

MYD-Y7Z010/007S Board

Linux 开发手册

版本 V1.0

版本记录

版本号	说明	时间
V1.0	初始版本	2018/1/2
V1.1		
V1.2		

目录

目录.....	1
第 1 章概述及软件资源介绍	2
1.1 概述.....	2
1.2 软件资源.....	2
第 2 章 Linux 开发环境搭建.....	3
2.1 建立工作目录.....	3
2.2 设置交叉编译工具	3
2.3 安装工具.....	3
第 3 章 Linux 系统编译.....	4
3.1 编译 Bootloader.....	4
3.2 编译 Linux 内核	4
3.3 构建根文件系统.....	4
3.4 制作文件系统.....	5
3.4.1 制作 ramdisk.....	5
3.4.2 修改根文件系统.....	6
第 4 章 Linux 系统烧写	7
4.1 创建启动 TF 卡	7
4.2 准备烧写文件.....	8
4.3 开始烧写.....	8
第 5 章 Linux 应用程序.....	9
5.1 Led	9
5.2 CAN	9
附录一 免责声明.....	10
附录二 售后服务与技术支持.....	11
附录三 技术支持说明	12

第 1 章概述及软件资源介绍

1.1 概述

MYD-Y7Z010/007S Board 提供了丰富的系统资源和软件资源，本手册将从环境搭建开始，一步步介绍如何进行 MYD-Y7Z010/007S Board Linux 开发。本手册中开发主机上的命令以 Ubuntu 为例进行讲解。

1.2 软件资源

类别	名称	备注	源码
Tool chains	gcc 4.6.1	gcc version 4.6.1 (Sourcery CodeBench Lite 2011.09-50)	
Boot loader	boot.bin	一级引导程序 包括 FSBL 和 u-boot	Yes
Linux Kernel	Linux 3.15.0	专为 MYD-Y7Z010/007S Board 的硬件制定的 Linux 内核	Yes
Driver	USB host	USB Host 驱动	Yes
	Ethernet	千兆以太网驱动(3 个网口)	Yes
	MMC/SD/TF	MMC/SD/TF 卡驱动	Yes
	UART	串口驱动	Yes
	UARTlite	Uartlite 串口驱动	Yes
	CAN	CAN 驱动	Yes
	LCD	LCD 驱动	Yes
	HDMI	HDMI 驱动	Yes
	LED	LED 驱动	Yes
File system	GPIO	GPIO 驱动	Yes
	Ramdisk	Ramdisk 系统镜像	
	Rootfs	Buildroot 制作	Yes

表 1-1

第 2 章 Linux 开发环境搭建

2.1 建立工作目录

拷贝 MYD-Y7Z010/007S Board 光盘中的资料到主机中：

```
$ mkdir -p <WORKDIR>  
$ cp -a <DVDROM>/04-Linux_Source/* <WORKDIR>
```

2.2 设置交叉编译工具

```
$ cd <WORKDIR>/Toolchain  
$ tar xvjf Sourcery_CodeBench_Lite_for_Xilinx_GNU_Linux.tar.bz2  
$ export PATH=$PATH:<WORKDIR>/Toolchain/  
CodeSourcery/Sourcery_CodeBench_Lite_for_Xilinx_GNU_Linux/bin  
$ export CROSS_COMPILE=arm-xilinx-linux-gnueabi-
```

执行完“export”命令后输入 arm 按 Tab 键来检查是否设置成功，该设置只对当前终端有效，如需永久修改，请将以上 export 命令添加到用户启动脚本文件：~/.bashrc。

2.3 安装工具

安装其他必要工具：

```
$ sudo apt-get install build-essential git-core libncurses5-dev \  
flex bison texinfo zip unzip zlib1g-dev gettext \  
gperf libstdc++-dev libstdc++0-dev libwxgtk2.6-dev \  
uboot-mkimage \  
g++ xz-utils
```

第 3 章 Linux 系统编译

3.1 编译 Bootloader

进入 Bootloader 目录，解压 U-boot 源码：

```
$ cd <WORKDIR>/Bootloader
$ tar jxvf u-boot-xlnx.tar.bz2
$ cd u-boot-xlnx
```

开始编译：

```
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- distclean
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- \
zynq_myd_config
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi-
```

编译完成后，在当前目录下会生成“u-boot”ELF 文件。使用此镜像制作 boot.bin 时需要重命名为“u-boot.elf”。

3.2 编译 Linux 内核

进入 Kernel 目录，解压内核源码：

```
$ cd <WORKDIR>/Kernel
$ tar xvjf linux-xlnx.tar.bz2
$ cd linux-xlnx
```

开始编译：

```
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- distclean
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- \
zynq_myd_defconfig
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- uImage
$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- dtbs
```

编译完成后，会在 arch/arm/boot 目录下生成 ulmage 文件，在 arch/arm/boot/dts/生成 zynq_myd.dtb 文件。

3.3 构建根文件系统

这里使用 buildroot 来构建根文件系统。

首先进入文件系统目录，解压 buildroot 源码：

```
$ cd <WORKDIR>/Filesystem
```

```
$ tar jxvf buildroot-2015.02.tar.bz2
$ cd buildroot-2015.02
```

将目录中的 zynq_myd_config 文件重命名为.config:

```
$ cp zynq_myd_config .config
```

配置 buildroot:

```
$ make menuconfig
```

设置交叉编译工具路径:

```
Toolchain --->
  () Toolchain path
```

进入 Toolchain path 选项, 填入编译器路径:

```
<WORKDIR>/Toolchain/CodeSourcery/Sourcery_CodeBench_Lite_for_Xilinx_
GNU_Linux
```

退出配置, 开始编译:

```
$ make
```

编译完成之后将在 “buildroot-2015.02/output/images” 目录内生产 rootfs.tar 文件。

3.4 制作文件系统

3.4.1 制作 ramdisk

(1)挂载 Ramdisk

新建目录 tmp, 并将 uramdisk.image.gz 拷贝至该目录

```
$ cd <WORKDIR>/Filesystem
$ mkdir tmp
$ cp uramdisk.image.gz tmp/
$ cd tmp/
```

去掉 mkimage 生成的 64 bytes 的文件头, 生成新的 ramdisk.image.gz

```
$ dd if=uramdisk.image.gz of=ramdisk.image.gz bs=64 skip=1
```

gunzip 解压 ramdisk.image.gz 生成 ramdisk.image

```
$ gunzip ramdisk.image.gz
```

新建挂载目录 “ramdiskdir”, 并将 ramdisk.image 挂载

```
$ mkdir -p ramdiskdir
$ sudo mount -o loop,rw ramdisk.image ramdiskdir
```

进入 ramdiskdir 目录, 根据需要做修改

(2)重新生成 ramdisk

同步文件系统并卸载 ramdisk

```
$ sync
```

```
$ sudo umount ramdiskdir
```

用 gzip 压缩 ramdisk.image, 生成 ramdisk.image.gz

```
$ gzip -9 ramdisk.image
```

用 mkimage 添加文件头, 生成新的 uramdisk.image.gz

```
$ mkimage -A arm -T ramdisk -C gzip -n Ramdisk -d ramdisk.image.gz  
uramdisk.image.gz
```

删除临时文件 ramdisk.image.gz

```
$ rm ramdisk.image.gz
```

3.4.2 修改根文件系统

(1)拆 tar 包

```
$ cd <WORKDIR>/Filesystem
```

```
$ mkdir -p rootfs
```

```
$ sudo tar xvf rootfs.tar -C ./rootfs/
```

(2)修改文件夹, 加入新文件。

(3)打包 tar

```
$ cd ./rootfs
```

```
$ sudo tar cvf ../rootfs.tar ./*
```

第 4 章 Linux 系统烧写

这里提供的烧写方案为通过 TF 卡启动 Linux，进入 Ramdisk 文件系统中的脚本烧写 Bootloader，内核到 QSPI-Flash，烧写根文件系统到 eMMC。

4.1 创建启动 TF 卡

(1) TF 卡格式化

请使用光盘目录 03-Tools 目录下的 HP USB Disk Storage Format Tool 2.0.6 工具格式化 TF 卡。

- ①把 TF 卡插入 USB 读卡器，然后将读卡器跟电脑连接
- ②打开 HP USB Disk Storage Format Tool，出现类似提示如下：

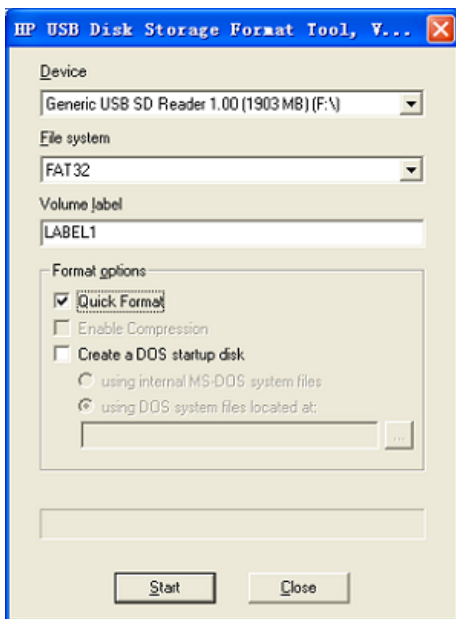


图 5-1

- ③选择“FAT32”系统格式
- ④点击“Start”
- ⑤等待格式化完成，点击“OK”

注意：HP USB Disk Storage Format Tool 会将清除 TF 卡上所有的数据，格式化前请做好备份。

4.2 准备烧写文件

将产品资料光盘 Image 文件夹中的镜像文件复制到刚才格式化的 TF 卡中, 所需的文件如下表所示:

文件名称	说明
boot.bin	系统启动程序, 包括 fsbl、和 u-boot。具体的制作方法将在《MYD-Y7Z010/007S Board 可编程逻辑开发手册》中描述
7z010.bit	XC7Z010 Bitstream 文件, U-Boot 将自动判断 SoC 型号选择加载
7z007.bit	XC7Z007S Bitstream 文件, U-Boot 将自动判断 SoC 型号选择加载
ulmage	Linux 内核
devicetree.dtb	设备树文件
uramdisk.image.gz	ramdisk 文件系统
uEnv.txt	U-boot 环境变量文件
rootfs.tar	根文件系统, 准备烧写到 eMMC 的文件系统

表 4-2

4.3 开始烧写

- (1)将开发板的 SW1 的开关 1 拨到 OFF 端, 开关 2 拨到 ON 端;
- (2)插入已存入烧写文件的 TF 卡, 连接串口, 设置波特率为 115200, 开发板上电;
- (3)开发板将引导进入 Ramdisk 文件系统, 进入 Linux 命令行, 输入命令开始更新:

```
welcome to myir board
myir login: root
[root@myir ~]#/updatesys.sh /mnt/mmcblk1p1
```

脚本将把 BOOT.bin, 7z010.bit/7z007s.bit, devicetree.dtb, ulmage 烧写到 QSPI-Flash, 把 rootfs.tar 烧写到 eMMC。

- (4)烧写完成之后, 将 SW1 的开关 1 拨到 ON 端, 开关 2 拨到 ON 端, 重新上电即可。

第 5 章 Linux 应用程序

MYD-Y7Z010/007S Board 提供了常用外设的演示程序，程序以及源码都位于“<WORKDIR>/Examples/”，请根据目录内的 Makefile 或 README 文件进行编译：

```
$ cd <WORKDIR>/Examples/  
$ export CROSS_COMPILE=arm-xilinx-linux-gnueabi-  
$ make
```

连接调试串口 J8，设置波特率为 115200，数据位为 8，停止位为 1，无奇偶校验，将对应的可执行程序拷贝到开发板，在开发板上运行程序时需要注意权限，如果无法执行请使用如下命令：

```
# chmod +x <program-to-be-executed>
```

5.1 Led

本例程演示如何使用 Linux API 操作开发板上的 LED，详情请参考源码。

将目录“<WORKDIR>/Examples/led-test”中的可执行程序 led-test 拷贝至开发板，把调试串口 J8 连到 PC 上。执行以下命令后，开发板上 LED 闪烁：

```
# ./led-test
```

5.2 CAN

本例程演示如何使用 Linux API 操作开发板上的 can，详情请参考源码。

将目录“<WORKDIR>/Examples/can-test”中的可执行程序 can-test 拷贝至开发板，把调试串口 J8 连到 PC 上。执行以下命令：

（注：请将板子上的 can 接口与外部 can 设备连接，来测试 can 通讯）

```
# ./can-test
```

附录一 免责声明

本产品手册（以下简称“手册”）发布时，会尽可能的完全与正确。内容若有变动，恕不另行通知。本手册例子中所用公司、人名和数据若非特别声明，均属虚构。

未得到深圳市米尔电子有限公司（简称“米尔电子”）明确的书面许可，不得为任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。

深圳市米尔电子有限公司 版权所有

附录二 售后服务与技术支持

MYiR 深圳市米尔电子有限公司

销售邮箱: sales.cn@myirtech.com

公司网址: www.myir-tech.com

深圳总部

联系电话: 0755- 25622735 / 18926526796

公司地址: 深圳市龙岗区坂田街道发达路云里智能园 2 栋 6 楼 604 室

上海办事处

联系电话: 021-60317628 / 15901764611

办公地址: 上海市普陀区中江路 106 号北岸长风 I 座 1402

北京办事处

联系电话: 010-84675491 / 13269791724

办公地址: 北京市昌平区东小口镇中滩村润枫欣尚 1 号楼 505 室

武汉研发中心

联系电话: 027-59621648

办公地址: 武汉市东湖高新区金融港一路 7 号光谷智慧园 15 栋 01 号楼 3 楼

联系技术支持

通过技术支持邮件, 客户可以更加方便通过截图, 附着代码等方式详细准确的描述技术问题, 以便于我公司技术支持人员提供准确的技术协助信息。在邮件中, 请客户附着详细的联系方式、产品内容、购买时间和方式等, 以便我公司技术人员快速响应。

- 技术支持邮箱: support.cn@myirtech.com
- 邮件回复时间: 12 小时内 (工作日)
- 技术支持电话:
 - NXP/ATMEL 产品线: 0755-22316235
 - TI/Xilinx 产品线: 027-59621648
 - 充电桩产品线: 0755-22919239

附录三 技术支持说明

MYIR 的理念是“**我们的成功源于客户的成功**”。

为了协助客户更加快速高效地使用我公司产品，MYIR 通过各地办事处提供完善周到的技术支持服务，客户可以通过如下指引获取相应技术信息和技术支持服务：

1. 产品开发资料：

MYIR 的所有开发板都提供配套资料光盘，资料光盘内容一般涉及如下内容：

- 产品使用手册
- 产品原理图(PDF 格式)
- 完整的例程代码、BSP 包
- 板载主要芯片技术手册
- 相应开发工具链（GNU 工具或 MDK 等第三方工具评估板）

2. 技术支持范围

MYIR 对所销售的产品提供 6 个月的免费技术支持服务，技术支持服务范围：

- 所购买产品的软硬件资源，硬件保修
- 协助客户正确地使用和调试光盘类容中提供的例程代码
- 客户对于产品文档，操作、嵌入式软硬件平台使用的问题

由于嵌入式开发的特殊性，以下情况不在我们的免费技术支持服务范围，将根据情况酌情处理：

- 用户自行开发中遇到的软硬件问题，对硬件的修改和造成损坏
- 用户自行裁减编译运行嵌入式操作系统遇到的问题
- 用户自己在平台中自行开发、修改的程序
- 修改光盘的软件代码遇到的问题

如欲了解米尔电子更多产品，请参阅米尔电子网站，致电或电邮我们，感谢您对我公司产品的关注！