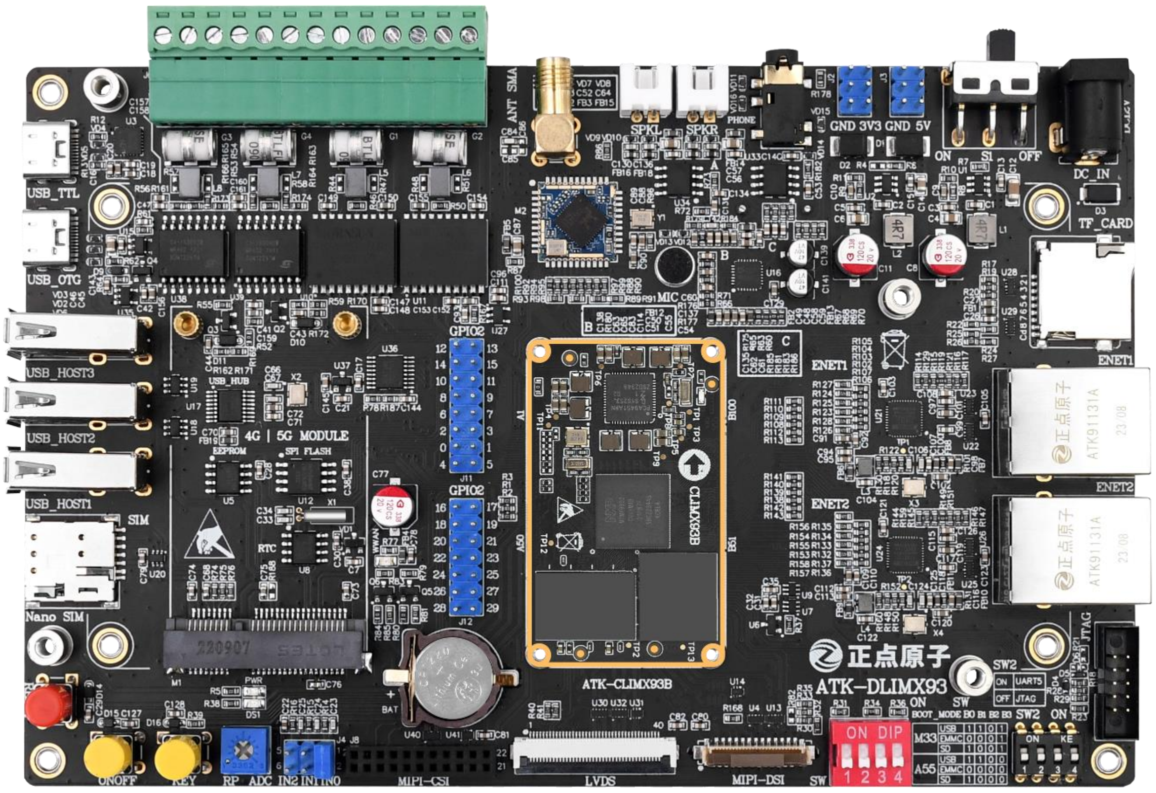
ATK-DLIMX93

Factory System Source Code Use Guide

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.com
Forum	:	http://www.openedv.com/forum.php
Videos	:	www.yuanzige.com
Fax	:	+86 - 20 - 36773971
Phone	:	+86 - 20 - 38271790

Disclaimer

Revision History:

Version	Version Update Notes	Responsible person	Proofreading	Date
V1.0	release officially	ALIENTEK	ALIENTEK	2024.04.30

Catalogue

Chapter 1. Describes the environment

This document mainly introduces the compilation process of the factory system source code of the ATK-DLIMX93 development board, and guides the user to master the factory source code compilation method of the development board, so as to facilitate the subsequent secondary development. This document is not intended to explain how each part of compilation works.

The environment used in this document:

- Windows 10 64bits. It is not recommended to use Windows 32 bit to develop, Windows 32 bit support memory size is limited, the system performance is limited. This document is a 64-bit operating system for the introduction.
- Ubuntu20.04. Ubuntu recommends using 20.04, otherwise errors may occur due to different installation environments.
- The reader is required to use FileZilla or WinSCP to transfer files between Ubuntu and Windows.
- This document is written for the default factory system source code of the ATK-DLIMX93 development board. If the user uses other source code versions or other versions of the cross compiler, it may be due to the source code configuration, compiler support instructions and other different, resulting in compilation errors, need to solve their own, this document compilation instructions for reference only. For the convenience of development, please use the default factory system source code.

1.1 Software Versions

Source code/tools	Version	Note
U-Boot and SPL	2023.04	
Linux kernel	6.1.55	
Qt	6.5.0	
aarch64-poky-linux-gcc	12.3.0	Cross-compilation toolchain
UUU	1.5.163	NXP burn tool
Development environment	Ubuntu20.04	Virtual machine image provided by ALIENTEK

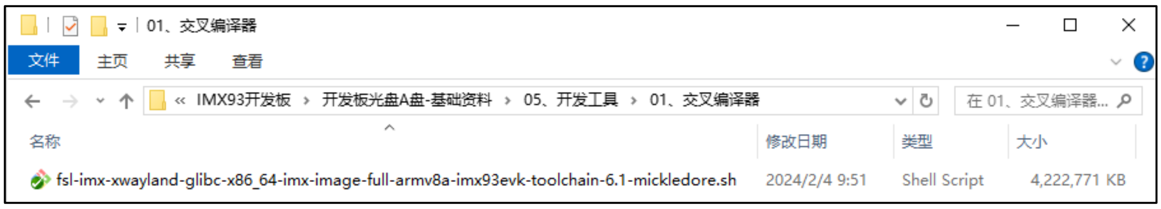
Chapter 2. Installing the ARM cross-compiler toolchain

2.1 Description of the cross-compilation toolchain

The factory system source code of ATK-DLIMX93 development board needs to be compiled by ARM cross compilation tool chain, and finally executable binary files are generated and burned to the development board. This section describes how to install the cross-compilation toolchain.

The ARM cross-compilation tool chain provided by the ATK-DLIMX93 development board is located in the data path of the network disk:

IMX93 development board \ Development board CD A disk - basic information \5_tools \ 1_Factory_system_cross_compiler


01、交叉编译器
文件 主页 共享 查看
« IMX93开发板 › 开发板光盘A盘-基础资料 › 05、开发工具 › 01、交叉编译器
在 01、交叉编译器...

名称	修改日期	类型	大小
fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh	2024/2/4 9:51	Shell Script	4,222,771 KB

Figure 2.1-1 The ATK-DLIMX93 cross-compilation toolchain

fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh toolchain is a cross-compilation toolchain generated based on the official NXP yocto file system. It is mainly used to compile filesystem related commands and programs (including Qt and AI related programs).

The size of the toolchain installation package is about 4GB, and the installation space is about 24GB. Please ensure that the environment storage space is sufficient before installation.

2.2 Source cross-compilation toolchain installation

Before installing, first copy it to the Ubuntu virtual machine. The ATK-DLIMX93 development board is developed using Ubuntu20.04. It is suggested that users use Ubuntu20.04 in a unified way. The user environment is consistent with the author's environment, which can facilitate the use of problems in the future.

fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh is a cross-compile toolchain installation package, Copy this file to Ubuntu20.04 virtual machine. This article has copied the cross-compiler tool to Ubuntu virtual machine.

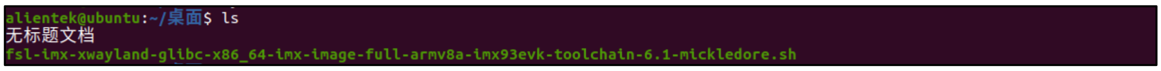

```
alientek@ubuntu:~/桌面$ ls
无标题文档
fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh
```

Figure 2.2-1 Copy the cross-build toolchain installation package to your Ubuntu system

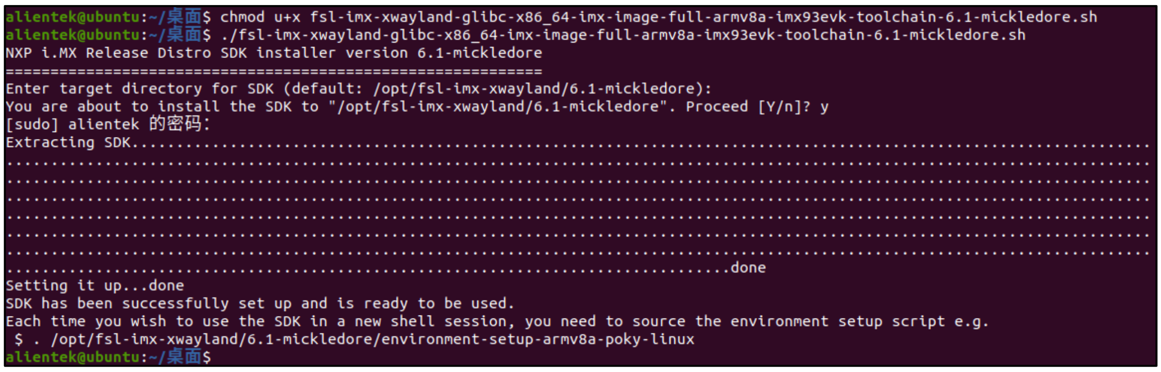
Execute the following command to change the script permissions.

```
chmod u+x fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh
```

Directly execute the script to install the cross-compiler tool, press the Enter key twice in succession to confirm, and then enter the user password. The directory of this installation is the default installation directory specified by the script, and the cross-compilation of the following kernel compilation environment is operated according to this installation directory, so it is recommended that users also install the default directory /opt/fsl-imx-xwayland/6.1-mickledore.

Execute the following command to install the cross-compilation toolchain.

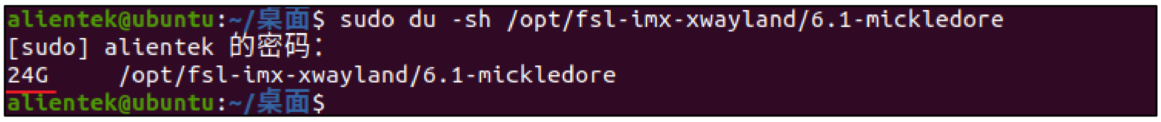
```
./fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh
```



```
alientek@ubuntu:~/桌面$ chmod u+x fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh
alientek@ubuntu:~/桌面$ ./fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh
NXP i.MX Release Distro SDK installer version 6.1-mickledore
============================================================
Enter target directory for SDK (default: /opt/fsl-imx-xwayland/6.1-mickledore):
You are about to install the SDK to "/opt/fsl-imx-xwayland/6.1-mickledore". Proceed [Y/n]? y
[sudo] alientek 的密码:
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/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux
alientek@ubuntu:~/桌面$
```

Figure 2.2-2 Install the cross-compile toolchain to the default directory.

The installed toolchain directory size is 24GB. This toolchain contains Qt, AI and other related libraries, so it needs enough space.

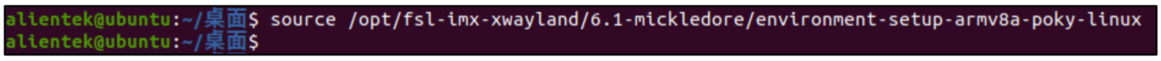


```
alientek@ubuntu:~/桌面$ sudo du -sh /opt/fsl-imx-xwayland/6.1-mickledore
[sudo] alientek 的密码:
24G     /opt/fsl-imx-xwayland/6.1-mickledore
alientek@ubuntu:~/桌面$
```

Figure 2.2-3 Total toolchain size

It's also very easy to use, following the instructions printed above, just enable the environment variables. But in different terminals or switch users need to re-enable the environment variable can be used.

```
source /opt/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux
```

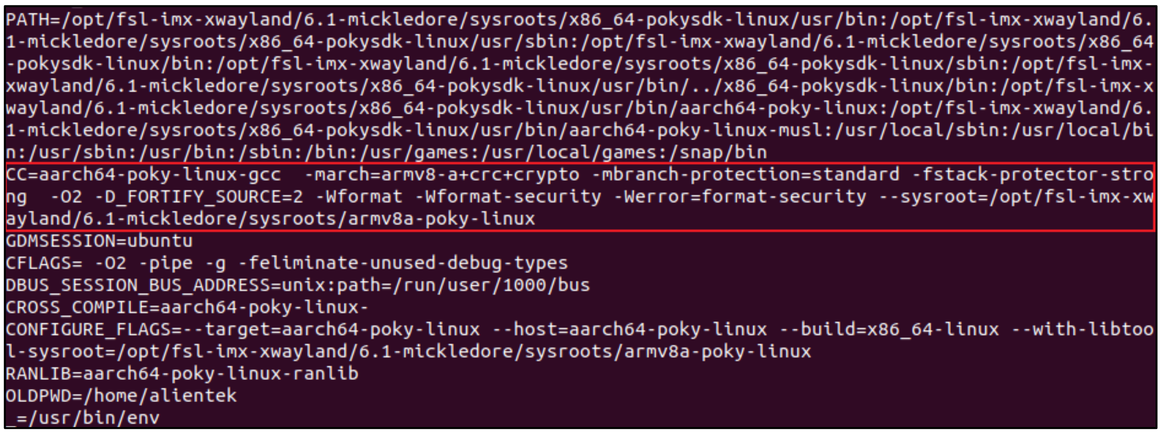


```
alientek@ubuntu:~/桌面$ source /opt/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux
alientek@ubuntu:~/桌面$
```

Figure 2.2-4 Enable environment variables

After enabling environment variables, you can use the env directive to see which environment variables are in effect. The following screenshot shows that gcc has configured the parameters for compilation with this environment variable enabled. Below is a screenshot of the environment variables section.

```
env
```



```
PATH=/opt/fsl-imx-xwayland/6.1-mickledore/sysroots/x86_64-pokysdk-linux/usr/bin:/opt/fsl-imx-xwayland/6.
1-mickledore/sysroots/x86_64-pokysdk-linux/usr/sbin:/opt/fsl-imx-xwayland/6.1-mickledore/sysroots/x86_64
-pokysdk-linux/bin:/opt/fsl-imx-xwayland/6.1-mickledore/sysroots/x86_64-pokysdk-linux/sbin:/opt/fsl-imx-
xwayland/6.1-mickledore/sysroots/x86_64-pokysdk-linux/usr/bin/../x86_64-pokysdk-linux/bin:/opt/fsl-imx-x
wayland/6.1-mickledore/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux:/opt/fsl-imx-xwayland/6.
1-mickledore/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux-musl:/usr/local/sbin:/usr/local/bi
n:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
CC=aarch64-poky-linux-gcc  -march=armv8-a+crc+crypto -mbranch-protection=standard -fstack-protector-stro
ng  -O2 -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -Werror=format-security --sysroot=/opt/fsl-imx-xw
ayland/6.1-mickledore/sysroots/armv8a-poky-linux
GDMSESSION=ubuntu
CFLAGS= -O2 -pipe -g -feliminate-unused-debug-types
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
CROSS_COMPILE=aarch64-poky-linux-
CONFIGURE_FLAGS=--target=aarch64-poky-linux --host=aarch64-poky-linux --build=x86_64-linux --with-libtoo
l-sysroot=/opt/fsl-imx-xwayland/6.1-mickledore/sysroots/armv8a-poky-linux
RANLIB=aarch64-poky-linux-ranlib
OLDPWD=/home/alientek
_=/usr/bin/env
```

Figure 2.2-5 Look at the enabled environment variables

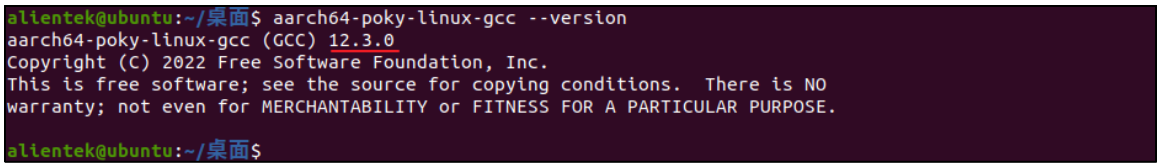
2.3 Query cross compiler

Now that you have the cross-compilation toolchain installed, you can verify that it was installed by querying the GCC version that comes with the toolchain.

Query the GCC version of the cross compiler.

```
aarch64-poky-linux-gcc --version
```

The result is shown in the following screenshot:



```
alientek@ubuntu:~/桌面$ aarch64-poky-linux-gcc --version
aarch64-poky-linux-gcc (GCC) 12.3.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

alientek@ubuntu:~/桌面$
```

Figure 2.3-1 Query cross compiler version

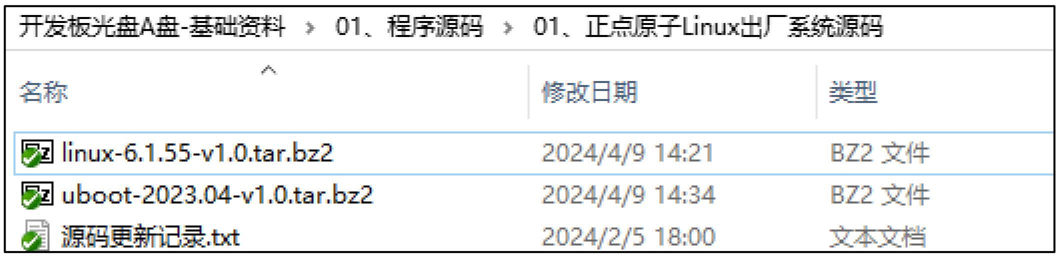
It can be seen from the above figure that the GCC version of the cross-compiler is 12.3.0, which proves that the ARM cross-compilation tool chain of the development board has been installed successfully.

Chapter 3. Factory system source code compilation and image construction

The factory system source code is to adapt to ATK-DLIMX93 development board hardware resources, the original source code version provided by NXP official transplantation, modification. Users can carry out secondary development and debugging based on the factory system source code. For example, if the hardware resources of the design base board and the development board are different, they can modify a part of the factory system source code and directly compile and debug.

The factory system source code is located in the network disk data path:

IMX93 development board \ development board CD A disk - basic information \1_codes \ 1_ ALIENTEK_Linux_factory_system_source_code



开发板光盘A盘-基础资料 › 01、程序源码 › 01、正点原子Linux出厂系统源码

名称	修改日期	类型
linux-6.1.55-v1.0.tar.bz2	2024/4/9 14:21	BZ2 文件
uboot-2023.04-v1.0.tar.bz2	2024/4/9 14:34	BZ2 文件
源码更新记录.txt	2024/2/5 18:00	文本文档

Figure 2.3-1 Factory system source code example

As shown in the figure above, the factory source code will be iterated if necessary, and information such as version number will change from the example in the figure, so users can use it directly. For the convenience of driver development learning, Linux kernel source code does not do Git management, only retain a release number. In order to facilitate the explanation of this document, the v1.0 version is taken as an example to write here, and the specific source code version is subject to the latest release. Pay attention to change the following decompress source code instructions related to the source code version.

The following sections describe how the source code is compiled.

Before compiling, please create a directory named "ATK-DLIMX93" in any directory of Ubuntu environment. All the following factory source code compilation operations will be carried out in the ATK-DLIMX93 directory.

In addition, install the following tools on ubuntu20.04 to avoid the lack of libraries at compile time:

```
sudo apt-get install make gcc libssl-dev g++ git libncurses5-dev libncursesw5-dev libyaml-dev
sudo apt-get install u-boot-tools python3-pyelftools device-tree-compiler bison flex expect
```

3.1 Compile the U-Boot source code and flash.bin firmware

The U-Boot and SPL bin files (u-boot.bin and u-boot-spl.bin) can be compiled from the U-Boot source.

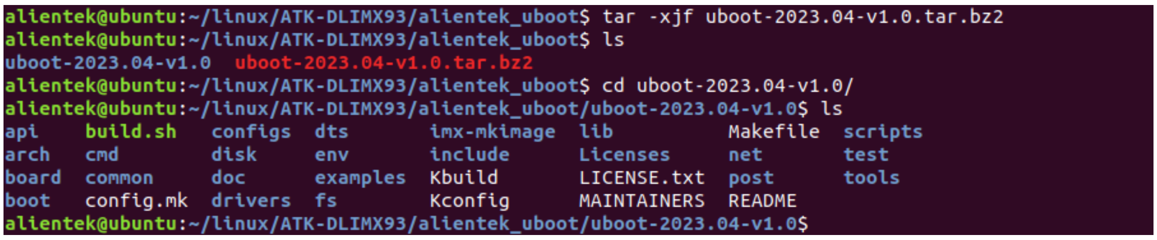
To build uboot, you need to install the library by entering the following:

```
sudo apt-get install bison flex
```

Create a directory called "alientek_uboot" in the ATK-DLIMX93 directory of Ubuntu, and copy the factory U-Boot source package uboot-2023.04-v1.0.tar.bz2 to the directory called alientek_uboot.

The unzip command is as follows: (Please confirm the source code version you downloaded, do not directly copy the unzip command below)

```
tar -xjf uboot-2023.04-v1.0.tar.bz2
```



```
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot$ tar -xjf uboot-2023.04-v1.0.tar.bz2
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot$ ls
uboot-2023.04-v1.0  uboot-2023.04-v1.0.tar.bz2
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot$ cd uboot-2023.04-v1.0/
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot/uboot-2023.04-v1.0$ ls
api    build.sh   configs  dts       imx-mkimage  lib          Makefile  scripts
arch   cmd        disk     env       include      Licenses     net       test
board  common     doc      examples  Kbuild       LICENSE.txt  post      tools
boot   config.mk  drivers  fs        Kconfig      MAINTAINERS  README
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot/uboot-2023.04-v1.0$
```

Figure 3.1-1 Unzipped U-Boot source code

When the factory source code is compiled, the user does not have to modify anything. To facilitate your compilation, ALIENTEK provides a one-button build script build.sh, you can directly run the script to compile the first time, the script is as follows:

```
# Check if the cross-compile toolchain exists, using arm-poky-linux-gnueabi- (gcc-12.3.0)
if [ ! -e "/opt/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux" ]; then
    echo ""
    echo "请先安装正点原子 IMX93 开发板光盘 A-基础资料->05、开发工具->01、交叉编译器
->fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh"
    echo ""
exit 1
fi

# Use the GCC 12.3.0 cross-compiler in the Yocto SDK to compile the factory Linux source code, you can not specify ARCH, etc., directly execute Make
source /opt/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux
#!/bin/bash
# Clear before compiling
make distclean
make imx93_11x11_evk_defconfig
make all -j16
# Create a tmp directory in the current directory to hold the compiled object files
if [ ! -e "./tmp" ]; then
    mkdir tmp
fi
rm -rf tmp/*
```

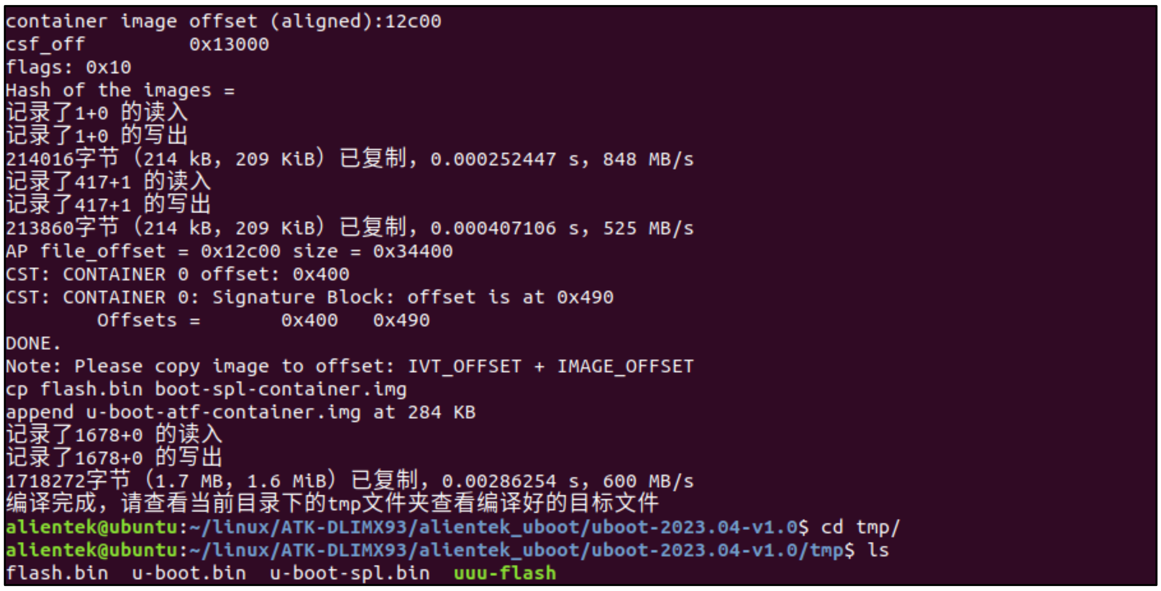
```
# Copy the compiled u-boot.bin and u-boot-spl.bin into the current tmp directory
mv u-boot.bin tmp
mv spl/u-boot-spl.bin tmp

# Compile the flash.bin file
#clean imx-mkimage
# Determines if the file exists and deletes it
echo "进入 imx-mkimage/iMX9"
cd imx-mkimage/iMX9

delete_if_exist() {
    if [ -e "$1" ]; then
        echo "删除文件: $1"
        rm -f "$1"
    fi
}

# List the files to delete
files_to_delete=(
    "flash.bin"
    "head.hash"
    "u-boot.bin"
    "u-boot-spl.bin"
    "u-boot-hash.bin"
    "u-boot-spl-ddr.bin"
    "boot-spl-container.img"
    "u-boot-atf-container.img"
)

# Iterate over the list of files and perform the delete operation
for file in "${files_to_delete[@]}"; do
    delete_if_exist "$file"
done
sync
sync

echo "拷贝 tmp 里的 u-boot.bin 和 u-boot-spl.bin 到 imx-mkimage/iMX9"
cd -
cp tmp/u-boot.bin tmp/u-boot-spl.bin imx-mkimage/iMX9
sync

echo "进入 imx-mkimage 编译 flash.bin"
```

```
cd imx-mkimage
make SOC=iMX9 REV=A1 flash_singleboot
sync
sync
cp iMX9/flash.bin ../tmp
cp iMX9/uuu-uboot ../tmp
echo "编译完成，请查看当前目录下的 tmp 文件夹查看编译好的目标文件"
```

This script contains actions to enable the cross-compiler, compile the U-Boot source code, and package flash.bin. Execute the build.sh script.

```
./build.sh
```

After the final compilation, the relevant files will be stored in the tmp directory, as shown in the following screenshot:

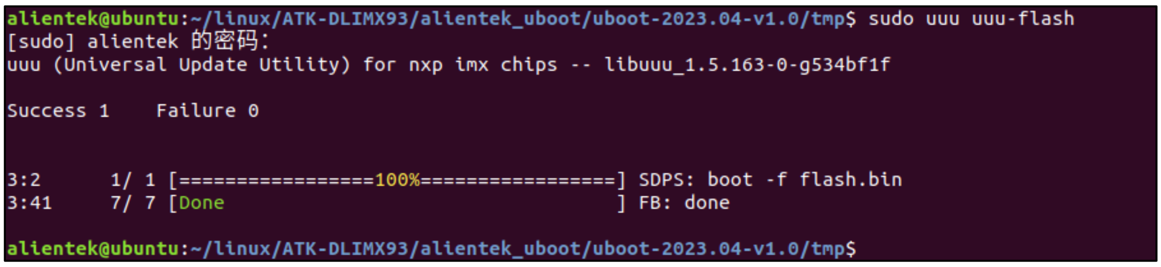


```
container image offset (aligned):12c00
csf_off         0x13000
flags: 0x10
Hash of the images =
记录了1+0 的读入
记录了1+0 的写出
214016字节 (214 kB, 209 KiB) 已复制, 0.000252447 s, 848 MB/s
记录了417+1 的读入
记录了417+1 的写出
213860字节 (214 kB, 209 KiB) 已复制, 0.000407106 s, 525 MB/s
AP file_offset = 0x12c00 size = 0x34400
CST: CONTAINER 0 offset: 0x400
CST: CONTAINER 0: Signature Block: offset is at 0x490
        Offsets =       0x400   0x490
DONE.
Note: Please copy image to offset: IVT_OFFSET + IMAGE_OFFSET
cp flash.bin boot-spl-container.img
append u-boot-atf-container.img at 284 KB
记录了1678+0 的读入
记录了1678+0 的写出
1718272字节 (1.7 MB, 1.6 MiB) 已复制, 0.00286254 s, 600 MB/s
编译完成，请查看当前目录下的tmp文件夹查看编译好的目标文件
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot/uboot-2023.04-v1.0$ cd tmp/
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot/uboot-2023.04-v1.0/tmp$ ls
flash.bin  u-boot.bin  u-boot-spl.bin  uuu-flash
```

Figure 3.1-2 Compilation completed

The flash.bin file is the firmware that can be burned to the development board. This firmware is generated by the imx-mkimage tool and consists of u-boot.bin, u-boot-spl.bin, bl31.bin, tee.bin and other files.

After compiling flash.bin, the user can use the uuu burning tool and the UUU-flash burning script to update the flash.bin burning to the development board. Set the dial switch of the development board to USB burning mode, connect the USB_OTG and USB_TTL interfaces, mount the OTG of the development board to the virtual machine, and open the USB_TTL serial port to view the specific burning information. The burn command is as follows:



```
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot/uboot-2023.04-v1.0/tmp$ sudo uuu uuu-flash
[sudo] alientek 的密码：
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.5.163-0-g534bf1f

Success 1    Failure 0


3:2      1/ 1 [=================100%=================] SDPS: boot -f flash.bin
3:41     7/ 7 [Done                                   ] FB: done

alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_uboot/uboot-2023.04-v1.0/tmp$
```

Figure 3.1-3 Write flash.bin using uuu-flash

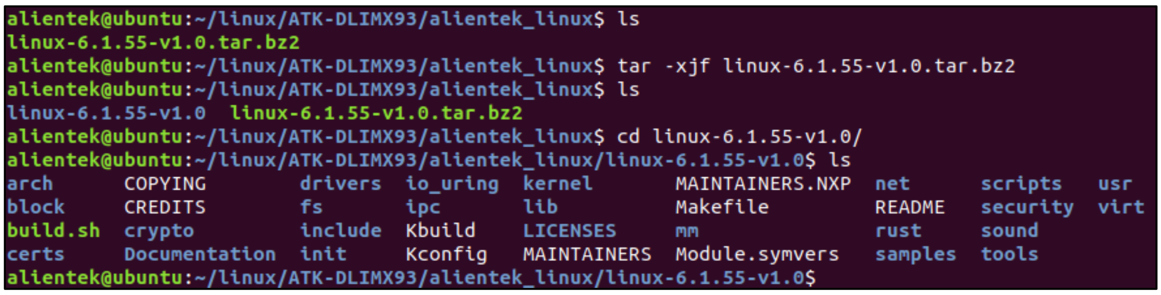
If you don't want to use UUU, you can copy flash.bin to the development board for firmware update. For firmware update, please refer to the development board disc A - Basic information \10_user manual \ [ALIENTEK] ATK-DLIMX93 Firmware Update Reference Documentation.pdf

3.2 Compile factory kernel sources and modules

Create a directory called "alientek_linux" in the ATK-DLIMX93 directory of Ubuntu, and copy the factory linux source package linux-6.1.55-v1.0.tar.bz2 to this directory. Execute the unzip command and navigate to the kernel source directory. (Please confirm the source code version you downloaded, do not directly copy the unzip command below)

```
tar -xjf linux-6.1.55-v1.0.tar.bz2
cd linux-6.1.55-v1.0
```

The result is shown in the following screenshot:



```
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux$ ls
linux-6.1.55-v1.0.tar.bz2
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux$ tar -xjf linux-6.1.55-v1.0.tar.bz2
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux$ ls
linux-6.1.55-v1.0  linux-6.1.55-v1.0.tar.bz2
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux$ cd linux-6.1.55-v1.0/
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux/linux-6.1.55-v1.0$ ls
arch       COPYING        drivers  io_uring  kernel       MAINTAINERS.NXP  net      scripts   usr
block      CREDITS        fs       ipc       lib          Makefile         README   security  virt
build.sh   crypto         include  Kbuild    LICENSES     mm               rust     sound
certs      Documentation  init     Kconfig   MAINTAINERS  Module.symvers   samples  tools
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux/linux-6.1.55-v1.0$
```

Figure 3.2-1 Unzip the linux source code

When the factory source code is compiled, the user does not have to modify anything. Adaptation in ATK - DLIMX93 development board hardware resources of the Linux kernel source device tree path for the arch/arm64 / boot/DTS/freescale/DTS /, The device tree files are imx93-11x11-atk.dts, imx93-11x11-atk-mipi-dsi-5.5-720x1280.dts, imx93-11x11-atk-mipi-dsi-5.5-1080x1920.dts, and imx93-11x 11-atk-mipi-dsi-10.1-800x1280.dts, imx93-11x11-atk-lvds-10.1-1280x800.dts.

To facilitate your compilation, ALIENTEK provides a one-button build script build.sh, you can directly run the script to compile the first time, the script is as follows:

```
# Check if the cross-compile toolchain is present, using aarch64-poky-linux- (gcc-12.3.0)
if [ ! -e "/opt/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux" ]; then
    echo ""
    echo "请先安装正点原子 ATK-DLIMX93 开发板光盘 A-基础资料->05、开发工具->01、交叉编译器
```

```
->fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx93evk-toolchain-6.1-mickledore.sh"
    echo ""
fi

# Use the GCC 12.3.0 cross-compiler in the Yocto SDK to compile the factory Linux source code, you can not specify ARCH, etc., directly execute Make
source /opt/fsl-imx-xwayland/6.1-mickledore/environment-setup-armv8a-poky-linux
#!/bin/bash
# Clear before compiling
make clean

# Configure the defconfig file
make imx_v8_defconfig -j 16

# Start compiling Image
#make menuconfig
make Image -j 16

# Compile the device tree of all kinds of display devices. If the user does not have a screen, it will choose imx93-11x11-atk.dtb to load when starting
make freescale/imx93-11x11-atk.dtb
make freescale/imx93-11x11-atk-mipi-dsi-5.5-720x1280.dtb
make freescale/imx93-11x11-atk-mipi-dsi-5.5-1080x1920.dtb
make freescale/imx93-11x11-atk-mipi-dsi-10.1-800x1280.dtb
make freescale/imx93-11x11-atk-lvds-10.1-1280x800.dtb

# Compiling kernel modules
make modules -j16

# Create a tmp directory in the current directory to hold the compiled object files
if [ ! -e "./tmp" ]; then
    mkdir tmp
fi
rm -rf tmp/*

# Install the compiled module into the tmp directory and remove the module debug information with INSTALL_MOD_STRIP=1
make modules_install INSTALL_MOD_PATH=tmp INSTALL_MOD_STRIP=1 -j16
```

```
40
41 # Delete the source directory in the module directory
42 rm -rf tmp/lib/modules/6.1.55/source
43
44 # Delete the build directory in the module's directory
45 rm -rf tmp/lib/modules/6.1.55/build
46
47 # Navigate to the module directory
48 cd tmp/lib/modules
49
50 # Compressed kernel module
51 tar -jcvf ../../modules.tar.bz2 .
52 cd -
53 rm -rf tmp/lib
54
55 # Copy Image into the tmp directory
56 cp arch/arm64/boot/Image tmp
57
58 # Copy all the compiled device tree files into the current tmp directory
59 cp arch/arm64/boot/dts/freescale/imx93-11x11-atk*.dtb tmp
60 echo "编译完成，请查看当前目录下的 tmp 文件夹中编译好的目标文件"
```

In lines 2 to 7, it checks whether the corresponding cross-compilation toolchain is installed on the current system. If not, it does not run the script and prompts.

Line 10, enables the cross-compilation toolchain.

On line 13, all compilation is cleared, and you can comment this directive if you want.

On line 16, select the configuration file.Running this command will generate a ".config "file in your linux source directory.

On line 19, the kernel menu configuration is enabled, and the default comment is not to run this instruction. It can be opened until the kernel configuration needs to be changed, and then saved as./arch/arm64/configs/imx_v8_defconfig

On line 20, the compilation kernel is run.

On lines 23 through 27, the compile ALIENTEK device tree is run.

On line 30, the compilation kernel module is run.

On lines 33 through 36, a tmp directory is created to hold the required files.

On line 39, the compiled module is packaged into the tmp directory.

On line 42, delete the source directory.

On line 45, delete the build directory.

On line 51, kernel modules are packaged

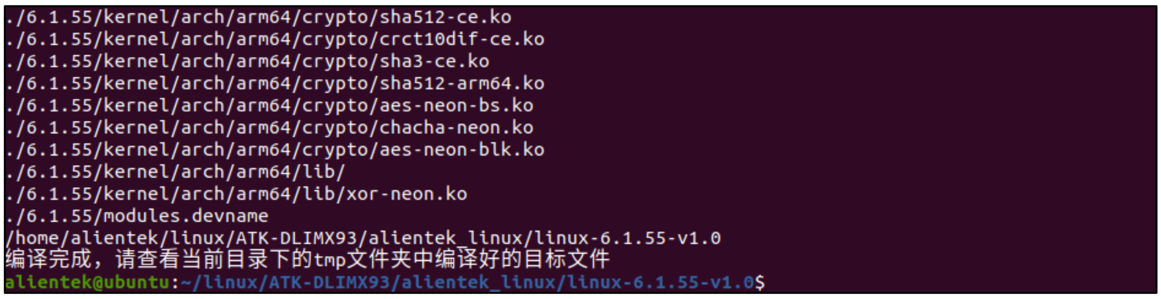
On line 56, copy Image to the tmp directory.

On line 59, copy the device tree required by the ATK-DLIMX93 development board to the tmp directory.

The above script can be run as a single command (must have.config file), you don't need to use this script every time. For example, if I need to clear the kernel compilation, I can copy line 13 of the script and run it. To compile, run the script as follows:

```
./build.sh
```

The following figure shows the compiled result:



```
./6.1.55/kernel/arch/arm64/crypto/sha512-ce.ko
./6.1.55/kernel/arch/arm64/crypto/crct10dif-ce.ko
./6.1.55/kernel/arch/arm64/crypto/sha3-ce.ko
./6.1.55/kernel/arch/arm64/crypto/sha512-arm64.ko
./6.1.55/kernel/arch/arm64/crypto/aes-neon-bs.ko
./6.1.55/kernel/arch/arm64/crypto/chacha-neon.ko
./6.1.55/kernel/arch/arm64/crypto/aes-neon-blk.ko
./6.1.55/kernel/arch/arm64/lib/
./6.1.55/kernel/arch/arm64/lib/xor-neon.ko
./6.1.55/modules.devname
/home/alientek/linux/ATK-DLIMX93/alientek_linux/linux-6.1.55-v1.0
编译完成，请查看当前目录下的tmp文件夹中编译好的目标文件
alientek@ubuntu:~/linux/ATK-DLIMX93/alientek_linux/linux-6.1.55-v1.0$
```

Figure 3.2-2 linux source compilation results

If you need to reconfigure the factory kernel, enter the following command to open the kernel menuconfig menu for configuration selection:

```
make menuconfig
make savedefconfig
cp defconfig arch/arm64/configs/imx_v8_defconfig
```

After modifying the kernel menu, we replace the modified defconfig file with the original factory source configuration file arch/arm64/configs/imx_v8_defconfig

Yes, we can look at the file in the tmp directory. The compiled kernel Image image and device tree file are as follows:

Image name	Mirror image effect	Image file and path
kernel	Linux image	Image
Device tree	Development board basic peripheral functions, such as Ethernet, TF card, etc	imx93-11x11-atk.dtb
	Support MIPI 5.5 inch 720x1280 screen	imx93-11x11-atk-mipi-dsi-5.5-720x1280.dtb
	Support MIPI 5.5 inch 1080x1920 screen	imx93-11x11-atk-mipi-dsi-5.5-1080x1920.dtb
	Support MIPI 10.1 inch 800x1280 screen	imx93-11x11-atk-mipi-dsi-10.1-800x1280.dtb
	Support LVDS 10.1 inch 1280x800 screen	imx93-11x11-atk-lvds-10.1-1280x800.dtb
Kernel module	The modules that need to be loaded when the kernel is booted are in this archive	modules.tar.bz2

The factory system has multiple device trees compatible with multiple screens of the ALIENTEK, imx93-11x11-atk.dtb is the base tree, and other device tree files contain this device tree. Users need to modify the device tree from the imx93-11x11-atk.dts device tree file. And compile the corresponding device tree file to update to the development board for use.

To verify the modified function, the compiled file can be copied to the corresponding directory on the development board system. The Image and dtb files are located in the factory system of the development board /run/media/mmcblk0p1 directory, and the kernel module is located in the factory system of the development board /lib/modules/6.1.55.

For more firmware update operations, please refer to the development board CD A - Basic Information \10_user_manual \ [ALIENTEK] ATK-DLIMX93 Firmware Update Reference Documentation.pdf

3.3 Compile the factory QT GUI comprehensive interface

Please refer to the development board CD A disk - Basic information \10_user_manual \ [ALIENTEK] ATK-DLIMX93 Factory interface QtUI instructions.pdf