

XOS SDK FAQ 问题速查

目录

目录

一) 开发环境、编译问题

- 1) SDK 简介
- 2) 编译环境
- 3) img2simg 安装问题, (仅在: ubuntu24.04 系统出现)
- 4) pip3 安装 xlrd 问题, (仅在: ubuntu24.04 系统出现)
- 5) 报 mkfs.ubifs 命令找不到等错误,
- 6) 关于并行编译的核心数设置
- 7)编译 SPL 时,
- 8) /bin/fakeroot: 175: cpio: not found
- 9) python: not found
- 10) mkstrip (ubuntu24.04 系统出现)
- 11) 个别机器上编译时好时坏, 报如下两种编译 错误。
- 12)如何减少 10XD SDK 的大小?

二) 烧写问题

- 1) 下载工具卡死在 PDL 握手阶段, 如下图
- 2) 下载工具卡死在 SDL 握手阶段, 如下图
- 3) 下载工具卡死在下载 IMG 握手阶段, 如下图

三) 启动相关

- 1) 10XD 的启动模式是怎么设置的?
- 2) 10XD 的 ROMCODE 打印信息代表什么意思?

四) 应用相关

- 1) 新字库的简单测试方法

一) 开发环境、编译问题

1) SDK 简介

请首先阅读，SDK 目录下的 `readme.txt` 文件，简单了解编译方法、文档、预编译固件等

2) 编译环境

参考“[旷明 XOS 开发环境搭建指南.pdf](#)”，安装需要的软件

Plain Text

关于编译主机或服务器的 OS 系统，

推荐: `ubuntu22.04` ，或 `ubuntu20.04`

其他: `ubuntu24.04`，请参考“[Xos 环境搭建-ubuntu2024_2022.pdf](#)”安装开发环境。

不推荐: `ubuntu18.04` 及以下的系统不支持。其他 PC linux 环境未经测试。

3) `img2simg` 安装问题，（仅在: `ubuntu24.04` 系统出现）

错误现象:

```
andrew@andrewpc:~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00$ sudo apt-get install -y fakeroot img2simg mtd-utils
Reading package lists ... Done
Building dependency tree ... Done
Reading state information... Done
Package img2simg is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  android-sdk-libsparse-utils
E: Package 'img2simg' has no installation candidate
```

改用下面:

```
sudo apt install -y android-sdk-libsparse-utils
```

4) pip3 安装 xlrd 问题，（仅在：ubuntu24.04 系统出现）

错误现象：

```
zope.interface 0.1
andrew@andrewpc:~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00$ pip3 install xlrd==1.2.0
error: externally-managed-environment

× This environment is externally managed
╰─> To install Python packages system-wide, try apt install
python3-xyz, where xyz is the package you are trying to
install.

If you wish to install a non-Debian-packaged Python package,
create a virtual environment using python3 -m venv path/to/venv.
Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make
sure you have python3-full installed.

If you wish to install a non-Debian packaged Python application,
it may be easiest to use pipx install xyz, which will manage a
virtual environment for you. Make sure you have pipx installed.

See /usr/share/doc/python3.12/README.venv for more information.

note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this
t the risk of breaking your Python installation or OS, by passing --break-system-packages.
hint: See PEP 668 for the detailed specification.
```

方案 1：使用 python3 虚拟环境（推荐）

- a) python3 -m venv ~/.venv #创建虚拟环境（指定路径，比如 ~/.venv）
- b) source ~/.venv/bin/activate #激活虚拟环境
- c) pip install xlrd==1.2.0 #在虚拟环境中安装 xlrd==1.2.0

d)使用虚拟环境的 Python（可选）

若你的 SDK / 脚本需要调用安装的 xlrd，需使用虚拟环境的 Python 路径：

示例：用虚拟环境的 Python 执行脚本

```
~/.venv/bin/python3 your_script.py
```

e)退出虚拟环境（用完后）

```
deactivate
```

```
/* log***** */
```

```
andrew@andrewpc:~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00$ python3 -m venv
~/.venv
```

```
andrew@andrewpc:~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00$ source
~/.venv/bin/activate
```

```
(.venv) andrew@andrewpc:~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00$ pip install
xlrd==1.2.0
```

```
Collecting xlrd==1.2.0
```

```
Using cached xlrd-1.2.0-py2.py3-none-any.whl.metadata (1.3 kB)
```

```
Using cached xlrd-1.2.0-py2.py3-none-any.whl (103 kB)
```

```
Installing collected packages: xlrd
```

Successfully installed xlrd-1.2.0

```
(.venv) andrew@andrewpc:~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00$ pip list
```

Package Version

```
pip 24.0
```

```
xlrd 1.2.0
```

注意：ubuntu24.04 系统下的编译，需要在安装了 xlrd==1.2.0 的环境下执行，所以，要激活 python3 虚拟环境后，再编译。

```
source ~/.venv/bin/activate #激活 python3 虚拟环境
```

```
cd ~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00
```

```
make distclean #清除
```

```
make project_demo_ld_defconfig #配置项目，生成.config
```

```
make xos -j$(nproc) #编译，用-j$(nproc)更安全
```

5) 报 mkfs.ubifs 命令找不到等错误，

```
code 3227 31 11/26
mkimg path: /home/andrew/prj_sdk/sdk/X-AIOS-LT00/out/qm10xd_linux/mkimg_part/system!
ubifs_lebsize=0x20000
ubifs_miniosize=2048
ubifs_maxlebcnt=148
/home/andrew/prj_sdk/sdk/X-AIOS-LT00/build/script/common/mk-image.sh: line 191: mkfs.ubifs: command not found
/home/andrew/prj_sdk/sdk/X-AIOS-LT00/build/script/common/mk-image.sh: line 200: ubinize: command not found
ubifs_lebsize=0x40000
ubifs_miniosize=2048
ubifs_maxlebcnt=73
/home/andrew/prj_sdk/sdk/X-AIOS-LT00/build/script/common/mk-image.sh: line 191: mkfs.ubifs: command not found
/home/andrew/prj_sdk/sdk/X-AIOS-LT00/build/script/common/mk-image.sh: line 200: ubinize: command not found
ubifs_lebsize=0x40000
ubifs_miniosize=4096
ubifs_maxlebcnt=74
/home/andrew/prj_sdk/sdk/X-AIOS-LT00/build/script/common/mk-image.sh: line 191: mkfs.ubifs: command not found
/home/andrew/prj_sdk/sdk/X-AIOS-LT00/build/script/common/mk-image.sh: line 200: ubinize: command not found
Install img2simg First
sudo apt-get install img2simg
make[4]: Leaving directory '/home/andrew/prj_sdk/sdk/X-AIOS-LT00/output'
```

参考“[旷明 XOS 开发环境搭建指南.pdf](#)”，确认 mtd-utils 工具已安装成功：

```
sudo apt-get install -y mtd-utils
```

然后，检查 mkfs.ubifs 和 ubinize 命令，确认已存在。

6) 关于并行编译的核心数设置

根据核心数目来设置 make 的-j 参数，以并行编译加快速度。通常，-j 的参数可以设置为 CPU 核心数，或者核心数的 2 倍，具体取决于系统资源和项目特性。

比如，设置-j 参数为 核心数目

```
make xos -j$(nproc)
```

查看处理器数量（逻辑核心数）方法：

```
grep -c '^processor' /proc/cpuinfo
```

或

```
lscpu
```

编译机器，推荐 CPU 核心数大于等于 8 个 core；内存大于等于 16GB.

7)编译 SPL 时，

```
COPY    spl/u-boot-spl.bin
MKMOLCHIP spl/u-boot-spl-header.bin
---spl starting merge---
LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:59: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:59: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+)\]', row_arr[bitspos_position])
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:63: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:63: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+):(\d+)\]', row_arr[bitspos_position])
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:110: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:110: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+)\]', row_arr[bitspos_position - 1])
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:114: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl_gen_dmc_reg.txt:114: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+):(\d+)\]', row_arr[bitspos_position - 1])
spl/u-boot-spl-header.bin size: 27860
LT00_DDR2_1200Mbps_QFN88_DDR2_ESMTmerge_bin_spl.bin size: 4004
LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:59: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:59: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+)\]', row_arr[bitspos_position])
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:63: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:63: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+):(\d+)\]', row_arr[bitspos_position])
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:110: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:110: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+)\]', row_arr[bitspos_position - 1])
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:114: SyntaxWarning: invalid escape sequence '['
/home/andrew/work/X-AI05-LT00-V1.0.3/X-AI05-LT00/base/soc/qm10xd/linux/bsp/ramboot/.build/LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl_gen_dmc_reg.txt:114: SyntaxWarning: invalid escape sequence '['
res = re.findall('\[(\d+):(\d+)\]', row_arr[bitspos_position - 1])
spl/u-boot-spl-header.bin size: 27860
LT00_DDR2_1200Mbps_QFN88_DDR2_UNISmerge_bin_spl.bin size: 4016
---spl ending merge---
MC_IMAGE_BUILDER spl/u-boot-spl-header.img
ecc_strength = 8,ecc_step_size = 1024,eraseblokc_size = 20000,page_size = 2048,oob_size=64
---spl image end---
CFECHK u-boot.cfg
```

```
(.venv) andrew@andrewpc:~/work/X-AI05-LT00-V1.0.3/X-AI05-LT00$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.3 LTS
Release:        24.04
Codename:       noble
(.venv) andrew@andrewpc:~/work/X-AI05-LT00-V1.0.3/X-AI05-LT00$ python --version
Python 3.12.3
(.venv) andrew@andrewpc:~/work/X-AI05-LT00-V1.0.3/X-AI05-LT00$
```

8) /bin/fakeroot: 175: cpio: not found

```
Bash
sudo apt install -y cpio
```

9) python: not found

```
andrew@andrewpc:~/work/try1220/X-AIOS-LT00$ make distclean
rm -rf output
rm -rf .config
rm -rf .config.old
rm -rf .config.env
rm -rf .config.env.unset
rm -rf .config.file
rm -rf inc/generated/autoconf.h
rm -rf product/.BoardConfig.mk
rm -rf product/.Kconfig
andrew@andrewpc:~/work/try1220/X-AIOS-LT00$ make project_demo_ld_defconfig
CONFIG_FILE found: product/demo_ld/project_demo_ld_defconfig
start create kconfig link
Using project type: demo_ld
/bin/sh: 1: python: not found
make: *** [Makefile:26: project_demo_ld_defconfig] Error 127
```

可选这三种方法之一解决:

- 1) `sudo apt install python-is-python3` (推荐)
- 2) `alias python='python3'`
- 3) 在 `/usr/bin/python` 和 `/usr/bin/python3` 之间建立符号链接

10) mkstrip (ubuntu24.04 系统出现)

```
4918 '/home/andrew/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00/core/res/res/pcm/1_16_8000.pcm' -> './usr/media/boot.pcm'
4919 ~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00/base/soc/qml0xd/linux/prebuilts/demo_ld/bsp/rootfs ~/work/X-AIOS-LT00
4920 /tmp/tmp.dtTjFtL3L ~/work/X-AIOS-LT00-V1.0.3/X-AIOS-LT00/base/soc/qml0xd/linux/prebuilts/demo_ld/bsp/rootf
4921 cpio: dev/console: Cannot mknod: Operation not permitted
4922 cpio: dev/null: Cannot mknod: Operation not permitted
4923 4019 blocks
4924 /bin/bash: line 3: mkstrip: command not found
4925 .
```

待解决, 暂时不影响运行。(未 strip 掉 lib 目录库的调试信息)

11) 个别机器上编译时好时坏, 报如下两种编译 错误。

错误 1:

```

2122 AS spl/arch/arm/lib/memset.o
2123 CC pdl/arch/arm/lib/cache-cp15.o
2124 ../arch/arm/lib/memset.S: Assembler messages:
2125 ../arch/arm/lib/memset.S:46: Error: thumb conditional instruction should be in IT block -- `stmiaqe ip!,{r1,r3,r8,lr}'
2126 ../arch/arm/lib/memset.S:47: Error: thumb conditional instruction should be in IT block -- `stmiaqe ip!,{r1,r3,r8,lr}'
2127 ../arch/arm/lib/memset.S:48: Error: thumb conditional instruction should be in IT block -- `stmiaqe ip!,{r1,r3,r8,lr}'
2128 ../arch/arm/lib/memset.S:49: Error: thumb conditional instruction should be in IT block -- `stmiaqe ip!,{r1,r3,r8,lr}'
2129 ../arch/arm/lib/memset.S:51: Error: thumb conditional instruction should be in IT block -- `ldmfdeq sp!,{r8,pc}'
2130 ../arch/arm/lib/memset.S:56: Error: thumb conditional instruction should be in IT block -- `stmiane ip!,{r1,r3,r8,lr}'
2131 ../arch/arm/lib/memset.S:57: Error: thumb conditional instruction should be in IT block -- `stmiane ip!,{r1,r3,r8,lr}'
2132 ../arch/arm/lib/memset.S:59: Error: thumb conditional instruction should be in IT block -- `stmiane ip!,{r1,r3,r8,lr}'
2133 ../arch/arm/lib/memset.S:106: Error: thumb conditional instruction should be in IT block -- `stmiane ip!,{r1,r3}'
2134 ../arch/arm/lib/memset.S:108: Error: thumb conditional instruction should be in IT block -- `strbne rl,[ip],#1'
2135 ../arch/arm/lib/memset.S:114: Error: thumb conditional instruction should be in IT block -- `strbne rl,[ip],#1'
2136 ../arch/arm/lib/memset.S:115: Error: thumb conditional instruction should be in IT block -- `strbne rl,[ip],#1'
2137 ../arch/arm/lib/memset.S:117: Error: thumb conditional instruction should be in IT block -- `strbne rl,[ip],#1'
2138 ../arch/arm/lib/memset.S:123: Error: thumb conditional instruction should be in IT block -- `strbit rl,[ip],#1'
2139 ../arch/arm/lib/memset.S:124: Error: thumb conditional instruction should be in IT block -- `strble rl,[ip],#1'
2140 make[9]: *** [../scripts/Makefile.build:325: spl/arch/arm/lib/memset.o] Error 1
2141 make[8]: *** [../scripts/Makefile.spl:432: spl/arch/arm/lib] Error 2

```

错误 2:

```

5 HOSTLD tools/ldgprep
6 HOSTLD tools/oc-image-builder
7 -----> ../arch/arm/gnu/boots-spl.lds ----->
8 LD pdl/arch/arm/gnu/build-in.o
9 CC pdl/arch/arm/mach-molchip/init.o
10 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: unrecognized -march target: armv6
11 arm-molv2-linux-uclibgnueabi-gcc.bs_real: note: valid arguments are: armv4 armv5 armv5te armv6 armv6k armv6z armv6zk armv6z2 armv6-m armv6-m armv7 armv7-a armv7-r armv7-m armv7e-m armv8-a
12 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: missing argument to '-march'
13 make[9]: *** [../scripts/Makefile.build:289: pdl/arch/arm/mach-molchip/init.o] Error 1
14 make[8]: *** [../scripts/Makefile.pdl:369: pdl/arch/arm/mach-molchip] Error 2
15 make[8]: *** Waiting for unfinished jobs....
16 CC pdl/arch/arm/gnu/armv7/cortex_7.o
17 CC spl/arch/arm/mach-molchip/init.o
18 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: unrecognized -march target: armv6
19 arm-molv2-linux-uclibgnueabi-gcc.bs_real: note: valid arguments are: armv4 armv5 armv5te armv6 armv6k armv6z armv6zk armv6z2 armv6-m armv6-m armv7 armv7-a armv7-r armv7-m armv7e-m armv8-a
20 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: missing argument to '-march'
21 make[9]: *** [../scripts/Makefile.build:290: pdl/arch/arm/gnu/armv7/cortex_7.o] Error 1
22 make[8]: *** [../scripts/Makefile.pdl:369: pdl/arch/arm/gnu/armv7] Error 2
23 make[7]: *** [home/AM/shaoqiang/SDE/N-A100-L100-V1.0.3/sde/N-A100-L100/base/soc/qm10xd/linux/bsp/zamboot/Makefile:1469: pdl/a-boot-pbi.bin] Error 2
24 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: unrecognized -march target: armv6
25 make[7]: *** Waiting for unfinished jobs....
26 arm-molv2-linux-uclibgnueabi-gcc.bs_real: note: valid arguments are: armv4 armv5 armv5te armv6 armv6k armv6z armv6zk armv6z2 armv6-m armv6-m armv7 armv7-a armv7-r armv7-m armv7e-m armv8-a
27 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: missing argument to '-march'
28 LD spl/arch/arm/gnu/build-in.o
29 make[9]: *** [../scripts/Makefile.build:290: spl/arch/arm/mach-molchip/init.o] Error 1
30 make[8]: *** Waiting for unfinished jobs....
31 CC spl/arch/arm/mach-molchip/string.o
32 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: unrecognized -march target: armv6
33 arm-molv2-linux-uclibgnueabi-gcc.bs_real: note: valid arguments are: armv4 armv5 armv5te armv6 armv6k armv6z armv6zk armv6z2 armv6-m armv6-m armv7 armv7-a armv7-r armv7-m armv7e-m armv8-a
34 arm-molv2-linux-uclibgnueabi-gcc.bs_real: error: missing argument to '-march'
35 make[9]: *** [../scripts/Makefile.build:290: spl/arch/arm/mach-molchip/string.o] Error 1
36 make[8]: *** [../scripts/Makefile.pdl:429: spl/arch/arm/mach-molchip] Error 2
37 make[7]: *** Waiting for unfinished jobs....

```

原因：多线程编译，竞争传递参数有问题，将 spl 和 pdl 的编译改成单线程编译可以解决该问题，具体修改见下图，去掉下图中的-j 4

```

100 fi
101
102 boot_build:
103 @echo "-----task [boot] build"
104 @if [ -d "$(BSP_DIR)/uboot/" ]; then \
105 make -C $(BSP_DIR)/uboot/$ (UBOOT_VER) ARCH=arm CROSS_COMPILE=$(CROSS) distclean || exit 1; \
106 make -C $(BSP_DIR)/uboot/$ (UBOOT_VER) ARCH=arm CROSS_COMPILE=$(CROSS) $(UBOOT_CONFIG) || exit 1; \
107 make -C $(BSP_DIR)/uboot/$ (UBOOT_VER) ARCH=arm CROSS_COMPILE=$(CROSS) -j 4 || exit 1; \
108 else \
109 echo "no $(BSP_DIR)/boot, ignore boot_build"; \
110 fi
111
112 ramboot:
113 @echo "-----task [ramboot] build"
114 @if [ -d "$(BSP_DIR)/ramboot/" ]; then \
115 make -C $(BSP_DIR)/ramboot ARCH=arm CROSS_COMPILE=$(CROSS) distclean; \
116 make -C $(BSP_DIR)/ramboot ARCH=arm CROSS_COMPILE=$(CROSS) $(UBOOT_CONFIG); \
117 make -C $(BSP_DIR)/ramboot ARCH=arm CROSS_COMPILE=$(CROSS) -j 4; \
118 else \
119 echo "no $(BSP_DIR)/ramboot, ignore built ramboot"; \
120 fi
121

```

12)如何减少 10XD SDK 的大小?

10XD SDK 默认是包含交叉工具链的，可以将交叉工具链安装到系统中。然后删除 SDK tools/toolchain/目录下的工具链。删除工具链后，SDK 大小可以减少约 549MB。

10XD 工具链可以安装到系统如下目录：

Plain Text

```

/opt/toolchain/arm-molv2-linux-uclibgnueabi/bin/arm-molv2-
linux-uclibgnueabi-gcc

```

如使用安装到系统中的交叉工具链则需要做如下修改：

a) 修改 **CMake** 工具链配置：

编辑 build/cmake/toolchain/toolchain_arm-linux-molv2.cmake 文件：

```
Plain Text  
set(tools /opt/toolchain/arm-molv2-linux-uclibcgnueabi)
```

b)修改环境配置脚本：

编辑 build/script/qm10xd/envsetup4qm102d.sh 文件：

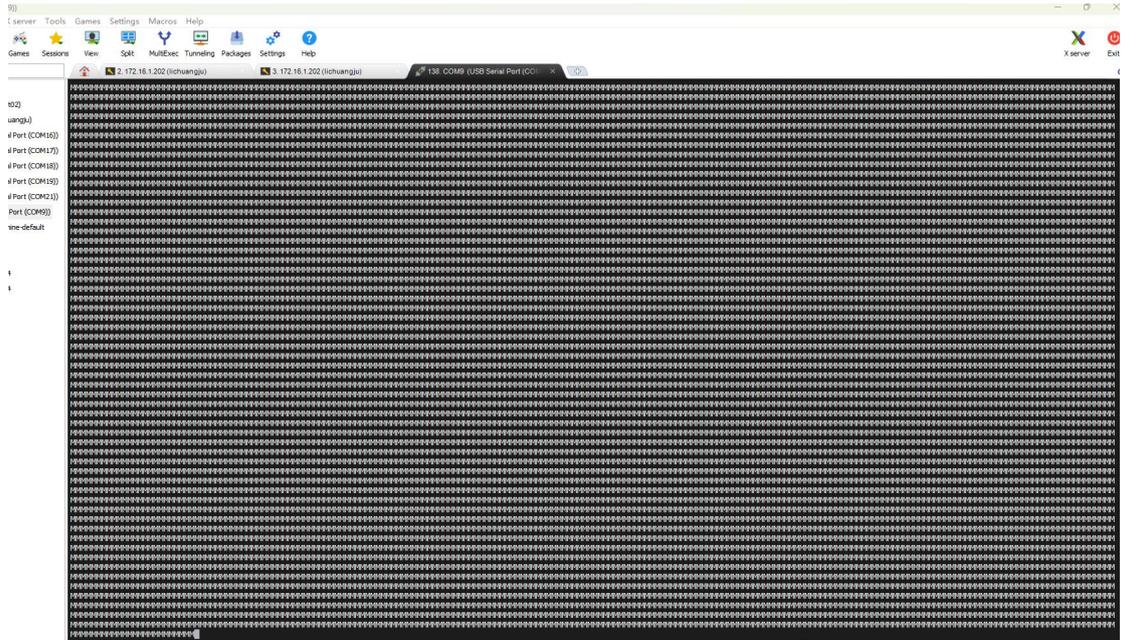
```
Plain Text  
export LT00_ARM_GNUC_BIN_PATH=/opt/toolchain/arm-molv2-linux-  
uclibcgnueabi
```

二) 烧写问题

1) 下载工具卡死在 **PDL** 握手阶段，如下图

时间	通道	级别	
10:53:35	COM1	消息	开始下载。
10:53:35	COM1	消息	开始下载“PDL”……
10:53:35	COM1	消息	开始握手……

原因：板子是否已切换到下载模式，确认方法，用串口工具看是否有如下 MMM 信息输出。



2) 下载工具卡死在 SDL 握手阶段，如下图

时间	通道	级别	消息
11:01:02	COM9	消息	开始下载。
11:01:02	COM9	消息	开始下载“PDL”……
11:01:02	COM9	消息	开始握手……
11:01:02	COM9	提示	握手成功。
11:01:02	COM9	消息	开始连接……
11:01:02	COM9	提示	连接成功。
11:01:02	COM9	提示	修改波特率成功。
11:01:02	COM9	消息	开始下载“E:\FAE\韬智\Image\u-boot-pdl-header.img”……
11:01:02	COM9	提示	下载“E:\FAE\韬智\Image\u-boot-pdl-header.img”成功。
11:01:02	COM9	提示	下载“PDL”成功。
11:01:02	COM9	消息	开始下载“SDL”……
11:01:02	COM9	消息	开始握手……

原因：pdl 下载地址填的不对，It00 pdl 下载地址为：0x27000000

3) 下载工具卡死在下载 IMG 握手阶段，如下图

时间	通道	级别	
11:03:54	COM9	消息	开始下载。
11:03:54	COM9	消息	开始下载“PDL”……
11:03:54	COM9	消息	开始握手……
11:03:54	COM9	提示	握手成功。
11:03:54	COM9	消息	开始连接……
11:03:54	COM9	提示	连接成功。
11:03:54	COM9	提示	修改波特率成功。
11:03:54	COM9	消息	开始下载“E:\FAE\韬智\Image\u-boot-pdl-header.img”……
11:03:55	COM9	提示	下载“E:\FAE\韬智\Image\u-boot-pdl-header.img”成功。
11:03:55	COM9	提示	下载“PDL”成功。
11:03:55	COM9	消息	开始下载“SDL”……
11:03:55	COM9	消息	开始握手……
11:03:56	COM9	提示	握手成功。
11:03:56	COM9	消息	开始连接……
11:03:56	COM9	提示	连接成功。
11:03:56	COM9	消息	开始下载“E:\FAE\韬智\Image\u-boot-sdl.bin”……
11:03:57	COM9	提示	下载“E:\FAE\韬智\Image\u-boot-sdl.bin”成功。
11:03:57	COM9	提示	下载“SDL”成功。
11:03:57	COM9	消息	开始下载“IMG”……
11:03:57	COM9	消息	开始握手……

原因：sdl 下载地址填的不对，It00 sdl 下载地址为：0x40000000.

三) 启动相关

1) 10XD 的启动模式是怎么设置的？

UART下载模式由strapping pin boot0单独控制。

Boot0	启动模式
0	BOOT_MODE
1	UART Download

strapping pin boot1 boot2控制BOOT_MODE。

boot2	boot1	启动模式
0	0	SPI nor flash boot(default)
0	1	SPI nand flash boot
1	0	USB boot
1	1	SD Card boot

2) 10XD 的 ROMCODE 打印信息代表什么意思?

ROMCODE 启动过程中，会打印 LOG 信息反馈执行状态：

SYSTEM Debug

字段	说明
SIT	system init
RBE	执行成功，即将跳转到可执行文件
RVS	RSA 解签成功
MLE	有错误，进入到 main loop
WHP	有预想不到的错误
PKF	计算公钥 hash 时错误
ADF	AES 错误

HSF	Hash efuse state 错误, 比如 efuse ulock
HCF	公钥 hash 与 efuse 中的不匹配
DAF	计算文件 hash 时出错
RDF	RSA 解密错误

SPI NOR Debug

字段	说明
OFB	nor flash boot
NAF	nor 控制器错误
SHF	nor 读取 header 信息出错
SHI	nor 读取到的 header 信息有误, 比如 magic num 不匹配
ND	Nor boot failed
NRF	Nor register read failed
dma cmd err	nor opcode 不是预期内的值
CBR	Nfc buf_p2 busy

SPI NAND Debug

字段	说明
NFB	nand flash boot
NIF	nand identify failed
NFF	nand flash boot failed
CBR	nfc buf_p2 busy

SD Card Debug

字段	说明
SDB	SD Card boot
DIF	SD mci init fail
SIF	SD init fail
SPF	SD read spl fail
SMF	SD magic number fail
SRS	SD read success
SDF	SD Card boot fail

USB Debug

字段	说明
URF	Usb receive data fail
USF	Usb send string fail
UAF	Usb send ack fail
USC	usb set configuration finish
UPO	Usb serial process operated
UAHBE	Usb ahb error interrupt
URD	Reset detected interrupt
URS	USB Reset interrupt
UEM	usb speed enumeration done

四) 应用相关

1) 新字库的简单测试方法

基于性能和内存限制，建议字库精简到 2MB 以内；如果要测试新的字库是否可用，可参考如下步骤，快速确认：

- 1、将字库文件下载下来放置到 `core/res/res/ttf` 目录，替换 `2312_v9.ttf`
- 2、编译模拟器 `make distclean && make project_xxx_defconfig && make xos-sim`
备注：`project_xxx_defconfig`，请更换为正确的工程配置名字
- 3、运行模拟器 `./out/simulator/xos/bin/qxosui`
- 4、观察运行结果是否出现下面截图所示画面内容

可在模拟器的 UI 界面，显示数字字母和汉字，注意这仅是一个示例，实际显示内容跟你运行工程的应用界面有关。



