

最近在调试ADV7401，调试的过程难免要反复修改寄存器，然后看结果现象。传统的做法是修改驱动代码寄存器值->编译->下载->运行->看结果，这一过程比较费时间，而我每次需要改的可能只是1个bit，有没有更便捷的方法呢？

在浏览TI E2E论坛时发现有人用一个叫i2c-tools的工具，没有使用过，从官网下载后发现很好用也很便捷，非常适合81xx平台的使用，现简单介绍使用方法：

一、下载i2c-tools，交叉编译

1、从开源网站<http://dl.lm-sensors.org/i2c-tools/releases/>下载i2c-tools，我使用的是3.0.2的版本。

2、解压后的目录结构如下：

```
zhaochenhui@server-rd3:~/i2c-tools-3.0.2$ ls -l
total 60
-rw-r--r-- 1 zhaochenhui arm-ti 2301 Nov 29 2008 CHANGES
-rw-r--r-- 1 zhaochenhui arm-ti 17987 Jun 27 2007 COPYING
-rw-r--r-- 1 zhaochenhui arm-ti 843 Jan 8 16:45 Makefile
-rw-r--r-- 1 zhaochenhui arm-ti 1931 Nov 2 2007 README
drwxr-xr-x 2 zhaochenhui arm-ti 4096 Nov 29 2008 espram
drwxr-xr-x 2 zhaochenhui arm-ti 4096 Nov 29 2008 esprouter
drwxr-xr-x 2 zhaochenhui arm-ti 4096 Nov 29 2008 include
drwxr-xr-x 2 zhaochenhui arm-ti 4096 Nov 29 2008 py-usb
drwxr-xr-x 2 zhaochenhui arm-ti 4096 Nov 29 2008 stub
drwxr-xr-x 2 zhaochenhui arm-ti 4096 Jan 10 16:54 tools
-rw-r--r-- 1 zhaochenhui arm-ti 24 Nov 29 2008 version.h
zhaochenhui@server-rd3:~/i2c-tools-3.0.2$
```

http://blog.csdn.net/mantis_1984

最有用的是tools/文件夹，接下来修改Makefile，主要是修改交叉编译器，如下图所示

```
COMPILE_PREFIX := /opt/DVRRDK_04.00.00.03/ti_tools/cgt_a8/arago/linux-devkit/bin/ar  
m-arago-linux-gnueabi-  
CC := $(COMPILE_PREFIX)gcc
```

http://blog.csdn.net/mantis_1984

然后保存退出。直接make，进行编译。编译完成后在tools/文件夹下会有如下几个可执行文件，i2cdetect, i2cdump, i2cget, i2cset。如下图所示：

```
chaochenhui@server-t33:~/i2c-tools-3.0.2/tools$ ls  
Module.mk  i2cbusses.o  i2cdetect.c  i2cdump.o  i2cget      i2cget.o  i2cset.c  util.h  
i2cbusses.c  i2cdetect.o  i2cdump.c  i2cget.o  i2cset.o  util.o  
i2cbusses.h  i2cdetect.o  i2cdump.o  i2cget.c  i2cset.o  util.c  
chaochenhui@server-t33:~/i2c-tools-3.0.2/tools$
```

http://blog.csdn.net/mantis_1984

将i2cdetect, i2cdump, i2cget, i2cset拷贝到你的nfs调试目录，就可以进行操作了。

二、使用举例

1、用i2cdetect检测有几组i2c总线在系统上，输入：./i2cdetect -l

```
/opt/autorun # ./i2cdetect -l
i2c-1  i2c      OMAP I2C adapter      I2C adapter
i2c-2  i2c      OMAP I2C adapter      I2C adapter
/opt/autorun #
```

http://blog.csdn.net/mantis_1984

由上图可知，系统中存在两组总线分别i2c-1和i2c-2。

2、用i2cdetect检测挂载在i2c总线上器件，输入 ./i2cdetect -r -y 1 (检测i2c-1上的挂载情况)

```
/opt/autorun #
/opt/autorun # ./i2cdetect -r -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- UU -- -- -- -- -- --
20: UU -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: 30 -- -- -- 34 -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- 64 -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
/opt/autorun #
```

http://blog.csdn.net/mantis_1984

由上图可知，0x18,0x20, 0x30,0x34, 0x64地址有挂载。而这几个分别是0x18 (tlv3104) , 0x20 (adv7401) , 0x30, 0x34 (sii9135) , 0x64 (这个不知道是谁，可能是系统的)

3、用i2cdump查看器件所有寄存器的值，这个很有用，输入 ./i2cdump -f -y 1 0x20 （查看adv7401寄存器值）

```
./i2cdump -f -y 1 0x20
No size specified (using byte-data access)
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  0123456789abcdef
00: 0f c8 04 00 76 00 02 7f 80 80 00 00 36 7c 00 00  ???..v.?????..6|..
10: 4d 19 c0 ed 12 00 00 41 93 f1 00 00 00 47 00 00  M?????..A???.G..
20: 00 00 00 c0 00 00 00 58 00 00 00 e1 ae f3 a4 f6  ...?..X...?????
30: 67 02 41 84 00 02 00 01 80 c0 17 80 58 a2 6a a0  g?A?.?.?????X?j?
40: 90 01 7e a4 ff b6 12 be 00 00 00 00 00 ef 08 08  ??~?.???.....???
50: 08 24 80 00 00 00 00 08 00 00 00 00 00 01 00 00  ?S?....?....?....
60: 00 00 20 00 00 00 00 03 01 00 00 c0 00 00 00 00  .. ....???.?....
70: 00 00 00 10 04 01 00 3f ff ff ff 1e c0 00 00 00  ...????.?....???..
80: 00 c0 04 00 0e 02 0b 63 5a 08 10 00 40 00 40 00  .??,???cZ??,0.0.
90: 00 00 00 00 00 00 00 00 00 00 00 00 18 22 22 00  .....?""..
a0: 03 ff 00 00 00 00 00 00 00 00 00 00 00 00 00 01  ?.....?
b0: 00 92 78 33 26 d1 00 00 4c 00 40 02 9f ff 00 02  .?x3&?.L.0???.?
c0: 00 00 00 00 00 81 00 00 00 04 07 48 5b 00 01 b4  ....?....??H[.??
d0: 00 10 ff ff 7f 7f 3e 08 3c 08 3c 9b ac 4c 08 03  .?..??>?<?<??L??
e0: 14 80 80 80 80 25 04 63 65 14 63 55 55 00 00 4a  ??????%?ce?cUU..J
f0: 44 0c 32 01 15 e0 69 10 00 03 a0 40 04 e7 ed 00  D?2????i?.??0???.
/opt/autorun #
```

http://blog.csdn.net/mantis_1984

这样就可以对比自己设置的是否一致了。

3、用i2cset设置单个寄存器值，用i2cget读取单个寄存器值，可以在初期调试时发挥很大作用，一旦有预期的现象出现，就可以用i2cdump读出整个寄存器的值，然后固化到代码中。

```
/opt/autorun #
/opt/autorun # ./i2cset -f -y 1 0x20 0x77 0x3f
/opt/autorun # ./i2cget -f -y 1 0x20 0x77
0x3f
/opt/autorun # ## AVERAGE: ENCODE [ 27.0] FPS,
```

http://blog.csdn.net/mantis_1984

上图是i2cset和i2cget使用方法：

```
./i2cset -f -y 1 0x20 0x77 0x3f （ 设置i2c-1上0x20器件的0x77寄存器值为0x3f ）  
./i2cget -f -y 1 0x20 0x77 （ 读取i2c-1上0x20器件的0x77寄存器值 ）
```


图片保存的不太好，凑合看吧！

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/mantis_1984/article/details/18254767

文章标签：linux i2c c语言
个人分类：linux

查看更多>>


想对作者说点什么？ 我来说两句

 **kangear** 2015-05-13 15:50:50 #1楼
很方便，可以快速检测I2C设备。可惜图片保存的太不好看了。；)

上一页 1 下一页

i2cdetect/i2cdump 查看系统的i2c bus数量以及i2c device挂载地址

ASUS的i2cdetect tool放在device/asus/Z00L/i2c-tools目录下，用mmm device/asus/Z00L/i2c-tools就会生成相应的bin文件，push到...

 **u013308744** 2015-12-01 14:03:07 阅读数：6531

树莓派学习笔记——I2C Tools 学习笔记

通过i2cdetect -I指令可以查看树莓派上的I2C总线，从返回的结果来看树莓派含有两个I2C总线，通过阅读相关的资料，树莓派1代使用I2C0，而树莓派2代使用I2C1。...

 **xukai871105** 2013-11-10 18:31:19 阅读数：36824

i2c-tools的使用方法举例 - CSDN博客

i2c-tools的使用方法 最近在调试ADV7401,调试的过程难免要反复修改寄存器,然后看结果现象。传统的做法是修改驱动代码寄存器值->编译->下载->运行->看...

2018-5-3

I2C-tool使用 - CSDN博客

举报内容: **I2C-tool使用** 举报原因: 色情 政治 抄袭 广告 招聘 骂人 其他 原文地址: 原因补充: 最多只允许输入30个字加入CSDN,享受更精准的内容推荐,与500...