http://bo-srv101/redmine/issues/3256

该issue的描述如下

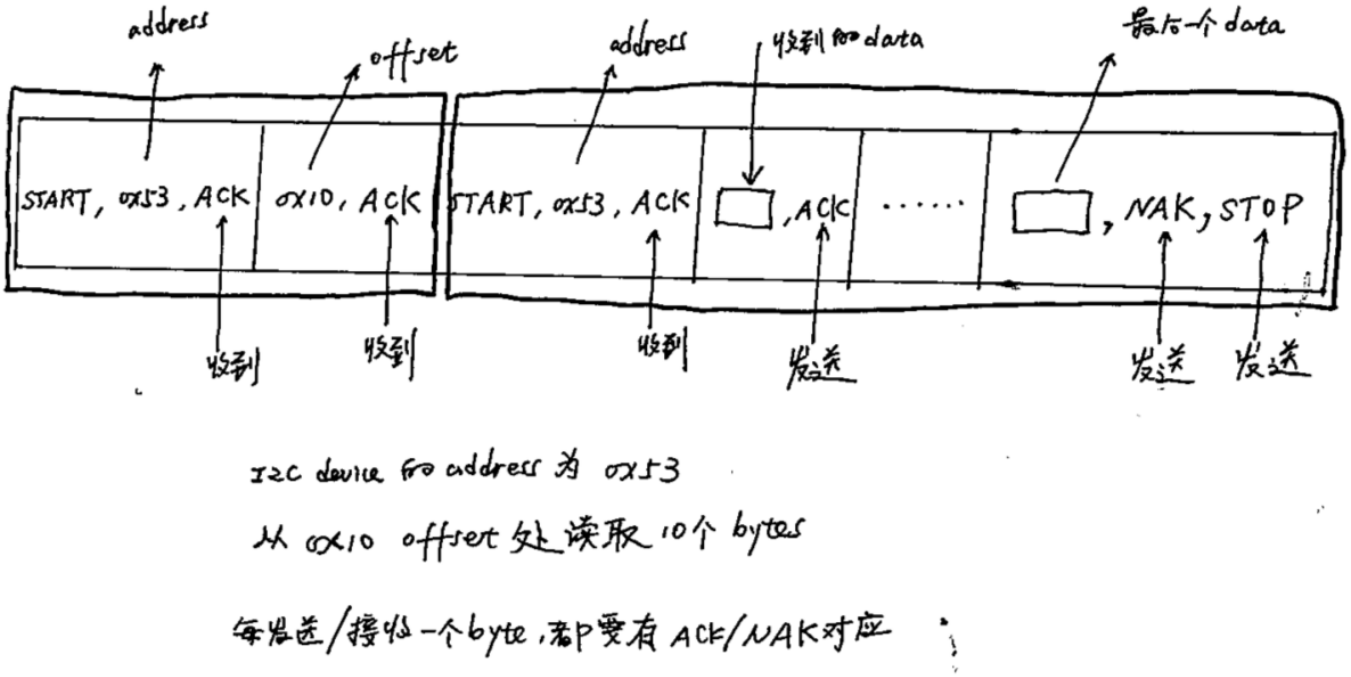
The i2c-pxa driver used for I2C support on granite2 supports repeated start for a write followed by a read (likely the more common case), but does not support repeated start for a read followed by a write. We currently have a customer that uses I2C to communicate between the 6270 and a separate custom board using both write followed by read and read followed by write repeated start sequences.

Both customer testing and inspection of the i2c-pxa driver show the driver supports repeated start on a write followed by a read but does not support repeated start on a read followed by a write. The code for supporting a repeated start for transitioning from a write to another command (read or write) is in i2c_pxa_irq_txempty() with "repeated start" in the comments. To support a repeated start when transitioning from a read to another command (write or read), code similar to this needs to be added to the i2c_pxa_irq_rxfull() function.

```
int i2c_master_sendrecv(const struct i2c_client *client,
                                                const char *sndbuf,
                                                int sndcount,
                                                char *rcvbuf,
                                                int rcvcount)
{
        int                             ret;
        struct i2c_adapter      *adap = client->adapter;
        struct i2c_msg          msg2;

        msg0.addr = msg1.addr = client->addr;
        msg0.flags = msg1.flags = client->flags & I2C_M_TEN;
        msg1.flags |= I2C_M_RD;
        msg0.len = sndcount;
        msg0.buf = (char *)sndbuf;
        msg1.len = rcvcount;
        msg1.buf = (char *)rcvbuf;

        ret = i2c_transfer(adap, msg, 2);

        return (ret == 2) ? 0 : -1;
}

int i2c_master_recvsend(const struct i2c_client *client,
                                                char *rcvbuf,
                                                int rcvcount,
                                                const char *sndbuf,
                                                int sndcount)
{
        int                             ret;
        struct i2c_adapter      *adap = client->adapter;
        struct i2c_msg          msg2;

        msg0.addr = msg1.addr = client->addr;
        msg1.flags = msg0.flags = client->flags & I2C_M_TEN;
        msg0.flags |= I2C_M_RD;
        msg0.len = rcvcount;
        msg0.buf = (char *)rcvbuf;
        msg1.len = sndcount;
        msg1.buf = (char *)sndbuf;

        ret = i2c_transfer(adap, msg, 2);

        return (ret == 2) ? 0 : -1;
}
```

根据描述，i2c_master_sendrecv()在现有i2c-pxa driver中已经支持，但i2c_master_recvsend()部支持！

i2c_master_sendrecv()就有点像对eeprom的读操作。



msg0发送的是eeprom的offset，而msg1则是要读取的内容。
这里关键是msg0不能发送STOP signal.
i2c_master_recvsend()在eeprom的场景中好像没有。