



Руководство (v1.3)

**По работе с драйвером модулей
«PCIe – 1553UDx»
«mPCIe – 1553UD2»
«ХМС – 1553UDx»**

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

Для драйверов версии 1.3

ОС WINDOWS



06.02.2017

ООО «Новомар» 2017 г.

Оглавление

1. Введение	3
2. Установка драйвера	4
2.1 Расшифровка названия драйвера.	5
3. Список доступных IOCTL вызовов по версиям драйверов.....	6
4. IOCTL вызовы.....	7
4.1 IOCTL_MIL1553UDx_CLEAR_DMA.....	8
4.2 IOCTL_MIL1553UDx_DISABLE_INTERRUPT	9
4.3 IOCTL_MIL1553UDx_DISABLE_SUBADDR.....	10
4.4 IOCTL_MIL1553UDx_ENABLE_SUBADDR.....	11
4.5 IOCTL_MIL1553UDx_GET_DEVICE_VER.....	12
4.6 IOCTL_MIL1553UDx_GET_NBLOCK_RAW_DMA	13
4.7 IOCTL_MIL1553UDx_GET_RESOURCE_INFO.....	14
4.8 IOCTL_MIL1553UDx_READ_BAR.....	16
4.9 IOCTL_MIL1553UDx_READ_DMA_CH_DATA_BLOCKS	17
4.10 IOCTL_MIL1553UDx_READ_REG	18
4.11 IOCTL_MIL1553UDx_READ_SUBADDR_BUF.....	19
4.12 IOCTL_MIL1553UDx_REGISTER_INTERRUPT	20
4.13 IOCTL_MIL1553UDx_SUBADDR_IRQ_RECEIVE	22
4.14 IOCTL_MIL1553UDx_SUBADDR_IRQ_TRANSMIT.....	23
4.15 IOCTL_MIL1553UDx_WRITE_BAR	24
4.16 IOCTL_MIL1553UDx_WRITE_REG.....	25
4.17 IOCTL_MIL1553UDx_WRITE_SUBADDR_BUF.....	26
5. Обновление драйвера.....	27
6. Обновление руководства	27

1. Введение

Данный драйвер поддерживает следующие модули:

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

Модуль «mPCIe-1553UD» данным драйвером не поддерживается. Для «mPCIe-1553UD» используется драйвер:

<http://novomar-spb.ru/mpcie-1553ud.html#soft>

Для того, чтобы узнать тип модуля установленного в компьютер, воспользуйтесь вызовом [IOCTL_MIL1553UDx_GET_DEVICE_VER](#).

Основой драйвера для модулей «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2», является файл Mil_1553UDx_YY_VerX.sys, где YY является разрядностью ОС (x86 или x64), для которой разработан драйвер, а X версия драйвера.

Драйвер разрабатывался и тестировался для операционных систем Microsoft Windows XP 32 bit edition, Microsoft Windows 7 32 bit edition и Microsoft Windows 7 64 bit edition.

При установке драйвера на ОС Windows XP, система выдаст предупреждение о том, что данное ПО не прошло сертификацию для Windows XP. Это связано с тем, что компания Microsoft более не поддерживает Windows XP. Нажмите кнопку «Продолжить». Драйвер установится и начнёт свою корректную работу.

Драйвер поддерживает как работу одного, так и одновременную работу нескольких устройств и присваивает им уникальные символьные имена PCIe1553UDxY*, где Y – индекс устройства, начиная с 0. Т.е. при одновременной работе двух и более устройств, одно будет иметь имя PCIe1553UDx0, второе PCIe1553UDx1 и т.д.

*Для модулей «XMC-1553UDx» символьное имя также будет PCIe1553UDxY.

2. Установка драйвера

Установка драйвера производится стандартными средствами установки оборудования ОС Windows.

Для установки драйвера следует открыть «Диспетчер устройств», выбрать устройство с одним из идентификаторов:

- PCI\VEN_A203&DEV_9472&REV_03 (PCIe-1553UD, PCIe-1553UD2, PCIe-1553UD4);
- PCI\VEN_A203&DEV_9473&REV_23 (mPCIe-1553UD2);
- PCI\VEN_A203&DEV_9474&REV_23 (XMC-1553UD2, XMC-1553UD4);

Для того, чтобы узнать тип модуля установленного в компьютер, воспользуйтесь вызовом [IOCTL_MIL1553UDx_GET_DEVICE_VER](#).

и нажать установить драйвер. Идентификатор можно просмотреть в свойствах устройства, во вкладке «Сведения», выбрав пункт «ИД оборудования».

Далее следует выбрать кнопку «Выполнить поиск драйверов на этом компьютере», указать путь к директории драйвера и нажать «Далее».

Если система отобразит ошибку, что не удалось найти драйвер для этого устройства, значит устройство выбрано неверно. Проверьте идентификатор устройства.

Если система отобразит ошибку о том, что устройство не может начать работу (код 10), перезагрузите компьютер.

Если система отобразит ошибку о том, что не удалось проверить цифровую подпись драйвера (код 52), проверьте наличие обновления ОС KB3033929. В Windows 7 наличие обновления можно проверить по следующему пути: «Пуск → Панель управления → Система и безопасность → Просмотр установленных обновлений → Поиск: установленные обновления».

Если система отобразит сообщение, что драйвер установлен, то можно приступать к работе с устройством.

Модуль теперь можно найти в «Диспетчере устройств» в ветке «Multifunction Adapters» под именем «Novomar(R) Mil-1553UDx Controller».

2.1 Расшифровка названия драйвера.

<i>Mil_1553UDx_x86_Ver1_0.sys</i>			
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>

1	<i>Mil_1553UDx</i>	Название драйвера
2	<i>X86</i>	Разрядность ОС, для которой предназначен драйвер
2	<i>Ver1_0</i>	Версия драйвера
3	<i>sys</i>	Расширение файла

3. Список доступных IOCTL вызовов по версиям драйверов

Название вызова	Краткое описание
Список вызовов, доступных в драйвере версии до 1.2 и ниже	
<u>IOCTL_MIL1553UDx_CLEAR_DMA</u>	Очистка буфера DMA
<u>IOCTL_MIL1553UDx_DISABLE_INTERRUPT</u>	Запрет прерываний
<u>IOCTL_MIL1553UDx_DISABLE_SUBADDR</u>	Запрет подадреса
<u>IOCTL_MIL1553UDx_ENABLE_SUBADDR</u>	Разрешение подадреса
<u>IOCTL_MIL1553UDx_GET_DEVICE_VER</u>	Чтение номера ревизии устройства.
<u>IOCTL_MIL1553UDx_GET_NBLOCK_RAW_DMA</u>	Количество новых данных в буфере DMA
<u>IOCTL_MIL1553UDx_GET_RESOURCE_INFO</u>	Получение информации о ресурсах и состоянии.
<u>IOCTL_MIL1553UDx_READ_BAR</u>	Чтение данных из регистрового пространства устройства (BAR)
<u>IOCTL_MIL1553UDx_READ_DMA_CH_DATA_BLOCKS</u>	Чтение программного счётчика записанных блоков данных в DMA.
<u>IOCTL_MIL1553UDx_READ_REG</u>	Чтение регистров устройства
<u>IOCTL_MIL1553UDx_READ_SUBADDR_BUF</u>	Чтение данных из буфера подадреса
<u>IOCTL_MIL1553UDx_REGISTER_INTERRUPT</u>	Ожидание прерываний.
<u>IOCTL_MIL1553UDx_SUBADDR_IRQ_RECEIVE</u>	Прерывания в случае успешного приёма данных
<u>IOCTL_MIL1553UDx_SUBADDR_IRQ_TRANSMIT</u>	Прерывания в случае успешной передачи данных
<u>IOCTL_MIL1553UDx_WRITE_BAR</u>	Запись регистров устройства
<u>IOCTL_MIL1553UDx_WRITE_REG</u>	Запись регистров устройства
<u>IOCTL_MIL1553UDx_WRITE_SUBADDR_BUF</u>	Запись данных в буфер подадреса

4. ЮСТЛ вызовы.

В данном описании приняты следующие соглашения и допущения:

- 1) по умолчанию, смещения указываются в шестнадцатеричном виде (hex);
- 2) по умолчанию, размеры указываются в десятичном виде;
- 3) по умолчанию, установленный флаг возврата говорит об удачном выполнении запроса, сброшенный же флаг возврата означает, что
 - запрос был отклонен ОС;
 - запрос не был выполнен драйвером из-за неправильных входных параметров.

Во всех примерах вызовов есть три неописанные переменные, это:

`m_hDevice` – дескриптор устройства на котором должна выполняться операция. Чтобы извлечь данные о дескрипторе устройства, используйте функцию **CreateFile**.

`lpBytesReturned` – указатель на переменную, которая получает размер переданных данных.

`Xxx` – параметр который должен быть равен либо `NULL`, либо ссылке на структуру `OVERLAPPED`, для того чтобы операция выполняется как перекрывающаяся (асинхронная) операция.

В случае удачного выполнения функция возвращает ненулевое значение. В случае ошибки возвращаемое значение равно нулю. Чтобы получить дополнительную информацию об ошибке, вызовите **GetLastError**

Вызовы сгруппированы по алфавиту.

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.

4.1 IOCTL_MIL1553UDx_CLEAR_DMA

Назначение:

По данному вызов программный счётчик вычитанных пакетов данных DMA уравнивается с регистром DMA_INDEX*(адреса 1040h, 1044h, 1048h, 104Ch).

*См. Раздел 5.1.2 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

Действие:

-

Примечание:

ВНИМАНИЕ!!!

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

Вход:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nCh</i> : Номер блока DMA	Номер блока DMA соответствует номеру канала (1-4). Значение 0 означает выбор блоков DMA всех каналов.

Выход:

Если вызов заканчивается неудачно с ошибкой STATUS_INVALID_PARAMETER, значит в параметре *m_nCh* был задан несуществующий номер канала. Если значение параметра *m_nCh* входит в диапазон 0-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4).

Пример вызова:

```
MIL1553Udx_CLEAR_DMA InputBuf;
InputBuf.m_nCh = 1;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_CLEAR_DMA,
    &InputBuf, sizeof(InputBuf),
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```


4.2 IOCTL_MIL1553UDx_DISABLE_INTERRUPT

Назначение:

Запрет прерывания *m_nInt* канала *m_nCh*. (Функции разрешения прерывания отсутствует, т.к. прерывание разрешается в вызове `IOCTL_MIL1553Udx_REGISTER_INTERRUPT`).

Действие:

Функция запрещает прерывание *m_nInt*, записывая 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).

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

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

Примечание:

Варианты `InterruptName`:

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

Вход – структура `MIL1553Udx_INT_NTFCN_IN`:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nInt</i> : номер прерывания	См. п. примечание
0x04	4	<i>m_hEvt</i> :	Не используется
0x08	4	<i>m_nCh</i> : номер канала.	1-4

Выход:

-

Пример вызова:

```
HANDLE Event= CreateEvent(NULL, TRUE, FALSE, name);
MIL1553Udx_INT_NTFCN_IN InputBuf;
InputBuf.m_nInt = INT_HDATx;
InputBuf.m_nCh = 1;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_DISABLE_INTERRUPT,
    &InputBuf, sizeof(InputBuf),
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.3 IOCTL_MIL1553UDx_DISABLE_SUBADDR

Назначение:

Запрет подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости номера подадреса функция устанавливает в единицу 31 бит регистра $RT_RCV_REGn^{(*)**}$ и сбрасывает в ноль 30 бит регистра $RT_TR_REGn^{(*)***}$.

*n – номер запрашиваемого подадреса

** см. Раздел 6.1.12 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

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

Примечание:

Если вызов заканчивается неудачно с ошибкой STATUS_INVALID_PARAMETER, значит в параметре m_nCh был задан несуществующий номер канала или значение параметра m_nNum выходит за границы диапазона 1-30.

Если значение параметра m_nCh входит в диапазон 0-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4).

Вход – структура MIL1553Udx_SUBADDR:

Смещение	Размер	Назначение	Примечание
0x00	4	m_nCh : номер канала.	1-4
0x04	4	m_nRT : приём или передача	Не используется
0x08	4	m_nNum : номер подадреса	1-30

Выход:

-

Пример вызова:

```
MIL1553Udx_SUBADDR InputBuf;
InputBuf.m_nCh = 1;
InputBuf.m_nNum = 1;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_DISABLE_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.4 IOCTL_MIL1553UDx_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 $RT_RCV_REGn^{(*)**}$ и устанавливает в единицу 30 бит регистра $RT_TR_REGn^{(*)***}$.

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

*n – номер запрашиваемого подадреса

** см. Раздел 6.1.12 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

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

Примечание:

Если вызов заканчивается неудачно с ошибкой STATUS_INVALID_PARAMETER, значит в параметре *m_nCh* был задан несуществующий номер канала или значение параметра *m_nNum* выходит за границы диапазона 1-30.

Если значение параметра *m_nCh* входит в диапазон 0-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4).

Вход – структура MIL1553Udx_SUBADDR:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nCh</i> : номер канала.	1-4
0x04	4	<i>m_nRT</i> : выбор режима работы. Если 1, то подадрес разрешается на приём, если 0, то подадрес разрешается на передачу.	
0x08	4	<i>m_nNum</i> : номер подадреса	1-30

Выход:

-

Пример вызова:

```
MIL1553Udx_SUBADDR InputBuf;
InputBuf.m_nCh = 1;
InputBuf.m_nRT = 1;
InputBuf.m_nNum = 1;

if (::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_ENABLE_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.5 IOCTL_MIL1553UDx_GET_DEVICE_VER

Назначение:

Чтение номера ревизии и типа устройства.

Действие:

Функция читает значение ревизии устройства из конфигурационного адресного пространства PCI и тип устройства из регистрового пространства платы.

Примечание:

-

Вход:

—

Выход – структура MIL1553Udx_GET_REV_OUT:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nRev</i> : Номер ревизии устройства	
0x04	4	<i>m_nType</i> : Тип устройства	1 – «PCIe-1553UD» 2 – «PCIe-1553UD2» 4 – «PCIe-1553UD4» 8 – «mPCIe-1553UD2» 16 – «XMC-1553UD2» 32 – «XMC-1553UD4»

Пример вызова:

MIL1553Udx_GET_REV_OUT Output;

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_GET_DEVICE_VER,
    NULL, NULL,
    &Output, sizeof(Output),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.6 IOCTL_MIL1553UDx_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 m-nBlock.

*См. Раздел 5.1.2 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

Примечание:

Если вызов заканчивается неудачно с ошибкой STATUS_INVALID_PARAMETER, значит в параметре m_nCh был задан несуществующий номер.

Если значение параметра m_nCh входит в диапазон 0-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4).

Вход – структура MIL1553Udx_GET_NBLOCK_DMA_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nCh</i> : Номер канала	1-4

Выход – структура MIL1553Udx_GET_NBLOCK_DMA_OUT:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nBlock</i> : Кол-во новых блоков канала <i>m_nCh</i>	

Пример вызова:

```
MIL1553Udx_GET_NBLOCK_DMA_OUT Output;
MIL1553Udx_GET_NBLOCK_DMA_IN Input;
Input.m_nCh = 1;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_GET_NBLOCK_RAW_DMA,
    &Input, sizeof(Input),
    &Output, sizeof(Output),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()

nBlock = Output.m_nBlock;
```

4.7 IOCTL_MIL1553UDx_GET_RESOURCE_INFO

Назначение:

Получение информации о ресурсах и состоянии модуля.

Действие:

Функция читает значение ревизии устройства из конфигурационного адресного пространства PCI.

Примечание:

Если вызов заканчивается неудачно с ошибкой STATUS_INVALID_PARAMETER, значит в параметре *m_nCh* был задан несуществующий номер.

Если значение параметра *m_nCh* входит в диапазон 0-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4).

Вход – структура MIL1553Udx_GET_RESOURCE_INFO_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nCh</i> : Номер шины	1-4

Выход – структура MIL1553Udx_GET_RESOURCE_INFO_OUT:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nBus</i> : Номер шины	
0x04	4	<i>m_nSlot</i> : Номер слота	
0x08	4	<i>m_nFunction</i> : Номер функции	
0x0C	4	<i>m_nDMAPhysAddr</i> : Физический адрес буфера DMA канала <i>m_nCh</i> .	
0x10	4	<i>m_nDataBufSize</i> : Размер буфера DMA канла <i>m_nCh</i> .	В байтах.*
0x14	4	<i>m_BarPhysAddr</i> : Физический адрес пространства регистра BAR	
0x18	128	<i>m_szCompileDate</i> : Текстовая строка, заканчивающаяся нулем, с информацией о времени и дате компиляции драйвера	

Пример вызова на следующей странице:

Пример вызова:

```
MIL1553Udx_GET_RESOURCE_INFO_OUT Output;  
MIL1553Udx_GET_RESOURCE_INFO_IN Input;  
Input.m_nCh = 1;  
  
if(::DeviceIoControl(  
    m_hDevice,  
    IOCTL_MIL1553Udx_GET_RESOURCE_INFO,  
    &Input,sizeof(Input),  
    &Output,sizeof(Output),  
    &lpBytesReturned,  
    xxx))  
else  
    dw = GetLastError()
```

4.8 IOCTL_MIL1553UDx_READ_BAR

Назначение:

Чтение данных из регистрового пространства устройства (BAR).

Действие:

Функция читает необходимое количество данных из желаемого адреса регистрового пространства устройства (BAR) в переменную pData.

Примечание:

Размер регистрового пространства (BAR) равен:

- для модулей с 1 каналом **0h-1FFCh**;
- для модулей с 2 каналами **0h-2FFCh**;
- для модулей с 3 каналами **0h-3FFCh**;
- для модулей с 4 каналами **0h-4FFCh**;

Рекомендуется использовать если данные больше 32 бит. Если меньше или равны 32 битам, то рекомендуется использовать вызов [IOCTL_MIL1553UDx_READ_REG](#).

Вход – структура MIL1553Udx_READ_BAR_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nOffset</i> : Смещение до нужного блока данных	см. примечание
0x04	4	<i>m_nSize</i> : Размер блока данных	

Выход:

Смещение	Размер	Назначение	Примечание
0x00	<i>m_nSize</i>	Считанный блок данных	

Пример вызова:

```
MIL1553Udx_READ_BAR_IN Input;
Input.m_nOffset = 0x0;
Input.m_nSize = 16;
Input.m_nBar = 0;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_READ_BAR,
    &Input, sizeof(Input),
    &Data, sizeof(Data),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```


4.9 IOCTL_MIL1553UDx_READ_DMA_CH_DATA_BLOCKS

Назначение:

Чтение данных из буфера DMA канала *m_nCh*. Данная функция читает любое требуемое значение новых блоков данных DMA по запрашиваемому каналу.

Действие:

По номеру запрашиваемого канала функция читает данные из соответствующего буфера DMA. Новые данные кладутся в переменную *pData* до тех пор, пока либо не наберётся запрашиваемое количество блоков, либо блоки с новыми данными не закончатся. Прочитанных блоков данных может быть меньше, чем запрашиваемых!

Примечание:

Если вызов заканчивается неудачно с ошибкой *STATUS_INVALID_PARAMETER*, значит в параметре *m_nCh* был задан несуществующий номер.

Если значение параметра *m_nCh* входит в диапазон 0-4, проверьте модель вашего модуля. В разных моделях, разное кол-во каналов (1, 2, 4).

Вход – структура *MIL1553Udx_READ_DMA_BLOCKS_CH_IN*:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nBlocks</i> : Запрашиваемое кол-во блоков данных для чтения	
0x04	4	<i>m_nCh</i> : Номер канала, с которого следует вычитать пришедшие данные.	1-4

Выход:

Смещение	Размер	Назначение	Примечание
0x01	<i>m_nBlocks*128</i>	Считанные блоки данных	

Пример вызова:

```
MIL1553Udx_READ_DMA_BLOCKS_CH_IN InputBuf;
//Размер запрашиваемых данных
UINT32 nOutputSize = 2*128;
InputBuf.m_nBlocks = 2;
InputBuf.m_nCh = 1;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_READ_DMA_CH_DATA_BLOCKS,
    &InputBuf, sizeof(InputBuf),
    &pData, nOutputSize,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
//размер прочитанных данных
Blocks = lpBytesReturned/128;
```

4.10 IOCTL_MIL1553UDx_READ_REG

Назначение:

Чтение данных из регистрового пространства устройства (BAR).

Действие:

Функция читает данные из желаемого адреса регистрового пространства устройства (BAR) в переменную pData.

Примечание:

Размер регистрового пространства (BAR) равен:

- для модулей с 1 каналом **0h-1FFCh**;
- для модулей с 2 каналами **0h-2FFCh**;
- для модулей с 3 каналами **0h-3FFCh**;
- для модулей с 4 каналами **0h-4FFCh**;

Рекомендуется использовать если данные меньше, либо равны 32 битам. Если больше 32 бит, то рекомендуется использовать функцию [IOCTL_MIL1553UDx_READ_BAR](#).

Вход – структура MIL1553Udx_READ_REG_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nAdress</i> : Адрес регистра	см. примечание

Выход:

Смещение	Размер	Назначение	Примечание
0x00	4	Считанный блок данных	

Пример вызова:

```
MIL1553Udx_READ_REG_IN InputBuf;
InputBuf.m_nAdress = 0x0;
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_READ_REG,
    &InputBuf, sizeof(InputBuf),
    &pData, sizeof(pData),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.11 IOCTL_MIL1553UDx_READ_SUBADDR_BUF

Назначение:

Чтение данных из буфера подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости от выбранного номера буфера и номера подадреса функция вычитывает блок данных запрашиваемого подадреса из буфера `RT_DATA_BUF0n(*)**` или `RT_DATA_BUF1n(*)***`.

*n – номер запрашиваемого подадреса

** см. Раздел 6.1.14 документа «Руководство по программированию «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

*** см. Раздел 6.1.15 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

Примечание:

-

Вход – структура `MIL1553Udx_RD_SUBADDR_IN`:

Смещение	Размер	Назначение	Примечание
0x00	4	<code>m_nCh</code> : Номер канала	1-4
0x04	4	<code>m_nNum</code> : Номер подадреса	1-30
0x08	4	<code>m_nBufNum</code> : Выбор буфера. Если 0, то <code>RT_DATA_BUF0n</code> , если 1, то <code>RT_DATA_BUF1n</code>	

Выход:

Смещение	Размер	Назначение	Примечание
0x00	64	Считанный блок данных	

Пример вызова:

```
MIL1553Udx_RD_SUBADDR_IN InputBuf;
```

```
InputBuf.m_nBufNum = 0;
```

```
InputBuf.m_nNum = 1;
```

```
InputBuf.m_nCh = 1;
```

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_READ_SUBADDR_BUF,
    &InputBuf, sizeof(InputBuf),
    pData, nSize,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError();
```

4.12 IOCTL_MIL1553UDx_REGISTER_INTERRUPT

Назначение:

Разрешение и ожидание прерывания *m_nInt* канала *m_nCh*.

Действие:

Функция передаёт драйверу событие (event) *m_hEvt* и информацию об ожидаемом драйвере. Драйвер 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, *hEvent* переводится в сигнальное состояние.

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

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

Примечание:

Варианты *m_nInt*:

- **INT_HDATx** – прерывание по заполнению 1/2 буфера данных контроллера *m_nCh*;
- **INT_QDATx** – прерывание по заполнению 1/16 буфера данных контроллера *m_nCh*;
- **RTx_INT_MC** – прерывание по приёму КУ контроллера *m_nCh*;
- **RTx_INT_ERR** – прерывание по ошибке сообщения контроллера *m_nCh*;
- **RTx_INT_REN** – прерывание от подадреса приёма контроллера *m_nCh* (Прерывание не будет срабатывать, если подадрес не разрешён на приём данных. Для разрешения воспользуйтесь вызовом [IOCTL_MIL1553UDx_ENABLE_SUBADDR](#));
- **RTx_INT_TEN** – прерывание от подадреса передачи контроллера *m_nCh* (Прерывание не будет срабатывать, если подадрес не разрешён на передачу данных. Для разрешения воспользуйтесь вызовом [IOCTL_MIL1553UDx_ENABLE_SUBADDR](#));
- **INT_BCx** – Прерывание контроллера *m_nCh* режима КШ.

Вход – структура MIL1553Udx_INT_NTFCN_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nInt</i> : номер прерывания	См. п. примечание
0x04	4	<i>m_hEvt</i> : Event для ожидания прерывания	
0x08	4	<i>m_nCh</i> : номер канала.	1-4

Выход:

–

Пример вызова на следующей странице:

Пример вызова:

```
MIL1553Udx_INT_NTFCN_IN InputBuf;
InputBuf.m_nInt = InterruptName;
InputBuf.m_hEvt = hEvent;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_REGISTER_INTERRUPT,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.13 IOCTL_MIL1553UDx_SUBADDR_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» и «mPCIe-1553UD2».

Примечание:

-

Вход – структура MIL1553Udx_SUBADDR_IRQ:

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nAction	Разрешение или запрет прерывания	ENABLE DISABLE
UINT32 nNum	Номер подадреса	1-30

Выход:

-

Пример вызова:

```
MIL1553Udx_SUBADDR_IRQ InputBuf;
InputBuf.m_nCh = nCh;
InputBuf.m_nAction = nAction;
InputBuf.m_nNum = nNum;

if (::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_SUBADDR_IRQ_RECEIVE,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.14 IOCTL_MIL1553UDx_SUBADDR_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» и «mPCIe-1553UD2».

Примечание:

-

Вход – структура MIL1553Udx_SUBADDR_IRQ:

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nAction	Разрешение или запрет прерывания	ENABLE DISABLE
UINT32 nNum	Номер подадреса	1-30

Выход:

-

Пример вызова:

```
MIL1553Udx_SUBADDR_IRQ InputBuf;
InputBuf.m_nCh = nCh;
InputBuf.m_nAction = nAction;
InputBuf.m_nNum = nNum;

if (::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_SUBADDR_IRQ_TRANSMIT,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.15 IOCTL_MIL1553UDx_WRITE_BAR

Назначение:

Запись данных в регистровое пространство устройства (BAR).

Действие:

Функция записывает необходимое количество данных по желаемому адресу в регистровое пространство устройства (BAR).

Примечание:

Размер регистрового пространства (BAR) равен:

- для модулей с 1 каналом **0h-1FFCh**;
- для модулей с 2 каналами **0h-2FFCh**;
- для модулей с 3 каналами **0h-3FFCh**;
- для модулей с 4 каналами **0h-4FFCh**;

Рекомендуется использовать если данные больше 32 бит. Если меньше или равны 32 битам, то рекомендуется использовать функцию [IOCTL_MIL1553UDx_WRITE_REG](#).

Вход – структура MIL1553Udx_WRITE_BAR_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nOffset</i> : Смещение блока данных на запись	см. примечание
0x04	4	<i>m_nSize</i> : Размер блока данных	В байтах
0x08	<i>m_nSize</i>	Блок данных на запись	

Выход:

–

Пример вызова:

```
MIL1553Udx_WRITE_BAR_IN nInputSize;
pInput->m_nOffset = 0x200;
pInput->m_nSize = 0x40;
pInput->m_nBar = 0;
memcpy(pInput->m_Data, pData, 0x40);

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_WRITE_BAR,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```


4.16 IOCTL_MIL1553UDx_WRITE_REG

Назначение:

Запись данных в регистровое пространство устройства (BAR).

Действие:

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

Примечание:

Размер регистрового пространства (BAR) равен:

- для модулей с 1 каналом **0h-1FFCh**;
- для модулей с 2 каналами **0h-2FFCh**;
- для модулей с 3 каналами **0h-3FFCh**;
- для модулей с 4 каналами **0h-4FFCh**;

Рекомендуется использовать если данные меньше, либо равны 32 битам. Если больше 32 бит, то рекомендуется использовать функцию [IOCTL_MIL1553UDx_WRITE_BAR](#).

Вход – структура MIL1553Udx_WRITE_REG_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nAddress</i> : Адрес регистра	см. примечание
0x04	4	<i>m_nData</i> : Записываемый блок данных	В байтах

Выход:

–

Пример вызова:

```
MIL1553Udx_WRITE_REG_IN InputBuf;
InputBuf.m_nAdress = 0x0;
InputBuf.m_nData = 0x1234;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553Udx_WRITE_REG,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

4.17 IOCTL_MIL1553UDx_WRITE_SUBADDR_BUF

Назначение:

Запись данных в буфер подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости от выбранного номера буфера и номера подадреса функция записывает в буфер RT_DATA_BUF0n^(*)** или RT_DATA_BUF1n^(*)** данные из переменной nData.

*n – номер запрашиваемого подадреса

** см. Раздел 6.1.14 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

*** см. Раздел 6.1.15 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

Примечание:

-

Вход – структура MIL1553Udx_WR_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nCh</i> : Номер канала	1-4
0x04	4	<i>m_nNum</i> : Номер подадреса	1-30
0x08	4	<i>m_nBufNum</i> : Выбор буфера. Если 0, то RT_DATA_BUF0n, если 1, то RT_DATA_BUF1n	
0x0c	64	<i>nData</i> : Блок данных на запись	

Выход:

-

Пример вызова:

```
MIL1553Udx_WR_SUBADDR_IN InputBuf;
pInput.m_nCh = 1;
pInput.m_nBufNum = 0;
pInput.m_nNum = 1;
memcpy(&pInput.m_nData, pData, 0x40);

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553UDx_WRITE_SUBADDR_BUF,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5. Обновление драйвера

Версия драйвера	Дата	Изменение
1.0	17.07.2015	- Драйвер создан
1.1	18.11.2016	- Добавлена поддержка модуля «mPCIe-1553UD2»
1.2	25.01.2017	- Добавлена поддержка модулей «XMC-1553UDx»
1.3	06.02.2017	- Добавлена проверка валидности адреса записи/чтения в функциях IOCTL_MIL1553UDx_WRITE_REG IOCTL_MIL1553UDx_WRITE_BAR IOCTL_MIL1553UDx_READ_REG IOCTL_MIL1553UDx_READ_BAR

6. Обновление руководства

Версия документа	Дата	Изменение
1.0	22.07.2015	- Документ создан
1.1	18.11.2016	- Изменены главы Введение , Установка драйвера
1.2	28.11.2016	- Изменена глава Установка драйвера .
1.3	06.02.2017	- Добавлено описание поддержки модулей «XMC-1553UDx»; - Изменены описания вызовов: IOCTL_MIL1553UDx_WRITE_REG IOCTL_MIL1553UDx_WRITE_BAR IOCTL_MIL1553UDx_READ_REG IOCTL_MIL1553UDx_READ_BAR