



I2C 应用开发教程

一、概述

应用发送一个数据包给 i2c 驱动实现与 i2c 设备的通讯。数据包中包含读写命令，设备地址，设备寄存器地址，以及数据等信息。

本文档讲解如何组建数据包以及如何调用 i2c 驱动接口函数实现读写指定寄存器功能

二、i2c 通讯相关数据结构介绍

```
struct i2c_msg {  
    __u16 addr; //i2c 设备地址  
    __u16 flags; //标志（读、写）  
    __u16 len; //buf 的大小  
    __u8 * buf; //数据，包括寄存器地址，发送、接收的数据  
};
```

```
struct i2c_rdwr_ioctl_data {  
    struct i2c_msg __user * msgs; //指向 struct i2c_msg 的指针  
    __u32 nmsgs; // struct i2c_msg 的个数  
};
```

该结构体是实际传递给驱动接口的数据，要实现相关的操作，需要将的参数设置为相应的值，将 struct i2c_rdwr_ioctl_data 传递给驱动接口函数。

三、驱动接口函数介绍

1、打开接口

```
fd = open( "/dev/i2c-0", O_RDWR );
```

2、IOCTL 控制

read()、write()方法都只适合于单开始信号时序，不具有太强的通用性

下面只介绍 ioctl 接口函数

```
ioctl ( fd, CMD, unsigned long arg );
```

CMD :

I2C_RETRIES 设置收不到 ACK 时的重试次数

I2C_TIMEOUT 设置超时时限的 jiffies

I2C_SLAVE 设置从机地址

I2C_SLAVE_FORCE 强制设置从机地址

I2C_TENBIT 选择地址位长:=0 for 7bit , != 0 for 10 bit

I2C_FUNCS 获取适配器支持的功能

I2C_RDWR 读写 (one STOP only)后面主要用该命令实现读写

I2C_PEC != 0 to use PEC with SMBus

I2C_SMBUS SMBus transfer

arg : 传递的参数

后续介绍使用 ioctl(fd , I2C_RDWR , unsigned long arg)实现读写



四、实现 i2c 读写功能

3、实现写入数据到指定寄存器

```
struct i2c_rdwr_ioctl_data    i2c_data;
struct i2c_msg                msgs;

i2c_data.nmsgs = 1;
i2c_data.msgs = &msgs;
unsigned char tbuf[] = { //i2c 要发送的数据，寄存器地址+数据
    REG >> 8,           //16 位寄存器用 2 字节, 8 位寄存器使用 1 字节
    REG & 0xff,
    0x66,               //要写入的数据
};
msgs.flags = 0; //写标志
msgs.addr = DEVICE_I2C_ADDR; //设备 i2c 地址
msgs.len = sizeof(tbuf); //msgs.buf 的大小
msgs.buf = tbuf; //数据包，包含寄存器及数据
组建数据包后，将 i2c_data 传递给驱动提供的读写函数即可
ioctl(fd, I2C_RDWR, (unsigned long) &i2c_data);
```

4、实现从指定寄存器读取数据

读取数据需要用到 2 个 struct i2c_msg，分别执行写、读命令
一个用来实现写入寄存器地址，另一个用来接收读取的数据

```
struct i2c_rdwr_ioctl_data i2c_data;
struct i2c_msg msgs[2];

i2c_data.nmsgs = 2;
i2c_data.msgs = msgs;
unsigned char tbuf[] = {
    reg >> 8,           //16 位寄存器用 2 字节, 8 位寄存器使用 1 字节
    reg & 0xff,
};

msgs[0].flags = 0; //写标志
msgs[0].addr = DEVICE_I2C_ADDR; //i2c 地址
msgs[0].len = sizeof(tbuf); //msgs.buf 的大小
msgs[0].buf = tbuf; //寄存器地址

unsigned char rbuf ;
msgs[1].flags = I2C_M_RD; //读标志，实际值为 0x1
msgs[1].addr = DEVICE_I2C_ADDR; //i2c 地址
msgs[1].len = 1; //读入的数据长度，此处读入 1 字节
msgs[1].buf = &rbuf; //读入数据的存放地址
组建数据包后，将 i2c_data 传递给驱动提供的读写函数
即可实现读取寄存器的值
```



```
ioctl(fd, I2C_RDWR, (unsigned long) &i2c_data);
```

五、实现读写指定寄存器功能的测试程序 sample.c