



Руководство (v3.4)

**По работе с драйвером модуля
«mPCIe-1553RT2»**

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

Для драйверов версии 1.6 и ниже

ОС WINDOWS



28.11.2016

ООО «Новомар» 2016 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.	Режим “CANONICAL”	7
4.1	“CanonicalDMA”	7
4.2	“CanonicalChMem”	7
5.	IOCTL вызовы	8
5.1	IOCTL_MIL1553DEV_CLEAR_DMA	9
5.2	IOCTL_MIL1553DEV_DISABLE_CANONICAL_CH_MEM	10
5.3	IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA	11
5.4	IOCTL_MIL1553DEV_DISABLE_SUBADDR	12
5.5	IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM	13
5.6	IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA	14
5.7	IOCTL_MIL1553DEV_ENABLE_SUBADDR	15
5.8	IOCTL_MIL1553DEV_GET_DEVICE_VER	16
5.9	IOCTL_MIL1553DEV_GET_INT_FLAGS	17
5.10	IOCTL_MIL1553DEV_GET_NBLOCK_RAW_DMA	18
5.11	IOCTL_MIL1553DEV_GET_RESOURCE_INFO	19
5.12	IOCTL_MIL1553DEV_READ_BAR	20
5.13	IOCTL_MIL1553DEV_READ_DMA_CH_DATA_BLOCKS	21
5.14	IOCTL_MIL1553DEV_DMA_COUNT	22
5.15	IOCTL_MIL1553DEV_READ_DMA_DATA_BLOCKS	23
5.16	IOCTL_MIL1553DEV_READ_DMA_DATABUF	24
5.17	IOCTL_MIL1553DEV_READ_DMA_STATUS	25
5.18	IOCTL_MIL1553DEV_READ_REG	26
5.19	IOCTL_MIL1553DEV_READ_SUBADDR	27
5.20	IOCTL_MIL1553DEV_READ_SUBADDR_BUF	28
5.21	IOCTL_MIL1553DEV_WRITE_BAR	29
5.22	IOCTL_MIL1553DEV_WRITE_REG	30
5.23	IOCTL_MIL1553DEV_WRITE_SUBADDR	31
5.24	IOCTL_MIL1553DEV_WRITE_SUBADDR_BUF	32
6.	Обновление драйвера	33
7.	Обновление руководства	34

1. Введение

Основой драйвера для модуля «**mPCIe – 1553RT2**», является файл `Mil1553RT2_YY_VeX.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 **Mil1553RT2x**, где X – индекс устройства, начиная с 0. Т.е. при одновременной работе двух и более устройств, одно будет иметь имя `Mil1553RT20`, второе `Mil1553RT21` и т.д.

2. Установка драйвера

Установка драйвера производится стандартными средствами установки оборудования системы Windows.

Для установки драйвера следует открыть «Диспетчер устройств», выбрать устройство с идентификатором **PCI\VEN_A203&DEV_9470&REV_12** 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 приступить к работе с устройством.

Модуль теперь можно найти в «Диспетчере устройств» в ветке «Multifunction Adapters» под именем «Novomar(R) MIL1553RT2 Controller».

2.1 Расшифровка названия драйвера.

<i>MIL1553RT2_x86_Ver1_0.sys</i>			
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>

- | | | |
|---|--------------------------|--|
| 1 | <i>MIL1553RT2</i> | Название поддерживаемого модуля |
| 2 | <i>X86</i> | Разрядность ОС, для которой предназначен драйвер |
| 3 | <i>Ver1_0</i> | Версия драйвера |
| 4 | <i>sys</i> | Расширение файла |

3. Список доступных IOCTL вызовов по версиям драйверов

Название вызова	Краткое описание
Список вызовов, доступных в драйвере версии до 1.5 и ниже	
IOCTL_MIL1553DEV_CLEAR_DMA	Сброс счётчиков DMA
IOCTL_MIL1553DEV_DISABLE_CANONICAL_CH_MEM	Выключение режима “CanonicalChMem”
IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA	Выключение режима “CanonicalDMA”
IOCTL_MIL1553DEV_DISABLE_SUBADDR	Запрет подадреса
IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM	Включение режима “CanonicalChMem”
IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA	Включение режима “CanonicalDMA”
IOCTL_MIL1553DEV_ENABLE_SUBADDR	Разрешение подадреса.
IOCTL_MIL1553DEV_GET_DEVICE_VER	Номер ревизии устройства.
IOCTL_MIL1553DEV_GET_INT_FLAGS	Флаги прерываний.
IOCTL_MIL1553DEV_GET_NBLOCK_RAW_DMA	Кол-во новых блоков данных в DMA
IOCTL_MIL1553DEV_GET_RESOURCE_INFO	Информация о ресурсах
IOCTL_MIL1553DEV_READ_BAR	Чтение регистров
IOCTL_MIL1553DEV_READ_DMA_CH_DATA_BLOCKS	Чтение данных из буфера DMA
IOCTL_MIL1553DEV_DMA_COUNT	Программный счётчик DMA
IOCTL_MIL1553DEV_READ_DMA_DATA_BLOCKS	Чтение данных из буфера DMA
IOCTL_MIL1553DEV_READ_DMA_DATABUF	Чтение данных из буфера DMA
IOCTL_MIL1553DEV_READ_DMA_STATUS	Чтение статуса DMA.
IOCTL_MIL1553DEV_READ_REG	Чтение регистров
IOCTL_MIL1553DEV_READ_SUBADDR	Чтение регистра подадреса
IOCTL_MIL1553DEV_READ_SUBADDR_BUF	Чтение данных из буфера подадреса.
IOCTL_MIL1553DEV_WRITE_BAR	Запись регистров
IOCTL_MIL1553DEV_WRITE_REG	Запись регистров
IOCTL_MIL1553DEV_WRITE_SUBADDR	Запись регистра подадреса
IOCTL_MIL1553DEV_WRITE_SUBADDR_BUF	Запись в буфер подадреса

4. Режим «CANONICAL»

В данном драйвере предусмотрен режим CANONICAL, обеспечивающий следующие два блока функциональности (запрещаемые/разрешаемые независимо).

В данных режимах недоступны некоторые функции. Для того чтобы понять доступна ли функция в данных режимах смотрите пункт “примечание” в описании функций. Если об этом ничего не сказано, то функция доступна как в данном режиме, так и вне его.

4.1 «CanonicalDMA»

Независимое чтение информации из буфера DMA по шинам (режим включается/выключается для всех шин устройства одновременно):

- чтение полученной информации с одной шины устройства не оказывает влияние на другие шины;
- при чтении с шины пользователь получает данные только этого шины.

При включении/выключении режима указатель буфера DMA устройства обнуляется.

Для включения используется функция [IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA](#), для выключения [IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA](#).

4.2 «CanonicalChMem»

“Простая” работа с буферами передачи (режим включается/выключается для каждой шины независимо).

В этом блоке реализована следующая функциональность:

- для всех подадресов шины включается режим использования одного буфера на передачу;
- при записи в буфер передачи данные становятся доступны для многократного считывания со стороны контроллера шины (КШ);
- обеспечивается целостность доступных со стороны КШ данных.

Обращение по записи/чтению возможно только к буферу 0 любого подадреса.

Для включения используется функция

[IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM](#), для выключения функция [IOCTL_MIL1553DEV_DISABLE_CANONICAL_CH_MEM](#).

5. 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.

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.

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.

При старте драйвера запрещена работа DMA. Для разрешения воспользуйтесь функцией библиотеки 4.2.1 EnableDMA* или установите в единицу нулевой бит регистра DMA_DATA_BASE самостоятельно.**

*См. Раздел 4.2.1 документа “Руководство по работе с библиотекой модуля “mPCIe-1553RT2”.

**См. Раздел 5.1.1 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

5.1 IOCTL_MIL1553DEV_CLEAR_DMA

Назначение:

По вызову данной функции программный счётчик вычитанных пакетов данных DMA уравнивается с регистром DMA_INDEX*(адрес 0x1008).

*См. Раздел 5.1.2 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Действие:

-

Примечание:**ВНИМАНИЕ!!!**

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

Вход:

-

Выход:

-

Пример вызова:

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_CLEAR_DMA,
    NULL, 0,
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.2 IOCTL_MIL1553DEV_DISABLE_CANONICAL_CH_MEM

Назначение:

Выключение режима “CanonicalChMem”* работы с буферами передачи.

*см [раздел 4.2](#) данного документа.

Действие:

Выключается режим “CanonicalChMem”, после которого становятся доступны новые функции и недоступны некоторые старые. Режим выключается только для запрашиваемого канала.

15 и 16 биты регистра MILn^(*)_CTRL_REG_PCI***, а также 31 биты регистров MILn^(*)_RT_TR_REGm^(**)**** установленные при включении режима не сбрасываются.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

***См. Раздел 5.1.5 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.13 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Для того чтобы понять работает какой-либо вызов с данным режимом или нет, смотрите раздел “примечание” просматриваемого вызова. Если про данный режим ничего не сказано, значит вызов может работать как с данным режимом, так и без него.

Вход:

–

Выход:

–

Пример вызова:

```
MIL1553_CANONICAL_CH_MEM_IN InputBuf;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_DISABLE_CANONICAL_CH_MEM,
    &InputBuf, sizeof(InputBuf),
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.3 IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA

Назначение:

Выключение режима “CanonicalDMA”* работы с DMA.

*см. [раздел 4.1](#) данного документа.

Действие:

Два программных счётчика пакетов, содержащие количество прочитанных пакетов по соответствующему каналу обнуляются. Вместо них возвращается один счётчик на оба канала. Он приравнивается к регистровому счётчику пакетов (регистр DMA_INDEX* (адрес 0x1008)).

Выключается режим “CanonicalDMA”, после которого становятся доступны новые функции и недоступны некоторые старые. Режим включается для обоих каналов одновременно.

*См. Раздел 5.1.2 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Для того чтобы понять работает какой-либо вызов с данным режимом или нет, смотрите раздел “примечание” просматриваемого вызова. Если про данный режим ничего не сказано, значит вызов может работать как с данным режимом, так и без него.

Вход:

—

Выход:

—

Пример вызова:

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA,
    NULL, NULL,
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.4 IOCTL_MIL1553DEV_DISABLE_SUBADDR

Назначение:

Запрет подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости от выбранного режима передачи или приёма и номера подадреса функция устанавливает в единицу 31 бит регистра

$MILn^{(*)}_RT_RCV_REGm^{(**)***}$ или сбрасывает в ноль тот же бит регистра

$MILn^{(*)}_RT_TR_REGm^{(**)****}$.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

*** см. Раздел 6.1.12 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.13 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

-

Вход – структура MIL1553_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	m_nRT : выбор регистра. Если 1, то регистр передачи RT_TR_REGn , если 0, то регистр приёма RT_RCV_REGn .	0-1
0x04	4	m_nNum : номер подадреса	1-31

Выход:

-

Пример вызова:

```
MIL1553_SUBADDR_IN InputBuf;
InputBuf.m_nRT = nRT;
InputBuf.m_nNum = nNum;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_DISABLE_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.5 IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM

Назначение:

Включение режима “CanonicalChMem”* работы с буферами передачи.

*см [раздел 4.2](#) данного документа.

Действие:

Биты 15 и 16 регистра MILn^(*)_CTRL_REG_PCI*** сбрасываются в ноль.

Также устанавливается в единицу 28 бит всех регистров MILn^(*)_RT_TR_REGm^(**)****.

Включается режим “CanonicalChMem”, после которого становятся доступны новые функции и недоступны некоторые старые. Режим включается только для запрашиваемого канала.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

***См. Раздел 5.1.5 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.13 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Для того чтобы понять работает какой-либо вызов с данным режимом или нет, смотрите раздел “примечание” просматриваемого вызова. Если про данный режим ничего не сказано, значит вызов может работать как с данным режимом, так и без него.

Вход:

–

Выход:

–

Пример вызова:

```
MIL1553_CANONICAL_CH_MEM_IN InputBuf;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM,
    &InputBuf, sizeof(InputBuf),
    NULL, 0,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.6 IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA

Назначение:

Включение режима “CanonicalDMA”* работы с DMA.

*см. [раздел 4.1](#) данного документа.

Действие:

Один программный счётчик вычитанных пакетов данных DMA обнуляется. Он заменяется на два, каждый из которых приравнивается к регистровому счётчику пакетов (регистр DMA_INDEX* (адрес 0x1008)) и содержит количество пакетов по соответствующей шине.

Включается режим “CanonicalDMA”, после которого становятся доступны новые функции и недоступны некоторые старые. Режим включается для обоих каналов одновременно.

*См. Раздел 5.1.2 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Для того чтобы понять работает какой-либо вызов с данным режимом или нет, смотрите раздел “примечание” просматриваемого вызова. Если про данный режим ничего не сказано, значит вызов может работать как с данным режимом, так и без него.

Вход:

—

Выход:

—

Пример вызова:

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA,
    NULL, NULL,
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.7 IOCTL_MIL1553DEV_ENABLE_SUBADDR

Назначение:

Разрешение подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости от выбранного режима передачи или приёма и номера подадреса функция сбрасывает в ноль 31 бит регистра

$MILn^{(*)}_RT_RCV_REGm^{(**)***}$ или устанавливает в единицу тот же бит регистра

$MILn^{(*)}_RT_TR_REGm^{(**)****}$.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

*** см. Раздел 6.1.12 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.13 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

-

Вход – структура MIL1553_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nRT</i> : выбор регистра. Если 1, то регистр передачи RT_TR_REGn, если 0, то регистр приёма RT_RCV_REGn.	0-1
0x04	4	<i>m_nNum</i> : номер подадреса	1-31

Выход:

-

Пример вызова:

```
MIL1553_SUBADDR_IN InputBuf;
InputBuf.m_nRT = nRT;
InputBuf.m_nNum = nNum;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_ENABLE_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.8 IOCTL_MIL1553DEV_GET_DEVICE_VER

Назначение:

Чтение номера ревизии устройства.

Действие:

Функция читает значение ревизии устройства из конфигурационного адресного пространства PCI.

Примечание:

-

Вход:

—

Выход:

Смещение	Размер	Назначение	Примечание
0x00	4	Номер ревизии устройства	

Пример вызова:

```
UINT32 Output;
```

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_GET_DEVICE_VER,
    NULL, NULL,
    &Output, sizeof(Output),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

```
Revision = Output;
```


5.9 IOCTL_MIL1553DEV_GET_INT_FLAGS

Назначение:

Чтение флагов прерываний.

Действие:

В переменную `pIntFlags` читается значение регистра `INTERRUPT*` (адрес `100Ch`). После этого флаги сбрасываются.

В переменные `nRT0Value` и `nRT1Value` читаются значения 20-23 бит регистров `MIL0_HW_STAT_REG2**` и `MIL1_HW_STAT_REG2**` соответственно. Флаги переписываются последующими данными.

* см. Раздел 5.1.3 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

** см. Раздел 6.1.10 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

-

Вход:

-

Выход – структура `MIL1553_INT_FLAGS_OUT`:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nIntValue</i> : Текущее значение регистра прерывний <code>INTERRUPT</code>	
0x04	4	<i>m_nRT1Value</i> : Текущее значение регистра прерывний <code>MIL1_HW_STAT_REG2</code>	
0x08	4	<i>m_nRT0Value</i> : Текущее значение регистра прерывний <code>MIL0_HW_STAT_REG2</code>	

Пример вызова:

`MIL1553_GET_INT_FLAGS_OUT` Output;

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_GET_INT_FLAGS,
    NULL, NULL,
    &Output, sizeof(Output),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

```
nIntValue = Output.m_nIntValue;
nRT1Value = Output.m_nRT1Value;
nRT0Value = Output.m_nRT0Value;
```

5.10 IOCTL_MIL1553DEV_GET_NBLOCK_RAW_DMA

Назначение:

Чтение количества новых блоков данных, накопленных в буфере DMA устройства по всем каналам (блок - 128 байт).

Действие:

Функция вычитает из значения регистра DMA_INDEX*(адрес 0x1008) значение программного счётчика вычитанных пакетов. Полученное количество блоков возвращается в переменной nValue.

*См. Раздел 5.1.2 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Не доступна в режиме “CanonicalDMA”. Воспользуйтесь вызовом [IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA](#).

Вход:

–

Выход – структура MIL1553_GET_NBLOCK_DMA_OUT:

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

Пример вызова:

```
MIL1553_GET_NBLOCK_DMA_OUT Output;
```

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_GET_NBLOCK_RAW_DMA,
    NULL,NULL,
    &Output, sizeof(Output),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()

nValue = Output.m_nValue;
```

5.11 IOCTL_MIL1553DEV_GET_RESOURCE_INFO

Назначение:

Получение информации о ресурсах и состоянии.

Действие:

Функция читает значение ревизии устройства из конфигурационного адресного пространства PCI.

Примечание:

-

Вход:

—

Выход – структура MIL1553_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_nDataBufPhysAddr</i> : Физический адрес буфера DMA	
0x10	4	<i>m_nDataBufSize</i> : Размер буфера DMA	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:

MIL1553_GET_RESOURCE_INFO_OUT Output;

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_GET_DEVICE_VER,
    NULL,NULL,
    &Output, sizeof(Output),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

```
pBus = Output.m_nBus;
pSlot = Output.m_nSlot;
nFunction = Output.m_nFunction;
pBufAddr = Output.m_nDataBufPhysAddr;
pBufSize = Output.m_nDataBufSize;
pBarAddr = Output.m_BarPhysAddr;
pStr = (char*)Output.m_szCompileDate;
```

5.12 IOCTL_MIL1553DEV_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:

Рекомендуется использовать если данные больше 32 бит. Если меньше или равны 32 битам, то рекомендуется использовать вызов [IOCTL_MIL1553DEV_READ_REG](#).

Вход – структура MIL1553_READ_BAR_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nOffset</i> : Смещение блока данных на чтение	
0x04	4	<i>m_nSize</i> : Размер блока данных	

Выход:

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

Пример вызова:

```
MIL1553_READ_BAR_IN Input;
Input.m_nOffset = nAddr;
Input.m_nSize = nSize;
Input.m_nBar = nBar;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_BAR,
    &Input, sizeof(Input),
    &Data, sizeof(Data),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.13 IOCTL_MIL1553DEV_READ_DMA_CH_DATA_BLOCKS

Назначение:

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

Действие:

По номеру запрашиваемого канала функция перебирает блоки данных в DMA, проверяя из какого канала пришли данные. Если из нужного то она кладёт данные в переменную pData до тех пор, пока либо не наберёт запрашиваемое количество блоков, либо блоки с новыми данными не закончатся.

Примечание:

Доступна только в режиме "CanonicalDMA". Воспользуйтесь вызовом [IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA](#).

Вход – структура MIL1553_READ_DMA_CH_DATA_BLOCKS_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 шины, пришедшие данные с которой следует вычитать.	0-1

Выход:

Смещение	Размер	Назначение	Примечание
0x00	<i>m_nBlocks</i> *128	Считанный блок данных	

Пример вызова:

```
MIL1553_READ_DMA_CH_DATA_BLOCKS_IN InputBuf;
//Размер запрашиваемых данных
UINT32 nOutputSize = nBlocks*128;
InputBuf.m_nBlocks = nBlocks;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_DMA_CH_DATA_BLOCKS,
    &InputBuf, sizeof(InputBuf),
    &Data, nOutputSize,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()

//размер прочитанных данных
pBlocks = lpBytesReturned/128;
```

5.14 IOCTL_MIL1553DEV_DMA_COUNT

Назначение:

Чтение программного счётчика записанных блоков данных в DMA.

Действие:

Функция забирает значение счётчика из внутренних переменных драйвера.

Примечание:

Не доступна в режиме "CanonicalDMA". Воспользуйтесь вызовом [IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA](#).

Вход:

–

Выход – MIL1553_DMA_COUNT_OUT:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nData</i> : Значение счётчика	В байтах

Пример вызова:

```
MIL1553_DMA_COUNT_OUT OutputBuf;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_DMA_COUNT,
    NULL, NULL,
    &OutputBuf, sizeof(OutputBuf),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()

nCount = OutputBuf.m_nCount;
```

5.15 IOCTL_MIL1553DEV_READ_DMA_DATA_BLOCKS

Назначение:

Чтение данных из буфера DMA. Данная функция читает любое требуемое количество новых блоков данных DMA.

Действие:

Чтобы воспользоваться функцией, рекомендуется узнать количество новых блоков данных с помощью вызова [IOCTL_MIL1553DEV_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:

Не доступна в режиме "CanonicalDMA". Воспользуйтесь вызовом [IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA](#).

Вход – структура MIL1553_READ_DMA_DATA_BLOCKS_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nBlocks</i> : Запрашиваемое кол-во блоков данных для чтения	

Выход:

Смещение	Размер	Назначение	Примечание
0x00	<i>m_nBlocks</i> *128	Считанный блок данных	

Пример вызова:

```
MIL1553_READ_DMA_DATA_BLOCKS_IN InputBuf;
//Размер запрашиваемых данных
UINT32 nOutputSize = nBlocks*128;
InputBuf.m_nBlocks = nBlocks;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_DMA_COUNT,
    NULL, NULL,
    &OutputBuf, sizeof(OutputBuf),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()

//размер прочитанных данных
pBlocks = lpBytesReturned/128;
```

5.16 IOCTL_MIL1553DEV_READ_DMA_DATABUF

Назначение:

Чтение данных из буфера DMA. Данная функция вычитывает только один новый блок данных.

Действие:

Данная функция читает данные из буфера DMA только по одному блоку (блок - 128 байт). Для того, чтобы проверить есть ли готовые данные надо вызвать функцию

[IOCTL_MIL1553DEV_READ_DMA_STATUS](#).

. Если возвращаемое ей значение равно '1', значит данные есть и их следует вычитать вызвав данную функцию, и снова проверить статус готовности данных DMA. И так пока статус не станет равен нулю.

Примечание:

Не доступна в режиме "CanonicalDMA". Воспользуйтесь функцией [IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA](#).

Вход:

–

Выход:

Смещение	Размер	Назначение	Примечание
0x00	128	Считанный блок данных	В байтах

Пример вызова:

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_DMA_DATABUF,
    NULL, NULL,
    &OutputBuf, 128,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```


5.17 IOCTL_MIL1553DEV_READ_DMA_STATUS

Назначение:

Чтение статуса DMA.

Действие:

Функция сравнивает программный счётчик вычитанных пакетов данных из DMA со значением регистра DMA_INDEX*(адрес 0x1008), если значения не равны, то новые данные появились, их следует вычитать и в OutputBuf вернётся 1. Если равны, то новых данных нет и в OutputBuf вернётся 0.

*См. Раздел 5.1.2 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Не доступна в режиме “CanonicalDMA”. Воспользуйтесь функцией [IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA](#).

Вход:

–

Выход – MIL1553_READ_DMA_STATUS:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nStatus</i> : Статус новых данных	

Пример вызова:

```
MIL1553_READ_DMA_STATUS OutputBuf;
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553_READ_DMA_STATUS,
    NULL, NULL,
    &OutputBuf, sizeof(OutputBuf),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.18 IOCTL_MIL1553DEV_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:

Рекомендуется использовать если данные меньше, либо равны 32 битам. Если больше 32 бит, то рекомендуется использовать функцию [IOCTL_MIL1553DEV_READ_BAR](#).

Вход – структура MIL1553_READ_REG_IN:

Смещение	Размер	Назначение	Примечание
0x00	4	<i>m_nAddress</i> : Адрес регистра	

Выход :

Смещение	Размер	Назначение	Примечание
0x00	4	Считанный блок данных	

Пример вызова:

```
MIL1553_READ_REG_IN InputBuf;
InputBuf.m_nAdress = nAddr;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_REG,
    &InputBuf, sizeof(InputBuf),
    pData, sizeof(pData),
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.19 IOCTL_MIL1553DEV_READ_SUBADDR

Назначение:

Чтение данных из регистра подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости от выбранного режима передачи или приёма и номера подадреса функция вычитывает регистры $MILn^{(*)}_{RT_RCV_REGm^{(**)}}^{***}$ или $MILn^{(*)}_{RT_TR_REGm^{(**)}}^{****}$.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

*** см. Раздел 6.1.12 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.13 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

-

Вход – структура MIL1553_WRRD_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00		<i>m_nCh</i> : Номер канала	
0x04	4	<i>m_nRT</i> : Выбор регистра. Если 1, то регистр передачи RT_TR_REGn, если 0, то регистр приёма RT_RCV_REGn.	
0x08	4	<i>m_nNum</i> : Номер подадреса	
0x0c	4	<i>m_nBufNum</i>	Не используется
0x10	4	<i>m_nData</i>	Не используется
0x14	4	<i>m_nSize</i> : Размер данных на запись	Не используется
0x18		Блок данных на запись	Не используется

Выход :

Смещение	Размер	Назначение	Примечание
0x00	4	Считанный блок данных	

Пример вызова:

```
MIL1553_WRRD_SUBADDR_IN InputBuf;
InputBuf.m_nRT = nRT;
InputBuf.m_nNum = nNum;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    pData, 4,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.20 IOCTL_MIL1553DEV_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 буфер $MILn^{(*)}_RT_DATA_BUF0m^{(**)***}$ или $MILn^{(*)}_RT_DATA_BUF1m^{(**)****}$.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

*** см. Раздел 6.1.14 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.15 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Доступна только в режиме “CanonicalChMem”. Воспользуйтесь функцией [IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM](#).

Вход – структура MIL1553_WRRD_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00		<i>m_nCh</i> : Номер канала	
0x04	4	<i>m_nRT</i>	Не используется
0x08	4	<i>m_nNum</i> : Номер подадреса	
0x0c	4	<i>m_nBufNum</i> : Выбор буфера. Если 0, то RT_DATA_BUF0n, если 1, то RT_DATA_BUF1n	
0x10	4	<i>m_nData</i>	Не используется
0x14	4	<i>m_nSize</i> : Размер данных на запись	Не используется
0x18		Блок данных на запись	Не используется

Выход :

Смещение	Размер	Назначение	Примечание
0x00	4	Считанный блок данных	

Пример вызова:

```
MIL1553_WRRD_SUBADDR_IN InputBuf;
InputBuf.m_nBufNum = nBufNum;
InputBuf.m_nNum = nNum;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_SUBADDR_BUF,
    &InputBuf, sizeof(InputBuf),
    pData, 4,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError();
```

5.21 IOCTL_MIL1553DEV_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:

Рекомендуется использовать если данные больше 32 бит. Если меньше или равны 32 битам, то рекомендуется использовать функцию [IOCTL_MIL1553DEV_WRITE_REG](#).

Вход – структура MIL1553_WRITE_BAR_IN:

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

Выход:

–

Пример вызова:

```
MIL1553_WRITE_BAR_IN nInputSize;
pInput->m_nOffset = nAddr;
pInput->m_nSize = nSize;
pInput->m_nBar = nBar;
pInput->m_Data = Data;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_WRITE_BAR
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.22 IOCTL_MIL1553DEV_WRITE_REG

Назначение:

Запись данных в регистровое пространство устройства (BAR).

Действие:

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

Примечание:

Рекомендуется использовать если данные меньше, либо равны 32 битам. Если больше 32 бит, то рекомендуется использовать функцию [IOCTL_MIL1553DEV_WRITE_BAR](#).

Вход – структура MIL1553_READ_BAR_IN:

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

Выход:

–

Пример вызова:

```
MIL1553_WRITE_REG_IN InputBuf;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_WRITE_REG,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.23 IOCTL_MIL1553DEV_WRITE_SUBADDR

Назначение:

Запись данных в регистр подадреса.

Действие:

В зависимости от выбранного канала, а также в зависимости от выбранного режима передачи или приёма и номера подадреса функция записывает в регистры MILn^(*)_RT_RCV_REGm^(**)*** или MILn^(*)_RT_TR_REGm^(**)**** данные из переменной nData.

*n – номер запрашиваемого канала.

**m – номер запрашиваемого подадреса

*** см. Раздел 6.1.12 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.13 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

-

Вход – структура MIL1553_WRRD_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00		<i>m_nCh</i> : Номер канала	
0x04	4	<i>m_nRT</i> : Выбор регистра. Если 1, то регистр передачи RT_TR_REGn, если 0, то регистр приёма RT_RCV_REGn.	
0x08	4	<i>m_nNum</i> : Номер подадреса	
0x0c	4	<i>m_nBufNum</i>	Не используется
0x10	4	<i>m_nData</i> : Данные для записи.	
0x14	4	<i>m_nSize</i> : Размер данных на запись	Не используется
0x18		Блок данных на запись	Не используется

Выход:

-

Пример вызова:

```
MIL1553_WRRD_SUBADDR_IN InputBuf;
InputBuf.m_nRT = nRT;
InputBuf.m_nNum = nNum;
InputBuf.m_nData = nData;
InputBuf.m_nCh = nCh;

if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_WRITE_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    NULL, NULL,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```

5.24 IOCTL_MIL1553DEV_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 $MILn^{(*)}_RT_DATA_BUF0m^{(**)}^{***}$ или $MILn^{(*)}_RT_DATA_BUF1m^{(**)}^{***}$ данные из переменной $nData$.

* n – номер запрашиваемого канала.

** m – номер запрашиваемого подадреса

*** см. Раздел 6.1.14 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

**** см. Раздел 6.1.15 документа “Руководство по программированию модуля “mPCIe-1553RT2”.

Примечание:

Доступна только в режиме “CanonicalChMem”. Воспользуйтесь функцией [IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM](#).

Вход – структура MIL1553_WRRD_SUBADDR_IN:

Смещение	Размер	Назначение	Примечание
0x00		m_nCh : Номер канала	
0x04	4	m_nRT	Не используется
0x08	4	m_nNum : Номер подадреса	
0x0c	4	$m_nBufNum$: Выбор буфера. Если 0, то RT_DATA_BUF0n , если 1, то RT_DATA_BUF1n	
0x10	4	m_nData : Данные для записи.	
0x14	4	m_nSize : Размер данных на запись	Не используется
0x18		Блок данных на запись	Не используется

Выход:

–

Пример вызова:

```
MIL1553_WRRD_SUBADDR_IN InputBuf;
InputBuf.m_nRT = nRT;
InputBuf.m_nNum = nNum;
InputBuf.m_nCh = nCh;
```

```
if(::DeviceIoControl(
    m_hDevice,
    IOCTL_MIL1553DEV_READ_SUBADDR,
    &InputBuf, sizeof(InputBuf),
    &Data, 4,
    &lpBytesReturned,
    xxx))
else
    dw = GetLastError()
```


6. Обновление драйвера

Версия драйвера	Дата	Изменение
1.4	18.03.2013	- Исправлена критическая ошибка в вызове IOCTL_MIL1553DEV_READ_DMA_CH_DATA_BLOCKS . Теперь проверяется 31 бит, а не 27.
1.5	03.07.2013	- Вызовы IOCTL_MIL1553DEV_READ_SUBADDR_BUF и IOCTL_MIL1553DEV_WRITE_SUBADDR_BUF теперь читают/пишут до 64 байт данных. - Вызов IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM стал сбрасывать в ноль 15 и 16 биты. - Исправлены ошибки в функциях IOCTL_MIL1553DEV_READ_DMA_DATA_BLOCKS , IOCTL_MIL1553DEV_GET_NBLOCK_RAW_DMA при переходе регистрового счётчика пакетов DMA на новый круг.
1.6	28.11.2016	- Обновлена цифровая подпись

7. Обновление руководства

Версия документа	Дата	Изменение
1.1	06.01.2013	Исправлена ошибка расхождения регистрового и программного счётчиков принятых пакетов данных по DMA. Изменено максимальное значение программного счётчика.
2	08.02.2013	<p>1 Добавлен новый раздел “Расшифровка названия драйвера”.</p> <p>2 Добавлены новые вызовы:</p> <p>IOCTL_MIL1553DEV_DISABLE_CANONICAL_CH_MEM IOCTL_MIL1553DEV_DISABLE_CANONICAL_DMA IOCTL_MIL1553DEV_DISABLE_SUBADDR IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM IOCTL_MIL1553DEV_ENABLE_CANONICAL_DMA IOCTL_MIL1553DEV_ENABLE_SUBADDR IOCTL_MIL1553DEV_GET_NBLOCK_RAW_DMA IOCTL_MIL1553DEV_READ_DMA_CH_DATA_BLOCKS IOCTL_MIL1553DEV_READ_DMA_DATA_BLOCKS IOCTL_MIL1553DEV_READ_SUBADDR IOCTL_MIL1553DEV_READ_SUBADDR_BUF IOCTL_MIL1553DEV_WRITE_SUBADDR IOCTL_MIL1553DEV_WRITE_SUBADDR_BUF</p> <p>3 В диспетчере устройств драйвер теперь отображается как “Novomar(R) MIL1553RT2 Controller”, а не “Novomar(R) MIL1553 Controller”.</p>
3	04.03.2013	<p>1 Описание режима “Canonical” вынесено из раздела “Введение” в отдельный раздел “Режим “Canonical””.</p> <p>2. Переделано описание всех вызовов. Появились разделы “Назначение” – краткое описание назначения вызова; “Действие” – описание действий вызова; “Примечание” – какие-либо важные заметки о действии вызова. “Пример вызова” – код для запуска вызова.</p> <p>3. Исправлена ошибка в вызове IOCTL_MIL1553DEV_ENABLE_CANONICAL_CH_MEM. Не записывались 15 и 16 биты.</p> <p>4. Исправлена ошибка в вызове IOCTL_MIL1553DEV_WRITE_REG. Теперь вызов не требует выходного буфера.</p>
3.01	01.07.2013	Исправлены ошибки в вызовах IOCTL_MIL1553DEV_DISABLE_SUBADDR и IOCTL_MIL1553DEV_ENABLE_SUBADDR
3.2	23.12.2014	<p>Обновлены разделы:</p> <ol style="list-style-type: none"> 1. Введение 2. Установка драйвера <ol style="list-style-type: none"> 2.1 Расшифровка названия драйвера <p>Добавлен раздел 6 Обновления драйвера</p>

Продолжение на следующей странице:

3.3	24.02.2015	Обновлён раздел “Введение” Добавлен раздел “ Список доступных ЮСТЛ вызовов по версиям драйверов ”
3.4	28.11.2016	Обновлён раздел Установка драйвера