旷明 XOS SDK 快速上手指南

旷明 XOS SDK 快速

上手指南

部门	
文档编号	
版 本 号	V0.0.1
作　者	QM

更新记录

日期	更新人	版本	备注
2025/12/29	QM	V0.0.1	初稿

目录

1. 引言

1.1 编写目的

本文用于指导用户如何快速上手 LT00/10XD SDK 开发环境，包括 SDK 代码获取、环境搭建、工程编译、配置和烧录等关键步骤，实现能基于此 SDK 快速实现定制化开发。

1.2 预期读者和阅读建议

本文档可提供给客户、研发人员、技术支持工程师和测试工程师使用。

1.3 缩略术语

词语	解释
SDK	Software Development Kit
XOS	旷明统一操作系统

1.4 参考资料

《旷明 XOS 开发环境搭建指南》

《旷明 XOS 烧录升级指南》

2. XOS 简介

XOS 是旷明自主开发的解决方案系统，支持 10XV、10XD、10XH 等芯片。

3. 开发板

XOS 支持 10XV、10XD、10XH 等几种芯片，每种芯片都有配套的开发板，具体见下文介绍。

3.1 10XD 开发板

开发板	存储	封装	DDR	屏幕	说明
Mingberry Pi QM102D_REF	SPI NAND	QFN 88	64MB DDR2	MIPI	WIFI,USB,MIC,SPK, RS232
Mingberry Pi MC3302_REF	SPI NAND	QFN 88	64MB DDR2	MIPI	WIFI,USB,MIC,SPK, RS232

102D_REF_V10 模组是 10XD 的开发板，SDK 的演示程序就是运行在该开发板上的。
102D_REF_V10 模组硬件说明请参阅《102D_REF_V10 模组使用指南》。

3.1.1 购买链接

芯片及开发板 - lingxi.team

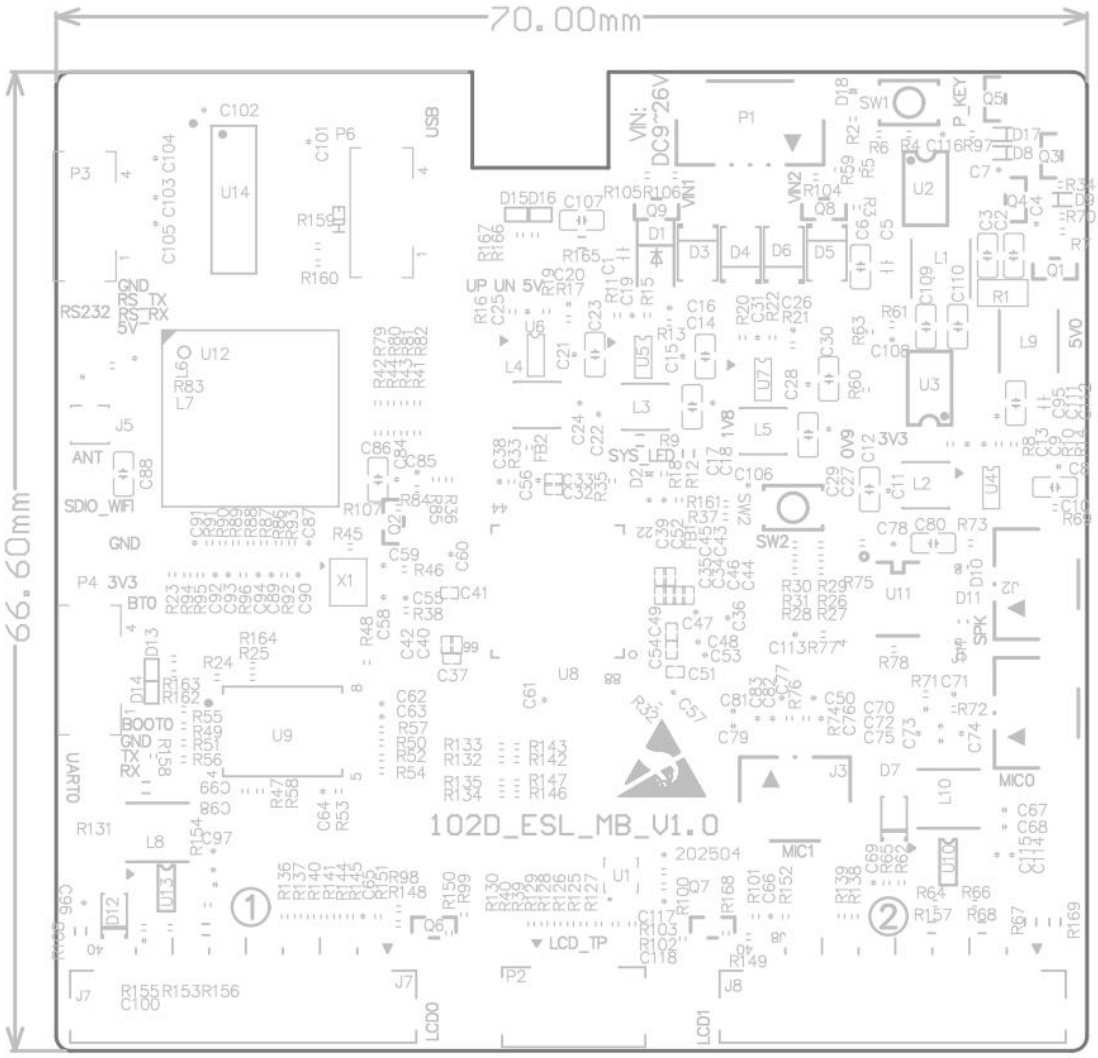
3.1.2 开发板标识

102D_ESL_MB_V1.0

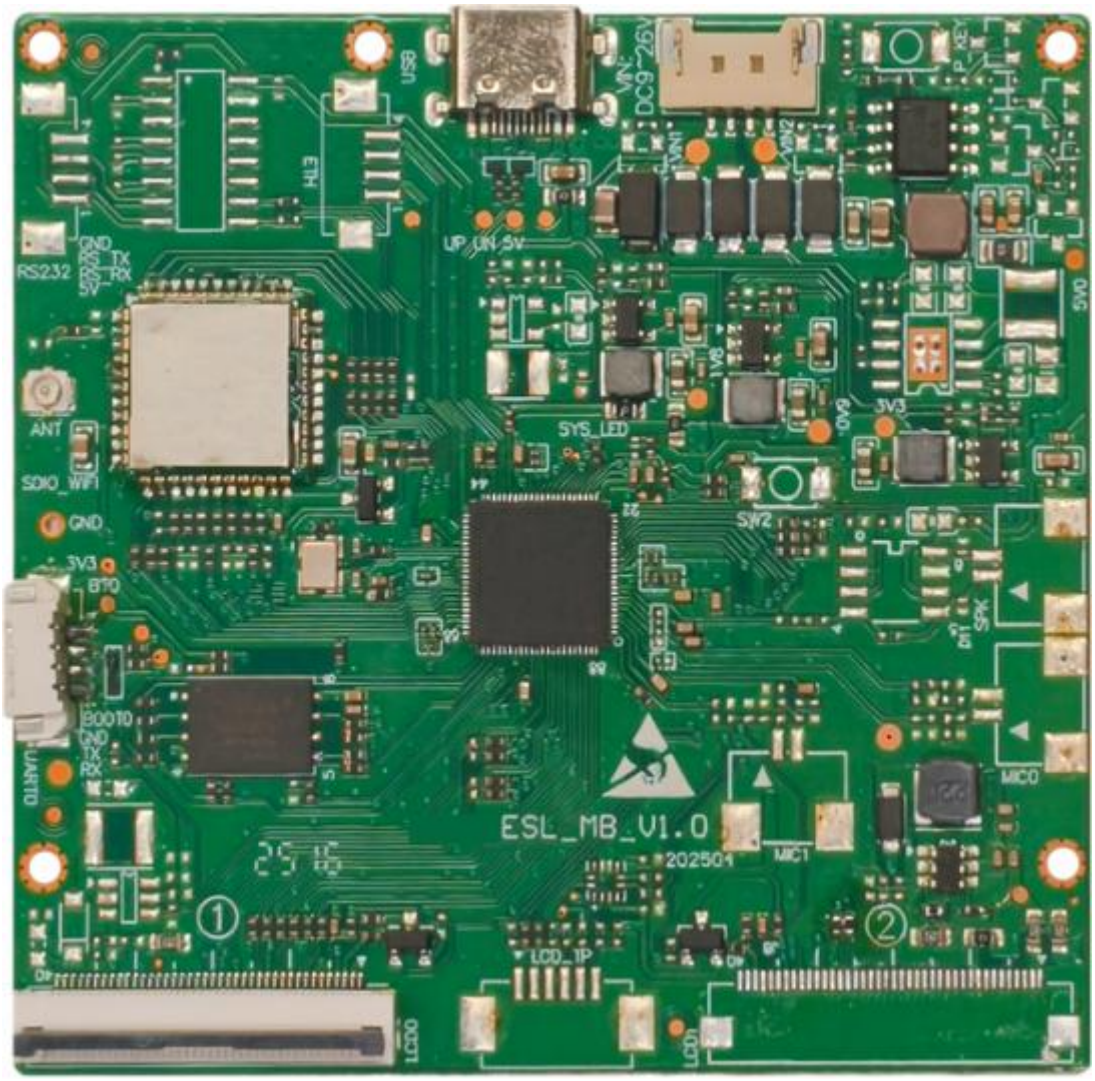
1） 规格

- 10.1 寸屏 800*1280，MIPI 接口（需外购）
- 10.1 寸 TP ，I2C 接口（需外购）
- SPI NAND Flash 1Gb
- USB2.0
- 100M Ethernet
- RS232+UART
- SPK+MIC

2） 器件布局


70.00mm
66.60mm
102D_ESL_MB_V1.0
202504
VIN: DC9~26V
UP UN 5V
RS232
RS_TX
RS_RX
SDIO_WIFI
ANT
BOOT0
UART0
SYS_LED
LCD_TP
LCD0
LCD1
MIC0
MIC1
SPK
P_KEY
USB

3）实物图



4） 原理图&用户指南

QM102D 开发板 - 文档详情 - lingxi.team

5） 模式切换选择

BOOT2(LCD_CLK)	BOOT1(LCD_HSYNC)	BOOT0(SFC_CS1N)	启动模式
0	0	0	SPI NOR flash CS0 boot (default)
0	1	0	SPI NAND flash CS0 boot
1	0	0	USB boot
1	1	0	SD1 boot
X	X	1	UART download

启动模式	**跳线**	**备注**
UART0	UART0 上的 BOOT0 线接 3.3V	下载 uboot
Nor	焊接 R27,R30,去掉 R26,R29	

Nand	焊接 R26,R30,去掉 R27,R29	默认配置
USB	焊接 R27,R29,去掉 R26,R30	
SD Nand	焊接 R26,R29,去掉 R27,R30	

4. SDK 代码获取

请从旷明官方合作渠道获取最新版本的 XOS SDK，XOS SDK 一般包含 docs、downloadtool、prebuilts、sdk 等几个目录。

- docs 目录：SDK 开发使用相关文档
- prebuilts 目录：预编译的 image
- sdk 目录：XOS SDK 软件包，一般以*.tag.gz 压缩包形式提供，类似：X-AIOS-abc-Vx.y.z.tar.gz，其中：
  - abc 是区分不同产品类别，比如 ESL-价签；CAM-相机；
  - x,y,z 区别不同的版本号，比如 x,yz 分别为 1,2,1 时，表示当前 SDK 版本号为 V1.2.1

  具体 SDK 版本以实际获取的 SDK 版本为准。

开发前需要解压 XOS SDK 软件包，命令如下：

```
Plain Text
tar -xvf X-AIOS-abc-Vx.y.z.tar.gz
```

解压后，XOS 软件包的目录结构如下：

```
SQL
~/xos/$ tree -L 1
├── base------ BSP，包括 uboot,kernel,rootfs....
├── build------XOS 编译架构目录
├── core------XOS APP,MultiMedia,FMk 等
├── docs------文档目录
├── inc------SDK .h 文件目录
├── Kconfig -> build/kconf/Kconfig------menuconfig 配置入口文件
├── Makefile -> build/make/Makefile------make 命令入口文件
├── out------编译生成的 image 目录
├── product------项目配置目录
```

```
└── tools------工具目录
```

5. 编译 SDK 代码

XOS SDK 提供了两种 OS 系统供用户选择：Linux 和 RTT。

本章主要介绍如何快速搭建 XOS SDK 编译环境并编译固件。

5.1 搭建开发环境

XOS SDK 开发环境基于 Ubuntu 主机开发，需要安装必要的开发工具及依赖库，再根据目标平台提供的编译工具链，就可以编译 XOS SDK。

5.1.1 安装 linux 系统

Linux 系统可以是一台运行 linux 系统的电脑或是运行在 windows 上的虚拟机。QM 推荐的 Linux 发行版本为 Ubuntu 20.04 LTS 64bit、Ubuntu 22.04 LTS 64bit。

推荐的软硬件配置如下：

磁盘空间	剩余空间大于 50G
内存	16G+
CPU	Quad+ core
系统	Ubuntu 22.04.3 LTS 64bit、Ubuntu 20.04 LTS 64bit

Ubuntu 的安装教程在网上有很多可参考，安装过程中常见的问题都可以在网上找到解决方案。

5.1.2 软件包安装

1）安装开发工具

```SQL
sudo apt-get update
sudo apt-get install -y cmake mtools scons curl git flex bison
gperf build-essential zip unzip
sudo apt-get install -y libsdl2-dev libpng-dev libcurl4-openssl-
dev libssl-dev
```

```
sudo apt-get install -y git-core gnupg flex bison gperf build-
essential zip curl zlib1g-dev
sudo apt-get install -y gcc-multilib g++-multilib libc6-dev-i386
lib32ncurses5-dev
sudo apt-get install -y x11proto-core-dev libx11-dev lib32z-dev
sudo apt-get install -y libgl1-mesa-dev libxml2-utils xsltproc
libncurses5 liblz4-tool
sudo apt-get install -y ccache gdb u-boot-tools device-tree-
compiler
sudo apt-get install -y fakeroot
sudo apt-get install -y mtd-utils
sudo apt-get install -y android-sdk-libsparse-utils img2simg
sudo apt-get install -y cpio
```

2）安装 Python3 及依赖库

Xos 的 uboot 编译需要使用 Python3，并需安装 xlrd 特定的依赖库：

```
Plain Text
sudo apt-install -y python3 python3-pip
pip3 install xlrd==1.2.0
```

注意：由于 Ubuntu 24.04 上无法安装 xlrd==1.2.0，所以使用 openpyxl 代替，就需要安装 openpyxl，安装命令如下：

```
Plain Text
sudo apt install -y python3-openpyxl
```

3）默认 shell 配置

编译脚本默认使用的是 bash，要求系统的默认 shell 为 bash。Ubuntu 默认 shell 一般为 dash，可通过如下命令来修改：

```
Plain Text
# sudo dpkg-reconfigure dash
#在弹出的界面选择<NO>
```

4）一键安装脚本

为方便用户搭建开发环境，XOS SDK 提供也提供一个一键安装脚本 install_xos_env.sh。

[install_xos_env.sh]

有可用软件源的网络环境中，Ubuntu 系统是用 apt-get 工具完成软件安装，“一键安

装” 脚本也需要用到此 apt-get 工

具。

在命令行执行一键安装脚本的方法：

```
Plain Text
sudo ./install_xos_env.sh
```

install_xos_env.sh 会自动检查当前系统的版本、环境，在软件源可以正常访问的情况下，逐个安装 XOS SDK 需要的软件工具，安装成功后会有提示信息：

install_xos_env.sh 目前已支持的系统有：

- Ubuntu 20.04、22.04、24.04

5.1.3 SDK 编译工具链

SDK 编译工具链，都是在 XOS SDK 中提供，工具链在 SDK 的目录：tools/toolchain 下，根据芯片不通过，工具链可能不同。

5.2 编译 XOS

5.2.1 编译命令

- 完整编译

```
Plain Text
$ make distclean-------清除编译
$ make project_xxx_defconfig-------选择项目配置文件
$ make xos -------开始全编译
编译成功后，编译结果在：out/xxxxxxx_linux/qmimages 目录下，xxxxxx 是该 project 选用的芯片名字，比如 qm10xv，qm10xd，qm10xh 等
qmimages 目录下，一般包括可烧录的 boot、刷机包、编程器烧录固件、以及用于调试 backup***.tar.gz 文件备份包。
关于 boot 烧录、刷机、量产时编程器烧录的使用方法，参考《旷明 XOS 烧录升级指南》等专项文档。
```

- 模块编译

```
Plain Text
```

(1). 编译 uboot
编译命令：*$ make xos-uboot*
编译产物：*spl, uboot image*
(2). 编译 kernel
编译命令：*$ make xos-kernel*
编译产物：out/qm10xd_linux/image/zImage-dtb
(3). 编译 APP
编译命令：*$ make xos-qxosui*
编译产物：output/qxosui.elf
(4). 打包文件系统和升级包
编译命令：*$ make xos-package*
编译产物：out/qm10xd_linux/qmimages 刷机脚本，文件系统，OTA 升级包等

- 其他命令

```
Plain Text
$ make menuconfig-------menuconfig 配置
$ make savedefconfig--------.config 配置生成 defconfig
$ make distclean-------清除编译
$ make xos -j <jobs> -------多线程编译,例如:make xos -j 16
```

备注：全编译完成后，才可以进行模块编译。

5.2.2 编译烧写 image

在 SDK 根目录，运行如下命令：

```
Plain Text
make distclean                                  #clean 清除 Q
make project_XXXXX_defconfig                    #配置项目，生
成.config
make xos -jxx                                   #编译，其中 xx 是选择编
译使用的 cpu 数
```

其中项目配置命令是根据你要编译的配置项目来定，如编译 demo_ld 的配置命令为：make project_demo_ld_defconfig.

XOS SDK 默认项目对应的配置命令见下表：

XOS SDK	编译配置	备注

X-AIOS-10XD	make project_demo_ld_defconfig	
X-AIOS-10XD-PUB	make project_demo_ld_defconfig	
X-AIOS-10XV	make project_camera_defconfig	

5.2.3 编译 APP 应用模拟器版本

在 SDK 根目录，运行如下命令：

```
Plain Text
$ make distclean-------清除编译
$ make project_xxx_defconfig-------选择项目配置文件
$ make xos-sim-------开始编译模拟器版本
```

其中，项目配置的使用同上。

模拟器编译结果在：out/simulator/xos/bin/qxosui
可在 X86 的 ubuntu 下直接运行，或用 gdb 加载运行，可用于快速开发 APP 应用。

Gdb 加载运行示例：
gdb ./out/simulator/xos/bin/qxosui #加载
r #运行
q #退出
补充说明：1、当编译板子固件、和编译模拟器交替执行时，先删除或重命名备份根目录下的 out 目录，再执行编译命令

5.2.4 10XD 编译模块

在 LT00/QM10XD 芯片平台上支持 qxosui、uboot、kernel 模块编译，执行模块编译后

会将下载需要的文件打包放置到 out/qm10xd_linux/qmimages 目录。

需要注意这里的模块编译前需要先执行全系统的编译，模块编译需要在全编译的基础上再单独执行模块的编译命令。

1. 编译 qxosui

在 SDK 根目录，运行如下命令：make xos-qxosui

2. 编译 uboot

在 SDK 根目录，运行如下命令：make xos-uboot

3. 编译 kernel

在 SDK 根目录，运行如下命令：make xos-kernel

6. 烧录

6.1 开发烧录方法

编译好的固件，烧录 boot 及刷机方法参考文档《旷明 XOS 烧录升级指南》。这只简单介绍将 SDK 软件程序烧写到硬件板子的步骤:

1）用工具烧写 boot

将硬件开发切换到下模式，然后通过工具烧写 boot（10xd 为 spl 和 uboot）

2）通过 u 盘烧写 kernel、rootfs、data 等其他分区。

将硬件板子切换到正常启动模式，通过 usb 转接线插上 u 盘，给板子上电，进入 uboot 命令行（上电时不停的向串口输入 enter 键）。然后，输入 run update 命令，就开始 U 盘自动烧写。

10XD U 盘烧写的 log 如下:

```
RESET TYPE :Soft reset
SPL (Jan 05 2026 - 16:47:32 +0800)
SPINAND : F35SQA001G is found


U-Boot 2017.09 (Jan 05 2026 - 16:47:20 +0800)

NAND:  *-End SPI Nand flash driver probe.
128 MiB
Unknown command 'sf' - try 'help'
Unknown command 'sf' - try 'help'
ret=1 run command:sf read 0x41000000 0x300000 0x800

show welcome done !
skip update fw... 1
boot#
uboot#
uboot#
uboot#
uboot#
uboot#
uboot#
uboot#
uboot#
uboot#run update
starting USB...
USB0:   Core Release: 4.00a
scanning bus 0 for devices... 2 USB Device(s) found
       scanning usb for storage devices... 1 Storage Device(s) found
reading script.ini
1398 bytes read in 142 ms (8.8 KiB/s)
## Executing script at 41000000
script data:env default -a;
fatload usb 0:1 40008000 u-boot-spl-header.img;nand erase.part boot-spl;nand write 40008000 boot-spl 8000;
fatload usb 0:1 40008000 u-boot.bin;nand erase.part boot-uboot;nand write 40008000 boot-uboot 9b790;
fatload usb 0:1 40008000 logo.img;nand erase.part logo;nand write 40008000 logo 73000;
fatload usb 0:1 40008000 misc.img;nand erase.part misc;nand write 40008000 misc 1000;
fatload usb 0:1 40008000 recovery.img;nand erase.part recovery;nand write 40008000 recovery 27b8c4;
fatload usb 0:1 40008000 ## Resetting to default environment
reading u-boot-spl-header.img
32768 bytes read in 159 ms (201.2 KiB/s)

Erasing at 0x0 -- 100% complete.
OK

NAND write:  32768 bytes written: OK
reading u-boot.bin
636816 bytes read in 655 ms (949.2 KiB/s)

Erasing at 0xe0000 -- 100% complete.
OK

NAND write:  636816 bytes written: OK
reading logo.img
```

6.2 量产烧录方法

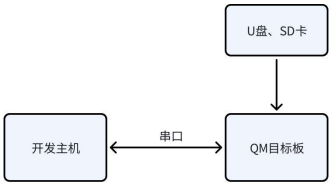
两种量产方式：

a. 工具烧录 boot，再加刷机模式，需夹具； （适用于 spi nor、spi nand 和 emmc）

b. 编程器烧录完整固件到 flash，然后再贴片； （适用于 spi nor 和 spi nand）

7. 调试

由于嵌入式单板的资源有限，不能再单板上运行开发和调试工具，通常需要交叉编译调试的方式进行开发和调试，即“宿主机+目标板”的形式。一般情况下宿主机和目标板的处理架构不相同。宿主机需要建立适合于目标板的交叉编译环境。程序在宿主机上经过“编译-连接-定位”得到可执行文件。将可执行文件烧写到目标板中，然后在目标板上运行，典型的开发环境如下所示：



XOS 开发一般使用 UART0 作为调试串口，串口的波特率一般为：115200

7.1 使用 GDB 调试 APP 应用模拟器版本

XOS SDK 支持通过 GDB 调试模拟器版本 APP 应用程序，方便用户调试 APP 应用程序。具体命令如下：

```
Plain Text
#gdb ./out/simulator/xos/bin/qxosui
```

输出信息如下：

```
Python
qm@build01:~/demo-test$ gdb ./out/simulator/xos/bin/qxosui
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.2) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./out/simulator/xos/bin/qxosui...
(gdb)
```

可以通过 b 命令设置断点

```
Plain Text
```

```
(gdb) b main
Breakpoint 1 at 0xea322
```

执行 run 命令，运行程序

```
Plain Text
(gdb) r
Starting program: /home/lichuangju/demo-
test/out/simulator/xos/bin/qxosui
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-
gnu/libthread_db.so.1".

Breakpoint 1, 0x000055555563e322 in main ()
(gdb)
```

关于更多 GDB 的使用方法，参考 GDB 官方文档。

7.2 CoreDump 调试

Coredump 是一款调试复杂软件问题的核心工具。CoreDump 通过操作系统在进行异常终止时，生成的内存快照文件，记录程序崩溃瞬间的寄存器状态、堆栈信息、内存布局等关键数据。

XOS SDK 默认支持 CoreDump 功能。

执行下列命令解除系统对 CoreDump 文件大小的限制：

```
Plain Text
ulimit -c unlimited
```

qxosui 运行过程中，当发生段错误时会自动在/data 目录下生成完整的 CoreDump 文件。

```
Plain Text
Segmentation fault (core dumped)
```

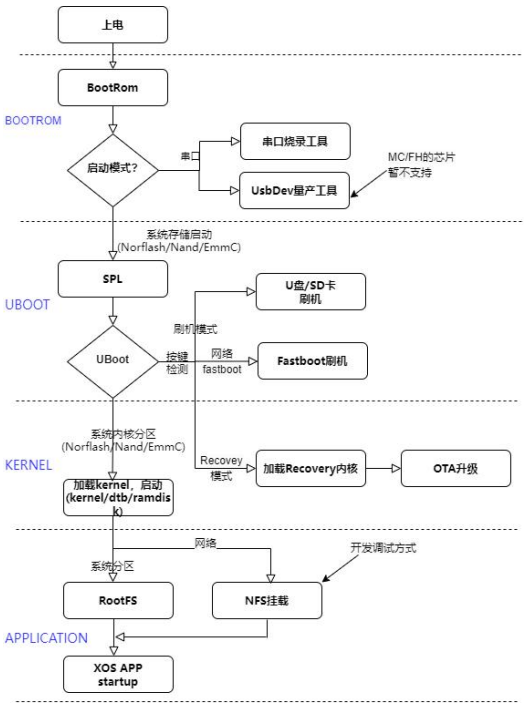
将 CoreDump 文件 core.xxxx 通过 u 盘或 adb 复制到开发主机。在开发主机上运行如下命令进行调试：

```
Plain Text
gdb ./out/qm10xd_linux/xos/bin/qxosui core.xxxx
```

关于更多 GDB 的使用方法，参考 GDB 官方文档。

8. 开机流程

XOS 开机流程图如下：



9. 文档资源

9.1 随 SDK 发布的文档

SDK 发布包里有 SDK 相关的使用说明文档，在 SDK 的 docs 目录中。

10. FAQ

在使用 SDK 时，遇到各种问题，请先阅读该文档和 FAQ 文档《XOS SDK FAQ 问题》。