



**Руководство (v1.0)**

**По работе с драйвером модуля  
“mPCIe-1553UD”**

Интерфейс ГОСТ Р 52070-2003  
(MIL-STD-1553B)

Для драйверов версии 1.0 и ниже

**ОС МСВС 3.0 (ядро 2.4.32)**



**23.03.2015**

**ООО “Новомар” 2015 г**

## Оглавление

1. Введение.....	4
2. Установка драйвера.....	4
3. Подключение файла с командами к разрабатываемому проекту.....	4
4. Список доступных команд по версиям драйверов.....	5
5. Описание использования команд драйвера.....	6
6. Общие команды для всех режимов.....	7
6.1. IOCTL_ENABLE_DMA.....	7
6.2. IOCTL_DISABLE_DMA.....	7
6.3. IOCTL_RESETPOINT.....	7
6.4. IOCTL_GET_NBLOCK_RAW_DMA.....	8
6.5. IOCTL_RD_RAW_DMA.....	8
6.6. IOCTL_SET_ADDRESS.....	9
6.7. IOCTL_ENABLE_CHANNEL.....	9
6.8. IOCTL_DISABLE_CHANNEL.....	9
6.9. IOCTL_SET_TIMER_TIMEOUT.....	10
6.10. IOCTL_SWITCH_MODE.....	10
6.11. IOCTL_VERSION.....	10
7. Работа в режиме контроллера канала.....	11
7.1. IOCTL_WRITE_BC_CONTR_REG.....	11
7.2. IOCTL_READ_BC_CONTR_REG.....	11
9. Работа в режиме оконечного устройства.....	12
9.1. IOCTL_ENABLE_PADDR.....	12
9.2. IOCTL_DISABLE_PADDR.....	12
9.3. IOCTL_WR_REG_PA_RT.....	13
9.4. IOCTL_RD_REG_PA_RT.....	13
9.5. IOCTL_WR_BLOCK_BUF_PA.....	14
9.6. IOCTL_RD_BLOCK_BUF_PA.....	14

10.	Работа в режиме монитора канала. ....	15
10.1.	IOCTL_WR_REG_PA_DIS .....	15
10.2.	IOCTL_RD_REG_PA_DIS .....	15
11.	Команды прямого доступа к регистрам платы. ....	16
11.1.	IOCTL_WR_REG_OFFSET .....	16
11.2.	IOCTL_RD_REG_OFFSET .....	16
11.3.	IOCTL_WR_REG .....	17
11.4.	IOCTL_RD_REG .....	17
12.	Прерывания. ....	18
12.1.	IOCTL_DISABLE_INT_HDAT .....	18
12.2.	IOCTL_DISABLE_INT_QDAT .....	18
12.3.	IOCTL_DISABLE_INT_FLASH .....	19
12.4.	IOCTL_DISABLE_INT_BC .....	19
12.5.	IOCTL_DISABLE_INT_MC .....	19
12.6.	IOCTL_DISABLE_INT_ERR .....	20
12.7.	IOCTL_ENABLE_INT_IO_SA_PID_SIG .....	20
13.	Обновление драйвера. ....	21
14.	Обновление руководства .....	21

## 1. Введение.

Драйвер для модуля “mPCIe – 1553UD”, поддерживает одновременную работу не более 100 устройств.

Устройствам следует присвоить уникальные символьные имена вида “man\_ud\_dev\_x”, где x — индекс устройства, начиная с 0. Устройства находятся в папке “/dev” .

Каждая плата представляет отдельный файл устройства в файловой системе.

Взаимодействие с драйвером происходит посредством ioctl-команд, перечень которых находится в файле “ioctl\_man.h”.

Существует четыре режима работы платы: режим контроллера канала, режим оконечного устройства, режим монитора канала, режим монитора канала с адресацией.

## 2. Установка драйвера.

1. Создайте папку /modules в корне файловой системы.
2. Поместите в созданную папку каталог man\_ud\_drv, содержащую файлы drvmanud.ko, version.txt и ioctl\_man.h.
3. В консоли введите команду “insmod /modules/man\_ud\_drv/drvmanud.ko”.
4. Драйвер установлен, но необходимо создать соответствующий файл устройства в каталоге /dev командой:

```
mknod /dev/ man_ud_dev_B c N B
```

, где

- N - major номер, присвоенный драйверу. Узнать этот номер можно в файле /proc/devices;
- B – minor номер устройства. Первому устройству следует назначить 0, второму 1 и т.д.

Пример для первого устройства в системе и major номером 253:

```
mknod /dev/ man_ud_dev_0 c 253 0
```

Для второго устройства команда будет выглядеть вот так:

```
mknod /dev/ man_ud_dev_1 c 253 1
```

**Если Вы хотите проверить, загружен ли драйвер в ядро:**

1. В терминале введите команду “lsmod”.
2. Найдите в выведенном списке “drvmanud”.
3. Если есть — драйвер успешно установлен, если нет — драйвер не работает.

**Для обновления версии драйвера:**

1. Посмотреть версию драйвера (прикладывается в файле version.txt в папке с драйвером) .
2. Сравнить с версией текущего (файл /modules/man\_ud\_drv/version.txt).
3. Выбрать более позднюю.
4. Заменить файлы drvmanud.ko и version.txt в папке /modules/man\_ud\_drv на новые (сохраняя имена!).
5. Перезагрузить компьютер.

## 3. Подключение файла с командами к разрабатываемому проекту.

1. Скопировать файл “ioctl\_man.h” из папки /modules/man\_ud\_drv в папку с проектом.
2. В начале проекта добавить строку:

```
#include «ioctl_man.h»
```
3. Использовать команды.

Формат описан ниже; примеры использования приложены.

#### 4. Список доступных команд по версиям драйверов.

Название вызова	Краткое описание
Список вызовов, доступных в драйвере версии до 1.0 и ниже	
<a href="#">IOCTL_ENABLE_DMA</a>	Разрешить работу DMA
<a href="#">IOCTL_DISABLE_DMA</a>	Запретить работу DMA
<a href="#">IOCTL_RESETPOINT</a>	Сбросить указатель DMA
<a href="#">IOCTL_GET_NBLOCK_RAW_DMA</a>	Кол-во новых блоков данных в DMA
<a href="#">IOCTL_RD_RAW_DMA</a>	Чтение данных из буфера DMA
<a href="#">IOCTL_SET_ADDRESS</a>	Установка адреса ОУ
<a href="#">IOCTL_ENABLE_CHANNEL</a>	Разрешение работы ОУ
<a href="#">IOCTL_DISABLE_CHANNEL</a>	Запрет работы ОУ
<a href="#">IOCTL_SET_TIMER_TIMEOUT</a>	Установка таймера
<a href="#">IOCTL_SWITCH_MODE</a>	Установка режима работы
<a href="#">IOCTL_VERSION</a>	Информация о плате
<a href="#">IOCTL_WRITE_BC_CONTR_REG</a>	Запись инструкции, операции или данных
<a href="#">IOCTL_READ_BC_CONTR_REG</a>	Чтение инструкции, операции или данных
<a href="#">IOCTL_ENABLE_PADDR</a>	Разрешение подадреса
<a href="#">IOCTL_DISABLE_PADDR</a>	Запрет подадреса
<a href="#">IOCTL_WR_REG_PA_RT</a>	Запись в регистр подадреса
<a href="#">IOCTL_RD_REG_PA_RT</a>	Чтение регистра подадреса
<a href="#">IOCTL_WR_BLOCK_BUF_PA</a>	Запись данных в буфер передачи
<a href="#">IOCTL_RD_BLOCK_BUF_PA</a>	Чтение данных из буфера передачи
<a href="#">IOCTL_WR_REG_PA_DIS</a>	Запись в регистр подадреса
<a href="#">IOCTL_RD_REG_PA_DIS</a>	Чтение регистра подадреса
<a href="#">IOCTL_WR_REG_OFFSET</a>	Запись в регистр
<a href="#">IOCTL_RD_REG_OFFSET</a>	Чтение регистра
<a href="#">IOCTL_WR_REG</a>	Запись в регистр
<a href="#">IOCTL_RD_REG</a>	Чтение регистра
<a href="#">IOCTL_DISABLE_INT_HDAT</a>	Отключение прерывания HDAT
<a href="#">IOCTL_DISABLE_INT_QDAT</a>	Отключение прерывания QDAT
<a href="#">IOCTL_DISABLE_INT_FLASH</a>	Отключение прерывания FLASH
<a href="#">IOCTL_DISABLE_INT_BC</a>	Отключение прерывания BC
<a href="#">IOCTL_DISABLE_INT_MC</a>	Отключение прерывания MC
<a href="#">IOCTL_DISABLE_INT_ERR</a>	Отключение прерывания ERR
<a href="#">IOCTL_ENABLE_INT_IO_SA_PID_SIG</a>	Включение прерываний и сигналов.

## 5. Описание использования команд драйвера.

Структура ioctl-команды:

*int ioctl(int fd, IOCTL\_..., unsigned long param),*

где:

int fd – дескриптор файла, полученный при вызове функции open для файла устройства;

IOCTL\_... – команда из набора, описанного в файле ioctl\_man.h;

unsigned long param – параметр, передаваемый с командой. Содержимое параметра зависит от команды (см. описание команд ниже).

Команда, в случае удачного выполнения запроса, возвращает 0.

В случае неудачного выполнения запроса возвращается значение, отличное от 0, что означает:

1. запрос был отклонен ОС (-EBUSY и т.п.);

2. -EACCESS - запрос не был выполнен драйвером (команда отсутствует, не может быть выполнена в данном режиме платы, некорректные параметры для данной платы/команды).

**После загрузки драйвера запрещена работа устройства, работа DMA, все прерывания, все подадреса, все командные слова, указатель DMA сброшен в 0, во все подадреса передачи записаны 0.**

## 6. Общие команды для всех режимов.

### 6.1. IOCTL\_ENABLE\_DMA

**Назначение:**

Начать (продолжить) работу DMA платы.

**Входные параметры:**

param – не используется

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_ENABLE_DMA, 0)) {
//ошибка
}
else {
// DMA разрешен
}
```

### 6.2. IOCTL\_DISABLE\_DMA

**Назначение:**

Остановить работу DMA платы.

**Входные параметры:**

param – не используется

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_DMA, 0)) {
//ошибка
}
else {
// DMA запрещен
}
```

### 6.3. IOCTL\_RESETPOINT

**Назначение:**

Сбросить указатель данных буфера DMA платы.

Режим работы DMA не изменяется.

**Входные параметры:**

param – не используется

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_RESETPOINT, 0))
//ошибка
}
else {
// указатель DMA сброшен
}
```

## 6.4. IOCTL\_GET\_NBLOCK\_RAW\_DMA

### Назначение:

Получить кол-во блоков данных, накопленных в буфере DMA платы по всем каналам (блок - 128 байт).

### Входные параметры:

param – указатель на переменную типа unsigned int, в которую будет помещено кол-во блоков

### Пример вызова:

```
#include "ioctl_man.h"
unsigned int nb;
if (ioctl(fd, IOCTL_GET_NBLOCK_RAW_DMA, &nb)) {
//ошибка
}
else {
// nb = кол-во непрочитанных блоков в буфере DMA
}
```

## 6.5. IOCTL\_RD\_RAW\_DMA

### Назначение:

Считать из буфера DMA кол-во блоков DMA\_STR.number\_block, поместив блоки в массив по указателю DMA\_STR.buf.

После выполнения команды указатель буфера DMA сдвигается на соответствующую кол-ву прочитанного позицию.

### Входные параметры:

param – указатель на структуру типа DMA\_STR

### Пример вызова:

```
#include "ioctl_man.h"
struct {
unsigned int number_block; // Кол-во запрашиваемых/полученных
блоков
char b[10*128]; // Место под 10 блоков
} d;
d.number_block = 10;
if (ioctl(fd, IOCTL_RD_RAW_DMA, &d)) {
//ошибка
}
else {
// d.number_block = кол-во блоков, скопированных из буфера
DMA (может быть меньше
//запрошенных!)
//d.buf0]- d.buf d.number_block*128] – скопированная из буфера
DMA информация
}
```



## 6.6. IOCTL\_SET\_ADDRESS

**Назначение:**

Установка адреса ОУ AD\_STR.ad ( 0...31) в канале.

**Входные параметры:**

param – указатель на структуру типа AD\_STR

**Пример вызова:**

```
#include "ioctl_man.h"
AD_STR sad;
sad.ad = 3;
if (ioctl(fd, IOCTL_SET_ADDRESS, &sad)) {
//ошибка
}
else {
// Для устройства установлен адрес 3
}
```

## 6.7. IOCTL\_ENABLE\_CHANNEL

**Назначение:**

Разрешение работы ОУ канала.

**Входные параметры:**

param – не используется

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_ENABLE_CHANNEL, 0)) {
//ошибка
}
else {
// канал разрешен, шины A, B включены
}
```

## 6.8. IOCTL\_DISABLE\_CHANNEL

**Назначение:**

Запрет работы ОУ канала.

**Входные параметры:**

param – не используется

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_CHANNEL, 0)) {
//ошибка
}
else {
// канал запрещен, шины A,B выключены
}
```

## 6.9. IOCTL\_SET\_TIMER\_TIMEOUT

### Назначение:

Установка разрешения таймера TT.timer (0-выкл.,1,2,4,8,16,32,64 ), таймаута TT.timeout\_rcv (17,60, 85,110) приема и таймаута TT.timeout\_tr (6,8,11,13,18,61,86,111) передачи канала. (Все значения в мкс).

### Входные параметры:

param – указатель на структуру типа AD\_STR

### Пример вызова:

```
#include "ioctl_man.h"
TT tit;
tit.timer = 4;
tit.timeout_rcv = 17;
tit.timeout_tr = 6;
if (ioctl(fd, IOCTL_SET_TIMER_TIMEOUT, &tit)) {
//ошибка
}
else {
// В канале установлено разрешение таймера 4 мкс, таймаут
приема 17 мкс и таймаут передачи 6 мкс.
}
```

## 6.10. IOCTL\_SWITCH\_MODE

### Назначение:

Установка режима работы. RT\_MODE, BC\_MODE, BM\_MODE или BM\_ADDR\_MODE.

### Входные параметры:

param – указатель типа short

### Пример вызова:

```
#include "ioctl_man.h"
MODE_STR mode;
mode.m = BM_ADDR_MODE;
if (ioctl(fd, IOCTL_SWITCH_MODE, &mode)) {
//ошибка
}
else {
//режим работы установлен RT (оконечное устройство)
}
```

## 6.11. IOCTL\_VERSION

### Назначение:

Запрос информации о плате.

### Входные параметры:

Param – переменная типа VERSION.

### Пример вызова:

```
#include "ioctl_man.h"
VERSION ver;
if (ioctl(fd, IOCTL_VERSION, &ver)) {
//ошибка
}
```

```

}else {
// структура ver содержит информацию о плате }

```

## 7. Работа в режиме контроллера канала.

### 7.1. IOCTL\_WRITE\_BC\_CONTR\_REG

#### Назначение:

Запись (BC\_RAM\_STR.data) инструкции (INSTR), операции(OPER) или данных(DATA) (BC\_RAM\_STR.type) по смещению (BC\_RAM\_STR.offset).

#### Входные параметры:

param — указатель на структуру типа BC\_RAM\_STR

#### Пример вызова:

```

#include «ioctl_man.h»
BC_RAM_STR bc;
bc.type = INSTR;
bc.offset = 2;
bc.data = xxxxxx;
if (ioctl(fd, IOCTL_WRITE_BC_CONTR_REG, &bc)) {
//ошибка
}
else {
//3-е слово инструкции стало равно bc.data
}

```

### 7.2. IOCTL\_READ\_BC\_CONTR\_REG

#### Назначение:

Чтение (BC\_RAM\_STR.data) инструкции (INSTR), операции(OPER) или данных(DATA) (BC\_RAM\_STR.type) по смещению (BC\_RAM\_STR.offset).

#### Входные параметры:

param — указатель на структуру типа BC\_RAM\_STR

#### Пример вызова:

```

#include «ioctl_man.h»
BC_RAM_STR bc;
bc.type = INSTR;
bc.offset = 2;
if (ioctl(fd, IOCTL_READ_BC_CONTR_REG, &bc)) {
//ошибка
}
else {
//bc.data стало равно 3-ему слову инструкции
}

```

## 9. Работа в режиме оконечного устройства.

### 9.1. IOCTL\_ENABLE\_PADDR

**Назначение:**

Разрешение подадреса номер SADDR\_DATA\_PA.paddr приема/передачи SADDR\_DATA\_PA.daddr (MIL\_RT\_RCV\_REG/MIL\_RT\_TR\_REG) .

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA\_PA.

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA_PA sadpa;
sadpa.daddr = MIL_RT_RCV_REG;
sadpa.paddr = 3;
// поле data структуры SADDR_DATA в данной команде не
используются
if (ioctl(fd, IOCTL_ENABLE_PADDR, &sadpa)) {
//ошибка
}
else {
// Подадрес 3 на прием разрешен
}
```

### 9.2. IOCTL\_DISABLE\_PADDR

**Назначение:**

Запрет подадреса номер SADDR\_DATA\_PA.paddr приема/передачи SADDR\_DATA\_PA.daddr (MIL\_RT\_RCV\_REG/MIL\_RT\_TR\_REG) .

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA\_PA.

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA_PA sadpa;
sadpa.daddr = MIL_RT_TR_REG;
sadpa.paddr = 5;
// поле data структуры SADDR_DATA в данной команде не
используются
if (ioctl(fd, IOCTL_DISABLE_PADDR, &sadpa)) {
//ошибка
}
else {
// Подадрес 5 на передачу запрещен
}
```

### 9.3. IOCTL\_WR\_REG\_PA\_RT

**Назначение:**

Запись в регистр SADDR\_DATA\_PA.daddr (*MIL\_RT\_RCV\_REG*, *MIL\_RT\_TR\_REG*) поадреса SADDR\_DATA\_PA.paddr значения SADDR\_DATA\_PA.data.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA\_PA.

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA_PA sadpa;
sadpa.daddr = MIL_RT_RCV_REG;
sadpa.paddr = 3;
sadpa.data = 0;
// поле buf1 структуры SADDR_DATA в данной команде не
// используются
if (ioctl(fd, IOCTL_WR_REG_PA, &sadpa)) {
//ошибка
}
else {
// Значение 0 записано в регистр MIL_RT_RCV_REG поадреса
3
}
```

### 9.4. IOCTL\_RD\_REG\_PA\_RT

**Назначение:**

Чтение из регистра SADDR\_DATA\_PA.daddr (*MIL\_RT\_RCV\_REG*, *MIL\_RT\_TR\_REG*) поадреса SADDR\_DATA\_PA.paddr значения и копирование его в SADDR\_DATA\_PA.data.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA\_PA.

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA_PA sadpa;
sadpa.daddr = MIL_RT_TR_REG;
sadpa.paddr = 7;
// поле buf1 структуры SADDR_DATA в данной команде не
// используются
if (ioctl(fd, IOCTL_RD_REG_PA, &sadpa)) {
//ошибка
}
else {
// sadpa.data = значение регистра MIL_RT_TR_REG поадреса 7
}
```

## 9.5. IOCTL\_WR\_BLOCK\_BUF\_PA

### Назначение:

Запись блока данных SBLOCK\_BUF\_PA.buf[32] в буфер передачи номер SBLOCK\_BUF\_PA.buf (0, 1) подадреса SBLOCK\_BUF\_PA.paddr.

Режим работы должен быть выставлен в VM\_ADDR\_MODE или RT\_MODE.

### Входные параметры:

Param – переменная типа SBLOCK\_BUF\_PA

### Пример вызова:

```
#include "ioctl_man.h"
SBLOCK_BUF_PA sbbpa;
sbbpa.num_buf = 0;
sbbpa.paddr = 3;
sbbpa.buf[0] = 0x5555;
sbbpa.buf[31] = 0xaaaa;
if (ioctl(fd, IOCTL_WR_BLOCK_BUF_PA, &sbbpa)) {
//ошибка
}
else {
// Буфер sbbpa.buf записан в буфер 0 подадреса передачи 3
}
```

## 9.6. IOCTL\_RD\_BLOCK\_BUF\_PA

### Назначение:

Чтение блока данных из буфера передачи номер SBLOCK\_BUF\_PA.buf (0, 1) подадреса SBLOCK\_BUF\_PA.paddr и копирование его в SBLOCK\_BUF\_PA.buf[32]

Режим работы должен быть выставлен в VM\_ADDR\_MODE или RT\_MODE.

### Входные параметры:

Param – переменная типа SBLOCK\_BUF\_PA

### Пример вызова:

```
#include "ioctl_man.h"
SBLOCK_BUF_PA sbbpa;
sbbpa.num_buf = 1;
sbbpa.paddr = 8;
if (ioctl(fd, IOCTL_RD_BLOCK_BUF_PA, &sbbpa)) {
//ошибка
}
else {
// sbbpa.buf[32] = содержимому буфера 1 подадреса передачи 8
}
```

## 10. Работа в режиме монитора канала.

### 10.1. IOCTL\_WR\_REG\_PA\_DIS

**Назначение:**

Запись в регистр SADDR\_DATA\_PA.daddr (*MIL\_DIS\_RCV\_REG*, *MIL\_DIS\_TR\_REG*) подадреса SADDR\_DATA\_PA.paddr значения SADDR\_DATA\_PA.data.

Режим работы должен быть выставлен в *BM\_ADDR\_MODE* или *BM\_MODE*.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA\_PA.

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA_PA sadpa;
sadpa.daddr = MIL_DIS_RCV_REG;
sadpa.paddr = 3;
sadpa.data = 0;
// поле buf1 структуры SADDR_DATA в данной команде не
// используются
if (ioctl(fd, IOCTL_WR_REG_PA, &sadpa)) {
//ошибка
}
else {
// Значение 0 записано в регистр MIL_RT_RCV_REG подадреса
3
}
```

### 10.2. IOCTL\_RD\_REG\_PA\_DIS

**Назначение:**

Чтение из регистра SADDR\_DATA\_PA.daddr (*MIL\_DIS\_RCV\_REG*, *MIL\_DIS\_TR\_REG*) подадреса SADDR\_DATA\_PA.paddr значения и копирование его в SADDR\_DATA\_PA.data.

Режим работы должен быть выставлен в *BM\_ADDR\_MODE* или *BM\_MODE*.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA\_PA.

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA_PA sadpa;
sadpa.daddr = MIL_DIS_TR_REG;
sadpa.paddr = 7;
// поле buf1 структуры SADDR_DATA в данной команде не
// используются
if (ioctl(fd, IOCTL_RD_REG_PA, &sadpa)) {
//ошибка
}
else {
// sadpa.data = значение регистра MIL_RT_TR_REG подадреса 7
}
```

## 11. Команды прямого доступа к регистрам платы.

### 11.1. IOCTL\_WR\_REG\_OFFSET

**Назначение:**

Запись в регистр по смещению SADDR\_DATA.daddr относительно базового адреса значения SADDR\_DATA.data.

**ВНИМАНИЕ.**

**Рекомендуется использовать только опытным пользователям.**

**Использование ошибочного смещения может привести к неисправности платы. Запись в регистры не контролируется.**

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA sad;
sad.daddr = 0x2000;
sad.data = 0;
// поля buf1 и channel структуры SADDR_DATA в данной
команде не используются
if (ioctl(fd, IOCTL_WR_REG_OFFSET, &sad)) {
//ошибка
}
else {
// Значение 0 записано в регистр 0x2000
}
```

### 11.2. IOCTL\_RD\_REG\_OFFSET

**Назначение:**

Чтение из регистра по смещению SADDR\_DATA.daddr относительно базового адреса значения и копирование его в SADDR\_DATA.data.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA sad;
sad.daddr = 0x2000;
// поля buf1 и channel структуры SADDR_DATA в данной
команде не используются
if (ioctl(fd, IOCTL_RD_REG_OFFSET, &sad)) {
//ошибка
}
else {
// В sad.data – содержимое регистра 0x2000
}
```



### 11.3. IOCTL\_WR\_REG

**Назначение:**

Запись в регистр SADDR\_DATA.daddr значения SADDR\_DATA.data.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA sad;
sad.daddr = MIL_FREE_TIMER;
sad.data = 0;
// поле buf1 структуры SADDR_DATA в данной команде не
// используются
if (ioctl(fd, IOCTL_WR_REG, &sad)) {
//ошибка
}
else {
// Значение 0 записано в регистр MIL_FREE_TIMER (таймер
сброшен)
}
```

### 11.4. IOCTL\_RD\_REG

**Назначение:**

Чтение из регистра SADDR\_DATA.daddr значения и копирование его в SADDR\_DATA.data.

**Входные параметры:**

param – указатель на структуру типа SADDR\_DATA

**Пример вызова:**

```
#include "ioctl_man.h"
SADDR_DATA sad;
sad.daddr = MIL_FREE_TIMER;
// поле buf1 структуры SADDR_DATA в данной команде не
// используются
if (ioctl(fd, IOCTL_RD_REG, &sad)) {
//ошибка
}
else {
// sad.data = текущее значение таймера
}
```

## 12. Прерывания.

Механизм прерываний основан на использовании сигналов. На каждое прерывание можно назначить индивидуальные сигнал и процесс. Примеры использования приложены к драйверу.

Возможные прерывания:

SINTHIOSA\_INTR\_HDAT — заполнение буфера DMA на 1/16

SINTHIOSA\_INTR\_QDAT — заполнение буфера DMA на 1/2

SINTHIOSA\_INTR\_FLASH — прерывание контроллера FLASH

SINTHIOSA\_INTR\_MC — прерывание при получении Команды Управления

SINTHIOSA\_INTR\_ERR — прерывание по ошибке

SINTHIOSA\_INTR\_SADDR — прерывание поадреса

SINTHIOSA\_INTR\_BC — прерывание контроллера канала

Команды:

### 12.1. IOCTL\_DISABLE\_INT\_HDAT

**Назначение:**

Отключить посылку сигнала по прерыванию HDAT.

**Входные параметры:**

Param – не используется.

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_INT_HDAT, 0)) {
//ошибка
}
else {
//
}
```

### 12.2. IOCTL\_DISABLE\_INT\_QDAT

**Назначение:**

Отключить посылку сигнала по прерыванию QDAT.

**Входные параметры:**

Param – не используется.

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_INT_QDAT, 0)) {
//ошибка
}
else {
//
}
```

### 12.3. IOCTL\_DISABLE\_INT\_FLASH

**Назначение:**

Отключить посылку сигнала по прерыванию FLASH.

**Входные параметры:**

Param – не используется.

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_INT_FLASH, 0)) {
//ошибка
}
else {
//
}
```

### 12.4. IOCTL\_DISABLE\_INT\_BC

**Назначение:**

Отключить посылку сигнала по прерыванию FLASH.

**Входные параметры:**

Param – не используется.

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_INT_BC, 0)) {
//ошибка
}
else {
//
}
```

### 12.5. IOCTL\_DISABLE\_INT\_MC

**Назначение:**

Отключить посылку сигнала по прерыванию MC.

**Входные параметры:**

Param – не используется.

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_INT_CH_MC, 0)) {
//ошибка
}
else {
//
}
```

## 12.6. IOCTL\_DISABLE\_INT\_ERR

**Назначение:**

Отключить посылку сигнала по прерыванию ERR.

**Входные параметры:**

Param – не используется.

**Пример вызова:**

```
#include "ioctl_man.h"
if (ioctl(fd, IOCTL_DISABLE_INT_CH_ERR, 0)) {
//ошибка
}
else {
//
}
```

## 12.7. IOCTL\_ENABLE\_INT\_IO\_SA\_PID\_SIG

**Назначение:**

Включение прерываний и назначение сигналов.

**Входные параметры:**

param – указатель на структуру типа SINTHIOSA

**Пример вызова:**

```
#include "ioctl_man.h"
SINTHIOSA sin;
sin.intr = SINTHIOSA_INTR_QDAT; // прерывание, по которому
будет отправлен сигнал
sin.proc_pid = pid; // pid процесса, на который Вы хотите
получить данный сигнал
sin.sig_no = signal; // сигнал, который Вы хотите получить
if (ioctl(fd, IOCTL_ENABLE_INT_IO_SA_PID_SIG, &sin)) {
//ошибка
}
else {
//
}
```

### 13. Обновление драйвера.

<b>Версия драйвера</b>	<b>Дата</b>	<b>Изменения</b>
1.0	23.03.2015	Драйвер создан

### 14. Обновление руководства.

<b>Версия документа</b>	<b>Дата</b>	<b>Изменения</b>
1.0	23.03.2015	Документ создан