

东阳的菜鸟房~ 吾日三省吾身：还年轻么？高工资么？有首付么？好，滚去加班学习！

目录视图 摘要视图 订阅

个人资料



dongyang626



访问: 3589次

积分: 132分

排名: 千里之外

原创: 7篇 转载: 0篇

译文: 0篇 评论: 0条

文章搜索

文章分类

- 个人(1)
- 嵌入式--驱动开发(2)
- 嵌入式系统使用相关(2)
- 生活生了么? (1)
- 系统原理(1)

文章存档

- 2012年05月(1)
- 2012年04月(1)
- 2012年02月(1)
- 2011年12月(4)

阅读排行

- Busybox提示/bin/sh:can (929)
- 一个用GPIO口模拟MDC/ (878)
- ATmega168烧写Bootloa (455)
- PowerPc系统时钟频率的 (365)
- 2011都要过了, 2012还 (353)
- 安家啦 安家啦~~ (258)
- GPIO模拟I2C的几个小注 (167)

评论排行

- 安家啦 安家啦~~ (0)
- 一个用GPIO口模拟MDC/ (0)
- Busybox提示/bin/sh:can (0)
- 2011都要过了, 2012还 (0)

新版论坛系列介绍之二——功能介绍篇 公告: CSDN博客频道博客搬家功能上线! CSDN产品客服新浪微博正式上线公告
中国最大规模移动开发者高水平盛会 没有重量只有质量: iPad版《程序员杂志》应用上线 “第一次亲密接触”——有奖征文活动

一个用GPIO口模拟MDC/MDIO的驱动，实现对switch的控制——基于powerpc mpc8308

分类: 嵌入式--驱动开发 2011-12-15 21:23 878人阅读 评论(0) 收藏 举报

第一次写技术水文，不知道从何开始...

先写版权声明吧...谁让咱参考了呢：本文中部分代码学习了以下博客，在此对这位博主表示感谢

<http://blog.csdn.net/zhaoqiaoshi>

该驱动的基本功能在文章标题中就有体现了，通过CPU直接控制GPIO口，模拟MDC/MDIO发送数据帧，继而控制switch交换芯片(或者是普通phy)

开发板: Freescale 8308

内核: Linux 2.6.29

首先要明确一点，29的内核有直接的GPIO驱动，而且help中还注明了用于控制swich等，不过由于没找到调用该驱动中各个函数的文章，

就没有使用该驱动，因为跑去看人家提供好的函数怎么用的时间，还不如自己写呢。。

首先还要明确一点.. 选用的GPIO口的物理地址.. 该物理地址依赖于一个叫IMMR的东东。

查阅uboot下的mpc8308erdb.h可知：

IMMR = 0xE0000000，也就是所有的硬件地址均是在此基础上的偏移！



所有的对外设的控制，都是对相应寄存器的操作，查阅mpc8308erm.pdf:

0xC00 GPIO direction register (GPDIR) R/W

0xC04 GPIO open drain register (GPODR) R/W

后面的省略了..

所以 GPIO寄存器的实际位置为：0xE0000000 + 0XC00

要注意一点，实际开发板上22,23引脚是否被用于GPIO还需要确定下，

pdf中：SICRH寄存器列出了所有GPIO口的实际用途：其中4，5位为22，23引脚，需将其设置为：0b11

0x00114 System I/O configuration register low (SICRL) R/W 0x0000_00001

0x00118 System I/O configuration register high (SICRH) R/W 0x0014_50002

貌似基本的准备的差不多了，下面开始编程。。。

具体对swich芯片的设置，由应用程序来设，所以第一步是将该驱动注册成一个字符设备：

```
[html]
01. GPIO_MAJOR = register_chrdev(0, DEVICE_NAME, &gpio_fops);
02.     if ( GPIO_MAJOR < 0 )
03.     {
04.         printk("Can't get major %d\n", GPIO_MAJOR);
05.         return GPIO_MAJOR;
06.     }
```

此处用的是旧的注册方法，有时间的话，会将其改为用新方法注册。

register_chrdev的第一个参数0，表示设备号由系统自动分配。

分配内存空间，并通过ioremap, 将物理地址映射为可操作的内存地址..(应该没写错...)

```
[html]
01. if (!request_mem_region( gpio_phy_base, gpio_len, "gpio_test"))
02.     {
03.         printk("gpio_test: can't get I/O mem address 0x%x\n", gpio_phy_base);
04.         return -ENODEV;
05.     }
06.     gpio_virt_base = (uint32)ioremap( gpio_phy_base, gpio_len );
```

前面提到的SICRH也要进行这步操作，不然，怎么设置它的寄存器相应位为11呢~~~

下面代码为具体的设置寄存器的代码：

```
[html]
01. inline uint32 io_pin_mask_32(int pin )
02. {
03.     return (1 << (31 -pin));
04. }
05.
06.
07.
08. /*
09.  * 1 : output
10.  * 0 : input
11.  */
12. //void io_p_dir_set( mpc8308_io_t port, int pin, int data )
13. void io_p_dir_set( int pin, int data )
14. {
15.     uint32     pinmask;
16.
17.     //printk("\nio_p_dir_set: pin = %d, data = %d\n", pin, data);
18.
19.     /**/
20.     pinmask     = io_pin_mask_32( pin );
21.
22.     if( *((volatile uint32 *)(gpio_virt_base + io_p_addr.odr)) != 0 )
23.         *((volatile uint32 *)(gpio_virt_base + io_p_addr.odr)) &= ~pinmask; //清除上拉模
式
24.
25.     if( data )
26.     {
27.         *((volatile uint32 *)(gpio_virt_base + io_p_addr.dir)) |= pinmask;
28.     }
29.     else
30.     {
31.         *((volatile uint32 *)(gpio_virt_base + io_p_addr.dir)) &= ~pinmask;
32.     }
33.
34.     // printk( "gpio_virt_base[dir] %8x: %8x\n",  gpio_virt_base + io_p_addr.dir, *
(volatile uint32*)(gpio_virt_base + io_p_addr.dir));
35.     // printk( "gpio_virt_base[odr] %8x: %8x\n",  gpio_virt_base + io_p_addr.odr, *
(volatile uint32*)(gpio_virt_base + io_p_addr.odr));
```

```

36. // printk( "gpio_virt_base[dat] %8x: %8x\n", gpio_virt_base + io_p_addr.dat, *
(volatile uint32*)(gpio_virt_base + io_p_addr.dat));
37. }
38.
39. /*
40. * 1 : high level
41. * 0 : low level
42. */
43. //void io_p_dat_out( mpc8308_io_t port, int pin, int data )
44. void set_gpio_bit(int pin, int data)
45. {
46.     uint32         pinmask;
47.
48.     // printk( "\nio_p_dat_out: pin = %d, data = %d\n", pin, data);
49.
50.     /**/
51.     pinmask = io_pin_mask_32( pin );
52.
53.     /**/
54.     if( data )
55.     {
56.         *((volatile uint32 *)(gpio_virt_base + io_p_addr.dat)) |= pinmask;
57.     }
58.     else
59.     {
60.         *((volatile uint32 *)(gpio_virt_base + io_p_addr.dat)) &= ~pinmask;
61.     }
62.     /**/
63.     printk( "gpio_virt_base[dir] %8x: %8x\n",
64.             gpio_virt_base + io_p_addr.dir, *(volatile uint32*)
(gpio_virt_base + io_p_addr.dir));
65.     printk( "gpio_virt_base[odr] %8x: %8x\n",
66.             gpio_virt_base + io_p_addr.odr, *(volatile uint32*)
(gpio_virt_base + io_p_addr.odr));
67.     printk( "gpio_virt_base[dat] %8x: %8x\n",
68.             gpio_virt_base + io_p_addr.dat, *(volatile uint32*)
(gpio_virt_base + io_p_addr.dat));
69.     /**/
70. }
71.
72. /*
73. * 1
74. * 0
75. */
76. //int io_p_dat_in( mpc8308_io_t port, int pin )
77. int get_gpio_bit( int pin )
78. {
79.     uint32         pinmask = 0;
80.
81.     pinmask = io_pin_mask_32( pin );
82.
83.     return (((volatile uint32 *)
(gpio_virt_base + io_p_addr.dat)) & pinmask) ? 1 : 0;
84. }

```

上面代码确实基本沿用了文章开始博客中的代码..主要是感觉没必要改..比较好用...再次表示感谢...

寄存器现在可以设置了，那么下面就是重要的MDC/MDIO的模拟了。

这部分跟上面比较独立，可分开来看。当然，两者在 read、write函数中结合。

Management frame fields								
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
READ	L..	01	10	AAAAA	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
WRITE	L..	01	01	AAAAA	RRRRR	10	DDDDDDDDDDDDDDDD	Z

MCD/MDIO的基本知识在此就不做介绍了 这张图看懂就没什么问题了. 下面上下关键部分的代码，当初就是卡在几个关键代码上，找了无数文章都没有那一步....

网络上关于如何按照上图的格式来发送数据的文章非常多，所以那部分代码就不贴了，MDC/MDIO部分当时很卡我的是

“延时”等问题，这些地方1没代码 2没介绍..没这些代码又肯定不能通..

这个是读写1位 即发送1位数据帧过去

```
[html]
01. //Read one bit
02. static void read_bit(uint32* pdata)
03. {
04.     init_gpio_pin(MDIO, GPIO_DIR_IN,GPIO_INT_DISABLE);
05.     set_gpio_bit(MDC, 1);
06.     Delay(TIME);
07.     set_gpio_bit(MDC, 0);
08.     *pdata = get_gpio_bit(MDIO);
09. }
10.
11.
12. static void wrtie_bit(uint32 data)
13. {
14.     init_gpio_pin(MDIO, GPIO_DIR_OUT,GPIO_INT_DISABLE);
15.     if(data)
16.     { /*write 1*/
17.         set_gpio_bit(MDIO, 1);
18.         set_gpio_bit(MDC, 0);
19.         Delay(TIME);
20.         set_gpio_bit(MDC, 1);
21.     }
22.     else
23.     {
24.         set_gpio_bit(MDIO, 0);
25.         set_gpio_bit(MDC, 0);
26.         Delay(TIME);
27.         set_gpio_bit(MDC, 1);
28.     }
29. }
```

调整GPIO口状态，并设定MDC、MDIO为低电平，为上升沿做准备。

```
[html]
01. static void z_bit(void)
02. {
03.     init_gpio_pin(MDIO, GPIO_DIR_IN,GPIO_INT_DISABLE);
04.     set_gpio_bit(MDC, 0);
05.     set_gpio_bit(MDIO, 0);
06.     Delay(TIME);
07. }
```

这部分初始化代码包括了GPIO引脚寄存器具体的设置方法

```
[html]
01. void init_gpio_pin(uint32 PIN, uint32 DIR, uint32 INT)
02. {
03.     uint32 pinmask;
04.
05.     /**/
06.     pinmask = io_pin_mask_32(PIN);
07.
08.
09.     if( *((volatile uint32 *) (gpio_virt_base + io_p_addr.odr))!= 0 )
10.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.odr)) &= ~pinmask; //清除上拉模
式
11.
12.
13.     if(DIR)
14.         *((volatile uint32 *)
```

```

15.     (gpio_virt_base + io_p_addr.dir)) |= pinmask; // 1 output
16.     else
17.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.dir)) &= ~pinmask;
18.
19.     if(INT)
20.     {
21.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.ier)) |= pinmask;
22.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.imr)) |= pinmask;
23.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.icr)) |= pinmask;
24.     }
25.     else
26.     {
27.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.ier)) &= ~pinmask;
28.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.imr)) &= ~pinmask;
29.         *((volatile uint32 *) (gpio_virt_base + io_p_addr.icr)) &= ~pinmask;
30.     }
31.
32. }

```

MCD/MDIO部分完了，就到了最后一步了，通过Read、Write函数为应用层程序提供统一接口：

注册各函数。

```

[html]
01. static struct file_operations gpio_fops={
02.     .owner      = THIS_MODULE,
03.     .open       = gpio_open,
04.     .release    = gpio_release,
05.     .ioctl      = gpio_ioctl,
06.     .read       = gpio_read,
07.     .write      = gpio_write,
08. };

```

Read Write写的比较简单，目前只能对付用，还有待完善。

```

[html]
01. static ssize_t gpio_read(struct file *filp, char __user *buf, size_t cnt, loff_t *off)
02. {
03.     char * ms = "gpio_test";
04.     ssize_t len;
05.     int n;
06.     Phy_Desc phy_info= {0x1, 0x2, 0x3,0x0};
07.     len = sizeof((char*)ms);
08.
09.     copy_from_user(&phy_info, buf, cnt);
10.
11.     uint32 data = 0x00000000;
12.     mii_read(phy_info.phyaddr, phy_info.phyreg,&data);
13.
14.     phy_info.redata = data;
15.
16.     //printf("At function gpio_read the return data is phy_info.redata is 0x%x\n", ph
17.
18.     n = copy_to_user(buf, &phy_info,cnt);
19.
20.     //printf("Read done\n");
21.
22.     return n >= 0 ? len : -EFAULT;
23.
24.     return -EFAULT;
25. }
26.
27.
28. static ssize_t gpio_write(struct file *filp, const char __user *buf, size_t cnt, loff_t *of
29. {
30.     Phy_Desc phy_info= {0x0, 0x0, 0x0,0x0};
31.

```

```
32.         copy_from_user(&phy_info, buf,cnt);
33.
34.         //printf("\nphyadd:%x, regadd: %x, sedata:%x, redata:%x\n", phy_info.phyaddr, phy_
35.
36.         mii_write(phy_info.phyaddr,phy_info.phyreg, phy_info.sedata);
37.
38.         return SUCCESS;
39.     }
```

不知不觉写了1个小时。。。好累。。。而且感觉写的很不好。。。就贴了些代码，讲解的很凌乱，整篇文章逻辑也不行。。

看来还需要多积累啊~~~

上一篇: [安家啦 安家啦~~](#)

下一篇: [Busybox提示/bin/sh:can't access ttyjob control turned off 的问题，以及系统自动登录相关](#)

分享到:

[查看评论](#)

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#)

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

✉ 联系邮箱: webmaster@csdn.net

Copyright © 1999-2012, CSDN.NET, All Rights Reserved

