
-xplatform qws/linux-arm-gnueabihf-g++ -embedded arm -little-endian \
-qt-mouse-tslib \
-no-pch -v
$make
$sudo make install
```

■ Configure

The command line

```
$source ~/.profile
```

is used to make sure the $PATH$ includes the correct Altera SoC tool-chain path.

To go to the source code folder, type

```
$cd ~/Downloads/qt-everywhere-opensource-src-4.8.5
```

To execute the configure batch to create makefile, type

```
$ ./configure -prefix /usr/local/qt-4.8.5-tslib-altera-soc -release -shared \
-nomake examples -nomake tools -nomake docs -make demos \
-xplatform qws/linux-arm-gnueabihf-g++ -embedded arm -little-endian \
-qt-mouse-tslib \
-no-pch -v
```

Before executing the configure batch, system will prompt to ask the license type as shown in **Figure 5-5**, type in "o" and press ENTER. The parameter "-prefix /usr/local/qt-4.8.5-altera-soc" specifies the directory where to install the generated QT library. The parameter "-qt-mouse-tslib" specifies enabling touch-screen supporting via the generic **tslib** touch screen library. The parameter "-make demos" specifies to build demo code while build the QT project.

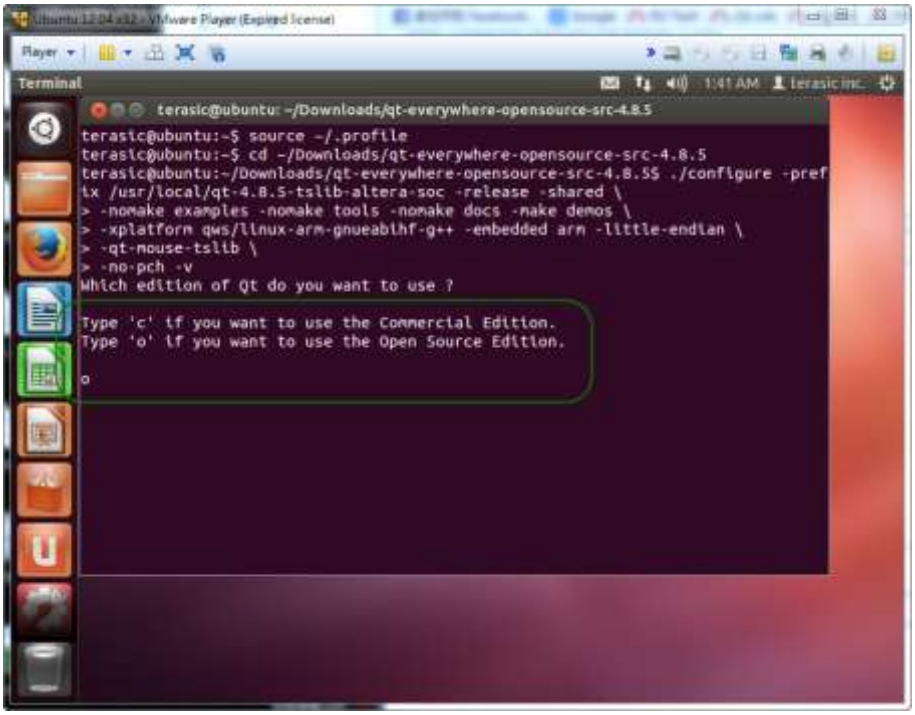

Figure 5-5 Ask License Type

After selecting the license type, system prompts to ask whether to accept the license term or not as shown in **Figure 5-6**. Type "yes" and press ENTER.

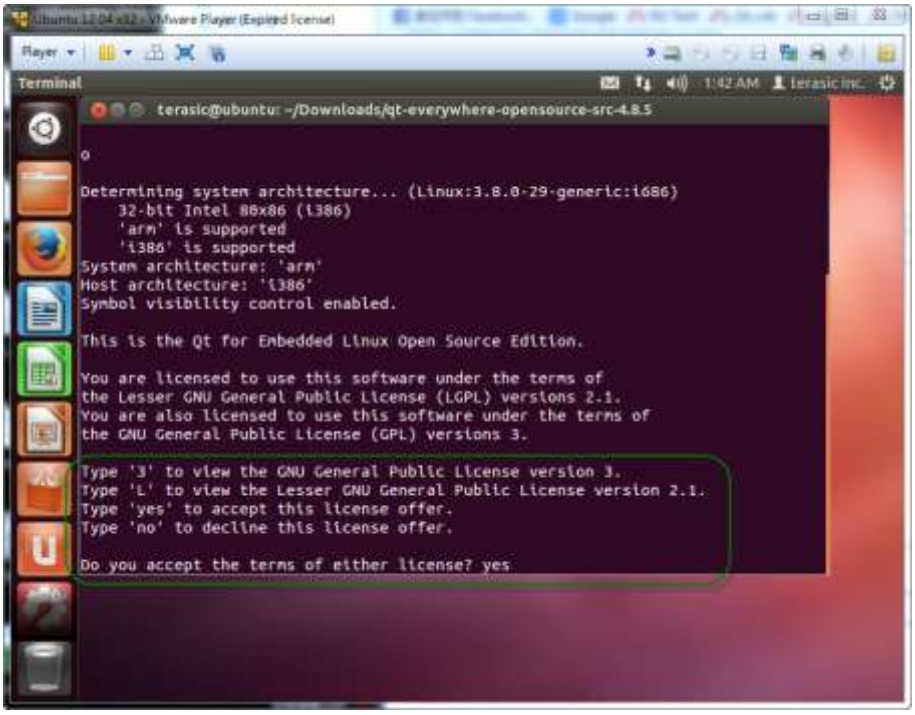

Figure 5-6 Type Yes to Accept This License Offer

When configuration is completed successfully, your will see the message “Qt is now configured for building. Just run ‘make’ as shown in **Figure 5-7**.

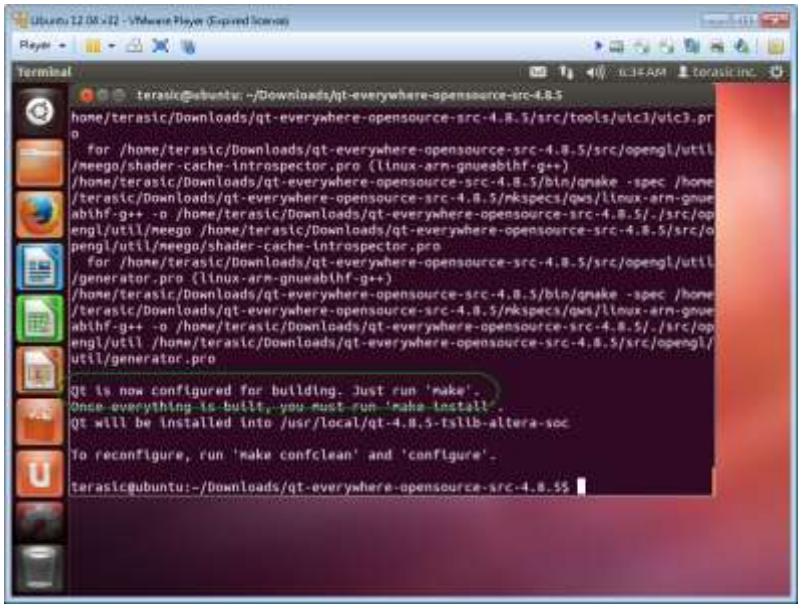

Figure 5-7 Qt Now Configured for Building

■ Make

In the terminal simply type:

```
$make
```

to build the QT library as shown in **Figure 5-8**. The build process will take several hours to complete.

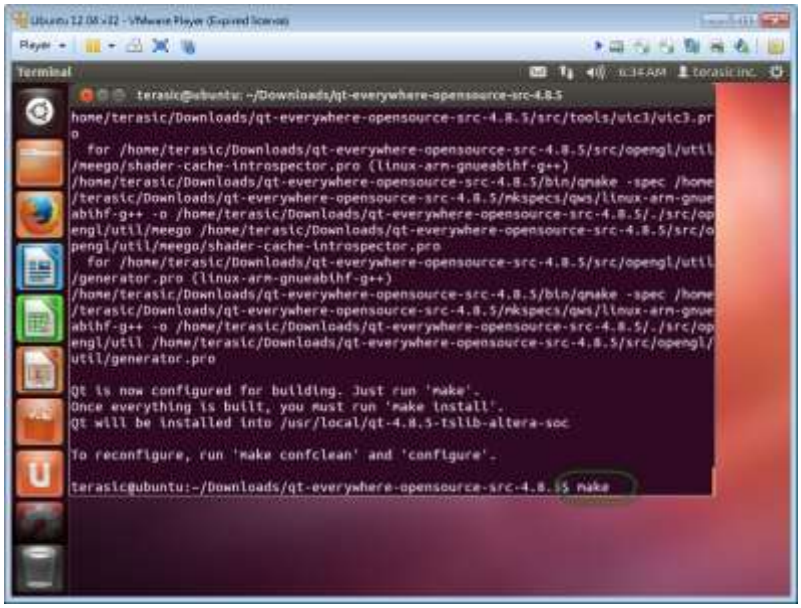

Figure 5-8 Type Make to Start the Build Process

Figure 5-9 shows the screenshot when the make process is completed.

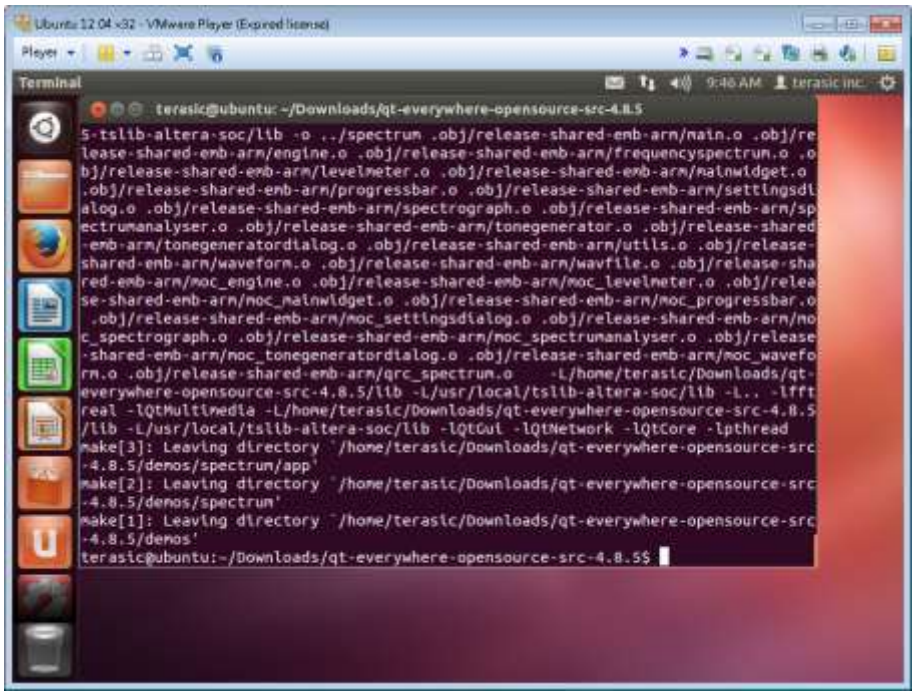

Figure 5-9 Make Process Now Complete

■ Install

Finally, to install the QT package please type in

```
$sudo make install
```

and type in “123” when asked for password for terasic as shown in **Figure 5-10**.

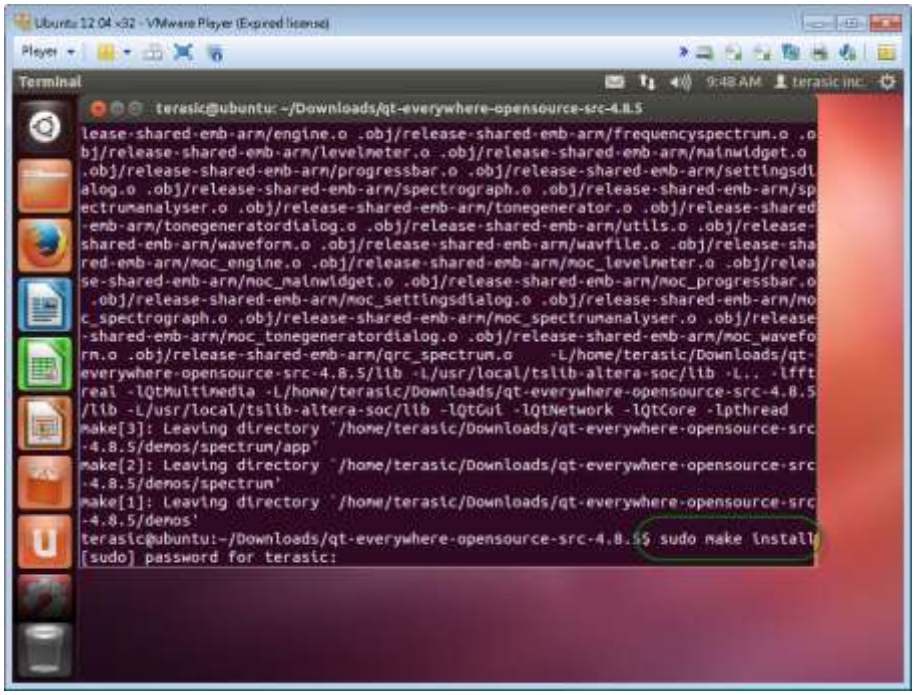

Figure 5-10 Input User Password to Continue the Installation

Figure 5-11 shows the screenshot when the installation is completed.

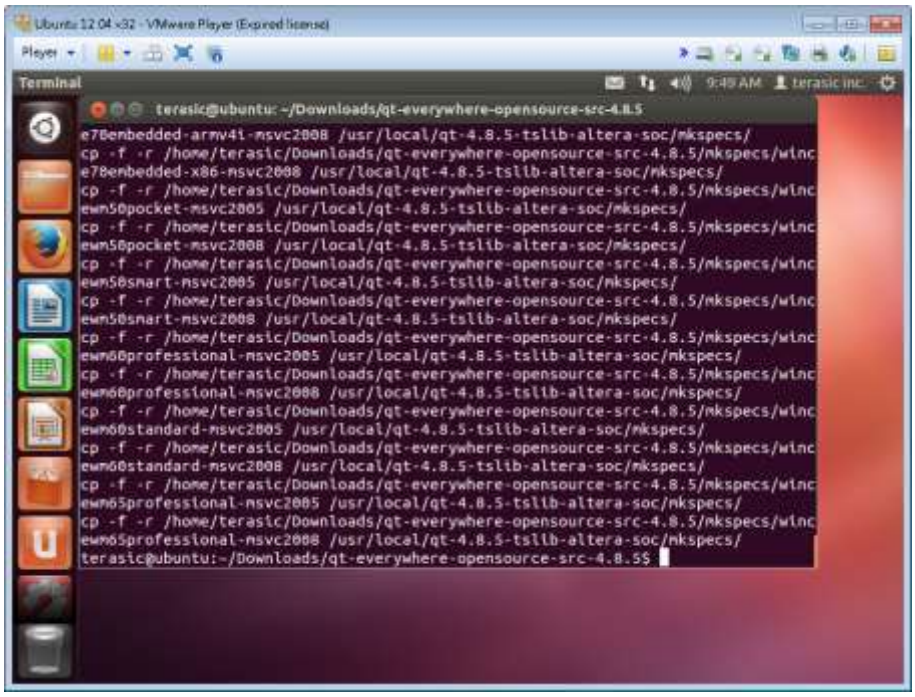

Figure 5-11 Installation is now complete

The QT package will be installed in the folder "/usr/local/qt-4.8.5-tslib-altera-soc" as shown in Figure 5-12.

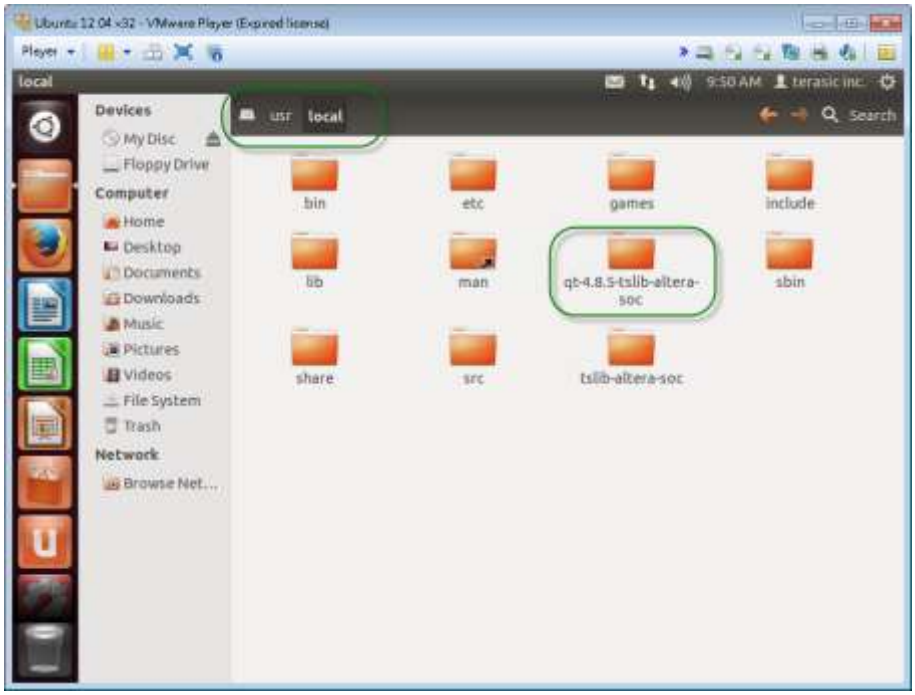

Figure 5-12 Installed Location of Altera SoC QT Package

5.4 Installation QT Library on Altera SoC Linux

This section describes how to install the generated **QT** library on the Altera SoC Linux. The installation is very straightforward. Just copy the installed folder on Host Linux to Altera SoC Linux and add some system environment variables in the file /etc/profile so the system knows where to find the QT library. To speed up the copying process, we will compress the folder first, copy the compressed file to Altera SoC Linux, and decompress it. In this tutorial, **vi** editor is used to modify the file "/etc/profile". Developers also can use other text editors to modify the file "/etc/profile".

Here is the installation procedures:

1. In Host Linux, enter Linux terminal and type "cd /usr/local" to go to the "/usr/local" folder.
2. Type "sudo tar -jcv -f qt-4.8.5-tslib-altera-soc.tar.gz2 qt-4.8.5-tslib-altera-soc" to compress the folder as **qt-4.8.5-tslib-altera-soc.tar.gz2**.

3. Copy the **qt-4.8.5-tslib-altera-soc.tar.gz2** to the folder "/usr/local" on Altera SoC Linux. For detailed copying operation, please refer to the section 8.1 Copy files to Altera SoC Linux.
4. In Altera SoC Linux, enter Linux terminal and type "cd /usr/local" to go to the "/usr/local" folder. Because the Linux had built-in the QT library in the folder "qt-4.8-5-tslib-altera-soc", type

   "mv qt-4.8-5-tslib-altera-soc qt-4.8-5-tslib-altera-soc-org"

   to rename the built-in folder.
5. Type " tar -jxv -f qt-4.8.5-tslib-altera-soc.tar.gz2" to decompress the library into the folder "/usr/local/qt-4.8-5-tslib-altera-soc".
6. Type "vi /etc/profile" to use **vi** editor to open the file "profile".
7. In **vi** editor, add the following content into the file "profile", type ":w" to save the file and type ":q" to exit **vi** editor. Actually, these statement had been built-in in /etc/profile, so developers just need to make sure they are there and type ":q" to exit **vi** editor.

```
export TSLIB_DIR=/usr/local/tslib-altera-soc
export QT_DIR=/usr/local/qt-4.8.5-tslib-altera-soc
export LD_LIBRARY_PATH=$PATH:$QT_DIR/lib:$TSLIB_DIR/lib
export QWS_MOUSE_PROTO=tslib:/dev/input/event0
```

Now, you have finished the installation of **QT** Library on Altera SoC Linux. You can test the **affine** application in the library as described in section **1.6 Execute QT Demo** to make sure the installation is correct.

Chapter 6

QT Creator Installation

QT Creator is a convenient tool to help create GUI-based programs. Users can simply follow our GUI application development explained in this tutorial to develop needed GUI applications. This chapter illustrates how to install QT Creator on the Host Linux. Also, we will show how to create, compile, and build a hello program running on Host Linux.

6.1 Download and Install QT Installer

■ Download QT Installer

As shown in **Figure 6-1**, click the Firefox web browser to open the web page http://download.qt-project.org/official_releases/qt/5.2/5.2.0/, then click "qt-linux-opensource-5.2.0-x86offline.run" to download the QT installer.

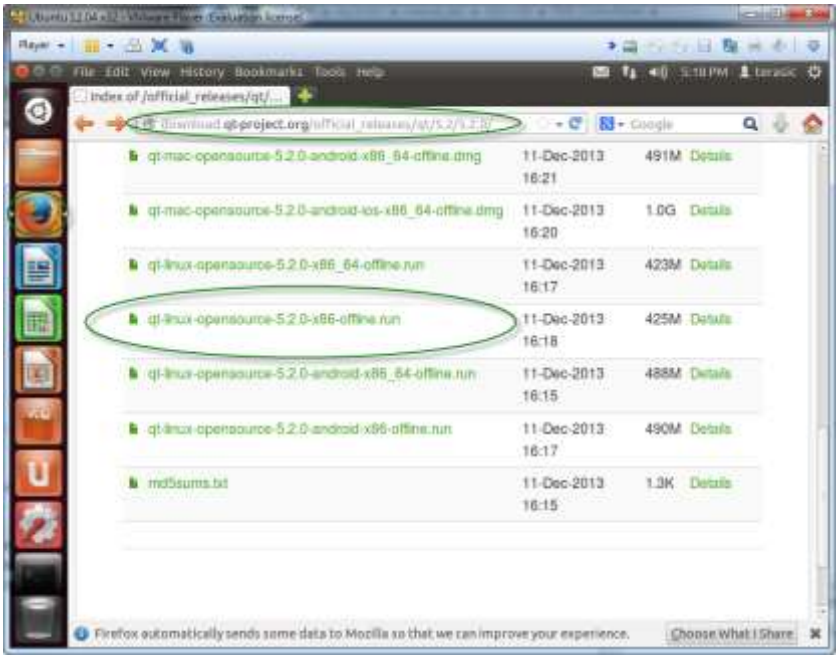

Figure 6-1 Web Page to Download QT

When an Opening dialog appears as shown in **Figure 6-2**, select “Save File” radio button, and click “OK”.

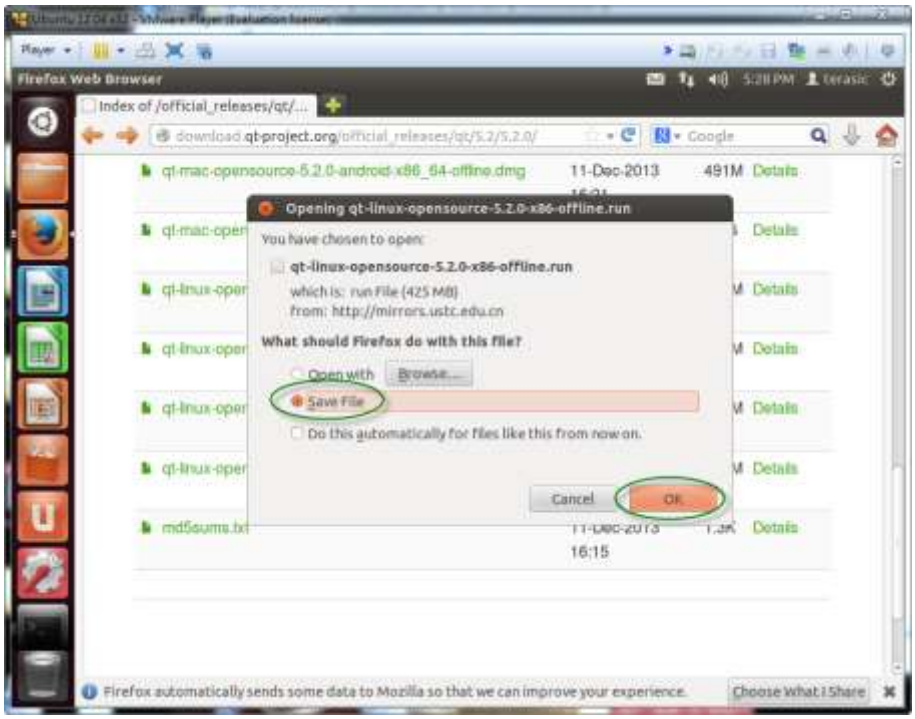

Figure 6-2 Opening Dialog

Figure 6-3 shows the download progress.

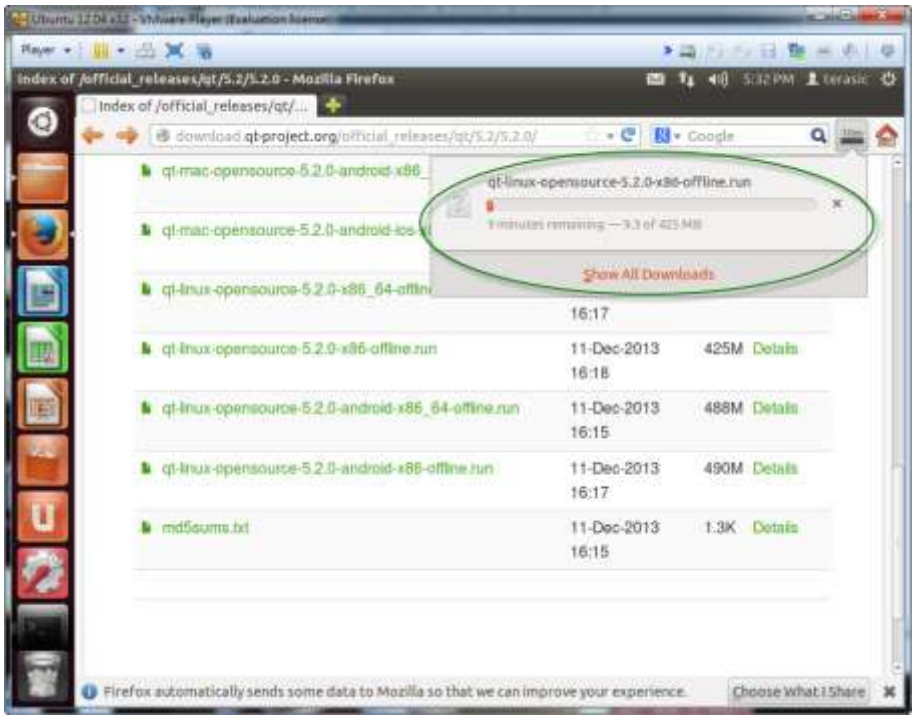

Figure 6-3 Download Process of qt-linux-opensource-5.2.0-x86-offline.run

The file is saved as "**qt-linux-opensource-5.2.0-x86-offline.run**", and saved under the folder "~/Download". When download is completed, click the Close icon, located on the left-top corner as shown in **Figure 6-4**, to close the Firefox web browser.

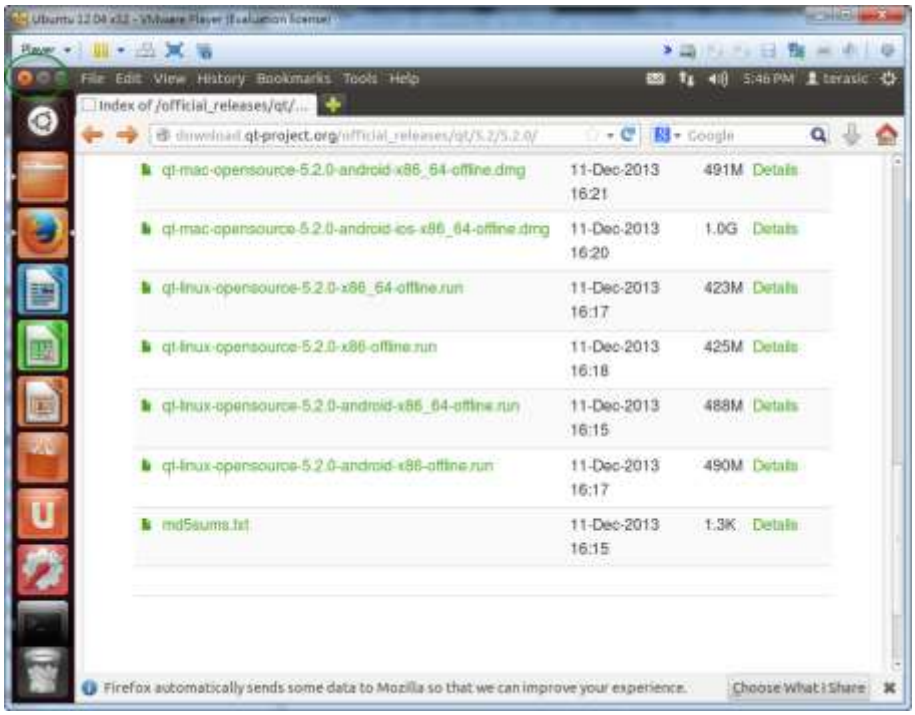

Figure 6-4 Click 'Close' Icon to Close Firefox Web Browser

■ Install QT

Type in the following commands to locate and launch the QT Installer, as shown in **Figure 6-5**.

```
$cd ~/Downloads/
$ls
$chmod +x qt-linux-opensource-5.2.0-x86-offline.run
$./qt-linux-opensource-5.2.0-x86-offline.run
```

The 2nd command line **ls** is used to check whether the qt-linux-opensource-5.2.0-x86-offline.run existed or not while the 3rd command line **chmod +x** is used to add "execution" attribute to the file such that it can be executed.

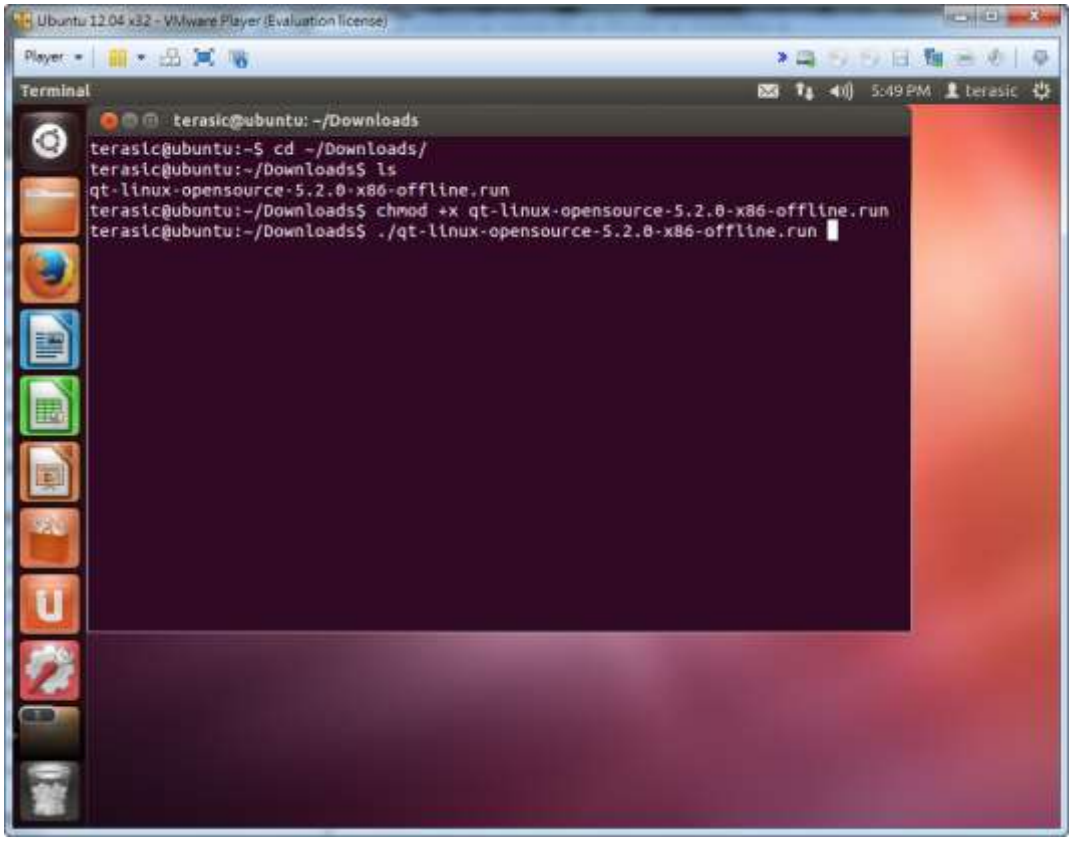

Figure 6-5 Locate and Launch the QT Installer

Figure 6-6 shows the **Welcome** dialog of the QT Installer.

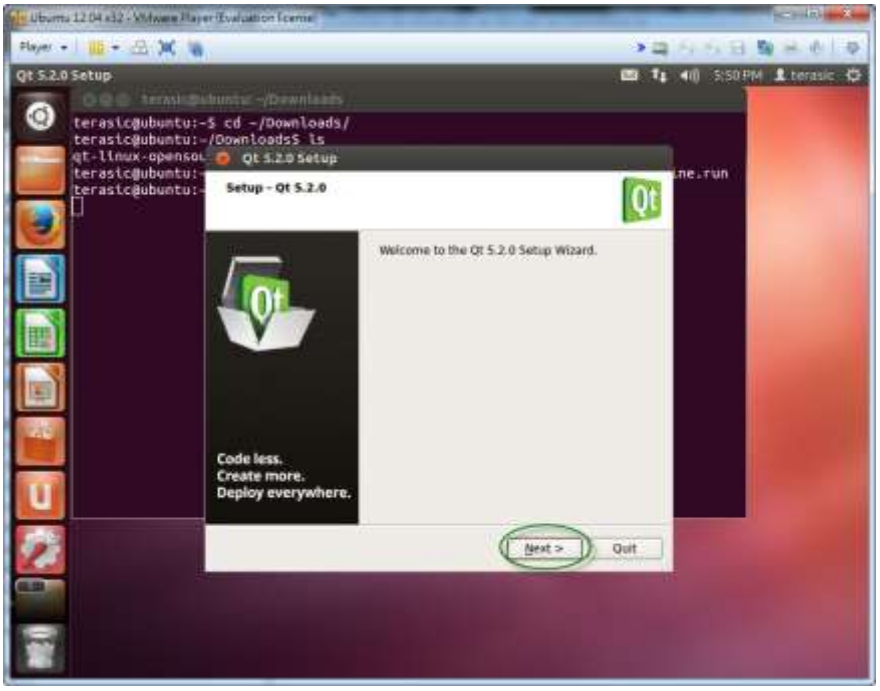

Figure 6-6 Welcome Dialog

In the **Installation Folder** dialog, please specify the folder where you wish to install Qt 5.2.0 and click "Next>" to go to the next step, as shown in **Figure 6-7**.

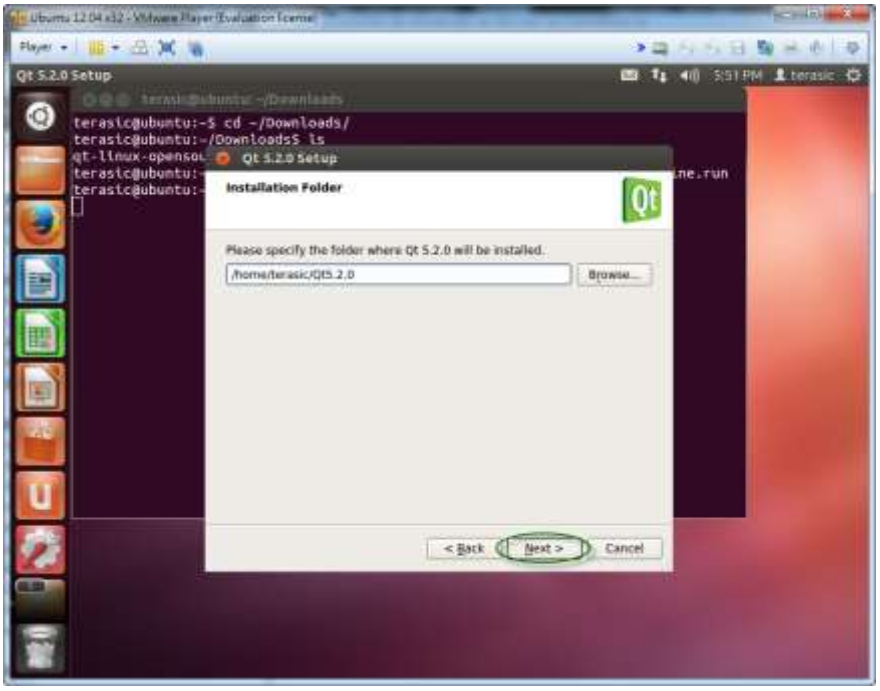

Figure 6-7 Specify the Folder to Install QT

In the **Select Components** dialog, keep default settings and click "Next >" to go to the next step, as shown in **Figure 6-8**.

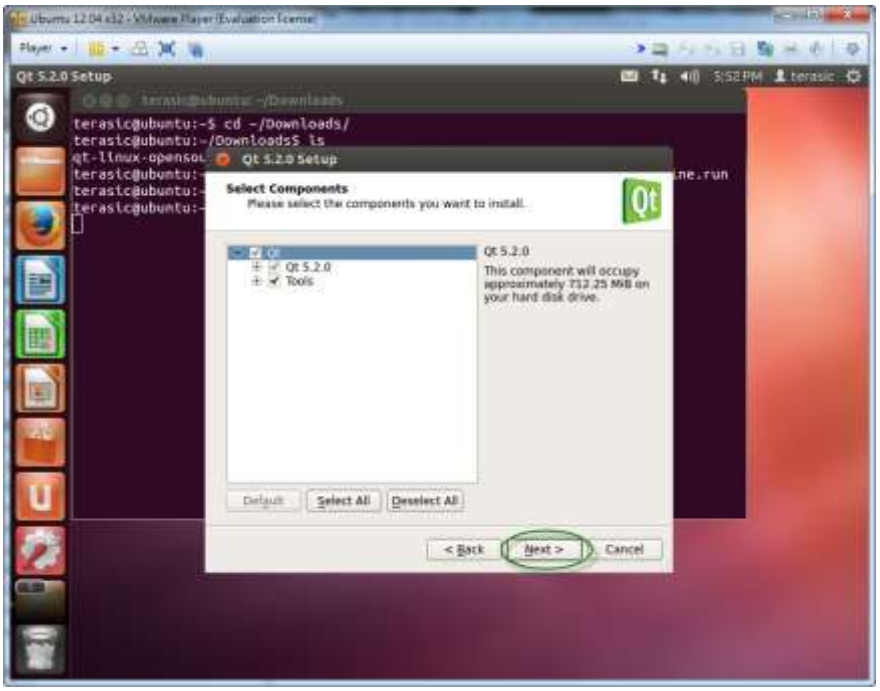

Figure 6-8 Select Components Dialog

In the **License Agreement** dialog, select a license and select the "I have read and agree.." radio button. Click "Next >" to go to the next step, as shown in **Figure 6-9**

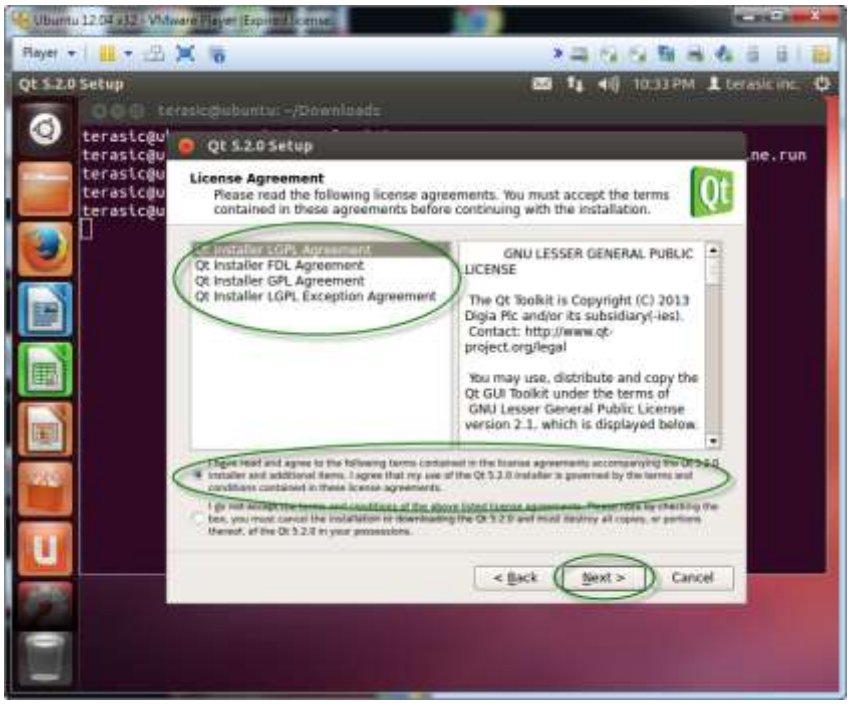

Figure 6-9 License agreement dialog of the QT installer

In the **Ready to Install** dialog, as shown in **Figure 6-10,** click "Install" to go to the next step.

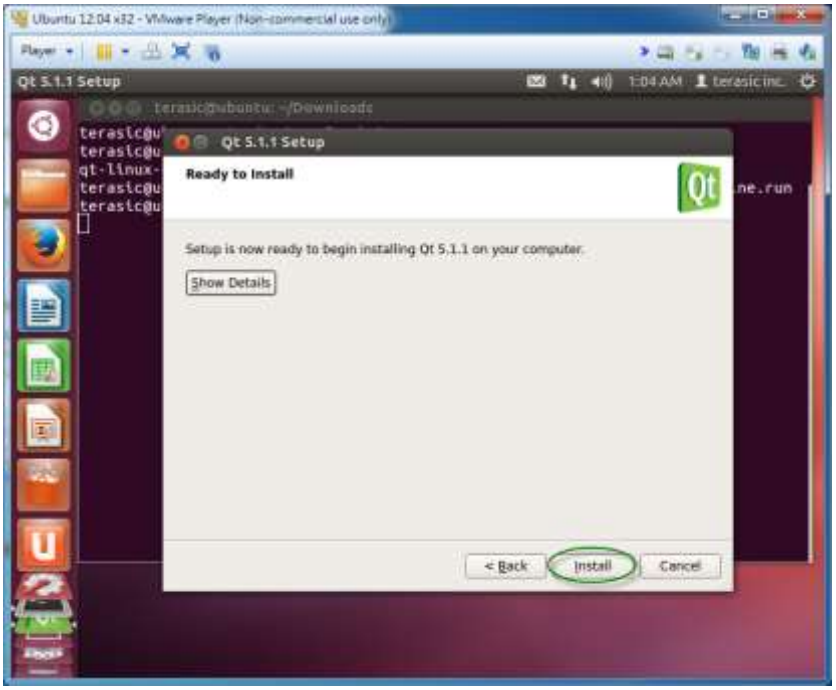

Figure 6-10 Ready to Install Dialog of QT Installer

In the **Completing the Qt 5.2.0 Wizard** dialog, as shown in **Figure 6-11,** click "Finish" to close the window.

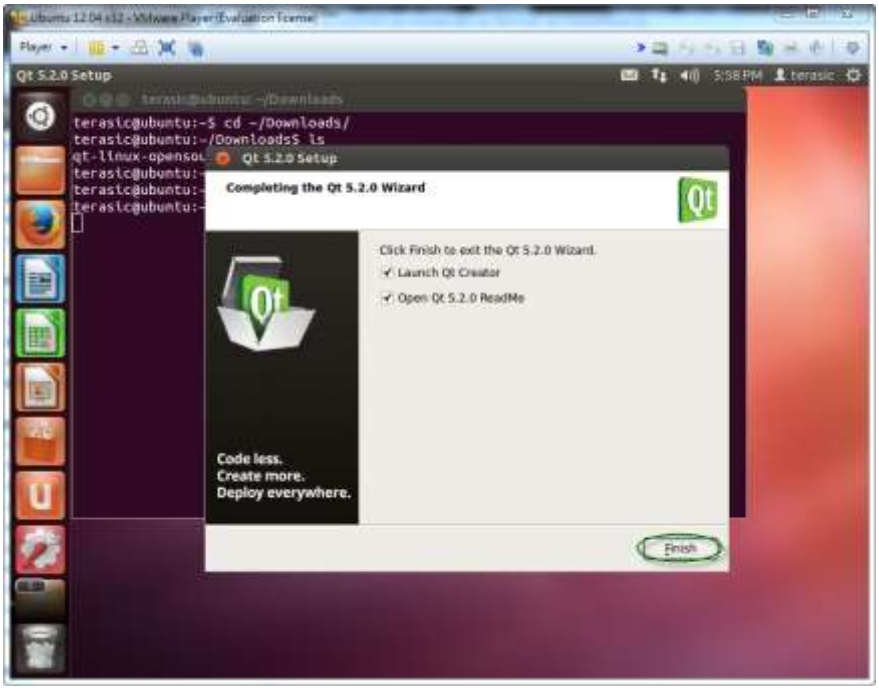

Figure 6-11 Completing Dialog of QT Installer

After installation has been completed, QT Creator is automatically launched as shown in **Figure 6-12.**

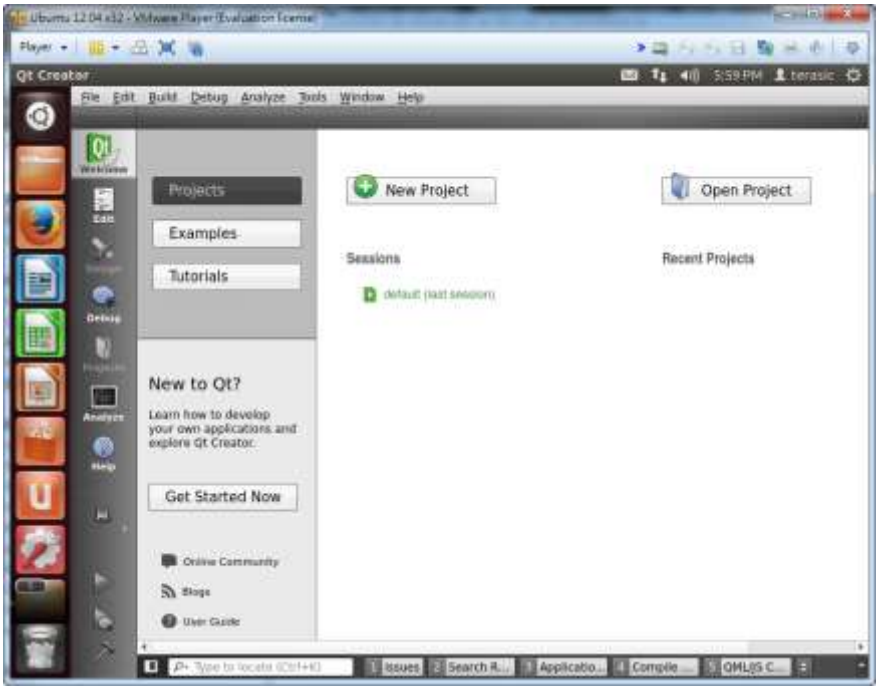

Figure 6-12 QT Creator GUI

Select the menu item “File→Exit”, as shown in Figure 6-13 to close the QT Creator.

Figure 6-13 Exit QT Creator

6.2 Launch QT Creator and Check Configure

This section describes how to launch QT Creator in Linux and where to check its Build & Run configuration settings.

■ Launch QT Creator

The Qt Creator program is located under “\home\Qt5.2.0\Tools\QtCreator\bin” as shown in Figure 6-14. To launch QT creator, simply double click “qtcreator” program icon.

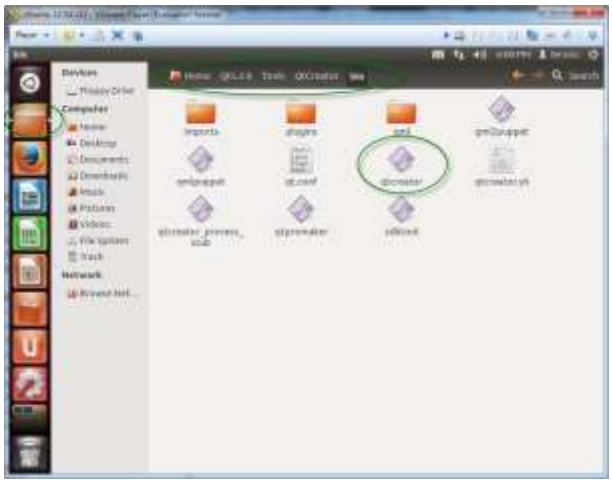

Figure 6-14 Launch QT Creator

■ Check Build & Run Configuration Settings

When QT Creator is launched, browse the menu and click “Tools→Options...”, as shown in **Figure 6-15,** to open the Option dialog.

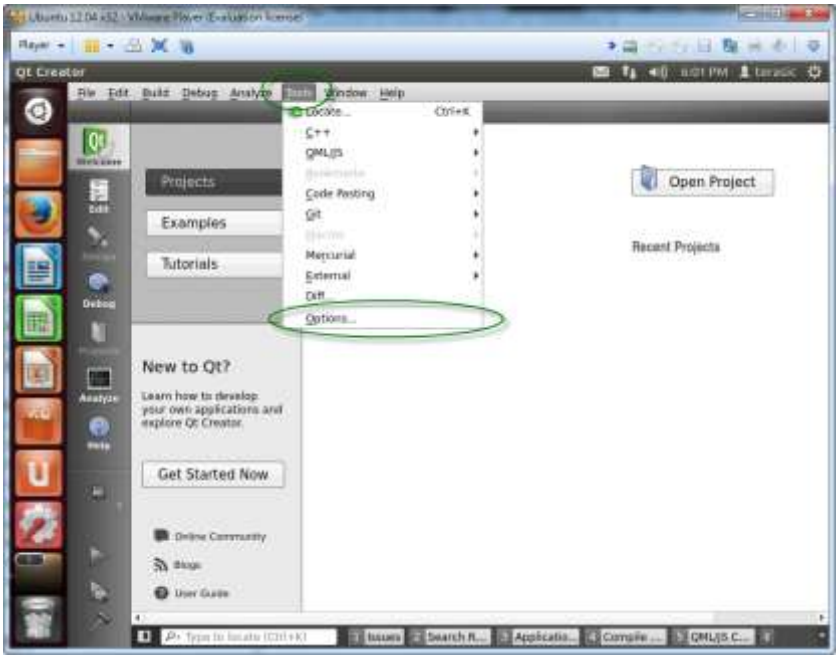

Figure 6-15 Open Option Dialog

In the **Options** dialog, first click “Build & Run” on the left and select the “Compilers” tab on the right to check if the “GCC” is detected as shown in **Figure 6-16.**

Figure 6-16 Compilers Options in QT Creator

Next, select "Qt Versions" tab (to the left of the "Compilers" tab) to check if the "Qt 5.2.0 GCC 32bit" is detected as shown in **Figure 6-17**.

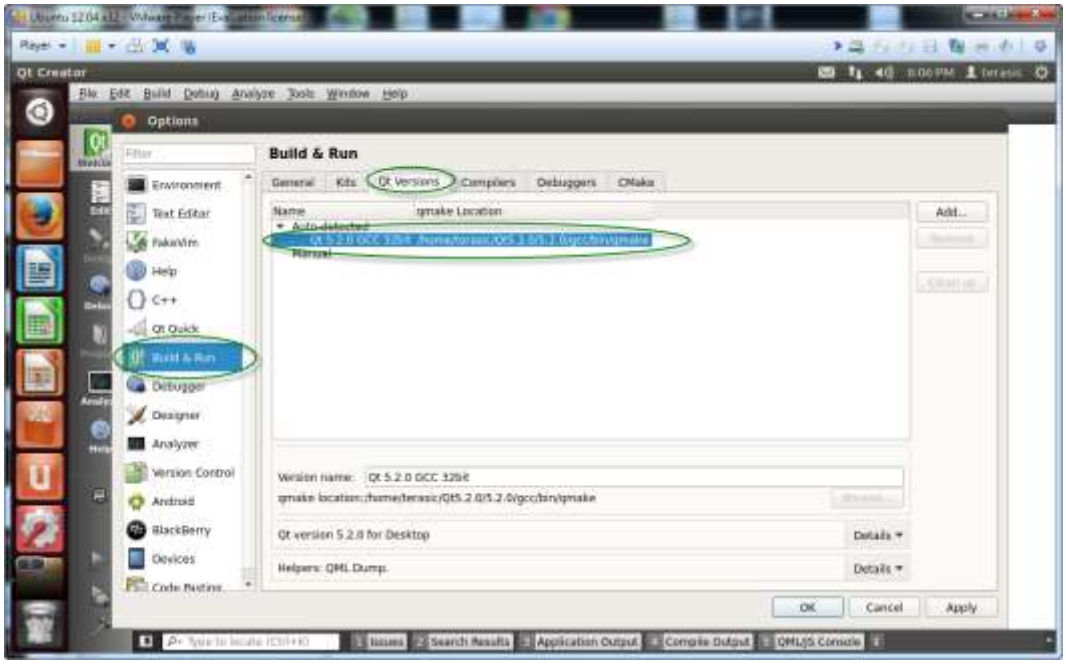

Figure 6-17 QT Version Option in QT Creator

Similarly, select "Kits" tab to check if the "Desktop Qt 5.2.0 GCC 32bit" is detected as shown in **Figure 6-18.**

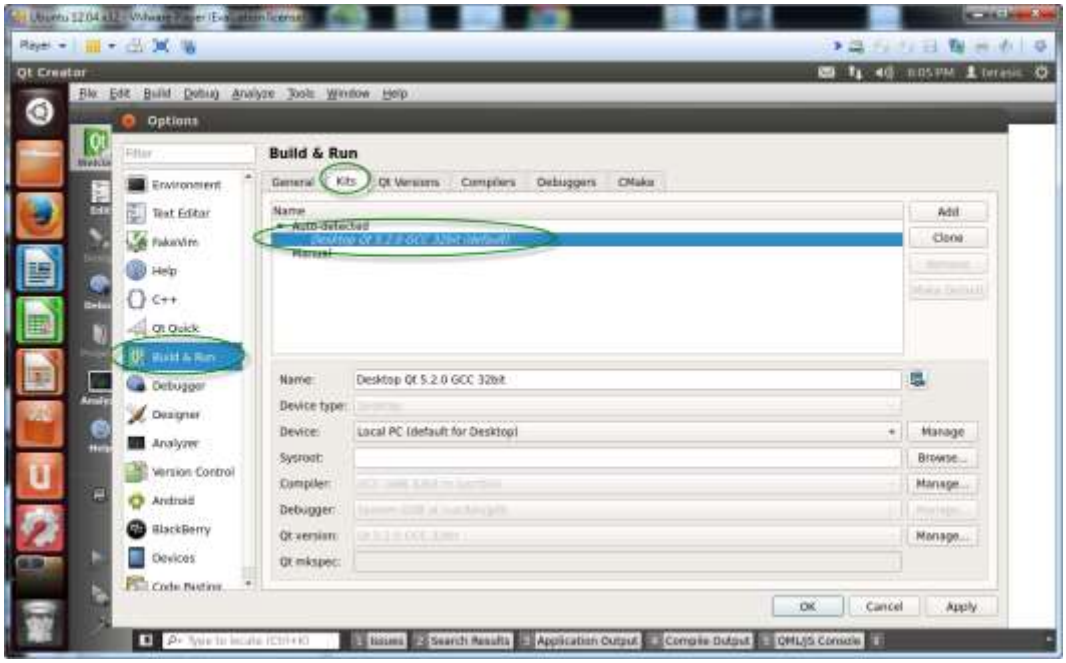

Figure 6-18 Kits Option in QT Creator

Finally, click "OK" to close the dialog as shown in **Figure 6-19**.

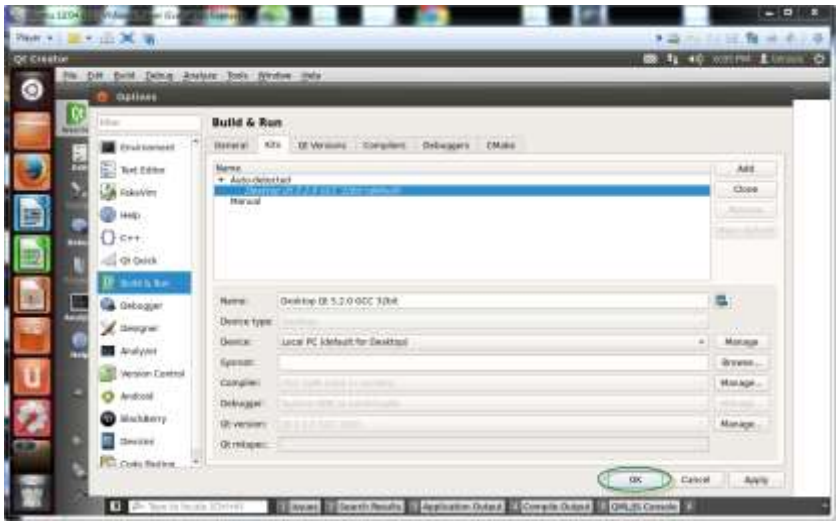

Figure 6-19 Click "OK" to Close Options Dialog

6.3 Hello Program

Now we are ready to create, build, and run our first program "Hello" in QT Creator on Host Linux. Please follow carefully with the following instructions.

■ Create a New Project

After launching the QT Creator, browse the menu and select item “File→New File or Project…” as shown in **Figure 6-20** to open a new dialog.

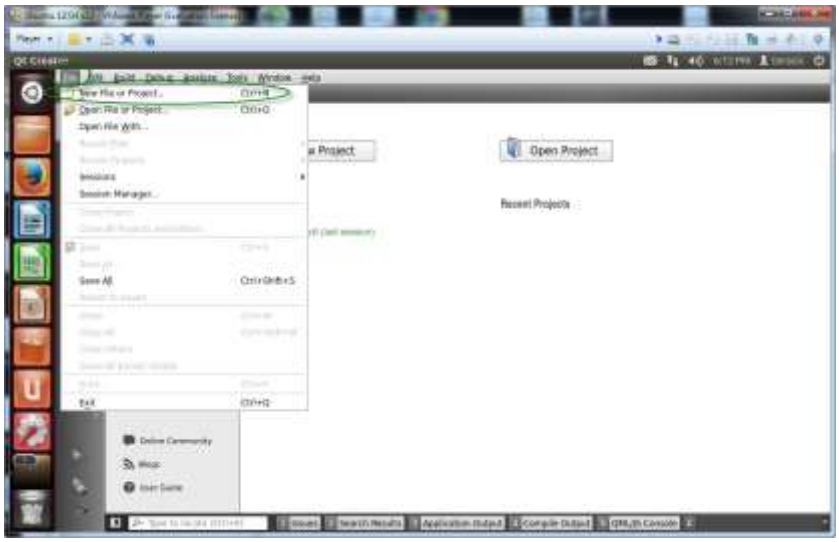

Figure 6-20 Open a New Project Dialog

In the **New** dialog, select “Applications” under Projects, and choose “Qt Widgets Application” as shown in **Figure 6-21**. Click “Choose…” to go to the next step.

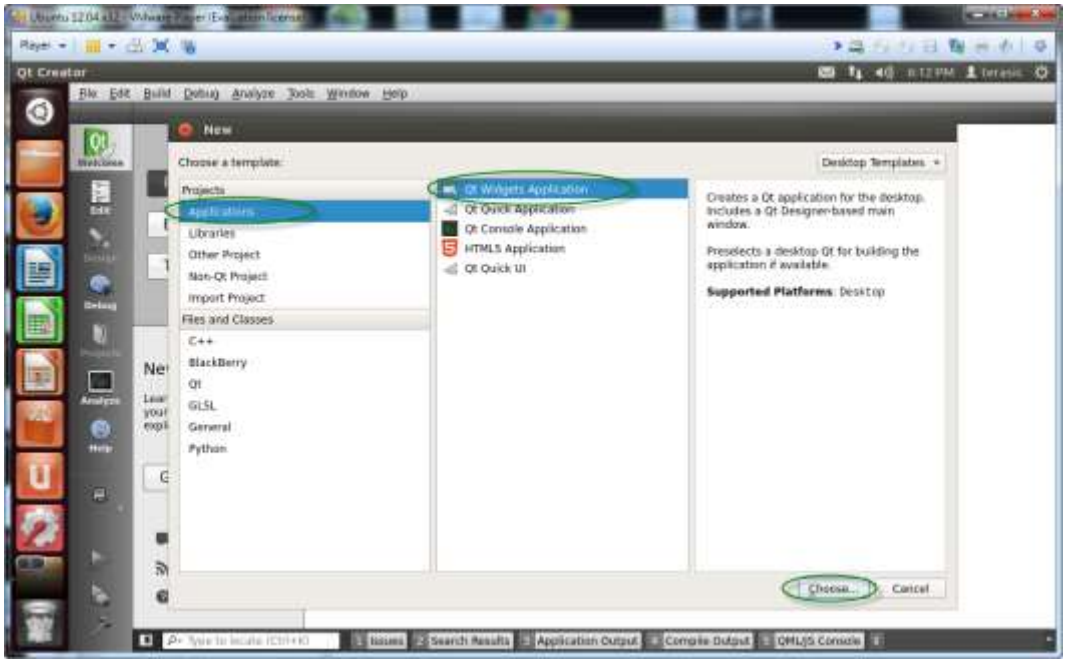

Figure 6-21 Dialog of Creating a New Project

In the **Qt Widgets Applicaton** Dialog, specify the **Name** as “hello”, click “Browse…” to create a folder “/home/**terasic**/my_hello” and set **Location in** to the created the folder, and then click “Next >” as shown in **Figure 6-22**. Note that in the path string, you should replace “terasic” with your linux user name.

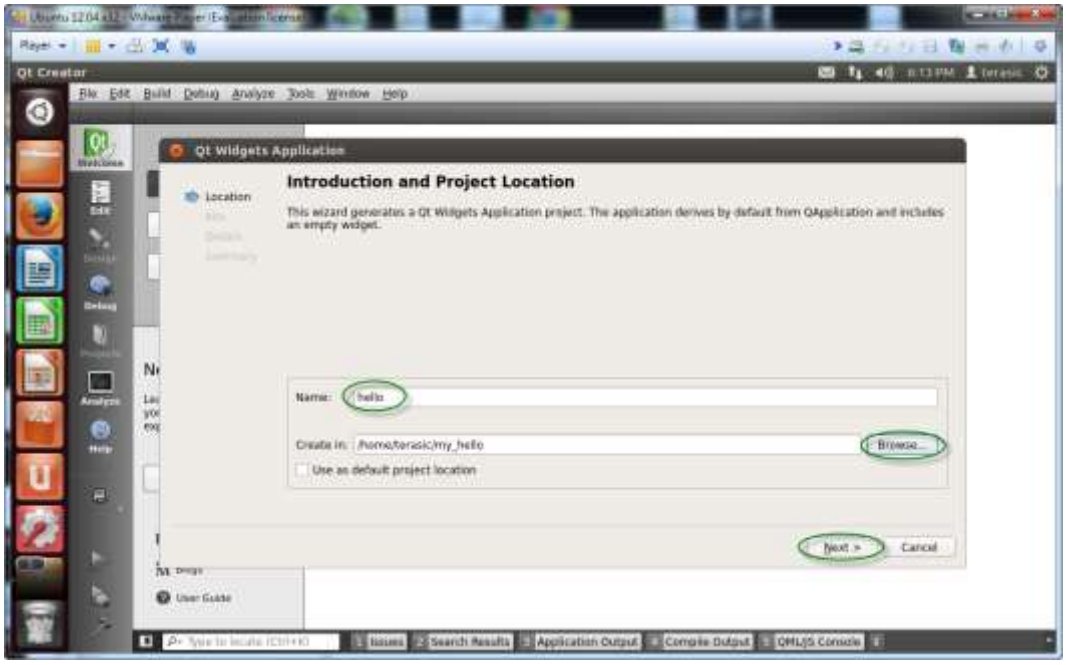

Figure 6-22 Project Name and Location Dialog

In the **Kit Selection** dialog, keep default settings and click “Next >” to go to next step as shown in **Figure 6-23**.

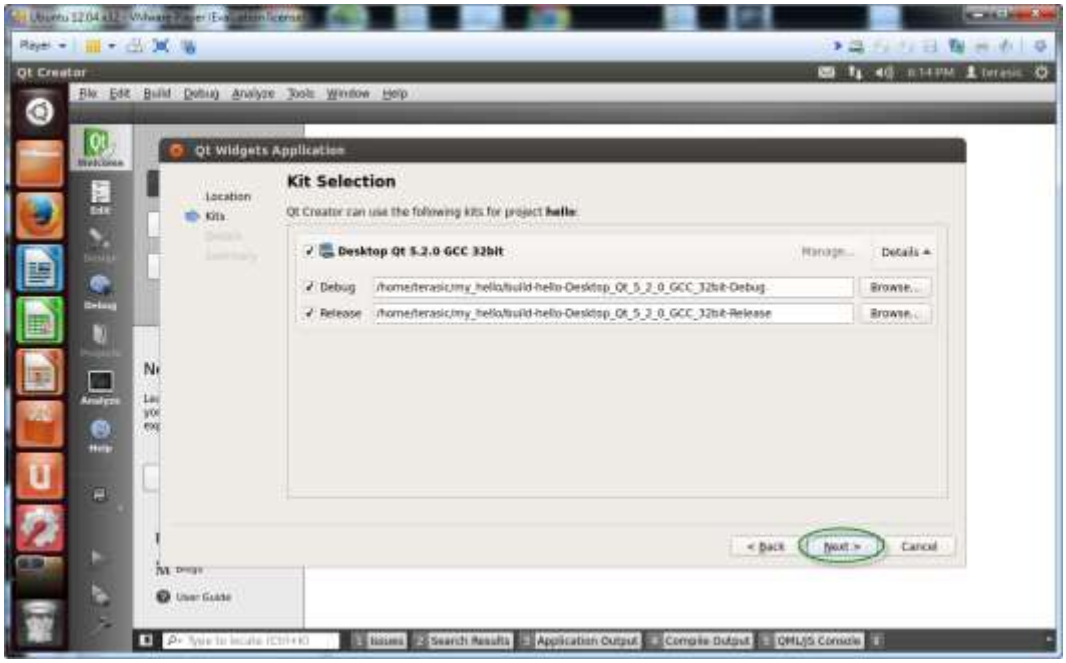

Figure 6-23 Kit Selection Dialog

In the **Class Information** dialog, click "Next >" as shown in **Figure 6-24** to proceed.

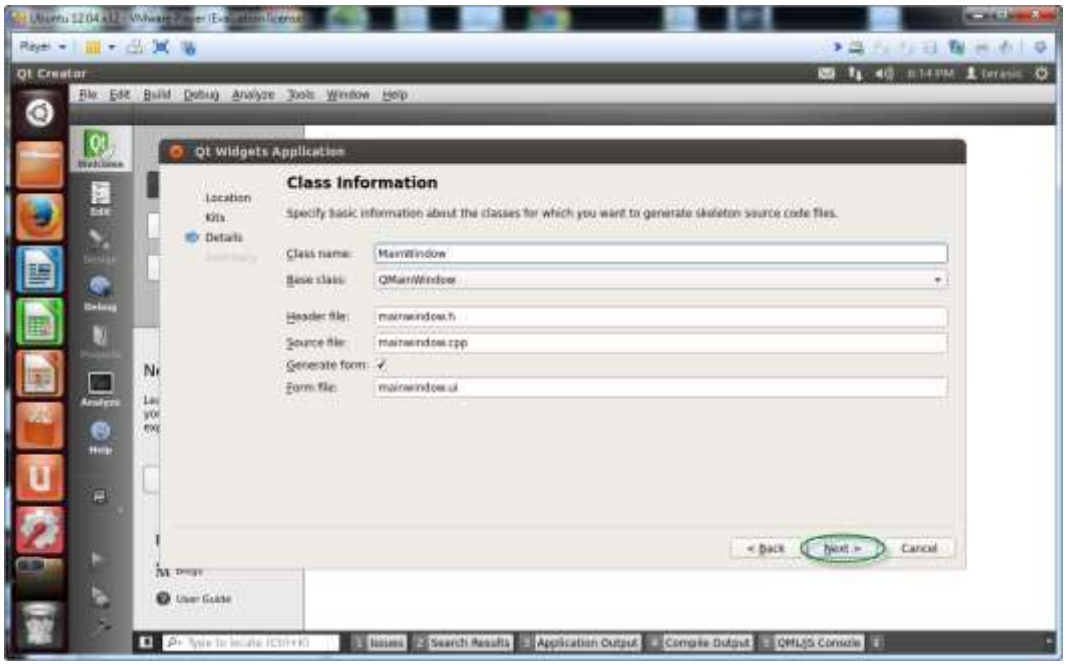

Figure 6-24 Class Information Dialog

In the **Project Management** dialog, click "Finish" as shown in **Figure 6-25** to proceed.

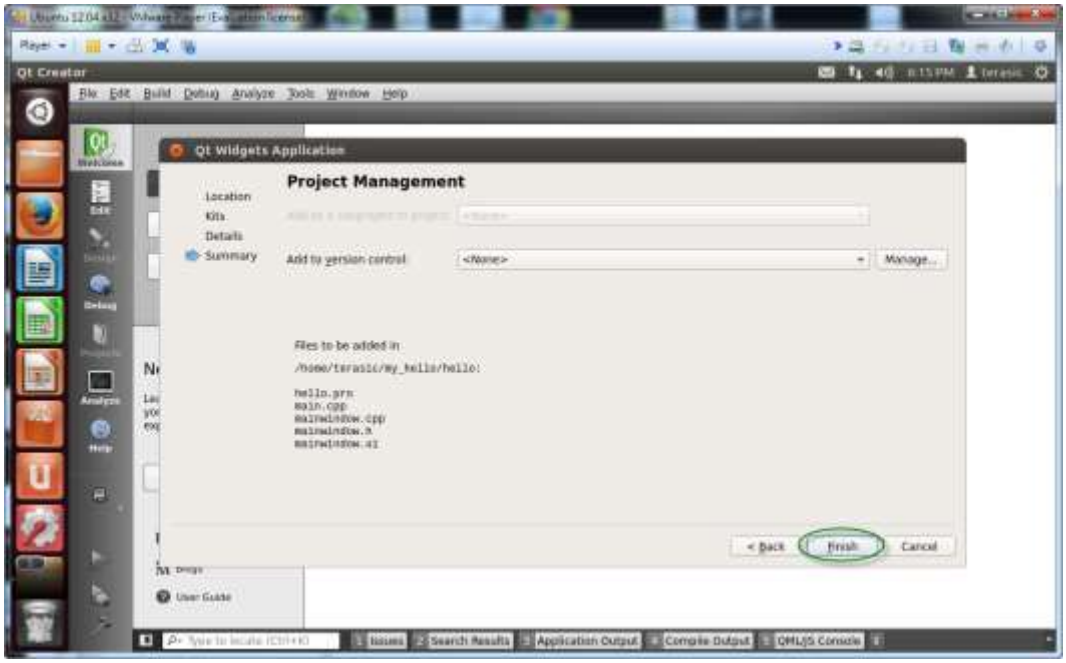

Figure 6-25 Final Dialog of Qt Gui Application

Figure 6-26 shows the hello project now has been created and the mainwindow.cpp and main.cpp files are included in the Sources folder under the hello project.

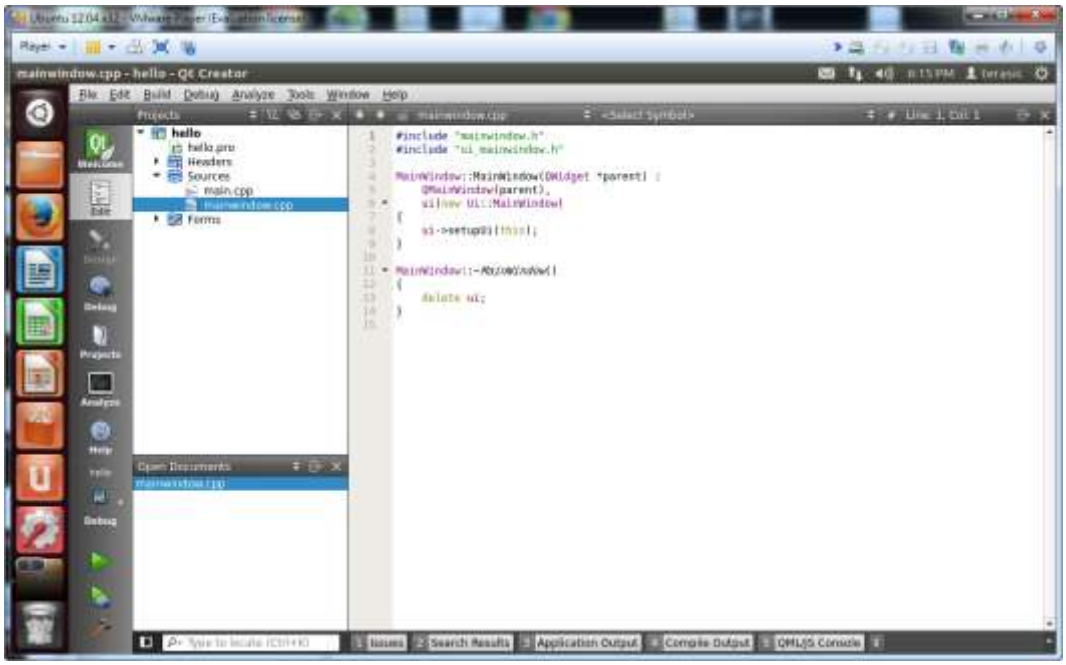

Figure 6-26 Mainwindow.cpp of the Hello project

■ GUI Design

Now, we start to design GUI. First, extend the **Forms** option under Projects, and double click "mainwindow.ui" to open the main GUI as shown in Figure 6-27.

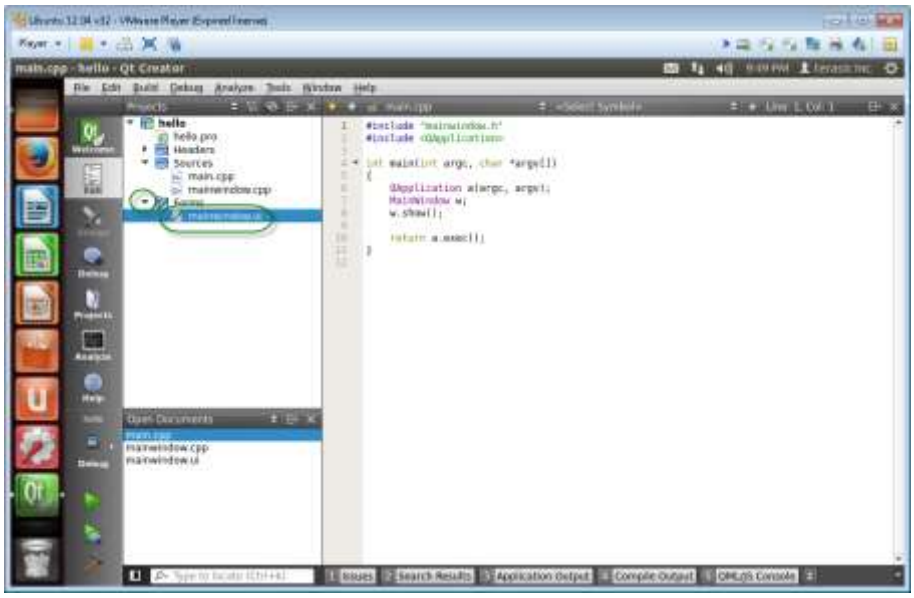

Figure 6-27 Open GUI

Figure 6-28 shows the opened main GUI under **Design** mode.

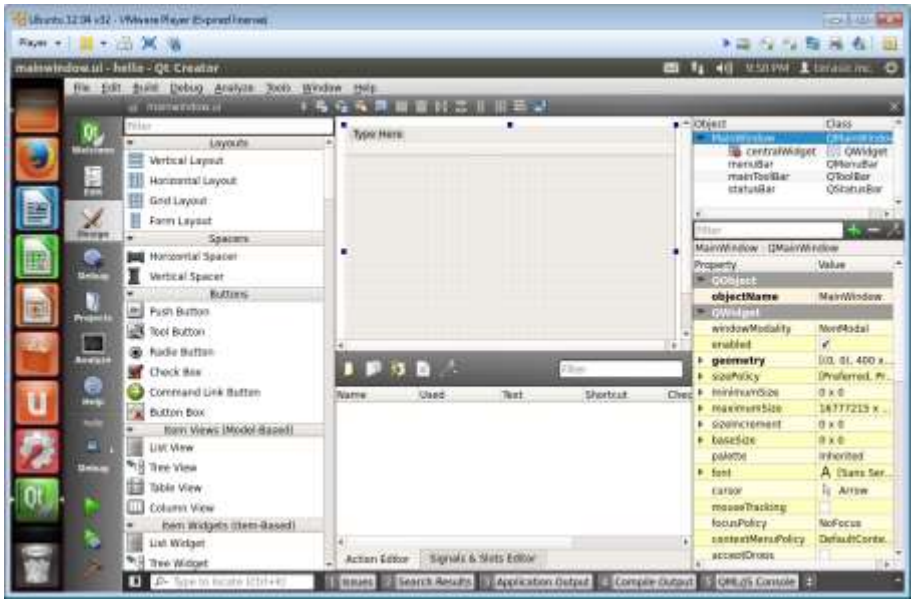

Figure 6-28 Main GUI

As shown in Figure 6-29, type "label" in the **Filter** edit box, and the **Label** component appears. Then, drag the **Label** component into Main GUI and drop it.

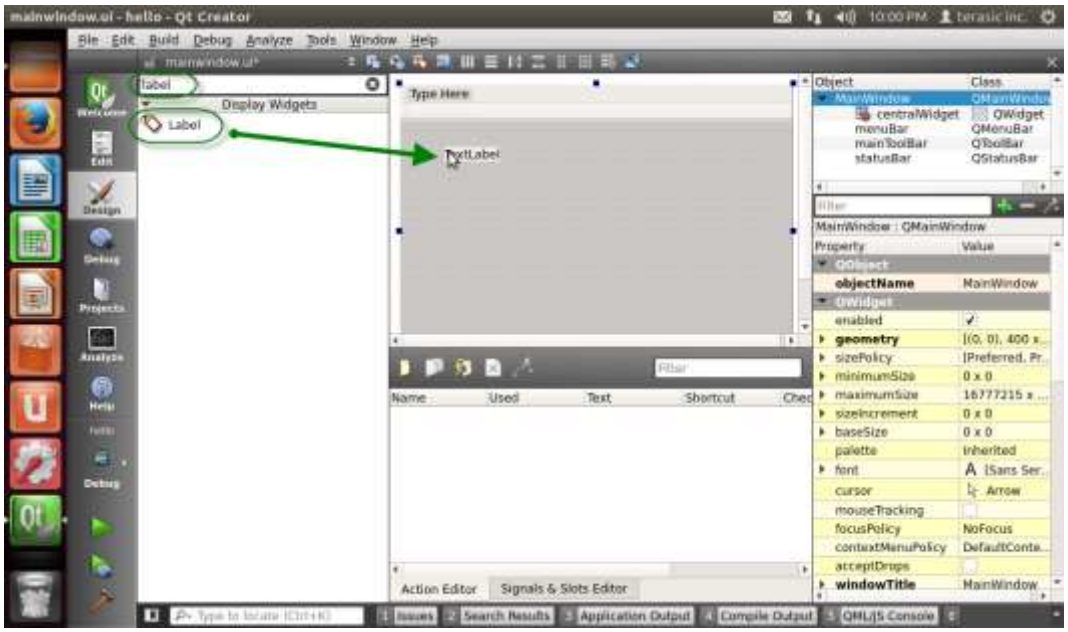

Figure 6-29 Add 'Label' GUI Component

Press the boundary of the **label** object to resize it as shown in **Figure 6-30**.

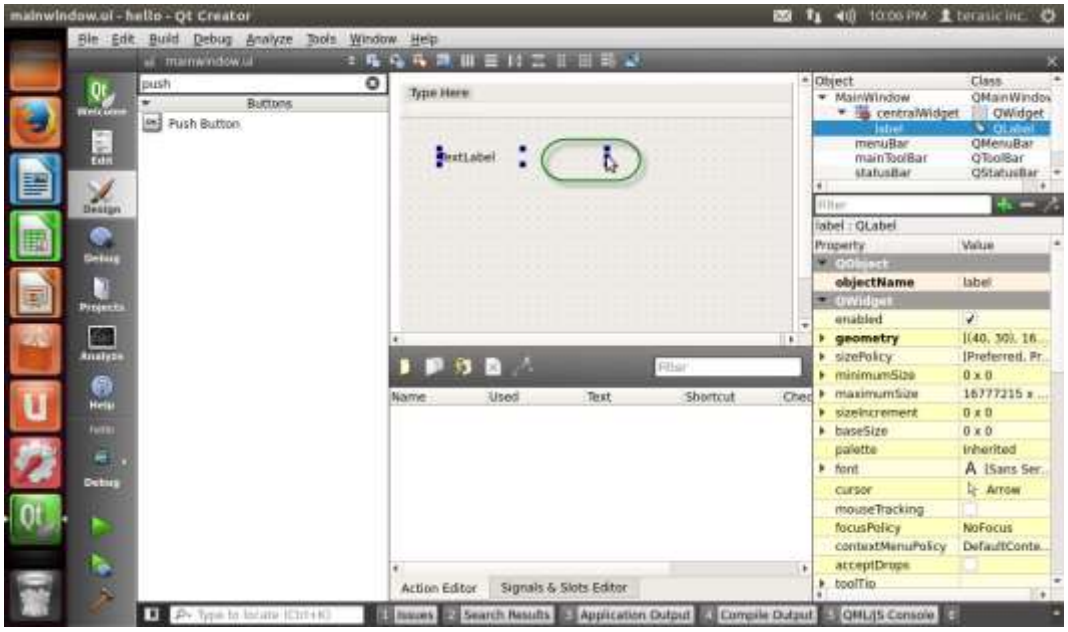

Figure 6-30 Resize label object

As shown in **Figure 6-31**, type "push" in the **Filter** edit box, and the **Push Button** component appears. Then, drag the **PushButton** component into Main GUI and drop it.

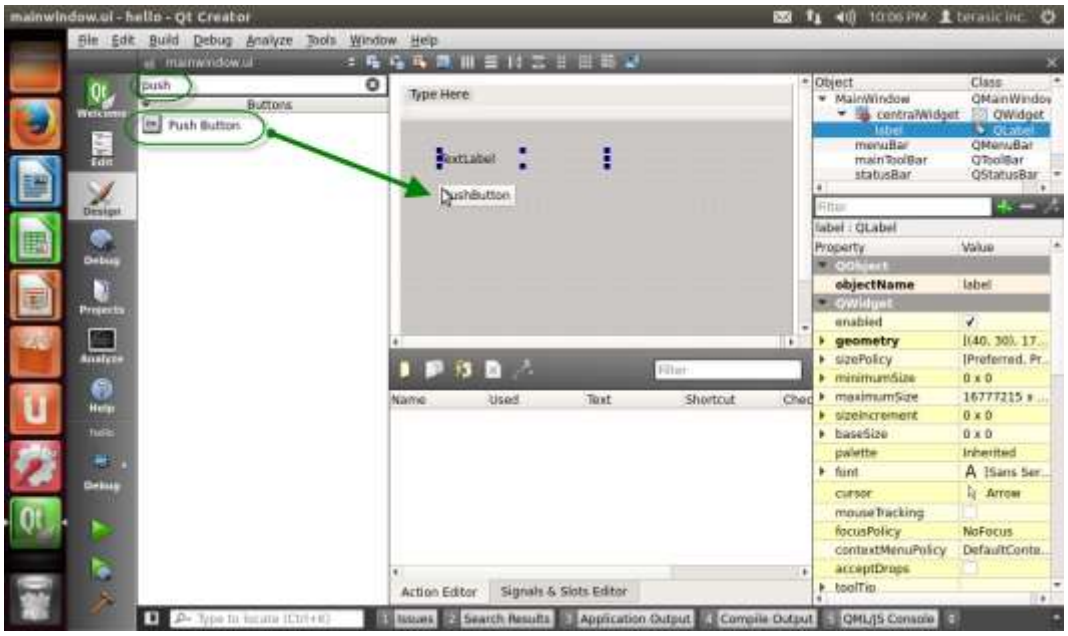

Figure 6-31 Add 'PushButton' GUI component

Press the boundary of the **PushButton** object to resize it as shown in **Figure 6-32**.

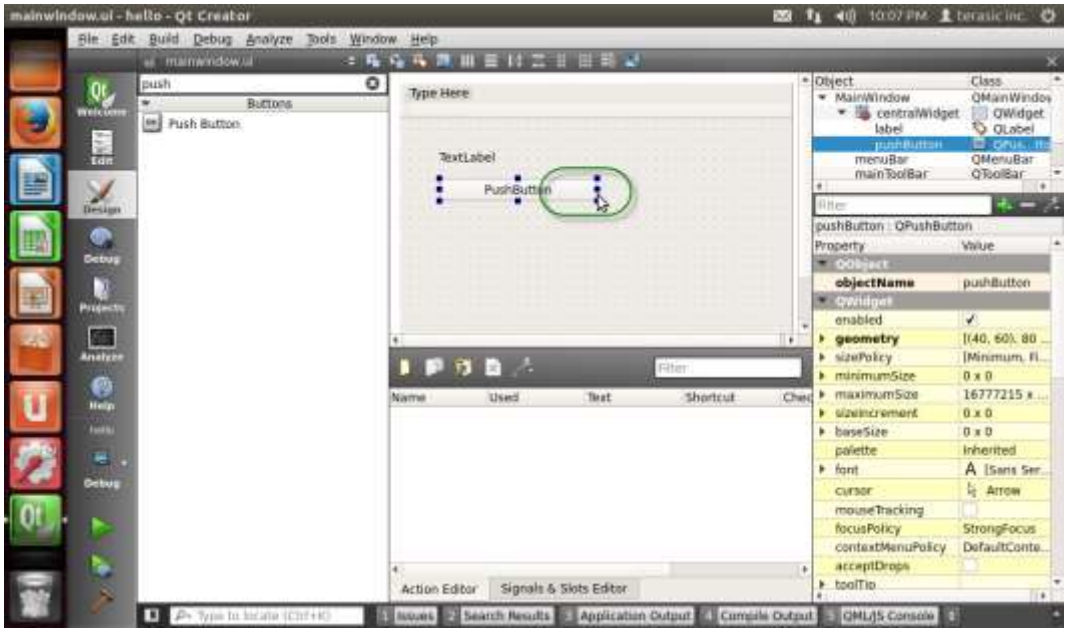

Figure 6-32 Resize 'PushButton' object

Right click the **PushButton** object to pop a menu, and select the "Go to slot…" item as shown in **Figure 6-33**.

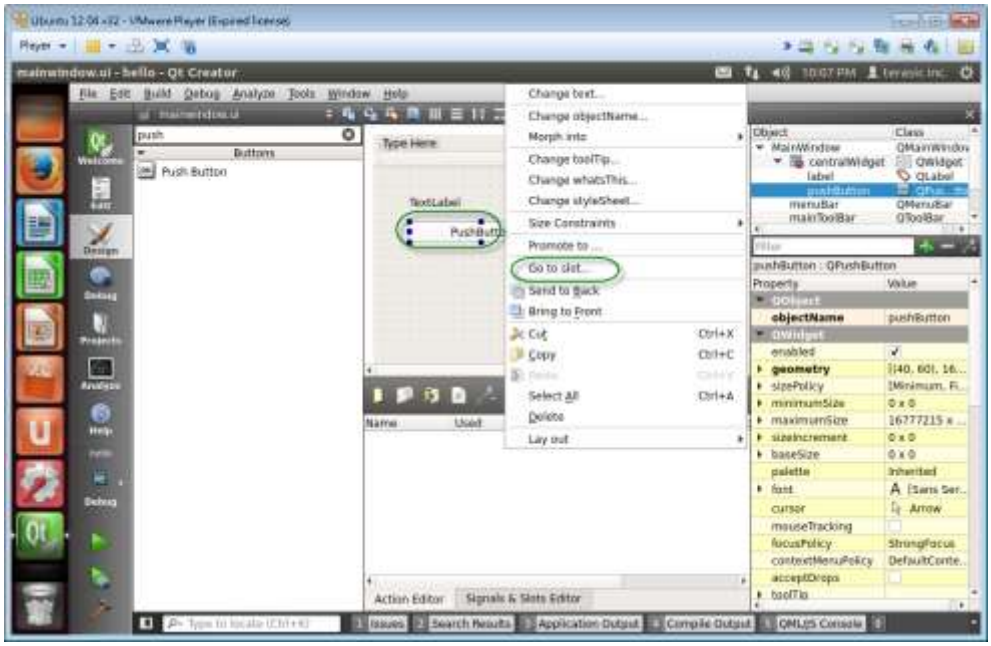

Figure 6-33 Launch 'Go to slot…" dialog for PustButton

When **Go to slot** dialog appears, select **click()** item and click **OK** as shown in **Figure 6-34**.

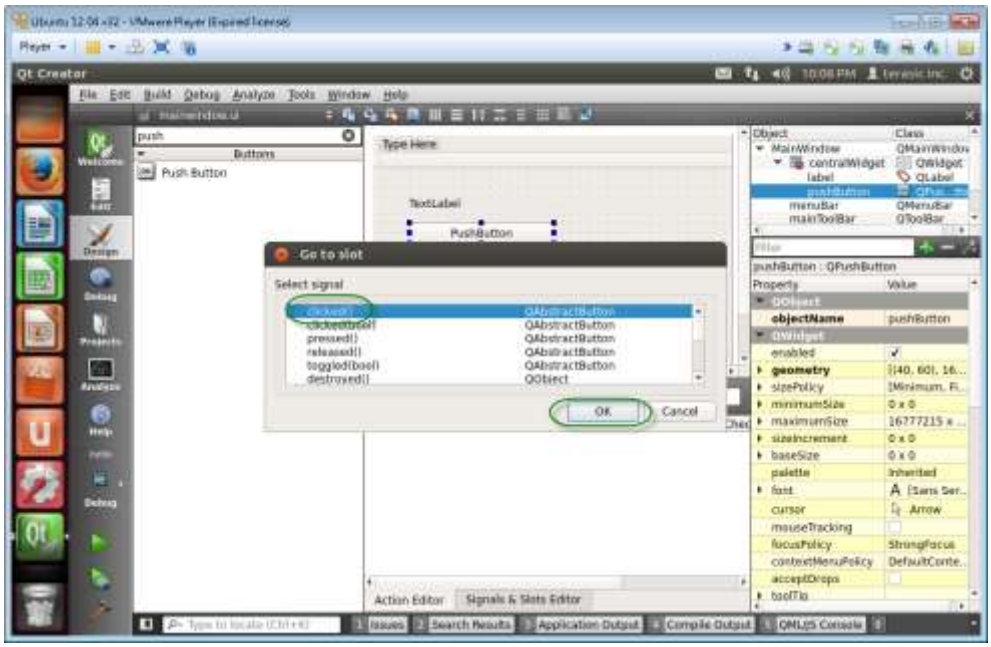

Figure 6-34 Select Signal for PushButton

A slot function "on_pushButton_clicked" is automatically created in mainwindow.cpp. Please add statement "ui->label->setText("Hello World!");" into the body of **on_pushButton_clicked** function

as shown in **Figure 6-35**.

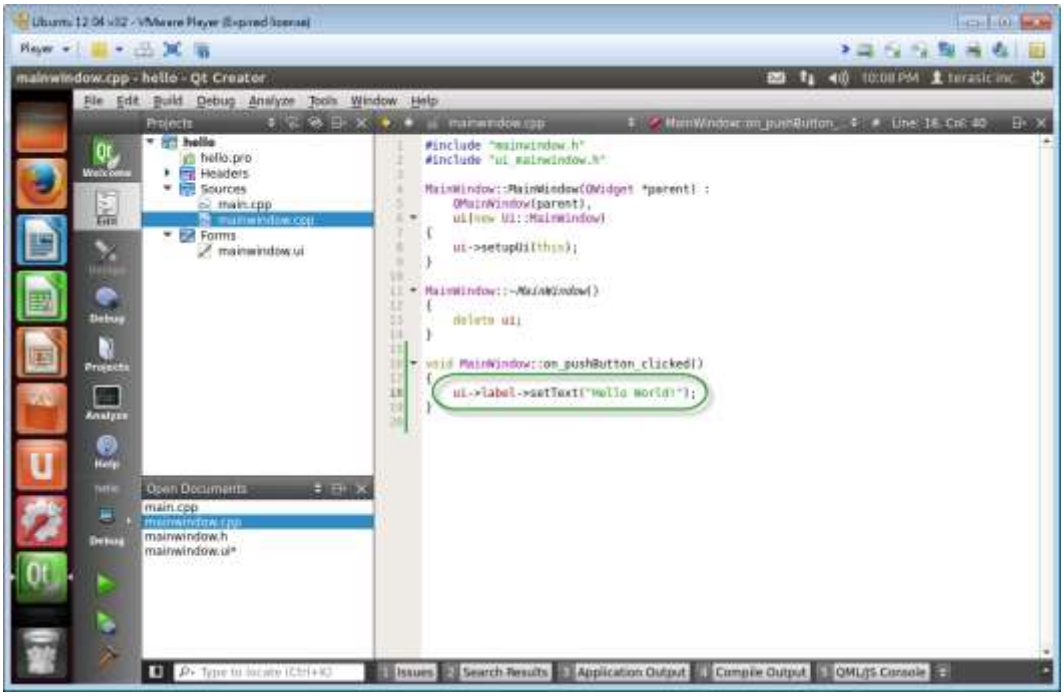

Figure 6-35 Implement Handle function for PushButton Clicked Signal

■ Build & Run

Click on the "Run" icon to build and run the Hello program. The Hello GUI program should appear as shown in **Figure 6-36**.

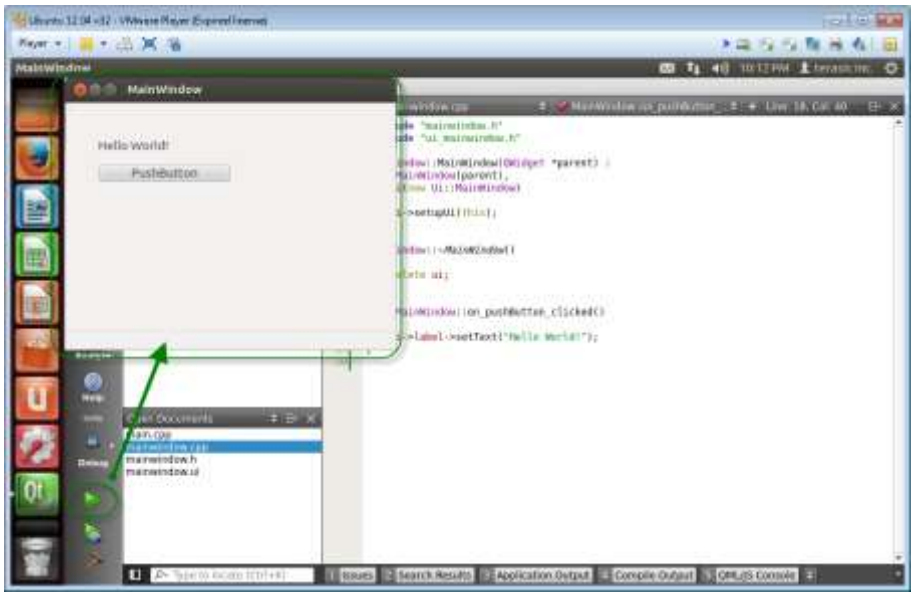

Figure 6-36 Run Hello Program

Click on the "PushButton", and then the label will display "Hello World!" as shown in **Figure 6-37**.

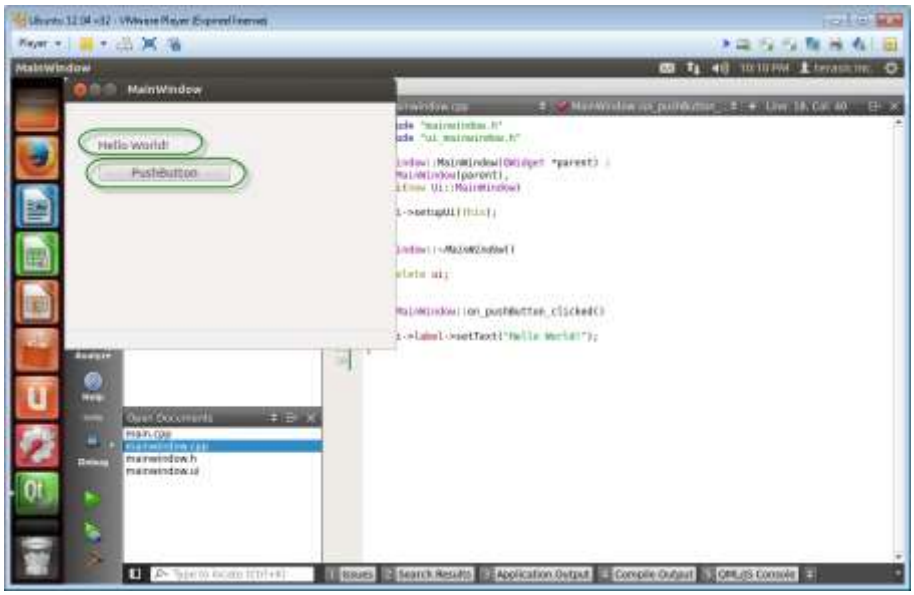

Figure 6-37 Test Hello Program

Clock the "Close" icon on the left-top corner, as shown in **Figure 6-38**, will close the Hello program.

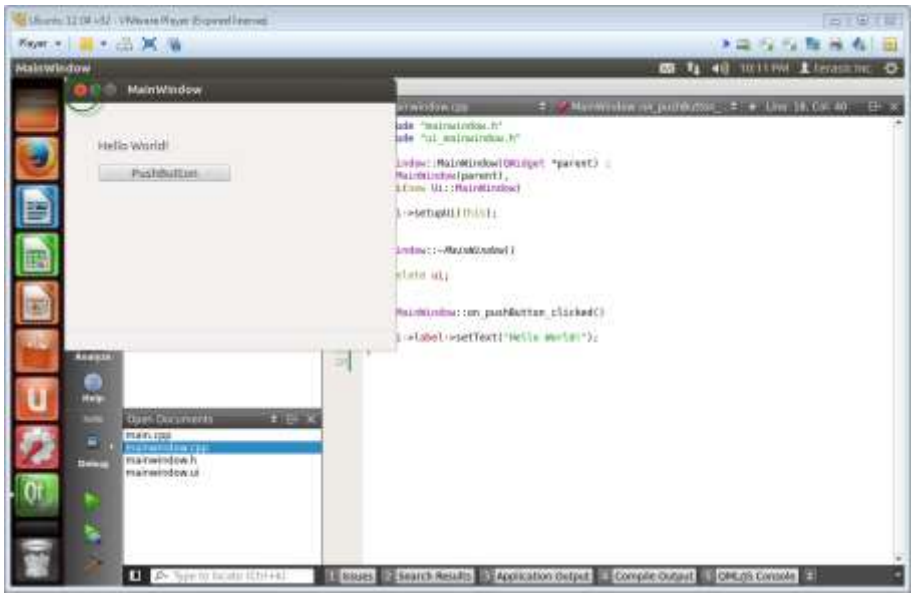

Figure 6-38 Close Hello Program

Chapter 7

Design Touch-Screen GUI Program

This chapter describes how to cross-compile the Hello project we created in previous chapter, such that the hello program can run on the Altera SoC Linux. Here we assume that Altera SoC Tool-Chain and **QT** and **tslib** touch-screen libraries for Altera SoC FPGA development board have been successfully installed on the Host Linux.

Here is a quick look of the 4 simple steps:

- Make sure QT and **stlib** libraries are installed
- Set up "Build & Run" in QT Creator
- Cross-compile the Hello project
- Execute Hello program

7.1 Installation of QT and stlib Libraries

This section describes how to install the pre-compiled **QT** and **tslib** libraries on the Host Linux. Developers can skip this section if they had cross-compiled these two libraries on the Host Linux. The pre-compiled **QT** and **tslib** libraries are available on the CD of TERASIC Linux BSP for touch-screen display as described in section 1.1 Linux BSP. The QT library is compressed in a file, named as:

qt-4.8.5-tslib-altera-soc.tar.gz2.

The **tslib** library is compressed in a file, named as:

tslib-altera-soc.tar.gz2.

■ Install QT Library

Please copy the file'qt-4.8.5-tslib-altera-soc.tar.gz2' to the folder "/user/local" in the Host Linux. In Linux terminal, type the below command to go to the /usr/local folder.

```
$cd /usr/local
```

Then, type the below command to decompress the library. The library will be installed on the folder "/usr/local/qt-4.8.5-tslib-altera-soc".

```
$tar -jxv -f qt-4.8.5-tslib-altera-soc.tar.gz2
```

■ Install tslib Library

Please copy file 'tslib-altera-soc.tar.gz2' to the folder "/user/local" in the Host Linux. In Linux terminal, type the below command to go to the /usr/local folder.

```
$cd /usr/local
```

Then, type the below command to decompress the library. The library will be installed in the folder "/usr/local/tslib-altera-soc".

```
$tar -jxv -f tslib-altera-soc.tar.gz2
```

7.2 Set up "Build & Run" in QT Creator

This section describes how to set up cross-compile and QT library for QT Creator.

■ Compiler Setup

First launch the QT Creator, and select the menu item "Tools→Options…" as shown in **Figure 7-1** to open the **Option** dialog.

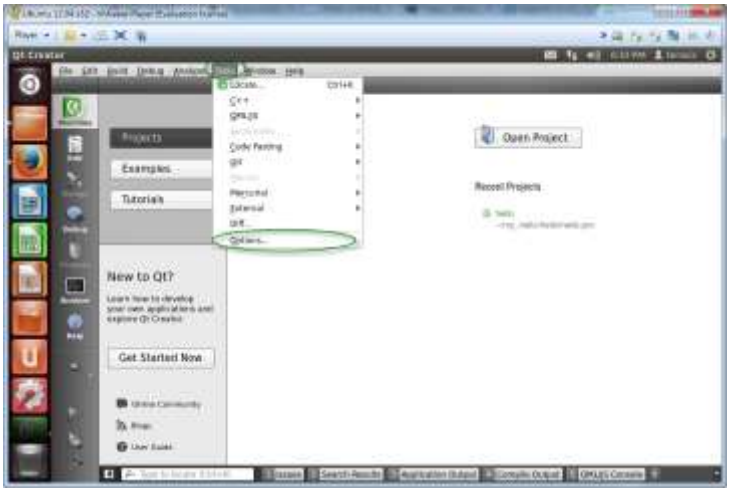

Figure 7-1 Open Option Dialog

To add the "GCC" compiler, in the Option dialog, select "Build & Run" item on the left and then select "Compilers" tab. Now you can select "GCC" item from the "Add" pull down menu as shown in **Figure 7-2**.

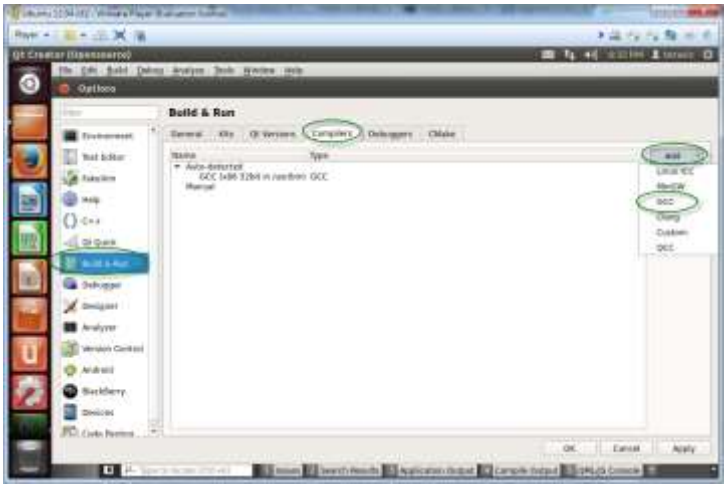

Figure 7-2 Add GCC Compiler

Next, you need to specify the GCC compiler details. First, please type in "GCC (Altera SoC)" in the Name option box and click "Browse…" to select the location of the compiler of Altera SoC ARM. The compiler is located at:
"/home/**terasic**/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux/
/bin/arm-linux-gnueabihf-g++". Note that in the path string, you should replace "terasic" with your linux user name. Then click "Apply" to finish the **Compiler** setup as shown in **Figure 7-3**.

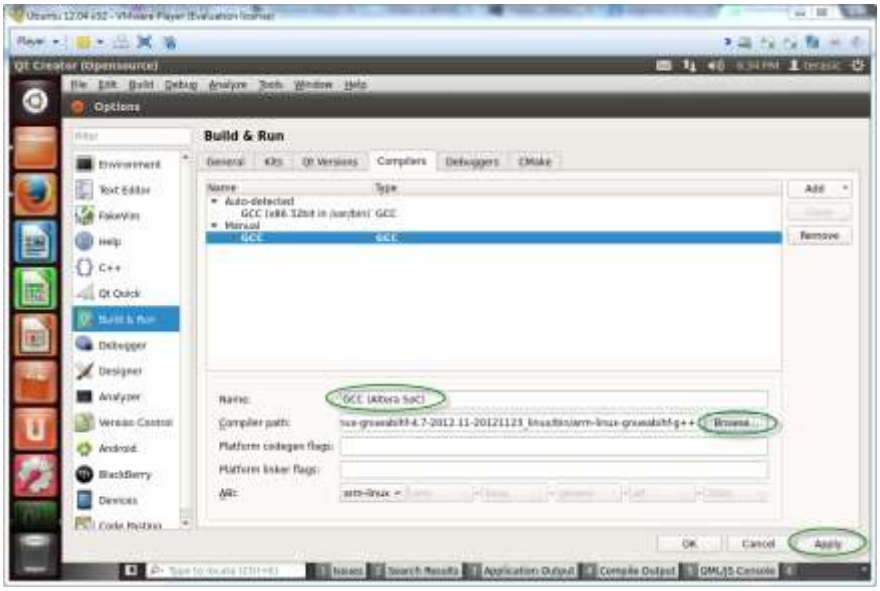

Figure 7-3 Specify Compiler Name and Path

■ Qt Verisons Setup

To add Qt Versions, first click on the "Qt Versions" tab and click "Add" as shown in **Figure 7-4**. When an **Open File** dialog appears to ask the location of qmake executable, please specify the location as: "**/usr/local/qt-4.8.5-tslib-altera-soc/bin/qmake**".

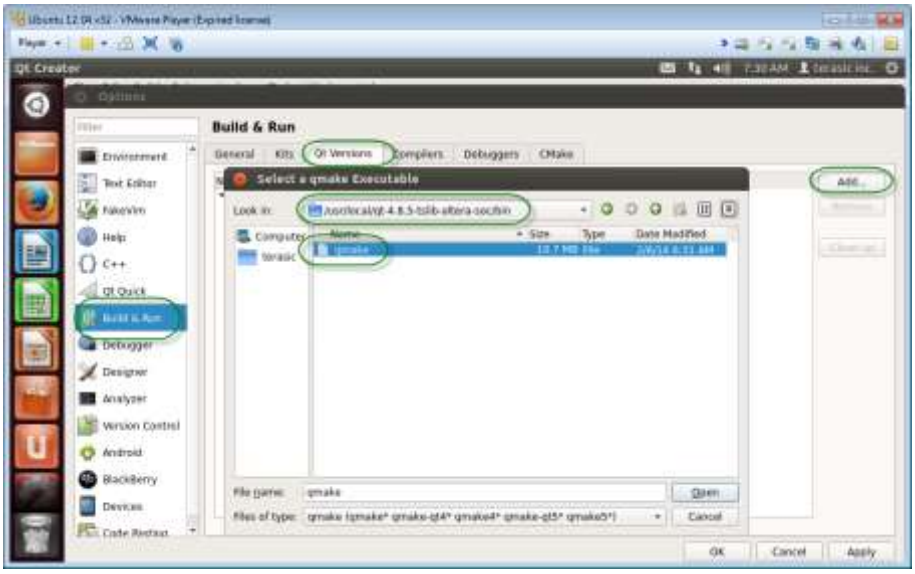

Figure 7-4 Add Qt Version

Finally, click "Apply" to complete the **Qt Versions** setup as shown in **Figure 7-5**.

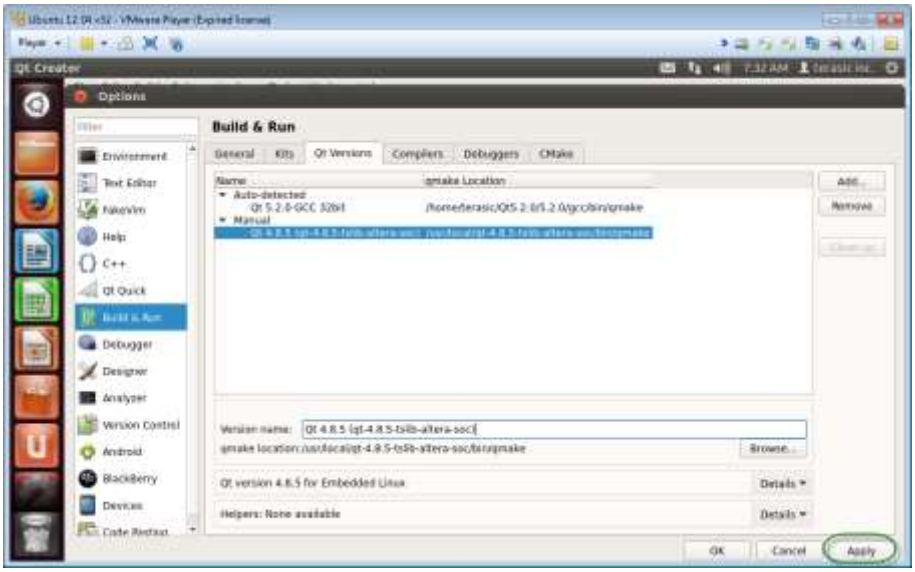

Figure 7-5 Apply the Qt Version

■ Kits Setup

To add Kits, first click on the "Kits" tab and click on "Add" as shown in **Figure 7-6**. Specify the kit detail as below:

- Name: Altera SoC FPGA Kit
- Device Type: Select "Generic Linux Device"
- Compiler: Select "GCC (Altera SoC)"
- Qt Version: Select "(Qt 4.8.5 (qt-4.8.5-tslib-atera-soc)"

Then, click "Apply" to finish **Kits** setup and click "OK" to finish Options setup.

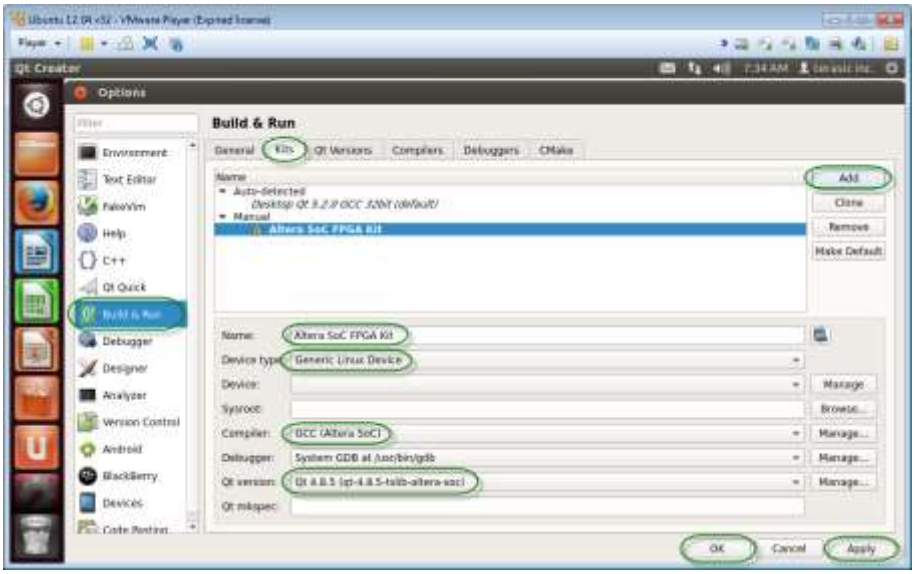

Figure 7-6 Add Altera SoC FPGA Kit

7.3 Cross-Compile the Hello Project

Now, after properly setting up "Build & Run" settings, we are ready to cross-compile the Hello project. First, please launch the QT Creator, and select the menu item "File→Recent Projects" to open the hello project as shown in **Figure 7-7**.

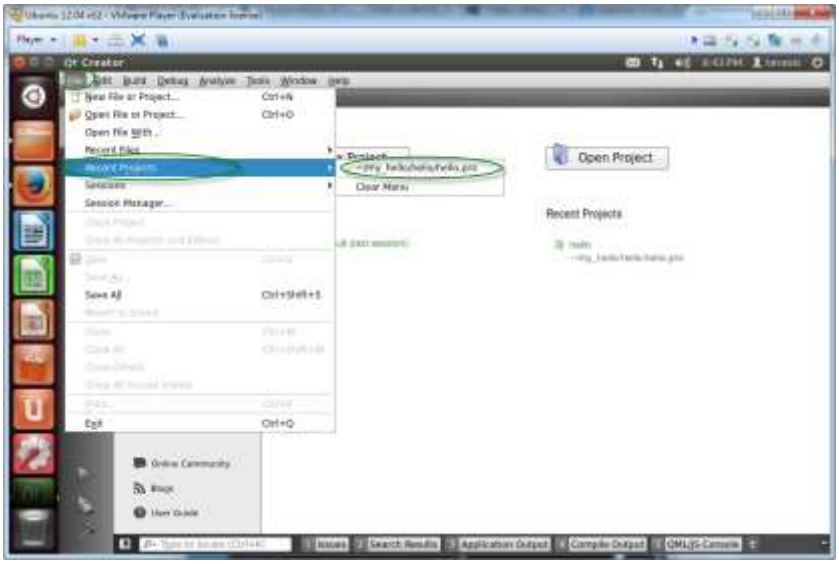

Figure 7-7 Open the Hello Project

To add "Altera SoC FPGA Kit" into the Hello project, click on the "Projects" icon and select "Altera SoC FPGA Kit" from the "Add Kit" pull-down menu, as shown in **Figure 7-8**.

Figure 7-8 Add Altera SoC FPGA Kit into the Hello Project

To specify release build for Altera SoC FPGA Kist, click on “Kit Selector icon and select “Altera SoC FPGA Kit” item under the **Kit** list menu and “Release” in **Build** list menu, as shown in **Figure 7-9**.

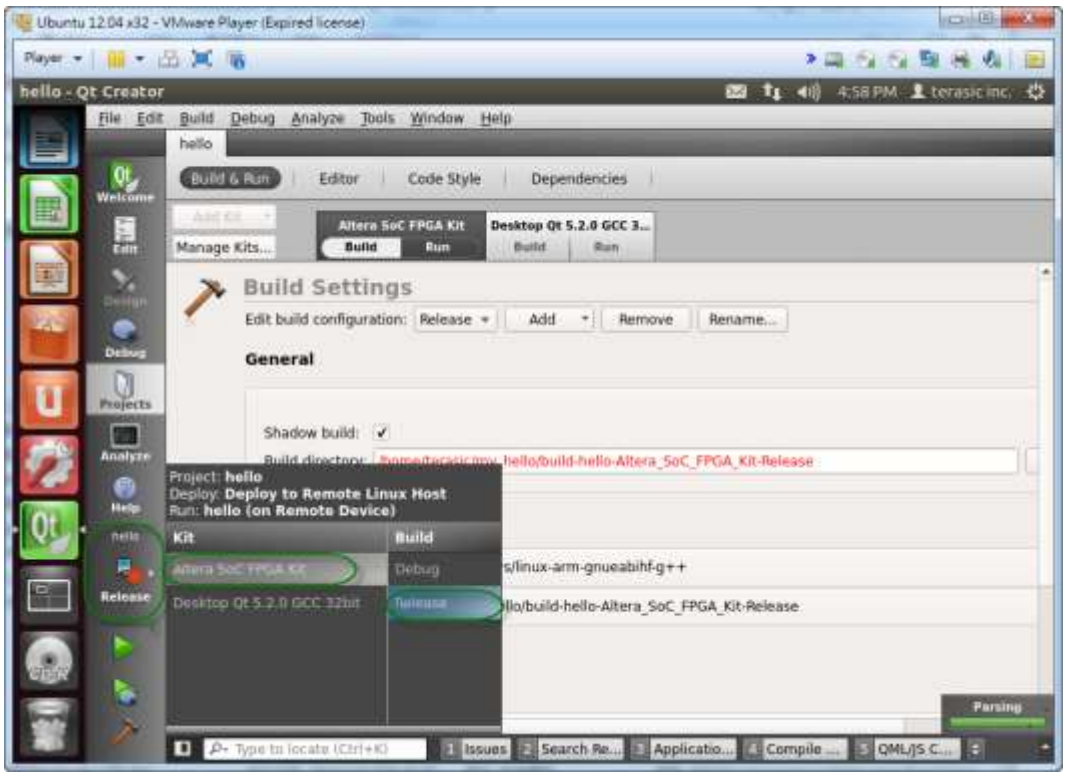

Figure 7-9 Specify Release Setting

To compile the hello project, click on "Compile Output" item on the bottom toolbar of the QT Creator and select the menu item "Build→Rebuild All" as shown in **Figure 7-10**. While compiling, the system message will be simulaneously displayed on the "Compile Output" Windows.

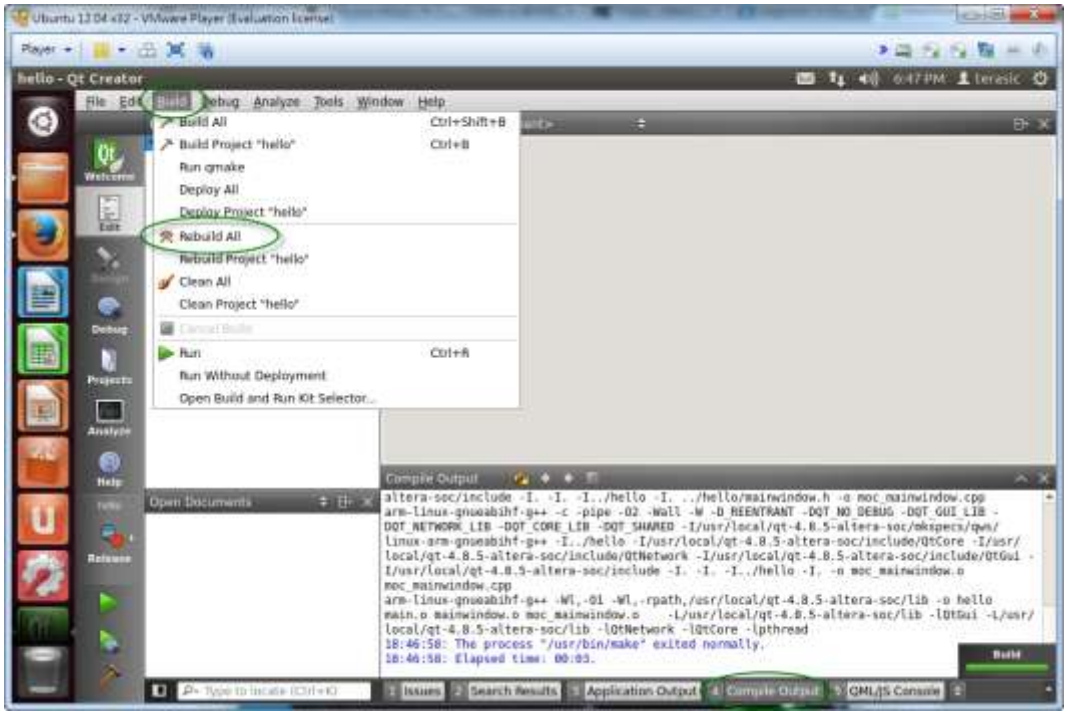

Figure 7-10 Build Hello Project

When the build process has been complete, the output execution file can be found under the folder "/home/**terasic**/my_hello/build-hello-Altera_SoC_FPGA_Kit-Release/hello". Note that in the path string, you should replace "terasic" with your linux user name as shown in **Figure 7-11**.

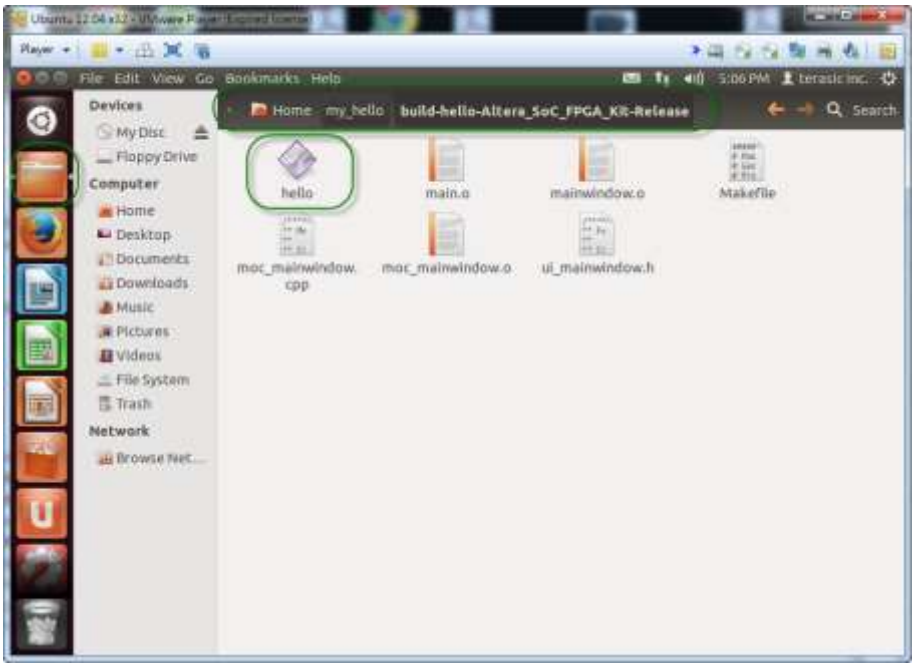

Figure 7-11 Location of the Output Hello Execution File

7.4 Execute Hello Program

To execute the Hello program on the Altera SoC FPGA board, first we need to copy the **hello** execution file to the Altera SoC Linux. For detailed copying operation, please refer to the section 8.1 Copy files to Altera SoC Linux. The cross-compile execution file for Altera SoC Linux is located at:

"/home/**terasic**/my_hello/build-hello-Altera_SoC_FPGA_Kit-Release/hello"

In the following, we assume that Linux Console is used and the hello execute file is copied to the "/home/root" folder of Altear SoC Linux.

■ Launch Hello Program

Before we launch the hello program, we first type in

```
$ cd /home/root
```

to go to the /home/root directory, as shown in **Figure 7-12**.

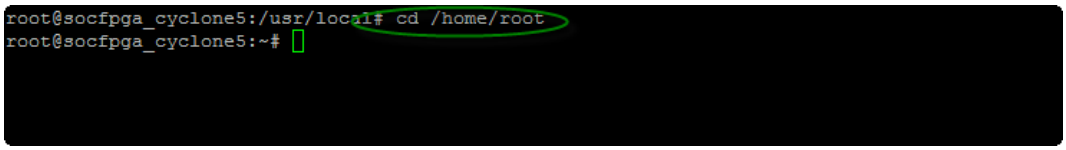

Figure 7-12 Go to /home/root Directory

Now launch hello program by typing in the following command:

```
$ ./hello -qws
```

as shown in **Figure 7-13**.

Figure 7-13 Launch Hello Program

If the **Hello** program is launched successfully, you should see the **Hello** Windows as shown in **Figure 7-14**. Touch "PushButton" on touch-screen and then the **label** content will become "Hello World!". Touch the **close** icon on the right-top corner of Windows can close **Hello** program.

Figure 7-14 Screenshot of the Hello Program

7.5 Quartus HPS Qsys Project

As described in section 1.1 Linux BSP, the Quartus HPS Qsys project is available on the CD-ROM. If developers need to add new controllers on FPGA, they can modify this Quartus project, generate .sof file, and translate the .sof to .rbf file. Finally, copy the .rbf to the mircoSD card for booting Linux. For details, please refer to the Control Panel development manual in Altera DE1-SoC CD. This CD is available on the web page: http://cd-de1-soc.terasic.com.

Chapter 8

Appendix

8.1 Copy files to Altera SoC Linux

This section describes how to use “scp” Linux command to copy files from Host Linux to Altera SoC Linux. First, please make sure both Host Linux and Altera SoC Linux are connected to DHCP Server as shown in Figure 8-1.

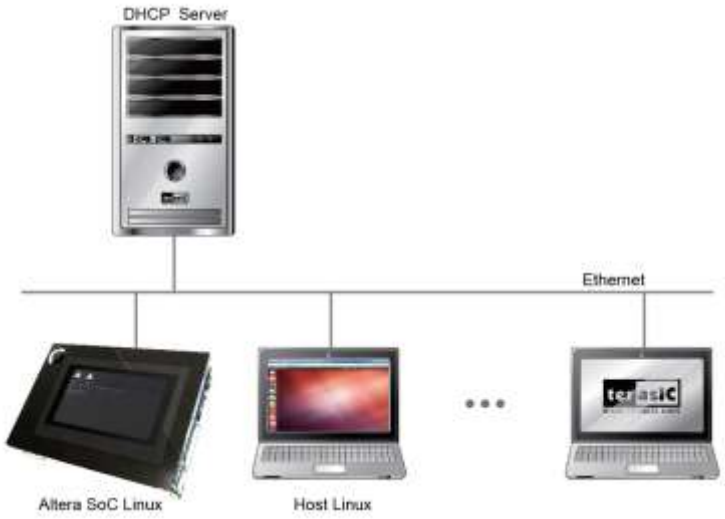

Figure 8-1 Connect to DHCP Server

Here we illustrate how to copy the file “tslib-altera-soc.tar.gz2” in the folder “/usr/local” of Host Linux to the folder “/usr/local” of the Altera SoC Linux by using **scp** Linux command. Here shows the copy procedures in step by steps:

1. Make sure the Altera SoC Board is connected to DHCP server, and the Linux booting

microSD card had been attached to the Altera SoC Board.

2. Power on the Altera SoC Board, and type “root” to login to the Altera SoC Linux as shown in Figure 8-2.
3. In terminal of Altera SoC Linux, type “ifconfig” to checked the assigned IP address as shown in Figure 8-3. Assumed that the assigned IP address is “192.168.1.182”
4. Make sure the Host Linux is connected to DHCP server. In Host Linux, enter Linux terminal, and type “cd /usr/local” to go to the folder “/usr/local” as shown in Figure 8-4.
5. Type “scp tslib-altera-soc.tar.gz2 root@192.168.1.182:/usr/local” in the terminal of Host Linux as shown in Figure 8-5. During the first time of the connection, system will ask “Are you sure you want to continue connecting (yes/no)?”. Please type in “yes” and press ENTER. After confirming, the system will ask you to input password for the remote machine, please type “**terasic**” (the default setting with the microSD image) and press ENTER.
6. Now, the file “tslib-altera-soc.tar.gz2” should be successfully copied to the “/usr/local” folder of Altera SoC Linux. You can check it by typing “cd /usr/local” in Altera SoC Linux to go to the “usr/local” folder. Then, type “ls” to list all files in the folder. The file “tslib-altera-soc.tar.gz2” will be expected to appear in the list as shown in Figure 8-5.

Figure 8-2 Login to Altera SoC Linux

```
root@socfpga:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 32:ba:9e:5a:6d:0e
          inet addr:192.168.1.182  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:419 errors:0 dropped:12 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:35801 (34.9 KiB)  TX bytes:1248 (1.2 KiB)
          Interrupt:152

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@socfpga:~#
```

Figure 8-3 Check Assigned IP Address

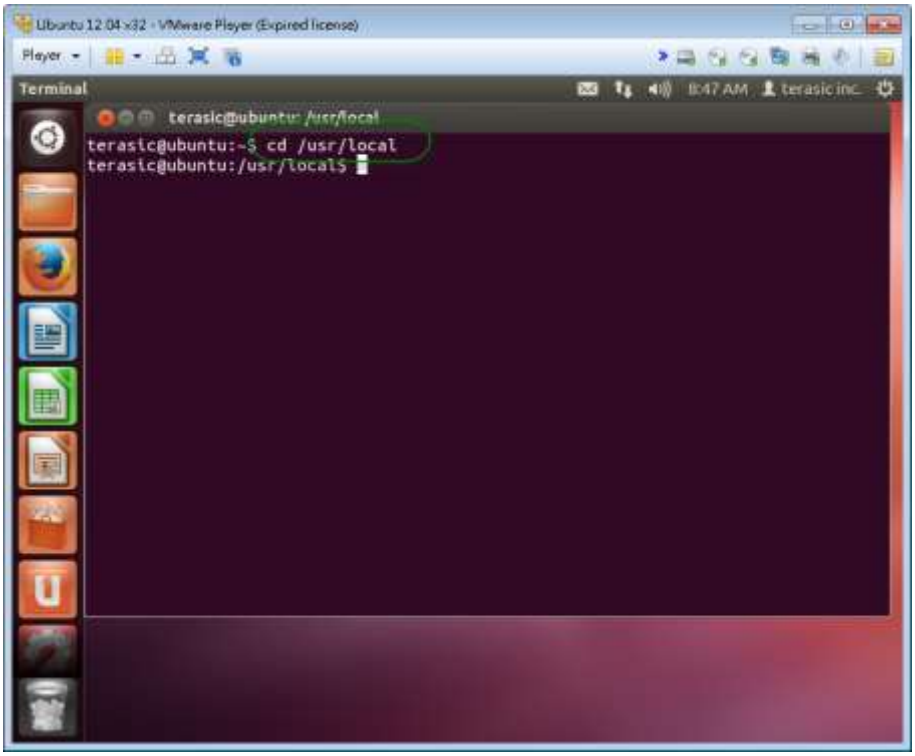

Figure 8-4 Go to /usr/local folder in Host Linux

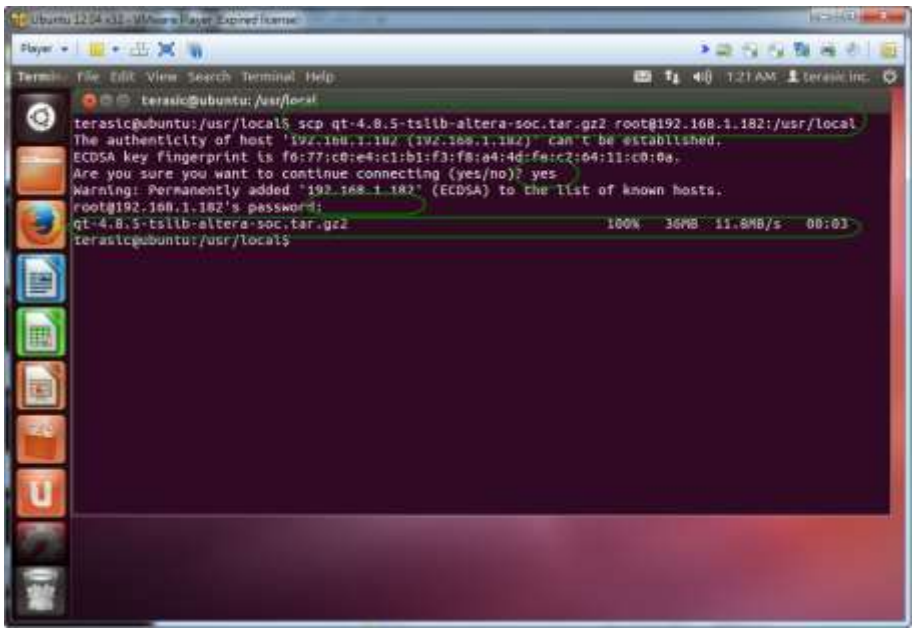

Figure 8-5 Execute 'scp' Command

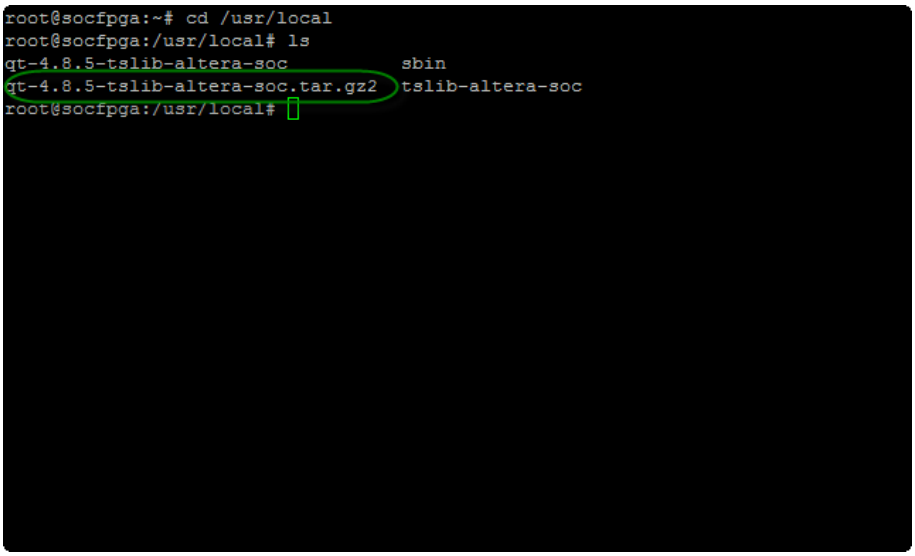

Figure 8-6 Check qt-4.8.5-tslbi-altera-soc.tar.gz2