

一、Tiny4412 开发板介绍

CPU: 三星 exynos4412(四核、主频 1.5GHz); 智能与环保

内核: ARM 架构 cortex-A9

ARM: V7 架构 cortex 系列:

A: 高端芯片, 操作系统开发; A5、A7、A8、A9、A15

R: 实时芯片

M: 微控制器 M3、M4 --- 裸机开发

学习操作系统: 应用层、内核层(Linux 架构和思想) --- Linux ARM9

资料:

exynos4412 数据手册;

arm-linux-gcc-4.5.1-v6-vfp-20120301.tar 编译软件;

SD-Flasher.exe SD 卡工具;

数据手册介绍:

1、42 页产品描述

四核、主频 1.4~1.5Ghz, 256k RAM、64k ROM。

8 路 IIC、4 路串口、3 路 SPI、2 个 USB

2、86 页地址映射: 地址的描述

3、99 页 GPIO 口: 304 个 IO 口;

每个 IO 口有 6 个寄存器控制:

①CON: 配置 IO 功能(输入、输出、串口等)

②DAT: 等同于 M3 数据寄存器(可以读取 IO 高低电平、可以写入高低电平)

③PUD: 配置上拉和下拉

④DRV: 增加电流

⑤CONPDN:非正常模式下配置功能

⑥PUDPND: 非正常模式下配置上拉和下拉

4、串口

1381 页 4 路串口

1395 页寄存器、地址描述

ULCON: 配置串口协议

UTRSTAT:判断状态

UTX、URX: 发送和接收的数据寄存器

UBRDIV: 计算波特率的

二、开发板的烧写

1、SD-Flasher.exe 右键管理员打开

2、scan: 扫描 SD 卡(yes--有分区、no--没有分区);

3、reformat: 格式化;

4、relayout: 重新分区;

5、找到 superboot4412.bin(引导程序)点击 fuse;

6、将光盘资料里面的 images 文件夹拷贝到 sd 卡;

三、交叉编译环境的搭建

1、介绍

最终的程序 --- 虚拟机上编译, arm 开发板上运行。

交叉编译工具: arm-linux-gcc

2、 安装流程

拷贝安装包到虚拟机里面；

解压到根目录 `tar -xvf arm-linux-gcc-4.5.1-v6-vfp-20120301.tgz -C /` ；

设置环境变量 `export PATH=$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin` ；

添加到开机自启动的文件里面 `~/.bashrc` ；

重启或者 `source ~/.bashrc` ；

命令：`arm-linux-gcc -v` //打印安装的交叉编译工具的版本号

3、 编译器类似于 gcc -E、 -S、 -o

四、 裸机启动流程

1、 支持四种启动方式

exynos4412: nand flash、 SD 卡、 EMMC、 USB

tiny4412:SD 卡、 EMMC 启动

2、 上电后程序先从 irom 运行，再到外部存储中搬运 BL1 代码，然后去搬运 BL2 代码到 DRAM 中运行。

上电执行 irom 的代码是三星编写的代码固化在硬件，不开源。

BL1 代码：三星编写，不开源 E4412_N.bl1.bin

BL2 代码：用户编写的程序。

烧写 BL2 的代码：三星提供，开源。

3、 烧写流程

`#ls /dev/sd*` //查看挂接的 sd 卡；

插入 SD 后多出 SDB(或 SDC) -- SD 卡注册产生的设备节点；

将 bl1，bl2 烧写到指定的 SD 卡扇区中；

例：烧写 led 灯程序到 SD 卡：

①在终端里进入 led 目录下，执行 makefile，将用户代码 led.s 编译生成 led.bin 文件；

②进入下一层目录 sd_fuse 里，执行 makefile，编译三星公司编写的 V310-EVT1-mkbl2.c 文件，生成 mkbl2 工具，利用该工具将 led.bin 文件转换成 BL2 代码 (bl2.bin)；

③进入下一层目录 tiny4412 里，执行 `sd_fusing.sh /dev/sdb`，其中 `sd_fusing.sh` 是一个脚本文件，`/dev/sdb` 是 SD 卡挂接点，目的是利用脚本文件将 BL2 代码烧写到 SD 卡里；

④将 SD 卡插入开发板里，设置启动方式为从 SD 卡启动；