

Geniatech NXP 系列 Linux 软件开发指南

概述

文档作为 Geniatech nxp 系列 Linux 软件开发指南,目的帮助软件开发工程师、技术支持工程师更快 上手 Geniatech NXP 平台 Linux 的开发及调试。后续会在此文档上持续更新 Geniatech NXP 系列的产品开发 指导。

产品版本

产品名称	内核版本	Yocto 版本
xpi-imx8mmevk	Linux4.14	Sumo



本文档(本指南)主要适用于以下工程师: 软件开发工程师



日期	版本	作者	审核	修改说明
2022-08-06	V1.00	1h	cf	1.初始化 Geniatech MXP linux 开发指导架构
				2.支持 xpi-imx8mmevk

目录

Geniatech NXP 系列 Linux 软件开发指南	1
概述	1
产品版本	1
读者对象	1
修订记录	1
目录	2
1. 支持列表	3
2. SDK 软件架构	4
2.1 SDK 概述	4
2.2 SDK 软件框架	5
2.3 SDK 开发流程	5
3. 开发环境搭建	6
3.1 概述	6
3.2 Linux 服务器开发环境搭建	7
3.2.1 发布包使用 Linux 服务器系统版本	7
3.2.2 依赖包安装	7
4. SDK 安装准备工作	8
4.1 SDK 获取	8
4.2 SDK 目录结构	9
4.3 SDK 更新及问题反馈	10
5. SDK 编译	11
5.1 选择项目	11
5.2 Uboot 编译	11
./build.sh -i uboot	11
5.3 Kernel 编译	11
./build.sh -i bootimg	11
5.4 Rootfs 编译	11
./build.sh -i rootfs	12
5.5 固件打包	12
./build.sh -i pack	12
5.6 应用部分编译	13
6. SDK 镜像烧写	14
6.1 概述	14
7. 问题反馈	16
7.1 反馈途径	16

1. 支持列表

1.1 产品列表

平台	产品型号	软件文档支持	功能说明
NXP	imx8mmevk	支持	
NXP	imx8mqevk	支持	

1.2 文档开发支持列表

产品	硬件板型	功能说明
xpi-imx8mmevk	开发板	hdmi out、ent、usbx4、wifi/bt、lte、rtc、引出 40pin 的引脚 排线

2. SDK 软件架构

2.1 SDK 概述

NXP Linux SDK 是基于 Yocto 的 Linux 开发 BSP。Yocto 项目是一个开源协作项目,能帮助开发者为不同 硬件架构的嵌入式产品,制作基于 Linux 的定制系统。Yocto Project 提供了灵活的工具集和开发环境,使世 界各地的嵌入式设备开发人员通过共享技术、软件堆栈、配置和用于创建这些定制 Linux 映像的最佳实践进 行协作。



Yocto 有以下几点优势:

1. 广泛采用:许多半导体、操作系统、软件和服务供应商在其产品和服务中采用并支持 Yocto 项目,也有很庞大的社区支持,想要要查看 Yocto 项目社区和参与 Yocto 项目的公司

2. 支持架构多: Yocto项目支持 Intel, ARM, MIPS, AMD, PPC 等架构。大多数 0DM、 0SV 和芯片供应商都创建并提供支持其硬件的 BSP。如果想增加自定义芯片,则可以创建支持该体系结构的 BSP。除了大量的架构支持外,Yocto项目还通过 Quick EMUlator (QEMU)完全支持各种设备仿真。

3. 适用于资源受限的嵌入式物联网设备: 与完整的 Linux 发行版不同, yocto 能让你自定义你的映像,你可以决定放哪些功能或者模块到你的映像中,比如说很多设备没有显示屏幕,那么像 X11, GTK+, Qt 或者 SDL 之类的组件就能不安装。最终的映像会足够小,而且没有多余的功能。

2.2 SDK 软件框架

SDK 软件框架, 从下至上分为 Bootloader 、 Linux Kernel 、 i.MX release layer 、Yocto Project community layers 四个层次。

各层次内容如下:

● Bootloader 层主要提供底层系统支持包, 如 Bootloader 、 U-Boot 、ATF 相关支持

 Kernel 层主要提供 Linux Kernel 的标准实现, Linux 也是一个开放的操作系统。该 SDK 的 Linux 核心为标准的 Linux4.14 内核,提供安全性,内存管理,进程管理, 网络协议栈等基础支持;主要 是通过 Linux 内核管理设备硬件资源,如 CPU 调度、缓存、 内存、I/0 等。

• i.MX release layer 层包括 meta-freescale, poky,meta-openembedded layers 以及 meta-freescale-distros 。

• Yocto Project community layers 层包括为base和i.MX Arm参考板提供支持,为第三方和合作伙伴董事会提供支持,freescale发行版,提供FSL Community BSP的基本配置,POKY的基础组件,提供多种浏览器以及QT5 相关应用组件。

2.3 SDK 开发流程

Geniatech 发布的 NXP 平台 SDK 开发包针对多种不同产品形态开发的 SDK 。 可以基于本 SDK ,有效地 实现系统定制和应用移植开发 。

系统的开发环境和编译代码。下面将简单介绍下该流程:

5

1)检查系统需求: 在下载代码和编译前,需确保本地的开发设备能够满足需求,包括机器的硬件 能力, 软件系统,工具链等。目前 SDK 支持 Linux 操作系统环境下编译,并且提供 Linux 环境下的工具链 支持, 其他如 MacOS, Windows 等系统暂不支持。

2) 搭建编译环境: 介绍开发机器需要安装的各种软件包和工具

3)选择设备:在开发过程中,需要开发者根据自己的需求,选择对应的硬件板型

4) 下载源代码: 选定设备类型后, 需要安装 repo 工具用于批量下载源代码,

5)系统定制:开发者可以根据使用的硬件板子、产品定义,定制 U-Boot、Kernel、 Rootfs

6)编译与打包:介绍具备源代码后,选择产品及初始化相关的编译环境,然后执行编译命令,包括整体或模块编译以及编译清理等工作。

7) 烧录并运行:

3. 开发环境搭建

3.1 概述

本节主要介绍了如何在本地搭建编译环境来编译 NXP Yocto Linux SDK 源代码。当前 SDK 只支持在 Linux 环境下编译,并提供 Linux 下的交叉编译工具链。

一个典型的嵌入式开发环境通常包括 Linux 服务器、Windows PC 和目标硬件版,以 imx8mmevk 为例, 典型开发环境如图 3-1 所示。

• Linux 服务器上建立交叉编译环境, 为软件开发提供代码更新下载, 代码交叉编译服务。

• Windows PC 和 Linux 服务器共享程序,并安装 SecureCRT 或 puTTY, 通过网络远程登陆到 Linux 服务器,进行交叉编译以及代码的开发调试。

• Windows PC 通过串口和 USB 与目标硬件板连接, 可将编译后的镜像文件烧写到目标硬件板, 并 调试系统或应用程序。

6



图 3-1

注: 开发环境中使用了 Windows PC,实际上很多工作也可以在 Linux PC 上完成,如使用 minicom 代替 SecureCRT 或 puTTY 等,用户可自行选择。

3.2 Linux 服务器开发环境搭建

NXP Yocto Linux SDK 是在 Ubuntu 16.04 上开发测试的。因此,我们推荐使用 Ubuntu 16.04 的系统进行编译。其他版本没有具体测试,可能需要对软件包做相应调整。

除了系统要求外,还有其他软硬方面的要求。

• 硬件要求: 64 位系统, 硬盘空间大于 50G, 我们建议磁盘空间大于 120G。如果您进行多个构建, 将需要更大的硬盘空间。

• 软件包依赖:除了 python 2.7, make 3.8, git 1.7 之外, 还需要安装一些额外的软件包,将 在软件包安装章节中列出。

3.2.1 发布包使用 Linux 服务器系统版本

本 SDK 开发环境安装如下版本 Linux 系统, SDK 默认均以此 Linux 系统进行编译:

Ubuntu 16.04 LTS

3.2.2 依赖包安装

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。

平台	安装依赖
imx8mmevk	1. 安装依赖包
	sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc- multilib build-essential chrpath socat libsdll.2-dev
	2. Ubuntu 12.04 和 14.04 需要依赖的软件包
	7

sudo apt-get install libsdl1.2-dev xterm sed cvs subversion coreutils texi2html \ docbook-utils python-pysqlite2 help2man make gcc g++ desktop-fileutils \ libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf automake groff curl lzop asciidoc \ 3. Ubuntu 12.04 需要依赖的软件包 sudo apt-get install uboot-mkimage 4. Ubuntu 14.04 需要依赖的软件包 sudo apt-get install u-boot-tools 若编译遇到报错,可以视报错信息, 安装对应的软件包。

4. SDK 安装准备工作

4.1 SDK 获取

SDK 通过 Geniatech 对外服务器发布,客户需要对应的向我司申请 SDK。

4.1.1 SDK 下载链接

平台	下载链接
xpi-imx8mmevk	http://www.geniatech.net/hefei/sd-release/imx8mm_xpi_yocto-kernel-uboot-source-
	20210909/ <u>imx8mm_xpi_yocto-kernel-uboot-source-20210909.zip</u>

4.1.2 SDK 代码压缩包

为方便客户快速获取 sdk 源码, geniatech 通常会提供对应硬件的一个 init 初始化压缩包。开发者可以 通过可以通过一下方式。

以 xpi-imx8mmevk 为例:

unzip imx8mm_xpi_yocto-kernel-uboot-source-20210909.zip

注: 解压密码请向相关负责人获取

cd imx8mm_xpi_yocto-kernel-uboot-source-20210909

产品平台	软件压缩包	版本
xpi-imx8mmevk	imx8mm_xpi_yocto-kernel-uboot-source-20210909.zip	V1.00

4.2 SDK 目录结构

SDK 下载完成后, 在根目录下可以看到如下目录结构:

以 xpi-imx8mmevk 为例:

- fsl-setup-af.sh
- fsl-setup-release.sh
- geniatech
- loong
- ----- lunch. sh
- ---- overlay ---- README
- ------ README-IMXBSP
- ----- setup-environment
 - source
- fsl-setup-af.sh 增加了 AF meta-layer 的环境配置
- fsl-setup-release. sh yocto环境配置,编译前需要运行次脚本
- geniatech 目录存放 uboot, kernel 源码
- loong 目录存放 项目定制文件以及打包编译脚本
- lunch.sh 软链接,编译前运行选择要编译的项目
- overlay 项目 overaly
- README 为 Freescale Yocto BSP readme
- README-IMXBSP 为 BSP 4.14.98_2.0.0 Release readme
- setup-environment 环境配置脚本,编译前会调用
- source 目录存放所以应用程序菜谱

4.3 SDK 更新及问题反馈

联系 geniatech 相关负责人

5. SDK 编译

5.1 选择项目

■ xpi-imx8mmevk

执行 source lunch.sh, 会列出目前支持的项目如下:



然后选择相应的项目,输入序号 5,进行环境初始化;

5.2 Uboot 编译

■ xpi-imx8mmevk

./build.sh -i uboot

编译完,在build-xwayland-imx8mmevk-xpi/tmp/deploy/images/imx8mmevk 目录会生成 u-boot.bin 等 相关镜像文件 。

5.3 Kernel 编译

■ xpi-imx8mmevk

./build.sh -i bootimg

编译完,在build-xwayland-imx8mmevk-xpi/tmp/deploy/images/imx8mmevk 目录会生成 boot.img 以及 dtb 等相关镜像文件 。

5.4 Rootfs 编译

■ xpi-imx8mmevk

./build.sh -i rootfs

编译完,在build-xwayland-imx8mmevk-xpi/tmp/deploy/images/imx8mmevk目录下会生成fsl-imageqt5-validation-imx-imx8mmevk.tar.bz2 rootfs 压缩包以及fsl-image-qt5-validation-imximx8mmevk.sdcard.bz 烧录压缩包

5.5 固件打包

■ xpi-imx8mmevk

./build.sh -i pack

编译完,在loong/out/nxp-imx8/yocto/nxp-imx8_yocto_xwayland-imx8mmevk-xpi 目录下会生成用于烧录的固件包

5.6 应用部分编译

针对单独应用菜谱的编译,通过 bitbake 命令可以完成。具体使用方法参考 yocto 官 网。

6. SDK 镜像烧写

6.1 概述

本章节主要介绍如何将构建完成的镜像文件(image)烧写并运行在硬件设备上的流程。 NXP 提供 uuu 烧录工具完成镜像烧录。

工具	运行系统	描述
uuu	Windows	NXP 整包烧录固件工具

6.2 开发板介绍



6.3 烧写模式切换

模式	拨码开关
烧录模式	10101001
启动模式	01101001

6.4 烧录包介绍

- flash.bat
- fsl-image-qt5-validation-imx-imx8mmevk.sdcard
- ----- imx-boot-imx8mmevk-sd.bin-flash_evk
- —— uuu. auto
- uuu. exe
- flash.bat 烧录批处理
- fsl-image-qt5-validation-imx-imx8mmevk.sdcard 烧录整包
- imx-boot-imx8mmevk-sd.bin-flash_evk uboot镜像
- uuu. auto uuu 烧录脚本
- uuu. exe uuu windows 程序

将板子切换到烧录模式,通过双 USB A 接口的 USB 线连接电脑和板子的 USB 烧录端口,双击 flash.bat 后上电进行烧录,烧录完成后提示如下。



烧录完成后切换到启动模式重新上电即可进入系统。

7. 问题反馈

7.1 反馈途径

请联系相关负责人,上报给研发。