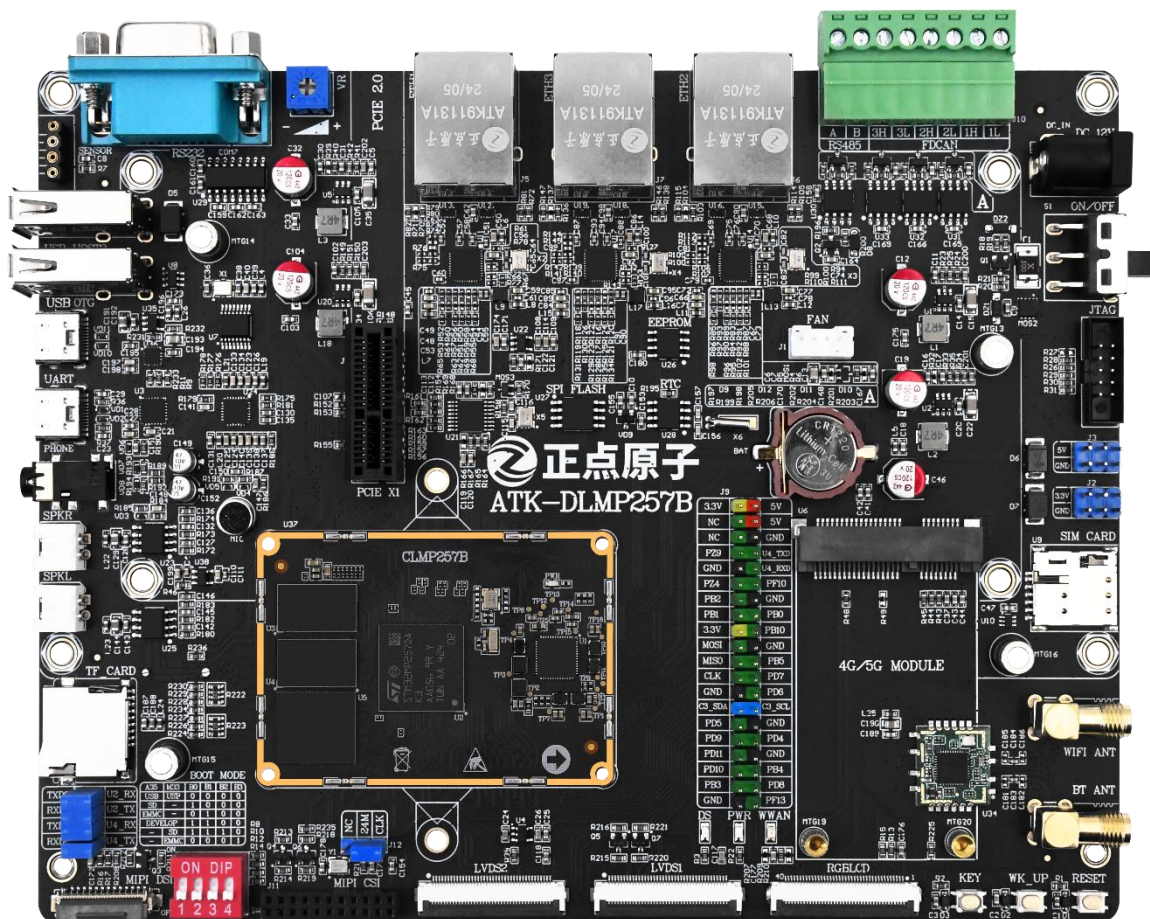


ATK-DLMP257B

Installs the Cross-compilation Toolchain Based
on the factory system

V1.0



1. Shopping:TMALL: <https://zhengdianyuanzi.tmall.com>TAOBAO: <https://openedv.taobao.com>**2. Download**Address: <http://www.openedv.com/docs/index.html>**3. FAE**Website : www.alientek.comForum : <http://www.openedv.com/forum.php>Videos : www.yuanzige.com

Fax : +86 - 20 - 36773971

Phone : +86 - 20 - 38271790



Disclaimer

The product specifications and instructions mentioned in this document are for reference only and subject to update without prior notice; Unless otherwise agreed, this document is intended as a product guide only, and none of the representations made herein constitutes a warranty of any kind. The copyright of this document belongs to Guangzhou Xingyi Electronic Technology Co., LTD. Without the written permission of the company, any unit or individual shall not be used for profit-making purposes in any way of dissemination.

In order to get the latest version of product information, please regularly visit the download center or contact the customer service of Taobao ALIENTEK flagship store. Thank you for your tolerance and support.

Revision History:

Version	Version Update Notes	Responsible person	Proofreading	Date
V1.0	release officially	ALIENTEK	ALIENTEK	2025.04.01

Catalogue

Introduction.....	1
Chapter 1. Installing the cross-compilation toolchain	2
1.1 Installing the compiler	3
1.2 Verifying the compiler version	3
Chapter 2. Uses the cross-compiler toolchain	5
2.1 .Simple compilation of c programs	6
2.2 Simple compilation of c++ programs.....	6
2.3 Its multiple uses	7

Introduction

This document is based on the Yocto system, that is, the installation and use of the cross-compilation toolchain on the factory system. `atk-image-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.x.s` is compiled based on Yocto of St. If the user needs to compile yocto, please compile based on the ST official website according to the information on the website, of course, the process is very hard! Here we do not need to re-create the wheel, directly use the cross compiler tool chain provided by the ALIENTEK is more convenient, direct development project, seize the market opportunity!

Chapter 1. Installing the cross-compilation toolchain

This chapter walks you through the installation of the cross-compilation toolchain, which can be used to build Qt projects, factory kernels, or source code such as Uboot. Therefore, you must first install the factory cross-compilation toolchain (or cross-compiler for short) for your system.

1.1 Installing the compiler

The network disk data A disk - basic data 05, development tools /01, factory system cross compiler /atk-image-openstlinux-weston-stm32mp2. Rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.0.sh. Copy x.h to Ubuntu.

The following picture, the author has copied to Ubuntu, Ubuntu and WIndows do not transfer files between the reader please see the other documentation.

```
alientek@alientek:~$ ls
公共  视频  文档  音乐  atk-image-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.0.sh  snap
模板  图片  下载  桌面  qt-opensource-linux-x64-5.12.12.run  vmware-tools-distrib
alientek@alientek:~$
```

Execute the following command to give the executable permission.

```
chmod +x atk-image-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.0.sh
```

```
alientek@alientek:~$ chmod +x atk-image-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.0.sh
alientek@alientek:~$
```

Install the toolchain directly by running the following command, and the script will ask you which path you want to install it in, which is usually the default /opt path. The installation takes some time and is completed as shown below.

```
./atk-image-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.0.sh
```

```
alientek@alientek:~$ ./atk-image-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-snapshot-20250115-v1.0.sh
ST OpenSTLinux - Weston - (A Yocto Project Based Distro) SDK installer version 5.0.3-snapshot
=====
Enter target directory for SDK (default: /opt/st/stm32mp2/5.0.3-snapshot):
You are about to install the SDK to "/opt/st/stm32mp2/5.0.3-snapshot". Proceed [Y/n]?
[sudo] password for alientek:
Extracting SDK.....done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
$ . /opt/st/stm32mp2/5.0.3-snapshot/environment-setup-cortexa35-ostl-linux
alientek@alientek:~$
```

1.2 Verifying the compiler version

Following the installation prompts, you'll need to run the following command each time you use the compiler in a new terminal.

```
source /opt/st/stm32mp2/5.0.3-snapshot/environment-setup-cortexa35-ostl-linux
```

```
alientek@alientek:~$ source /opt/st/stm32mp2/5.0.3-snapshot/environment-setup-cortexa35-ostl-linux
alientek@alientek:~$
```

To see the compiler version, execute the following command:

```
aarch64-ostl-linux-gcc -v
```



```

alientek@alientek:~$ aarch64-ostl-linux-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-ostl-linux-gcc
COLLECT_LTO_WRAPPER=/opt/st/stm32mp2/5.0.3-snapshot/sysroots/x86_64-ostl_sdk-linux/usr/libexec/aarch64-ostl-linux/gcc/aarch64-ostl-linux/13.3.0/lto-wrap
per
Target: aarch64-ostl-linux
Configured with: .././.././../work-shared/gcc-13.3.0-r0/gcc-13.3.0/configure --build=x86_64-linux --host=x86_64-ostl_sdk-linux --target=aarch64-ostl-linux --prefix=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr --exec_prefix=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr --bindir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/bin/aarch64-ostl-linux --sbindir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/bin/aarch64-ostl-linux --libexecdir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/libexec/aarch64-ostl-linux --datadir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/share --sysconfdir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/etc --sharedstatedir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/x/com --localstatedir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/var --libdir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/lib/aarch64-ostl-linux --includedir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/include --oldincludedir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/include --infodir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/share/info --mandir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64-ostl_sdk-linux/usr/share/man --disable-silent-rules --disable-dependency-tracking --with-libtool-sysroot=/home/alientek/mp2-v6.0-yocto/build-openstlinweston-stm32mp2/tmp-glibc/work/x86_64-nativesdk-ostl_sdk-linux/gcc-cross-canadian-aarch64/13.3.0/recipe-sysroot --with-gnu-ld --enable-shared --enable-languages=c,c++ --enable-threads=posix --enable-multilib --enable-c99 --enable-long-long --enable-symvers=gnu --enable-libstdc++-pch --program-prefix=aarch64-ostl-linux- --without-local-prefix --disable-install-liberty --disable-libssp --enable-libitm --enable-lto --disable-bootstrap --with-system-zlib --with-linker-hash-style=gnu --enable-linker-build-id --with-hppl=no --with-cloog=no --enable-checking=release --enable-headers=c global --without-isl --with-gxx-include-dir=/not/exist/usr/include/c++/13.3.0 --with-build-time-tools=/home/alientek/mp2-v6.0-yocto/build-openstlinweston-stm32mp2/tmp-glibc/work/x86_64-nativesdk-ostl_sdk-linux/gcc-cross-canadian-aarch64/13.3.0/recipe-sysroot-native/usr/aarch64-ostl-linux/bin --with-sysroot=/not/exist --with-build-sysroot=/home/alientek/mp2-v6.0-yocto/build-openstlinweston-stm32mp2/tmp-glibc/work/x86_64-nativesdk-ostl_sdk-linux/gcc-cross-canadian-aarch64/13.3.0/recipe-sysroot --with-plugin-ld=ld --enable-poison-system-directories --disable-static --enable-nls --with-glibc-version=2.28 --enable-initfini-array --enable-_cxa_atexit
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 13.3.0 (GCC)
alientek@alientek:~$

```

Chapter 2. Uses the cross-compiler toolchain

In this chapter, we will teach you how to use the cross-compilation toolchain.

2.1 .Simple compilation of c programs

For simpler C programs, we can use a compiler to compile them without linking to libraries.

Create a new main.c program.

```
#include <stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

Execute the following to compile.

```
$CC main.c -o main
```

```
alientek@alientek:~$ $CC main.c -o main
alientek@alientek:~$ ls main
main
alientek@alientek:~$
```

- \$CC is the C cross-compiler, you can print it with echo \$CC to see! For ARM 64 bit (aarch64) architecture of Linux system.
- main.c is the C source file to be compiled.
- -o main specifies the name of the output file, in this case an executable file named main.

Copy the compiled main executable file to the development board to execute, directly ./main execution.

2.2 Simple compilation of c++ programs

For simpler C++ programs, the compiler can be used directly to compile them without linking to libraries.

Create a new main.cpp program.

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
    return 0;
}
```

Execute the following to compile.

```
$CXX main.cpp -o main
```

```
alientek@alientek:~$ $CXX main.cpp -o main
alientek@alientek:~$ ls main
main
alientek@alientek:~$
```

- \$CXX is the C++ cross compiler, you can print echo \$CXX to see! For ARM 64 bit (aarch64) architecture of Linux system.
- main.cpp is the C++ language source file to be compiled.
- -o main specifies the name of the output file, in this case an executable file named main.

Copy the compiled main executable file to the development board to execute, directly./main execution

2.3 Its multiple uses

Learn about Makefiles and CMakeLists first.