

S720

SDK Compiling Environment Setup Guide

Issue 1.0



Copyright © Neoway Technology Co., Ltd. 2022. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Neoway Technology Co., Ltd.

neoway is the trademark of Neoway Technology Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

This document provides guide for users to use S720.

This document is intended for system engineers (SEs), development engineers, and test engineers.

THIS GUIDE PROVIDES INSTRUCTIONS FOR CUSTOMERS TO DESIGN THEIR APPLICATIONS. PLEASE FOLLOW THE RULES AND PARAMETERS IN THIS GUIDE TO DESIGN AND COMMISSION. NEOWAY WILL NOT TAKE ANY RESPONSIBILITY OF BODILY HURT OR ASSET LOSS CAUSED BY IMPROPER OPERATIONS.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE DUE TO PRODUCT VERSION UPDATE OR OTHER REASONS.

EVERY EFFORT HAS BEEN MADE IN PREPARATION OF THIS DOCUMENT TO ENSURE ACCURACY OF THE CONTENTS, BUT ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

Neoway provides customers complete technical support. If you have any question, please contact your account manager or email to the following email addresses:

Sales@neoway.com

Support@neoway.com

Website: <http://www.neoway.com>

Contents

About This Document.....	iv
Scope	iv
Audience	iv
Change History.....	iv
Conventions	iv
Related Document	iv
1 Configuring the Compiling Computer.....	5
2 Setting Up the SDK Compiling Environment.....	6
2.1 OS Requirements.....	6
2.2 Installing the Ubuntu14.04 64-bit Os.....	6
2.3 Installing the Tool Kit	6
2.4 Installing Python.....	7
2.5 Installing openjdk8	7
3 Testing the Compiling	9
4 Common Compiling Issues.....	10
4.1 System Library Not Found	10
4.2 Ubuntu Version Earlier than 14.04	10
4.3 Java Version Error.....	10

About This Document

Scope

This document is applicable to the S720 module and describes how to set up the S720 SDK compiling environment.




Audience

This document is intended for system engineers (SEs), development engineers, and test engineers.

Change History

Issue	Change	Author
1.0	Initial issue.	Luo Jian

Conventions

Symbol	Indication
	Indicates danger or warning. This information must be followed. Otherwise, a catastrophic module or user device failure or bodily injury may occur.
	Indicates caution. This symbol alerts the user to important points about using the module. If these points are not followed, the module or user device may fail.
	Indicates instructions or tips. This symbol provides advices or suggestions that may be useful when using the module.

Related Document

Neoway_S720_Datasheet

1 Configuring the Compiling Computer

The following configurations are recommended for a computer used to set up the S720 SDK compiling environment:

- Multi-core processor (CPU).

It is recommended to use a quad-core processor or a processor with more than four cores.

- 64-bit Linux OS.

It is recommended to use Ubuntu14.04 64-bit OS.

- Memory: 16 GB or above.

If the memory is lower than 16 GB, there is a high probability that "Exception in thread "main" java.lang.OutOfMemoryError: Java heap space" occurs during the compiling.

- Hard disk storage: 250 GB or above.
- Internet connection.

Used for system software download and update.

- Root permission.



- It is not recommended to use a virtual machine to set up a compiling server. During the use of the virtual machine, the virtual machine file will become larger and larger, which may easily cause the hard disk space of the host machine to be used up. If this issue occurs, you need to clear the hard disk space again; or you even need to delete the virtual machine, set up the compiling environment again, and then recompile the SDK code.
 - If a virtual machine must be used to set up a compiling environment for testing, allocate 16 GB of memory to the virtual machine. Otherwise, compiling errors will occur during the compiling.
-

2 Setting Up the SDK Compiling Environment

This chapter describes how to set up the SDK compiling environment on Ubuntu14.04 64-bit OS.

2.1 OS Requirements

It is recommended to select Ubuntu14.04 64-bit OS. The following command can be used to view the version of Ubuntu:

```
lsb_release -a
```

Additional required software packages:

- Python 2.6 - 2.7 provided in python.org
- GNU Make 3.81 - 3.82 provided in gnu.org
- Git 1.7 or later provided in git-scm.com



Ubuntu10.04-12.0 and later version Ubuntu16.04 are also supported, but the compiling support tools that are dependent on are slightly different for different versions.

2.2 Installing the Ubuntu14.04 64-bit Os

Make sure that the computer can be connected to Internet in order to update the Ubuntu14.04 system source and install the tool kit.

Step 1: Install the Ubuntu14.04 64-bit OS.

Step 2: Replace the default source of the system with the source of Alibaba Cloud or the source of NetEase 163, and run the **sudo apt-get update** command to update the source.

2.3 Installing the Tool Kit

If the compiling computer can be connected to Internet, enter the following command on the terminal to install the tool kit of the compiling dependent environment:

```
sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev  
gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev  
lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip libssl-dev
```



The required compiling support tool kits vary with different installed Ubuntu versions. You can access <https://source.android.com/setup/build/initializing> and download the complete tool kit.

2.4 Installing Python

There is a precompiled version of Python in `/prebuilts/python/` in the open source part of the SDK package. Therefore, there is no need to install Python separately.

If the system has another version of Python installed, it is recommended to delete it and install Python of version 2.7 or 2.6. The following command can be used to view the Python version:

```
python -V
```



You can download Python at the Python official website <https://www.python.org/>.

2.5 Installing openjdk8

Run the following commands to install openjdk8:

```
# Add the OpenJDK 8 ppa repository.  
sudo apt-get install --reinstall ca-certificates  
sudo add-apt-repository ppa:openjdk-r/ppa  
sudo apt-get update  
sudo apt-get install openjdk-8-jdk
```

```
support@neoway:~$ sudo add-apt-repository ppa:openjdk-r/ppa

More info: https://launchpad.net/~openjdk-r/+archive/ubuntu/ppa
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmpx6l6z4gb/secring.gpg' created
gpg: keyring `/tmp/tmpx6l6z4gb/pubring.gpg' created
gpg: requesting key 86F44E2A from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpx6l6z4gb/trustdb.gpg: trustdb created
gpg: key 86F44E2A: public key "Launchpad OpenJDK builds (all archs)" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
OK
support@neoway:~$
```

```
support@neoway:~$ sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libgif4 libice-dev libsm-dev libxt-dev
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
Suggested packages:
  default-jre equivs libice-doc libsm-doc libxt-doc openjdk-8-demo
  openjdk-8-source visualvm icedtea-8-plugin fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei ttf-telugu-fonts
  ttf-oriya-fonts ttf-kannada-fonts ttf-bengali-fonts
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libgif4 libice-dev libsm-dev libxt-dev
  openjdk-8-jdk
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
0 upgraded, 13 newly installed, 0 to remove and 429 not upgraded.
Need to get 38.5 MB of archives.
After this operation, 149 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
# set default to Java 8
sudo update-alternatives --config java
sudo update-alternatives --config javac
sudo update-alternatives --config javadoc
```


3 Testing the Compiling

After decompressing the SDK package, run `./build.sh S720` in the root directory of S720 SDK code to perform full compiling.

```
find ./bsp/out/$BOARD/dist/u-boot15/u-boot
find ./bsp/out/$BOARD/dist/chipram/u-boot-spl
find ./bsp/out/$BOARD/dist/chipram/fdl1.axf
find ./bsp/out/$BOARD/dist/modules/
not find ./bsp/out/$BOARD/dist/u-boot15/fdl2
not find ./bsp/out/$BOARD/dist/sml/bl32.elf
[symbols.vendor]
find ./out/target/product/$BOARD/symbols/vendor/
[symbols.system]
find ./out/target/product/$BOARD/symbols/system/
[symbols.idh]
not find ./bsp/out/$BOARD/dist/sml/bl31.elf
not find ./bsp/out/$BOARD/dist/trusty/lk.elf
===== copy symbols end =====

=====
Build Success !
=====

support@neoway:~/share/S720/S720_SDK_002$ ./build.sh S720
```



- A new full compiling can take from tens of minutes to several hours, depending on the performance of the compiling server.
- If the machine used for compiling supports multi-thread compiling, you can modify the `make -j` parameter in the `build.sh` script and use the `-j` parameter to speed up the compiling. The value after `-j` indicates multi-thread parallel compiling. The recommended value is twice the number of cores in the compiling server, mainly dependent on whether the compiler supports multi-thread parallel compiling, and also relating to the CPU. For example: `make -j8`. Do not use a large value for `-j`. This is to prevent compiling errors caused by irregular Makefile dependency writing. The larger the number of threads, the greater the probability of errors.

4 Common Compiling Issues

4.1 System Library Not Found

For example: fatal error: openssl/bio.h

Install openssl:

```
sudo apt-get install openssl
```

If a compiling error occurs after openssl/ssl.h is referenced and "xxx not found" is displayed, the solution is as follows:

```
sudo apt-get install libssl-dev build-essential zlibzlib-bin libidn1-dev libidn1
```

4.2 Ubuntu Version Earlier than 14.04

Google recommends using Ubuntu14.04 OS. If the Ubuntu OS of a version earlier than 14.04 is used, "GLIBC_2.17/2.18 not found" is reported.

If the Ubuntu OS of a version earlier than 14.04 needs to be used, install the corresponding dependency library. For details, refer to the following command:

```
sudo dpkg -i libc6_2.17-0ubuntu4_amd64.deb
```

4.3 Java Version Error

If the compiling Android version is inconsistent with the Java version, **make** will terminate, and the information similar to the following will be displayed.

```
*****
You are attempting to build with the incorrect version of java.
Your version is: WRONG VERSION.
The correct version is: RIGHT VERSION.
Please follow the machine setup instructions at
https://source.android.com/source/initializing.html
*****
```

The cause is that the specified JDK is not installed. In this case, make sure you have set environment variables and add the correct JDK to the beginning of the path, or remove the problematic JDK.