NAME

smb - SMB generic I/O device driver

SYNOPSIS

device smb

DESCRIPTION

The *smb* character device driver provides generic I/O to any smbus(4) instance. To control SMB devices, use */dev/smb?* with the ioctls described below. Any of these ioctl commands takes a pointer to *struct smbcmd* as its argument.

```
#include <sys/types.h>

struct smbcmd {
        u_char cmd;
        u_char reserved;
        u_short op;
        union {
                char    byte;
                char    buf[2];
                short   word;
        } wdata;
        union {
                char    byte;
                char    buf[2];
                short   word;
        } rdata;
        int  slave;
        char *wbuf;     /* use wdata if NULL */
        int  wcount;
        char *rbuf;     /* use rdata if NULL */
        int  rcount;
};
```

The *slave* field is always used, and provides the address of the SMBus slave device. The slave address is specified in the seven most significant bits (i.e., "left-justified"). The least significant bit of the slave address must be zero.

Ioctl *Description*

SMB_QUICK_WRITE	*QuickWrite* does not transfer any data. It just issues the device address with write intent to the bus.
SMB_QUICK_READ	*QuickRead* does not transfer any data. It just issues the device address with read intent to the bus.
SMB_SENDB	*SendByte* sends the byte provided in *cmd* to the device.
SMB_RECVB	*ReceiveByte* reads a single byte from the device which is returned in *cmd*.
SMB_WRITEB	*WriteByte* first sends the byte from *cmd* to the device, followed by the byte given in *wdata.byte*.
SMB_WRITEW	*WriteWord* first sends the byte from *cmd* to the device, followed by the word given in *wdata.word*. Note that the SMBus byte-order is little-endian by definition.
SMB_READB	*ReadByte* first sends the byte from *cmd* to the device, then reads one byte of data from the device. Returned data is stored in *rdata.byte*.
SMB_READW	*ReadWord* first sends the byte from *cmd* to the device, then reads one word of data from the device. Returned data is stored in *rdata.word*.
SMB_PCALL	*ProcedureCall* first sends the byte from *cmd* to the device, followed by the word provided in *wdata.word*. It then reads one word of data from the device and returns it in *rdata.word*.
SMB_BWRITE	*BlockWrite* first sends the byte from *cmd* to the device, then the byte from *wcount* followed by *wcount* bytes of data that are taken from the buffer pointed to by *wbuf*. The SMBus specification mandates that no more than 32 bytes of data can be transferred in a single block read or write command. This value can be read from the constant SMB_MAXBLOCKSIZE.
SMB_BREAD	*BlockRead* first sends the byte from *cmd* to the device, then reads a count of data bytes that the device is going to provide and then reads that many bytes. The count is returned in *rcount*. The data is returned in the buffer pointed to by *rbuf*.

The read(2) and write(2) system calls are not implemented by this driver.

ERRORS

The ioctl(2) commands can cause the following driver-specific errors:

[ENXIO]	Device did not respond to selection.
[EBUSY]	Device still in use.
[ENODEV]	Operation not supported by device (not supposed to happen).

[EINVAL]	General argument error.
[EWOULDBLOCK]	SMBus transaction timed out.

SEE ALSO

ioctl(2), smbus(4)

HISTORY

The **smb** manual page first appeared in FreeBSD 3.0.

AUTHORS

This manual page was written by Nicolas Souchu and extended by Michael Gmelin <freebsd@grem.de>.