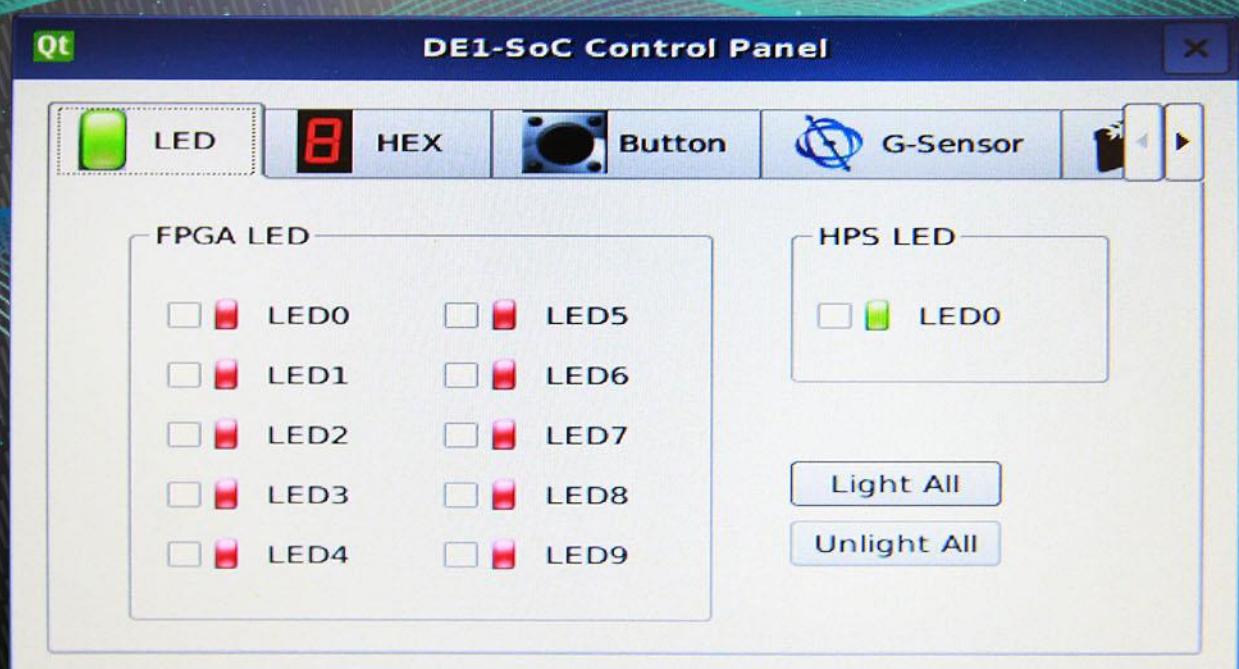


# DE1-SoC

## CONTROL PANEL



**terasic**  
www.terasic.com

**ALTERA**  
UNIVERSITY  
PROGRAM

**CONTENTS**

<b>CHAPTER 1      <i>OVERVIEW</i>.....</b>	<b>4</b>
1.1 SYSTEM BLOCK DIAGRAM .....	4
1.2 LEARNING TOPICS .....	5
1.3 SYSTEM REQUIREMENTS .....	6
1.4 DEVELOPMENT FLOW .....	6
1.5 SET UP THE CONTROL PANEL DEMO .....	6
<b>CHAPTER 2      <i>LINUX INSTALLATION</i>.....</b>	<b>13</b>
2.1 SYSTEM REQUIREMENTS .....	13
2.2 INSTALL VMWARE PLAYER .....	14
2.3 LAUNCH VMWARE.....	19
2.4 INSTALL LINUX UBUNTU DESKTOP .....	20
2.5 UPGRADE LINUX SOFTWARE PACKAGE .....	29
<b>CHAPTER 3      <i>QT CREATOR INSTALLATION</i>.....</b>	<b>34</b>
3.1 INSTALL TOOL-CHAIN FOR LINUX x86 .....	34
3.2 DOWNLOAD AND INSTALL QT INSTALLER .....	36
3.3 LAUNCH QT CREATOR AND CHECK CONFIGURE .....	45
3.4 HELLO PROGRAM.....	49
<b>CHAPTER 4      <i>ALTERA SOC TOOL-CHAIN INSTALLATION</i>.....</b>	<b>55</b>
4.1 DOWNLOAD AND INSTALL TOOL-CHAIN .....	55
4.2 SET UP TOOL-CHAIN PATH.....	56
<b>CHAPTER 5      <i>BUILD QT LIBRARY FOR ALTERA SOC</i>.....</b>	<b>58</b>
5.1 DOWNLOAD THE QT SOURCE CODE.....	58
5.2 CREATE A NEW ‘MKSPECS’ FOR ALTERA SOC.....	60
5.3 CONFIGURE, BUILD, AND INSTALL QT LIBRARY .....	62
<b>CHAPTER 6      <i>QT APP FOR ALTERA SOC</i>.....</b>	<b>68</b>
6.1 SET UP “BUILD & RUN” IN QT CREATOR.....	68
6.2 CROSS-COMPILE THE HELLO PROJECT .....	72
6.3 EXECUTE HELLO PROGRAM .....	76

<b>CHAPTER 7      <i>CONTROL PANEL QUARTUS PROJECT</i></b>	<b>85</b>
7.1 BUILD QUARTUS PROJECT OF CONTROL PANEL.....	86
7.2 TEST FPGA CONFIGURATION FILE .....	86
7.3 MORE ON THE QUARTUS PROJECT OF CONTROL PANEL.....	87
<b>CHAPTER 8      <i>CONTROL PANEL QT PROJECT</i></b>	<b>91</b>
8.1 COPY CONTROL PANEL QT PROJECT .....	97
8.2 BUILD CONTROL PANEL QT PROJECT .....	102
8.3 EXECUTE CONTROL PANEL PROGRAM.....	107
8.4 MORE ON THE CONTROL MODULE OF CONTROL PANEL.....	108

# Chapter 1

## Overview

The Altera Cyclone V SoC Development Kit (DE1-SoC), a robust platform that is built around Altera System-on-Chip (SoC) FPGA which combines the dual-core Cortex-A9 embedded cores with industry leading programmable logic for ultimate design flexibility. Altera's SoC FPGA integrates Hard Processor System (HPS) consisting of processor, peripherals and memory interfaces tied seamlessly with the FPGA fabric using a high-bandwidth interconnect backbone. Other than having the Altera SoC FPGA, DE1-SoC FPGA development board is equipped with hardware, such as DDR3 memory, video and audio capabilities, Ethernet networking, and much more that promise many exciting applications.

To demonstrate the power of the SoC FPGA on the DE1-SoC FPGA development board, we have created a “Control Panel” program, a board utility software that uses ARM to access various peripherals that belong both to the Hard Processor System (HPS) and the FPGA. This demo is particularly useful for anyone who is interested in the SoC FPGA structure and development of future SoC applications.

This scope of this tutorial is to help users understand and learn how to build the board utility “Control Panel” on the DE1-SoC FPGA development board in a step-by-step fashion. To boost user confidence while following the instructions, we provide screenshots of steps needed to be followed. This helps users debug if they encounter any problems during the learning process. Users should have basic concepts toward building their own applications once they have finished working with the whole tutorial. Also, users should find this tutorial self-explanatory and self-contained as we hope that lots of creative projects can be inspired as a result of this.

### 1.1 System Block Diagram

**Figure 1-1** shows the block diagram of the Control Panel where user can see the Control Panel program is running on Linux on the left-hand side. The program GUI is built based on QT library and it can access the FPGA resources through the AXI bus. The Linux Frame Buffer Display

hardware is implemented based on Altera VIP suite and HPS DDR3 SDRAM memory is used as frame buffer.

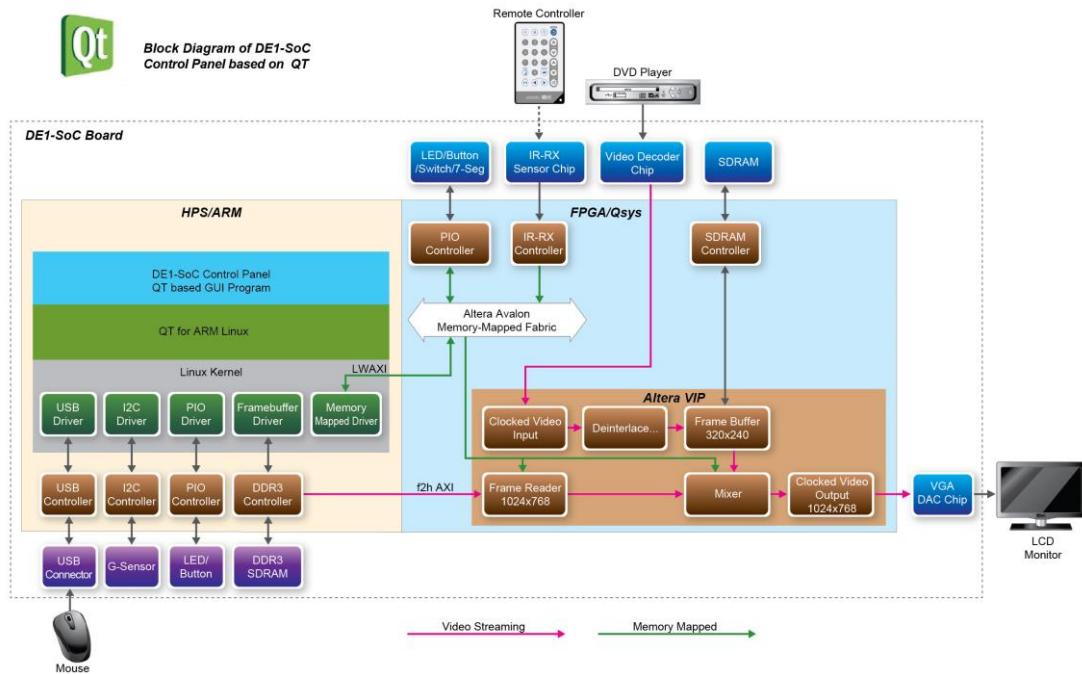


Figure 1-1 Block Diagram of the Control Panel

## 1.2 Learning Topics

To give users an outline what they will learn in this manual, here we provide a list of topics that will be covered and explained:

- Install a virtual machine on Windows Host
- Install a Linux x86 under virtual machine
- Install QT Designer on Linux x86
- Create and build a QT hello program for Linux x86
- Cross-compile to build QT library for Altera SoC ARM
- Cross-compile to build QT Application for Altera SoC ARM
- Create and build a Quartus Project for Altera SoC FPGA
- Launch QT application on Altera SoC FPGA board

## 1.3 System Requirements

Before starting this tutorial, please note that the following items are required to complete the Control Panel project:

- A x86 PC
- Altera DE1-SoC FPGA board
- Linux Ubuntu 12.04 x32 Installer
- Options if Windows Host is preferred
  - MS Windows Installer
  - VMware Player Installer
- QT 5.2.0 Designer Installer
- QT 4.8.5 open source
- Quartus 13.0 or later installer

## 1.4 Development Flow

Our Control Panel program GUI is built using the QT library. Considering that most users might not be familiar with the QT library, we believe it would be most helpful to first list all the steps required to complete the control panel project in the following:

1. Install Linux x86 on Windows Host
2. Install QT 5.2.0 Creator
3. Install Altera SoC Tool-Chain
4. Build QT 4.8.5 Library for Altera SoC ARM
5. Build QT App for Altera SoC FPGA Board
6. Build Quartus Project
7. Execute QT App on Altera SoC FPGA Board

## 1.5 Set up the Control Panel Demo

This section describes how to set up the Control Panel program on the Altera DE1-SoC Development Kit. There are five steps involved. Please carefully follow the instructions below:

1. Download microSD card writer utility – **Win32 Disk Imager**
2. Decompress the ControlPanel image file, and write the file into a 4GB microSD card with an ImageWriter Utility
3. Insert the microSD card intro Altera DE1-SoC FPGA Development Board, connect a USB keyboard and mouse to the USB host port, connect a LCD monitor to the VGA port, then power on the board to boot Linux System.
4. Login in Linux with ‘root’ user name
5. Type “./ControlPanel –qws” to launch Control Panel.

## ■ 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**.

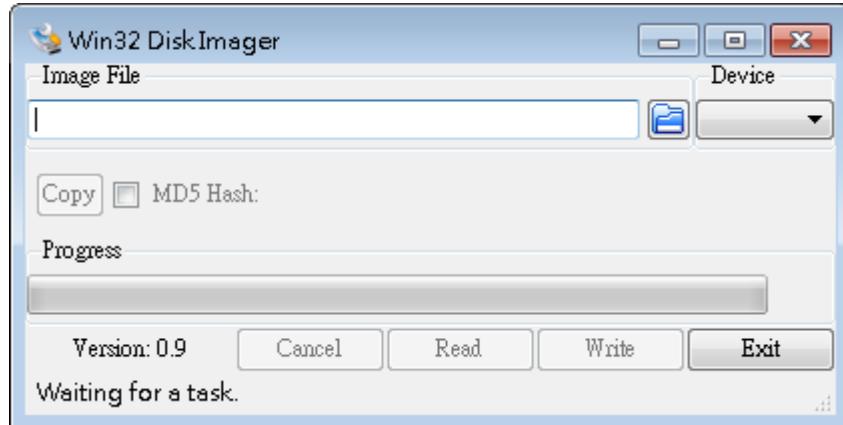


Figure 1-3 Win32 Disk Imager

## ■ Create Linux Booting microSD card

First, we need to prepare the microSD card image. The Control Panel microSD card image is stored at \sdcard\_image\DE1SOC\_ControlPanel\_SDCARD.zip in the compressed format. You need to decompress the file to get the image file DE1SOC\_ControlPanel\_SDCARD.img.

Then, insert a 4GB 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 DE1SOC\_ControlPanel\_SDCARD.img. Click “Write” to start writing the image file into the microSD card as shown in **Figure 1-4**.

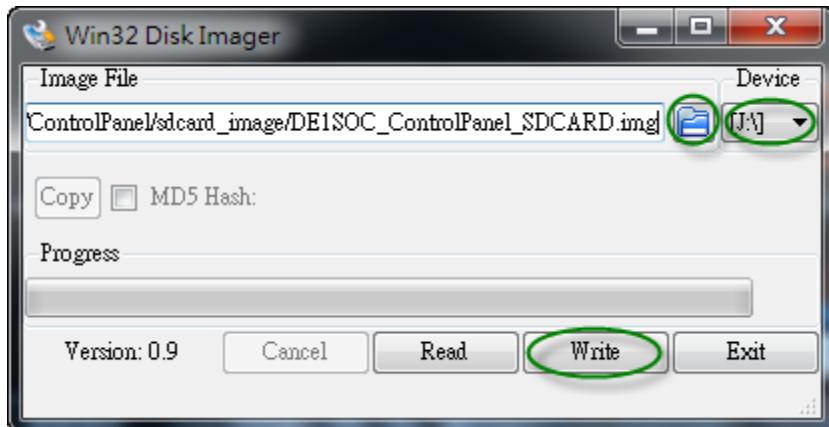


Figure 1-4 Write microSD card with a Given Image File

Before writing, a “Confirm overwrite” dialog appears as shown in **Figure 1-5**. Make sure the device is the microSD card, and click “Yes” to close the window.

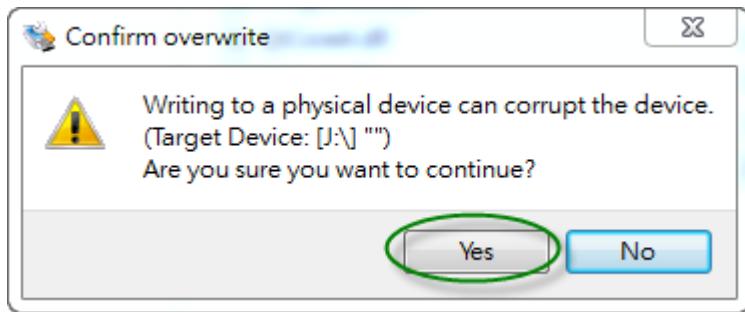


Figure 1-5 Confirm to Continue

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

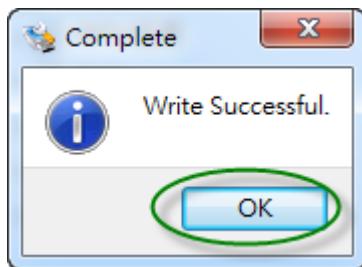
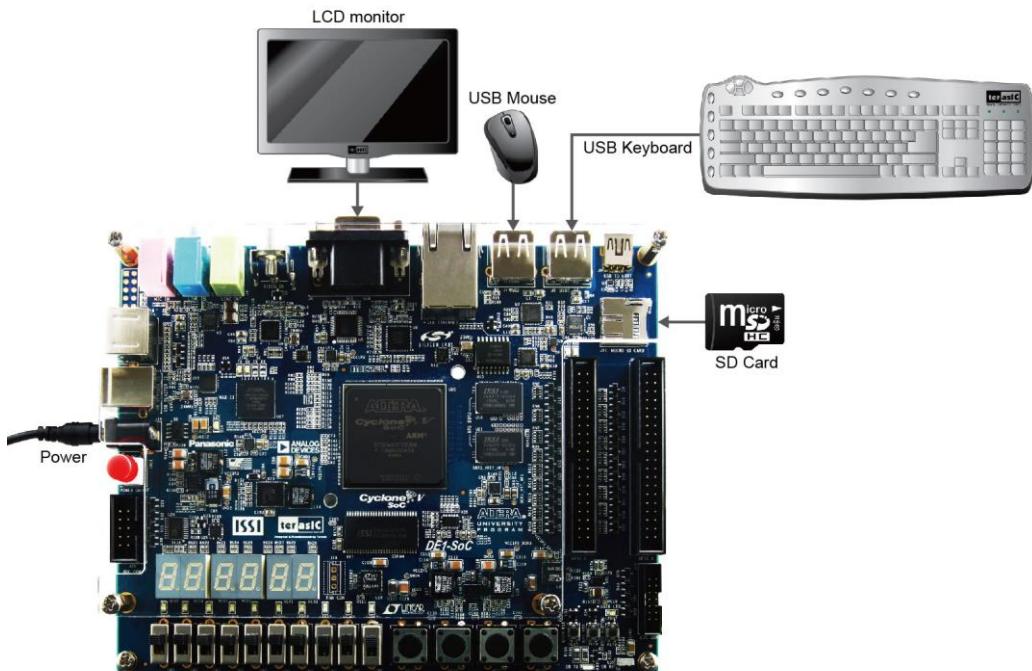


Figure 1-6 Write SD-card Complete

## ■ Set up Hardware and Boot Linux

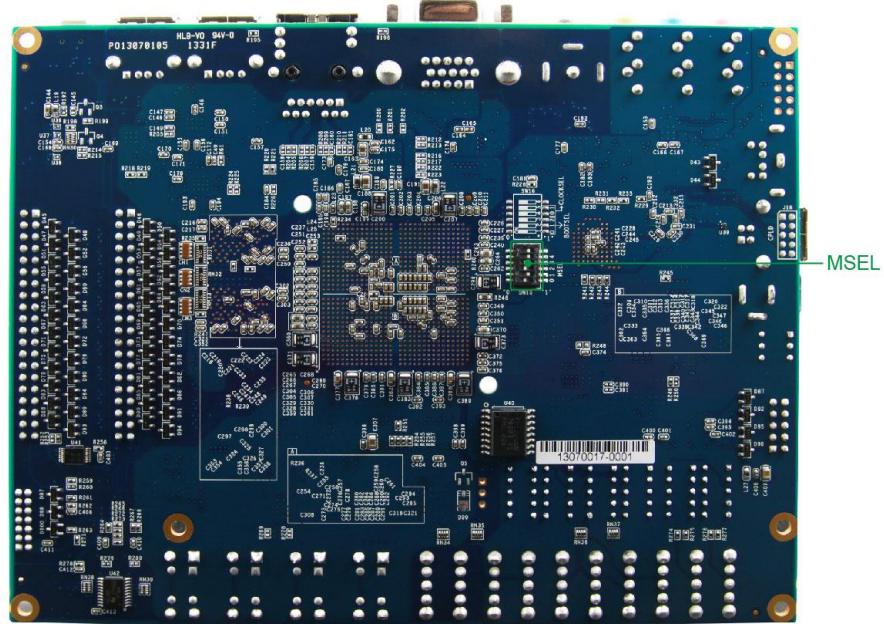
Connect the following items to the Altera DE1-SoC FPGA development board as shown in [Figure 1-7](#).

- USB mouse and keyboard
- LCD monitor
- Control Panel microSD Card
- DC Power



**Figure 1-7 Altera DE1-SoC FPGA Development Board Setup**

Please also make sure that MSEL[4:0]=00000 located on the back of the Altera DE1-SoC FPGA development board as shown in **Figure 1-8**.



**Figure 1-8 MSEL[4:0] on the Altera DE1-SoC FPGA Development Board**

After connecting the above peripherals, power on the Altera DE1-SoC FPGA board. The LCD monitor will now display Linux booting message as shown in **Figure 1-9**.

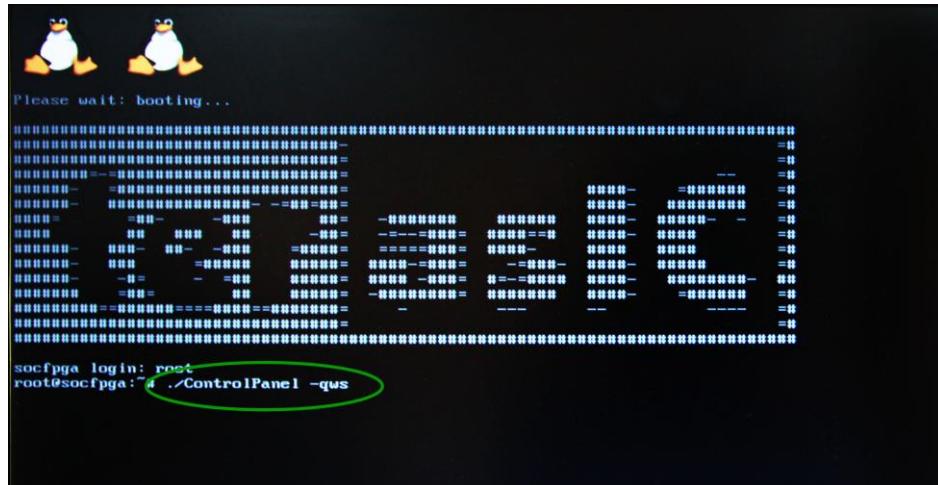


**Figure 1-9 Linux Booting Using Root Account**

When the Linux boot process is complete, a login prompt will appear as shown in **Figure 1-10**. Type in “root” and press ENTER to login Linux.

## ■ Launch Control Panel

After successful login as root, the current directory is now set at “/home/root”. The Control Panel execution file “ControlPanel” is pre-built in this directory. Please type in “./ControlPanel –qws” to launch the Control Panel as shown in **Figure 1-11**. Note that QT library has been preinstalled in the microSD card image therefore users can simply launch the Control Panel without installing the QT library.



**Figure 1-10 Launch the Control Panel**

The Control Panel window will appear on the LCD monitor as shown **Figure 1-11**. Now, you can use the USB mouse to operate the Control Panel.



**Figure 1-11 Screenshot of DE1-SoC Control Panel**

## Chapter 2

# *Linux Installation*

The Control Panel program runs on Linux in the Altera SOC ARM as shown in the block diagram in Chapter 1. In Chapter 2, we begin by installing Linux as a first step toward completing the Control Panel project. The Linux system can be installed on a x86 PC directly or on a virtual machine running on Microsoft Windows. This chapter also 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.

The Control Panel is a GUI-based program and QT Creator is a convenient tool to help create such GUI-based programs. 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.

Running Ubuntu on VMware Player in Microsoft Windows is an easier way when users wish to create a GUI application, such as oscilloscopes or logic analyzers, without the need to have a Desktop Linux. Users can simply follow our GUI application development method explained in this tutorial for more user-based GUI applications.

## 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**.

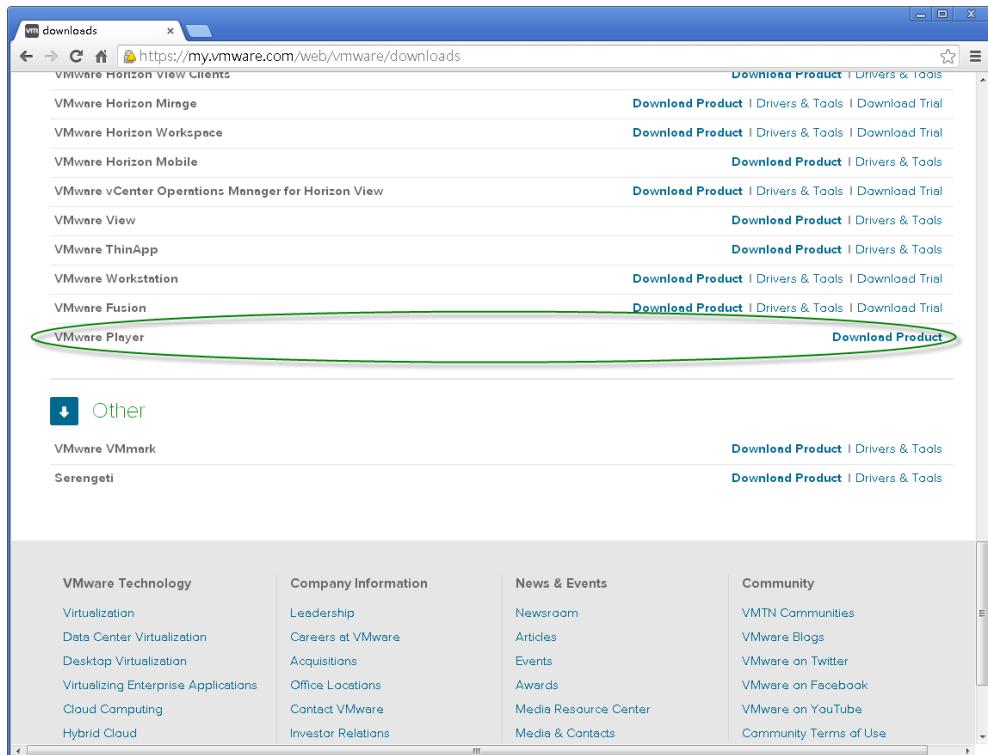
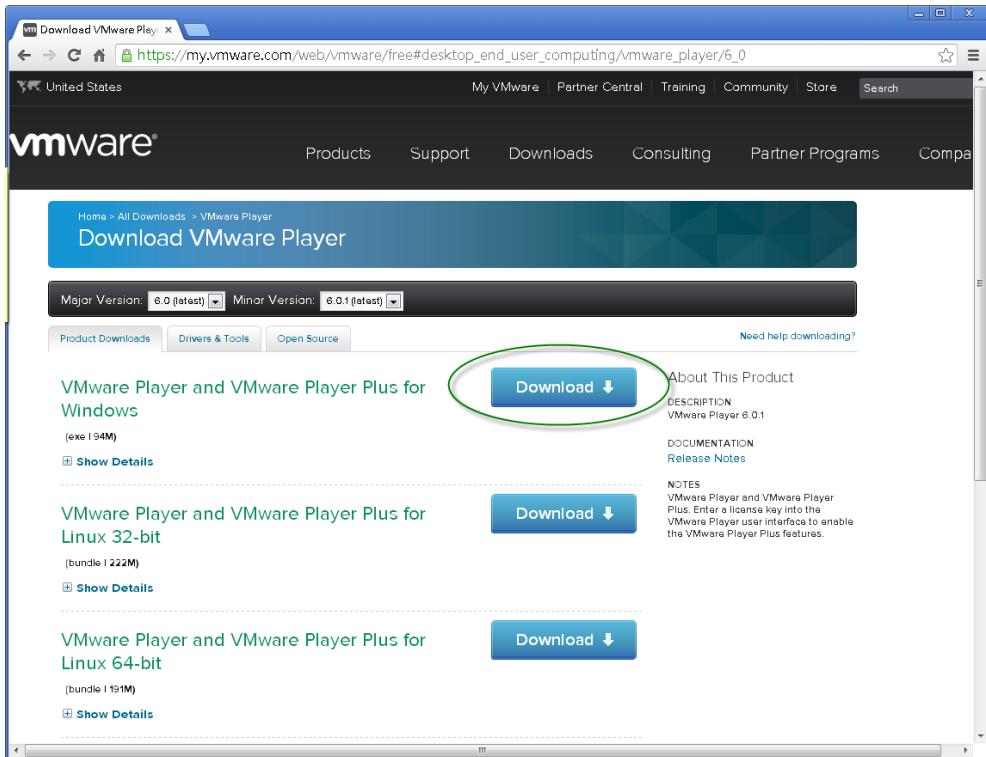


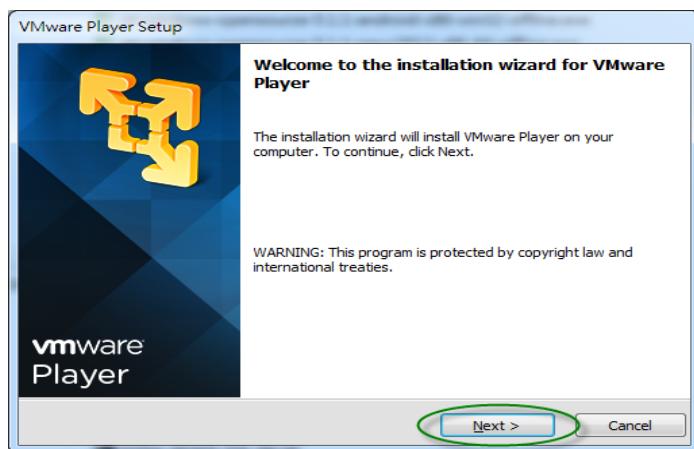
Figure 2-1 Download Web Page of VMware



**Figure 2-2 VMware Player Download Web Page**

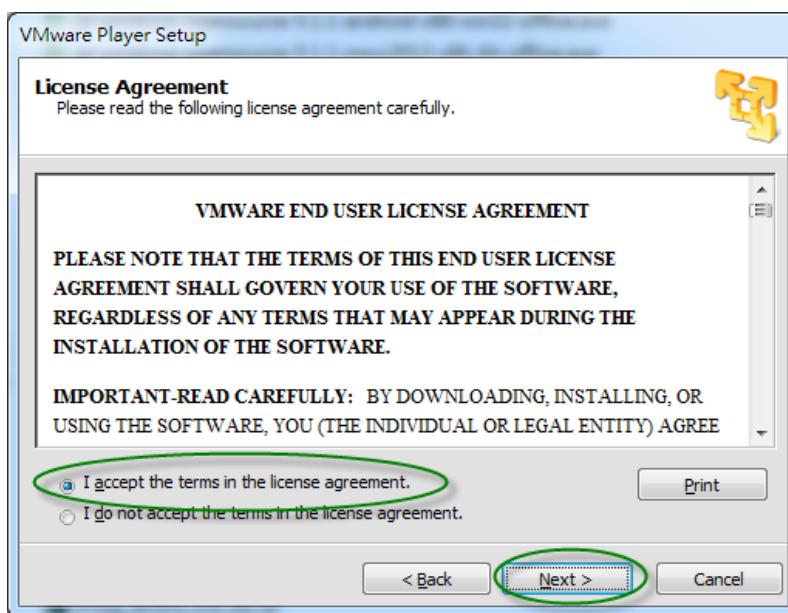
## ■ Install VMware

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.



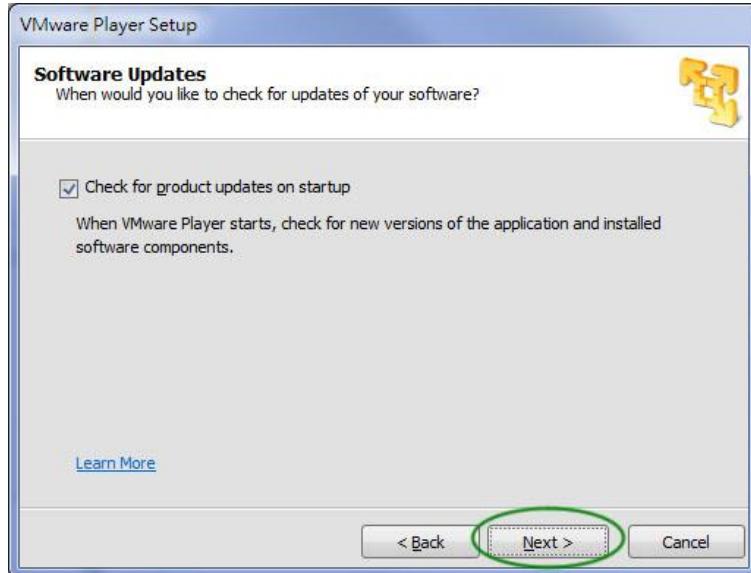
**Figure 2-4 License Agreement Dialog**

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.



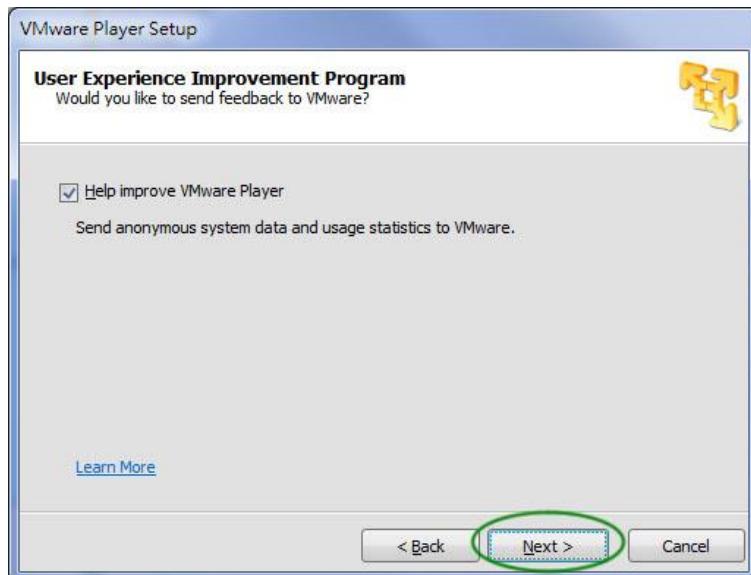
**Figure 2-5 Destination Folder Dialog**

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.



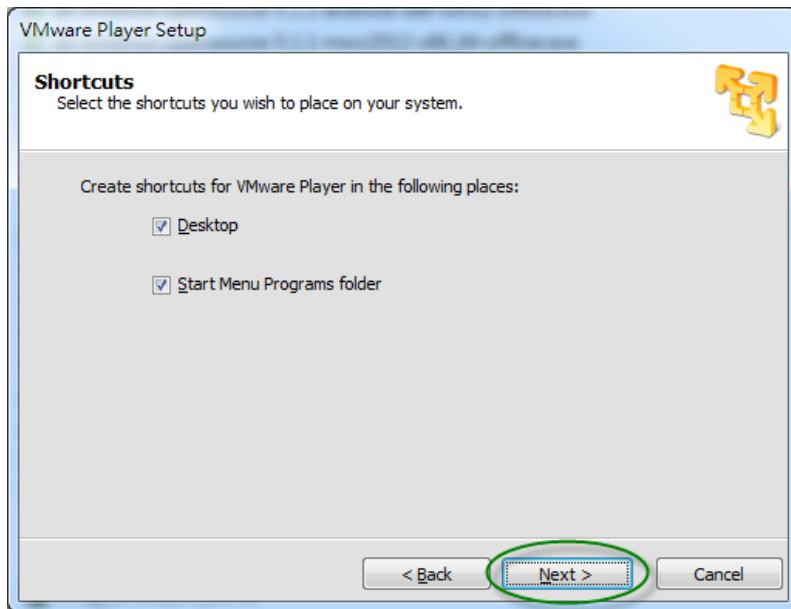
**Figure 2-6 Software Updates Dialog**

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.



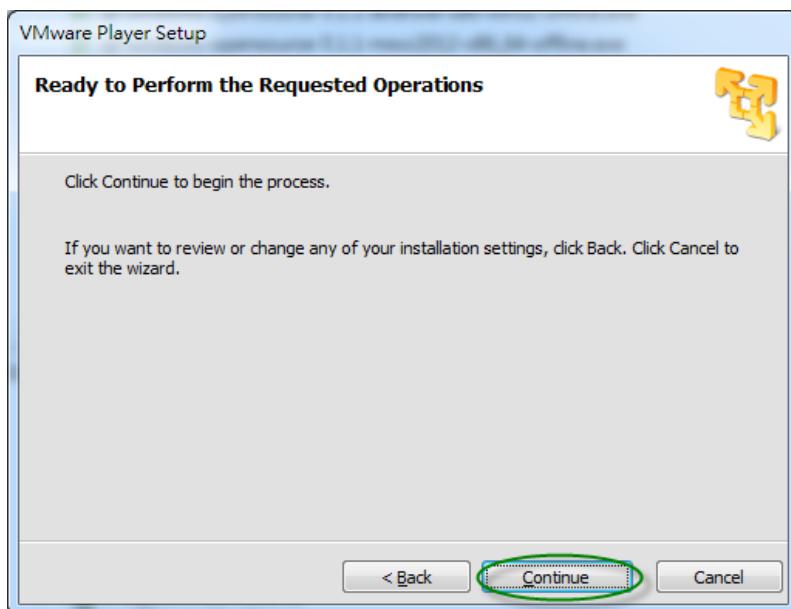
**Figure 2-7**

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.

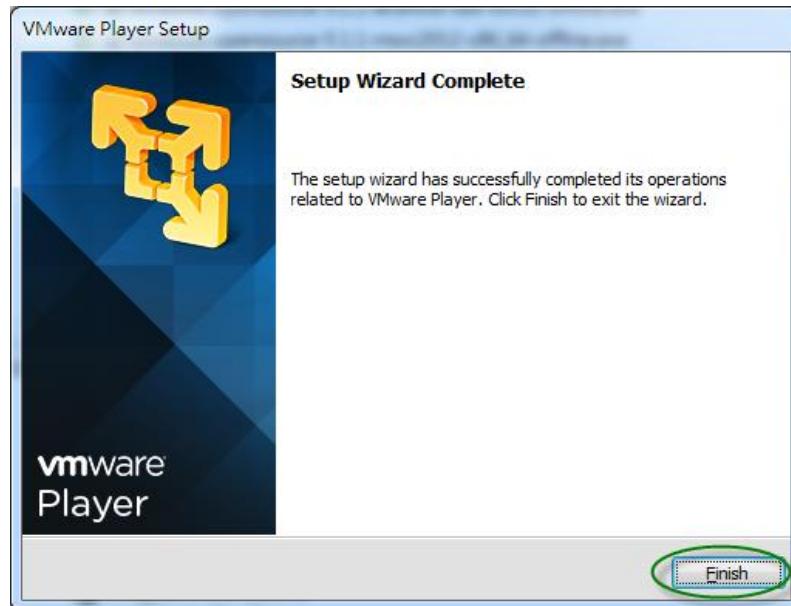


Figure 2-10 Setup Wizard Complete Dialog

## 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

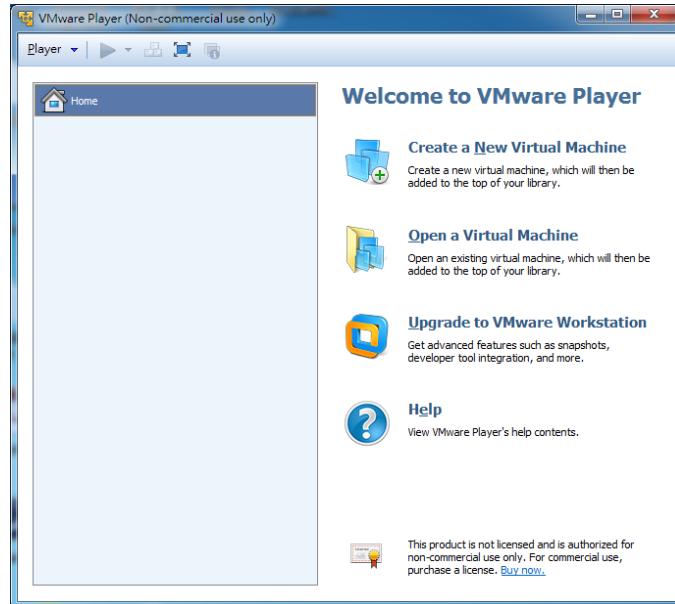


Figure 2-12 Main Window of VMware Player

## 2.4 Install Linux Ubuntu Desktop

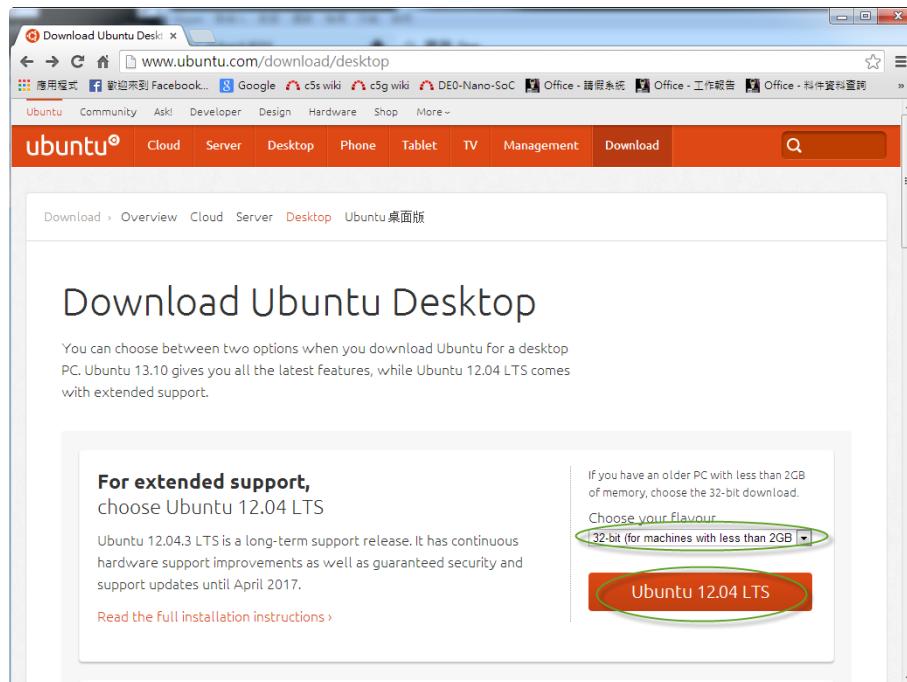
Now we have downloaded and installed the VMware player, we are ready to download Ubuntu image file and install it on the VMware Player. Note that the “Easy Install” mode is automatically applied when VMware Player detects if the installed OS is Ubuntu 12.04.

### ■ Download Ubuntu Linux Image

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

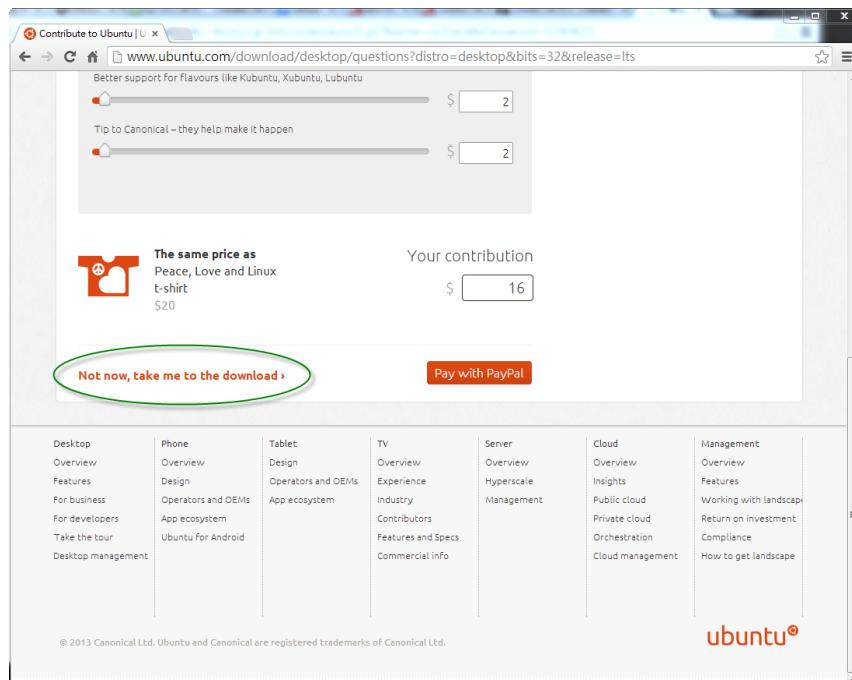
<http://www.ubuntu.com/download/desktop>

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.

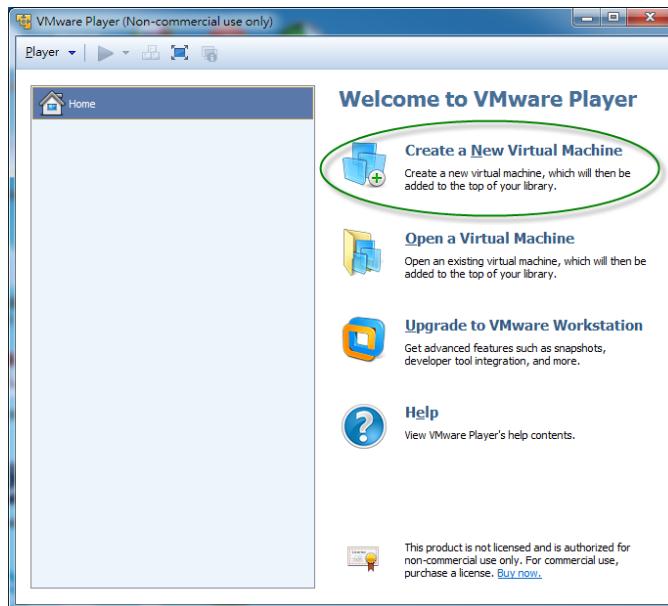
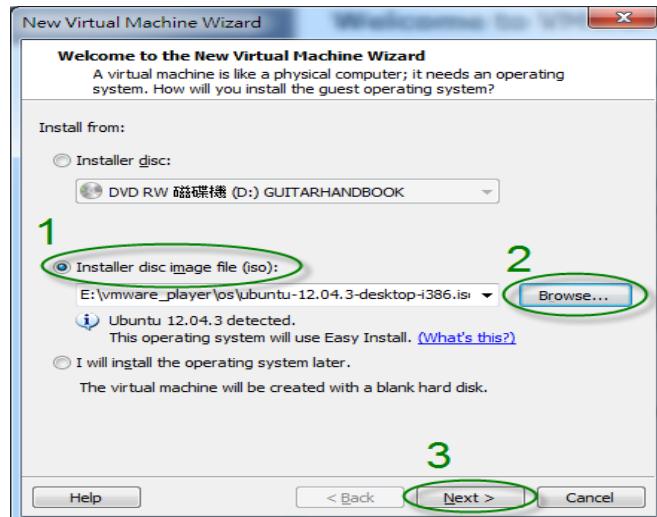


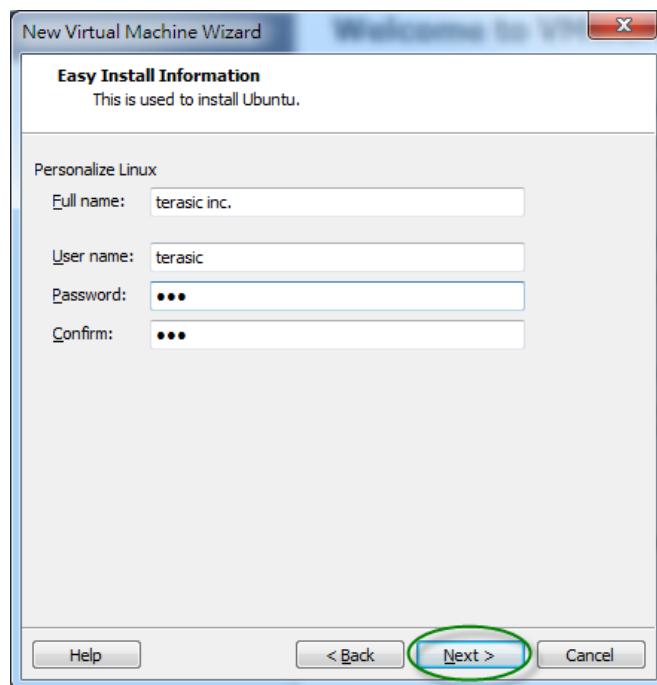
Figure 2-15 VMware Player Main Window

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**

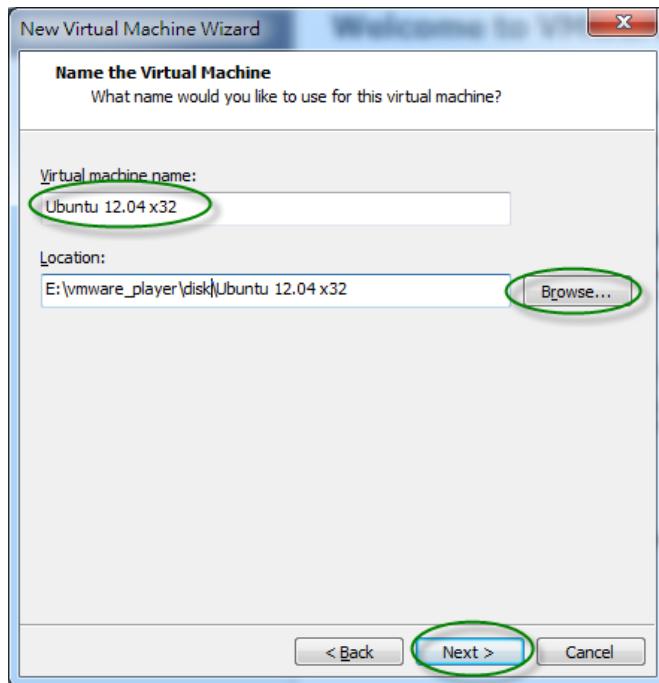
When the **Easy Install Information** dialog appears as shown in **Figure 2-17**, please specify your username and password. In this tutorial, the user name “terasic” and the password “123” are used. Click "Next >" button to go to the next step. **Please remember your username and password, as they will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

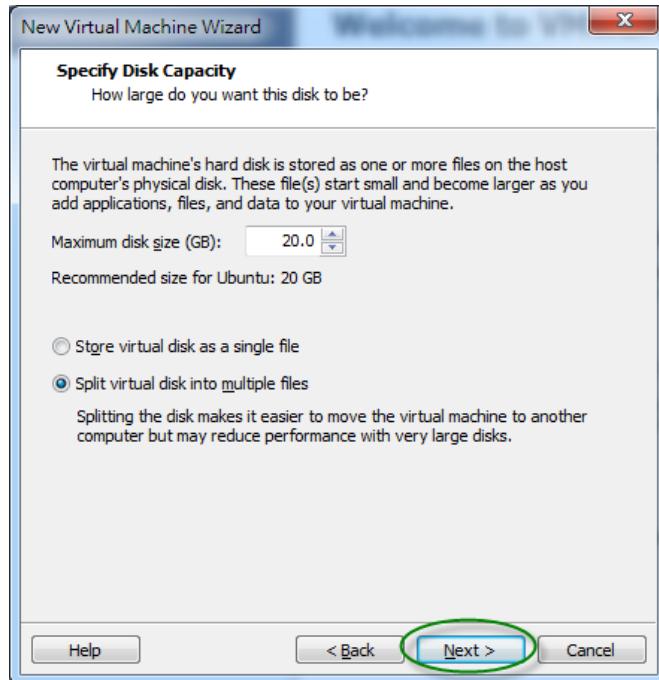


Figure 2-19 Specify Disk Capacity Dialog

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.

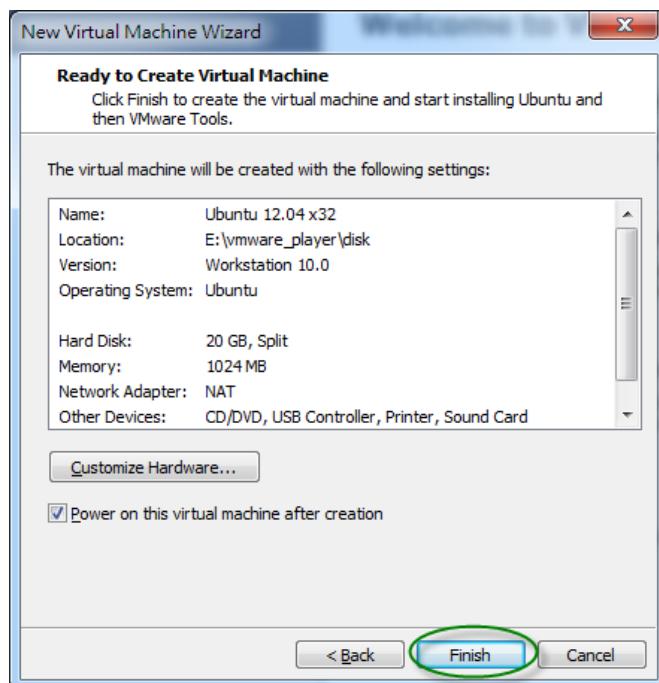
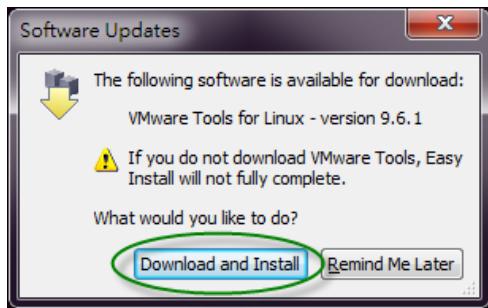


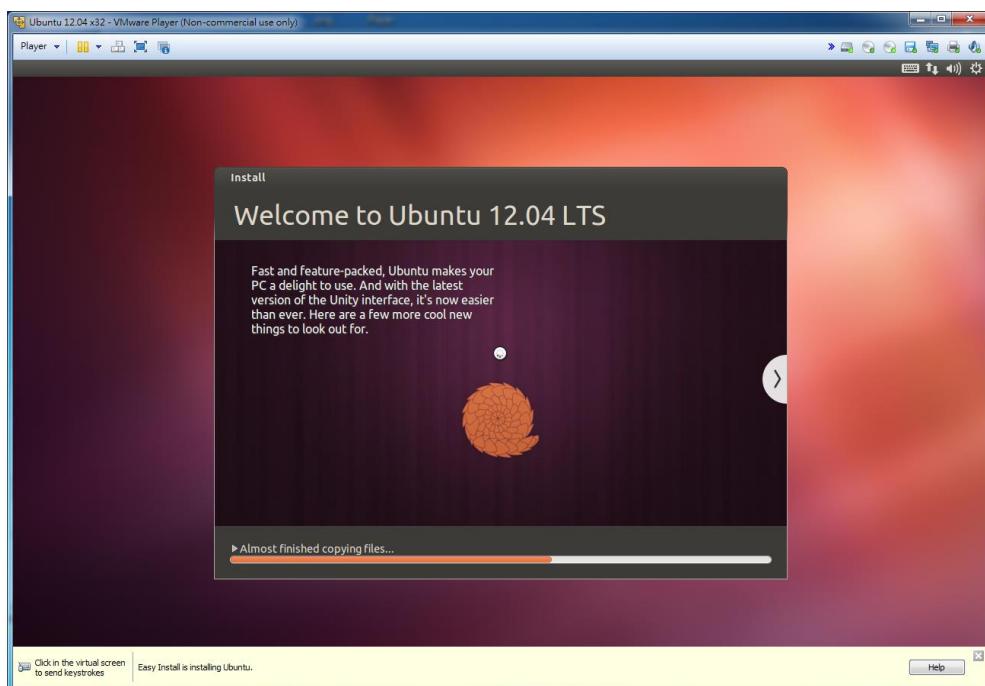
Figure 2-20 Ready to Create Virtual Machine Dialog

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



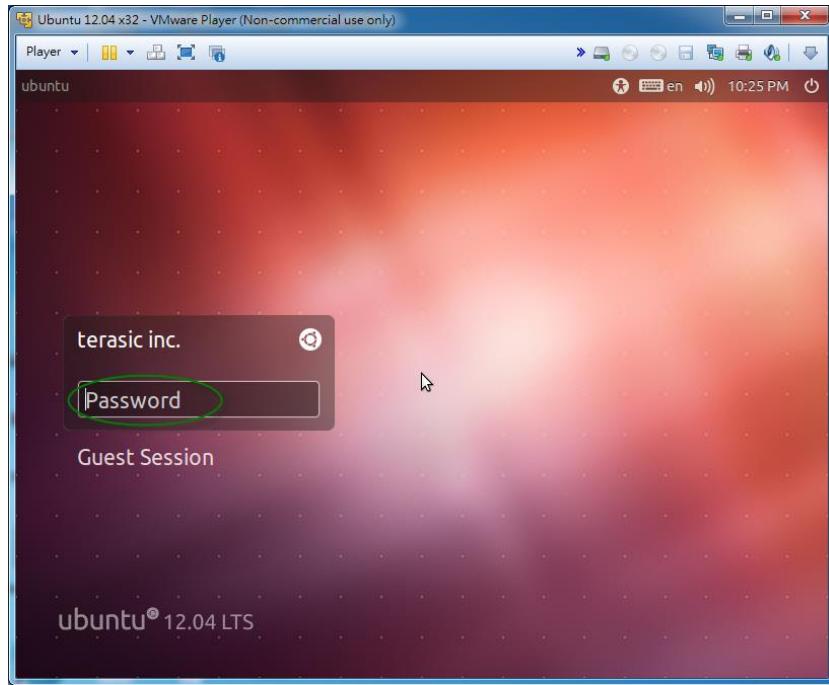
**Figure 2-21 Software Updates Dialog**

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



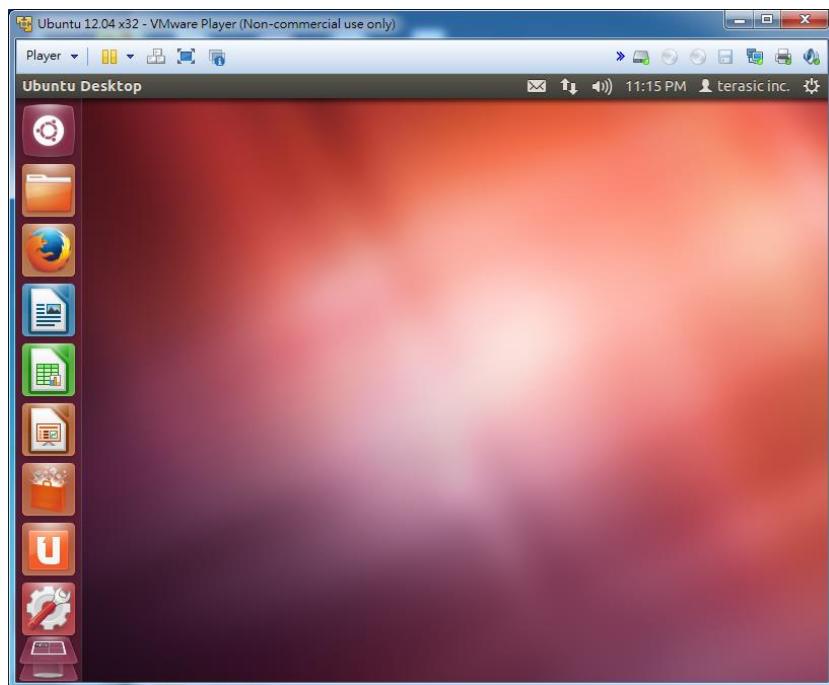
**Figure 2-22 Installation Progress**

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.

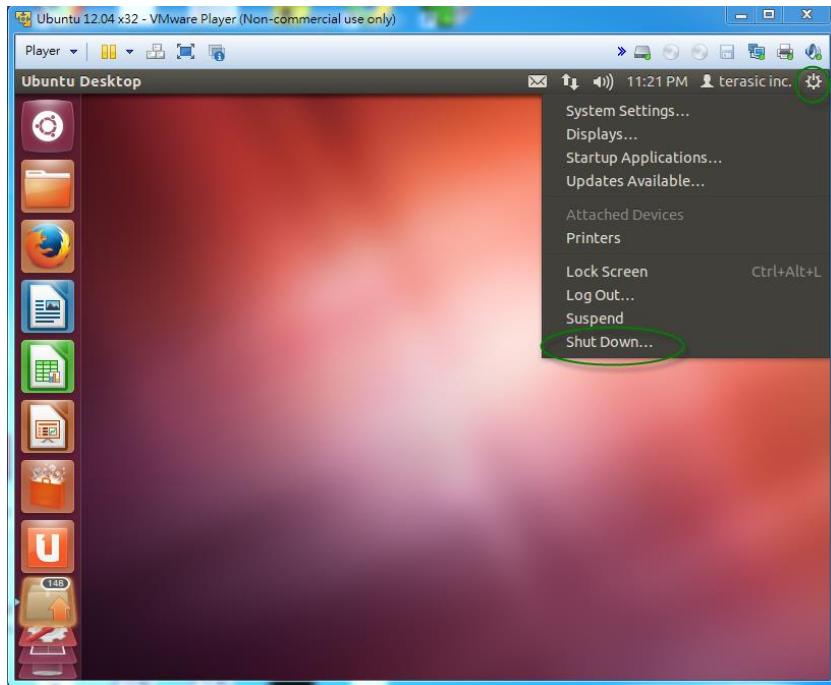


Figure 2-25 Shut Down Ubuntu Desktop

## ■ 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**.

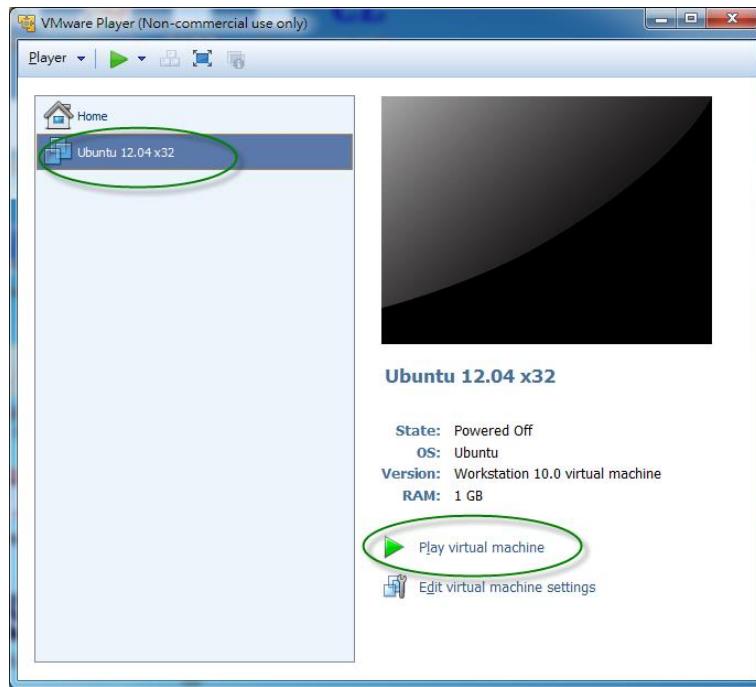
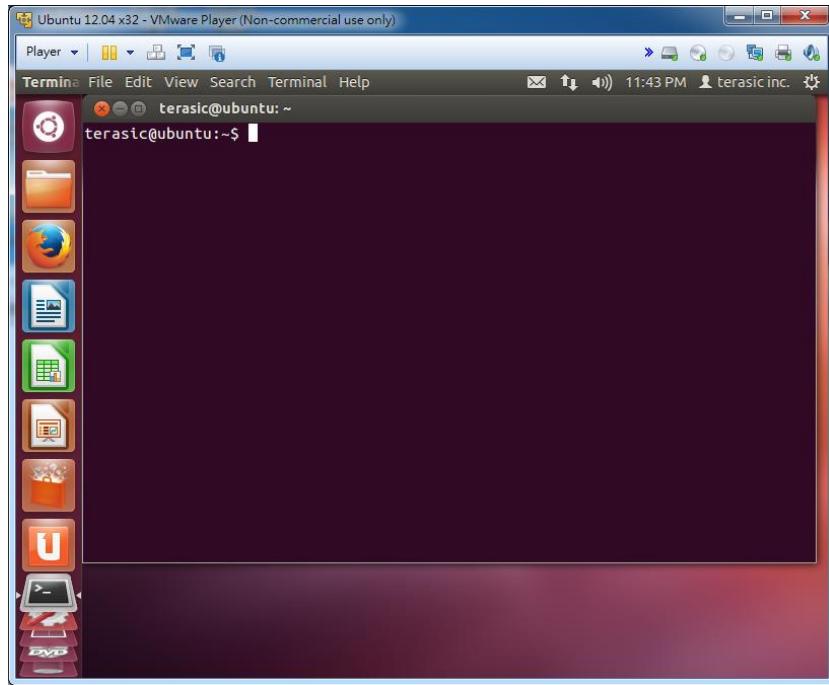


Figure 2-26 Start Ubuntu

## 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**.

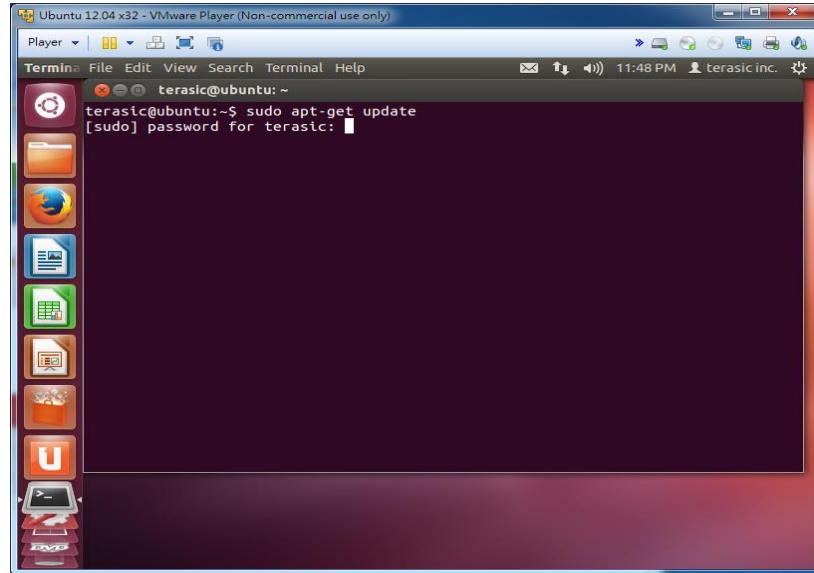


**Figure 2-27 Linux Terminal**

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

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

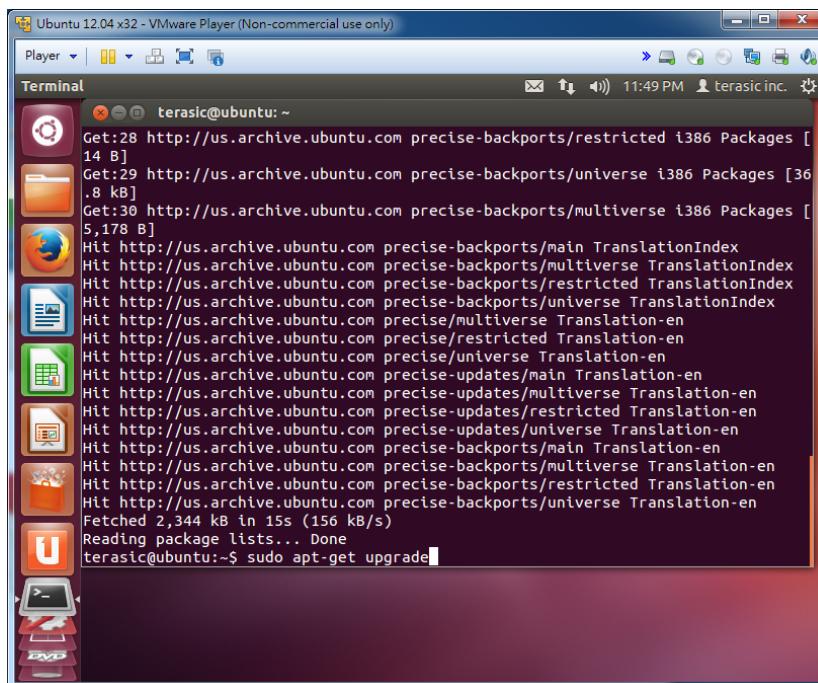
The system will prompt users to input a password. Please input your password (in this tutorial, the password is “123”) and press ENTER as shown in [Figure 2-28](#).



**Figure 2-28 Input Password for Terasic**

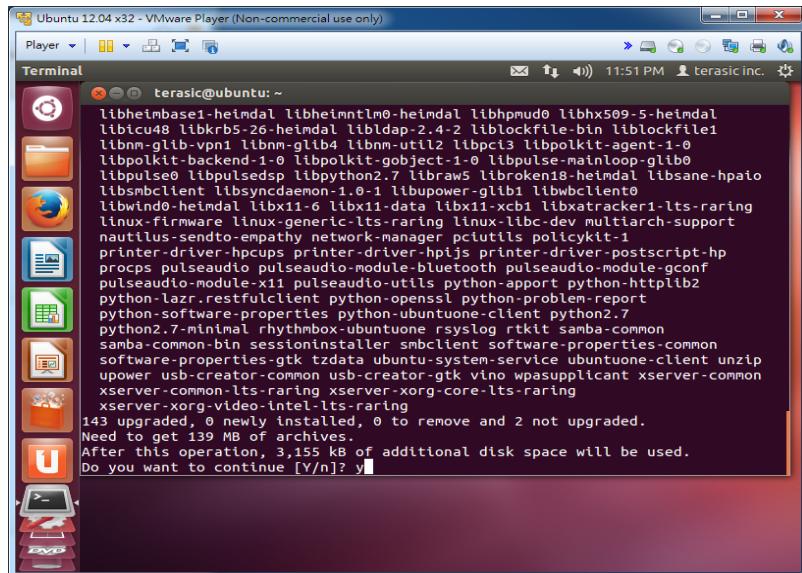
After finishing typing in “sudo apt-get update” and the password, now type in the following command and press ENTER, as shown in **Figure 2-29**.

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



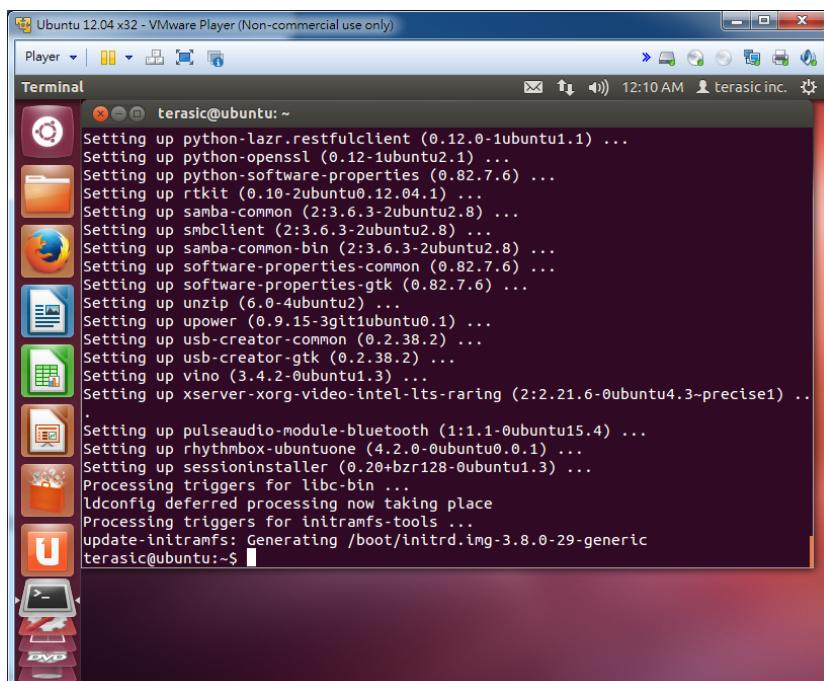
**Figure 2-29 Typing in the Command “sudo apt-get upgrade”**

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.



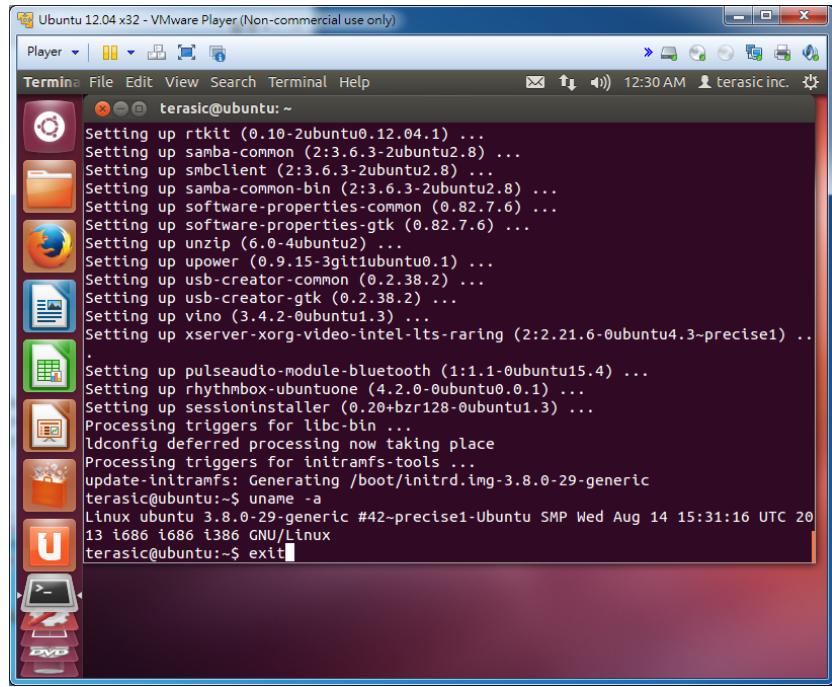
**Figure 2-30 Type 'y' to Continue the Upgrade Process**

**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**

## Chapter 3

# *QT Creator Installation*

The Control Panel is a GUI-based program and 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.

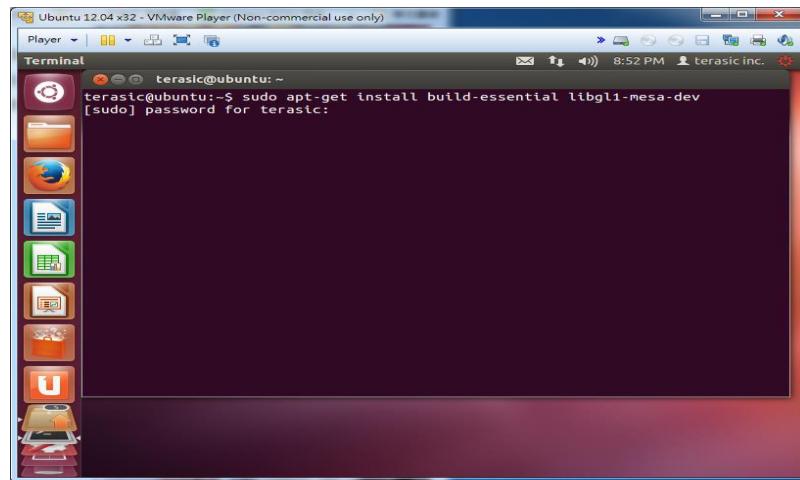
Chapter 3 illustrates how to install QT Creator on the Desktop Ubuntu which has been installed in previous chapter. Also, we will show how to create, compile, and build a hello program running on Ubuntu. In order for QT Creator to be able to build a project correctly, we also provide steps of installing necessary x86 GCC tool-chain needed for the QT Creator.

### **3.1 Install Tool-Chain for Linux x86**

In order for the QT Creator to be able to build a project correctly, a proper tool-chain is required. Use the following command to install the required tool-chain:

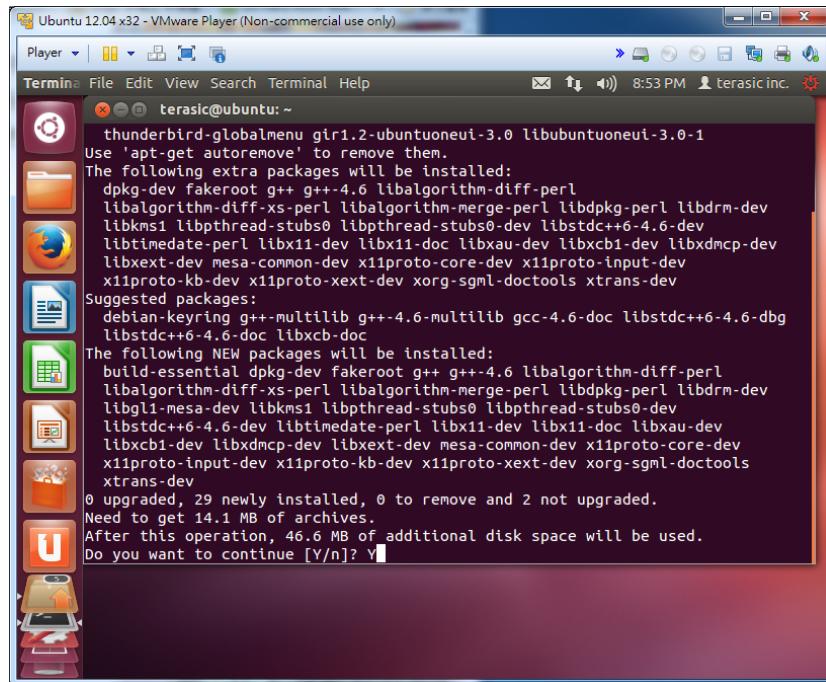
```
$sudo apt-get install build-essential libgl1-mesa-dev
```

System will prompt the user to input a password (in this tutorial, the password is “123”) as shown in **Figure 3-1**.



**Figure 3-1 Install Software Package**

Now system also prompts users to confirm to continue the installation process, as shown in **Figure 3-2**. Please type “y” and press ENTER to continue.



**Figure 3-2 Confirm to Continue with the Software Package Installation**

**Figure 3-3** shows the screenshot after the installation has been completed. Type “exit” to close the terminal.

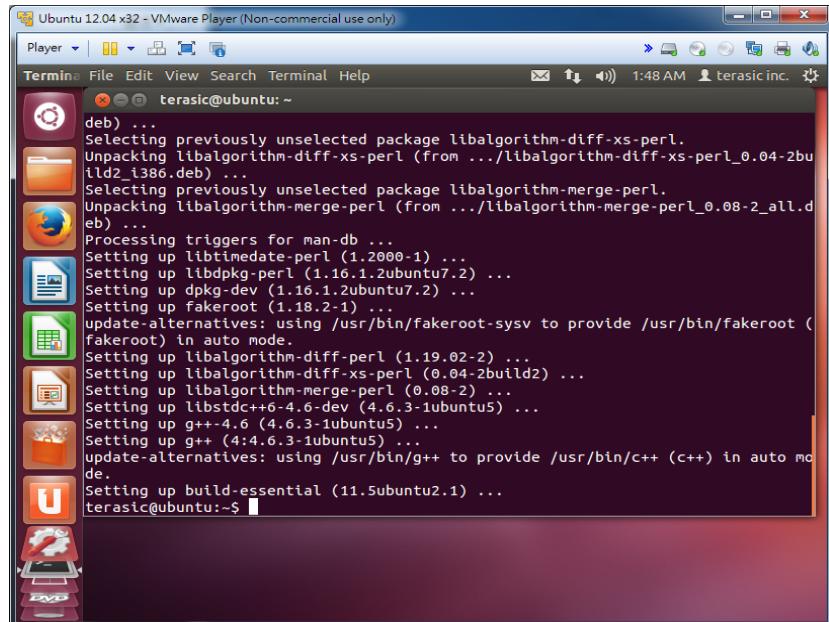


Figure 3-3 Software Package Installation is Now Complete

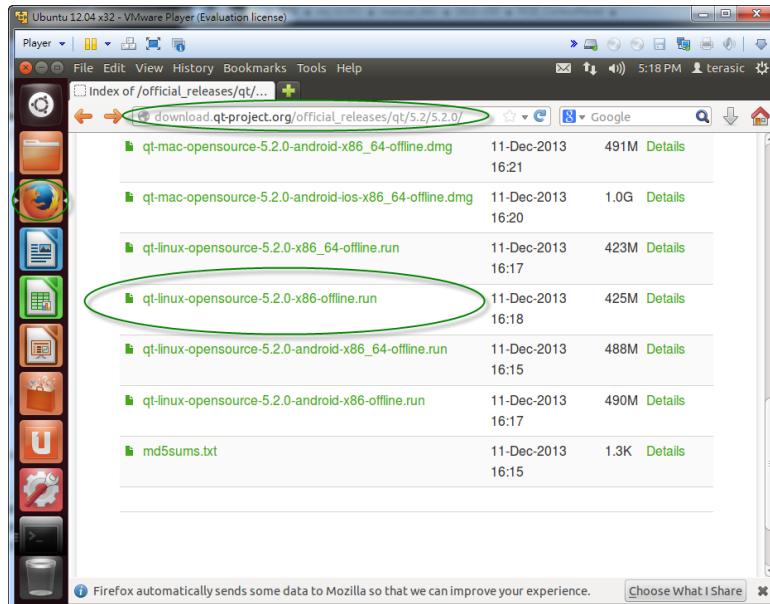
## 3.2 Download and Install QT Installer

### ■ Download QT Installer

As shown in **Figure 3-4**, click the Firefox web browser to open the web page

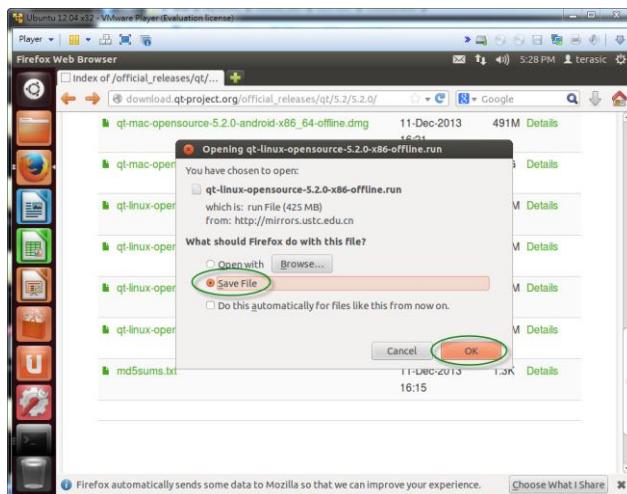
[http://download.qt-project.org/official\\_releases/qt/5.2/5.2.0/](http://download.qt-project.org/official_releases/qt/5.2/5.2.0/),

then click “qt-linux-opensource-5.2.0-x86offline.run” to download the QT installer.



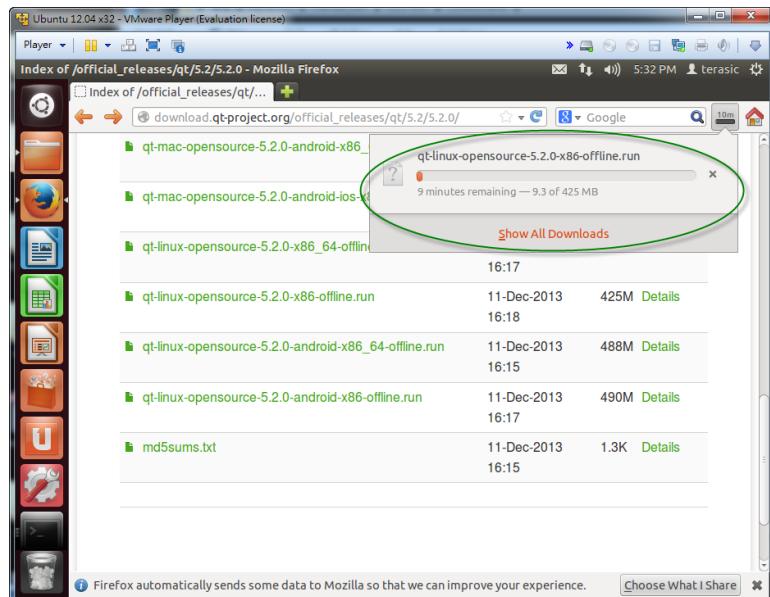
**Figure 3-4 Web Page to Download QT**

When an Opening dialog appears as shown in **Figure 3-5**, select “Save File” radio button, and click “OK”.



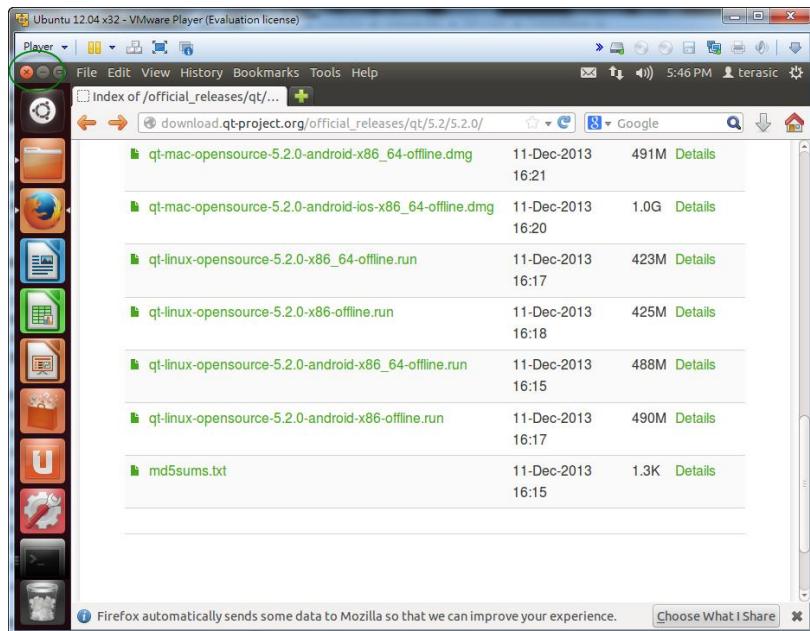
**Figure 3-5 Opening Dialog**

**Figure 3-6** shows the download progress.



**Figure 3-6 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 3-7**, to close the Firefox web browser.



**Figure 3-7 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 3-8**.

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

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

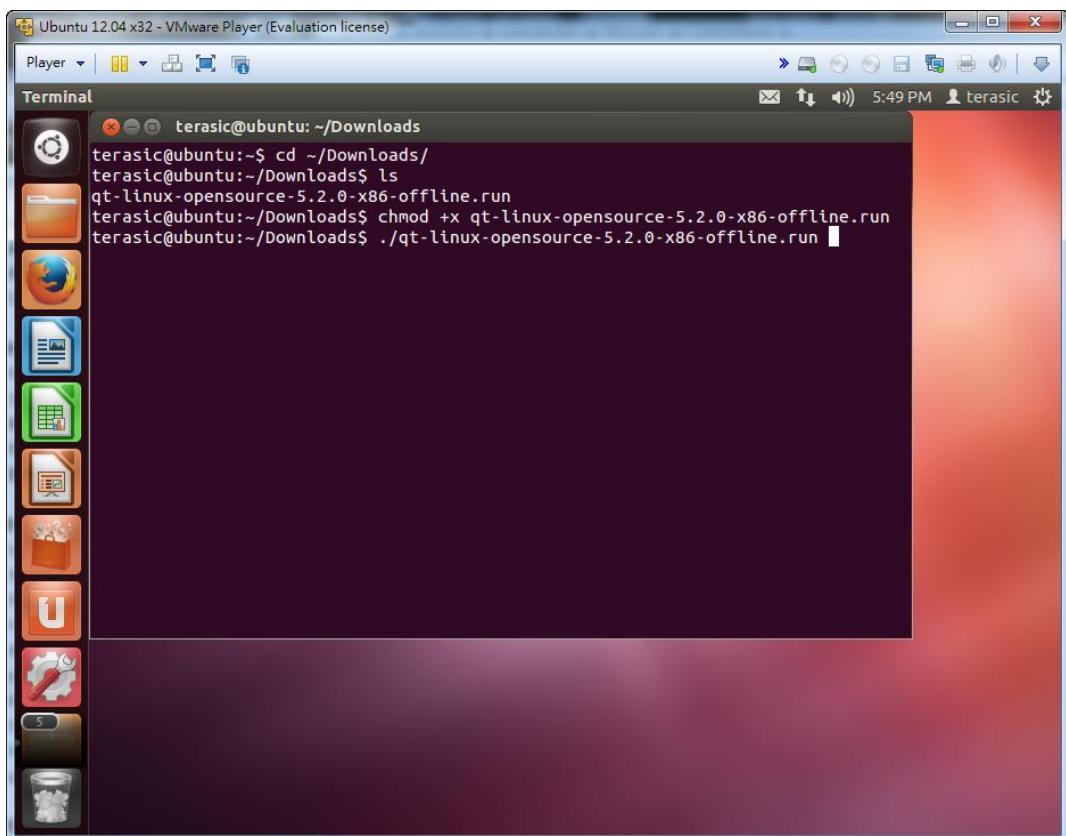
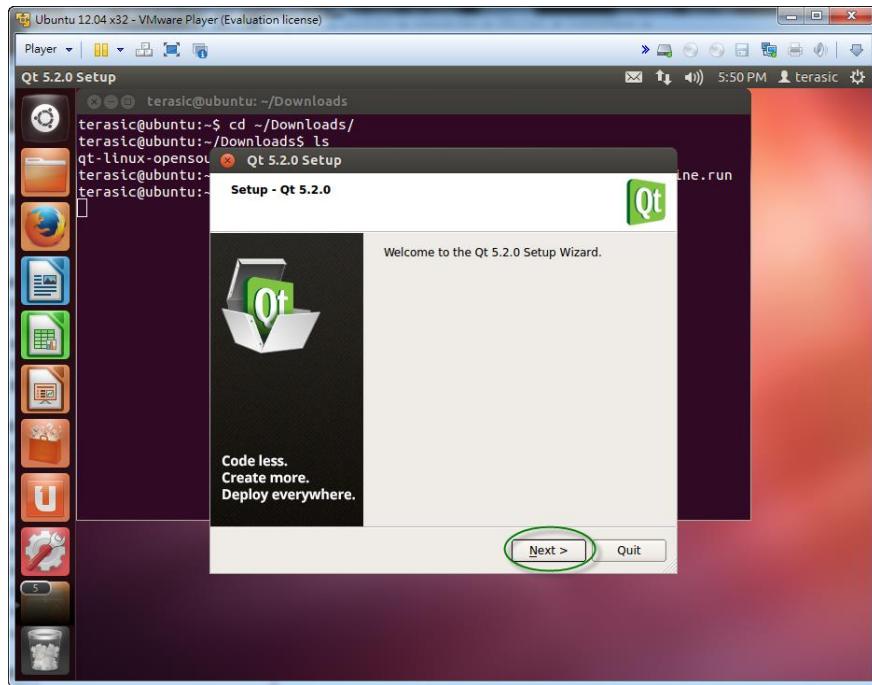


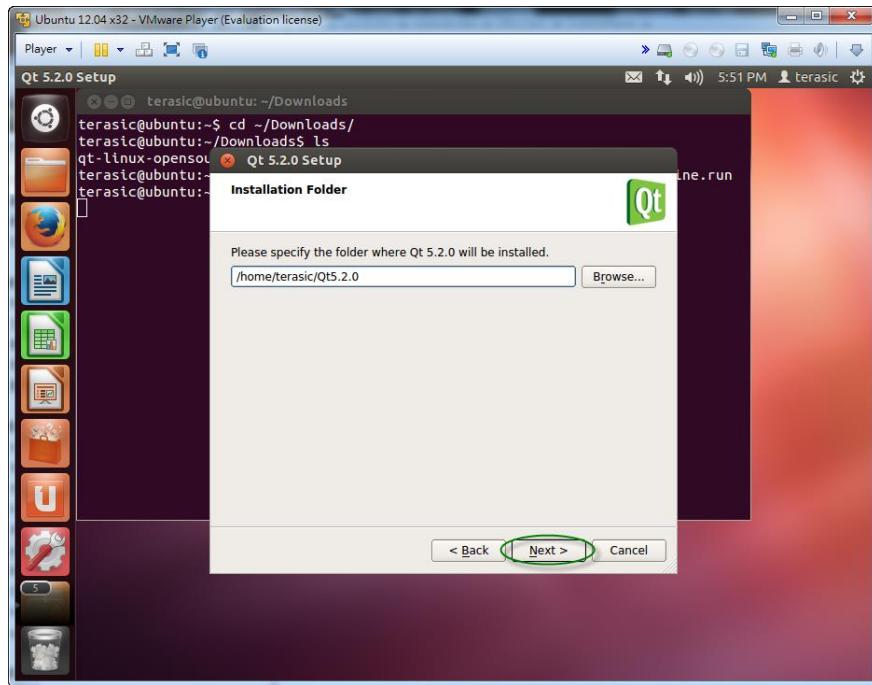
Figure 3-8 Locate and Launch the QT Installer

**Figure 3-9** shows the **Welcome** dialog of the QT Installer.



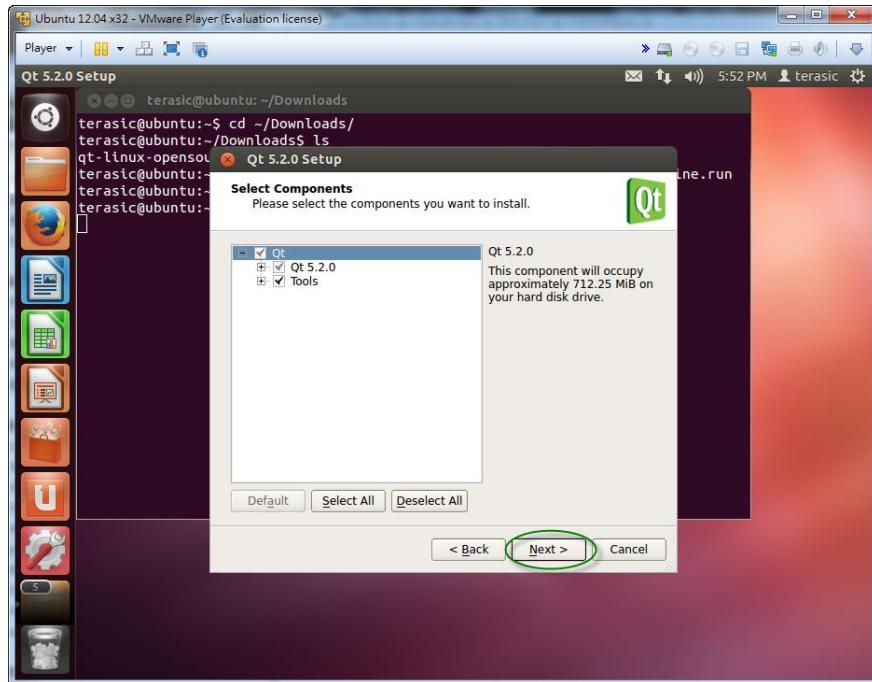
**Figure 3-9 Welcome Dialog**

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 3-10**.



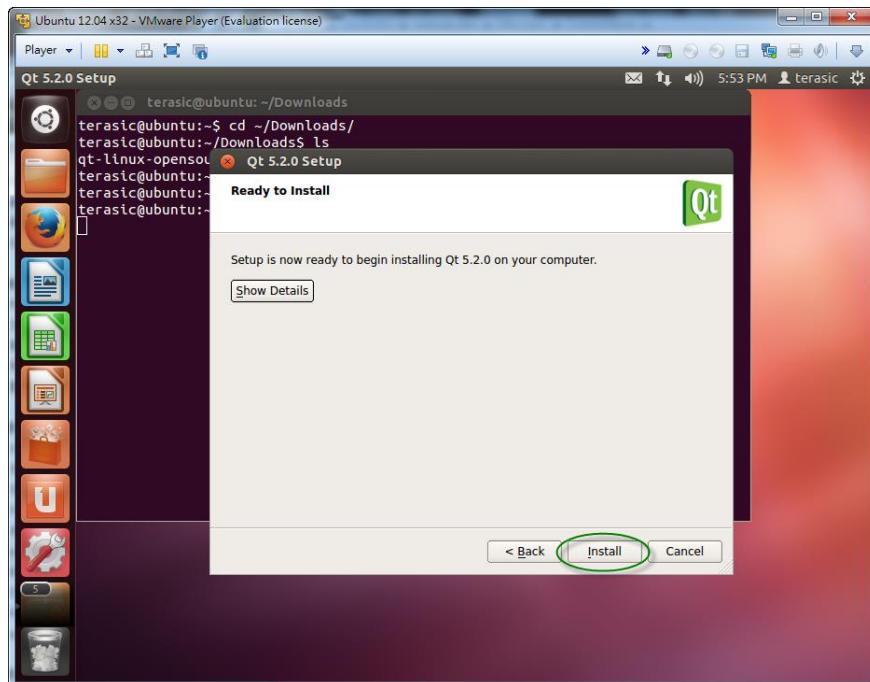
**Figure 3-10 Specify the Folder to Install QT**

In the **Select Components** dialog, keep default settings and click “Next >” to go to the next step, as shown in **Figure 3-11**



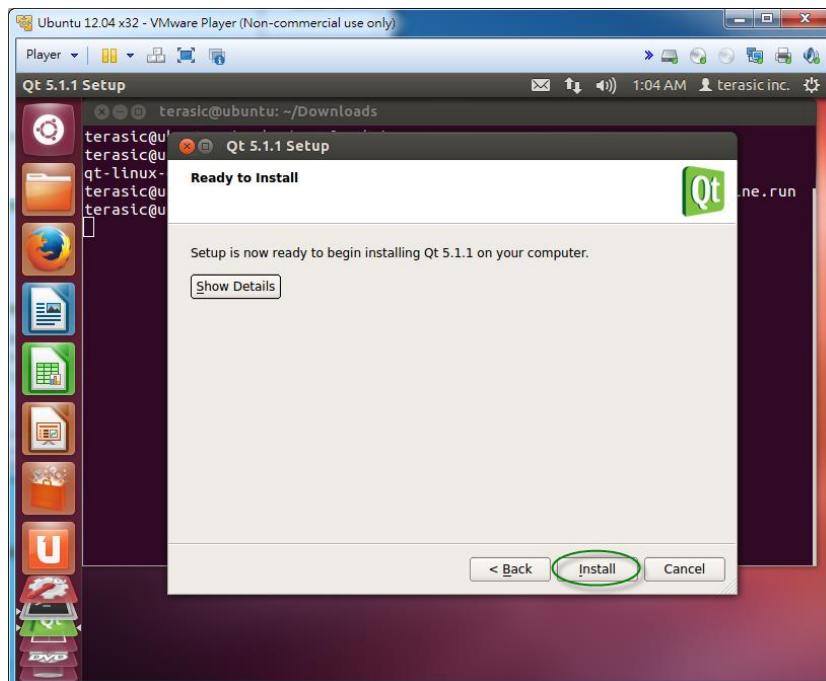
**Figure 3-11 Select Components Dialog**

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 3-12**



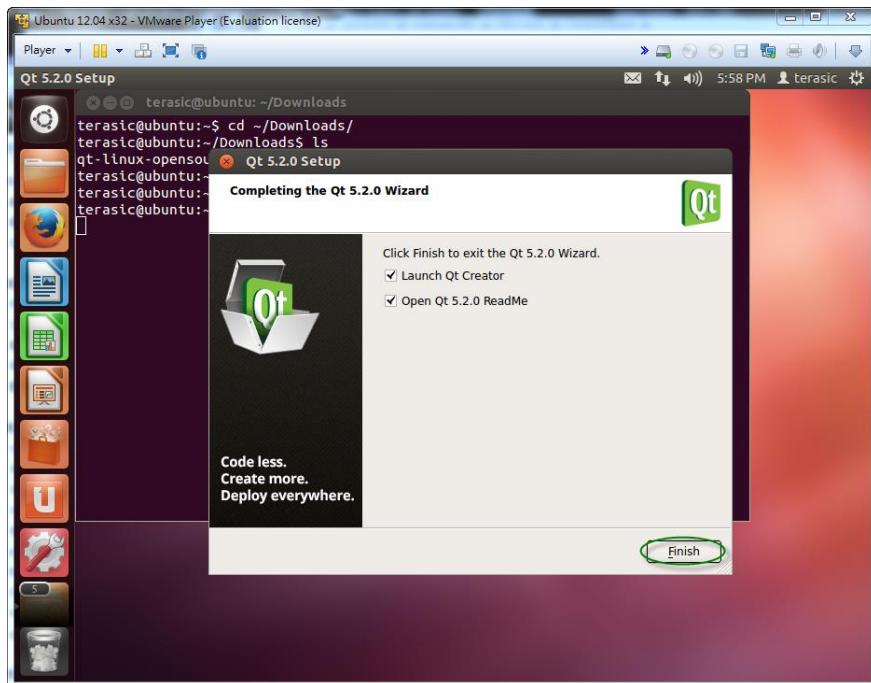
**Figure 3-12 License agreement dialog of the QT installer**

In the **Ready to Install** dialog, as shown in **Figure 3-13**, click “Install” to go to the next step.



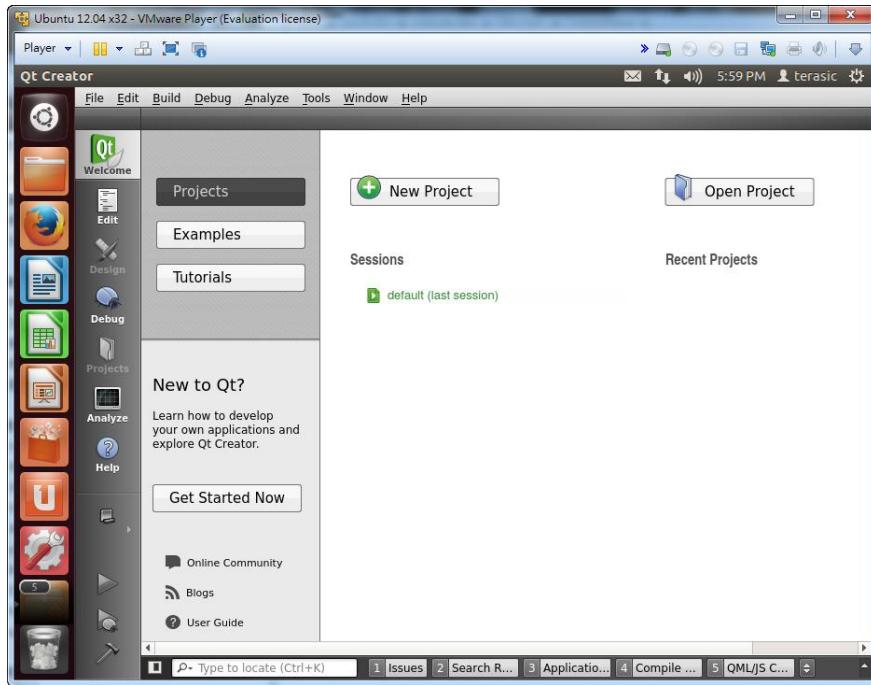
**Figure 3-13 Ready to Install Dialog of QT Installer**

In the **Completing the Qt 5.2.0 Wizard** dialog, as shown in **Figure 3-14**, click “Finish” to close the window.



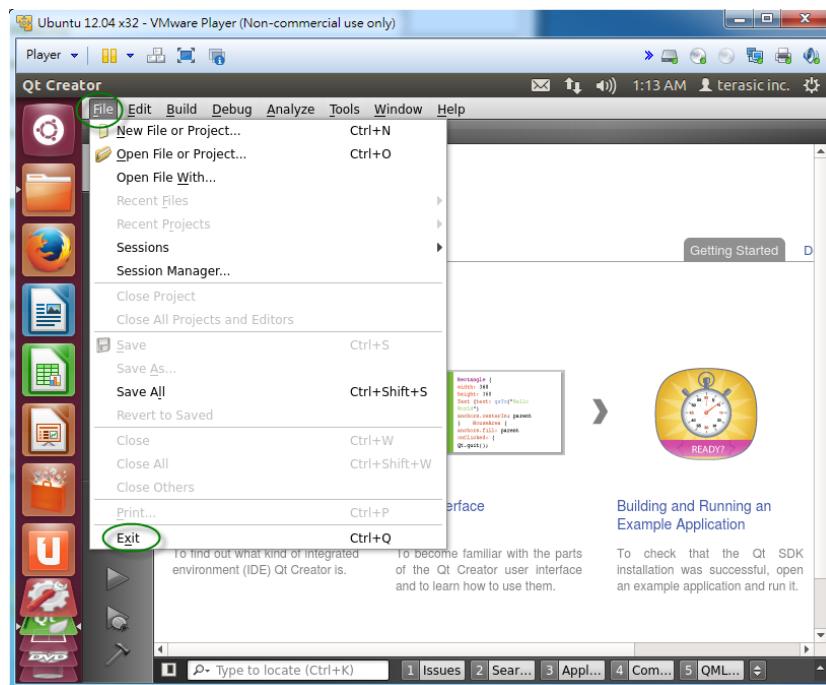
**Figure 3-14 Completing Dialog of QT Installer**

After installation has been completed, QT Creator is automatically launched as shown in **Figure 3-15**.



**Figure 3-15 QT Creator GUI**

Select the menu item “File→Exit”, as shown in **Figure 3-16**, to close the QT Creator.



**Figure 3-16 Exit QT Creator**

### 3.3 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 3-17**. To launch QT creator, simply double click “qtcreator” program icon.

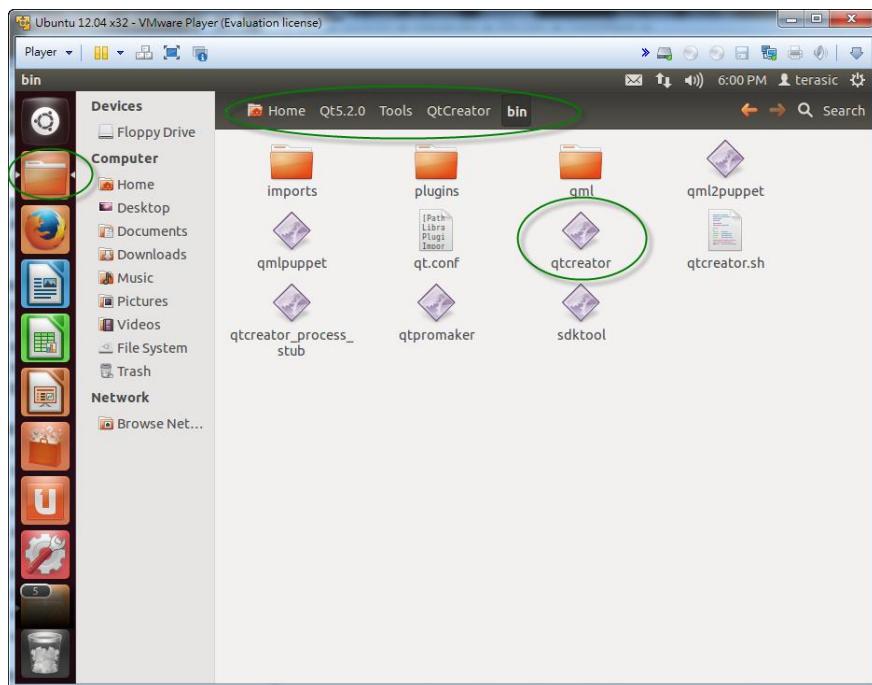
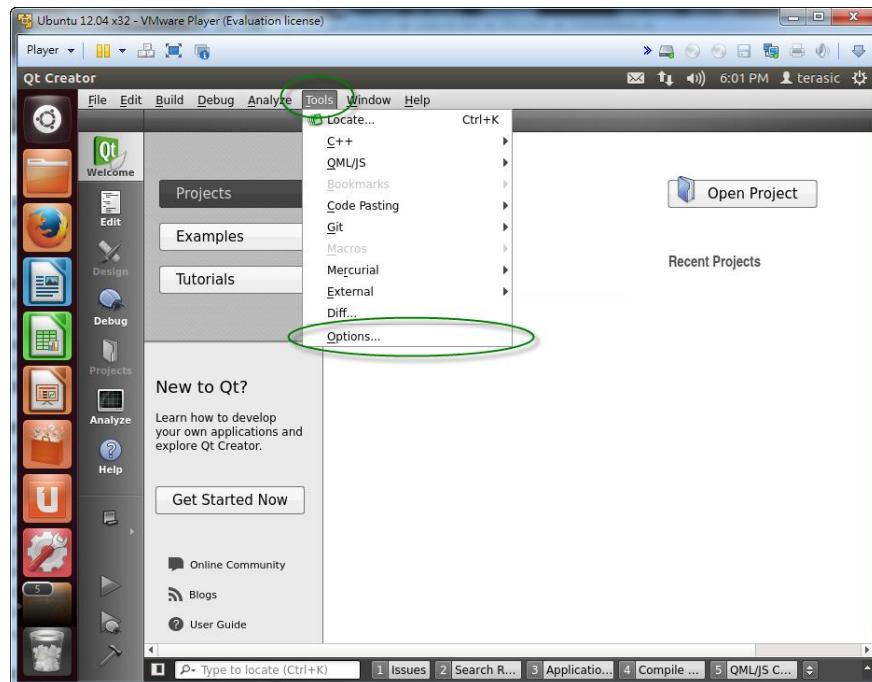


Figure 3-17 Launch QT Creator

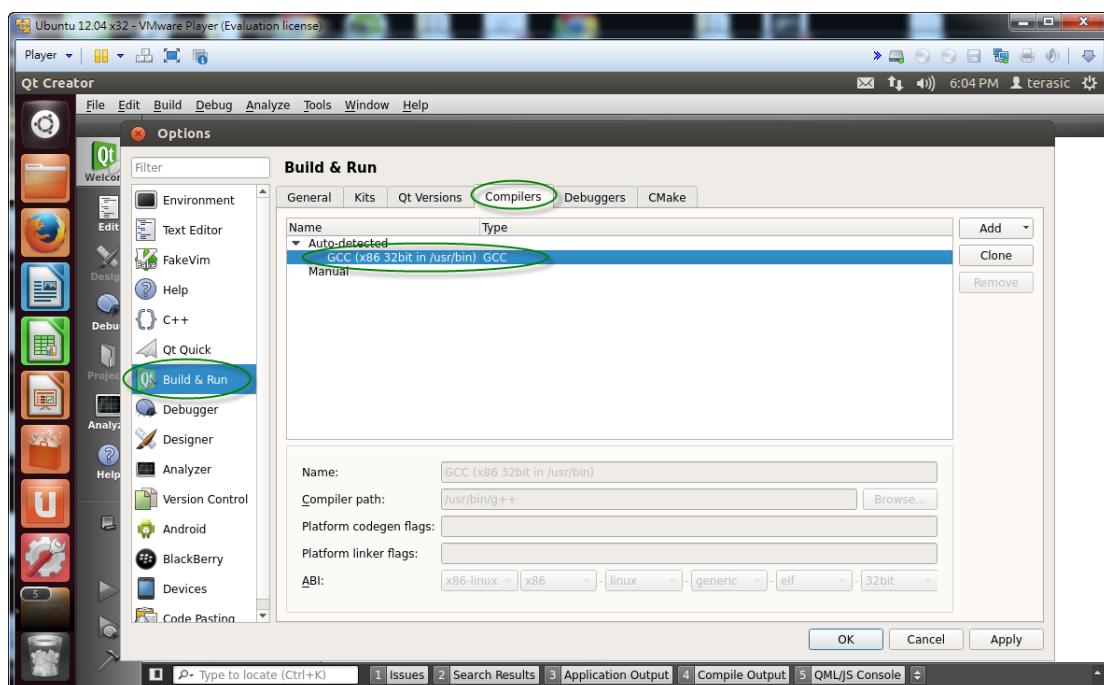
#### ■ Check Build & Run Configuration Settings

When QT Creator is launched, browse the menu and click “Tools→Options...”, as shown in **Figure 3-18**, to open the Option dialog.



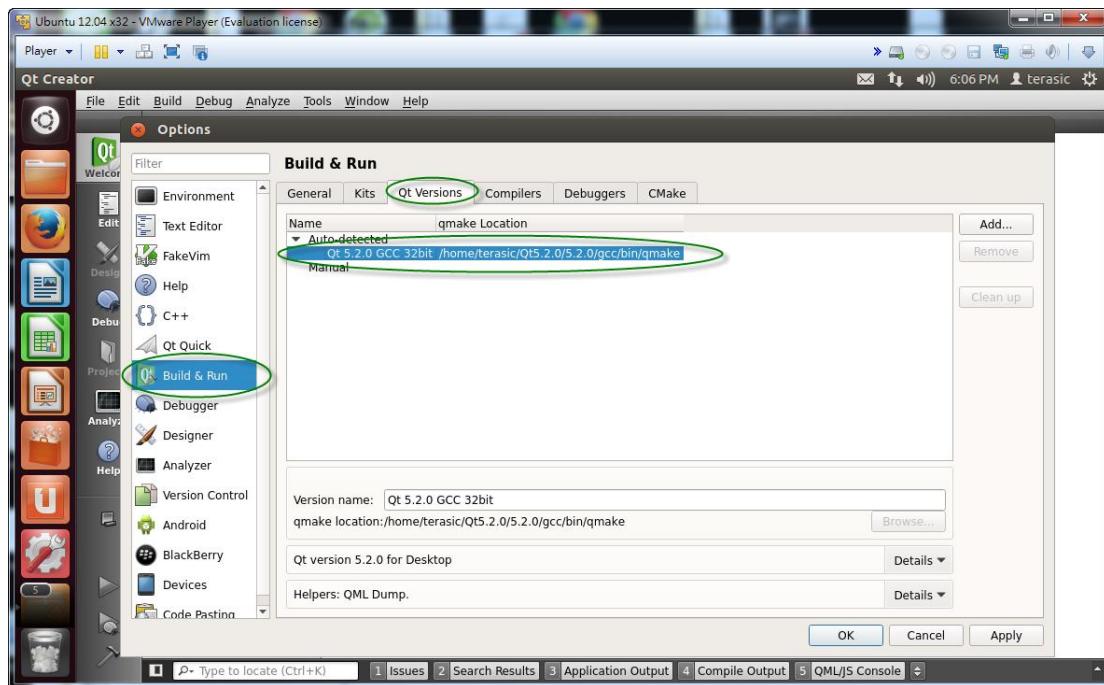
**Figure 3-18 Open Option Dialog**

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 3-19**.



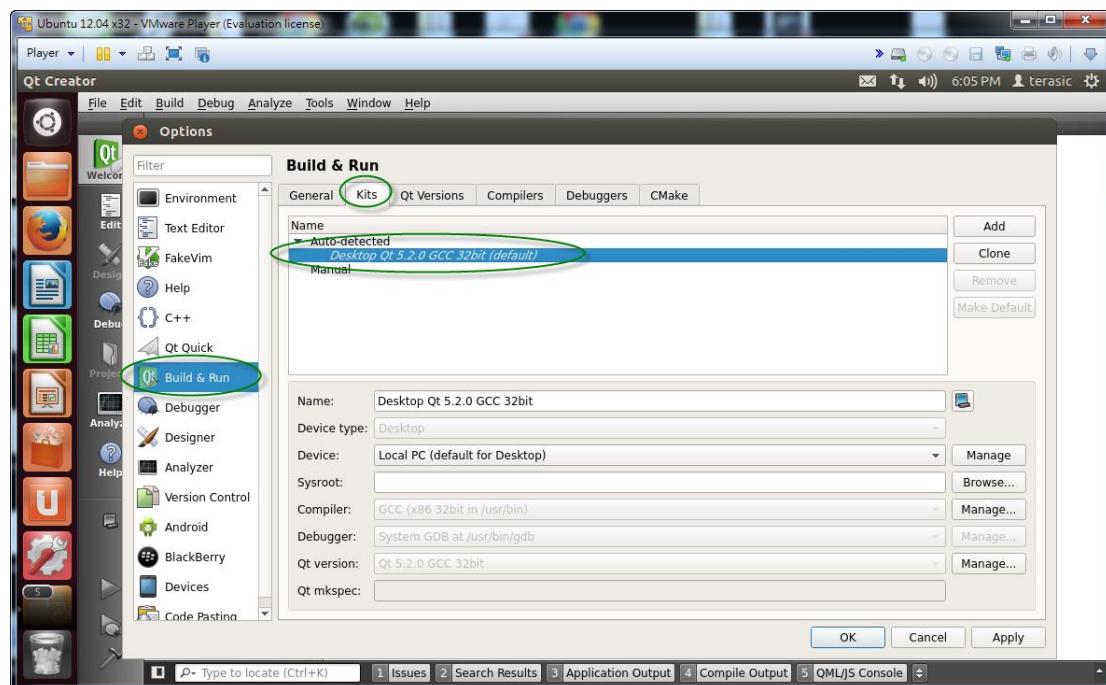
**Figure 3-19 Compilers Options in QT Creator**

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 3-20**.



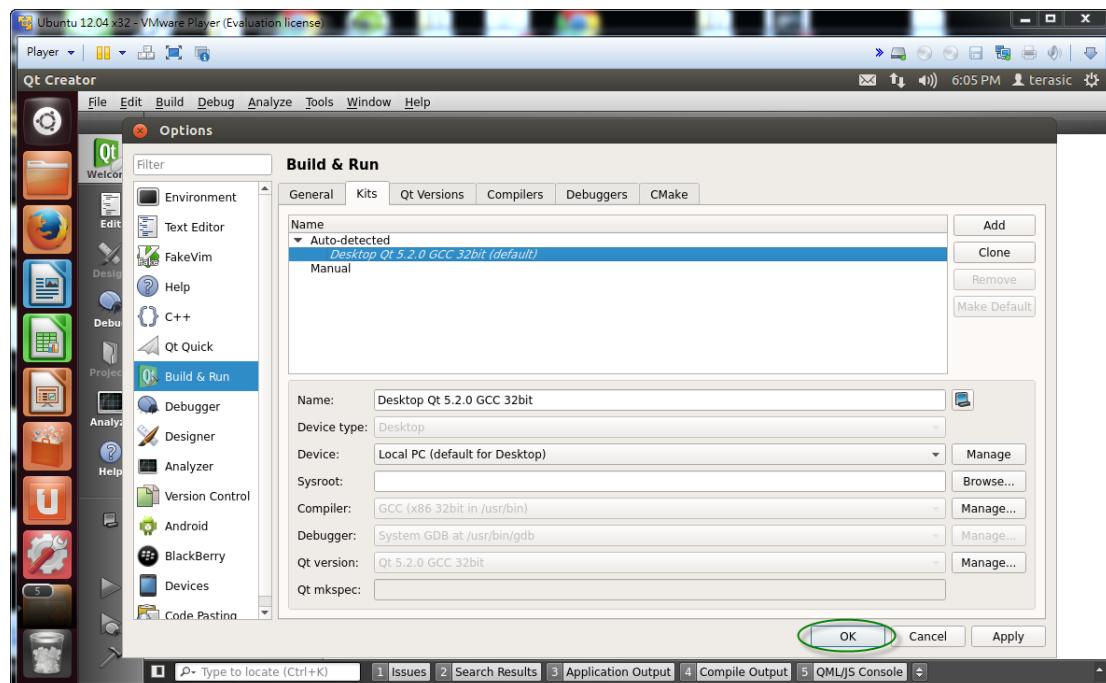
**Figure 3-20 QT Version Option in QT Creator**

Similarly, select “Kits” tab to check if the “Desktop Qt 5.2.0 GCC 32bit” is detected as shown in **Figure 3-21**.



**Figure 3-21 Kits Option in QT Creator**

Finally, click “OK” to close the dialog as shown in **Figure 3-22**.



**Figure 3-22 Click “OK” to Close Options Dialog**

## 3.4 Hello Program

Now we are ready to create, build, and run our first program “Hello” in QT Creator. 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 3-23** to open a new dialog.

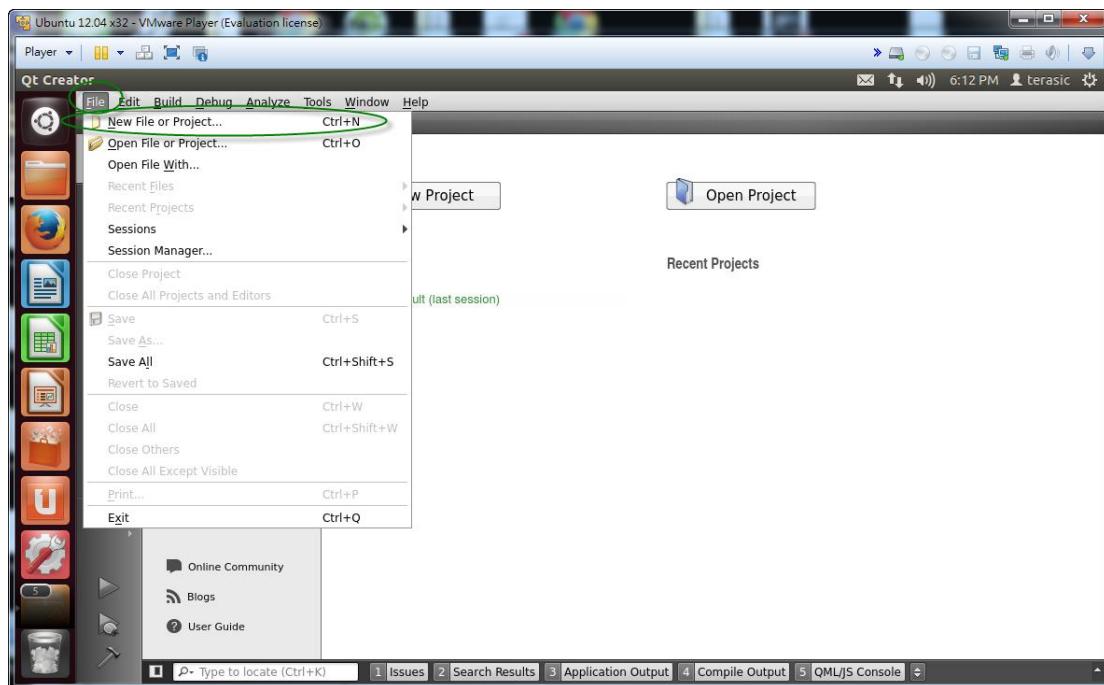
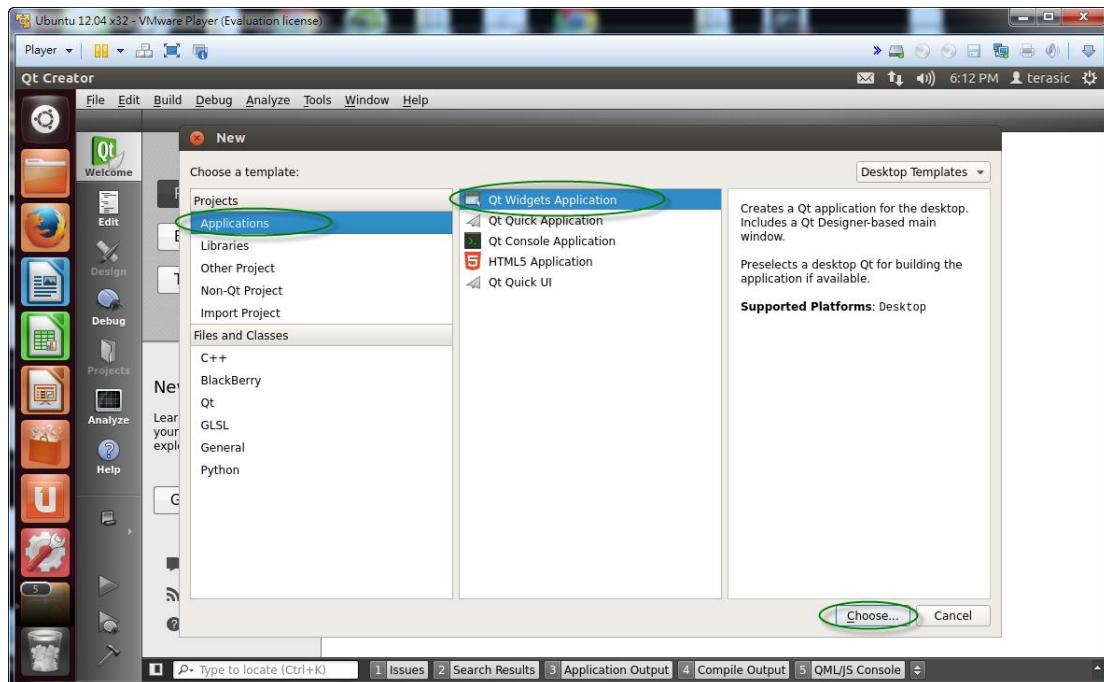


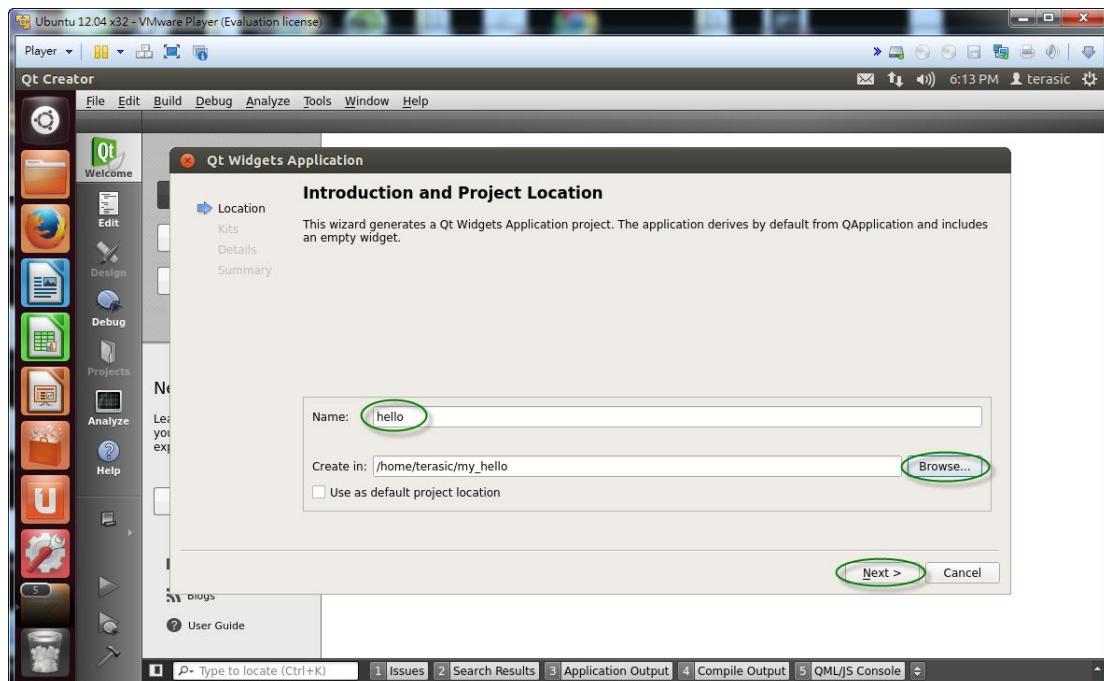
Figure 3-23 Open a New Project Dialog

In the **New** dialog, select “Applications” under Projects, and choose “Qt Gui Application” as shown in **Figure 3-24**. Click “Choose...” to go to the next step.



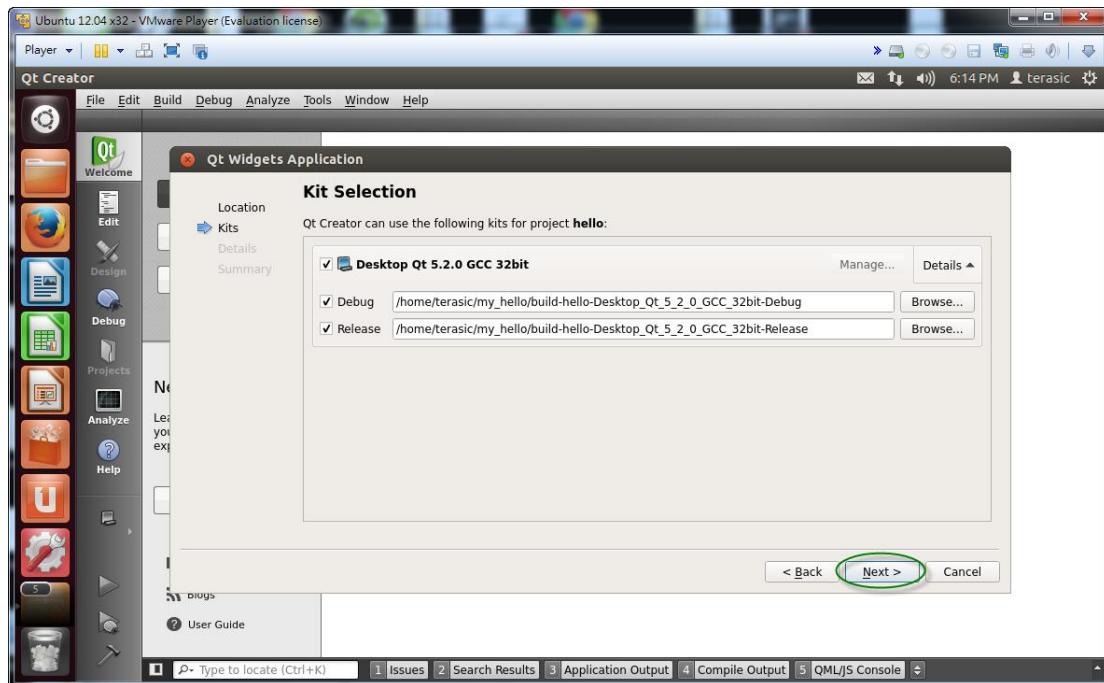
**Figure 3-24 Dialog of Creating a New Project**

In the **Qt Gui Application** Dialog, specify the project name and project location, then click “**Next >**” as shown in **Figure 3-25**.



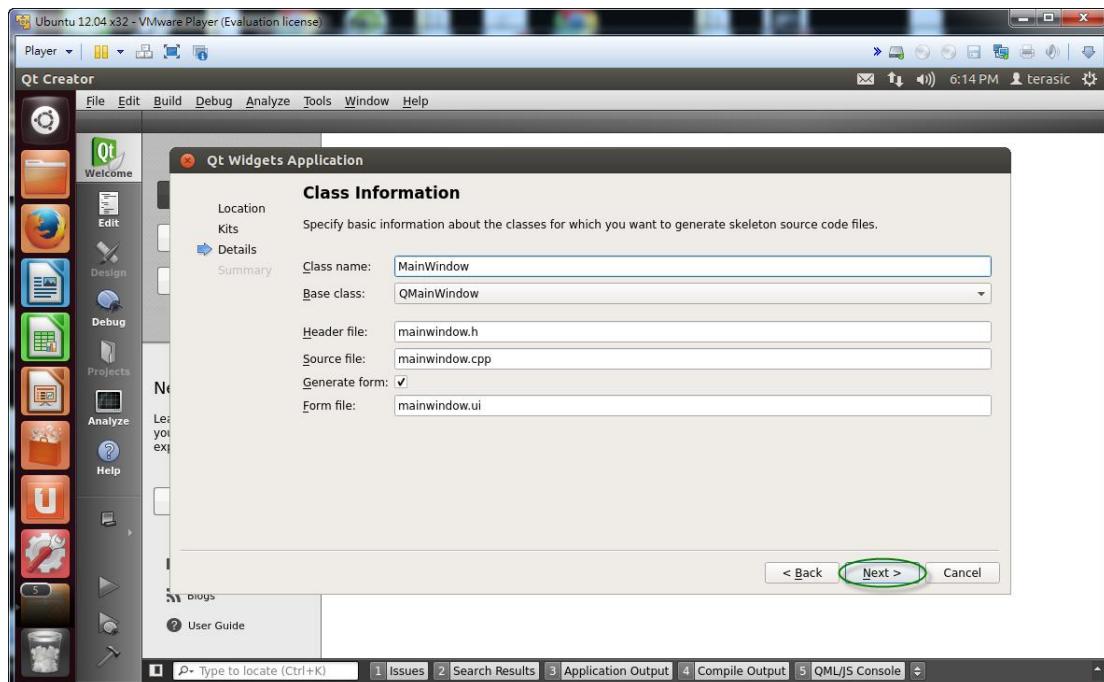
**Figure 3-25 Project Name and Location Dialog**

In the **Kit Selection** dialog, keep default settings and click “Next >” to go to next step as shown in **Figure 3-26**.



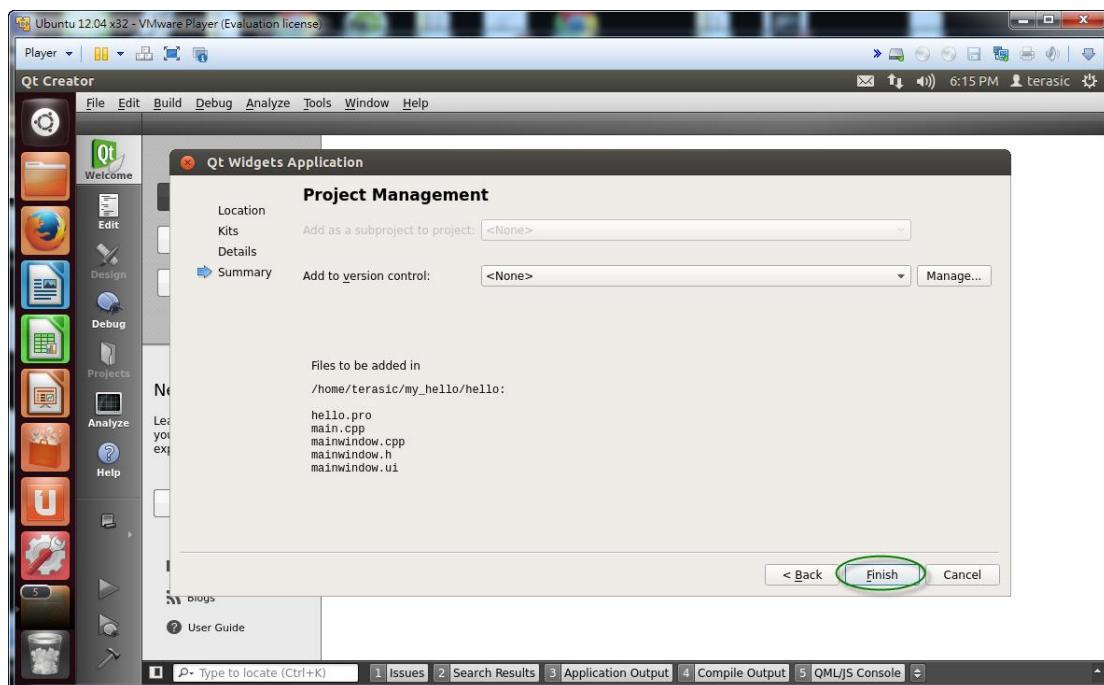
**Figure 3-26 Kit Selection Dialog**

In the **Class Information** dialog, click “Next >” as shown in **Figure 3-27** to proceed.



**Figure 3-27 Class Information Dialog**

In the **Project Management** dialog, click “Finish” as shown in [Figure 3-28](#) to proceed.



**Figure 3-28 Final Dialog of Qt Gui Application**

[Figure 3-29](#) 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.

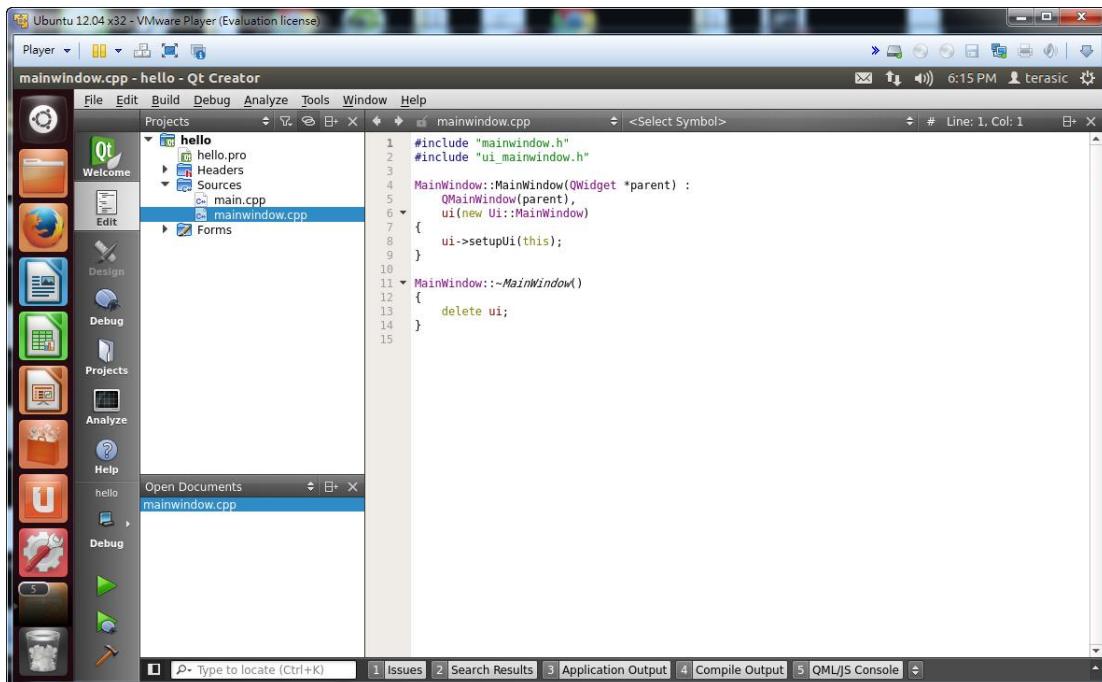
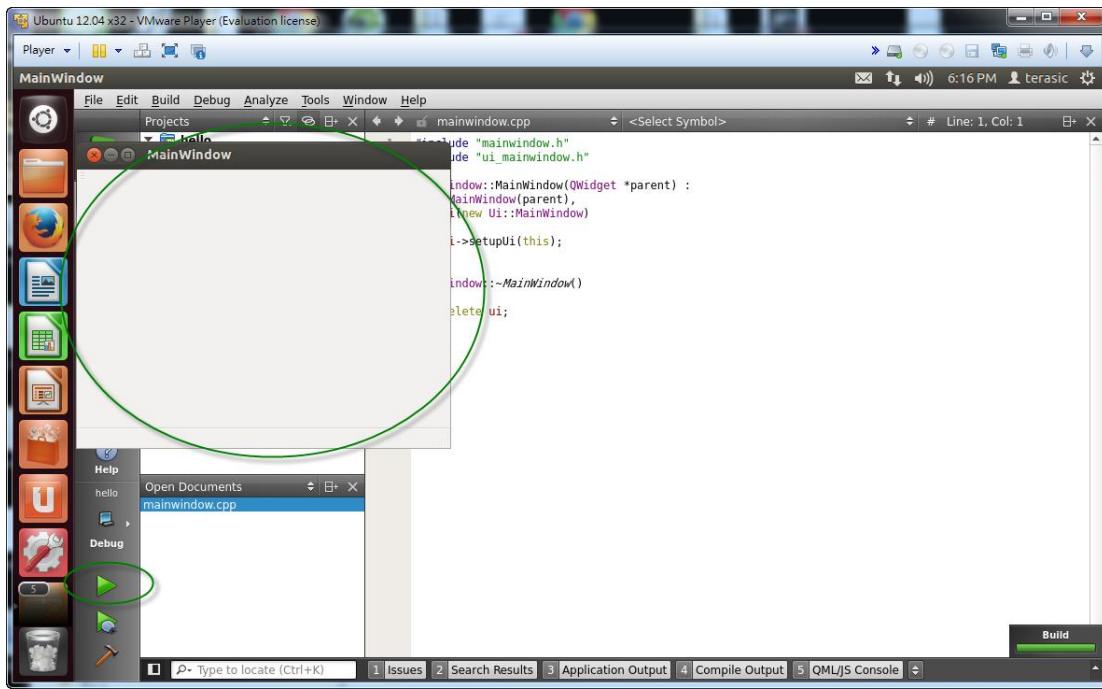


Figure 3-29 Mainwindow.cpp of the Hello project

## ■ Build & Run

Click on the “Run” icon to build and run the Hello program. The GUI program should appear as shown in **Figure 3-30**.



**Figure 3-30 Run Hello Program**

## Chapter 4

# *Altera SoC Tool-Chain Installation*

Altera SoC tool-chain is required while building the QT 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:

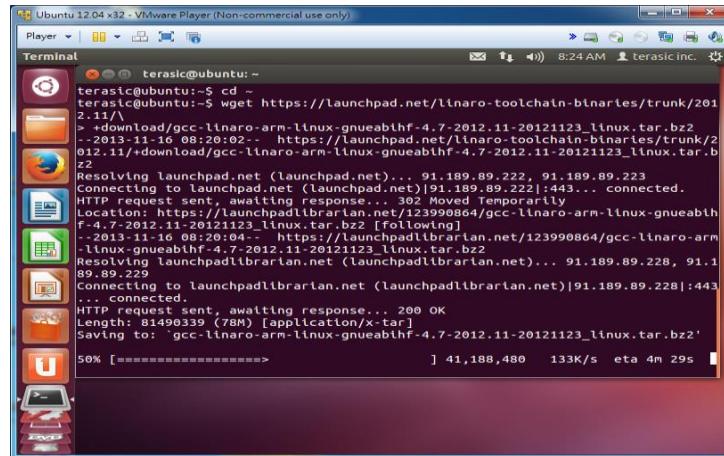
- Download the Altera SoC Tool-Chain and extract the file
- Include Tool chain path into the system environment variable \$PATH

## 4.1 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,

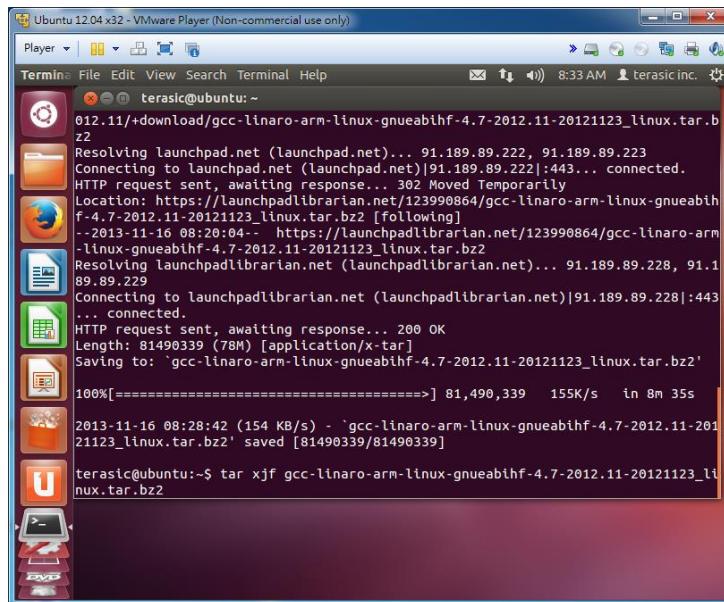
```
$cd ~  
$wget https://launchpad.net/linaro-toolchain-binaries/trunk/2012.11/\  
+download/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux.tar.bz2  
$tar xjf gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123_linux.tar.bz2
```

**Figure 4-1** 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 4-1 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 4-2** 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 4-2 Altera SoC Tool-Chain Download Finished and Extracted by the Tar Command**

## 4.2 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.

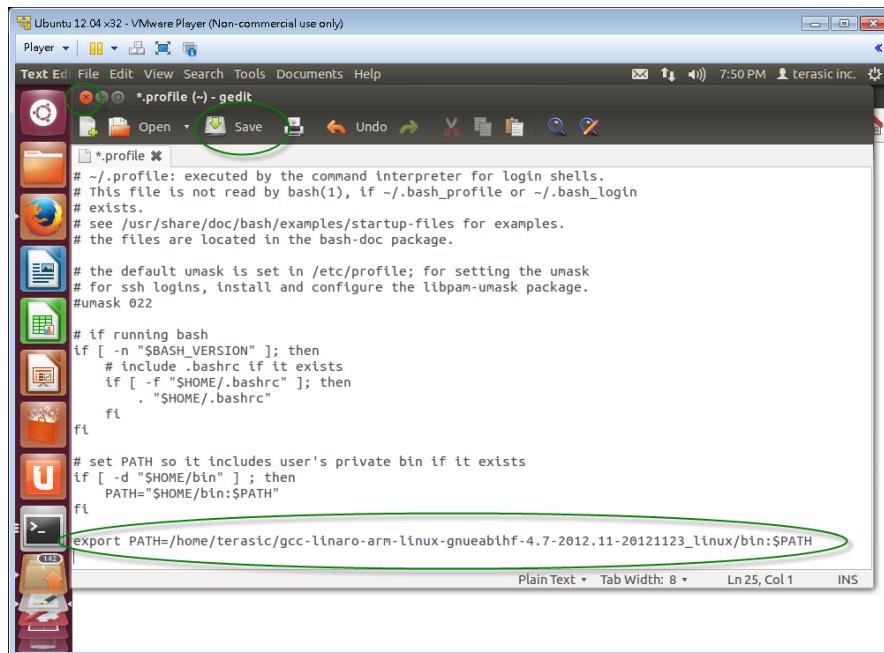
Launch terminal (CTRL+ALT+T) and type in the following command to open the batch file “`~./profile`”

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

**Figure 4-3** shows the screenshot that the “`~./profile`” 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 “terasic” 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 4-3 ./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 5

# *Build QT Library for Altera SoC*

Up to now, you should have successfully installed the Altera SoC tool-chain in the Ubuntu system. Now we start building QT library for Altera SoC ARM on the Ubuntu Linux.

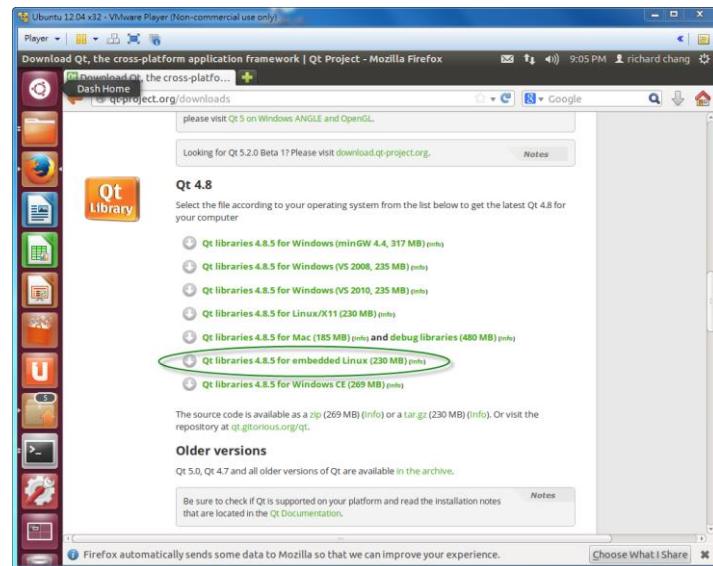
Here is a quick look of the 3 simple steps:

- Download and extract the QT source code
- Create a make configuration folder ‘mkspecs’ for Altera SoC
- Configure, build and install QT Library

### **5.1 Download the QT source code**

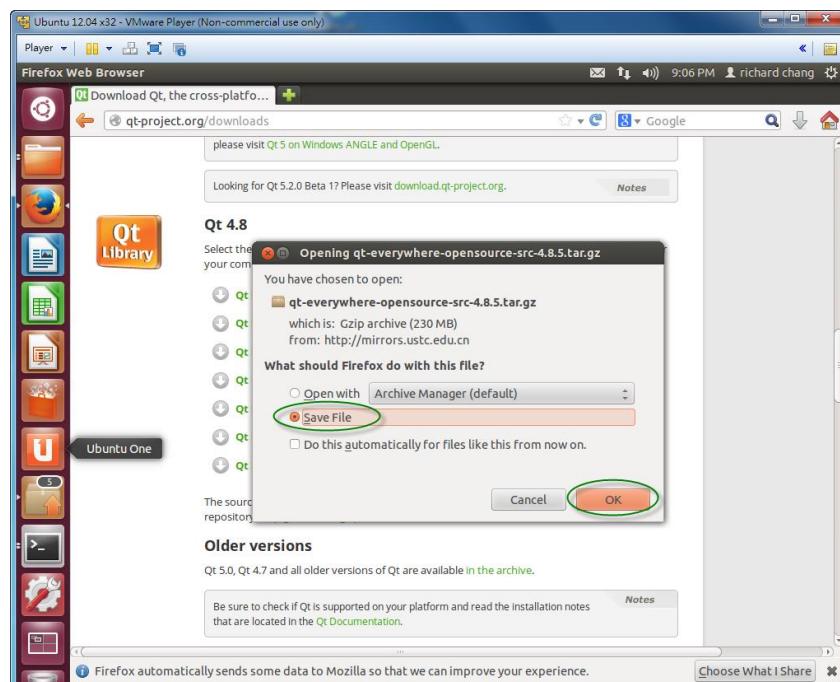
#### ■ **Download Compressed Source File**

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 download 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 -zxf qt-everywhere-opensource-src-4.8.5.tar.gz
```

Note the 2<sup>nd</sup> command line ‘ls’ checks if the source file exists and the 3<sup>rd</sup> command line is used to extract the source file, as shown in **Figure 5-3**. 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 to 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 2<sup>nd</sup> command line is to browse the folder where the folder “linux-arm-gnueabi-g++” exists and the 3<sup>rd</sup> command line is to duplicate the “linux-arm-gnueabi-g++” folder and rename it as “linux-arm-gnueabihf-g++” s shown in **Figure 5-3**

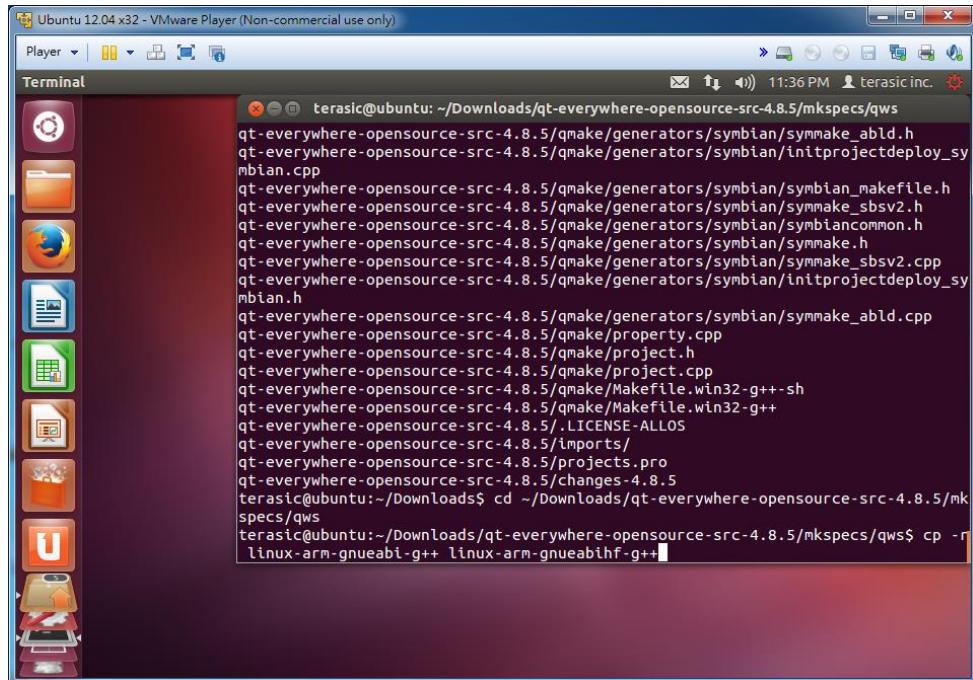
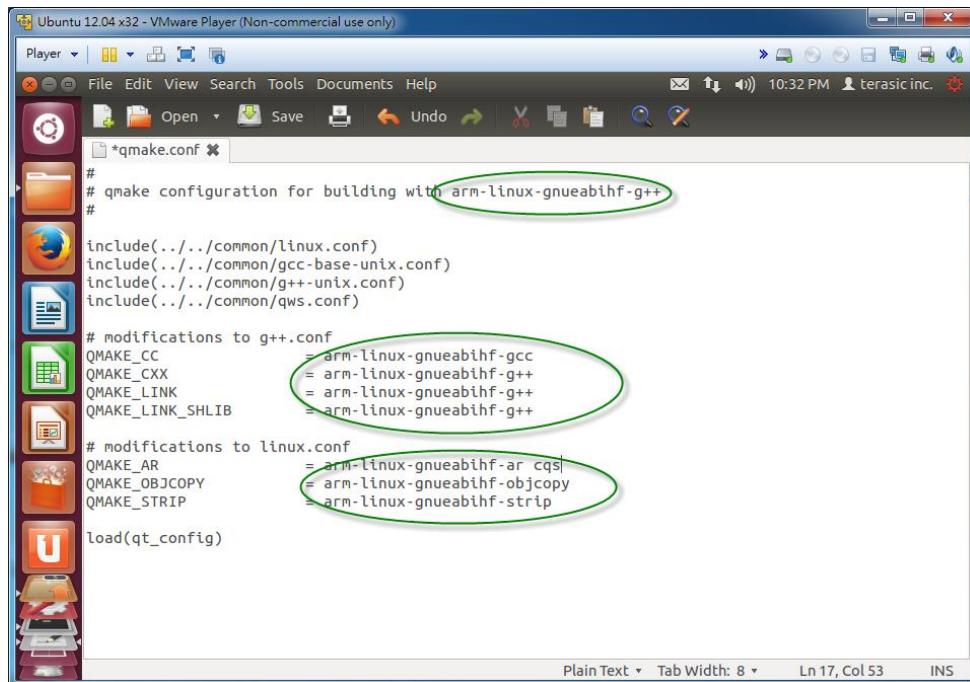


Figure 5-3 Duplicate Configuration Folder

## ■ 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, replace all “arm-none-linux-gnueabi” with “arm-linux-gnueabihf” as shown in **Figure 5-4**. 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-altera-soc -release -shared \
-nomake examples -nomake tools -nomake docs -make demos \
-xplatform qws/linux-arm-gnueabihf-g++ -embedded arm -little-endian \
-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.

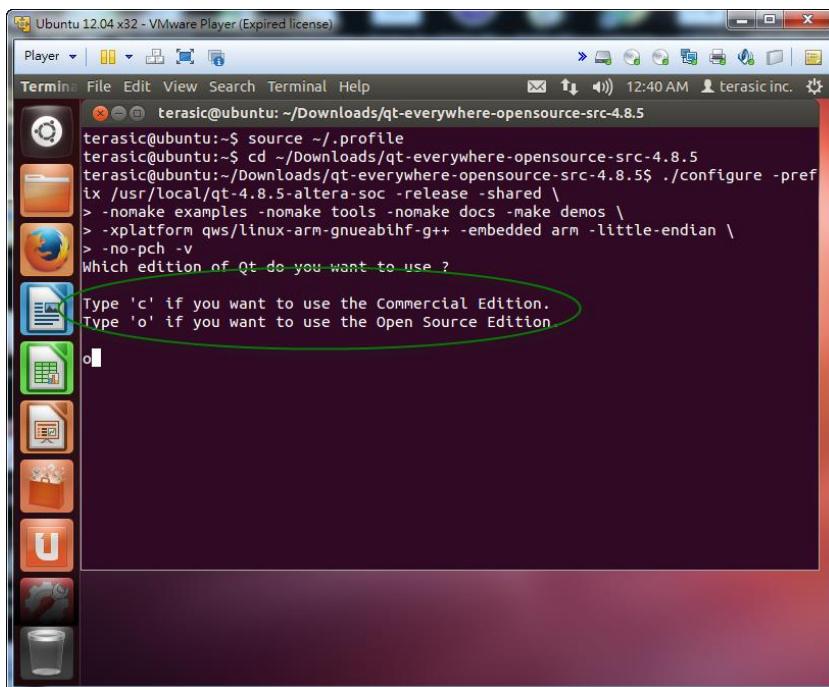
To go to the source code folder, type

```
$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-altera-soc -release -shared \
-nomake examples -nomake tools -nomake docs -make demos \
-xplatform qws/linux-arm-gnueabihf-g++ -embedded arm -little-endian \
-no-pch -v
```

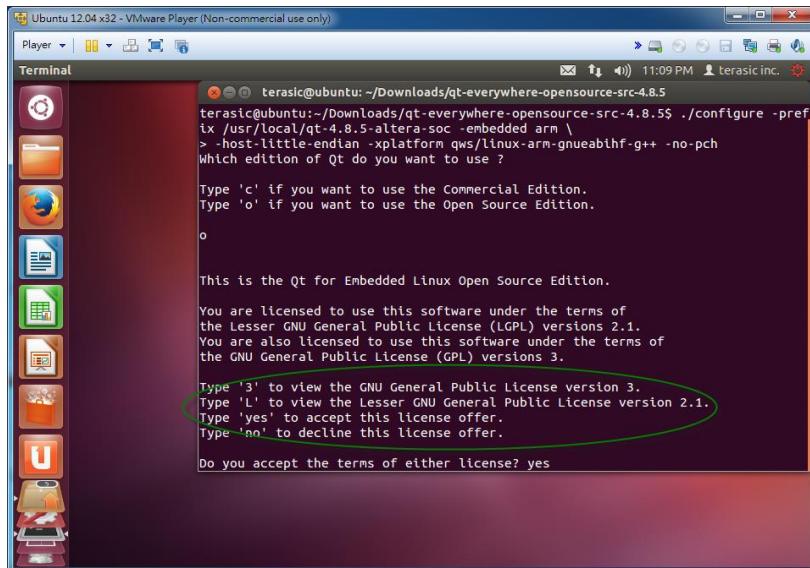
Before executing the configure batch, system will prompt to ask license type as shown in **Figure 5-5**, type in “o” and press ENTER.



**Figure 5-5 Ask License Type**

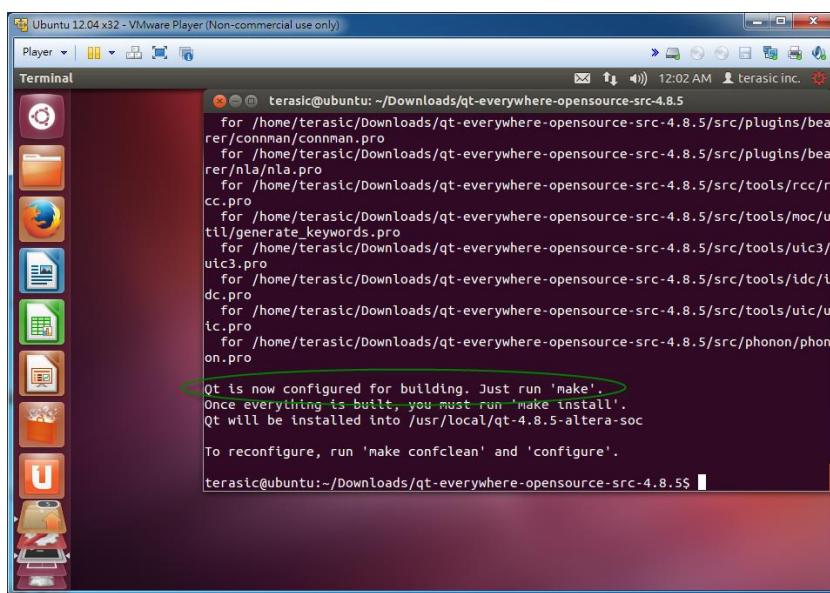
After selecting the license type, system prompts to ask whether to accept the license term or not as

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**.



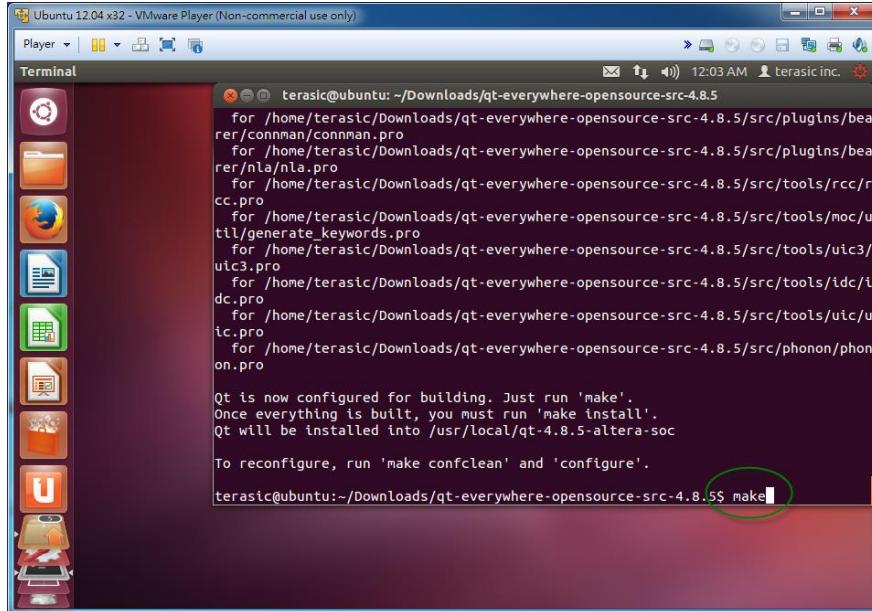
**Figure 5-7 Qt Now Configured for Building**

## ■ Make

In the terminal simply type:

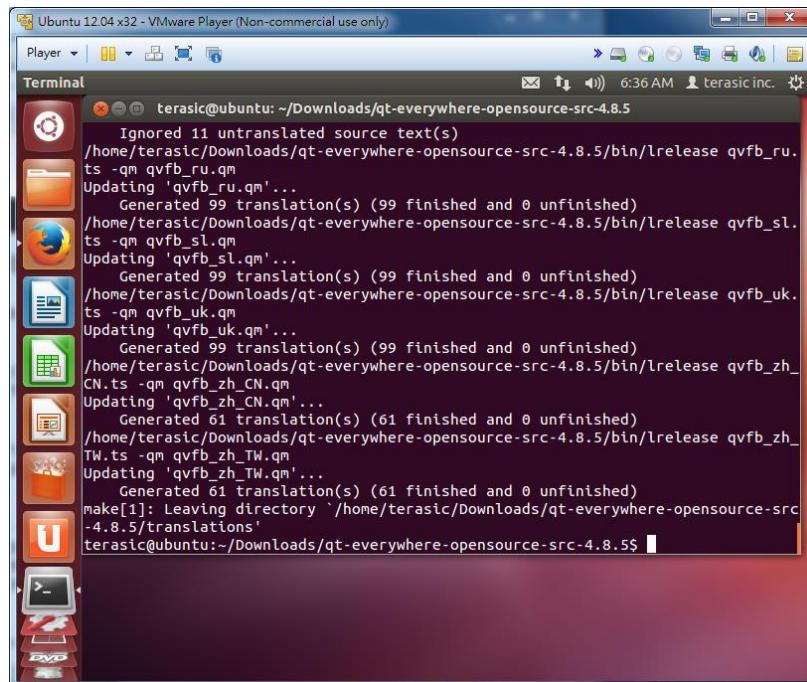
```
$make
```

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.



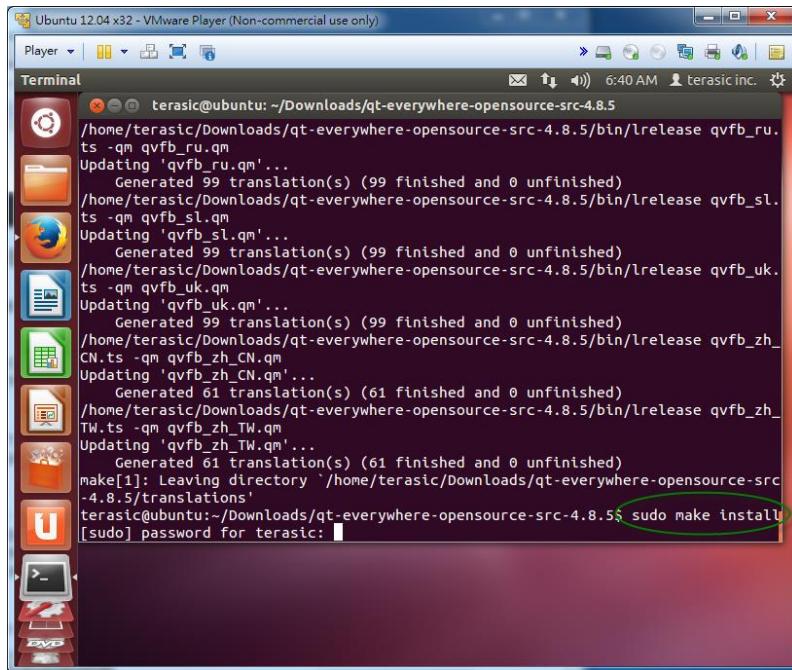
**Figure 5-9 Make Process Now Complete**

## ■ Install

Finally, to install the QT package please type in

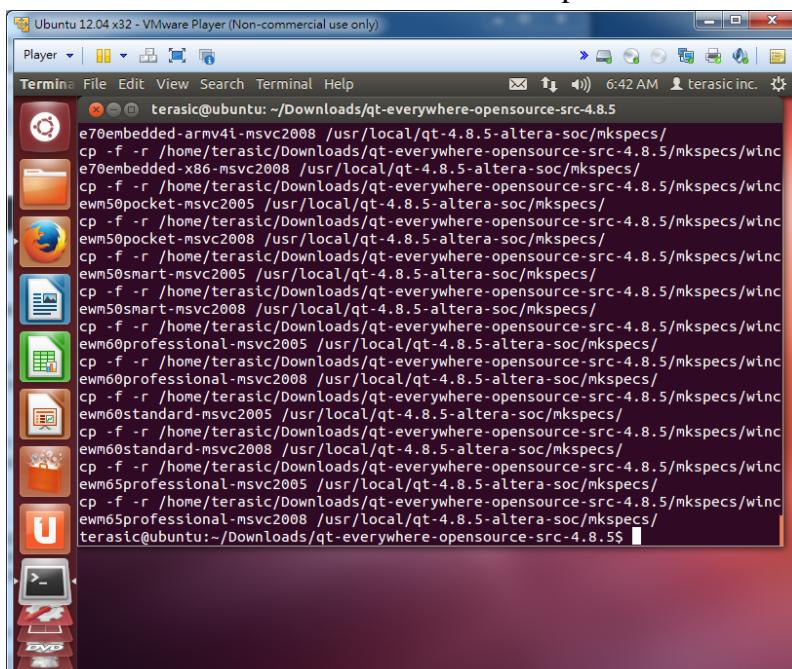
```
$sudo make install
```

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



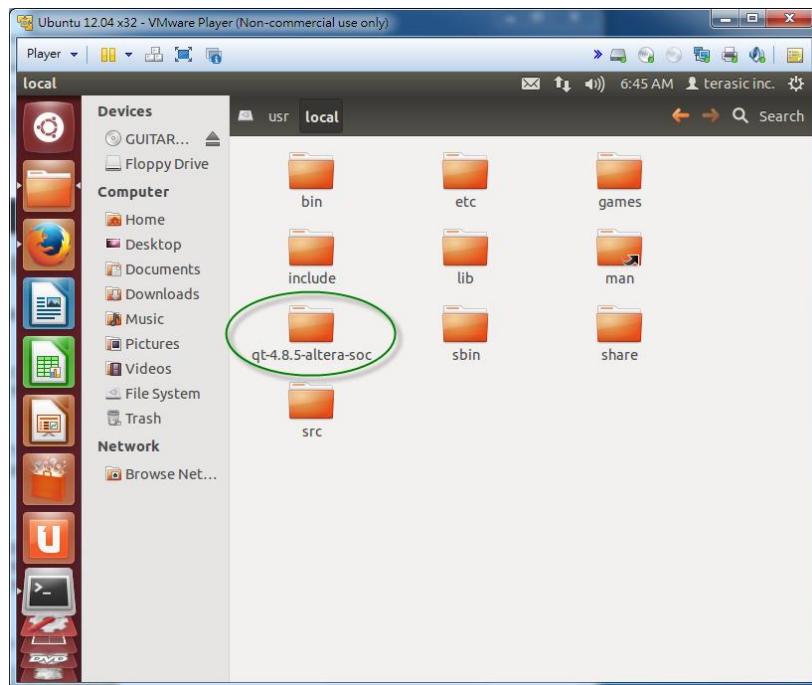
**Figure 5-10 Input User Password to Continue the Installation**

**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-altera-soc” as shown in **Figure 5-12**.



**Figure 5-12 Installed Location of Altera SoC QT Package**

## Chapter 6

# *QT App for Altera SoC*

Chapter 6 describes how to cross-compile the Hello project we created back in chapter 3, such that the hello program can run on the target platform – Altera DE1-SoC FPGA development board in this tutorial. Here we assume that Altera SoC Tool-Chain and QT library for Altera SoC FPGA development board have been successfully installed and are now available on the system.

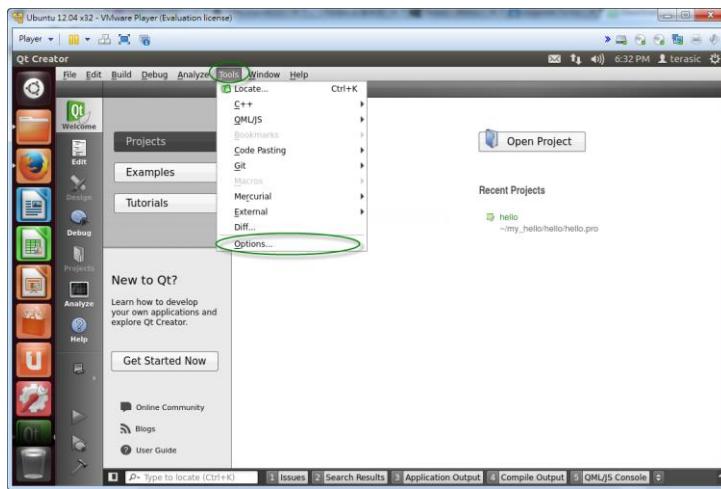
Here is a quick look of the 3 simple steps:

- Set up “Build & Run” in QT Creator
- Cross-compile the Hello project
- Execute Hello program

## **6.1 Set up “Build & Run” in QT Creator**

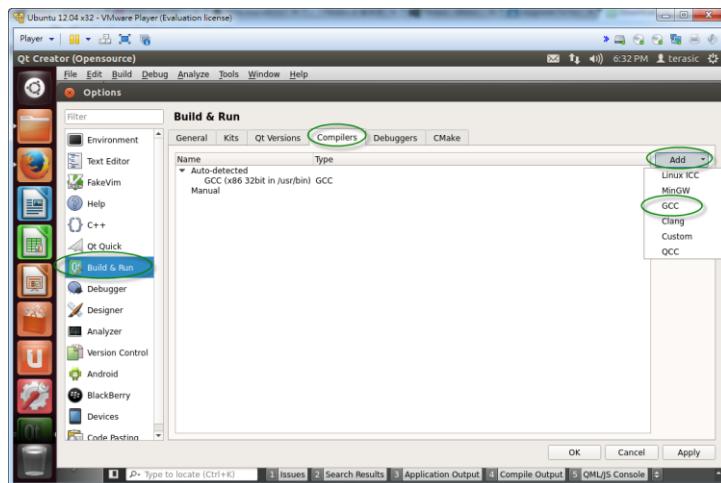
### ■ Compiler Setup

First launch the QT Creator, and select the menu item “Tools→Options...” as shown in **Figure 6-1** to open the **Option** dialog.



**Figure 6-1 Open Option Dialog**

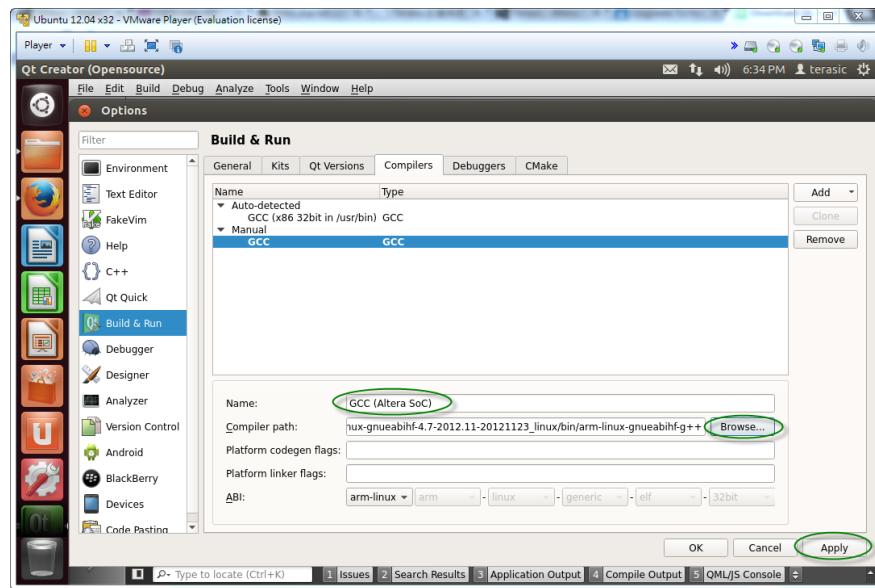
To add the “GCC” compiler, in the Option dialog, select “Build & Run” item on the left and then select “Compilers” tab. Now you can select “GCC” item from the “Add” pull down menu as shown in **Figure 6-2**.



**Figure 6-2 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

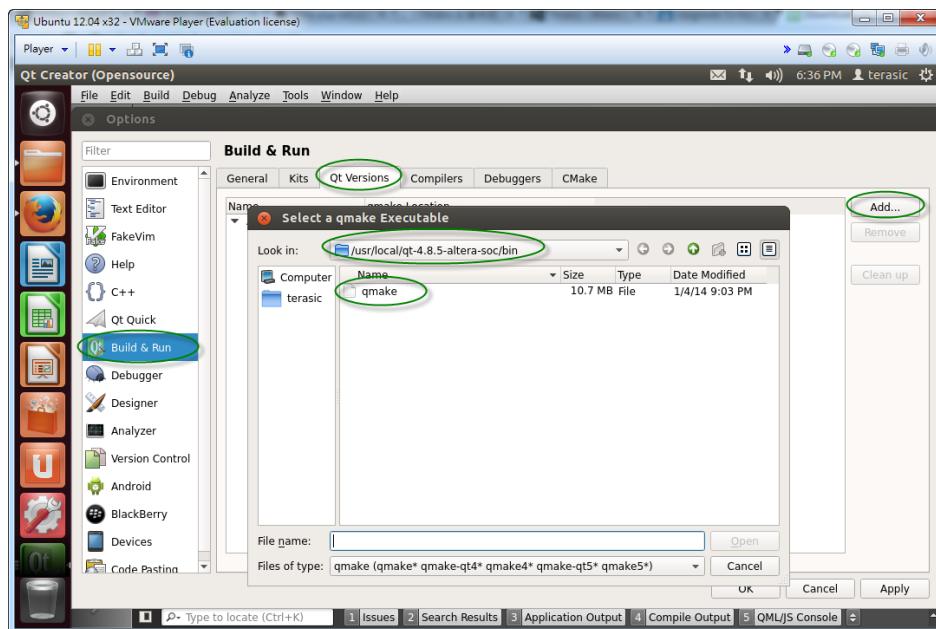
“/home/**terasic**/gcc-linaro-arm-linux-gnueabihf-4.7-2012.11-20121123\_linux/  
/bin/arm-linux-gnueabihf-g++”. Note that in the path string, you should replace “**terasic**” with your linux user name. Then click “Apply” to finish the Compiler setup as shown in **Figure 6-3**.



**Figure 6-3 Specify Compiler Name and Path**

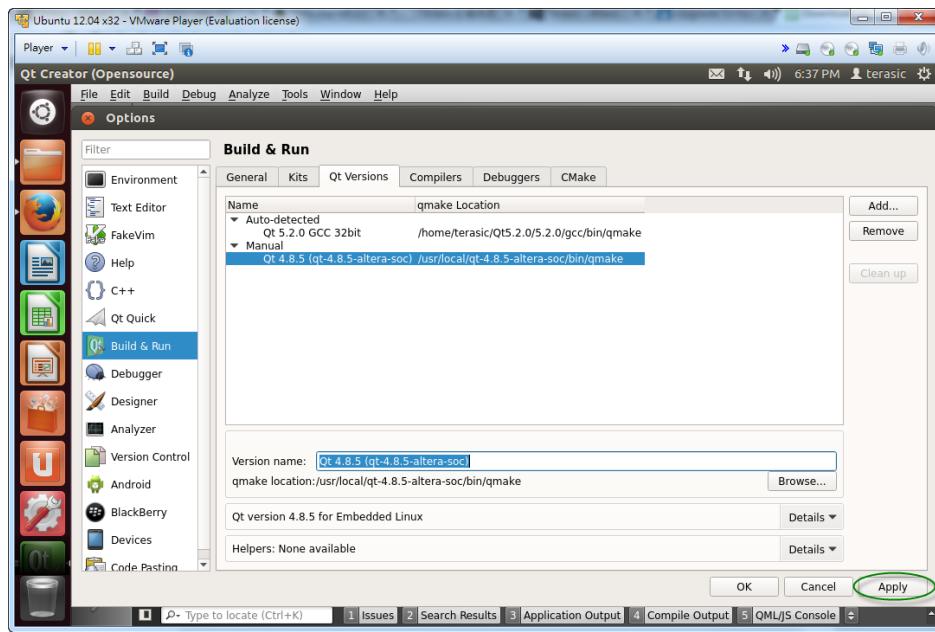
## ■ Qt Versions Setup

To add Qt Versions, first click on the “Qt Versions” tab and click “Add” as shown in [Figure 6-4](#). When an **Open File** dialog appears to ask the location of qmake executable, please specify the location as: “`/usr/local/qt-4.8.5-altera-soc/bin/qmake`”.



**Figure 6-4 Add Qt Version**

Finally, click “Apply” to complete the **Qt Versions** setup as shown in [Figure 6-5](#).



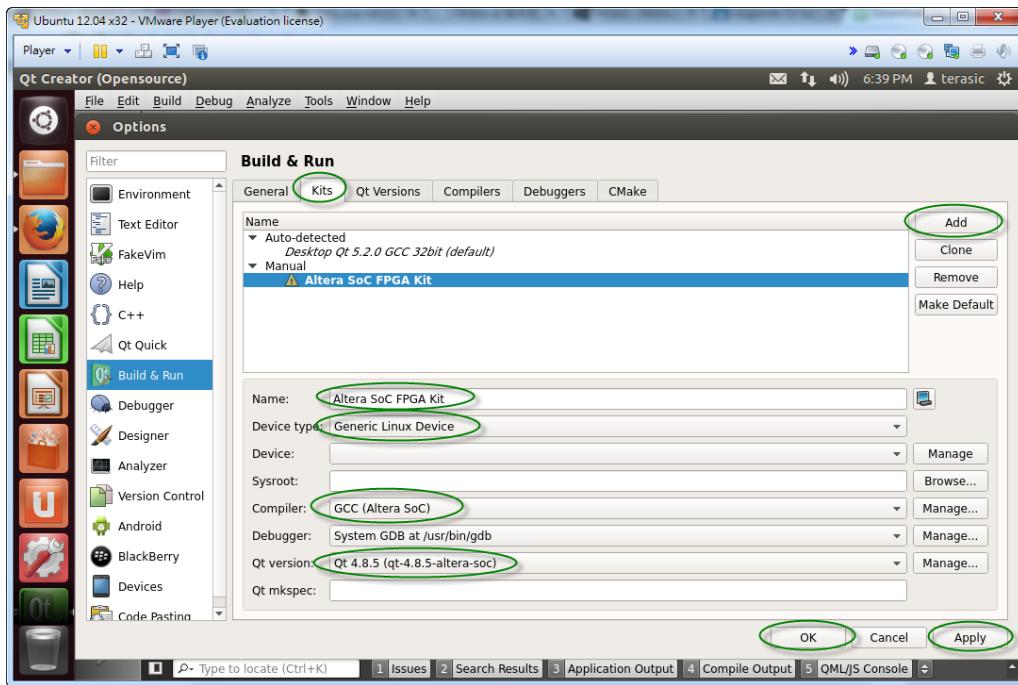
**Figure 6-5 Apply the Qt Version**

## ■ Kits Setup

To add Kits, first click on the “Kits” tab and click on “Add” as shown in **Figure 6-6**. 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-altera-soc))”

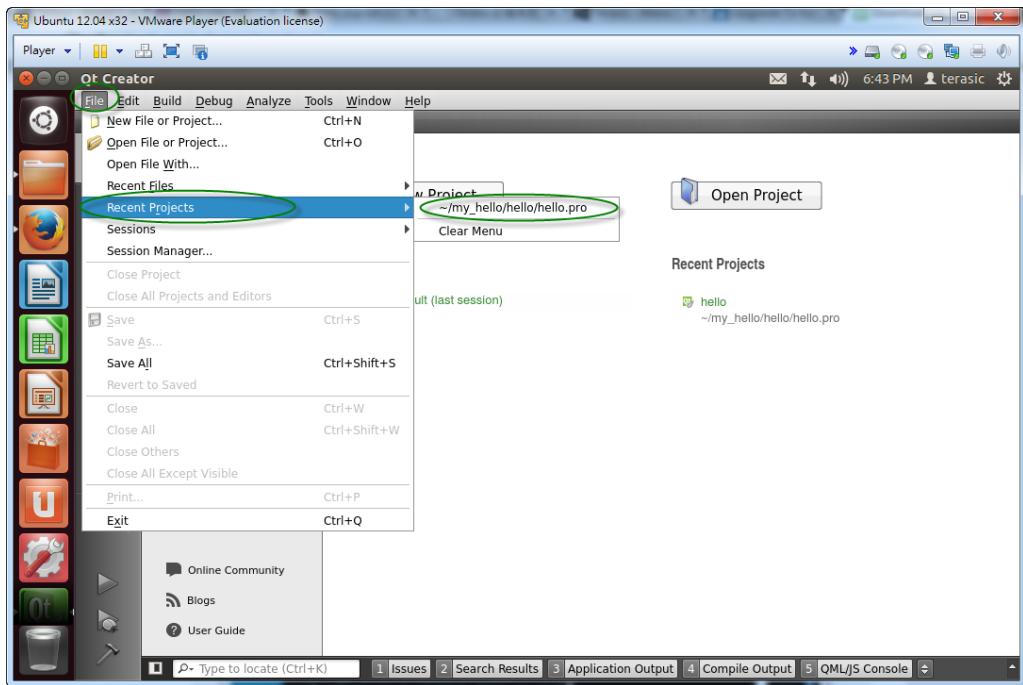
Then, click “Apply” to finish **Kits** setup and click “OK” to finish Options setup.



**Figure 6-6 Add Altera SoC FPGA Kit**

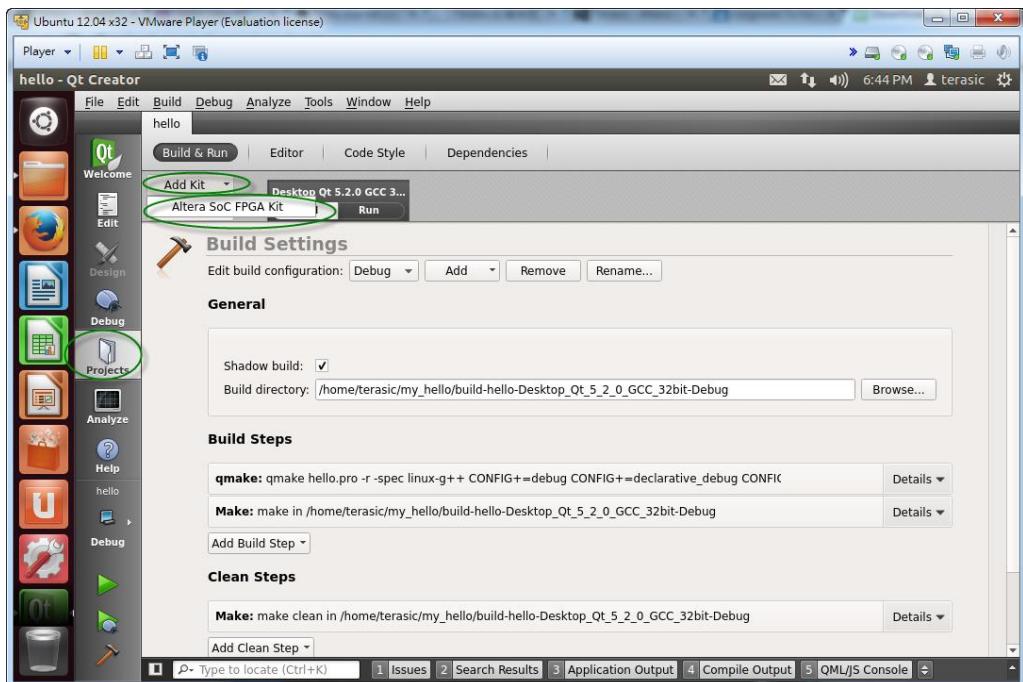
## 6.2 Cross-Compile the Hello Project

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 6-7**.



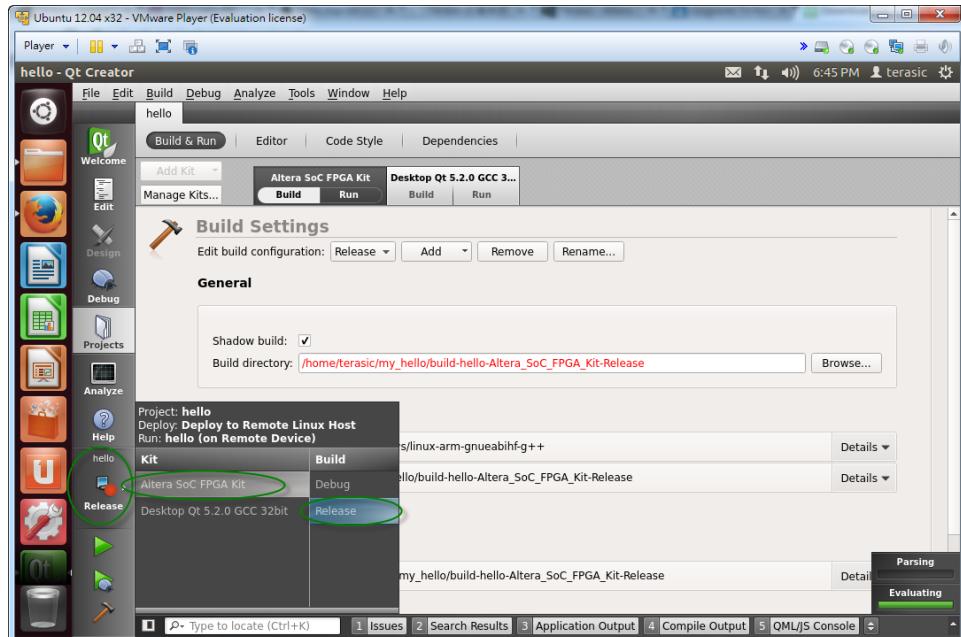
**Figure 6-7 Open the Hello Project**

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 6-8**.



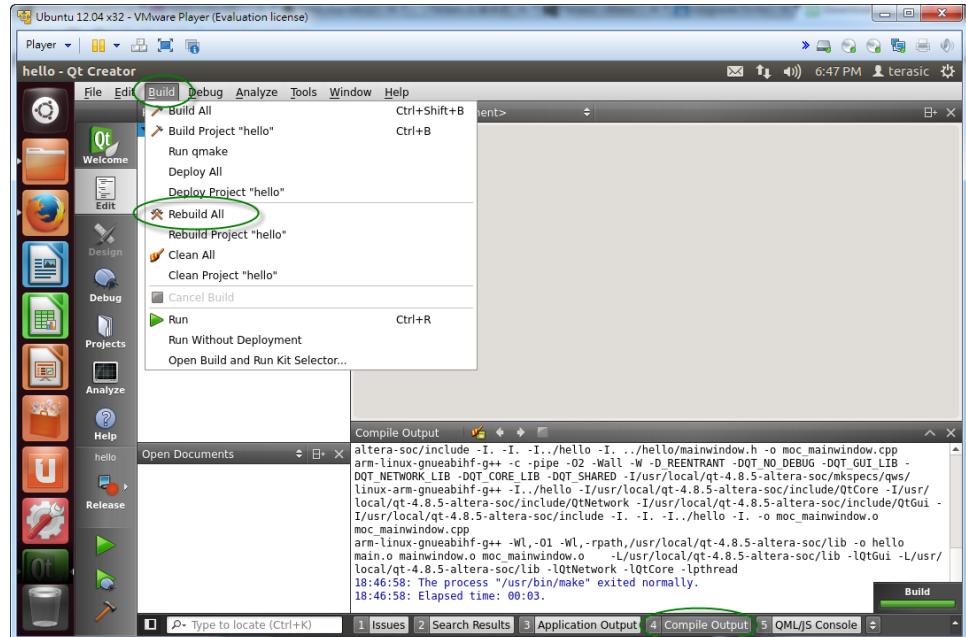
**Figure 6-8 Add Altera SoC FPGA Kit into the Hello Project**

To select **Release** type, click on “Release” icon and select “Altera SoC FPGA Kit” item under the **Kit** list menu and “Release” in **Build** list menu, as shown in **Figure 6-9**.



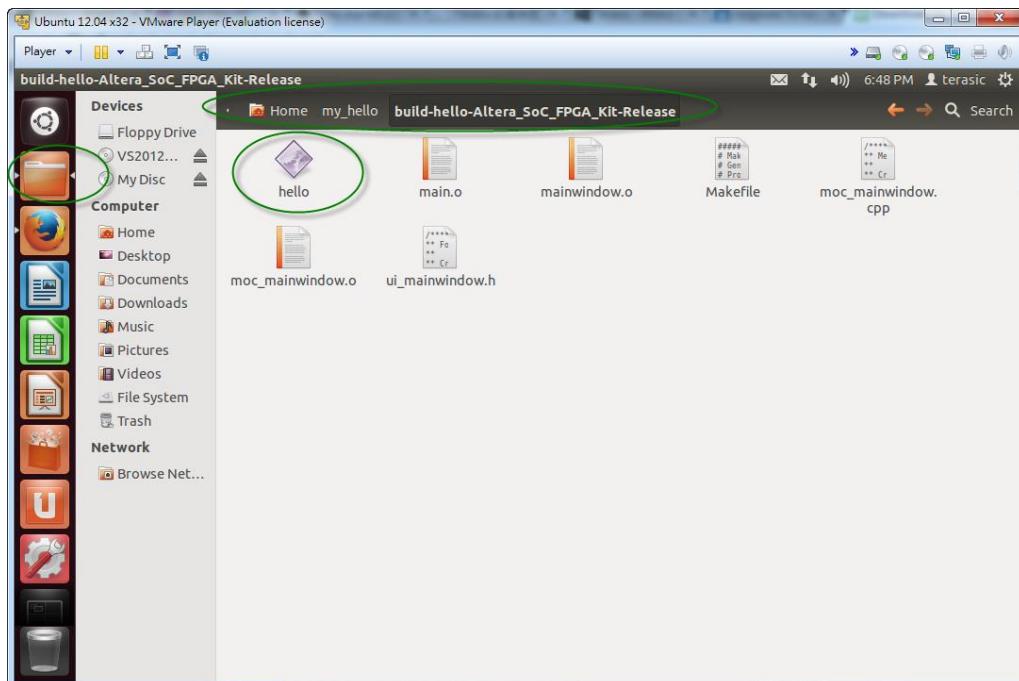
**Figure 6-9 Specify Release Setting**

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 6-10**. While compiling, the system message will be simultaneously displayed on the “Compile Output” Windows.



**Figure 6-10 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 6-11**.



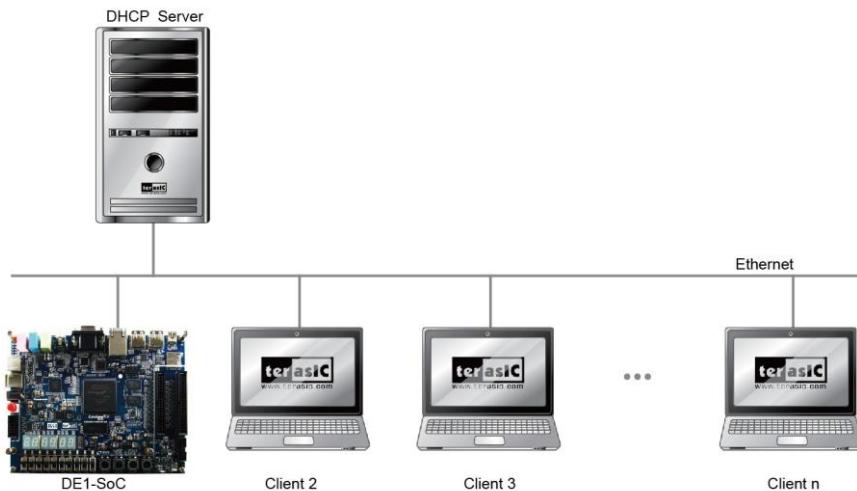
**Figure 6-11 Location of the Output Hello Execution File**

## 6.3 Execute Hello Program

To execute the Hello program on the Altera DE1-SoC FPGA board, we need to copy the **hello** execution file and **QT library for Altera SoC** to the Altera SoC FPGA Board which runs on Linux. In this demonstration, we use Linux “scp” command to remote copy these files from host PC to the Altera DE1-SoC FPGA development board.

### ■ Setup Network for Altera SoC FPGA Board

First, use a RJ45 cable to connect the Altera SoC FPGA development board to an Ethernet network with a DHCP server, like **Figure 6-12**.



**Figure 6-12 Connect Altera SoC FPGA Board To the Ethernet Network**

Then, use the microSD card which Control Panel is written to, described in chapter one, to boot the Altera SoC FPGA development board. After logging into Linux, type in “udhcpc” command to request an Ethernet IP Address from the DHCP server as shown in **Figure 6-13**.

```
root@socfpga_cyclone5:~# udhcpc
udhcpc (v1.20.2) started
Sending discover...
Sending select for 192.168.1.108...
Lease of 192.168.1.108 obtained, lease time 86400
/etc/udhcpc.d/50default: Adding DNS 192.168.1.21
/etc/udhcpc.d/50default: Adding DNS 192.168.1.31
root@socfpga_cyclone5:~#
```

**Figure 6-13 Linux UDHCPC Command**

To get a IP address from the UDHC server, type Linux “ifconfig” command to find the assigned IP address, as shown in **Figure 6-14**. In this case, the assigned IP address is “192.168.1.114”.

```
root@socfpga_cyclone5:~# ifconfig
eth0      Link encap:Ethernet HWaddr 6a:c9:47:7a:63:6b
          inet addr:192.168.1.114 Bcast:0.0.0.0  Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                  RX packets:3891 errors:0 dropped:154 overruns:0 frame:0
                  TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:356599 (348.2 KiB)  TX bytes:5858 (5.7 KiB)
                  Interrupt:152 Base address:0x4000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
                  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_cyclone5:~#
```

**Figure 6-14 Use Linux ifconfig Command to Check Assigned Ethernet IP Address**

## ■ Copy Files from PC Host to Altera SoC FPGA Board

Now, back to the host PC part. Please make sure the host PC is connected to the Ethernet which Altera DE1-SoC FPGA development board also is connected to. Launch a terminal, and type in:

```
$cd /home/terasic/my_hello/build-hello-Altera_SoC_FPGA_Kit-Release
```

to go to the directory where the hello execution file is located, as shown in **Figure 6-15**. **Note, in the path string, you should replace “terasic” with your linux user name.**

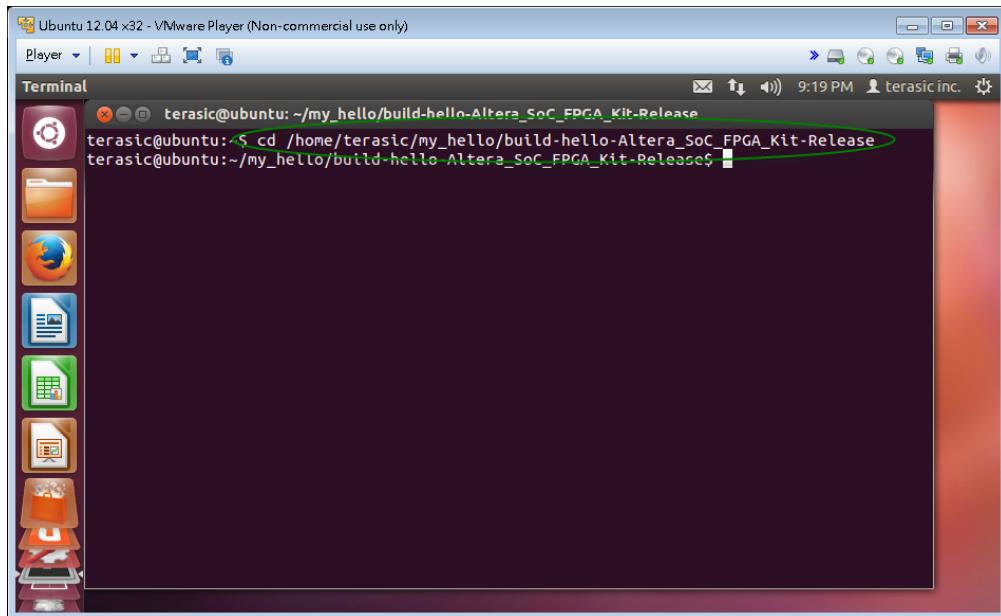


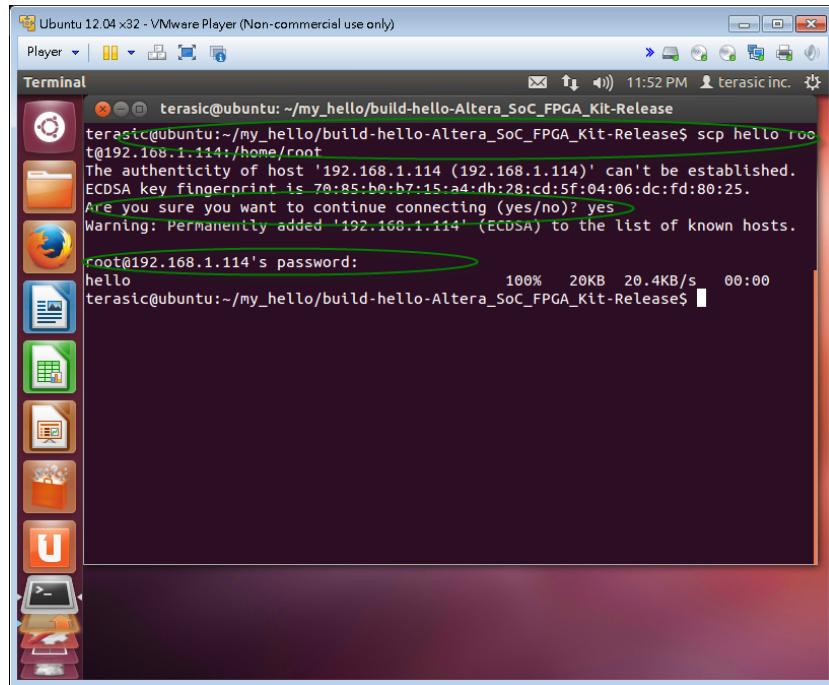
Figure 6-15 Go to the Hello Program Folder

Then, type:

```
$ scp hello root@192.168.1.114:/home/root
```

to remote copy the **hello** program to the /home/root directory of the file system of Altera DE1-SoC FPGA development board , as shown in **Figure 6-16**. Note that the IP address 192.168.1.114 should be changed to the actual IP address assigned.

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.

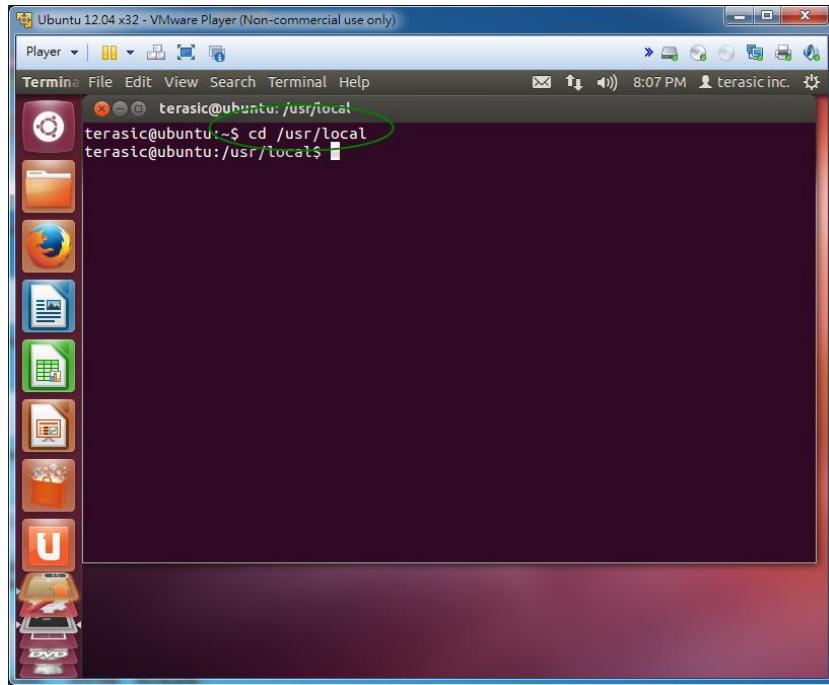


**Figure 6-16 Remote Copy the Hello Program**

Now, we are going to copy the QT library. Please type in:

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

to go to the directory where QT library is located, as shown in [Figure 6-17](#).

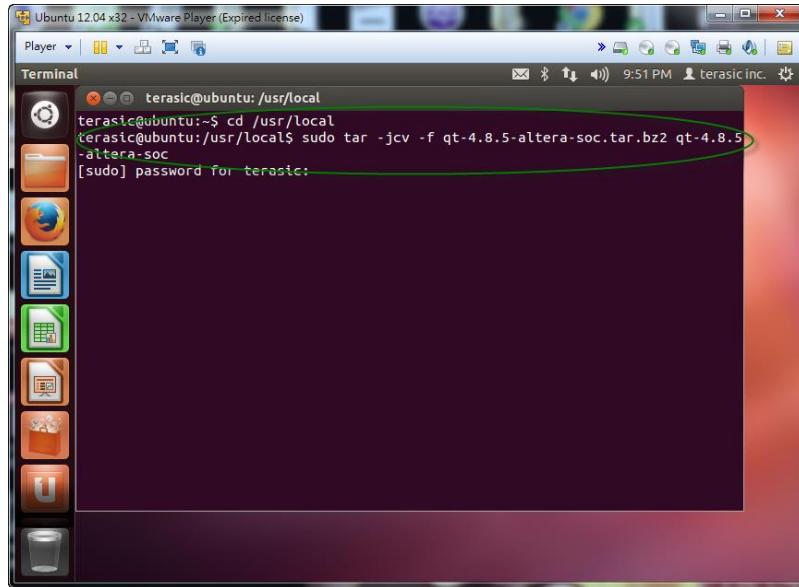


**Figure 6-17 Go to the QT Library Directory**

Please type in the following command to compress the QT library as shown in **Figure 6-18**.

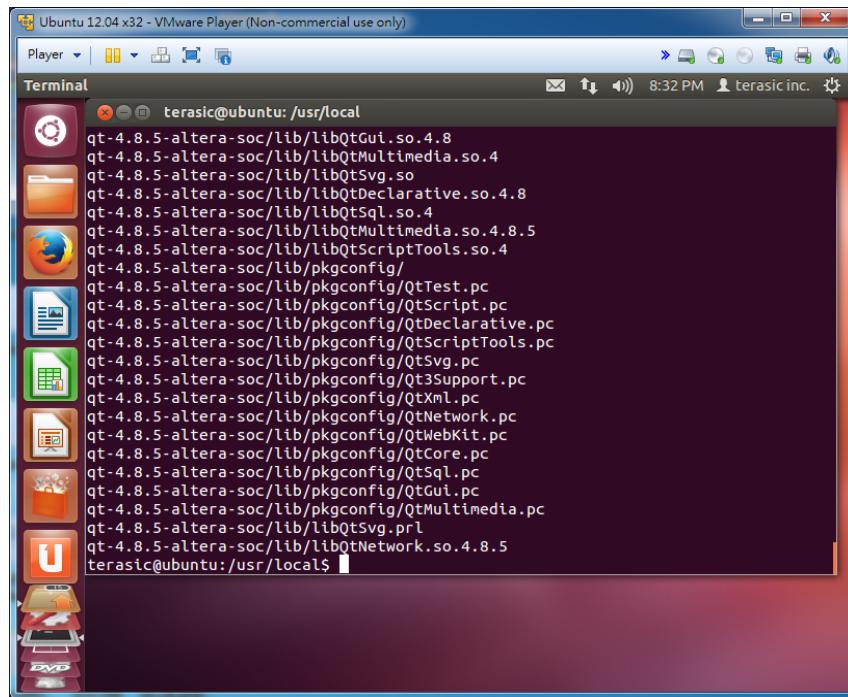
```
$sudo tar -jcv -f qt-4.8.5-altera-soc.tar.bz2 qt-4.8.5-altera-soc
```

When system prompts to request your password, please input your password and press ENTER. (In this tutorial, the password is “123”)



**Figure 6-18 Compressing QT Library**

**Figure 6-19** shows the screenshot when the compressing process is complete.

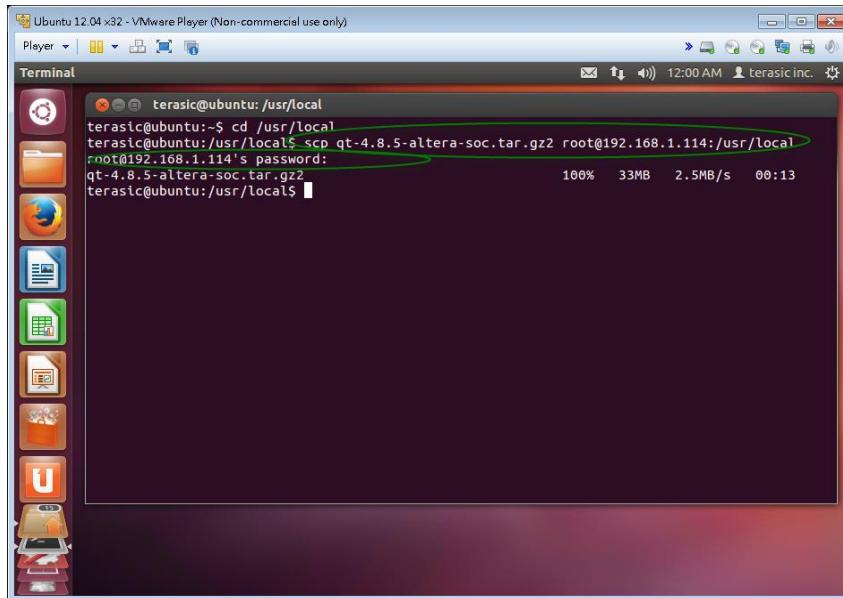


**Figure 6-19 Compressing the QT Library**

Type in the following command

```
$ scp qt-4.8.5-altera-soc.tar.bz2 root@192.168.1.114:/usr/local
```

to remote copy the compressed QT Library to the “/usr/local” directory of the file system of Altera SoC FPGA development board, as shown in **Figure 6-20**. Note that the ip address 192.168.1.108 should be changed to the ip address actually assigned. The system will ask you to provide the password of the remote machine, please type in “terasic” and press ENTER.



**Figure 6-20 Remote Copy Compressed QT Library to Altera SoC FPGA Development Board**

Once the copying process is done, we work in the Linux terminal on Altera SoC FPGA development board. Since the file sent earlier from host PC is in the compressed format, we need to decompress the QT Library before we can use it.

By typing the following command in the Linux terminal on Altera SoC FPGA development board

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

to go to the /usr/local directory, as shown in **Figure 6-21**.

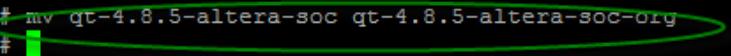
```
root@socfpga:~# cd /usr/local
root@socfpga:/usr/local#
```

**Figure 6-21 Go to /usr/local Directory in Altera SoC FPGA Board**

There is already a “qt-4.8.5-alteasoc” directory in /usr/local. We need to move(rename) that directory because we now need to decompress qt-4.8.5-altera-soc.tar.bz2. Please use the following command to rename “ qt-4.8.5-altera-soc “ directory name as “ qt-4.8.5-altera-soc-org ” as shown in **Figure 6-22**.

```
$ mv qt-4.8.5-altera-soc qt-4.8.5-altera-soc-org
```

```
root@socfpga:/usr/local# mv qt-4.8.5-altera-soc qt-4.8.5-altera-soc-org
root@socfpga:/usr/local#
```



**Figure 6-22 Rename Existing Qt Library Directory**

After renaming, we can now decompress the QT Library by typing the following command:

```
$ tar -jxv -f qt-4.8.5-altera-soc.tar.bz2
```

as shown in **Figure 6-23**.

```
root@socfpga:/usr/local# tar -jxv -f qt-4.8.5-altera-soc.tar.bz2
```



**Figure 6-23 Decompress QT Library**

## ■ 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 6-24**.

```
root@socfpga:/usr/local# cd /home/root
root@socfpga:~#
```



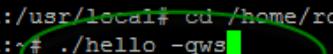
**Figure 6-24 Go to /home/root Directory**

Now launch hello program by typing in the following command:

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

as shown in **Figure 6-25**.

```
root@socfpga:/usr/local# cd /home/root
root@socfpga:~# ./hello -qws
```



**Figure 6-25 Launch Hello Program**

If the Hello program is launched successfully, you should see the Hello windows as shown in

**Figure 6-26.**



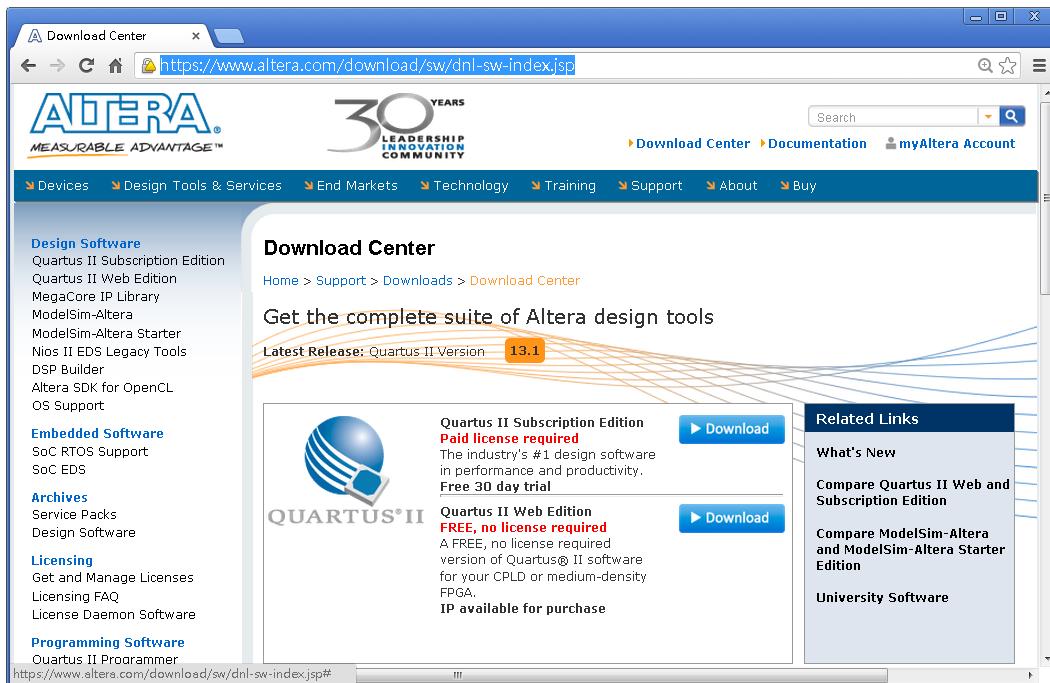
**Figure 6-26 Screenshot of the Hello Program**

## Chapter 7

# Control Panel Quartus Project

We have so far shown how to build and run hello program on the Altera SoC FPGA development board with QT library. In Chapter 7, we show how to build the Quartus project “Control Panel” over on the Windows host, translate the .sof to .rbf, and copy the .rbf file to the microSD Card such that the FPGA can be configured while booting Linux. We also introduce how to use **HPS** component in Qsys in order for the Linux program to access Qsys Avalon Memory-Mapped bus through **HPS** component.

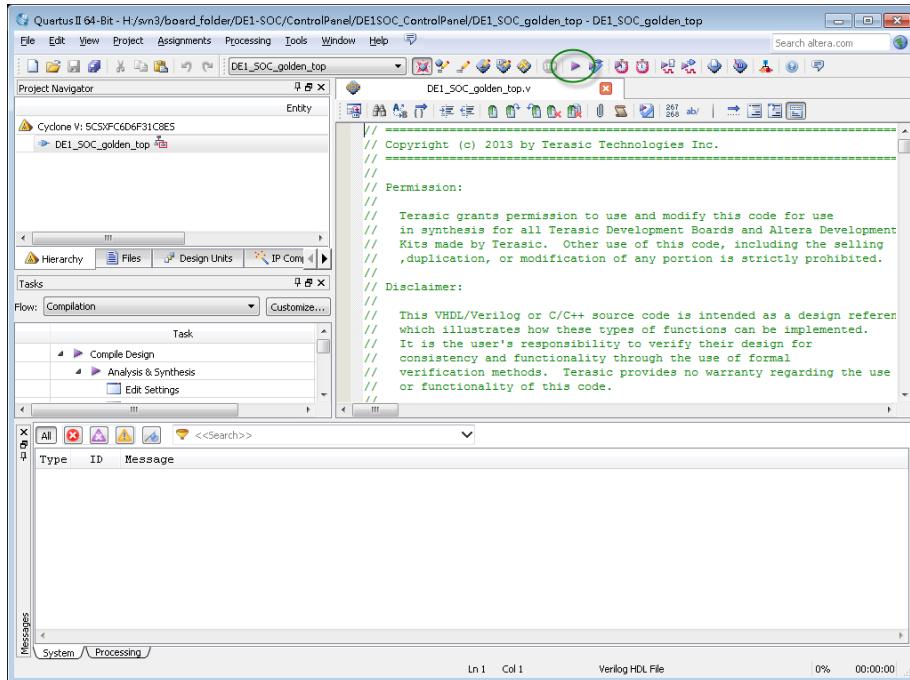
Quartus 13.0 or later which supports Cyclone V Device on Windows is required in the Control Panel Quartus Project can be downloaded from <https://www.altera.com/download/sw/dnl-sw-index.jsp> or as shown in **Figure 7-1**.



**Figure 7-1 Quartus Download Web Page**

## 7.1 Build Quartus Project of Control Panel

The Quartus project of Control Panel is located on the system CD: “**Demonstration\ControlPanel\Quartus**”. Please copy the folder to the Windows host and open it with Quartus, as shown in **Figure 7-2**. Click “Start Compilation” icon to start to compile.



**Figure 7-2 Quartus Project of the Control Panel**

When finished compiling, **DE1\_SOC\_golden\_top.sof** is generated under the Quartus project folder.

## 7.2 Test FPGA Configuration File

When Altera SoC FPGA boots from the microSD Card, it also reads **soc\_system.rbf** in the microSD Card to configure the FPGA. Therefore, we now show how to translate the .sof to .rbf.

### ■ Generate **soc\_system.rbf**

First, we need to translate the .sof to .rbf with the following command

```
$ quartus_cpf -c DE1_SOC_golden_top.sof soc_system.rbf
```

We have provided the batch file “sof\_to\_rbf.bat” which implements the translation command for user’s convenience. When the file translation is complete, **soc\_system.rbf** file is generated in the folder where the Quartus project is located.

## ■ Update microSD card

Next, we need copy the **soc\_system.rbf** to the microSD Card with the following procedures:

1. Insert the microSD Card in the Windows Host
2. Delete **soc\_system.rbf** in the microSD Card if there is one
3. Copy your **soc\_system.rbf** into the microSD Card

## ■ Test

Finally, boot the Altera DE1-SoC FPGA development board with the updated microSD Card, login to Linux, and launch control panel with the command “./ControlPanel ./qws” to test if everything works well.

# 7.3 More on the Quartus Project of Control Panel

The Control Panel circuit is built by QSyS. The HPS Linux program accesses **Avalon Memory Mapped Bus** through the **HPS** component. Also, FPGA controller can use HPS resource, like DDR3 SDRAM, through the **HPS** component. The HPS configuration is board-peripheral-dependent, so users don’t need to modify the HPS setting if the board has not been changed.

## ■ Linux Access FPGA Controller

The HPS Linux program accesses **Avalon Memory Mapped Slave** ports through HPS **AXI Master** port as shown **Figure 7-3**. If you wish that Linux program can access a Qsys instance, just connect its **Avalon Memory Mapped Slave** port of the controller to HPS **AXI Master** port.

Linux program access specific **Avalon Memory Mapped Slave** port based on its port address, so please make sure the port addresses are not overlapped with each other. We strongly recommend

that developer properly defines the port addresses and lock port addresses in Qsys.

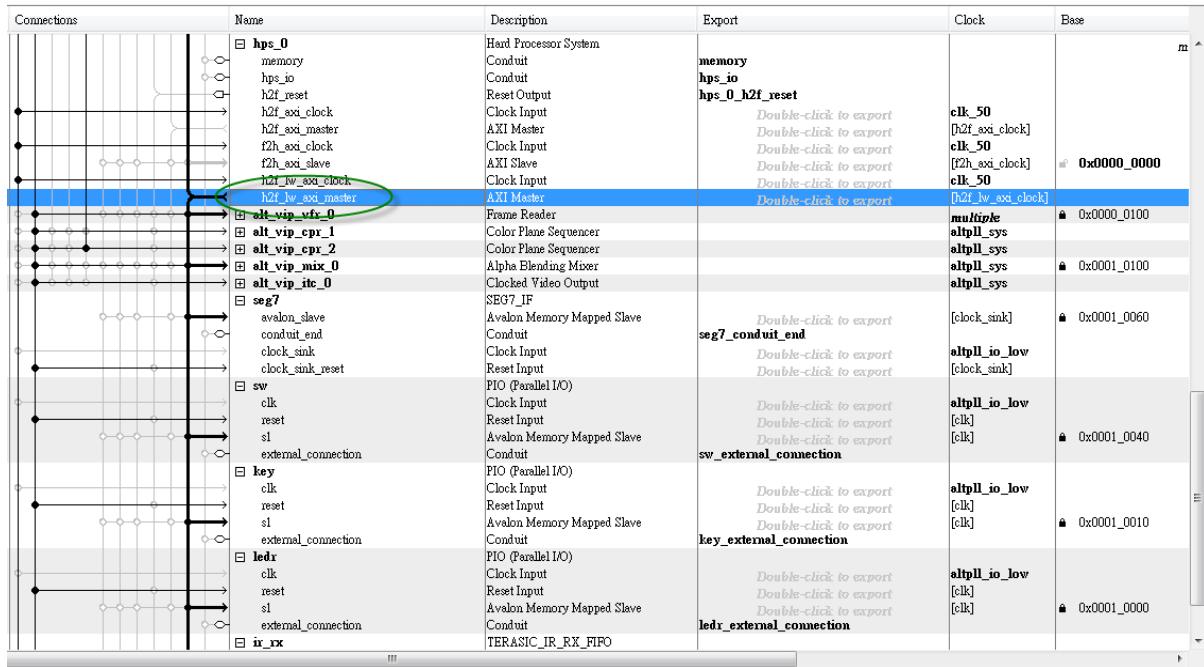


Figure 7-3 HPS AXI Master Port

In this tutorial, the slave port of LED, Button, Switch, IR, 7-segment, VIP frame reader, VIP clocked video input, and VIP alpha blending mixer controllers are connected to the **HPS h2f\_lw\_axi\_master AXI Master** port. Use Qsys “Address Map” tab can view the connection summary and associated address, as shown in [Figure 7-4](#).

System Contents		Address Map	Project Settings
alt_vip_vfr_0_avalon_slave	hps_0_h2f_lw_axi_master	0x0000_0100 - 0x0000_017f	
ledr.s1		0x0001_0000 - 0x0001_000f	
key.s1		0x0001_0010 - 0x0001_001f	
sw.s1		0x0001_0040 - 0x0001_004f	
seg7.avalon_slave		0x0001_0060 - 0x0001_007f	
alt_vip_ct_0_control		0x0001_0080 - 0x0001_00bf	
alt_vip_mx_0_control		0x0001_0100 - 0x0001_01ff	
ur_rx.avalon_slave		0x0001_0200 - 0x0001_0207	
timer.s1			
onchip_memory2.s1			
sram.s1			
i2c_scl.s1			
i2c_sda.s1			
jtag_uart.avalon_jtag_slave			
td_status.s1			
td_reset_n.s1			
clock_crossing_io_slow.s0			
cpu.jtag_debug_module			
sysid.control_slave			
spi_0.spi_control_port			
audio.avalon_slave			
uart.s1			
timer_stamp.s1			
mm_clock_crossing_bridge_1.s0			
hps_0_f2h_axi_slave		0x0000_0000 - 0xffff_ffff	
i2c_scl.s1 via clock_crossing_io_slow			
td_reset_n.s1 via clock_crossing_io_slow			
spi_0.spi_control_port via mm_clock_cr...			
sysid.control_slave via clock_crossing_i...			
td_status.s1 via clock_crossing_io_slow			
timer.s1 via clock_crossing_io_slow			
i2c_sda.s1 via clock_crossing_io_slow			

Figure 7-4 HPS h2f\_lw\_axi\_master Connection and Address Map

## ■ Hardware of Linux Frame Buffer Display

VIP Frame Reader is used to implement the Frame Buffer Display of Linux. The Frame Reader uses HPS's DDR3 SDRAM as frame buffer, so we need to connect the **Avalon Memory Mapped Master** port of Frame Reader instance to the **AXI Slave** port of HPS instance, as shown in **Figure 7-5.**

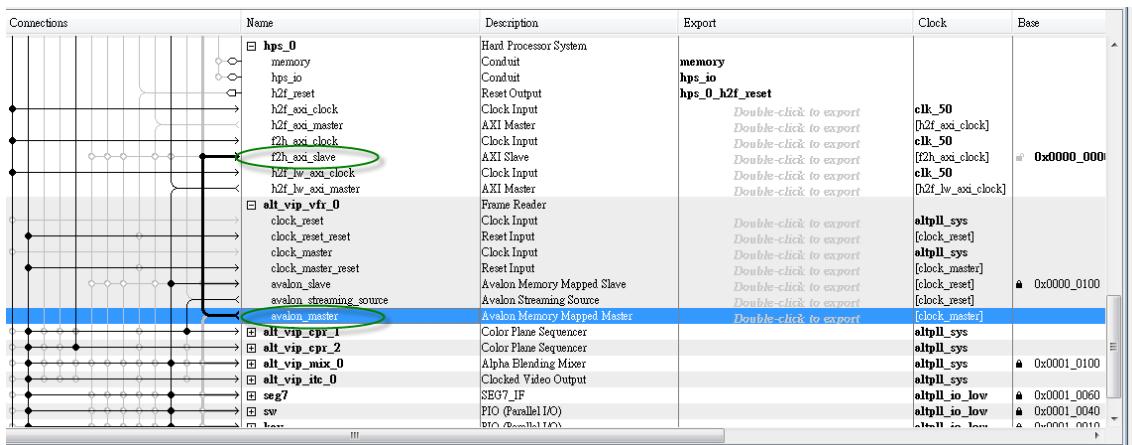


Figure 7-5 HPS AXI Master Port

Figure 7-6 shows Frame Reader instance setting. The display resolution is set as 1024x768.

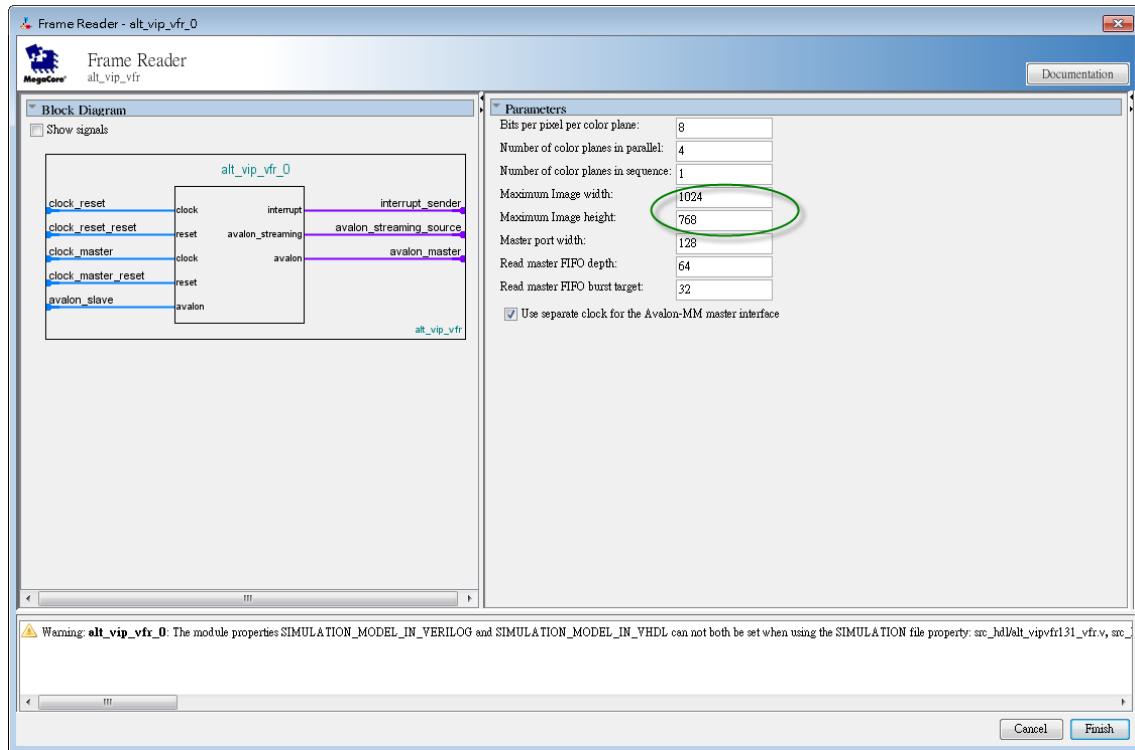


Figure 7-6 Frame Reader Setting

## Chapter 8

# Control Panel QT Project

Chapter 8 describes how to build the Control Panel QT project by the QT Creator on the Linux host. Here, we assume the QT library for Altera SoC ARM built in previous chapters is still in place. The **Shared Folder** feature of VMware Player is used so that the Control Panel QT project in the folder on the Windows host can be visible on the Linux host.

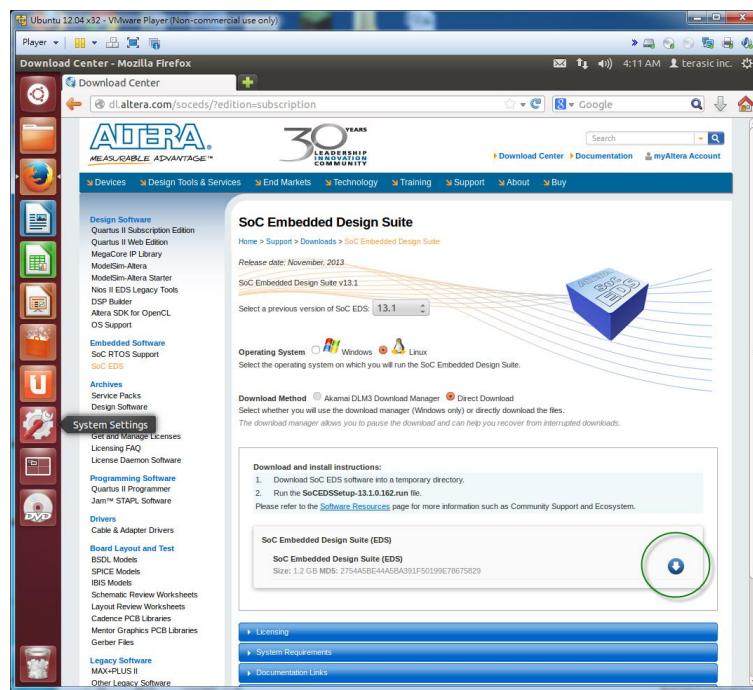
Control Panel QT project uses Altera **hwlib** library source code which is included in the Altera SoC EDS (Embedded Development Suite). Therefore make sure to install Altera SoC EDS before compiling the QT Project.

Altera SoC EDS can be downloaded from <https://www.altera.com/download/sw/dnl-sw-index.jsp> or the Altera Download Center by clicking on the **SoCEDS** icon as shown in **Figure 8-1**.



Figure 8-1 Altera SoC EDS Download Web Page

In the SoC Embedded Design Suite Download page, click the “Download” icon to download the SoC EDS installer as shown in **Figure 8-2**.



**Figure 8-2 Altera SoC EDS Download Web Page**

The downloaded SoC EDS installer “**SoCEDSSetup-13.1.0.162.run**” is located in the directory “**~/Downloads**”. To launch the installer, first launch Linux terminal (CTRL+ALT+T), and type in:

```
$ cd ~/Downloads
```

to go to “**~/Downloads**” directory.

Then, type in:

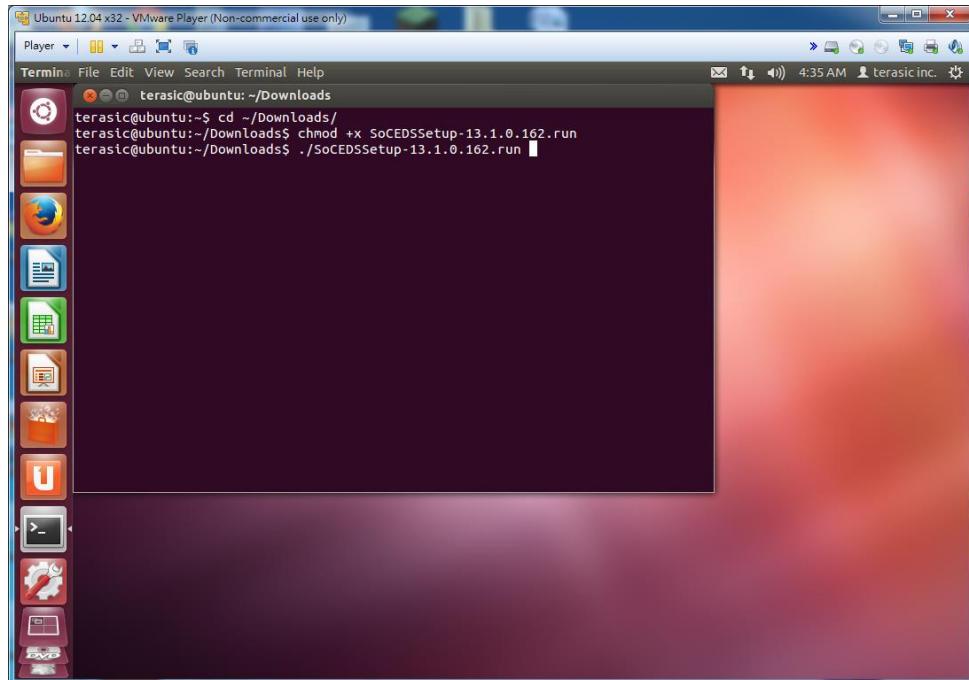
```
$ chmod +x SoCEDSSetup-13.1.0.162.run
```

to add execution attribute to the installer.

Finally, type in:

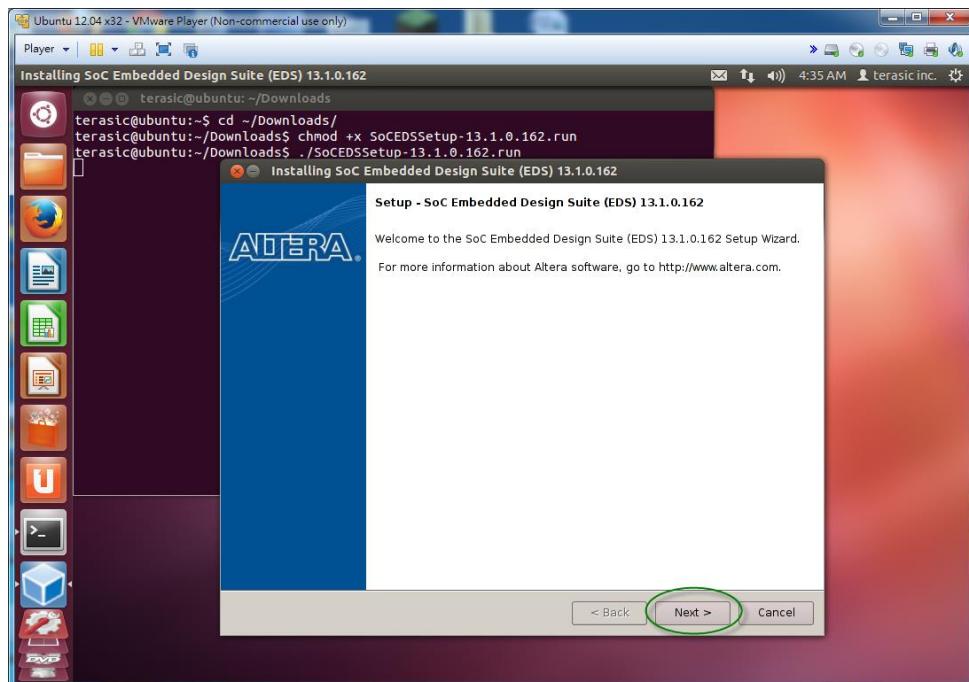
```
$ ./SoCEDSSetup-13.1.0.162.run
```

to launch SoC EDS installer as shown in **Figure 8-3**.



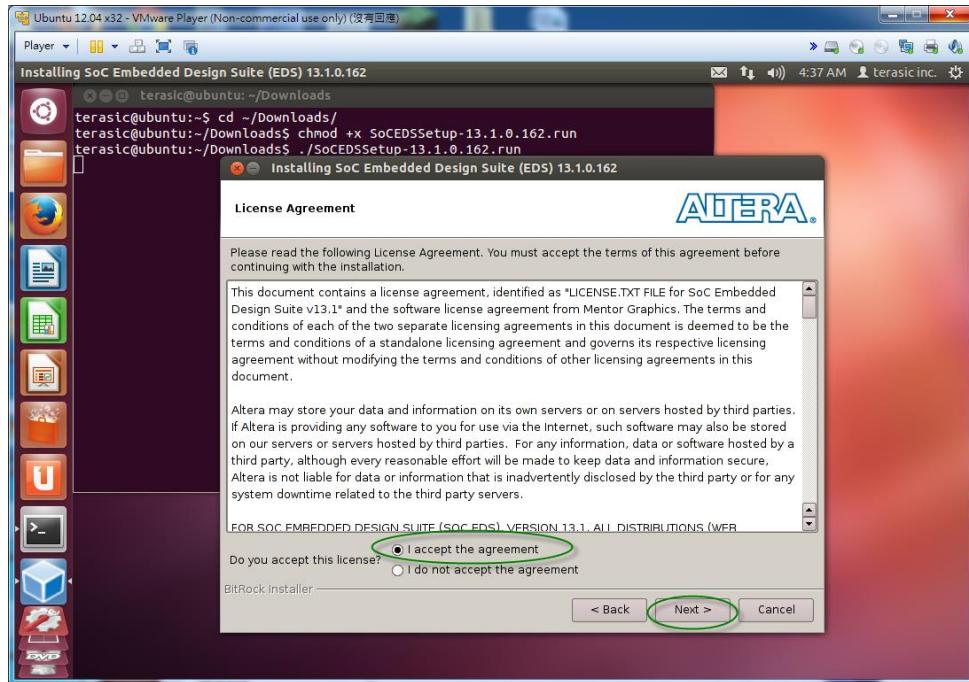
**Figure 8-3 Launch SoC EDS Installer**

When the installer is launched, a **Welcome** dialog appears as shown in **Figure 8-4**. Click “Next >” to go to the next step.



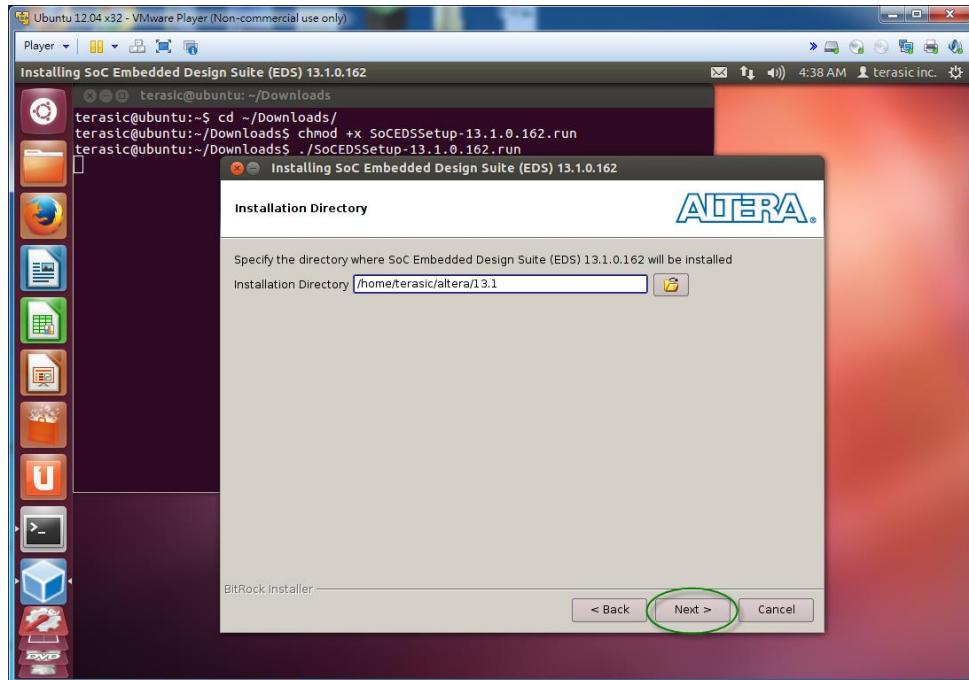
**Figure 8-4 Welcome Dialog of SoC EDS Installer**

When the **License Agreement** dialog appears as shown in **Figure 8-5**, please select the “I accept the agreement” radio button if you agree with the license and click “Next >” to go to the next step.



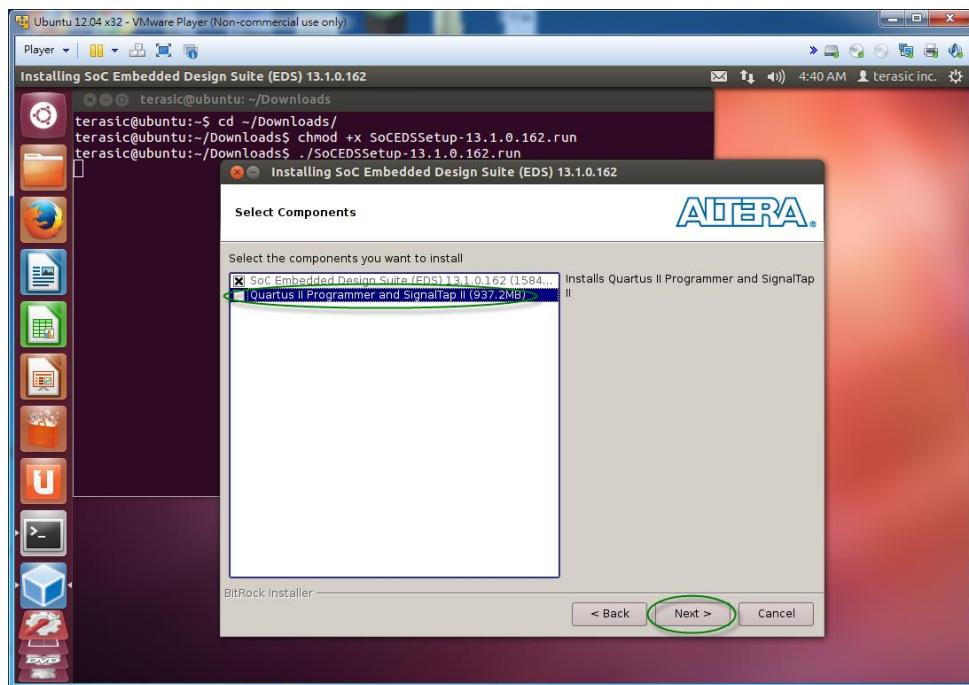
**Figure 8-5 License Agreement Dialog of SoC EDS Installer**

When the **Installation Directory** dialog appears as shown in **Figure 8-6**, we strongly recommend you keep the default directory and click “Next >” to proceed.



**Figure 8-6 Installation Directory Dialog of SoC EDS Installer**

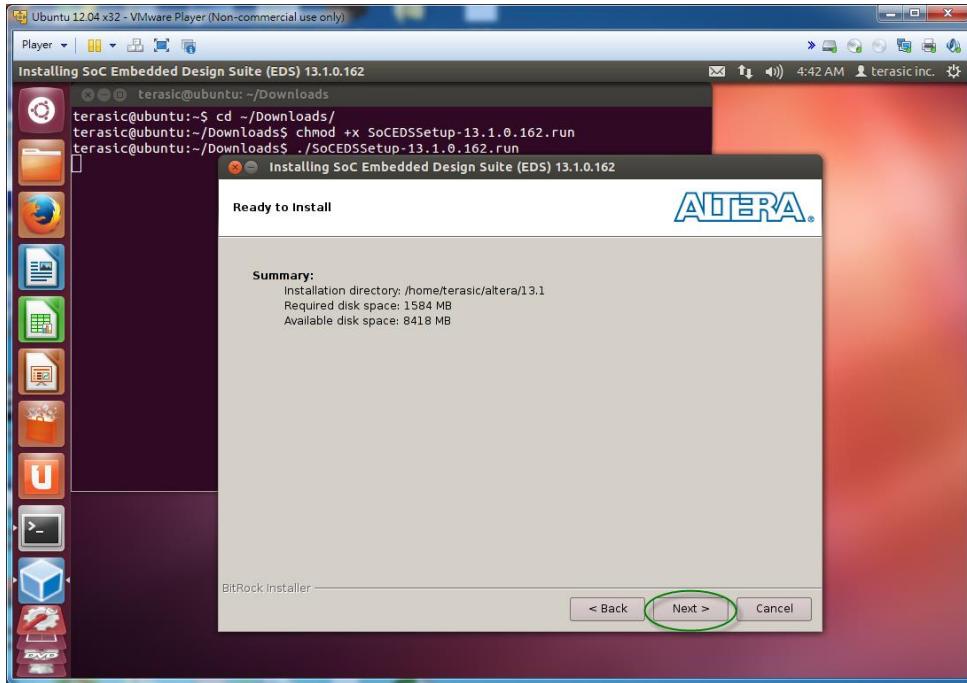
When the **Select Components** dialog appears as shown in **Figure 8-7**, please **uncheck** the “Quartus II Programmer and SignalTap II(937.2MB)” checkbox and click “**Next >**” to go to the next step.



**Figure 8-7 Select Components Dialog of SoC EDS Installer**

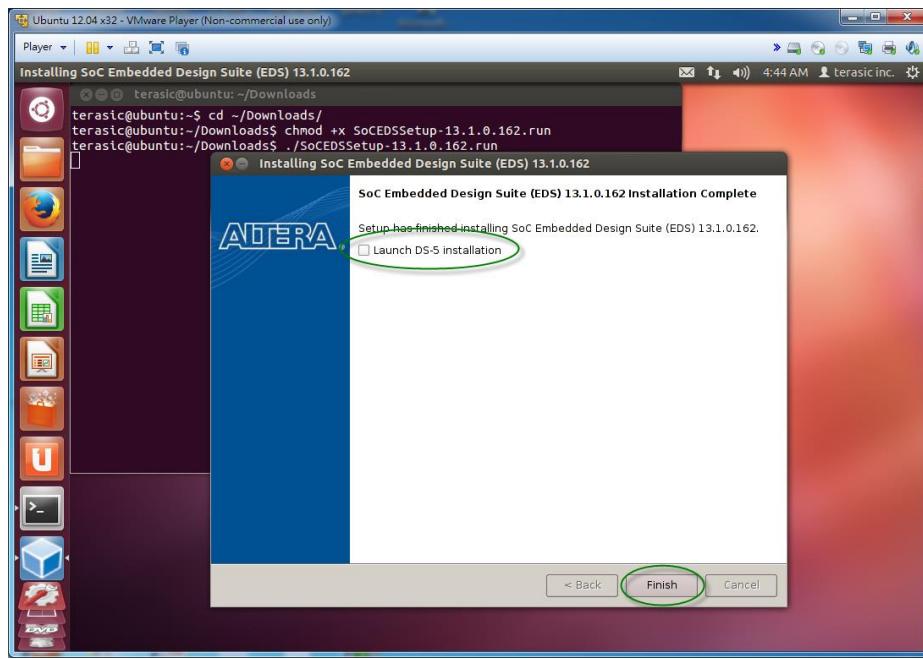
When the **Ready to Install** dialog appears as shown in **Figure 8-8**, please click “**Next >**” to go to

the next step.



**Figure 8-8 Ready to Install Dialog of SoC EDS Installer**

When the installation is complete, please uncheck the “Launch DS-5 Installation” checkbox and click “Finish” as shown in **Figure 8-9** to finish.



**Figure 8-9 Finish Dialog of SoC EDS Installer**

The source code of **hwlib** library can be found at the below location as shown in **Figure 8-10**.

/home/terasic/altera/13.1/embedded/ip/altera/hps/altera\_hps/hwlib

(Note, in the path string, the “terasic” should be replaced with your linux user name.)

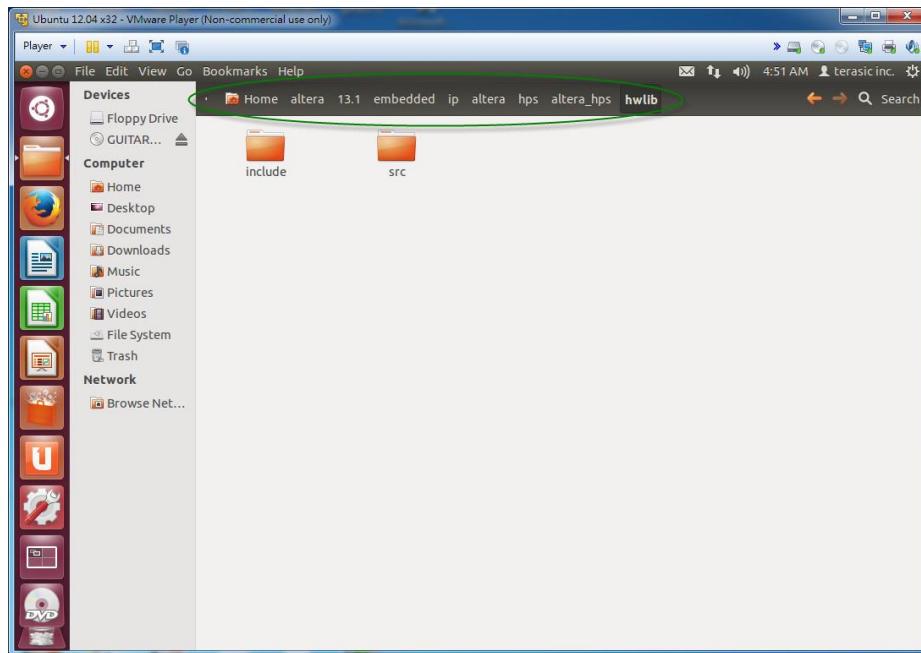


Figure 8-10 Directory Location of Altera ‘hwlib’ Library Source Code

## 8.1 Copy Control Panel QT Project

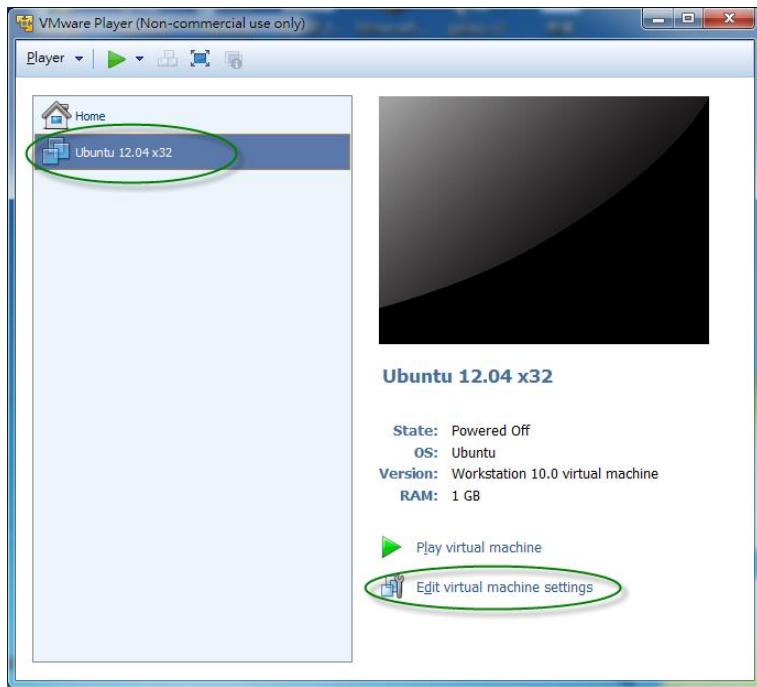
The Control Panel QT project is located on the System CD:

\Demonstration\ ControlPanel\ControlPanel\_QT

Here we show how to make the project visible under the Linux host by using VMware Player’s **Shared Folder** features.

### ■ Shared Folder Setup

Launch the VMware Player, select the “Ubuntu 12.04 x32”, and click “Edit virtual machine settings”, as shown in **Figure 8-11**.



**Figure 8-11 Launch Edit Virtual Machine Settings**

When the **Virtual Machine Settings** dialog appears as shown in [Figure 8-12](#), please select “Options” tab, click “Shared Folders” icons, check “Always enabled” radio button, and then click “Add...”.

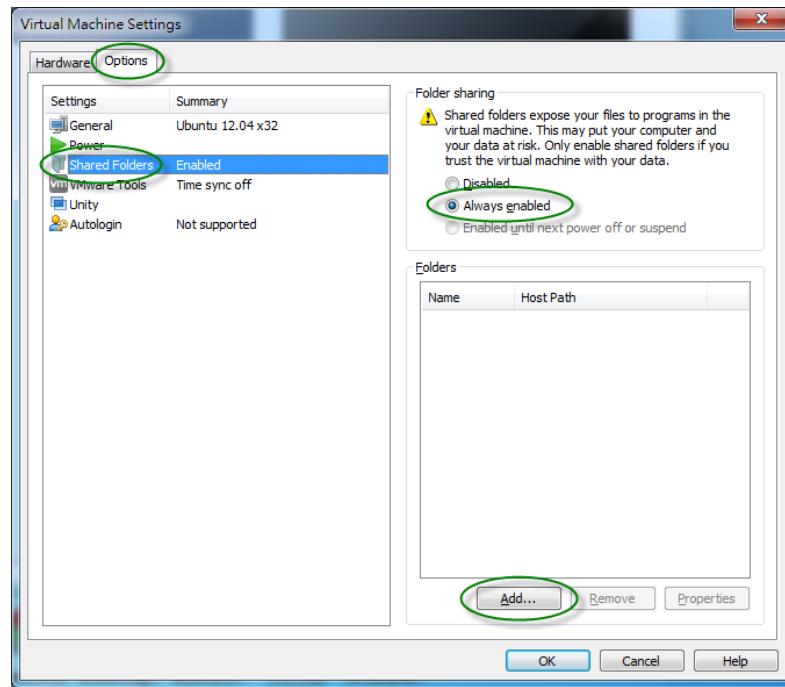


Figure 8-12 Add Shared Folders

**Add Shared Folder Wizard** is launched for shared folder settings. A Welcome dialog appears as shown in **Figure 8-13**, please click “Next >” to go to the next step.

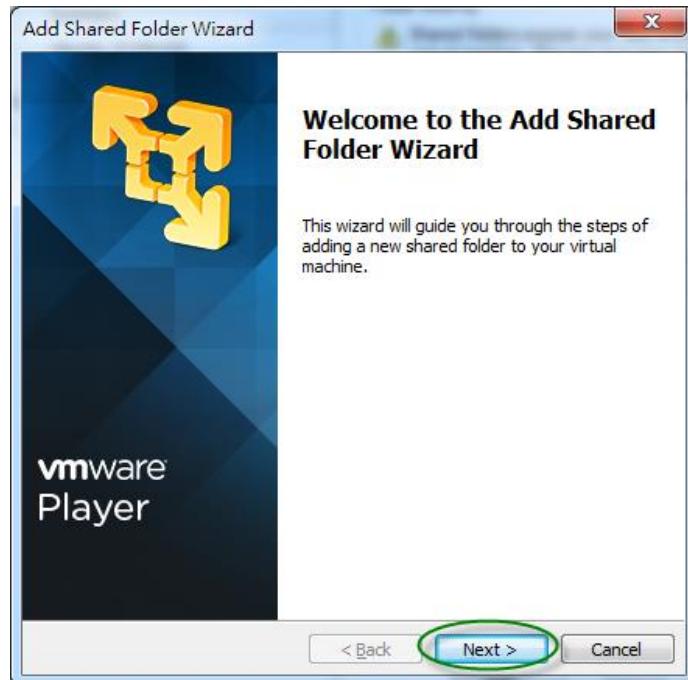
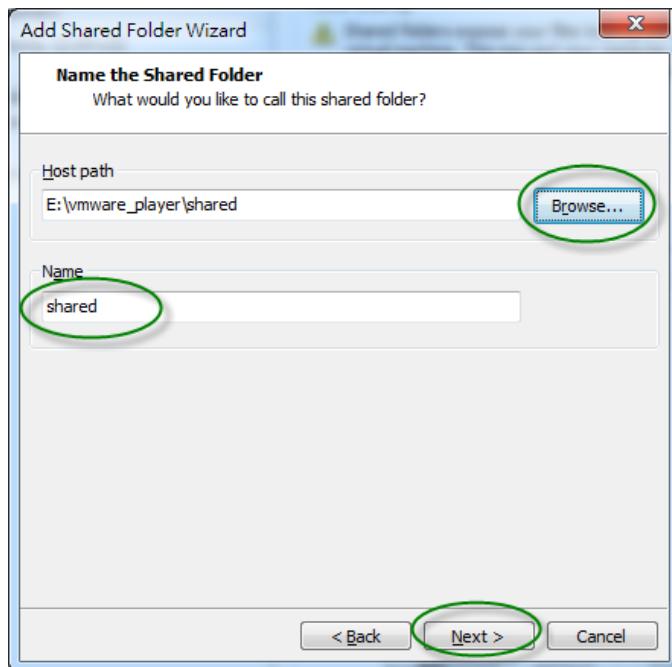


Figure 8-13 Welcome Dialog of Add Shared Folder Wizard

When the **Name the Shared Folder** dialog appears as shown in [Figure 8-14](#), click “Browse...” to select a folder as the shared folder, give a name to the shared folder(in this tutorial, **shared** is used ), and click “Next >” to proceed.



**Figure 8-14 Specify Shared Folder Location and Name**

When the **Specify Shared Folder Attributes** dialog appears as shown in [Figure 8-15](#), check “Enable this share” checkbox and click “Finish”.

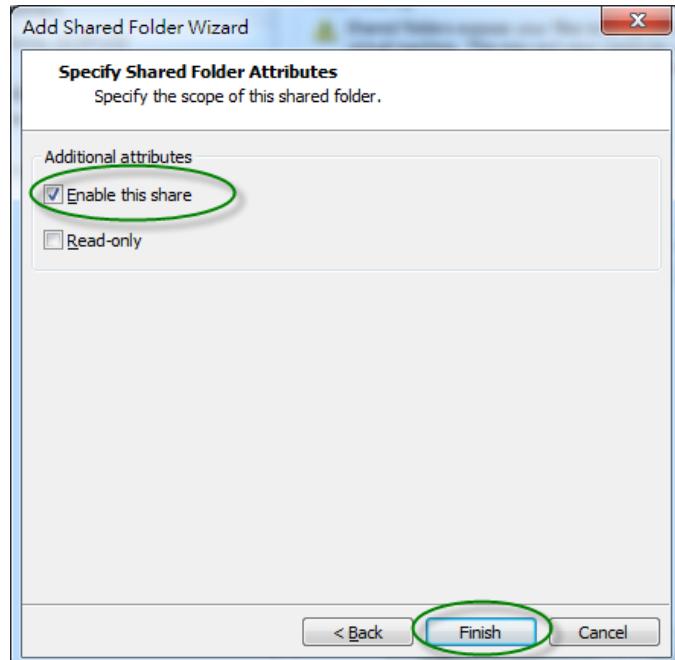


Figure 8-15 Finish Shared Folder Setup

When going back to the **Virtual Machine Settings** dialog as shown in **Figure 8-16**, check “OK” to complete the settings. Now the Shared Folder feature is applied.

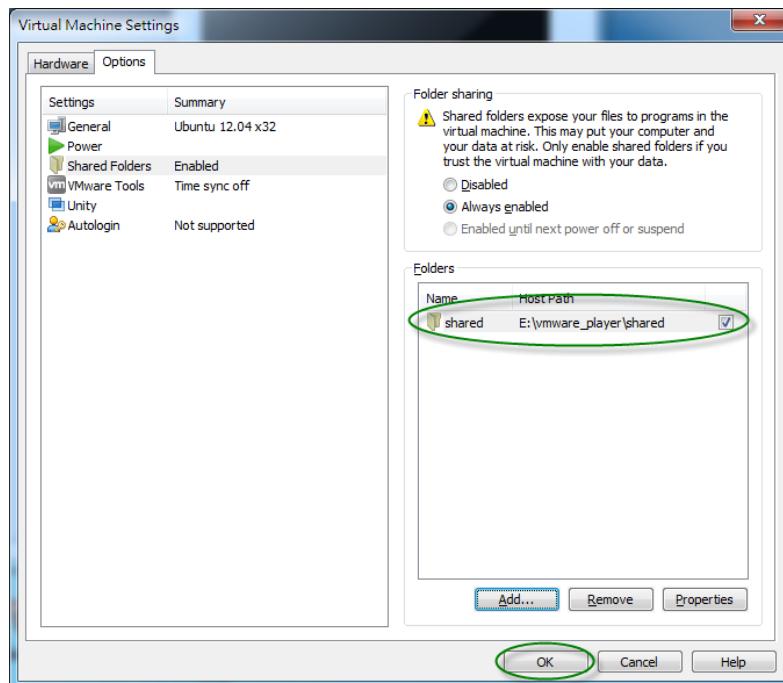


Figure 8-16 Complete Virtual Machine Setting

## ■ Copy Control Panel QT Project to Shared Folder

Copy the Control Panel QT project folder “ControlPanel\_QT” from the System CD to the shared folder on Windows host. Then, launch the “Ubuntu 12.04 x32” virtual machine. After logging in Ubuntu, launch a terminal (CTRL+ALT+T) and type the following command to go to the shared folder on Linux:

```
$cd /mnt/hgfs/shared
```

to go to the shared directory on Linux.

Now type in

```
$ls
```

To see the “ControlPanel\_QT” directory of Control Panel QT Project, as shown in [Figure 8-17](#).

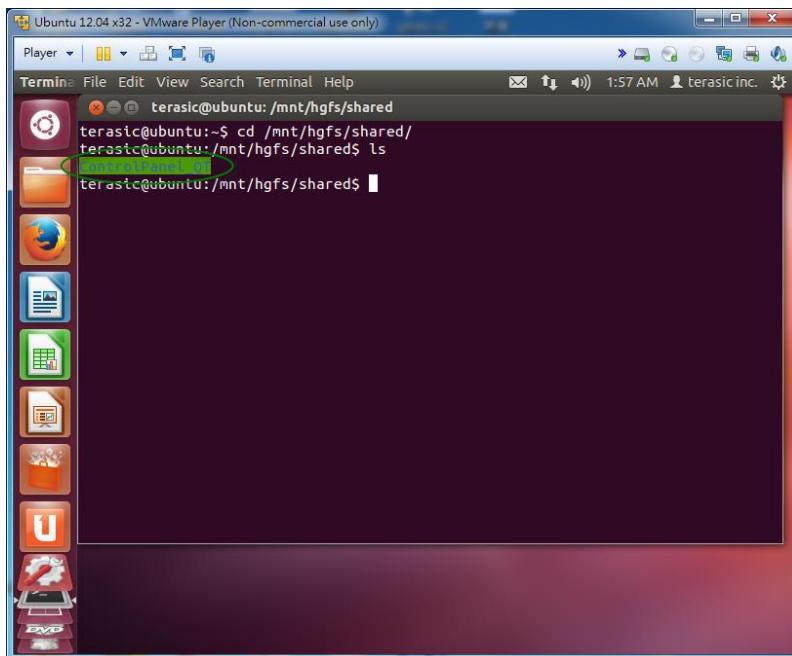
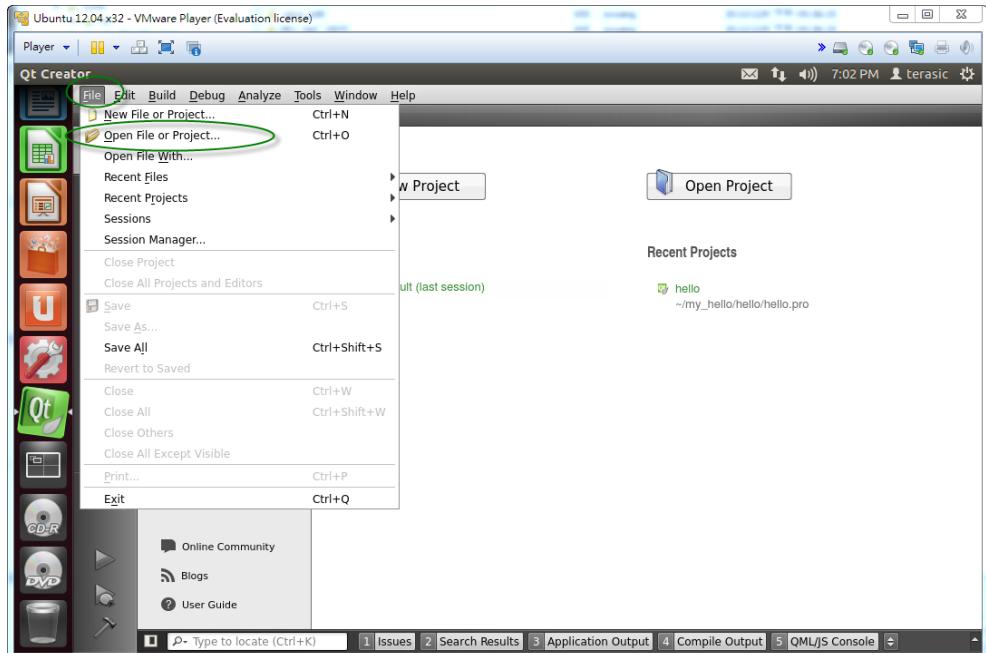


Figure 8-17 ‘ControlPanel\_QT’ Directory of Control Panel QT Project

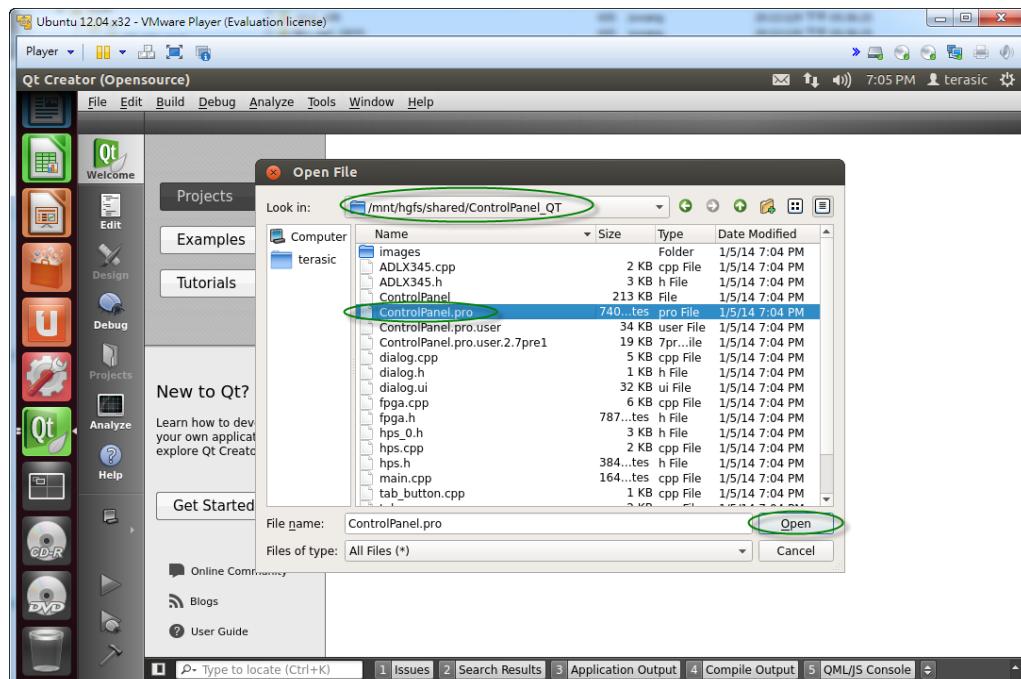
## 8.2 Build Control Panel QT Project

Now, we can launch the QT Creator, and select the menu item “Files→Open File or Project...” as shown in [Figure 8-18](#).



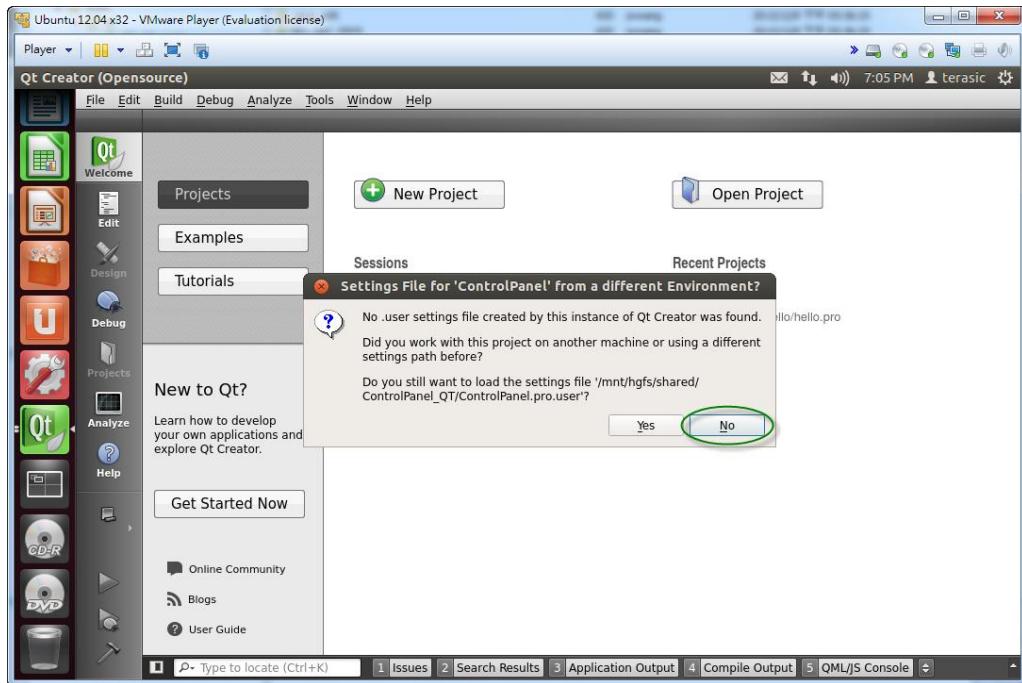
**Figure 8-18 Launch Open Project Dialog**

In the **Open File** dialog, go to the directory “/mnt/hgfs/shared/ControlPanel\_QT”, select “ControlPanel.pro”, and click “Open” as shown in **Figure 8-19**.



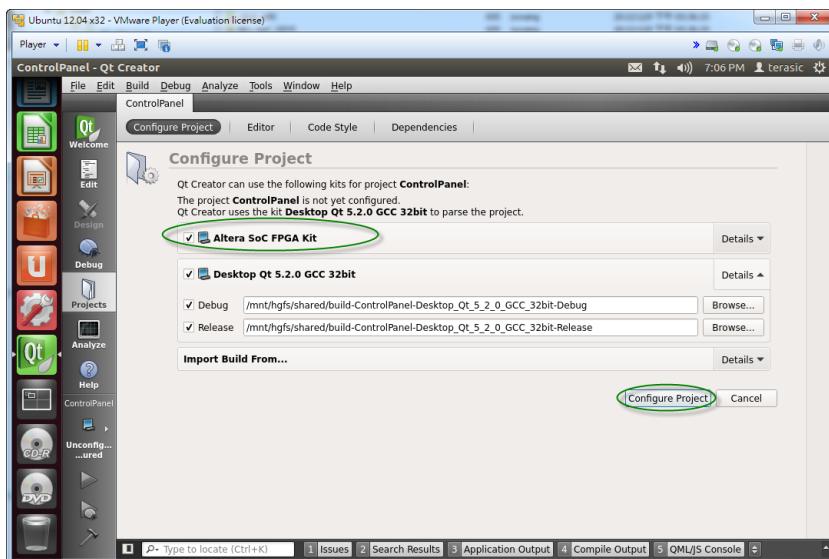
**Figure 8-19 Select Control Panel Project – ControlPanel.pro**

If the **Settings File** dialog appears as shown in **Figure 8-20**, please click “No” to proceed.



**Figure 8-20 Query .user Setting**

If Configure Project appears as shown in **Figure 8-21**, please check “Altera SoC FPGA Kit” and click “Configure Project”.



**Figure 8-21 Configure Project**

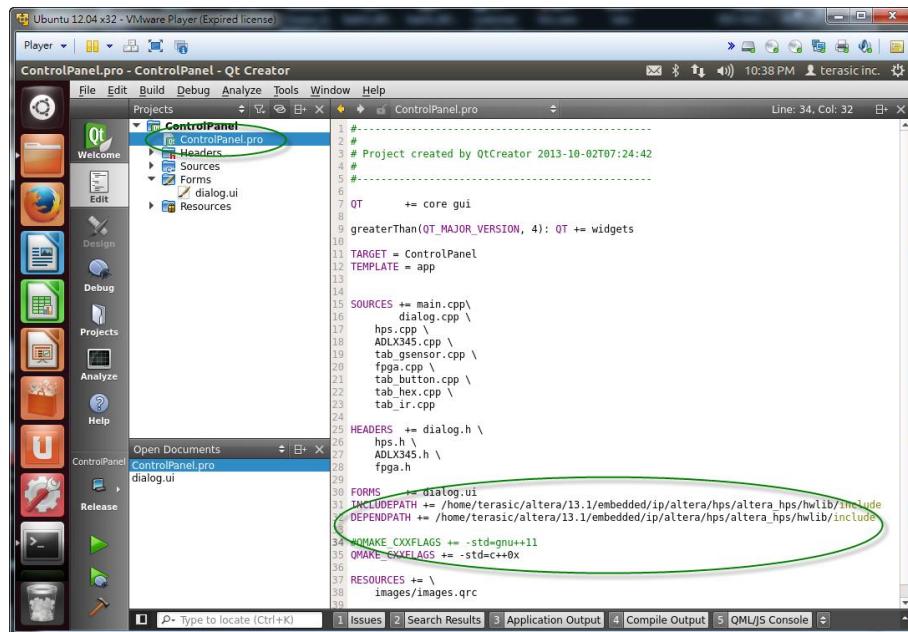
Now, to check if the include path is correct, please follow the instructions below:

First click on “Edit” icon on the left and double click “ControlPanel.pro” as shown in **Figure 8-22**, please make sure the **INCLUDEPATH** includes the correct path:

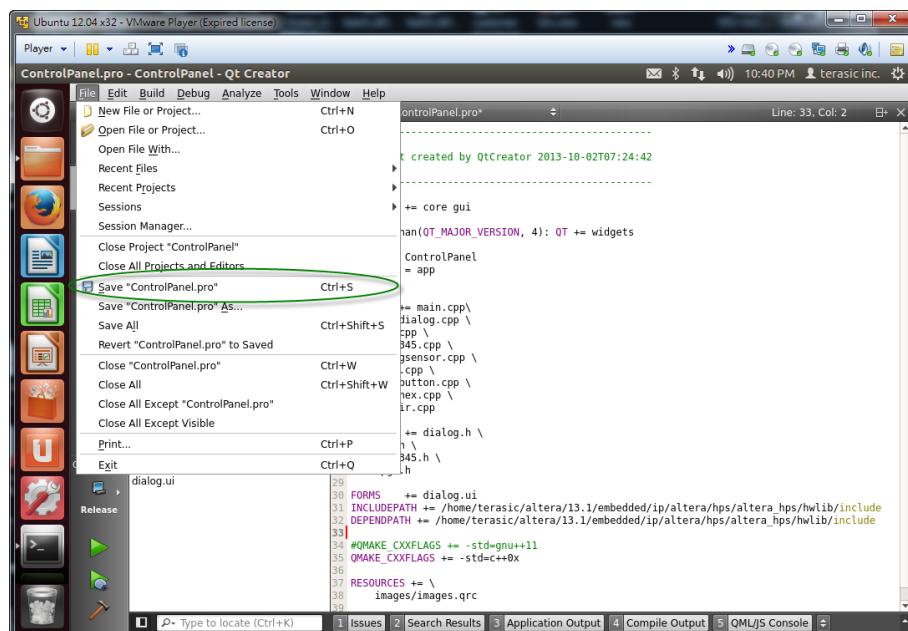
/home/terasic/altera/13.1/embedded/ip/altera/hps/altera\_hsp/hplib/include

(Note, in the path string, the “terasic” should be replaced with your linux user name.)

If the file “ControlPanel.pro” is modified, please correct the path and select the menu item “File→Save “ControlPanel.pro” as shown in **Figure 8-23**.

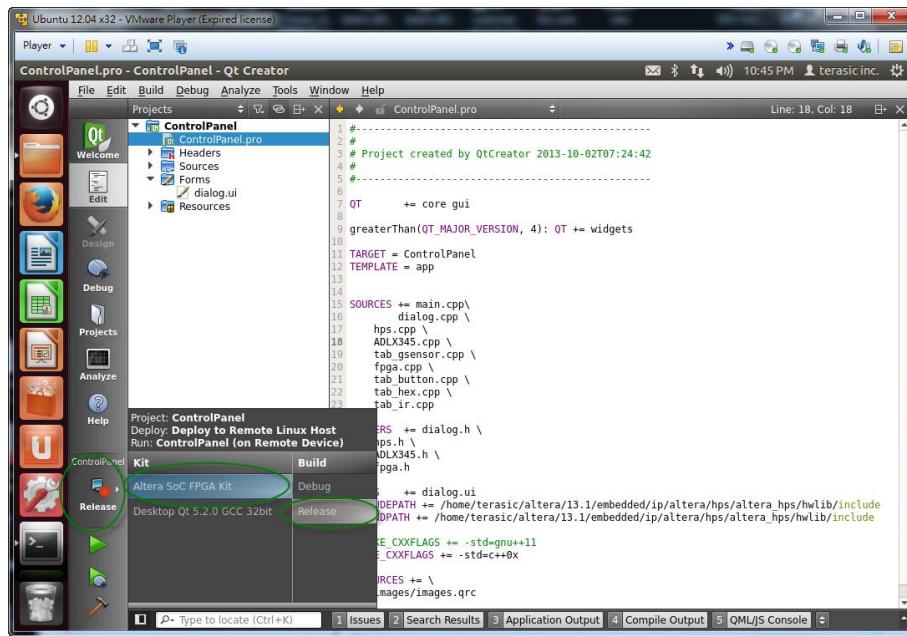


**Figure 8-22 Check Include Path in ControlPanel.pro**



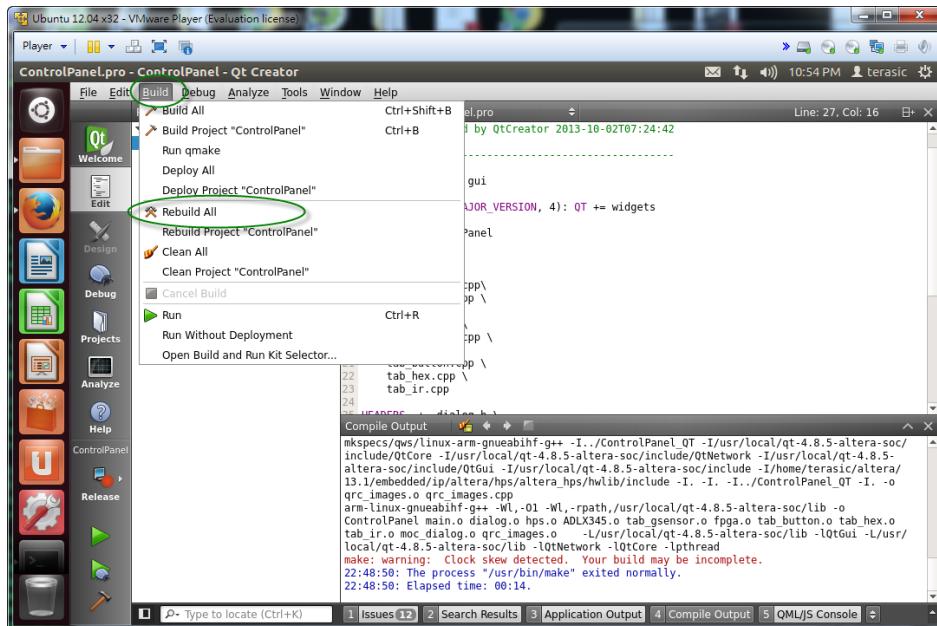
**Figure 8-23 Save ControlPanel.pro**

Click on the “Release” icon, and select “Altera Soc FPGA Kit” and “Release” Build as shown in **Figure 8-24**.



**Figure 8-24 Setting Release Kit and Build**

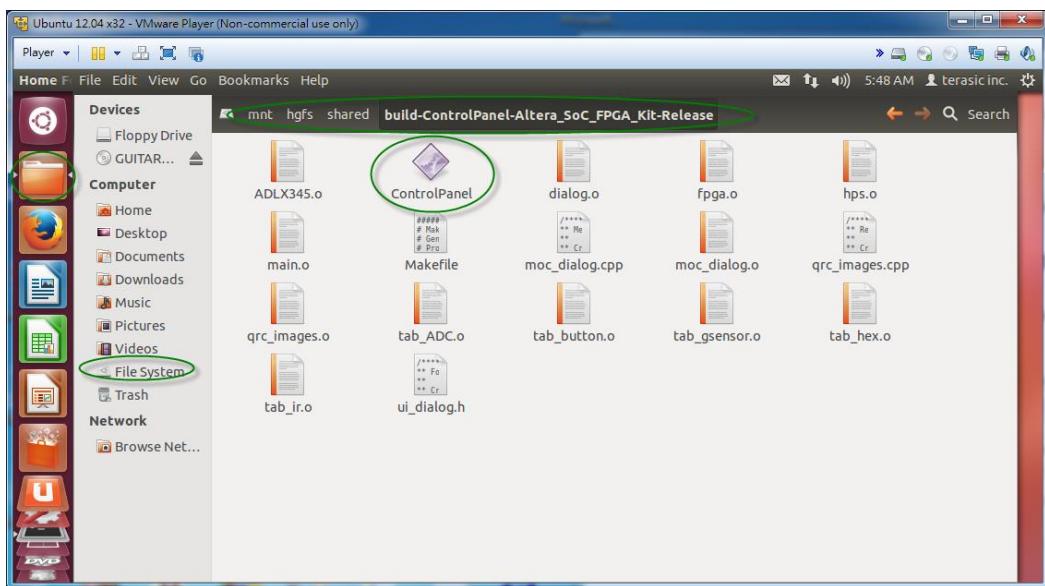
Then, select menu item “Build→Rebuild All”, as shown in **Figure 8-25**, to build the Control Panel Project.



**Figure 8-25 Build Control Panel**

If the build is successful, the ControlPanel execution file is generated in the below folder as shown in **Figure 8-26**.

```
/mnt/hgfs/shared/build-ControlPanel-Altera_SoC_FPGA_Kit-Release
```



**Figure 8-26 Generated Control Panel Execution File**

## 8.3 Execute Control Panel Program

To run the “ControlPanel” execution file on the Altera DE1-SoC FPGA development board, we need to copy it to the Linux on Altera SoC FPGA Board. This is similar to what we have done in Section 6.3 “Execute Hello Program” where we use Linux “scp” command to remote copy.

After “ControlPanel” execution file has been remotely copied to the “/home/root” directory of Linux running on Altera SoC FPGA Board, type “./ControlPanel –qws” to launch the Control Panel in the terminal of Altera SoC FPGA board. If you can see the ControlPanel windows as shown in **Figure 8-27**, it means you had successfully built the Control Panel QT Project.

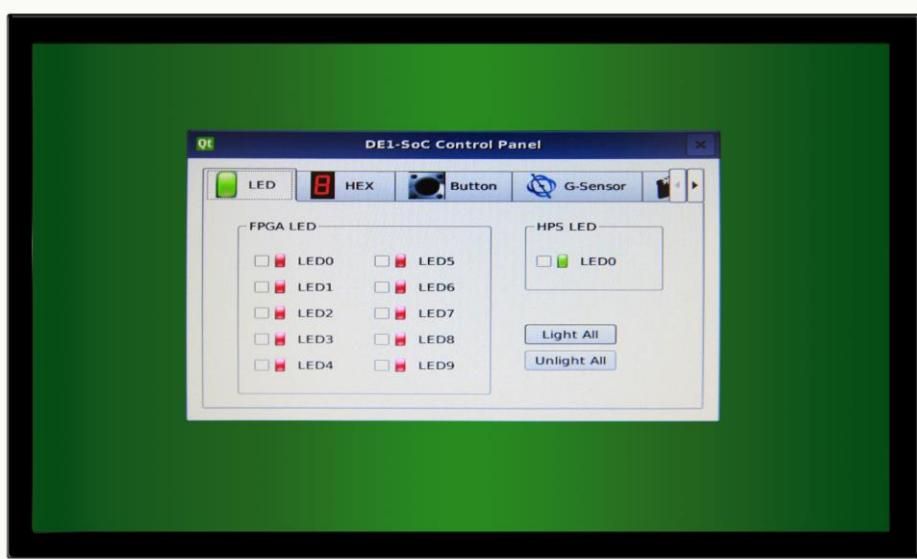


Figure 8-27 Screenshot of Control Panel

## 8.4 More on the Control Module of Control Panel

### ■ Control FPGA LED

Control Panel controls the FPGA Qsys components through the memory-mapped method. The device driver “/dev/mem” is used to access the physical address space which is mapped to Qsys address space.

Below shows how to turn on 10 LEDs which is controlled by Qsys PIO Controller whose address is 0x10000 in the Qsys system. Based on “/dev/mem” device driver and “mmap” function, we can calculate the LED address **led\_base**. With the address, we can directly write the value of the address to control the PIO Controller. Const ALT\_LWFPGASLVS\_OFST is defined “socal/hsp.h”

```
#include "socal/hps.h"

#define HW_REGS_BASE (ALT_STM_OFST)
#define HW_REGS_SPAN ( 0x04000000 )
#define HW_REGS_MASK ( HW_REGS_SPAN - 1 )
```

```
// Controller Base Address in QSYS
#define FPGA_LED_PIO_BASE 0x10000

int file
void *virtual_base;
uint8_t *led_base;

file = open( "/dev/mem", ( O_RDWR | O_SYNC ) );
virtual_base = mmap( NULL, HW_REGS_SPAN, ( PROT_READ | PROT_WRITE ), MAP_SHARED,
file, HW_REGS_BASE );
led_base= (uint8_t *)virtual_base + ( ( unsigned long )( ALT_LWFPGASLVS_OFST +
FPGA_LED_PIO_BASE ) & ( unsigned long )( HW_REGS_MASK ) );

*(uint32_t *)m_led_base = 0x3FF; // turn on 10 led
```

## ■ Control HPS LED

The way to control HPS LEDs is similar to the way to control FPGA LED. The register files of both PIO controllers are different. Macro alt\_setbits\_word is used to set value to the specific address. The HPS LED is controlled by GPIO1 GPIO Controller. ConstALT\_GPIO1\_SWPORTA\_DDR\_A DDR is used to define the direction register address of GPIO1 Controller, and const USER\_IO\_DIR is used to define the direction pin bits-mask associated to the HPS LED. Const ALT\_GPIO1\_SWPORTA\_DR\_ADDR defines the data register address of GPIO1 Controller, and const BIT\_LED is used to define the pin bits-mask associated to the HPS LED. Both ALT\_GPIO1\_SWPORTA\_DDR\_ADDR and ALT\_GPIO1\_SWPORTA\_DR\_ADDR are defined in “socal\hps.h”. Macro alt\_setbits\_word is defined in “socal\socal.h”

```
#include "socal/socal.h"
#include "socal/hps.h"

#define HW_REGS_BASE (ALT_STM_OFST)
#define HW_REGS_SPAN (0x04000000)
#define HW_REGS_MASK (HW_REGS_SPAN - 1)

#define USER_IO_DIR (0x01000000)
```

```
#define BIT_LED      (0x01000000)

int file
void *virtual_base;
uint8_t *led_base;

file = open( "/dev/mem", ( O_RDWR | O_SYNC ) );
virtual_base = mmap( NULL, HW_REGS_SPAN, ( PROT_READ | PROT_WRITE ), MAP_SHARED,
file, HW_REGS_BASE );
// configure LED as output pin
alt_setbits_word((void *) ( (char *)virtual_base +
( ( uint32_t )( ALT_GPIO1_SWPORTA_DDR_ADDR ) & ( uint32_t )( HW_REGS_MASK ) ),
USER_IO_DIR );
// turn on LED
alt_setbits_word((void *) ( (char *)virtual_base + ( ( uint32_t )( ALT_GPIO1_SWPORTA_DR_ADDR )
& ( uint32_t )( HW_REGS_MASK ) ) ), BIT_LED );
```

## ■ FPGA Class

All of FPGA control functions are encapsulated in the **FPGA** class and it is implemented in fpga.c and fpga.h. Below shows the public functions of this class:

```
bool LedSet(int mask);
bool HexSet(int index, int value);
bool KeyRead(uint32_t *mask);
bool SwitchRead(uint32_t *mask);
bool AdcRead(uint16_t szValue[8]);
bool VideoEnable(bool bEnable);
bool VideoMove(int x, int y);
bool IsVideoEnabled();
bool IrDataRead(uint32_t *scan_code);
bool IrlsDataReady(void);
```

## ■ HPS Class

All of HPS control functions are encapsulated in the **HPS** class and it is implemented in hps.cpp and hps.h. Below shows the public functions of this class:

```
bool LedSet(bool bOn);  
bool IsButtonPressed();  
bool GsensorQuery(int16_t *X, int16_t *Y, int16_t *Z);
```