1. **交叉编译工具 搭建（应用程序编译工具链）**

1.1、工具包: gcc-linaro-5.3-2016.02-x86\_64\_arm-linux-gnueabihf.tar.xz

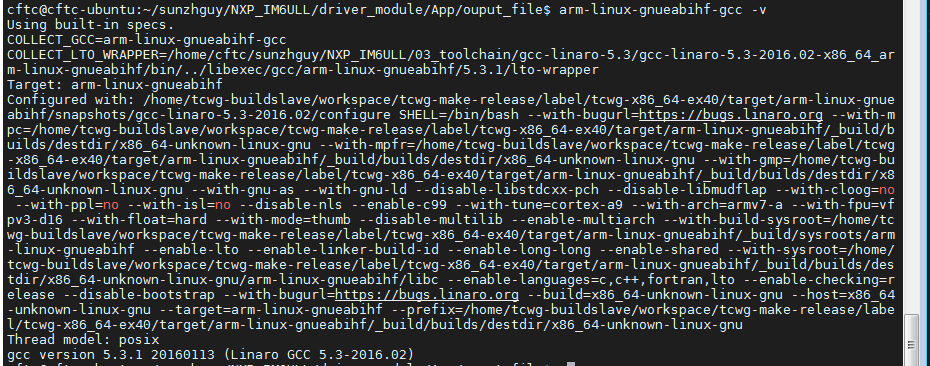
1.2、例解压目录如下: /home/cftc/sunzhguy/NXP\_IM6ULL/03\_toolchain

1.3、tar xvJf gcc-linaro-5.3-2016.02-x86\_64\_arm-linux-gnueabihf.tar.xz

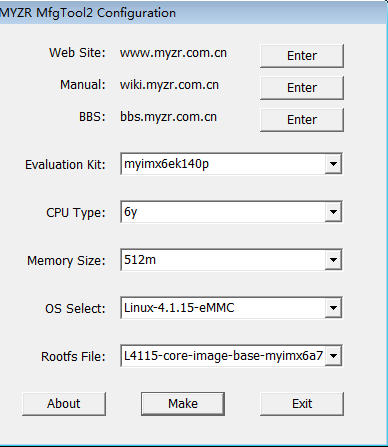
1.4、配置环境变量修改vim ~/.bashrc

export PATH=$PATH:/home/cftc/sunzhguy/NXP\_IM6ULL/03\_toolchain/gcc-linaro-5.3/gcc-linaro-5.3-2016.02-x86\_64\_arm-linux-gnueabihf/bin

安装成功显示如下:



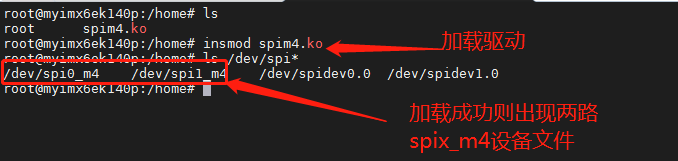
1. **Uboot、内核、文件系统烧录 (直接使用开发板测试略过)**



1. **加载SPI驱动 （IMX6与M4通讯传输驱动加载，上电启动开发板测试需手动加载）**

上电系统启动，进入系统，进入home 目录，加载spim4.ko

myimx6ek140p login: root

root@myimx6ek140p

1. **硬件接口连接说明 （IMX6 与M4的SPI通讯接口及同步IO接口定义）**

M4连接 开发板P21座子（SP1） M4连接管脚

4<SPI1\_SIN> +++ ECSPI1\_MOSI 3 | |2 ECSPI1\_MISO 2<SPI1\_SOUT>

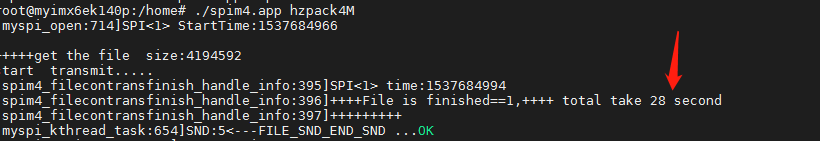
3<SPI1\_SCK> +++ ECSPI1\_SCLK 5| |4 ECSPI1\_CS0 1<SPI1\_PCS>

1. **应用程序测试源码**

应用程序源码 spim4\_test.c

arm-linux-gnueabihf-gcc spim4\_test.c -o spim4.app

4M文件测试总共需要48秒 hzpack4M(SPI传输的文件)



1. **操作接口说明**

SPI传输设备

1路: /dev/spi0\_m4

2路:/dev/spi1\_m4

#define SPI\_M4\_START         \_IO(SPI\_M4\_MAGIC, 0)                               //SPI\_M4 启动工作

#define SPI\_PROCESSINFO     \_IOWR(SPI\_M4\_MAGIC, 1, char[PROCESSINFO\_MSGSIZE])   //文件处理的流程信息

#define SPI\_HANDSHARKSTATUS \_IOR(SPI\_M4\_MAGIC, 2, char[SHARK\_STATUS\_MSGSIZE])   //握手状态

#define SPI\_HEART\_STATUS    \_IOR(SPI\_M4\_MAGIC, 3, char[HEART\_STATUS\_MSGSIZE])   //心跳状态

#define SPI\_FILE\_INFO(N)    \_IOW(SPI\_M4\_MAGIC, 4, char[FILE\_INFO\_MSGSIZE(N)])   //设置传输文件信息

#define SPI\_FILEREQ\_INFO    \_IOR(SPI\_M4\_MAGIC, 5, char[FILEREQ\_INFO\_MSGSIZE])   //文件请求发送信息

#define SPI\_FILEEND\_REQINF  \_IOR(SPI\_M4\_MAGIC, 6, struct  file\_end\_request\_info\_t)   //文件传输结束信息返回状态

1、open 打开所需传输的设备

2、启动文件传送之前需要提前设置传输文件信息IOTCTL 调用 SPI\_FILE\_INFO(N)设置

3、将发送的文件通过write 写入内核buffer

4、启动发送调用IOTCL SPI\_M4\_START指令启动SPIM4 发送

5、发送过程可以调用SPI\_PROCESSINFO 获取相应的传输状态，从而读取不同阶段返回结果，具体详细参考spim4\_test.c 应用程序源码。