1. Requirements

1.1 Hardware

1.1.1 CPU: Intel 2.5G I5 x64;

1.1.2 Memory: 16G DDR3;

1.1.3 Disk: 30GB;

1.1.4 Internet connected;

1.2 Software

1.2.1 OS: Ubuntu 12.04(64 bit);

1.2.2 Java JDK 6;

1.2.3 Eclipse;

1.2.4 ADT;

1.2.5 CDT;

1.2.6 Android SDK;

1.2.7 Android NDK;

1.2.8 Linux Source;

1.2.9 Android Source.

2. Initializing a Build Environment

Attention: Demonstration is in Ubuntu 12.04(32 bit).

2.1 Installing the JDK

The *Sun JDK* is no longer in Ubuntu's main package repository. In order to download it, you need to add the appropriate repository and indicate to the system which JDK should be used.

Input the following command in the Terminal:

\$ sudo gedit /etc/apt/sources.list

Add:

\$ deb http://us.archive.ubuntu.com/ubuntu/ hardy multiverse

Input in terminal:

\$ sudo add-apt-repository "deb http://archive.canonical.com/ lucid partner"

\$ sudo apt-get update

\$ sudo apt-get install sun-java6-jdk

2.2 Installing the Android SDK

Download the SDK from Web Browser:

http://developer.android.com/sdk/index.html

Choose the *adt-bundle-linux-x86-20130219.zip*.

adt-bundle-linuxx86-20130219.zip 418664018 bytes e56ebb5c8eb84eb3227cf7c255373f4b

Input in the terminal:

Linux 32-bit

\$ sudo mkdir ~/sdk

Copy the *adt-bundle-linux-x*86-20130219.zip to the */root/sdk* and extract the zip. Enter the folder */root/sdk/adt-bundle-linux-x*86-20130219/sdk/tools.



Double click the script file android.

It will show the Android SDK Manager. Choose the package with ticks.

🖷 Name	API	Rev.	Status	
7 🖉 🔲 Tools				
S X Android SDK Tools		21.1	🎒 Installed	
🜌 👬 Android SDK Platform-tools		16.0.2	🎒 Installed	
🖻 🖼 Android 4.2.2 (API 17)				
🖉 🖀 Android 4.1.2 (API 16)				
🖻 🖼 Android 4.0.3 (API 15)				
Android 4.0 (API 14)				
 				
🗆 🗎 Android 3.1 (API 12)				
• 🗆 🖀 Android 3.0 (API 11)				
- 🗆 🖀 Android 2.3.3 (API 10)				
🗹 🛤 Android 2.2 (API 8)				
🗆 🔚 Android 2.1 (API 7)				
🗆 🖀 Android 1.6 (API 4)				
🗆 🗃 Android 1.5 (API 3)				
- Extens		-		
C Dutras	1	8		
how: 🗹 Updates/New 🗹 Installed 🗌	Obsolete S	elect Ne	ew or <u>Updates</u>	Install 31 packages

Then click the *Install XX packages* button on the right. It will start download the required packages.

After the installation, add the SDK tools to PATH Environment Variable.

Input in the terminal:

\$ gedit /etc/profile
Input in the "profile"
\$ export ANDROID_SDK_PATH=/root/sdk/adt-bundle-linux-x86-20130219/sdk
\$ export PATH=.:\$ANDROID_SDK_PATH/platform-tools:\$PATH
Save and close the profile.

Input in the terminal:

\$ source /etc/profile

2.2 Installing the Eclipse

With the download of *adt-bundle-linux-x86-20130219.zip*, we can find the *Eclipse* in the /root/sdk/adt-bundle-linux-x86-20130219/eclipse.



Double click the *eclipse* can run it.

2.2.1 Installing the ADT online

Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.

ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, add packages based on the *Android Framework API*, debug your applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute your application.

Run the eclipse and click the *help>Install New Software*.

Then click the Add button.

	type or select a site	· .	Add
	Find mor	e software by working with the <u>"Available Software Sites"</u> pr	eferer
Name		Version	
🗆 🛈 Ther	e is no site selected.		
	Name:	Local	
	Location: http://	Archive	
Decails			
	y the latest versions of available softwa	re 📃 Hide items that are already installed	
Show on		and a first standard standard and a	
🗹 Show on 🗹 Group it	ems by category	What is <u>already installed</u> ?	
Show on Group it Show on	ms by category y software applicable to target environr	what is <u>aiready installed</u> ? nent	
Show on Group it Show on Contact	ens by category y software applicable to target environr Ill update sites during install to find requ	what is <u>arready installed</u> ? ment uired software	

Work with: adt - http://dl-ssl.google.com/andro	id/eclipse/	•	Add
Find m	ore software by working with the <u>"Ava</u>	ilable Software Sit	es" preferences
type filter text			0
Name	Version		
 DWD Developer Tools DWDK Plugins 			

Click Select All and Next.

2.2.2 Installing the NDK

The NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For certain types of apps, this can be helpful so you can reuse existing code libraries written in these languages, but most apps do not need the Android NDK.

Download the NDK from Web Browser:

http://developer.android.com/sdk/ndk/index.html

Choose the *android-ndk-r8e-linux-x86.tar.bz2*.

Linux 32-bit	android-ndk-r8e-linux-	461526099	26d774b0884bcd98de08eb4de41ab532
(X86)	x86.tar.bz2	bytes	

Input in the terminal:

\$ sudo mkdir ~/ndk

Copy the *android-ndk-r8e-linux-x86.tar.bz2* to the */root/sdk* and extract the zip. Input in the terminal:

\$tar -xvf android-ndk-r8e-linux-x86.tar.bz2

Finally, add the NDK folder to PATH Environment Variable.

2.2.3 Installing the CDT

The CDT Project provides a fully functional C and C++ Integrated Development Environment based on the Eclipse platform. Features include: support for project creation and managed build for various toolchains, standard make build, source navigation, various source knowledge tools, such as type hierarchy, call graph, include browser, macro definition browser, code editor with syntax highlighting, folding and hyperlink navigation, source code refactoring and code generation, visual debugging tools, including memory, registers, and disassembly viewers.

The installation of CDT is the same as ADT.

But the Location of CDT is:

http://download.eclipse.org/tools/cdt/releases/indigo

2.3 Installing the Cross Compiling Environment

The CPU X86 is using the CISC, while the ARM processor is RISO. Therefore, the cross compiling environment should be installed.

The CodeSourcery is an good choice for the cross compiling. It can be download from the following website.

<u>http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-e</u> <u>dition/</u>

Choose the ARM Processor GNU/Linux release.

ARM Processors
These releases support ARM, Thumb, and Thumb-2 compilations for all architectures in active use, including version 7 of the ARM architecture.
Read the FAQ >
Download the GNU/Linux Release >>
Download the EABI Release >
Download the SymbianOS release >

It will need your personal information. Finally, system will send you an Email with the download URL.

• G	NU C and C++ compilers	
• G	NU assembler and linker	
• c	and C++ runtime libraries	
• G	NU debugger	
Fi	irst Name	Last Name
E	mail	Country
		Choose Country
A	valid email address is required.	

Click the Download Sourcery CodeBench Lite 2012.09-64.

Sourcery CodeBench Lite Edition for ARM GNU/Linux hosted on IA32 Windows, IA32 GNU/Linux



Choose the IA32 GNU/Linux Installer.

Download	MD5 Checksum
Rec	ommended Packages
A32 GNU/Linux Installer	c6b76a7214e3eb404e6d8dadf4b2aab3
1A32 Windows installer	20583b80d0be222a41615d75fe776039
A	dvanced Packages
IA32 GNU/Linux TAR	b5c21a5b546a27912c9f7fd8ac9b4729
IA32 Windows TAR	7c876cb5ab7fb109f5efc33f488649fc
Source TAR	52456cf4d7bb81e267f91f30649f60a5

After download, enter the folder in terminal and use the following command:

\$ ln -s /lib/i386-linux-gnu/libc.so.6 /lib

\$ dpkg-reconfigure dash

Choose NO.

6	🔊 🖨 🗊 root@ubuntu: /home/siriux/Downloads
Pa	nckage configuration
t	
ſ	Configuring dash
	The system shell is the default command interpreter for shell scripts.
	Using dash as the system shell will improve the system's overall performance. It does not alter the shell presented to interactive users.
	Use dash as the default system shell (/bin/sh)?
	<yes> <no></no></yes>
t.	

\$ sh arm-2012.09-64-arm-none-linux-gnueabi.bin



Add the CodeSourvery to PATH. Input in the terminal: \$ gedit /etc/profile Input in the *profile*. \$export PATH=.:\$/root/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_LIinux/bin: \$PATH Save and close the *profile*. Input in the terminal: \$ source /etc/profile

3. Test the first cross compiling environment

Program a simple first.c file.

```
#include<stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}
Input in the terminal:
    $ arm-none-linux-gnueabi-gcc -static -o first first.c
    $ adb push first /data/local
    $ adb shell
    $ cd /data/local
    $ ./first
root@android:/data/local # ls
```

```
first
tmp
word_count.ko
word_count.o
root@android:/data/local # ./first
Hello world
```

Suggestions for modification: