Руководство (v1.0)

По работе с драйвером модулей «PCIe – 1553UDx» «mPCIe – 1553UD1» «mPCIe – 1553UD2» «XMC – 1553UDx» «CPCIS – 1553UDx»

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

Для драйверов версии 1.0

ОС QNX 6.5

ООО «Новомар» 2018 г.

Оглавление

1. Введение.

Данный драйвер поддерживает следующие модули:

- **«mPCIe-1553UD1» (1 канал передачи данных);**
- **«mPCIe-1553UD2» (2 канала передачи данных);**
- **«PCIe-1553UD1» (1 канал);**
- **«PCIe-1553UD2» (2 канала);**
- **«PCIe-1553UD4» (4 канала);**
- **«XMC-1553UD2» (2 канала);**
- **«XMC-1553UD4» (4 канала);**
- **«CPCIS-1553UD2» (2 канала);**
- **«CPCIS-1553UD4» (4 канала).**

Для того, чтобы узнать тип модуля установленного в компьютер, воспользуйтесь функцией DEVCTL_VERSION.

Данный драйвер поддерживает одновременную работу не более 100 модулей и присваивает им уникальные символьные имена вида “PCIe_1553UDx_Y”*, где Y — индекс устройства, начиная с 0.

*Для модулей **«XMC–1553UDx»** символьное имя также будет **PCIe_1553UDx_Y.**

Каждый модуль представляет отдельный файл устройства в файловой системе и отображается в каталоге “/dev”.

Взаимодействие с драйвером происходит посредством devctl-команд, перечень которых находится в файле “devctl_man.h”.

Существует четыре режима работы модуля: 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 Mil_1553UDx_drv, содержащий файлы drvmanud, version.txt и devctl_man.h.
3. Откройте файл /etc/rc.d/rc.local текстовым редактором.
4. Первая строка данного файла должна содержать **#!/bin/sh**
5. Далее, добавьте строку **/modules/Mil_1553UDx_drv/drvmanud > /modules/ Mil_1553UDx_drv/drvmanud.log &**
6. Сохраните файл.
7. Перезагрузите компьютер.
8. Посмотреть вывод информации при старте драйвера можно в файле /modules/Mil_1553UDx_drv/drvmanud.log

Замечание:

1. по умолчанию полная отладочная печать драйвера отключена
2. для включения отладочной печати вместо строки **/modules/Mil_1553UDx_drv/drvmanud > /modules/Mil_1553UDx_drv/drvmanud.log &** введите **/modules/Mil_1553UDx_drv/drvmanud -v > /modules/Mil_1553UDx_drv/drvmanud.log &**

9. При выборе стандартного загрузчика(qnxbasesm.ifs) на компьютерах, где поддерживается APIC контроллер (семейство процессоров INTEL), работа с MSI прерываниями модуля будет запрещена самой ОС QNX 6.5. Для того, чтобы ОС позволила драйверу проинициализировать APIC контроллер и настроить MSI прерывания необходимо выбрать загрузчик qnxbasesmp-apic.ifs(см. замечание ниже).

10. Вход в ОС QNX 6.5 обязательно должен быть осуществлен как root.

Замечание:

1. Для того чтобы ОС QNX 6.5 позволила настроить APIC контроллер драйверу, необходимо пересобрать загрузчик следующим образом:

- использовать **startup-apic** вместо **startup-bios**
- использовать **pci-bios=pci-bios-v2** вместо **pci-bios**

2. Пример загрузчика (файл qnxbasesmp-apic.ifs – необходимо скопировать его в директорию /.boot), а так же build файл (файл qnxbasesmp-apic.build) под QNX 6.5 прилагаются

3. При загрузке драйвера выводится сообщение о том, что включены ли MSI прерывания, либо нет. При работе с отключенными MSI прерываниями не рекомендуется в принципе использовать прерывания при работе с модулем.

Если Вы хотите проверить, загружен ли драйвер в системе:

1. Перейдите в каталог /dev.
2. Найдите в выведенном списке **&PCIe_1553UDx_Y**, где Y – номер устройства.
3. Если есть — все нормально, если нет — драйвер не загружен.

Для обновления версии драйвера:

1. Посмотрите версию драйвера (прикладывается в файле version.txt в папке с драйвером).
2. Сравните с версией текущего драйвера (файл /modules/Mil_1553UDx_drv/version.txt).
3. Выберите более позднюю.
4. Замените файлы devctl_man.h, drvmanud и version.txt в папке /modules/Mil_1553UDx_drv на новые (сохраняя имена!).
5. Перезагрузите компьютер.

Идентификаторы устройства:

- **PCI\VEN_A203&DEV_9472&REV_03 (PCIe-1553UDx, XMC-1553UDx, CPCIS-1553UDx);**
- **PCI\VEN_A203&DEV_9470&REV_03 (mPCIe-1553UD1);**
- **PCI\VEN_A203&DEV_9473&REV_03 (mPCIe-1553UD2).**

3. Подключение файла с командами к разрабатываемому проекту.

1. Скопируйте файл **devctl_man.h** из папки /modules/Mil_1553UDx_drv в папку с проектом.
2. В начале проекта добавьте строку:
   #include “devctl_man.h”
3. Можете начать использовать команды.

Формат описан ниже; примеры использования приложены.

ВНИМАНИЕ

Системная функция **open** открывает модули **«PCIe–1553UDx», «XMC–1553UDx»** , «**CPCIS-1553UDx**», **«mPCIe–1553UD1»** и **«mPCIe–1553UD2»** целиком, вне зависимости от модели модуля и количества существующих каналов на нём.

Если вызов драйвера не требует номер канала, значит, драйвер работает с модулем как с единым целым устройством.

Если вызов запрашивает номер канала, значит, драйвер работает только с одним конкретным каналом связи.

4. Список доступных команд по версиям драйверов.

Название вызова	Краткое описание
Список вызовов, доступных **в драйвере версии 1.0**	
DEVCTL_CLEAR_DMA	Сбросить указатель DMA
DEVCTL_DISABLE_DMA	Запретить работу DMA
DEVCTL_DISABLE_INTERRUPT	Отключение прерываний
DEVCTL_DISABLE_SUBADDR	Запрет подадреса
DEVCTL_ENABLE_DMA	Разрешить работу DMA
DEVCTL_ENABLE_INTERRUPT	Включение прерываний и сигналов
DEVCTL_ENABLE_SUBADDR	Разрешение подадреса
DEVCTL_GET_NBLOCK_RAW_DMA	Чтение кол-ва новых данных в блоке DMA
DEVCTL_READ_BC_CONTR_REG	Чтение инструкции, операции или данных
DEVCTL_RD_BLOCK_BUF_PA	Чтение данных из буфера подадреса
DEVCTL_RD_RAW_DMA	Чтение данных из буфера DMA
DEVCTL_RD_REG_OFFSET	Чтение инструкции, операции или данных
DEVCTL_SET_ADDRESS	Установка адреса ОУ
DEVCTL_SET_BUS	Выбор шины передачи данных
DEVCTL_SET_TIMER_BC_CONF_REG_PCI	Установка таймера КШ
DEVCTL_SET_TIMER_RTBM_CONF_REG_PCI	Установка таймера ОУ
DEVCTL_SET_TR_AUTO_MODE	Установка автоматического режима передачи данных
DEVCTL_SET_TR_BUF_READY	Установка бита готовности буфера подадреса к передаче данных
DEVCTL_SET_TR_PROG_MODE	Установка программного режима передачи данных
DEVCTL_SWITCH_MODE	Установка режима работы модуля
DEVCTL_SUBADDRESS_IRQ_RECEIVE	Прерывания по подадресу в случае приёма данных
DEVCTL_SUBADDRESS_IRQ_TRANSMIT	Прерывания по подадресу в случае передачи данных
DEVCTL_VERSION	Информация о плате
DEVCTL_WORK_ENABLE	Разрешение работы модуля
DEVCTL_WRITE_BC_CONTR_REG	Запись инструкций, операций или данных
DEVCTL_WR_BLOCK_BUF_PA	Запись данных в буфер подадреса
DEVCTL_WR_REG_OFFSET	Запись регистра

5. Описание использования команд драйвера.

Структура devctl-команды:

int devctl(int fd, DEVCTL_..., unsigned long param),
где:
int fd – дескриптор файла, полученный при вызове функции open для файла устройства;
DEVCTL_... – команда из набора, описанного в файле devctl_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. –EACCES - запрос не был выполнен драйвером (команда отсутствует, не может быть выполнена в данном режиме модуля, некорректные параметры для данного модуля/команды).

Примеры вызовов всех функций также есть в прикреплённом файле
“**QNX6_5_xPCIe_MIL1553_UDx_test_project.c**”.

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

6. Стандартные функции

6.1. DEVCTL_VERSION

Назначение:
Чтение информации о модуле.

Действие:
Функция заполняет все поля структуры VERSION.
Поле **ver.type** одно из самых важных. Исходя из его значения можно узнать тип устройства и количество каналов на нём:

1 – «PCIe-1553UD», «XMC-1553UD», «mPCIe-1553UD1» или «CPCIS-1553UD»

2 – «PCIe-1553UD2», «XMC-1553UD2», «mPCIe-1553UD2» или «CPCIS-1553UD2»

4 – «PCIe-1553UD4», «XMC-1553UD4» или «CPCIS-1553UD4»

Примечание:
-

Входные данные – структура **VERSION** (поля структуры заполнять не нужно, их заполнит драйвер):

Поле структуры	Описание
unsigned int **device_id**	идентификатор модуля
unsigned int **vendor_id**	идентификатор производителя
char **revision**	номер ревизии модуля
unsigned int type	**тип модуля и количество каналов MIL1553**
char **dev_name[30]**	имя модуля, назначенное драйвером
int **minor**	порядковый номер модуля, присвоенный драйвером
char **irq**	номер линии прерываний
long **size_dma**	размер буфера DMA
void***addr_dma_virt**	адрес буфера DMA
long **pci_bars**	номер регистровго пространства (BAR)

Расшифровка кодов ошибок

-

Пример вызова:

```
if(devctl(fd, DEVCTL_VERSION, &ver, sizeof(ver), NULL)){
  printf("DEVCTL_VERSION error.\n");
}
else{
  printf("Number of channels = %d,\nrevision = 0x%x, \nDMA size =
0x%x. \n", ver.type, ver.revision, ver.size_dma);
}
```

6.2. DEVCTL_VERSION_DRIVER

Назначение:

Чтение информации о версии драйвера и дате его создания(корректировка).

Действие:

Функция возвращает параметр следующего формата:

#define DRIVER_VER_AND_DATE 0x12181117

Примечание:

-

Поле структуры	Описание
31:28	Версия драйвера(целая часть)
27:24	Версия драйвера (дробная часть)
23:16	День создания драйвера
15:8	Месяц создания драйвера
7:0	Год создания драйвера

Расшифровка кодов ошибок

-

Пример вызова:

```
if(devctl(fd, DEVCTL_VERSION_DRIVER, &driver_ver,
sizeof(driver_ver), NULL)){
  printf("DEVCTL_VERSION_DRIVER error.\n");
}
else{
printf("Ver = %x.%x, Date = %x.%x.%x\n",(driver_ver&0xF0000000) >>
28,(driver_ver&0x0F000000) >>
24,(driver_ver&0xFF0000)>>16,(driver_ver&0xFF00)>>8,driver_ver&0
xFF);}
```

6.3. DEVCTL_PCI_LOCATION

Назначение:
Определение на каком слоте и шине находится устройство на шине PCI.

Действие:
Функция возвращает номер слота и номер шины на шине PCI.

Примечание:
-

Входные данные – структура **PCI_LOCATION**:

Поле структуры	Описание	Диапозон значений
unsigned int **pSlot**	Номер слота	
unsigned int **pBus**	Номер шины	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Некорректное значение адреса регистра

Пример вызова:

```
PCI_LOCATION loc;
if (devctl(fd, DEVCTL_WR_REG_OFFSET, & loc, sizeof(loc), NULL)){
    printf("Error read pci location.\n");
}
else{
    printf("pci_slot = 0x%x, pci_bus = 0x%x.\n", loc.pSlot, loc.pBus);
}
```

6.4. DEVCTL_WR_REG_OFFSET

Назначение:
Запись данных в регистровое пространство устройства (BAR).

Действие:
Функция записывает данные по желаемому адресу в регистровое пространство устройства (BAR).

Примечание:
-

Входные данные – структура **SADDR_DATA**:

Поле структуры	Описание	Диапозон значений
unsigned long **daddr**	Адрес начала блока памяти для записи	0h-47FFCh (для модулей с 4 каналами)
unsigned int **data**	Переменная с записываемыми данными	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Некорректное значение адреса регистра

Пример вызова:

```
SADDR_DATA add1;
add1.daddr = 0x2184;
add1.data = 0x80000000;
if (devctl(fd, DEVCTL_WR_REG_OFFSET, &add1, sizeof(add1),
NULL)){
    printf("Error writing to 0x%x reg.\n", add1.daddr);
}
else{
    printf("0x%x was written to 0x%x reg.\n", add1.data, add1.daddr);
}
```

6.5. DEVCTL_RD_REG_OFFSET

Назначение:
Чтение данных из регистрового пространства устройства (BAR).

Действие:
Функция читает данные из желаемого адреса регистрового пространства устройства (BAR) в переменную **data**.

Примечание:
-

Входные данные - структура **SADDR_DATA**:

Поле структуры	Описание	Диапозон значений
unsigned long **daddr**	Адрес начала блока памяти для записи	0h-47FFCh (для модулей с 4 каналами)
unsigned int **data**	Область памяти, куда будут прочитаны данные	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Некорректное значение адреса регистра

Пример вызова:

```
SADDR_DATA add1;
add1.daddr = 0x2184;
if (devctl(fd, DEVCTL_RD_REG_OFFSET, &add1, sizeof(add1),
NULL)){
    printf("Error reading from 0x%x reg.\n", add1.daddr);
}
else{
    printf("0x%x was read from 0x%x reg.\n", add1.data, add1.daddr);
}
```

7. Функции конфигурирования устройства

7.1. DEVCTL_ENABLE_DMA

Назначение:

Разрешение работы DMA для всех каналов. Работа DMA разрешается на весь модуль, а не на конкретный канал!

Действие:

Нулевой бит регистра DMA_DATA_BASE* (адрес 1000h) устанавливается в единицу.
*См. Раздел 5.1.1 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

После загрузки операционной системы работа не разрешена.

Входные данные

-

Расшифровка кодов ошибок

-

Пример вызова:

```
if(devctl(fd, DEVCTL_ENABLE_DMA, 0, 0, NULL)){
        printf("DEVCTL_ENABLE_DMA error.\n");
}
else{
        printf("DMA has been enabled.\n");
}
```

7.2. DEVCTL_DISABLE_DMA

Назначение:

Запрет работы DMA для всех каналов. .Работа DMA запрещается на весь модуль, а не на конкретный канал!

Действие:

Нулевой бит регистра DMA_DATA_BASE* (адрес 1000h) сбрасывается в ноль.
*См. Раздел 5.1.1 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные

-

Расшифровка кодов ошибок

-

Пример вызова:

```
if(devctl(fd, DEVCTL_DISABLE_DMA, 0, 0, NULL)){
        printf("DEVCTL_DISABLE_DMA error.\n");
}
else{
        printf("DMA has been disabled.\n");
}
```

7.3. DEVCTL_ENABLE_SUBADDR

Назначение:
Разрешение подадреса.

Действие:
Разрешается работа подадреса, в зависимости от выбранного канала, а также в зависимости от выбранного режима работы и номера подадреса.

Если выбран режим передачи, то функция устанавливает в единицу 31 бит регистра RT_RCV_REG$n^{(*)}$** и устанавливает в единицу 30 бит регистра RT_TR_REG$n^{(*)}$***.

Если выбран режим приёма, то функция сбрасывает в ноль 31 бит регистра RT_RCV_REG$n^{(*)}$** и сбрасывает в ноль 30 бит регистра RT_TR_REG$n^{(*)}$***.

*n – номер запрашиваемого подадреса
** см. Раздел 6.1.12 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*
*** см. Раздел 6.1.13 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные - структура **SADDR_DATA_PA**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned short **nRT**	Выбор режима работы.	TRANSMIT RECEIVE
unsigned long **paddr**	Номер подадреса	1-30

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра nRT не попадает в диапазон разрешённых значений; - Значение параметра paddr не попадает в диапазон разрешённых значений

Пример вызова:

```
SADDR_DATA_PA sadpa;
sadpa.nRT = RECEIVE;
 sadpa.paddr = 1;
sadpa.nCh = nCh;
if(devctl(fd, DEVCTL_ENABLE_SUBADDR, &sadpa, sizeof(sadpa), NULL)){
        printf("DEVCTL_ENABLE_SUBADDR error.\n");
}
else{
        printf("SubAddress number 1 was enable for receive.\n");
}
```

7.4. DEVCTL_DISABLE_SUBADDR

Назначение:
Запрет подадреса.

Действие:
В зависимости от выбранного канала, а также в зависимости номера подадреса функция устанавливает в единицу 31 бит регистра RT_RCV_REG$n^{(*)}$** и сбрасывает в ноль 30 бит регистра RT_TR_REG$n^{(*)}$***.
*n – номер запрашиваемого подадреса
** см. Раздел 6.1.12 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*
*** см. Раздел 6.1.13 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные - структура **SADDR_DATA_PA**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned short **nRT**	Не используется	
unsigned long **paddr**	Номер подадреса	1-30

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра paddr не попадает в диапазон разрешённых значений

Пример вызова:

```
SADDR_DATA_PA sadpa;
sadpa.paddr = 1;
sadpa.nCh = nCh;

if(devctl(fd, DEVCTL_DISABLE_SUBADDR, &sadpa, sizeof(sadpa),
NULL)){
        printf("DEVCTL_DISABLE_SUBADDR error.\n");
}
else{
        printf("SubAddress number 1 was DISABLE.\n");
}
```

7.5. DEVCTL_SWITCH_MODE

Назначение:
Выбор режима работы модуля канала nCh.

Действие:
Функция записывает значение из переменной **mode** в регистр CTRL_REG_PCI* (адреса 2000h-4000h-6000h-8000h) начиная с нулевого бита.
*См. Раздел 5.1.5 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные – структура MODE_STR:

Поле структуры	Описание	Диапозон значений
int **nCh**	Номер канала, размер буфера которого Вы хотите узнать	1-4
short **mode**	Выбранный режим работы устройства.	"**BM_ADDR_MODE**" - адресуемый монитор шины "**BC_MODE**" - контроллер шины "**RT_MODE**" - оконечное устройство "**BM_MODE**" - не адресуемый монитор шины

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра mode не попадает в диапазон разрешённых значений

Пример вызова:

```
MODE_STR mode;
mode.mode = BM_ADDR_MODE;;
mode.nCh = 1;

if(devctl(fd,DEVCTL_SWITCH_MODE, &mode, sizeof(mode), NULL)){
        printf("Error setting work mode.\n");
}
else{
        printf("Work mode was successfully set.\n");
}
```

7.6. DEVCTL_WORK_ENABLE

Назначение:
Разрешение/запрет работы модуля канала nCh.

Действие:
Если значение переменной **nFlag** равно **ON**, то функция устанавливает в единицу 23 бит регистра CTRL_REG_PCI* (адреса 2000h-4000h-6000h-8000h).
Если значение переменной **nFlag** равно **OFF**, то функция сбрасывает в ноль 23 бит регистра CTRL_REG_PCI*.
*См. Раздел 5.1.5 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные – структура **WORK_ENABLE**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала, работу которого требуется разрешить	1-4
short **nFlag**	Значение TRUE означает разрешение работы, FALSE запрет.	**ON** **OFF**

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4);

Пример вызова:

```
WORK_ENABLE WorkEn;
WorkEn.nFlag = ON;
 WorkEn.nCh = 1;

if(devctl(fd, DEVCTL_WORK_ENABLE, &WorkEn, sizeof(WorkEn),
NULL)){
    printf("Error work enabling.\n");
}
else{
    printf("work was successfully enabled.\n");
}
```

7.7. DEVCTL_SET_BUS

Назначение:
Выбор шины канала nCh.

Действие:
Данный вызов записывает значение переменной **nBus** в регистр CTRL_REG_PCI* (адреса 2000h-4000h-6000h-8000h) начиная со второго бита.
*См. Раздел 5.1.5 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
Если значение равно **NONE**, то обе шины устройства отключены.
Если значение равно **BUSA**, то разрешена работа с шиной "A".
Если значение равно **BUSB**, то разрешена работа с шиной "B".
Если значение равно **BOTH**, то разрешена работа с обеими шинами.

Входные данные – структура **BUS_SEL**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала, размер буфера которого Вы хотите узнать	1-4
unsigned int **nBus**	См. Разедел примечание.	**NONE** **BUSA** **BUSB** **BOTH**

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение переменной nMode не попадает в диапазон разрешённых значений

Пример вызова:

```
BUS_SEL BusSel;
BusSel.nBus = BOTH;
BusSel.nCh = 1;

if(devctl(fd, DEVCTL_SET_BUS, &BusSel, sizeof(BusSel), NULL)){
        printf("Error setting bus number.\n");
}
else{
        printf("Bus number was successfully set.\n");
}
```

7.8. DEVCTL_SET_ADDRESS

Назначение:

Выбор адреса удалённого устройства канала nCh.

Действие:

Данный вызов записывает значение переменной **ad** в регистр CTRL_REG_PCI* (адреса 2000h-4000h-6000h-8000h) начиная с восьмого бита.
Также при необходимости устанавливает в единицу 13 бит этого же регистра.
*см. Раздел 5.1.5 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные – структура **AD_STR**:

Поле структуры	Описание	Диапозон значений
int **nCh**	Номер канала, размер буфера которого Вы хотите узнать	1-4
short **ad**	Желаемый номер адреса оконечного устройства.	1-30

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение переменной ad не попадает в диапазон разрешённых значений

Пример вызова:

```
AD_STR sad;
sad.ad = addr;
sad.nCh=nCh;
if(devctl(fd, DEVCTL_SET_ADDRESS, &sad, sizeof(sad), NULL)){
        printf("DEVCTL_SET_ADDRESS error.\n");
}
else{
        printf("Address %d was assigned to the device.\n", sad.ad);
}
```

7.9. DEVCTL_SET_TIMER_RTBM_CONF_REG_PCI

Назначение:
Установка значений таймеров канала **nCh** для всех режимов кроме режима КШ.

Действие:
Данный вызов записывает значения из переменной **timeout_rcv** в биты с 0 по 1 регистра RTBM_CONF_REG_PCI* (адреса 2004h-4004h-6004h-8004h);
из переменной **timeout_tr** в биты с 8 по 10 этого же регистра;
из переменной **timer** в биты 16-18 этого же регистра.
*См. Раздел 6.1.1 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные – структура **TT**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала, размер буфера которого Вы хотите узнать	1-4
short **timeout_rcv**	Таймаут приёма	17, 60, 85, 110 мкс
short **timeout_tr**	Таймаут передачи	6, 8, 11, 13, 18, 61, 86, 111 мкс
short **timer**	Значение инкремента основного таймера	1, 2, 4, 8, 16, 32, 64 мкс

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение переменной **timeout_rcv** не попадает в диапазон разрешённых значений; - Значение переменной **timeout_tr** не попадает в диапазон разрешённых значений; - Значение переменной **timer** не попадает в диапазон разрешённых значений.

Пример вызова:

```
TT tt;
tt.nCh=nCh;
tt.timeout_rcv=17;
tt.timeout_tr=6;
tt.timer=2;

if(devctl(fd, DEVCTL_SET_TIMER_RTBM_CONF_REG_PCI, &tt,
sizeof(tt), NULL)){
printf("DEVCTL_SET_TIMER_RTBM_CONF_REG_PCI error.\n");
}else{ printf("Timeout values has been set.\n");}
```

7.10. DEVCTL_SET_TIMER_BC_CONF_REG_PCI

Назначение:
Установка значений таймеров канала **nCh** для режима КШ.

Действие:
Данный вызов записывает значения из переменной **timeout_rcv** в биты с 19 по 20 регистра BC_CONF_REG_PCI* (адреса 2008h-4008h-6008h-8008h);
из переменной **timeout_tr** в биты с 21 по 23 этого же регистра;
из переменной **timer** в биты 24-26 этого же регистра.
*См. Раздел 7.2.1 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные – структура **TT**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала, размер буфера которого Вы хотите узнать	1-4
short **timeout_rcv**	Таймаут приёма	17, 60, 85, 110 мкс
short **timeout_tr**	Таймаут передачи	6, 8, 11, 13, 18, 61, 86, 111 мкс
short **timer**	Значение инкремента основного таймера	1, 2, 4, 8, 16, 32, 64 мкс

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение переменной **timeout_rcv** не попадает в диапазон разрешённых значений; - Значение переменной **timeout_tr** не попадает в диапазон разрешённых значений; - Значение переменной **timer** не попадает в диапазон разрешённых значений.

Пример вызова:

```
TT tt;
tt.nCh=nCh;
tt.timeout_rcv=17;
tt.timeout_tr=6;
tt.timer=2;

if(devctl(fd, DEVCTL_SET_TIMER_BC_CONF_REG_PCI , &tt,
sizeof(tt), NULL)){
printf("DEVCTL_SET_TIMER_BC_CONF_REG_PCI  error.\n");
}else{  printf("Timeout values for BC mode has been set.\n");}
```

8. Функции для чтения данных

8.1. DEVCTL_GET_NBLOCK_RAW_DMA

Назначение:
Чтение количества новых блоков данных, накопленных в буфере DMA устройства по каналу nCh (блок - 128 байт).

Действие:
Функция вычитает из значения регистра DMA_INDEX*(адреса 0x1040; 0x1044; 0x1048; 0x104C) значение программного счётчика вычитанных пакетов. Полученное количество блоков возвращается в переменной **nBlock**.
*См. Раздел 5.1.2 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные структура **GET_NBLOCK**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned int **nBlock**	Переменная в которую будут считано количество новых блоков данных	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4)

Пример вызова:

```
GET_NBLOCK GetNBlock;
GetNBlock.nCh=1;
if(devctl(fd, DEVCTL_GET_NBLOCK_RAW_DMA, &GetNBlock,
sizeof(GetNBlock), NULL)){
        printf("Error getting NBLOCK RAW DMA.\n");
}
else{
        printf("There is %d of new data block(s).\n", GetNBlock.nBlock);
}
```

8.2. DEVCTL_RD_RAW_DMA

Назначение:
Чтение данных из буфера DMA канала nCh. Данная функция читает любое требуемое количество новых блоков данных DMA канала nCh.

Действие:
Чтобы воспользоваться данным вызовом, рекомендуется узнать количество новых блоков данных с помощью вызова DEVCTL_GET_NBLOCK_RAW_DMA. Значение возвращенное этой функцией, использовать в качестве входного параметра для переменной nBlocks.
После выполнения функции указатель буфера DMA сдвигается на соответствующую количеству прочитанных данных позицию.

Примечание:
Изначально в переменную **number_block** нужно указать какое количество блоков данных вы хотите прочитать. После успешного выполнения данного вызова, драйвер запишет в эту переменную количество прочитанных блоков DMA. Количество прочитанных блоков DMA может отличаться от количества запрашиваемых. Например если было запрошено 10 блоков данных, а в DMA всего 3 новых блоков данных, то данный вызов прочитает 3 блока данных и изменит значение переменной **number_block** с 10 на 3. Так же значение количества блоков может измениться если данные находятся в конце и начале буфера DMA. Например вызов DEVCTL_GET_NBLOCK_RAW_DMA определил, что в буфере DMA 10 новых блоков данных, но 2 из них находятся в конце буфера, а 8 записались в начало. В этом случае DEVCTL_RD_RAW_DMA следует вызывать 2 раза. При первом вызове прочитаются только 2 блока данных и программный счётчик переместится в начало буфера. При втором вызове прочитаются оставшиеся 8 блоков данных.

Входные данные – структура пользователя (см. пример):

Поле структуры	Описание	Диапозон значений
int **nCh**	Номер канала	1-4
int **b[n]**	Буфер в который будут считаны данные, где n – размер буфера	
unsigned int **number_block**	Запрашиваемое количество блоков данных для чтения. После чтения данных в данную переменную будет записано значение количества прочитанных блоков данных.	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4)

Пример вызова на следующей странице:

```
struct {
        unsigned int number_block;
        int nCh;
        int b[32];
} d;

int i=0;
d.number_block=1;
d.nCh=1;

if(devctl(fd, DEVCTL_RD_RAW_DMA, &d, sizeof(d), NULL)){
        printf("Error reading DMA.\n");
}
else{
        printf("Was read %d data blocks from DMA\n", d.number_block);
}

printf("Data:\n");
for (i; i<32; i++){
        printf("0x%x\n", d.b[i]);
}
```

8.3. DEVCTL_CLEAR_DMA

Назначение:

По вызову данной функции программный счётчик вычитанных пакетов данных DMA уравнивается с регистром DMA_INDEX*(адреса 0x1040; 0x1044; 0x1048; 0x104C).
*См. Раздел 5.1.2 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Действие:

-

Примечание:

ВНИМАНИЕ!!!

ФУНКЦИЮ ИСПОЛЬЗОВАТЬ ОЧЕНЬ АККУРАТНО! ВОЗМОЖНА ПОТЕРЯ ДАННЫХ. ПЕРЕД ВЫЗОВОМ ПРОВЕРЬТЕ ВЫЧИТАНЫ ЛИ ВСЕ ДАННЫЕ.

Входные данные – структура **CH_NUM**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала, размер буфера которого Вы хотите узнать	0-4 0 – очистить DMA по всем каналам

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4)

Пример вызова:

```
CH_NUM nCh;
nCh.nCh=1;
if(devctl(fd, DEVCTL_CLEAR_DMA, &nCh, sizeof(nCh), NULL)){
        printf("Error clearing DMA.\n");
}
else{
        printf("DMA index was successfully cleared.\n");
}
```

9. Функции MIL1553 режима ОУ

9.1. DEVCTL_WR_BLOCK_BUF_PA

Назначение:

Функция записывает данные из буфера **buf** в буфер подадреса канала nCh.

Действие:

Функция записывает 64 байта данных из переменной **buf** в буфер подадреса RT_DATA_BUF0n* (адреса 2800h, 4800h, 6800h, 8800h) или RT_DATA_BUF1n** (адреса 3000h, 5000h, 7000h, 9000h).

*См. Раздел 6.1.16 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

*См. Раздел 6.1.17 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные структура **SBLOCK_BUF_PA**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned long **num_buf**	Номер буфера	0 - RT_DATA_BUF0n 1 -RT_DATA_BUF1n
unsigned long **paddr**	Номер подадреса	1-30
unsigned short **buf[32]**	Область памяти размером 64 байта	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **num_buf** не входит в диапазон допустимых значений; - Значение параметра **paddr** не входит в диапазон допустимых значений.

Пример вызова:

```
SBLOCK_BUF_PA sbbpa;
sbbpa.num_buf = 0;
sbbpa.paddr = 1;
sbbpa.nCh=1;
for(i=0; i<31; i++)
   sbbpa.buf[i]=0x1234;
if(devctl(fd, DEVCTL_WR_BLOCK_BUF_PA, &sbbpa, sizeof(sbbpa),
NULL)){    printf("DEVCTL_WR_BLOCK_BUF_PA error.\n");
}
else{    printf("sbbpa.buf was written to %d buf, PAddress %d.\n",
sbbpa.num_buf, sbbpa.paddr);  }
```

9.2. DEVCTL_RD_BLOCK_BUF_PA

Назначение:
Функция читает данные из буфера подадреса канала nCh в буфер **buf**.

Действие:
Функция читает 64 байта данных из буфера подадреса RT_DATA_BUF0n* (адреса 2800h, 4800h, 6800h, 8800h) или RT_DATA_BUF1n** (адреса 3000h, 5000h, 7000h, 9000h) в буфер pData.
*См. Раздел 6.1.16 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*
*См. Раздел 6.1.17 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:
-

Входные данные - структура **SBLOCK_BUF_PA**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned long **num_buf**	Номер буфера	0 - RT_DATA_BUF0n 1 -RT_DATA_BUF1n
unsigned long **paddr**	Номер подадреса	1-30
unsigned short **buf[32]**	Область памяти размером 64 байта	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **num_buf** не входит в диапазон допустимых значений; - Значение параметра **paddr** не входит в диапазон допустимых значений.

Пример вызова:

```
SBLOCK_BUF_PA sbbpa;
sbbpa.num_buf = 0;
sbbpa.paddr = 1;
sbbpa.nCh=1;

if(devctl(fd, DEVCTL_RD_BLOCK_BUF_PA, &sbbpa, sizeof(sbbpa),
NULL)){
printf("DEVCTL_RD_BLOCK_BUF_PA error.\n");
}
else{
printf("Data has been read from %d buf, PAddress %d to sbbpa.buf.\n",
sbbpa.num_buf, sbbpa.paddr);
}
for (i; i<32; i++){printf("0x%x\n", sbbpa.buf[i]);}
```

9.3. DEVCTL_SET_TR_PROG_MODE

Назначение:

Включение режима передачи данных “Программный”* канала nCh.

Действие:

Функция читает регистр RT_TR_REGn**. Если 28 и 29 биты данного регистра не равны нулю, то функция сбрасывает их в ноль и устанавливает 31 бит этого же регистра в 1 (без данного шага 28 и 29 бит будут не доступны на запись). Записывает данные в регистр. Все остальные биты данного регистра не изменяются.

Далее 31 бит сбрасывается в 0 и данный регистр записывается ещё раз.

*См. Раздел 6.1.18 документа “*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx , «mPCIe-1553UD1» и «mPCIe-1553UD2».*

**См. Раздел 6.1.13 документа “*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные – структура **SET_TR_MODE**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned int **nNum**	Номер подадреса	1-30

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра nNum не входит в диапазон допустимых значений.

Пример вызова:

```
SET_TR_MODE SetTRMode;
SetTRMode.nCh=1;
SetTRMode.nNum=1;

if(devctl(fd, DEVCTL_SET_TR_PROG_MODE, &SetTRMode,
sizeof(SetTRMode), NULL)){
        printf("DEVCTL_SET_TR_PROG_MODE error.\n");
}
else{
        printf("Transmit mode was successfuly set.\n");
}
```

9.4. DEVCTL_SET_TR_AUTO_MODE

Назначение:

Включение режима передачи данных “Автомат”* канала nCh.

Действие:

Функция читает регистр RT_TR_REGn**.

Если режим передачи данных “Автомат” включён (28 бит данного регистра равен единице, 29 бит равен нулю), то функция завершает свою работу.

Если режим передачи данных “Автомат” выключен (28 бит данного регистра не равен единице, 29 бит не равен нулю), то функция устанавливает 28 бит в единицу, 29 бит сбрасывает в ноль.

Затем 31 бит этого регистра устанавливается в единицу (без данного шага 28 и 29 бит будут не доступны на запись) и данные записываются в регистр. Все остальные биты данного регистра не изменяются.

Далее 31 бит сбрасывается в 0 и данный регистр записывается ещё раз.

*См. Раздел 6.1.18 документа “*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

**См. Раздел 6.1.13 документа “*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные – структура **SET_TR_MODE**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned int **nNum**	Номер подадреса.	1-30

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **nNum** не входит в диапазон допустимых значений.

Пример вызова:

```
SET_TR_MODE SetTRMode;
SetTRMode.nCh=1;
SetTRMode.nNum=1;

if(devctl(fd, DEVCTL_SET_TR_AUTO_MODE, &SetTRMode,
sizeof(SetTRMode), NULL)){
        printf("DEVCTL_SET_TR_PROG_MODE error.\n");
}
else{
        printf("Transmit mode \"Auto\" was successfully set.\n");
}
```

9.5. DEVCTL_SET_TR_BUF_READY

Назначение:

Установка бита готовности буфера для передачи данных в режиме ОУ канала nCh. В режиме передачи данных “Программный” данные биты устанавливаются и сбрасываются вручную, а в режиме передачи данных “Автомат”, биты могут быть сброшены автоматически.*

*См. Раздел 6.1.18 документа “*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Действие:

Функция читает регистр RT_TR_REGn** и изменяет 22 и 23 биты (см. пункт примечание).

Затем 31 бит этого же регистра устанавливается в 1 (без данного шага 22 и 23 биты будут не доступны на запись). Записывает данные в регистр. Все остальные биты данного регистра не изменяются.

Далее 31 бит сбрасывается в 0 и данный регистр записывается ещё раз.

**См. Раздел 6.1.13 документа “*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

NONE = 0: 22 и 23 бит сбрасываются в ноль (оба буфера не готовы к передаче);
BUF0 = 1: 22 бит устанавливается в 1, а 23 сбрасывается в ноль(к передаче готов только буфер 0);
BUF1 = 2: 22 бит сбрасывается в ноль, а 23 устанавливается в 1(к передаче готов только буфер 1);
BOTH =3: 22 и 23 бит устанавливаются в единицу (оба буфера готовы к передаче).

Входные данные

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned int **nNum**	Номер подадреса.	1-30
unsigned int **nBuf**	Режим готовности буфера	См. примечание

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **nNum** не входит в диапазон допустимых значений; - Значение параметра **nBuf** не входит в диапазон допустимых значений.

Пример вызова:

```
SET_TR_MODE_BUF SetTRModeBuf;
SetTRModeBuf.nCh=1;
SetTRModeBuf.nNum=1;
SetTRModeBuf.nBuf= BUF0;

if(devctl(fd, DEVCTL_SET_TR_BUF_READY, &SetTRModeBuf,
sizeof(SetTRModeBuf), NULL)){
        printf("DEVCTL_SET_TR_BUF_READY error.\n");
}
else{   printf("Buffer was successfully set to ready state.\n");  }
```

10. Функции MIL1553 режима КШ

10.1. DEVCTL_READ_BC_CONTR_REG

Назначение:
Чтение информации в область памяти модуля канала nCh, предназначенную для режима КШ.

Действие:
- Если значение переменной **type** равно **INSTR**, функция читает информацию из регистра **BC_INSTR_RAM*** (адреса 0x18000, 0x28000, 0x38000, 0x48000) плюс смещение **offset** в переменную **data**;
- Если значение переменной **type** равно **OPER**, функция читает информацию из регистра **BC_OPERATION_RAM**** (адреса 0x1C000, 0x2C000h, 0x3C000, 0x4C000) плюс смещение **offset** в переменную **data**;
- Если значение переменной **type** равно **DATA**, функция читает информацию из регистра **BC_DATA_RAM***** (адреса 0x10000, 0x20000, 0x30000, 0x40000) плюс смещение **offset** в переменную **data**.

*См. Раздел 7.1.1 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx , «mPCIe-1553UD1» и «mPCIe-1553UD2».*
**См. Раздел 7.1.2 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*
***См. Раздел 7.1.3 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx , «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание: -

Входные данные – структура **BC_RAM_STR**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned long **type**	Номер подадреса.	DATA, INSTR, OPER
unsigned long **offset**	Смещение относительно начально адреса	
unsigned long **data**	Информация для записи	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **type** не входит в диапазон допустимых значений.

Пример вызова:

```
BC_RAM_STR bc;
bc.nCh = 1;
bc.type = INSTR;
bc.offset = 2;
if(devctl(fd, DEVCTL_READ_BC_CONTR_REG, &bc, sizeof(bc),
NULL)){       printf("DEVCTL_READ_BC_CONTR_REG error.\n");
}else{  printf("the 3-rd word of instructions = 0x%x.\n", bc.data);   }
```

10.2. DEVCTL_WRITE_BC_CONTR_REG

Назначение:
Запись информации в область памяти модуля канала nCh, предназначенную для режима КШ.

Действие:
- Если значение переменной **type** равно **INSTR**, функция записывает информацию из переменной **data** в регистр **BC_INSTR_RAM*** (адреса 0x18000, 0x28000, 0x38000, 0x48000) плюс смещение **offset**;
- Если значение переменной **type** равно **OPER**, функция записывает информацию из переменной **data** в регистр **BC_OPERATION_RAM**** (адреса 0x1C000, 0x2C000h, 0x3C000, 0x4C000) плюс смещение **offset**;
- Если значение переменной **type** равно **DATA**, функция записывает информацию из переменной **data** в регистр **BC_DATA_RAM***** (адреса 0x10000, 0x20000, 0x30000, 0x40000) плюс смещение **offset**.

*См. Раздел 7.1.1 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx , «mPCIe-1553UD1» и «mPCIe-1553UD2».*
**См. Раздел 7.1.2 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*
***См. Раздел 7.1.3 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание: -

Входные данные – структура **BC_RAM_STR**:

Поле структуры	Описание	Диапозон значений
unsigned int **nCh**	Номер канала	1-4
unsigned long **type**	Номер подадреса.	DATA, INSTR, OPER
unsigned long **offset**	Смещение относительно начально адреса	
unsigned long **data**	Область памяти для записи прочитанной информации	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **type** не входит в диапазон допустимых значений.

Пример вызова:

```
BC_RAM_STR bc;
bc.nCh = 1;
bc.type = INSTR;
bc.offset = 2;
bc.data = 1;
if(devctl(fd, DEVCTL_WRITE_BC_CONTR_REG, &bc, sizeof(bc),
NULL)){       printf("DEVCTL_WRITE_BC_CONTR_REG error.\n");
} else{ printf("bc.data was written to 3-rd word of instructions.\n");  }
```

11. Функции прерываний

11.1. DEVCTL_ENABLE_INTERRUPT

Назначение:

Разрешение и ожидание сигнала о прерывании **m_nInt** канала nCh.

Действие:

Драйвер сохраняет полученные **proc_pid** и **sig_no**, разрешает прерывание **m_nInt** и ожидает его. Для разрешения прерывания функция записывает данные либо только в регистр INTERRUPT MASK* (адрес 1010h), либо и в регистр INTERRUPT MASK и в регистр RT_INT_REG** (адреса 2034h, 4034h, 6034h, 8034h).

Как только ожидаемое прерывание обрабатывается драйвером, сигнал **sig_no** передаётся процессу с pid **proc_pid**.

* см. Раздел 5.1.4 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

** см. Раздел 6.1.8 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

Варианты **m_nInt**:

- **INT_HDATx -** прерывание по заполнению 1/2 буфера данных контроллера nCh (8192 пакета);
- **INT_QDATx -** прерывание по заполнению 1/16 буфера данных контроллера nCh (512 пакетов);
- **RTx_INT_MC -** прерывание по приёму КУ контроллера nCh;
- **RTx_INT_ERR** - прерывание по ошибке сообщения контроллера nCh;
- **RTx_INT_REN** – прерывание от подадреса приёма контроллера nCh Прерывание не будет срабатывать, если подадрес не разрешён на приём данных и если не разрешено прерывание конкретного подадреса. Для разрешения подадреса воспользуйтесь вызовом DEVCTL_ENABLE_SUBADDR. Для разрешения прерывания подадреса воспользуйтесь вызовом DEVCTL_SUBADDRESS_IRQ_RECEIVE.
- **RTx_INT_TEN** – прерывание от подадреса передачи контроллера nCh. Прерывание не будет срабатывать, если подадрес не разрешён на передачу данных и если не разрешено прерывание конкретного подадреса. Для разрешения подадреса воспользуйтесь вызовом DEVCTL_ENABLE_SUBADDR. Для разрешения прерывания подадреса воспользуйтесь вызовом DEVCTL_SUBADDRESS_IRQ_TRANSMIT.
- **INT_BCx** - Прерывание контроллера nCh режима КШ.

Входные данные - структура **SINTHIOSA**:

Поле структуры	Описание	Диапозон значений
int **m_nCh**	Номер канала	1-4
int **m_nInt**	Идентификатор прерывания	См. п. примечание
int **proc_pid**	pid процесса, на который Вы хотите получить данный сигнал	
int **sig_no**	сигнал, который Вы хотите получить	

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **m_nInt** не входит в диапазон допустимых значений.

Пример вызова:

```
static void sig_handler (int signo){
        printf("-----> signal %d\n", signo);
}

int main( int argc, char *argv[] ){
        int retval;
        int i;
        SADDR_DATA add1;
        unsigned long pid, sig;
        char *nodename = "/dev/PCIe_1553UDx_0";
        SINTHIOSA ss1;

        printf("Test signal \n");
        pid = getpid();
        sig = SIGDEFAULT;
        if (argc > 1) sig = atoi(argv[1]);
        if (argc > 2) pid = atoi(argv[2]);
        if (argc > 3) nodename = argv[3];
        if (SIG_ERR == signal(sig, sig_handler)) printf("set signal handler
error \n");

//opening
        i = open(nodename, O_RDWR);
        printf("Device opened 0 \n");

//set signal
        ss1.m_nCh = 1;
        ss1.m_nInt = INT_HDATx;
        ss1.proc_pid = pid;
        ss1.sig_no = sig;
        retval = devctl(i, DEVCTL_ENABLE_INTERRUPT, &ss1,
sizeof(ss1), NULL);
        if (retval == 0)                printf("signal set \n");
        else            printf("error set signal %d \n", retval);

//close
        close(i);
        printf("Device closed 0 \n");

        while (1) pause();
        exit(0);}
```

11.2. DEVCTL_DISABLE_INTERRUPT

Назначение:

Запрет прерывания **m_nInt** канала **nCh**. (Вызов для разрешения прерывания отсутствует, т.к. прерывание разрешается в вызове DEVCTL_ENABLE_INTERRUPT)

Действие:

Данный вызов запрещает прерывание **m_nInt,** записывая данные либо только в регистр INTERRUPT MASK* (адрес 1010h), либо и в регистр INTERRUPT MASK и регистр RT_INT_REG** (адреса 2034h, 4034h, 6034h, 8034h).

* см. Раздел 5.1.4 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

** см. Раздел 6.1.8 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx , «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

Варианты **m_nInt**:

- **INT_HDATx -** прерывание по заполнению 1/2 буфера данных контроллера nCh;
- **INT_QDATx -** прерывание по заполнению 1/16 буфера данных контроллера nCh;
- **RTx_INT_MC -** прерывание по приёму КУ контроллера nCh;
- **RTx_INT_ERR** - прерывание по ошибке сообщения контроллера nCh;
- **RTx_INT_REN** – прерывание от подадреса приёма контроллера nCh;
- **RTx_INT_TEN** – прерывание от подадреса передачи контроллера nCh;
- **INT_BCx** - Прерывание контроллера nCh режима КШ.

Входные данные - структура **SINTHIOSA**:

Поле структуры	Описание	Диапозон значений
int **m_nCh**	Номер канала	1-4
int **m_nInt**	Идентификатор прерывания	См. п. примечание

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **m_nInt** не входит в диапазон допустимых значений.

Пример вызова:

```
SINTHIOSA ss1;
ss1.m_nCh = 1;
ss1.m_nInt = INT_HDATx;

if(devctl(fd, DEVCTL_DISABLE_INTERRUPT, &ss1, sizeof(ss1),
NULL)){
        printf("DEVCTL_DISABLE_INTERRUPT error.\n");
}
else{   printf("Interrupt was successfully disabled.\n");}
```

11.3. DEVCTL_SUBADDRESS_IRQ_RECEIVE

Назначение:

Запрет или разрешение прерывания по подадресу **nNum** канала **nCh** в случае успешного приёма данных.

Действие:

Если **action** равно **ENABLE**, то 29 бит регистра RT_RCV_REG* (адреса 2100…217Ch, 4100…417Ch, 6100…617Ch, 8100…817Ch) устанавливается в единицу и прерывания разрешаются.
Если **action** равно **DISABLE**, то 29 бит регистра RT_RCV_REG* сбрасывается в ноль и прерывания запрещаются.
* см. Раздел 6.1.12 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные - структура **SADDR_INT**:

Поле структуры	Описание	Диапозон значений
int **m_nCh**	Номер канала	1-4
unsigned int **nNum**	Номер подадреса	1-30
short **action**	Разрешение или запрет прерывания	**ENABLE** **DISABLE**

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **nNum** не входит в диапазон допустимых значений.

Пример вызова:

```
SADDR_INT saddr_int;
saddr_int.nCh=nCh;
saddr_int.nNum=1;
saddr_int.action=DISABLE;

if(devctl(fd, DEVCTL_SUBADDRESS_IRQ_RECEIVE, &saddr_int,
sizeof(saddr_int), NULL)){
        printf("DEVCTL_SUBADDRESS_IRQ_RECEIVE error.\n");
}
else{
        printf("Receive Subaddress was enabled/disable for interrupts.\n");
}
```

11.4. DEVCTL_SUBADDRESS_IRQ_TRANSMIT

Назначение:

Запрет или разрешение прерывания по подадресу **nNum** канала **nCh** в случае успешной передачи данных.

Действие:

Если **action** равно **ENABLE**, то 24 бит регистра RT_TR_REG* (адреса 2180…21FCh, 4180…41FCh , 6180…61FCh , 8180…81FCh) устанавливается в единицу и прерывания разрешаются.
Если **action** равно **DISABLE**, то 24 бит регистра RT_TR_REG * сбрасывается в ноль и прерывания запрещаются.
* см. Раздел 6.1.13 документа "*Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx», «mPCIe-1553UD1» и «mPCIe-1553UD2».*

Примечание:

-

Входные данные - структура **SADDR_INT**:

Поле структуры	Описание	Диапозон значений
int **m_nCh**	Номер канала	1-4
unsigned int **nNum**	Номер подадреса	1-30
short **action**	Разрешение или запрет прерывания	**ENABLE** **DISABLE**

Расшифровка кодов ошибок

код	Расшифровка
-EACCES	Возможные причины ошибки: - nCh задан несуществующий номер канала. Если значение параметра nCh входит в диапазон 1-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4); - Значение параметра **nNum** не входит в диапазон допустимых значений.

Пример вызова:

```
SADDR_INT saddr_int;
saddr_int.nCh=nCh;
saddr_int.nNum=1;
saddr_int.action=DISABLE;

if(devctl(fd, DEVCTL_SUBADDRESS_IRQ_TRANSMIT, &saddr_int,
sizeof(saddr_int), NULL)){
        printf("DEVCTL_SUBADDRESS_IRQ_TRANSMIT error.\n");
}
else{
        printf("Transmit Subaddress was enabled/disable for interrupts.\n");
}
```

12. С чего начать

12.1. Необходимый набор вызовов для инициализации модуля на приём данных в режиме ОУ.

DEVCTL_SWITCH_MODE; Выбрать режим ОУ

DEVCTL_SET_ADDRESS;

DEVCTL_SET_BUS;

DEVCTL_ENABLE_SUBADDR; Второй входной параметр nRT должен иметь значение **RECEIVE** (подадрес разрешается на приём).

DEVCTL_WORK_ENABLE; При инициализации модуля, разрешать работу следует последним действием.

DEVCTL_ENABLE_DMA;

12.2. Набор вызовов для чтения принятых данных

DEVCTL_GET_NBLOCK_RAW_DMA

DEVCTL_RD_RAW_DMA

12.3. Необходимый набор вызовов для инициализации модуля на передачу данных в режиме ОУ.

DEVCTL_SWITCH_MODE; Выбрать режим ОУ

DEVCTL_SET_ADDRESS;

DEVCTL_SET_BUS;

DEVCTL_ENABLE_SUBADDR; Второй входной параметр nRT должен иметь значение **TRANSMIT** (подадрес разрешается на передачу).

DEVCTL_WORK_ENABLE; При инициализации модуля, разрешать работу следует последним действием.

DEVCTL_ENABLE_DMA;

12.4. Набор функций для передачи данных

DEVCTL_WR_BLOCK_BUF_PA

DEVCTL_SET_TR_PROG_MODE или DEVCTL_SET_TR_AUTO_MODE

DEVCTL_SET_TR_BUF_READY

12.5. Набор вызовов для инициализации модуля для работы в режиме КШ

DEVCTL_SWITCH_MODE Выбрать режим КШ
DEVCTL_ENABLE_DMA
DEVCTL_SET_BUS
DEVCTL_WORK_ENABLE

12.6. Набор вызовов для записи инструкций, операций и данных в режиме КШ и начало их выполнения

DEVCTL_WRITE_BC_CONTR_REG

13. Обновление драйвера

Версия драйвера	Дата	Изменение
1.0	18.01.2018	- Драйвер создан.

14. Обновление руководства

Версия руководства	Дата	Изменения
1.0	18.01.2018	- Документ создан