

i2c驱动(Linux3.4.2)

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149



dev(client)的几种创建方法

新内核下i2c的设备驱动框架

新内核下i2c的设备驱动编写

用户态直接访问i2c

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

设备的几种创建方法

新内核下的i2c驱动是采用bus-dev-drv模型设计，此模型在linux的地位举足轻重，这里的bus为i2c_bus_type

问：既然I2C驱动是bus-dev-drv模型，那么就需创建dev,怎么创建设备呢？有几种方法？

答：方法有四：

1. 定义一个i2c_board_info, i2c_register_board_info
2. 直接i2c_new_device, 或 i2c_new_probed_device
3. 从用户空间创建设备
4. 除设置i2c_driver的probe外,还设置它的detect

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

设备的几种创建方法

1. 定义一个*i2c_board_info*, *i2c_register_board_info*的核心要点:

定义一个*i2c_board_info*, 里面有: 名字, 设备地址

然后*i2c_register_board_info*(busnum, ...) //把它们放入__i2c_board_list
链表

list_add_tail(&devinfo->list, &__i2c_board_list);

问: 链表何时使用?

答: *i2c_register_adapter*

i2c_scan_static_board_info

i2c_new_device //创建dev(client)

使用限制: 必须在 *i2c_register_adapter* 之前 *i2c_register_board_info*

所以不适合我们动态加载 *insmod*

淘宝地址: 100ask.taobao.com 版权所有: <http://www.100ask.net/>

视频下载: <http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>

嵌入式交流群: 84174029 28664149, Gflash 免费获取群: 28664149

设备的几种创建方法

2. 关于i2c_new_device, 或 i2c_new_probed_device:

用法:

i2c_new_device(适配器, i2c_board_info结构体);

i2c_new_probed_device(适配器, i2c_board_info结构体, addr_list, NULL)

> probe(适配器, addr_list[i]) //确定设备是否真实存在

> info->addr = addr_list[i]

> i2c_new_device(adap, info)// ">" 表示调用关系

区别:

都是创建dev的方法, 区别是创建的前提不一样

i2c_new_device: 强制认为设备存在

i2c_new_probed_device : 对于"已经识别出来的设备"(probed_device), 才会创建("new")

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>

视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>

嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

设备的几种创建方法

3. 从用户空间创建设备

创建设备

```
echo at24c08 0x50 > /sys/class/i2c-adapter/i2c-0/new_device
```

at24c08 : i2c_driver的id_table里的name

0x50 : 设备地址

实质上间接调用i2c_new_device

删除设备

```
echo 0x50 > /sys/class/i2c-adapter/i2c-0/delete_device
```

实质上间接调用i2c_unregister_device

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>

视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>

嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

设备的几种创建方法

4.除设置i2c_driver的probe外,还设置它的detect函数

如官方例子: linux_Dlr\drivers\hwmon\lm90.c的i2c_driver结构体

```
static struct i2c_driver lm90_driver = {  
    .class    = I2C_CLASS_HWMON, // 表示去哪些适配器上找设备  
    .driver = {  
        .name    = "lm90",  
    },  
    .probe    = lm90_probe, // client的name=lm90_id的name将调用的函数  
    .remove   = lm90_remove, // 于lm90_probe作用相反  
    .id_table  = lm90_id,  
    .detect   = lm90_detect, // 用这个函数来检测设备确实存在  
    .address_list = normal_i2c, // 这些设备的地址  
}
```

方法是:去"class表示的这一类"I2C适配器, 用"detect函数"来确定能否找到"address_list"里的设备",如果能找到就调用i2c_new_device来注册i2c_client,这会 and i2c_driver的id_table比较, 如果匹配, 调用probe

设备的几种创建方法

第4种方法适用于不知道这个i2c设备在哪一条(i2c总线,i2c控制器)上,并且有一些I2C设备的地址是相同的,故用detect函数进一步分辨是哪一款,而前3种方法都要事先确定设备所在的适配器(I2C总线,I2C控制器)

问: 知道怎么构建dev,那么怎么注册/构建i2c_driver?

答: i2c_add_driver ,这是驱动程序的核心,一般在入口函数调用
出口函数一般做相反操作i2c_del_driver

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

新内核下i2c的设备驱动框架

```
i2c_add_driver
i2c_register_driver
    // at24cxx_driver放入i2c_bus_type的drv链表
    // 并且从dev链表里取出能匹配的i2c_client并调用probe
    driver_register
    ..
...
// 对于每一个适配器, 调用__process_new_driver
// 对于每一个适配器, 调用它的函数确定address_list里的设备是否存在
// 如果存在, 再调用detect进一步确定、设置, 然后i2c_new_device
i2c_for_each_dev(driver, __process_new_driver);
    __process_new_driver
        i2c_do_add_adapter
            i2c_detect(adap, driver);
            for (i = 0; address_list[i] != I2C_CLIENT_END; i += 1) {
                err = i2c_detect_address(temp_client, driver);
                // 判断这个设备是否存在: 简单的发出s信号确定有ACK
                if (!i2c_default_probe(adapter, addr))
                    return 0;
                ...
                memset(&info, 0, sizeof(struct i2c_board_info));
                info.addr = addr;

                // 故需设置info.type
                err = driver->detect(temp_client, &info);
                ...
            }
        i2c_new_device
```

i2c_add_driver流程图

新内核下i2c的设备驱动编写

在设备驱动的编写时创建dev的第4种方法并不常见，最常见的还是硬件模块(dev)和驱动模块(drv)分开写，即需要两个文件，匹配后调用驱动的probe，还是以mini2440的i2c为例

三部曲:

1.在设备文件中:调用i2c_new_device注册设备

在驱动文件中:

2. 分配和设置一个i2c_driver结构体

3. 注册i2c_driver



淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>

视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>

嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

新内核下i2c的设备驱动编写

1.在设备文件中调用i2c_new_device:

```
static struct i2c_board_info at24cxx_info = {  
    I2C_BOARD_INFO("at24c08", 0x50), // at24c08为需要和  
    i2c_driver的id_table下的名字一致, 0x50为设备地址  
}
```

宏I2C_BOARD_INFO在linux_Dir\include\linux\i2c.h定义

i2c_adap = i2c_get_adapter(0) // 获得适配器, 0表示第一个适配器

i2c_new_device(i2c_adap, &at24cxx_info) // 返回i2c_client结构体

i2c_get_adapter, i2c_new_device都是有i2c_core.c提供

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>

视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>

嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

新内核下i2c的设备驱动编写

2.在驱动文件中分配/设置i2c_driver

```
static const struct i2c_device_id at24cxx_id_table[] = {
    { "at24c08", 0 }, // at24c08为设备id即设备名,很重要, 需与
    //I2C_BOARD_INFO的dev_type一致
    {}
}

static struct i2c_driver at24cxx_driver = {
    .driver = {
        .name = "100ask", // 这个名字可随便设
        .owner = THIS_MODULE,
    },
    .probe = at24cxx_probe, // drv和dev匹配后将激发的函数
    .remove = at24cxx_remove, // 与probe相反, 一般做清理工作
    .id_table = at24cxx_id_table,
}
```

新内核下i2c的设备驱动编写

和旧内核下i2c驱动编写类似，也需提供对i2c的读和写

问: 怎么读写?

答: 在probe函数里注册字符设备/创建设备节点，在文件操作接口fops中提供读写函数, 片段如下:

```
struct file_operations at24cxx_fops = {  
    .owner = THIS_MODULE,  
    .read  = at24cxx_read,  
    .write = at24cxx_write,  
}  
register_chrdev(0, "at24cxx", &at24cxx_fops) // 参数二: proc/devices 下  
的名字  
class = class_create(THIS_MODULE, "at24cxx") // 参数二: 类名  
device_create(class, NULL, MKDEV(major, 0), NULL, "at24cxx") //  
dev/at24cxx
```

淘宝地址: 100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载: <http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash 免费获取群: 28664149

新内核下i2c的设备驱动编写

at24cxx_read:

```
copy_from_user(&addr, buf, 1)// 用户空间-内核空间  
data = i2c_smbus_read_byte_data(at24cxx_client, addr);  
copy_to_user(buf, &data, 1)// 内核空间-用户空间
```

at24cxx_write:

```
copy_from_user(ker_buf, buf, 2)// 拷贝2字节  
addr = ker_buf[0]  
data = ker_buf[1]  
i2c_smbus_write_byte_data(at24cxx_client, addr, data)
```

这里就没再构造i2c_msg结构然后调用i2c_tranfer,而是用smbus提供的API:i2c_smbus_read_byte_data和i2c_smbus_write_byte_data, 在内核文档smbus-protocol中, 这些API被推荐使用,也在这个文档中详细解释了smbus

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

用户态直接访问i2c

不自己写i2c驱动，而是调用内核提供的接口即i2c-dev.c，先下载i2c-tools并解压,把i2c-dev.h拷贝测试程序的目录，然后编写测试程序

1.先(可读写)打开设备节点

```
fd = open(argv[1], O_RDWR)// argv[1]为/dev/i2c-x(x=0,1,2...)
// 0表示第一个适配器，依此类推
```

2. ioctl设置设备地址

```
ioctl(fd, I2C_SLAVE, dev_addr)// dev_addr由用户设置
```

3. 读i2c(1个字节)

```
i2c_smbus_read_word_data(fd, addr);//addr为测试程序第4个参数
```

4. 写i2c(byte)

```
i2c_smbus_write_byte_data(fd, addr, data);// addr和data为测试程序提供的参数4和5，分别表示地址和欲设置的值
```

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>

视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>

嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149

用户态直接访问i2c

```
/ # ./i2c_usr_test  
./i2c_usr_test </dev/i2c-0> <dev_addr> r addr  
./i2c_usr_test </dev/i2c-0> <dev_addr> w addr val  
/ # ./i2c_usr_test /dev/i2c-0 0x50 r 0  
data: 2, 50, 0x32  
/ # ./i2c_usr_test /dev/i2c-0 0x50 w 0 0x61  
/ # ./i2c_usr_test /dev/i2c-0 0x50 r 0  
data: a, 97, 0x61
```

测试效果图

以上只分析了用户态测试程序的核心，具体代码还需看视频或源码

淘宝地址:100ask.taobao.com 版权所有: <http://www.100ask.net/>
视频下载:<http://pan.baidu.com/share/link?uk=2520074993&shareid=480546>
嵌入式交流群: 84174029 28664149, Gflash免费获取群:28664149