DE10-Standard

Bluetooth SPP

terasIC
www.terasic.com

Content

1. Overview 1

2. Setup Demonstration 3

3. Linux Application Project on DE10-Standard 6

4. Android Application Project 8

5. Appendix 13

1. Overview

This documents describe how to an Android Smart Phone can remotely control the LEDs on the DE10-Standard board. The Bluetooth SPP (Serial Port Profile) is used as communication protocol between Smart Phone and DE10-Standard. Classical USB Bluetooth dongle is used to expand DE10-Standard Bluetooth capability.

■ System Block Diagram

Figure 1 System Block Diagram shows the system block diagram of this demonstration. In the demonstration, we implement two Bluetooth application software. One is running on Android and one is running on DE10-Standard. These two application communicate with each other based on a propriety command set.

In DE10-Standard, the Bluetooth application is running on the Linux. A classical USB Bluetooth dongle is used to expand DE10-Standard Bluetooth capability. The SPP (Serial Port Profile) protocol is used as communication protocol. The BlueZ Bluetooth protocol software stack is used by the application to handle the Bluetooth protocol. The Linux BSP build in the BlueZ and Bluetooth USB dongle driver, so user don't need install to install any library or driver in this demonstration.

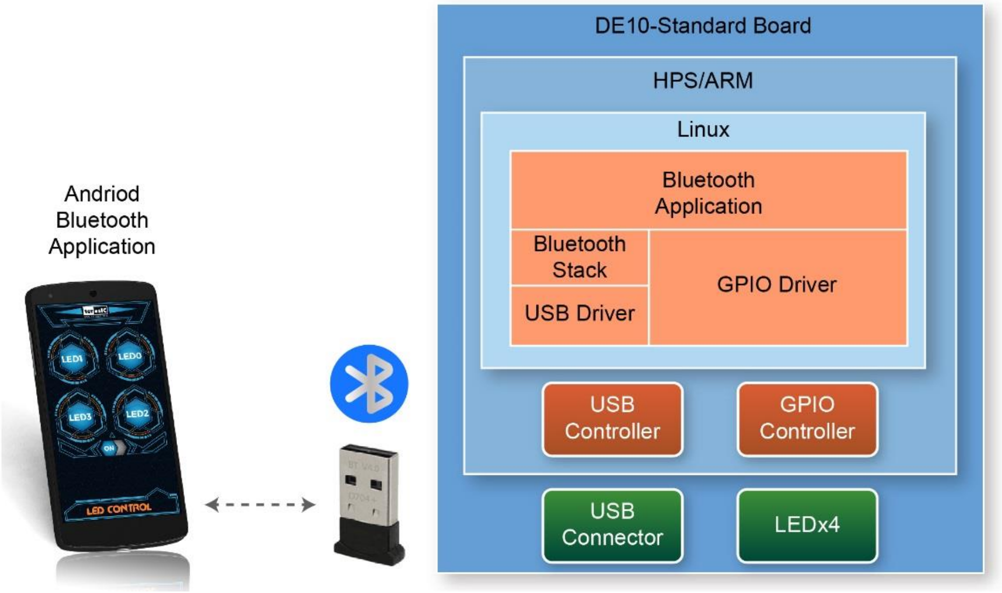


Figure 1 System Block Diagram

■ Command Set

There two application communicates based on the following command set defined by Terasic. The commands are send by from Android Smart Phone to the DE10-Standard.

Command	Description
“ATLED0\n”	Turn one LED0 only
“ATLED1\n”	Turn one LED1 only
“ATLED2\n”	Turn one LED2 only
“ATLED3\n”	Turn one LED3 only
“ATON\n”	Turn on all four LEDs.
“ATOFF\n”	Turn off all four LEDs.

2. Setup Demonstration

This section describe how to setup the Bluetooth remote control demonstrations.

Hardware Requirements

The following hardware items are required to perform this demonstration:

- DE10-Standard Board with shipped microSD card
- Classical Bluetooth USB dongle
- USB Mouse
- USB hub with two ports at least.
- Android Smart Phone with Bluetooth supporting
- A microSD card writer is required if reprogramming the microSD card is necessary.

The microSD card came with this kit already include the Linux application in the /home/root directory. If the microSD content is changed, please program the microSD card with the Linux Image File locate at http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1046&PartNo=5.

Execute Demonstration

Here are show the procedure to setup the demonstration. First, the Bluetooth Application should be launched on DE10-Standard. Then, launch the Bluetooth Application on the Android phone. In the first time, the Android need to discovery the DE10-Standard and pair with it. When Android phone connects with the DE10-Standard, users can controls the four LEDs on DE10-Standard.

Here are the procedure to launch the Bluetooth Application on the DE10-Standard:

1. Connect the HDMI port of DE10-Standard to a monitor as shown in **Figure 2**.
2. Connect a USB hub into the micro USB port of DE10-Standard.
3. Plug the USB mouse to the USB hub.
4. Plug a Bluetooth USB dongle into USB hub.
5. Insert the microSD card, came with this kit, into the DE10-Standard.
6. Make sure MSEL[4:0] is 01010.
7. Power on DE10-Standard.
8. When the LXDE desktop appears on the monitor, use a mouse to double click the BT_LED_AP icon to launch the Bluetooth Application as shown in **Figure 3**.

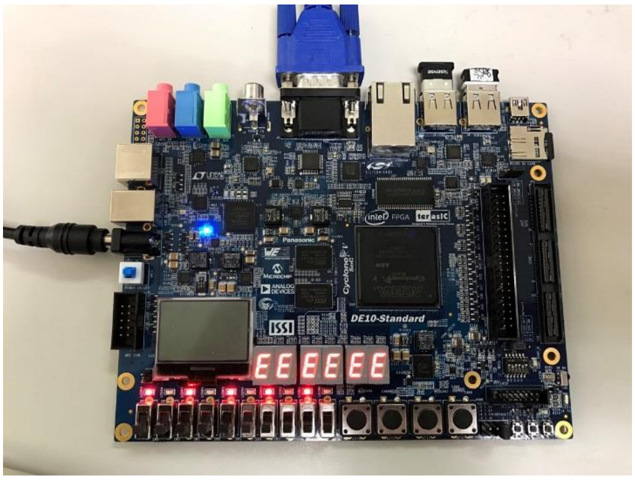
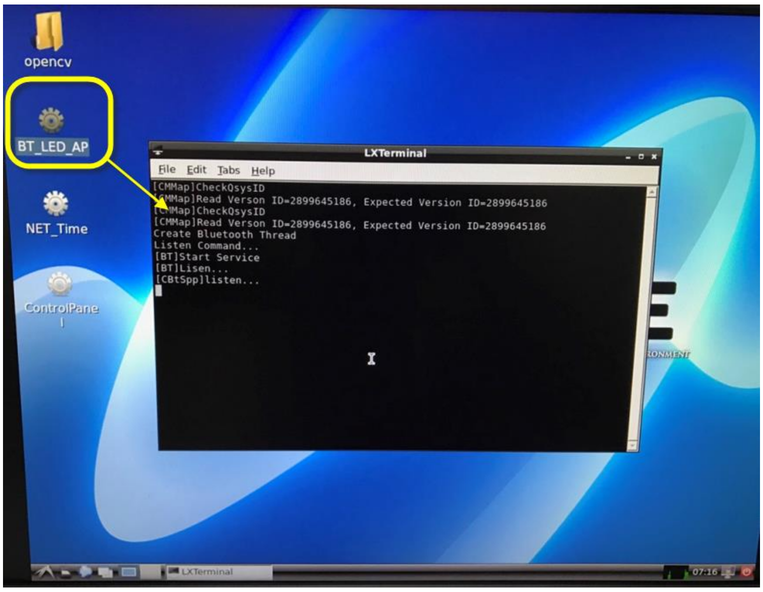


Figure 2 DE10-Standard Demo Setup



```
[CMMap]CheckQsysID
[CMMap]Read Verson ID=2899645186, Expected Version ID=2899645186
[CMMap]CheckQsysID
[CMMap]Read Verson ID=2899645186, Expected Version ID=2899645186
Create Bluetooth Thread
Listen Command...
[BT]Start Service
[BT]Lisen...
[CBtSpp]listen...
```

Figure 3 Launch BT_LED_AP on LXDE

Here are the procedure to setup the demonstration on Android Smart Phone:

1. Download the TerasicBluetooth App from the QR code shown **Figure 4**.
2. Install TerasicBluetooth.
3. Launch TerasicBluetooth.
4. For first time to connect DE10-Standard, click the ZOOM icon to discovery nearby devices, select DE10_Nano device, and pair it with password “1234” as shown in **Figure 5**.
5. Click the ZOOM icon and connect the paired DE10_Nano device.
6. In the TerasicBlueooth App GUI, click the LED0/LED1/LED2/LED3 and ON/OFF icons to control the on the DE10-Standard.

Note, the QR code link to:

http://www.terasic.com/downloads/demo/de10-nano/TerasicBluetooth.apk

Figure 4 BT_LED_APP QR-Code

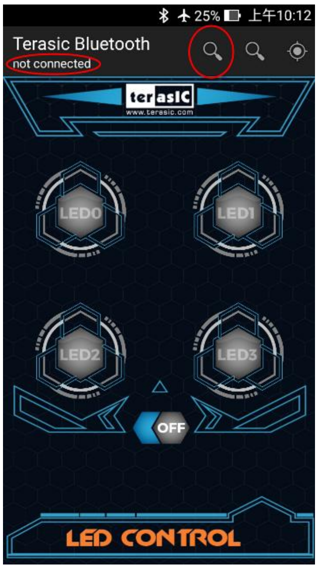


Figure 5 Search Bluetooth Device

3. Linux Application Project on DE10-Standard

The Linux BT_LED_AP is a C++ project. **Figure 6** show the main objects in the system. The CBtSppCommand object is used to receive the led control command from the Android smart phone. This object is running in a separated thread. The thread checks whether there is a coming command in polling method. The received commands are pushed into the CQueueCommand object.

The main program and the other objects are running in main thread. It will check the CQueueCommand whether there any queue command. If there are queue command, it will retrieve the command from the CQueueCommand object, and control CPIO_LED object to perform associated action.

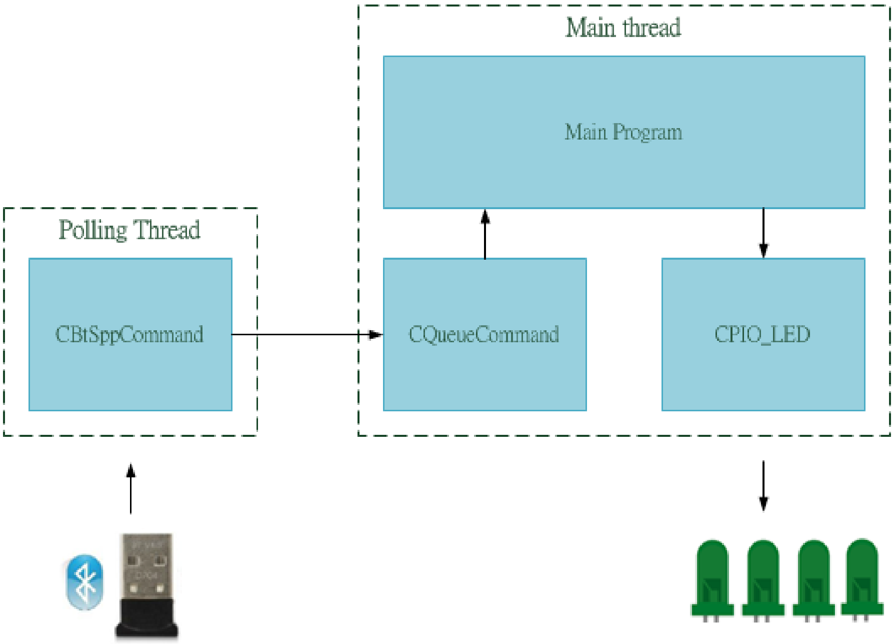


Figure 6 Major objects in the system

C++ Class

The major class used in this project as shown in the table below.

Class	Function Description	Implementation Files
BtSppCommand	Parse the raw data coming from Bluetooth. Derive from the CBtSpp class.	BtSppCommand.cpp/h
BtSpp	Provide Bluetooth SPP service based on the RFCOMM Bluetooth stack in Linux kernel	BtSpp.cpp/h
CQueueCommand	Queue received Bluetooth	QueueCommand.cpp/h

	command	
CQueue	Provide queue function	Queue.cpp/h
CPIO_LED	Control the LEDs from HPS side of DE10-Standard	PIO_LED.cpp/h

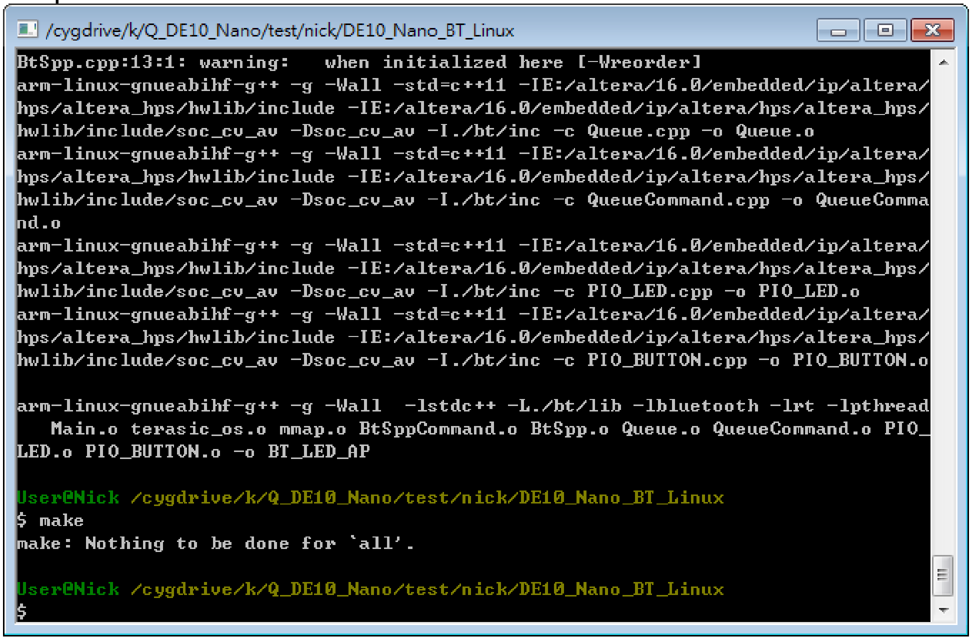
Build Project

Altera SoC EDS(Embedded Design Suite) is required to compile this C++ project. The C++ project is located at:

CD/Demonstrations/SoC_Advanced/Bluetooth_Spp/Linux_BT_App

Please follow the steps below to compile the project.

1. Make sure Altera SoC EDS v16.0 is installed on the host PC.
2. Copy the Linux_BT_App folder into the local hard disk of your host PC.
3. Launch Altera “SoC EDS Command Shell”.
4. In the shell, type “cd” command in the command shell to change the current directory to folder Linux_BT_App.
5. Type “make” to build the project, as shown in Figure 4
6. The “BT_LED_AP” binary file will be generated in the project directory if compile is successful.



```
BtSpp.cpp:13:1: warning:   when initialized here [-Wreorder]
arm-linux-gnueabihf-g++ -g -Wall -std=c++11 -IE:/altera/16.0/embedded/ip/altera/
hps/altera_hps/hwlib/include -IE:/altera/16.0/embedded/ip/altera/hps/altera_hps/
hwlib/include/soc_cv_av -Dsoc_cv_av -I./bt/inc -c Queue.cpp -o Queue.o
arm-linux-gnueabihf-g++ -g -Wall -std=c++11 -IE:/altera/16.0/embedded/ip/altera/
hps/altera_hps/hwlib/include -IE:/altera/16.0/embedded/ip/altera/hps/altera_hps/
hwlib/include/soc_cv_av -Dsoc_cv_av -I./bt/inc -c QueueCommand.cpp -o QueueComma
nd.o
arm-linux-gnueabihf-g++ -g -Wall -std=c++11 -IE:/altera/16.0/embedded/ip/altera/
hps/altera_hps/hwlib/include -IE:/altera/16.0/embedded/ip/altera/hps/altera_hps/
hwlib/include/soc_cv_av -Dsoc_cv_av -I./bt/inc -c PIO_LED.cpp -o PIO_LED.o
arm-linux-gnueabihf-g++ -g -Wall -std=c++11 -IE:/altera/16.0/embedded/ip/altera/
hps/altera_hps/hwlib/include -IE:/altera/16.0/embedded/ip/altera/hps/altera_hps/
hwlib/include/soc_cv_av -Dsoc_cv_av -I./bt/inc -c PIO_BUTTON.cpp -o PIO_BUTTON.o

arm-linux-gnueabihf-g++ -g -Wall  -lstdc++ -L./bt/lib -lbluetooth -lrt -lpthread
   Main.o terasic_os.o mmap.o BtSppCommand.o BtSpp.o Queue.o QueueCommand.o PIO_
LED.o PIO_BUTTON.o -o BT_LED_AP

User@Nick /cygdrive/k/Q_DE10_Nano/test/nick/DE10_Nano_BT_Linux
$ make
make: Nothing to be done for `all'.

User@Nick /cygdrive/k/Q_DE10_Nano/test/nick/DE10_Nano_BT_Linux
$
```

Figure 7 Screenshot of build all

Test BT_LED_AP Binary File

Copy the generated BT_LED_AP binary file into the /home/root directory of DE10-Standard Linux. Then following the demonstration setup procedure to perform the test.

4. Android Application Project

The Android TerasicBluetooth project is Java-based project built by Eclipse. The main function of the Android TerasicBluetooth project is to receive user's input from the GUI and send propriety commands to the spider robot through Bluetooth. The Android device should equity with Classical Bluetooth capacity for running this Bluetooth-based application. To open this project, Android Eclipse IDE is required. For installation details, please refer to the Appendix section in this document.

■ Android UI Layout and Components

Figure 8 shows the User Interface of this application software. Two kinds of build-in components are used in this application software. They are ImageView and ListView. The ImageView is used to implement button functions. When users click a button, the associated command will be sent to DE10-Standard through Bluetooth. When DE10-Standard receives the command, it will change the LED status. List View is used to log the command translated between Phone and DE10-Standard.

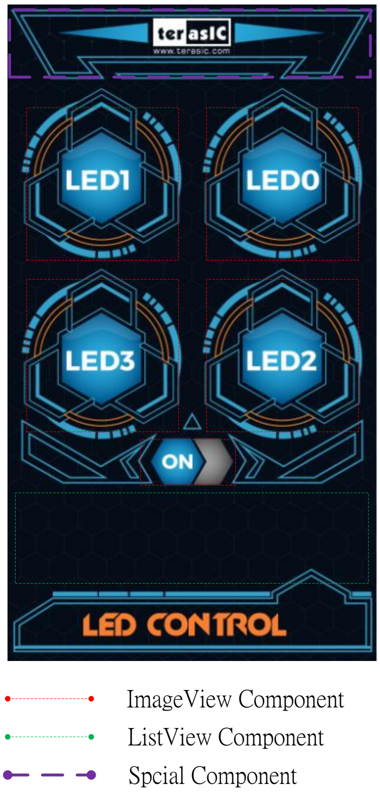


Figure 8 Android User Interface layout

■ Java Class

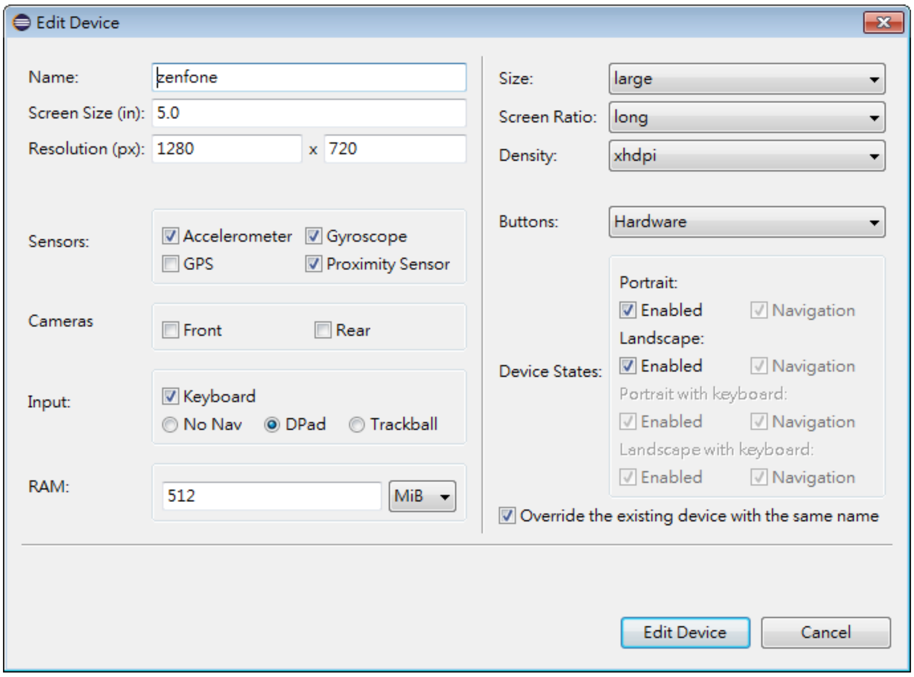
There are three Java Classes are used in this project. They are Bluetooth, BluetoothService and eviceListActivity. Blueotooth is the main Activity that handle .the GUI event. BluetoothService provides all the work for setting up, managing Bluetooth connector with other devices, and data transmission. DeviceListActivity appears as a dialog. It lists any paired devices and devices detected in the area after discovery.

If developers want to change GUI setting, they can modify the main.xml file under the layout folder in this application project.

■ Build Project

Both Android SDK and Eclipse ADT Plugin must be installed to complete the installation for this project prior to the development of Android. For installation details, please refer to the Appendix section in this document.

Beside, developers need to create a new Android Device for their Android Smart Phones. In the Eclipse GUI, select the menu item“windows->Android Virtual Devices Manager->Device Definitions->Create Device”to create a new device as shown in **Figure 9**.



Edit Device

Name: zenfone

Screen Size (in): 5.0

Resolution (px): 1280 x 720

Sensors: ☑ Accelerometer ☑ Gyroscope ☐ GPS ☑ Proximity Sensor

Cameras ☐ Front ☐ Rear

Input: ☑ Keyboard ○ No Nav ● DPad ○ Trackball

RAM: 512 MiB

Size: large

Screen Ratio: long

Density: xhdpi

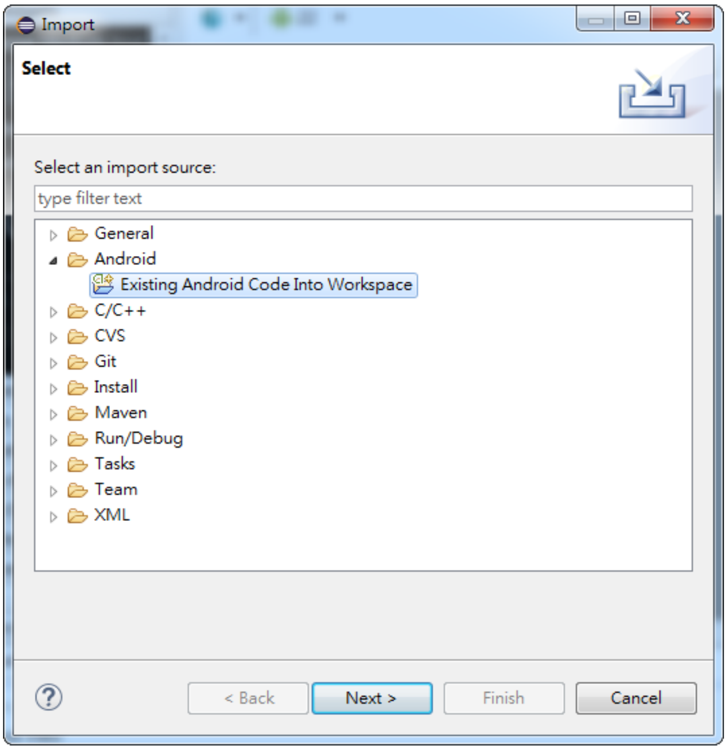
Buttons: Hardware

Device States:
Portrait: ☑ Enabled ☑ Navigation
Landscape: ☑ Enabled ☑ Navigation
Portrait with keyboard: ☑ Enabled ☑ Navigation
Landscape with keyboard: ☑ Enabled ☑ Navigation

☑ Override the existing device with the same name

Figure 9 New and Edit Android Device

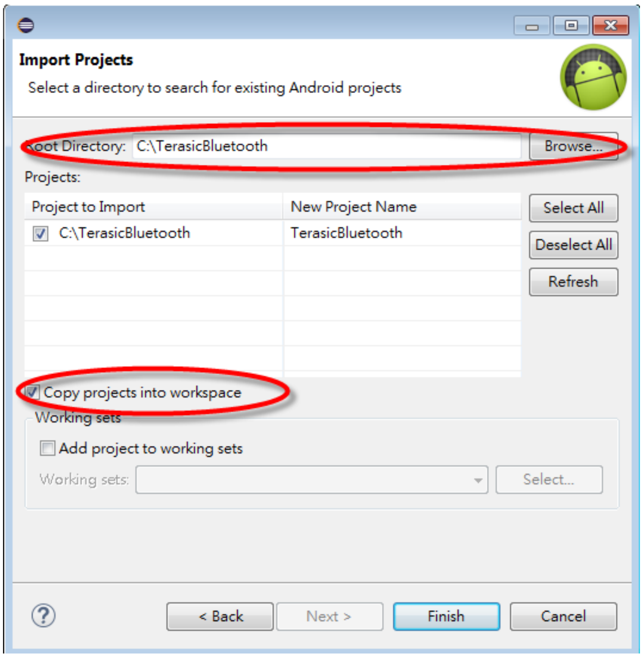
The Bluetooth project has to be imported prior to the start of building the project. In the Eclipsed, select the menu item “File->Import->Android->Existing Android Code into Workspace” to select the imported project type as shown in **Figure 10.**



Existing Android Code Into Workspace

Figure 10 Select Imported Project Type

In this Import Projects dialog, specify Bluetooth project location in the Root Directory edit box as shown in **Figure 11**. Then, click the Finish button.



Root Directory: C:\TerasicBluetooth

Project to Import	New Project Name
☑ C:\TerasicBluetooth	TerasicBluetooth

☑ Copy projects into workspace

□ Add project to working sets

Figure 11 Root Directory for Imported Project

After project is imported successfully, open the main gui file main.xml as shown in **Figure 12**. In this demonstration, Android Device is 5”1280x720 LCD, and API 19: Android 4.2.2 is used. Developers can change these setting according to their Android

Smart Phone. The binary file TerasicBluetooth.apk will be generated automatically when it will be download to the Android Smart Phone.

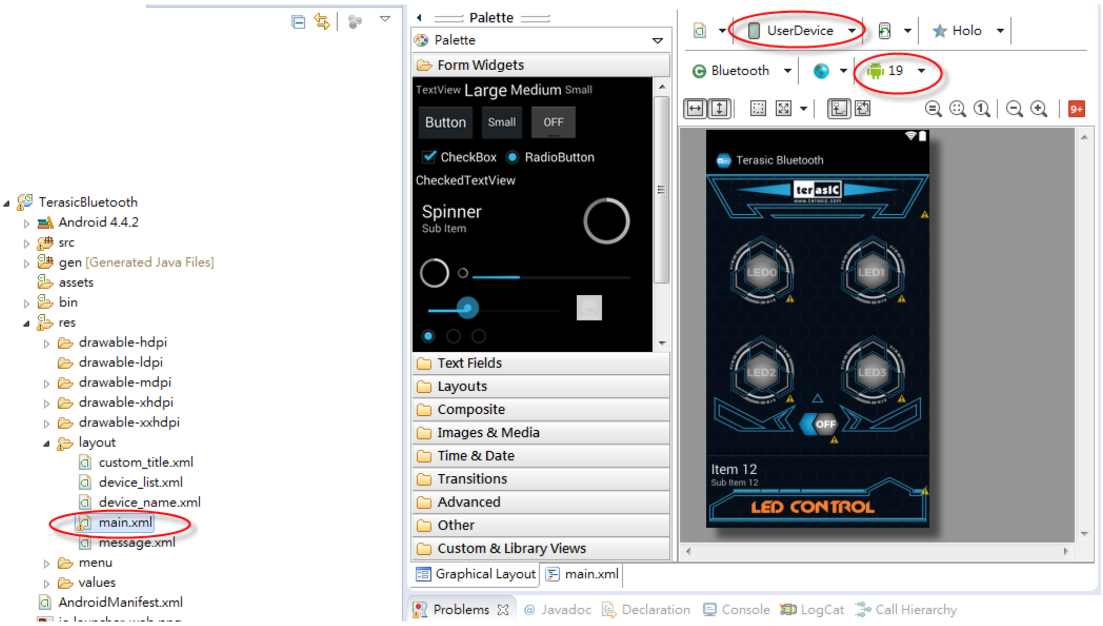


Figure 12 Project Main GUI

■ Download Binary File the Android Device

Note, before the TerasicBluetooth.apk binary file can be downloaded to Android mobile through Eclipse tool, the Debug mode of the users' Android Device must be enabled to allow the installation from an unknown source. The phone must also have the developer option turned on. The corresponding driver for users' Android Device also needs to be installed on the host PC.

Connect your host PC and Android Smart Phone with a USB cable. To update binary file, right click on the project folder to pop up a menu, then select the menu item "Run As→Android Application" to download the demo file to Android mobile, as shown in **Figure 13**

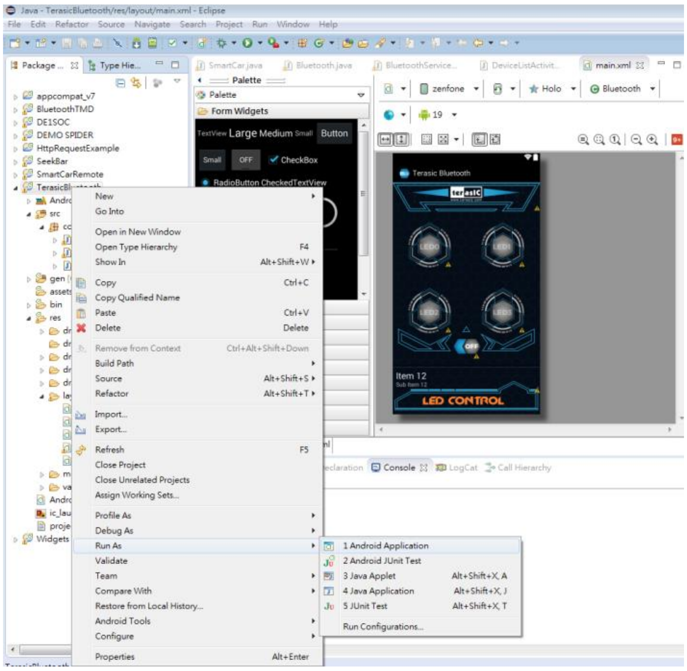


Figure 13 Select menu item "Run As→Android Application"

■ Run and Test Android Application

First, please launch the demo code on DE10-Standard. For details, please refer to Chapter 2. The following procedures below to the just download binary file.

1. Launch TerasicBluetooth application.
2. For first time to connect DE10-Standard, click ZOOM icon to discovery the DE10-Standard and pair it with pin-code "1234" as shown in **Figure 14**.
3. Click the ZOOM icon to connect the paired DE10-Standard.
4. In the TerasicBlueooth App GUI, click the LED0/LED1/LED2/LED3 and ON/OFF icons to control the on the DE10-Standard.

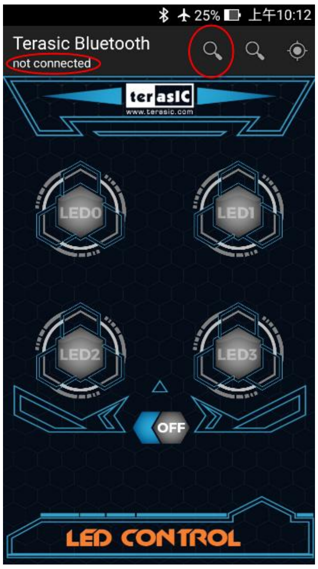


Figure 14 TerasicBluetooth Application UI

5. Appendix

The following items should be install on your host PC for developing Android Applications software. This section will describe how to download and install these items.

- Java JDK
- Eclipse
- Android SDK
- Android ADT

Install Java JDK

Go to web link below. In the web page, select Java Download icon as shown in. **Figure 15**

http://www.oracle.com/technetwork/java/javase/downloads/index.html

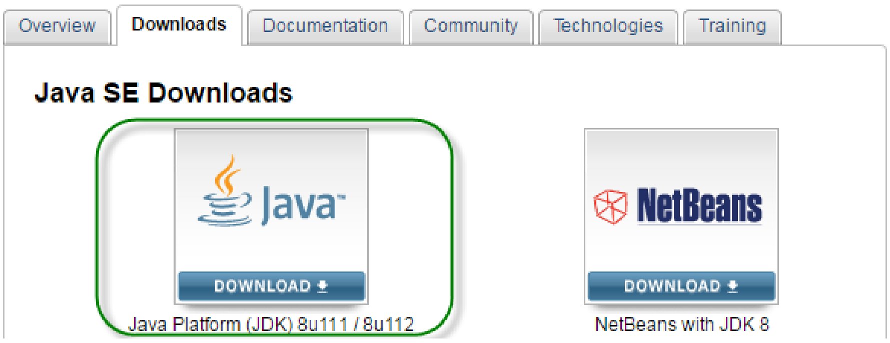
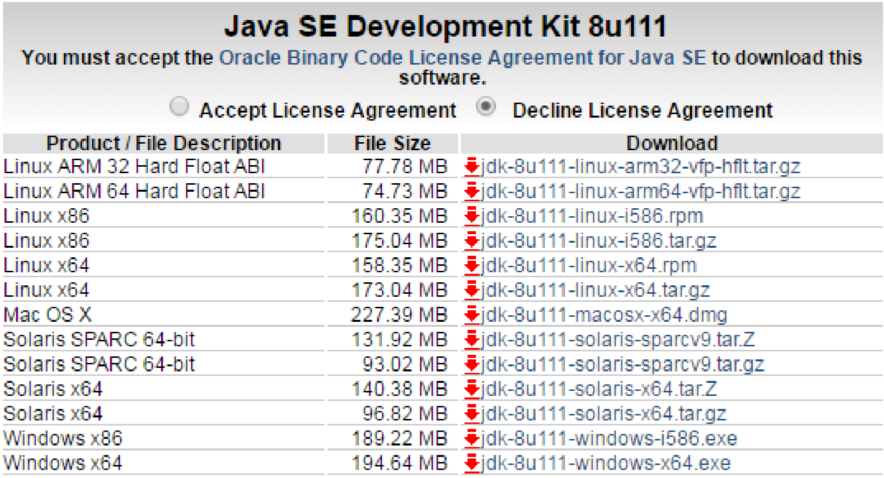


Figure 15 JAVA SE Download Web

In the download web as shown in **Figure 16**, select proper installer for your Host PC.



Java SE Development Kit 8u111

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ● Decline License Agreement

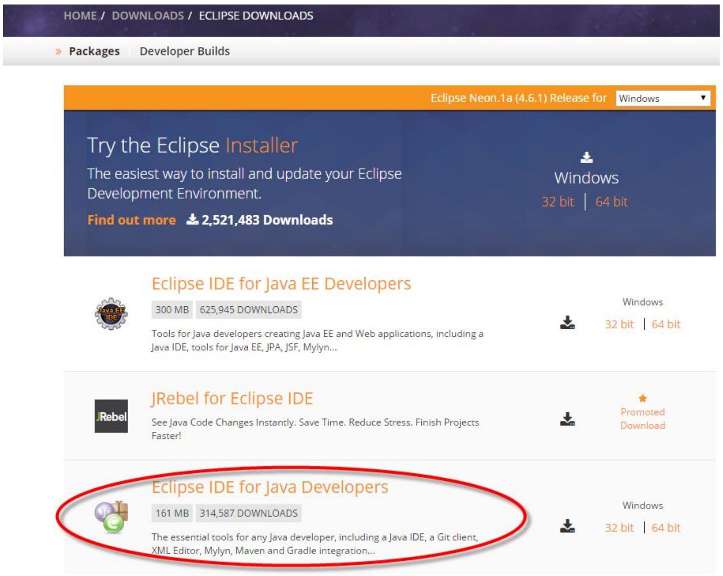
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.78 MB	jdk-8u111-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.73 MB	jdk-8u111-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.35 MB	jdk-8u111-linux-i586.rpm
Linux x86	175.04 MB	jdk-8u111-linux-i586.tar.gz
Linux x64	158.35 MB	jdk-8u111-linux-x64.rpm
Linux x64	173.04 MB	jdk-8u111-linux-x64.tar.gz
Mac OS X	227.39 MB	jdk-8u111-macosx-x64.dmg
Solaris SPARC 64-bit	131.92 MB	jdk-8u111-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	93.02 MB	jdk-8u111-solaris-sparcv9.tar.gz
Solaris x64	140.38 MB	jdk-8u111-solaris-x64.tar.Z
Solaris x64	96.82 MB	jdk-8u111-solaris-x64.tar.gz
Windows x86	189.22 MB	jdk-8u111-windows-i586.exe
Windows x64	194.64 MB	jdk-8u111-windows-x64.exe

Figure 16 JAVA SE Development Kit

Install Eclipse

Go to the web below. Select proper installer for your Host PC in the download page as shown in **Figure 17**. Execute the installer to install the Eclipse.

http://www.eclipse.org/downloads/eclipse-packages/



Try the Eclipse Installer

The easiest way to install and update your Eclipse Development Environment.

Find out more 2,521,483 Downloads

Eclipse IDE for Java EE Developers

300 MB 625,945 DOWNLOADS

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

JRebel for Eclipse IDE

See Java Code Changes Instantly. Save Time. Reduce Stress. Finish Projects Faster!

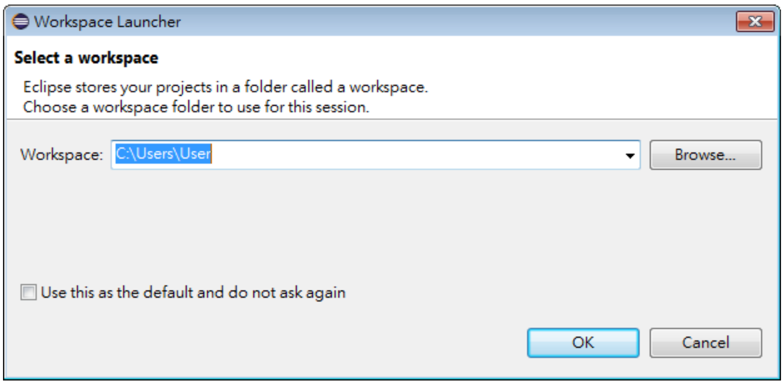
Eclipse IDE for Java Developers

161 MB 314,587 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration...

Figure 17 Eclipse Download Web

After Eclipse is installed successfully, launch the Eclipse. A the Workspace Launcher dialog will appear as shown in **Figure 18**, input project folder and then click OK.



Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\User

□ Use this as the default and do not ask again

Figure 18 Workspace directory for Eclipse

Install ADT (Android Development Tools)

In the Eclipse, select the menu item "Help->Install New Software …" as shown in **Figure 19**.

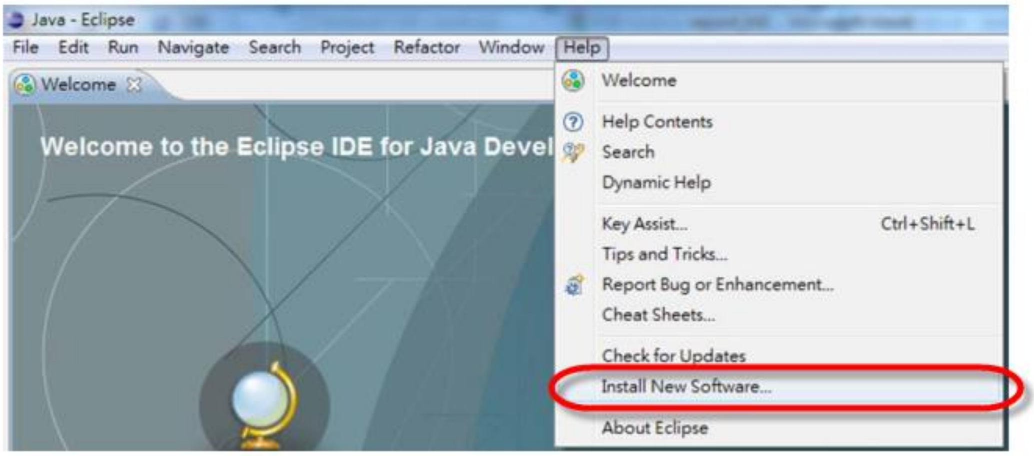
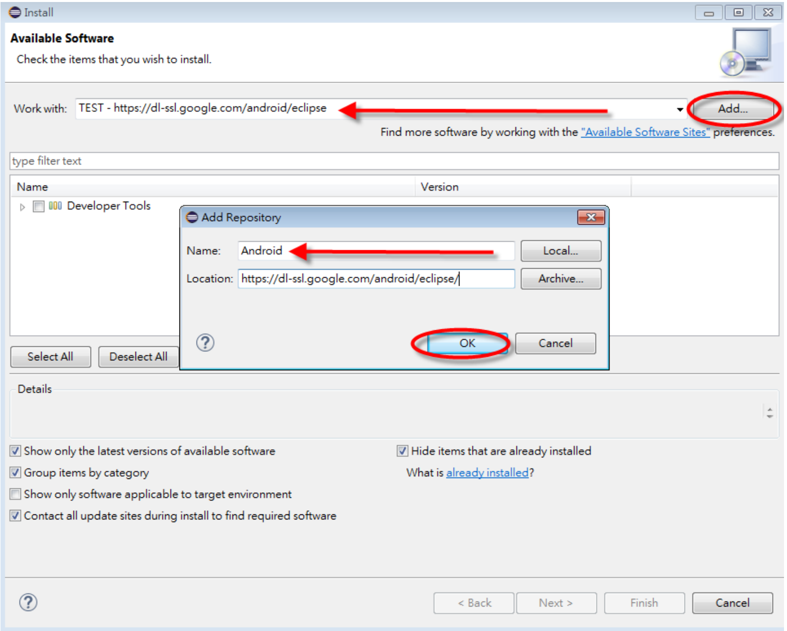


Figure 19 Install New Software

The **Install** dialog will appear as shown in **Figure 20**. In the dialog, click the "Add…" button to popup an **Add Repository** dialog. In the **Add Repository** dialog, type in the following information for Name and Location edit box:

Name: Android
Location: https://dl-ssl.google.com/android/eclipse/



Install

Available Software

Check the items that you wish to install.

Work with: TEST - https://dl-ssl.google.com/android/eclipse

Find more software by working with the "Available Software Sites" preferences.

Add Repository

Name: Android

Location: https://dl-ssl.google.com/android/eclipse/

☑ Show only the latest versions of available software
☑ Group items by category
□ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed

What is already installed?

Figure 20 Install ADT

In the **Install** Dialog, check the **Developer Tools** check box, then click the **Next** button to start downloading Android ADT as shown in **Figure 21**.

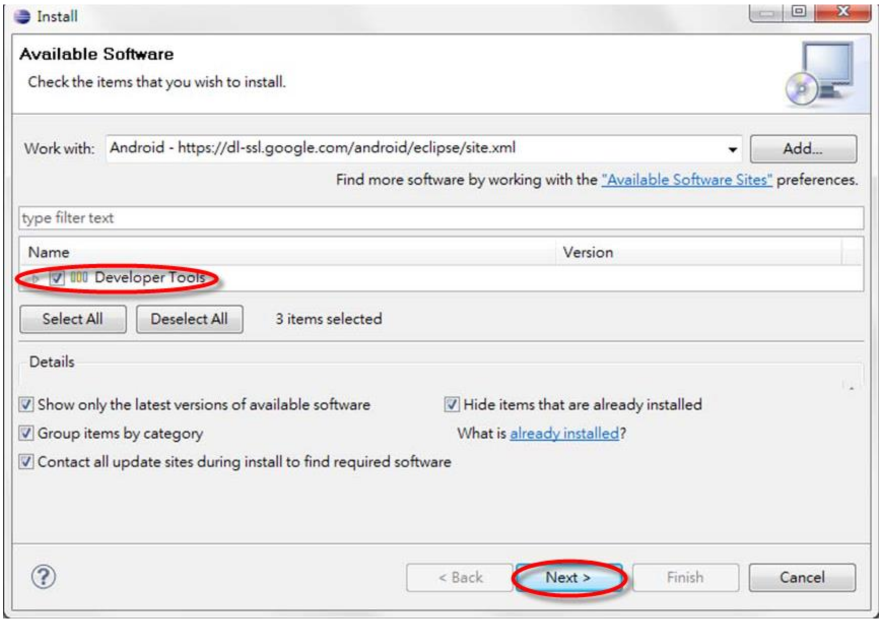

Install
Available Software
Check the items that you wish to install.
Work with: Android - https://dl-ssl.google.com/android/eclipse/site.xml
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
☑ Developer Tools
Select All
Deselect All
3 items selected
Details
☑ Show only the latest versions of available software
☑ Hide items that are already installed
☑ Group items by category
What is already installed?
☑ Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

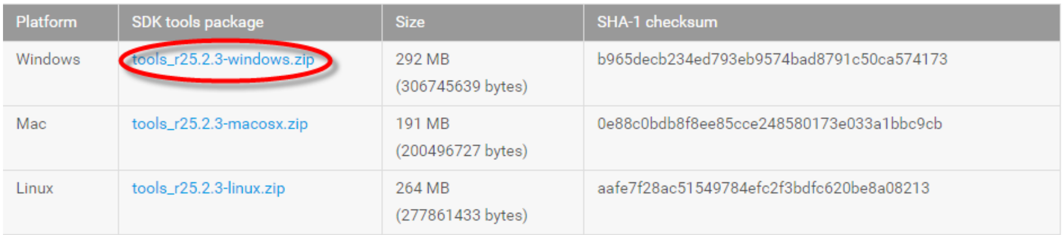
Figure 21 ADT Development Kit

After the installations completed, restart the Eclipse.

■ Install Android SDK

Go to the web link below to download the Android SDK as shown in **Figure 22**. Then uncompressed the downloaded zip file.

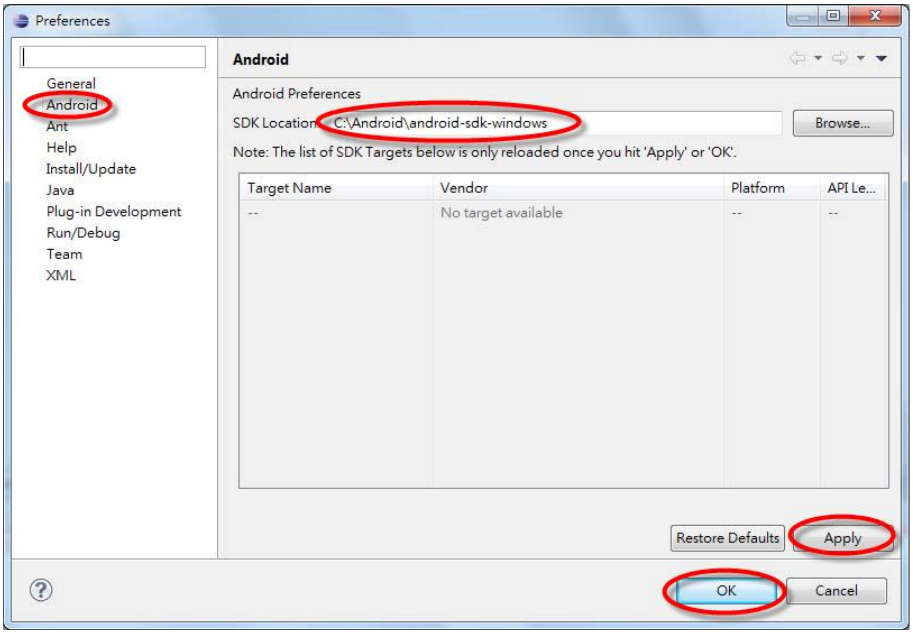
https://developer.android.com/studio/index.html



Platform	SDK tools package	Size	SHA-1 checksum
Windows	tools_r25.2.3-windows.zip	292 MB (306745639 bytes)	b965decb234ed793eb9574bad8791c50ca574173
Mac	tools_r25.2.3-macosx.zip	191 MB (200496727 bytes)	0e88c0bdb8f8ee85cce248580173e033a1bbc9cb
Linux	tools_r25.2.3-linux.zip	264 MB (277861433 bytes)	aafe7f28ac51549784efc2f3bdfc620be8a08213

Figure 22 Android SDK Download

In the Eclipse, select the menu item "Windows > Preference" to open the Preferences dialog as shown in **Figure 23**. In the dialog, select Android tab. In the SDK Location edit box, input the folder location where the uncompressed Android SDK is located.Figure 23



Android Preferences

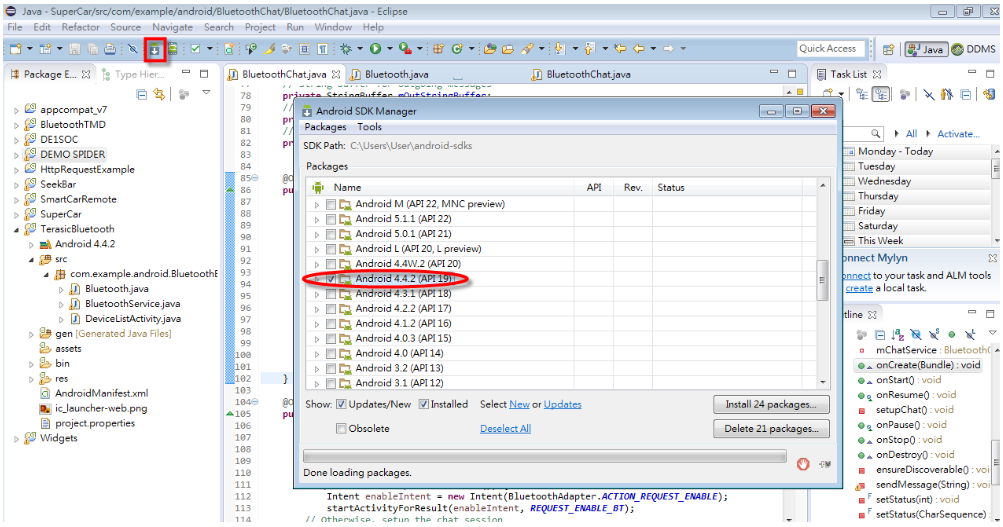
SDK Location: C:\Android\android-sdk-windows

Note: The list of SDK Targets below is only reloaded once you hit 'Apply' or 'OK'.

Target Name	Vendor	Platform	API Le...
--	No target available	--	--

Figure 23 Android SDK installation location

Then, we need to selected desired API for your Android Smart Phones. In Eclipse, select the menu "Windows→Android SDK Manager" to popup the **Android SDK Manager** dialog as shown in **Figure 24**. In the dialog, select the desired API package. In this demonstration, Android 4.2.2 (API 19) is used.



SDK Path: C:\Users\User\android-sdks

Packages

Name	API	Rev.	Status
□ Android M (API 22, MNC preview)			
□ Android 5.1.1 (API 22)			
□ Android 5.0.1 (API 21)			
□ Android L (API 20, L preview)			
□ Android 4.4W.2 (API 20)			
☑ Android 4.4.2 (API 19)			
□ Android 4.3.1 (API 18)			
□ Android 4.2.2 (API 17)			
□ Android 4.1.2 (API 16)			
□ Android 4.0.3 (API 15)			
□ Android 4.0 (API 14)			
□ Android 3.2 (API 13)			
□ Android 3.1 (API 12)			

Show: ☑ Updates/New ☑ Installed Select New or Updates

□ Obsolete Deselect All

Done loading packages.

```
Intent enableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(enableIntent, REQUEST_ENABLE_BT);
// Otherwise, setup the chat session
```

Figure 24 Android SDK Manager