

SDK 开发环境

用户指南

文档版本 V1.1

发布日期 2025-06-03

前言

概述

本文档介绍 sdk 开发环境及其搭建、u-boot、内核和根文件系统的烧写，以及创建网络开发环境和如何启动 Linux 开发应用程序。

本文档主要让客户快速熟悉 sdk 开发流程。



说明

未经特殊说明，以 XM7606V13 指代 XM7606V1 系列所有芯片，以 XM7206V11A 指代 XM7206V1 系列所有芯片。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。
2025-06-03	V1.1	新增 XM7206V1 系列芯片相关说明。

目 录

前 言	i
1 开发环境	1
1.1 嵌入式开发环境	1
1.2 开发环境	1
1.3 搭建开发环境	3
1.3.1 发布包使用的 Linux Server 版本	3
1.3.2 网络环境搭建	3
1.3.3 软件包安装	3
1.3.4 配置交叉编译工具路径	5
1.3.5 安装 SDK	5
1.3.6 开发板配套串口小板驱动安装	5
2 SDK	8
2.1 SDK 包目录结构	8
2.2 配置 SDK	10
2.3 整体编译 SDK	11
2.4 烧写镜像	11
2.5 运行 SDK	11
2.5.1 单板网络配置	11
2.5.2 KO 加载	12
2.5.3 程序运行	12
3 应用程序开发简介	13
3.1 编写代码	13
3.2 运行应用程序	13

插图目录

图 1-1 嵌入式开发图例..... 1

图 1-2 开发环境 2

表格目录

表 1-1 开发环境的各部分软件描述.....	2
-------------------------	---

1 开发环境

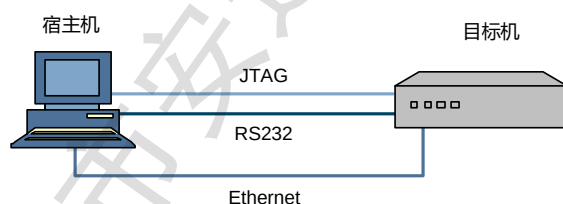
1.1 嵌入式开发环境

由于嵌入式单板的资源有限，不能在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+目标机（评估板）”的形式。宿主机和目标机一般采用串口连接，也可同时通过网口或者 JTAG 连接，如图 1-1 所示。

宿主机和目标机的处理器一般不相同。宿主机需要建立适合于目标机的交叉编译环境。程序在宿主机上经过“编译 - 链接 - 定位”得到可执行文件。通过一定的方法将可执行文件烧写到目标机中，然后在目标机上运行。

目标机上的 Bootloader 启动后，目标机中的操作信息通过串口或者网口输出到宿主机上显示。在宿主机上的控制台中输入命令，可以控制目标机。

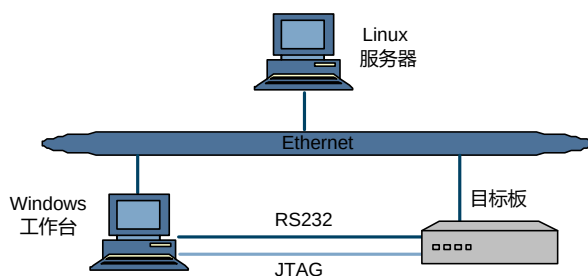
图1-1 嵌入式开发图例



1.2 开发环境

开发环境通常包括 Linux 服务器、Windows 工作台和 DEMO（目标板），三者同处于一个网络中，如图 1-2 所示。

图1-2 开发环境



在 Linux 服务器上建立交叉编译环境，Windows 工作台通过串口和网口与 XM7606V13 单板连接，开发人员可以在 Windows 工作台进行程序开发或者远程登录到 Linux 服务器进行程序开发。各部分具体软件介绍如表 1-1 所示。



说明

开发环境中使用了 Windows 工作台，实际上很多工作也可以在 Linux 服务器上完成，如使用 minicom 代替超级终端等，用户可自行选择。

表1-1 开发环境的各部分软件描述

软件		描述
Windows 工作台	操作系统	Windows10。
	应用软件	putty、超级终端、tftp 服务器、DS-5 等软件。
Linux 服务器	操作系统	无特别要求，可为 Ubuntu、Redhat、Debian 等。内核版本支持 2.6.18 及以上版本。安装时建议选择完全安装。
	应用软件	NFS、telnetd、samba、vim、arm 交叉编译环境（gcc 版本 12.2.0）等。 其他应用软件根据具体开发需要而定，通常系统都已默认安装，只要适当配置即可。
BOARD	引导程序	U-boot。
	操作系统	Linux：Linux 内核基于 Linux 标准内核 5.10.130 版本移植开发，根文件系统基于 busybox 1.36.1 版本制作而成。
	应用软件	包含 telnetd、gdb server 等 Linux 常用命令。

软件		描述
	程序开发库	glibc-2.37 版本。

1.3 搭建开发环境

推荐用户使用 64 位 Linux 服务器，本开发包在 32 位 Linux 服务器、较老版本的 Linux 服务器、偏冷门的 Linux 服务器上可能存在未知的兼容性问题。

推荐的硬件配置如下：

- CPU Intel(R) Xeon(R) CPU E5-2450 0 @ 2.10GHz 或更好 CPU
- DDR: >= 16GB
- Hard disk >= 600GB
- Gigabit Ethernet
- OS: Ubuntu 18.04 64bit

1.3.1 发布包使用的 Linux Server 版本

本示例安装如下版本 Linux 系统，本开发包默认均以此 Linux 系统进行编译：

```
Linux version 4.15.0-192-generic (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)) #203-Ubuntu SMP Wed Aug 10 17:40:03 UTC 2022
```



说明

ubuntu 系统安装步骤略。

1.3.2 网络环境搭建

请用户自行配置网络，并安装 nfs, samba, ssh 等网络组件。

1.3.3 软件包安装

操作系统安装好后，且用户已自行配置好网络环境，则可继续如下步骤完成相关软件包的安装：

步骤 1 配置默认使用 bash。

执行 `sudo dpkg-reconfigure dash` 选择 no

步骤 2 安装软件包。

执行: `sudo apt-get install make libc6:i386 lib32z1 lib32stdc++6 zlib1g-dev libncurses5-dev ncurses-term libncursesw5-dev g++ u-boot-tools:i386 texinfo texlive gawk libssl-dev openssl bc`

步骤 3 创建/etc/ld.so.preload 文件, 并执行 `echo "" > /etc/ld.so.preload`, 以解决 64bit linux server 上某些第三方库编译失败的问题。

步骤 4 安装 gperf 工具, 以解决 eudev-3.2.7 工具编译问题。下载网址:

<http://ftp.gnu.org/pub/gnu/gperf/gperf-3.1.tar.gz>;

```
tar -xzf gperf-3.1.tar.gz
```

```
cd gperf-3.1
```

```
./configure
```

```
make
```

```
make install
```

步骤 5 mtd-utils2.0.2 依赖以下几个库, 以 ubuntu 为例, 安装方式请参考下面命令:

```
sudo apt-get install zlib1g-dev liblz2-dev uuid-dev pkg-config
```

由于 mtd-utils2.0.2 通过 pkg-config 工具检查各个库是否正常安装, 因此请参考如下方式设置 pkg-config 工具搜索路径:

```
export PKG_CONFIG_PATH="$PKG_CONFIG_PATH:/usr/lib/x86_64-linux-gnu/pkgconfig"
```

mtd-utils2.0.2 依赖于 automake1.15.1 版本, 请参考如下方式安装 automake1.15.1 版本:

- 删除原有 automake 版本

```
sudo apt-get autoremove automake
```

- 下载 1.15.1 版本

- Windows 系统中使用浏览器访问 automake 工程 url 并下载对应源码包:

```
https://lists.gnu.org/archive/html/info-gnu/2017-06/msg00007.html;
```

- linux 系统中可以使用下面命令下载:

```
wget ftp://ftp.gnu.org/gnu/automake/automake-1.15.1.tar.gz
```

- 编译安装 automake1.15.1

```
tar zxvf automake-1.15.1.tar.gz
```

```
cd automake-1.15.1/
```

```
./configure
```

```
make
```

```
make install
```

----结束

1.3.4 配置交叉编译工具路径

须知

使用从网络等渠道得到的交叉编译工具可能存在与使用的内核并不配套，造成开发过程中出现一些不可预料的问题。

XM7606V13 发布包提供编译工具链 arm-gcc12.2.0-linux-gnueabi、aarch64-gcc12.2.0-linux 和 riscv-gcc10.2.0-linux。

XM7206V11A 发布包提供编译工具链 arm-gcc12.2.0-linux-uclibceabi 和 riscv-gcc10.2.0-linux。

配置步骤如下：

步骤 1 进入 sdk 目录。

执行如下命令：

```
cd sdk
```

步骤 2 配置工具链路径。

```
source ./build/env.sh
```

即可将 sdk 中的 tools/linux/toolchains 目录下的编译路径加入到 PATH 环境变量中，编译器即可在 sdk 中使用。

----结束

1.3.5 安装 SDK

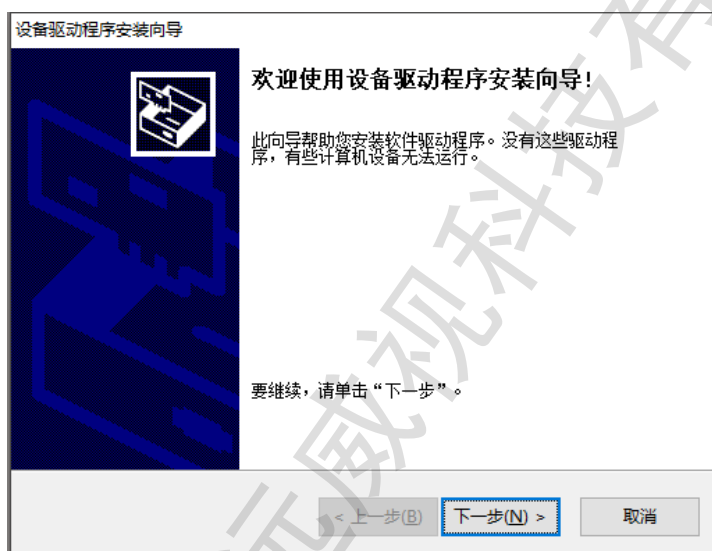
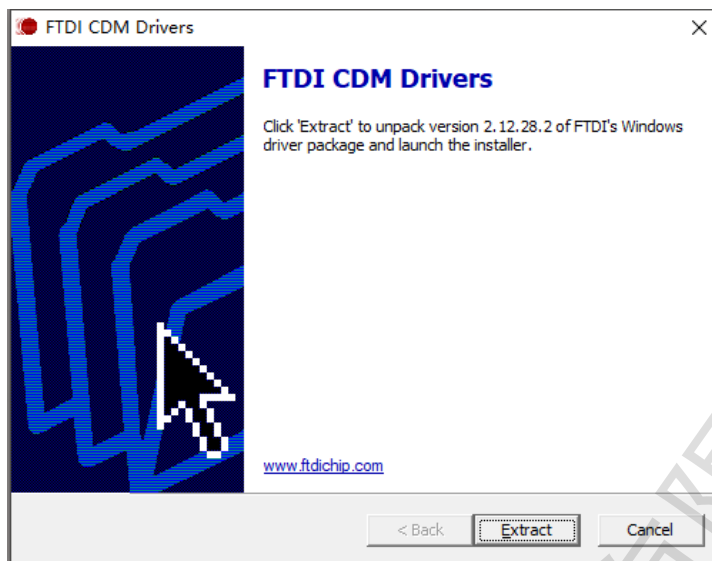
源码包中包含完整的 SDK，无需安装即可使用。

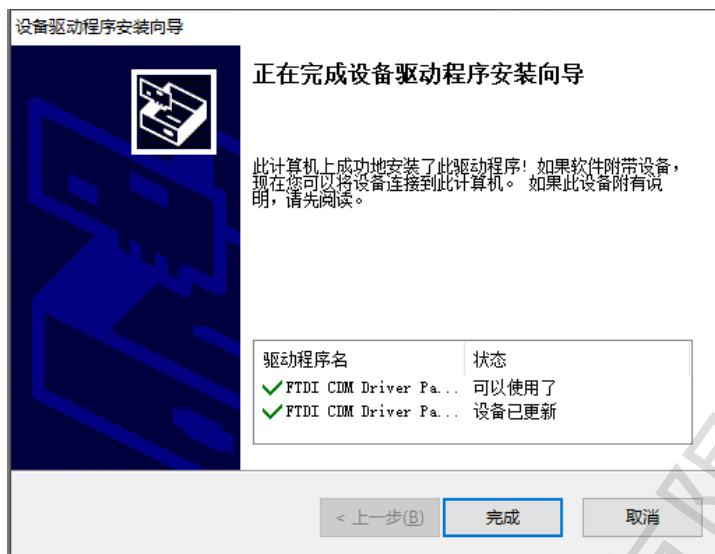
1.3.6 开发板配套串口小板驱动安装

FT232 Windows 串口驱动位于以下路径：

```
sdk/tools/windows/burning-tools/ftdi-driver/FTDI-CDM-Driver.exe
```

安装步骤如下图所示：





2 SDK

2.1 SDK 包目录结构

```
-- build
-- configs
|  |-- xm7xxx
|  |  |-- prebuilts
|  |  `-- reg
|  `-- xm7xxx
|      |-- prebuilts
|      `-- reg
-- open_source
|  |-- busybox
|  |-- can
|  |-- deflate
|  |-- dosfstools
|  |-- e2fsprogs
|  |-- ethtool
|  |-- eudev
|  |-- ext4_utils
|  |-- flatcc
|  |-- gdb
|  |-- gzip
|  |-- libuuid
|  |-- lz4
|  |-- lzma
|  |-- lzo
|  |-- mtd-utils
|  |-- ncurses
|  |-- squashfs
|  |-- toybox
|  |-- util-linux
|  |-- xz
|  |-- yaffs2
|  |-- zlib
```

```

|-- zstd
|-- sample
|   |-- audio
|   |-- cipher
|   |-- common
|   |-- fb
|   |-- ir
|   |-- ircut
|   |-- live_rtsp_server
|   |-- npu
|   |-- pq_bin
|   |-- region
|   |-- scene_auto
|   |-- tde
|   |-- vdec
|   |-- venc
|   |-- vgs
|   |-- vio
|-- vo
-- source
|   |-- bootloader
|   |   |-- product
|   |   |-- secureboot
|   |   |-- u-boot-2020.01
|   |-- gmp
|   |   |-- drv
|   |   |   |-- include
|   |   |   |-- init
|   |   |   |-- ir
|   |   |   |-- lsadc
|   |   |   |-- obj_linux-5.10.y
|   |   |-- osal
|   |   |-- include
|   |   |-- lib
|   |   |   |-- arm-gcc12.2.0-linux
|   |-- usr
|   |   |-- dump_tools
|   |   |-- isp
|-- initramdisk
|   |-- scripts
|   |-- toybox
|-- kernel

```

```

| | `-- linux-5.10.y
| | |-- rootfs
| | |-- busybox
| | |-- can-utils
| | |-- canutils
| | |-- dosfstools
| | |-- e2fsprogs
| | |-- ethtools
| | |-- eudev
| | |-- gdb
| | |-- mtd-utils
| | |-- regtools
| | `-- scripts
| `-- trusted-firmware
|     `-- trusted-firmware-a-2.8
|-- tools
|   |-- linux
|   |-- aq_board
|   |-- kconfig
|   |-- pq_board
|   |-- toolchains
|   |-- `-- utils
|   `-- windows
|       |-- aq_tools
|       |-- burning_tools
|       `-- pq_tools
|-- `-- version_config

```

2.2 配置 SDK

配置 SDK 的操作步骤如下：

步骤 1 配置环境变量。

```
source ./build/env.sh
```

步骤 2 拷贝 cfg.mk 配置文件，如：

```
cp configs/xm7606v13/xm7606v13rbb_linux-5.10_cfg.mk ./cfg.mk
```

步骤 3 用户通过 “make menuconfig” 进行 SDK 配置。

```
make menuconfig
```


步骤 4 选择需要的配置项。

步骤 5 选择完毕后，保存并退出。

----结束

说明

每个芯片不同板型的配置文件对应不同的 DDR 类型，参考 `cfg.mk` 中的 `CONFIG_XMEDIA_BOOT_REG_NAME` 配置项，请用户自行选择与硬件对应的配置编译。

2.3 整体编译 SDK

配置保存后，在 `sdk` 根目录下，使用如下命令编译 `sdk`。

```
make build -j
```

2.4 烧写镜像

参考文档《FastBurn 工具使用指南》。

2.5 运行 SDK

镜像烧写完成后，重新上下电，即可进入板端运行环境，可以运行板端程序，下面分别介绍一些板端常用的开发环境配置。

2.5.1 单板网络配置

如果单板上有网口，可以采用如下命令配置单板网络便于开发：

```
ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx // 配置单板 MAC 地址
```

```
ifconfig eth0 xx.xx.xx.xx netmask xx.xx.xx.xx //配置单板 IP 及掩码
```

```
route add default gw xx.xx.xx.xx // 配置路由
```

配置完成后，可以 `ping` 一下服务器 IP，确保网络畅通，之后可以通过 `nfs` 挂载到服务器，便于开发。

```
mount -t nfs -o nolock -o tcp xx.xx.xx.xx:/xx/xx /mnt
```

2.5.2 KO 加载

配置完网络之后，即可在单板上访问服务器目录，进入编译输出的 ko 目录加载 ko，ko 加载的参考

命令如下：

```
cd /mnt/sdk/out/xm7606v13/ko
```

```
./load xm7606v13 -i -osmem 256
```

ko 加载完成之后，可以 lsmod 看一下驱动的加载情况，如下：

```
~ # lsmod
Module                  Size  Used by    Tainted: G
xm_fb                   86016  0
xm_vo                   270336  1 xm_fb
xm_venc                 131072  0
xm_h265e                98304  1 xm_venc
xm_rc                   110592  2 xm_venc,xm_h265e
xm_jpege                73728  1 xm_venc
xm_vedu                24576  2 xm_h265e,xm_jpege
xm_chnl                 36864  3 xm_venc,xm_jpege,xm_vedu
xm_vi                   249856  0
xm_mcf                  69632  0
xm_vpss                 290816  2 xm_vi,xm_mcf
xm_isp                  69632  2 xm_vi,xm_vpss
xm_region               65536  2 xm_vo,xm_vpss
xm_gdc                  86016  2 xm_vi,xm_vpss
xm_vgs                 135168  6 xm_vo,xm_venc,xm_vi,xm_mcf,xm_vpss,xm_gdc
xm_tde                  57344  1 xm_region
xm_common              143360  16 xm_fb,xm_vo,xm_venc,xm_h265e,xm_rc,xm_jpege,xm_vedu,xm_chnl,xm_vi,xm_mcf,xm_vpss,xm_isp,xm_region,xm_gdc,xm_vgs,xm_tde
xm_osal                 61440  52 xm_fb,xm_vo,xm_venc,xm_h265e,xm_rc,xm_jpege,xm_vedu,xm_chnl,xm_vi,xm_mcf,xm_vpss,xm_isp,xm_region,xm_gdc,xm_vgs,xm_tde,xm_common
```

 说明

相较于上一代SDK，XM7606V13/XM7206V11A SDK 加载驱动时无需指定sensor类型，board类型及yuv类型。删除了sys_config.ko，管脚复用通过uboot表格进行配置。

2.5.3 程序运行

驱动加载完成后，即可在单板上运行应用程序。

3 应用程序开发简介

3.1 编写代码

用户可根据个人习惯选择代码编写工具。通常在 Windows 环境下使用 Source Insight, 在 Linux 环境下使用 Vim+ctags+cscope, 功能也相当强大。

3.2 运行应用程序

要运行编译好的应用程序, 首先需要将其添加到目标机中, 必须完成以下工作:

- 将应用程序和需要的库文件 (如果有) 等添加到目标机的根文件系统相应的目录中。通常将应用程序放到/bin 目录里, 库文件放到/lib 目录里, 配置文件则放到/etc 目录里。
- 制作包含新应用程序的根文件系统。



说明

如果可执行应用程序, 需要读写文件系统操作。请选择 yaffs2、jffs2 文件系统。

如果新添加的应用程序需要系统启动后自动运行, 请编辑/etc/init.d/rcS 文件, 添加需要启动的应用程序路径。