



Руководство (v1.0)

**По работе
со статической библиотекой
модулей**

**«PCIe – 1553UDx»
«mPCIe – 1553UD1»
«mPCIe – 1553UD2»
«XMC – 1553UDx»
«CPCIS – 1553UDx»**

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

Для библиотек версии 1.0

ОС LINUX



18.11.2017

ООО «Новомар» 2017 г.

Оглавление

1. Подключение библиотеки к разрабатываемому проекту	6
1.1 Расшифровка названия библиотеки	7
2. Функции для работы с драйвером.....	8
3. Список доступных функций по версиям библиотек	9
3.1 Стандартные функции	11
3.1.1 BOOL Open(CMIL1553UDxDevice *mil_device, int nIndex).....	11
3.1.2 void Close(CMIL1553UDxDevice *mil_device).....	11
3.1.3 BOOL GetDataBufSize(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pBufSize)	12
3.1.4 BOOL GetDataBufPhysAddr(CMIL1553UDxDevice mil_device, UINT32 nCh , UINT32 *pBufAddr).....	13
3.1.5 BOOL GetBarPhysAddr(CMIL1553UDxDevice mil_device, UINT32 *pBarAddr).....	14
3.1.6 BOOL GetPCILocation(CMIL1553UDxDevice mil_device, UINT32 *pBus, UINT32 *pSlot) 15	
3.1.7 BOOL GetRevision(CMIL1553UDxDevice mil_device, UINT32 *pType, UINT32 *pRevision).....	16
3.1.8 BOOL GetDriverCompileDate(CMIL1553UDxDevice mil_device, char *pStr).....	17
3.1.9 BOOL WriteBar(CMIL1553UDxDevice mil_device, UINT32 nAddr, const void *pData, UINT32 nSize).....	18
3.1.10 BOOL ReadBar(CMIL1553UDxDevice mil_device, UINT32 nAddr, void *pData, UINT32 nSize) 19	
3.1.11 BOOL ReadReg(CMIL1553UDxDevice mil_device, UINT32 nAddr, UINT32 *pData)....	20
3.1.12 BOOL WriteReg(CMIL1553UDxDevice mil_device, UINT32 nAddr, UINT32 nData)....	21
2.1 Функции конфигурирования устройства.....	22
3.1.13 BOOL EnableDMA(CMIL1553UDxDevice mil_device).....	22
3.1.14 BOOL DisableDMA(CMIL1553UDxDevice mil_device).....	23
3.1.15 UINT32 EnableSubAddr(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nRT, UINT32 nNum).....	24
3.1.16 UINT32 DisableSubAddr(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum) 25	
3.1.17 UINT32 ReadFreeTimer(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pData) 26	
3.1.18 UINT32 ResetFreeTimer(CMIL1553UDxDevice mil_device, UINT32 nCh)	27
3.1.19 UINT32 ReadHW_STAT_REG(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pData1, UINT32 *pData2)	28
3.1.20 UINT32 SetMode(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nMode)....	29
3.1.21 UINT32 WorkEnable(CMIL1553UDxDevice mil_device, UINT32 nCh, bool bFlag).....	30
3.1.22 UINT32 BusSelection(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nMode) 31	
3.1.23 UINT32 SetRTAddress(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 Address).....	32

3.2	Функции для чтения данных.....	33
3.2.1	UINT32 GetNBlockRawDMA(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *nValue).....	33
3.2.2	UINT32 ReadDMACHDataBlocks(CMIL1553UDxDevice mil_device, UINT32 nCh, void *pData, UINT32 nBlocks, UINT32 *pBlocks)	34
3.2.3	UINT32 ReadDMACHDataBlockCheck(CMIL1553UDxDevice mil_device, UINT32 nCh, void *pData, UINT32 *nCheck).....	36
3.2.4	UINT32 ClearDMA(CMIL1553UDxDevice mil_device, UINT32 nCh)	38
3.3	Функции MIL1553 режима ОУ	39
3.3.1	UINT32 WriteSubAddrBuf(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nBufNum, UINT32 nNum, UINT32 *pData, UINT32 nSize)	39
3.3.2	UINT32 ReadSubAddrBuf(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nBufNum, UINT32 nNum, UINT32 *pData, UINT32 nSize)	41
3.3.3	UINT32 SetTrProgMode(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum) 43	
3.3.4	UINT32 SetTrAutoMode(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum) 44	
3.3.5	UINT32 SetTrBufBitReady(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum, UINT32 nBuf).....	45
3.3.6	UINT32 ReadTrBufStatus(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum, UINT32 nBuf, BOOL *bReadyFlag)	47
3.3.7	UINT32 SetRT_TCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer) 48	
3.3.8	UINT32 SetRT_T1_TRCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer) 49	
3.3.9	BOOL SetRT_T1_RCVCK(CMIL1553UDxDevice mil_device, UINT32 nTimer)	50
3.3.10	UINT32 SetMC16_CSR(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) 51	
3.3.11	UINT32 SetTX_OAC(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) ..	51
3.3.12	UINT32 SetSET_INSTR(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) 51	
3.3.13	UINT32 SetSET_SR(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)	51
3.3.14	UINT32 SetSET_BSY(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) .	51
3.3.15	UINT32 SetSET_SF(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)	51
3.3.16	UINT32 SetSET_TF(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)	51
3.4	Функции MIL1553 режима КИШ	52
3.4.1	UINT32 WriteBCFrame(CMIL1553UDxDevice mil_device, UINT32 nCh, WR_MIL1553Udx *Str, UINT32 *pData).....	52
3.4.2	UINT32 SetBC_TCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer) 54	
3.4.3	UINT32 SetBC_T1_TRCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer) 55	

3.4.4	UINT32 SetBC_T1_RCVCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer) 56
3.4.5	UINT32 SetBCTTA(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nMode) 57
3.4.6	UINT32 SetTIME_SEL(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) 58
3.4.7	UINT32 SetBCSTRT(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) ...58
3.4.8	UINT32 SetBCRE(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag).....58
3.4.9	UINT32 SetBC2RE(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag).....58
3.4.10	UINT32 SetBCR1A(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)58
3.4.11	UINT32 SetBCR2A(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)58
3.4.12	UINT32 SetBCRSB(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)58
3.4.13	UINT32 SetMENDV(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) ...58
3.4.14	UINT32 SetBSYNDV(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)..58
3.4.15	UINT32 SetBCTTM(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)58
3.4.16	UINT32 SetTTSYNEN(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag) 58
3.4.17	UINT32 SetETTSYN(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)...58
3.4.18	UINT32 SetBCRME(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)58
3.4.19	UINT32 SetTIME_UTIL_REG_PCI(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTime)60
3.4.20	UINT32 SetGPF2(CMIL1553UDxDevice mil_device, UINT32 nCh, bool bFlag))61
3.5	Работа с Flash62
3.5.1	UINT32 AutoLoadingAfterStartOn(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nArea).....62
3.5.2	UINT32 AutoLoadingAfterStartOff(CMIL1553UDxDevice mil_device, UINT32 nCh) ...63
3.5.3	UINT32 WriteRegsValueToEEPROM(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nArea).....64
3.5.4	UINT32 ReadRegsValueFromEEPROM(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nArea).....65
3.6	Функции прерываний66
3.6.1	UINT32 Register_Interrupt(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 InterruptName, UINT32 hEvent).....66
3.6.2	UINT32 DisableInterrupt(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 InterruptName).....68
3.6.3	UINT32 SubAddrIRQ_Receive(UINT32 nCh, UINT32 nAction, UINT32 nNum).....70
3.6.4	UINT32 SubAddrIRQ_Transmit(UINT32 nCh, UINT32 nAction, UINT32 nNum)71
4.	С чего начать72
4.1	Необходимый набор функций для инициализации модуля на приём данных в режиме ОУ 72
4.2	Необходимый набор функций для инициализации модуля на передачу данных в режиме ОУ 72
4.3	Набор функций для подготовки данных к передаче в режиме ОУ72

4.4	Набор функций для чтения принятых данных	72
4.5	Набор функций для инициализации модуля для работы в режиме КШ.....	73
4.6	Набор функций для программирования работы КШ для начинающего пользователя.....	73
4.7	Набор функций программирования работы КШ для продвинутого пользователя	74
5.	Обновление библиотеки.....	75
6.	Обновление руководства.....	75

Данная библиотека поддерживает следующие модули:

- «PCIE-1553UD» (1 канал передачи данных);
- «mPCIE-1553UD1» (1 канал);
- «mPCIE-1553UD2» (2 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).

Модуль «mPCIE-1553UD» данной библиотекой не поддерживается.

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

1. Подключение библиотеки к разрабатываемому проекту.

Для того, чтобы получить доступ к функциям драйвера, в разрабатываемый проект следует включить статическую библиотеку.

Библиотеку можно добавлять в проекты под операционные системы, которые имеют unix архитектуру (linux, astra-linux,...).

Для удобства компиляции вашего проекта совместно с библиотекой необходимо создать Makefile, в котором будет прописано следующее:

```
all: LinuxMIL1553UDxLibTest
```

```
clean:
```

```
$(RM) LinuxMIL1553UDxLibTest
```

```
LinuxMIL1553UDxLibTest: LinuxMIL1553UDxLibTest.c LinuxMIL1553UDxDevice.h
```

```
$(CC) $< -o $@ -lmil1553udx_linux_ver1_0 -L./
```

Где в строке LinuxMIL1553UDxLibTest прописаны те файлы, которые будут скомпилированы совместно с библиотекой. В папке с проектом должны находиться так же:

- Makefile(файл для компиляции проекта)
- LinuxMIL1553UDxDevice.h(заголовочный файл, где хранятся описания функций и переменных в библиотеке)
- libmil1553udx_linux_ver1_0.a(сама библиотека)
- LinuxMIL1553UDxLibTest.c (пользовательский файл, который использует библиотеку)

Для очистки проекта наберите **make clean**, для сборки проекта вместе с библиотекой наберите **make**. Запуск пользовательского приложения осуществляется командой:

```
./LinuxMIL1553UDxLibTest
```

В вашем проекте необходимо прописать следующие строки для использования функций библиотеки:

```
#include "LinuxMIL1553UDxDevice.h"
```

1.1 Расшифровка названия библиотеки.

<i>libmil1553udx_linux_ver1_0.a</i>			
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>

1	<i>libmil1553udx</i>	Название библиотеки
2	<i>linux</i>	ОС, для которой написана библиотека
3	<i>ver1_0</i>	Версия библиотеки
4	<i>a</i>	Расширение файла

2. Функции для работы с драйвером

Все функции поделены на два типа BOOL и UINT32 по следующему критерию:

- если в функции несколько мест, где может произойти ошибка, то функция имеет тип UINT32. Каждой ошибке присвоен свой номер. Для расшифровки смотрите соответствующую таблицу;
- если в функции только одно место, где может произойти ошибка, то она имеет тип BOOL и в случае неудачи возвращает значение FALSE.

Обе функции в случае удачного выполнения возвращают значение 1/TRUE.

Каждая функция имеет пример вызова. Для того чтобы не перепечатывать код вручную, вы можете найти пример вызова функции по названию тестовой функции в файле *LinuxMIL1553UDxLibTest.c*

Пример:

Функция библиотеки:

3.1.3 BOOL GetDataBufSize(UINT32 *pBufSize)

Её пример вызова:

```
bool testGetDMASize(){
    CMIL1553UdxDevice objEx;
    UINT32 nSize;
    Open(&objEx,0);
    if (GetDataBufSize(objEx ,&nSize))
        //В переменной nSize появилось
        значение размера буфера DMA.
    Else
        //Ошибка
    Close(&objEx);
    return true;
}
```

Название тестовой функции testGetDMASize(). По этому названию, код данной функции можно найти в файле *LinuxMIL1553UDxLibTest.c*

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.

3. Список доступных функций по версиям библиотек

Название вызова	Краткое описание
Список вызовов, используемых в библиотеке версии 1.0	
SetGPF2	Установка сброс флага GPF2
ReadDMACHDataBlockCheck	Чтение данных из буфера DMA с проверкой
WriteBCFrame	Создание скриптов для работы КШ
Open	Открытие устройства
Close	Закрытие устройства
GetDatabufSize	Размер буфера DMA
GetDatabufPhysAddr	Физический адрес буфера DMA
GetBarPhysAddr	Физический адрес пространства регистров
GetPCILocation	Шина и слот, занимаемые устройством
GetRevision	Ревизия устройства
GetDriverCompileDate	Дата компиляции драйвера
WriteBar	Запись в регистры
ReadBar	Чтение регистров
ReadReg	Чтение регистров
WriteReg	Запись в регистры
EnableDMA	Разрешение работы
DisableDMA	Запрет работы DMA.
EnableSubAddr	Разрешение подадреса
DisableSubAddr	Запрет подадреса
ReadFreeTimer	Значение неуправляемого таймера
ResetFreeTimer	Сброс неуправляемого таймера
ReadHW_STAT_REG	Чтение статусных регистров
SetMode	Выбор режима работы устройства
WorkEnable	Разрешение/запрет работы устройства
BusSelection	Выбора шины
SetRTAddress	Выбор адреса удалённого устройства
GetNBlockRawDMA	Кол-во новых блоков данных
ReadDMACHDataBlocks	Чтение данных из буфера DMA
ClearDMA	Сброс счётчиков DMA
SetRT_TCK	Инкремент основного таймера
SetRT_T1_TRCK	Таймаут передачи
SetRT_T1_RCVCK	Таймаут приёма

SetMC16_CSR, SetTX_OAC, SetSET_INSTR, SetSET_SR, SetSET_BSY, SetSET_SF, SetSET_TF	Изменение битов регистра MIL_STAT_REG
WriteSubAddrBuf	Запись данных в буфер подареса
ReadSubAddrBuf	Чтение данных из буфера подареса
SetTrProgMode	Режим передачи “Программный” вкл.
SetTrAutoMode	Режим передачи “Автомат” вкл.
SetTrBufBitReady	Установка бита готовности буфера данных подадреса
ReadTrBufStatus	Статус бита готовности буфера данных подадреса
SetBC_TCK	Инкремент таймера КШ
SetBC_T1_TRCK	Тамаут передачи КШ
SetBC_T1_RCVCK	Таймаут приёма КШ
SetBCTTA	Сброс или загрузка основного таймера КШ.
SetTIME_SEL, SetBCSTRT, SetBCRE, SetBC2RE, SetBCR1A, SetBCR2A, SetBCRSB, SetMENDV, SetBSYNDV, SetBCTTM, SetTTSYNEN, SetETTSYN, SetBCRME	Изменение битов регистра BC_CONF_REG_PCI
SetTIME_UTIL_REG_PCI	Запись в регистр TIME_UTIL_REG_PCI
AutoLoadingAfterStartOn	Загрузка регистров из флеш
AutoLoadingAfterStartOff	Выключение загрузки регистров из флеш
WriteRegsValueToEEPROM	Запись значений регистров во флеш
ReadRegsValueFromEEPROM	Чтение значения регистров из флеш
Register Interrupt	Ожидание прерываний
DisableInterrupt	Запрет прерываний
SubAddrIRQ_Receive	Прерывания в случае успешного приёма данных
SubAddrIRQ_Transmit	Прерывания в случае успешной передачи данных

3.1 Стандартные функции

3.1.1 BOOL Open(CMIL1553UDxDevice *mil_device, int nIndex)

ВНИМАНИЕ

Функция Open открывает модули «mPCIe-1553UD1», «mPCIe-1553UD2», «PCIe-1553UDx», «CPCIS-1553UDx» и «ХМС-1553UDx» целиком, вне зависимости от модели модуля и количества существующих каналов на нём.

Если функция библиотеки не требует номер канала, значит функция работает с модулем как с единым целым устройством.

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

Открытие устройства. Для открытия устройства в функцию следует передать его индекс. Если устройство в системе, то индекс равен '0'.

Если нужно работать с несколькими устройствами одновременно, то следует объявить несколько экземпляров структуры:

Пример

```
CMIL1553UdxDevice m_Device0, m_Device1;
```

Функцию Open следует вызывать с разными индексами (0, 1, 2, 3), от разных экземпляров структуры:

```
Open(&m_Device0,0); Open(&m_Device1,1);
```

Вне зависимости от успешности открытия устройства, как файла, в строку *char lib_ver[100]*, которая является членом структуры CMIL1553UdxDevice, копируется версия используемой библиотеки.

3.1.2 void Close(CMIL1553UDxDevice *mil_device)

Закрытие устройства. Для закрытия устройства функцию Close следует вызывать от того же экземпляра структуры, что и функцию Open.

Пример:

```
CMIL1553UdxDevice m_Device0;  
Open(&m_Device0,0);  
Close(&m_Device0);
```

3.1.3 BOOL GetDataBufSize(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pBufSize)

Назначение:

Чтение размера буфера DMA канала Ch.

Действие:

Функция читает значения размера буфера DMA из внутренних переменных драйвера.

Примечание:

Если в переменную pBufSize записывается значение 0, значит в переменной 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).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
UINT32 *pBufSize	Указатель на переменную, в которую будет считано значение	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testGetDMASize(){
    CMIL1553UdxDevice objEx;
    UINT32 nSize, nCh=2;
    Open(&objEx,0);
    if (GetDataBufSize(objEx,nCh, &nSize))
        //В переменной nSize появилось
        значение размера буфера DMA.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.4 BOOL GetDataBufPhysAddr(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pBufAddr)

Назначение:

Чтение физического адреса буфера DMA канала nCh.

Действие:

Функция читает начальное значение адреса буфера DMA из внутренних переменных драйвера.

Примечание:

Если в переменную pBufAddr записывается значение 0, значит в переменной 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).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
UINT32 *pBufAddr	Указатель на переменную в которую будет считано значение	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testGetDmaAddress(){
    CMIL1553UdxDevice objEx;
    UINT32 nAddress, nCh=1;
    Open(&objEx,0);
    if (GetDataBufPhysAddr(objEx,nCh, &nAddress))
        //В переменной nAddress появилось
        начальное значение адреса буфера DMA.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.5 BOOL GetBarPhysAddr(CMIL1553UDxDevice mil_device, UINT32 *pBarAddr)**Назначение:**

Чтение физического адреса начала пространства регистров (BAR).

Действие:

Функция забирает начальное значение адреса буфера DMA из внутренних переменных драйвера.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 *pBarAddr	Указатель на переменную в которую будет считано значение	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testGetBarAddress(){
    CMIL1553UdxDevice objEx;
    UINT32 nAddress;
    Open(&objEx,0);
    if (GetBarPhysAddr(objEx,&nAddress))
        //В переменной nAddress появилось
        начальное значение адреса BAR.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.6 BOOL GetPCILocation(CMIL1553UDxDevice mil_device, UINT32 *pBus, UINT32 *pSlot)**Назначение:**

Чтение номера шины и слота, занимаемые устройством (функция всегда 0).

Действие:

Функция забирает значения номера шины и слота из внутренних переменных драйвера.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 *pBus	Указатель на переменную в которую будет считано значение	
UINT32 *pSlot	Указатель на переменную в которую будет считано значение	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testGetPCILoc(){
    CMIL1553UdxDevice objEx;
    UINT32 nBus, nSlot;
    Open(&objEx,0);
    if (GetPCILocation(objEx ,&nBus, &nSlot))
        //В переменной nBus появилось значение
        занимаемой шины, а в nSlot занимаемого слота.
    Else
        //Призошла ошибка.
    Close(&objEx);
    Return true;
}
```

3.1.7 BOOL GetRevision(CMIL1553UDxDevice mil_device, UINT32 *pType, UINT32 *pRevision)

Назначение:

Чтение типа и номера ревизии модуля.

Действие:

Функция читает значение ревизии из конфигурационного адресного пространства PCI и значение типа модуля из регистрового пространства BAR.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 * pType	Указатель на переменную в которую будет считано значение	1 – «PCie-1553UD», «XMC-1553UD» «CPCIS-1553UD» «mPCie-1553UD1» 2 – «PCie-1553UD2», «XMC-1553UD2» или «CPCIS-1553UD2» 4 – «PCie-1553UD4», «XMC-1553UD4» или «CPCIS-1553UD4» 8 – «mPCie-1553UD2»
UINT32 * pRevision	Указатель на переменную в которую будет считано значение	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testGetRevision(){
    CMIL1553UdxDevice objEx;
    UINT32 nRev, nType;
    Open(&objEx,0);
    if (GetRevision(objEx, &nType, &nRev))
        //В переменной nType тип устройства
        //в переменной nRev ревизия устройства
    else
        cout<<"I/O error"<<endl;
    Close(&objEx);
    return true;
}
```


3.1.8 BOOL GetDriverCompileDate(CMIL1553UDxDevice mil_device, char *pStr)

Назначение:

Чтение текстовой строки с информацией о времени и дате компиляции драйвера.

Действие:

Функция забирает значение времени и даты компиляции драйвера из внутренних переменных драйвера.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
char *pStr	Указатель на переменную в которую будет скопирован текст	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testGetDriverCompileDate(){
    CMIL1553UdxDevice objEx;
    string sStr;
    Open(&objEx,0);
    if (GetDriverCompileDate(objEx,&sStr))
        //В переменной sStr появилась
        дата и время компиляции драйвера.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.9 BOOL WriteBar(CMIL1553UDxDevice mil_device, UINT32 nAddr, const void *pData, UINT32 nSize)

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 [3.1.12 WriteReg](#).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nAddr	Адрес начала блока памяти для записи	см. примечание
const void *pData	Указатель на буфер с записываемыми данными	
UINT32 nSize	Размер данных	1 – 4128 байт

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testWriteBar(){
    CMIL1553UdxDevice objEx;
    UINT32 nAddr = 0x0, nData[2];
    nData[0] = 0x11223344;
    nData[1] = 0x55667788;
    Open(&objEx,0);
    if (WriteBar(objEx,nAddr, &nData, 8))
        //Данные записаны по желаемому адресу в BAR.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.10 BOOL ReadBar(CMIL1553UDxDevice mil_device, UINT32 nAddr, void *pData, UINT32 nSize)

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 функцию [3.1.11 ReadReg](#).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nAddr	Адрес начала блока памяти для чтения	см. примечание
const void *pData	Указатель на буфер куда будут записываться прочтенные данные	
UINT32 nSize	Размер данных	1 – 4128 байт

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testReadBar(){
    CMIL1553UdxDevice objEx;
    UINT32 pData[2], nAddr = 0x0;
    Open(&objEx,0);
    if (ReadBar(objEx,nAddr, &pData, 8))
        //В переменной pData появились
        данные прочтённые из желаемого адреса BAR.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.11 BOOL ReadReg(CMIL1553UDxDevice mil_device, UINT32 nAddr, UINT32 *pData)**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 [3.1.10 ReadBar](#).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nAddr	Адрес начала блока памяти для чтения	см. примечание
UINT32 *pData	Указатель на переменную в которую будет считаны данные	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testReadReg(UINT32 nAddr){
    CMIL1553UdxDevice objEx;
    UINT32 pData;
    Open(&objEx,0);
    if (ReadReg(objEx,nAddr, &pData))
        //В переменной pData появились
        данные прочтённые из желаемого адреса BAR.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

3.1.12 BOOL WriteReg(CMIL1553UDxDevice mil_device, UINT32 nAddr, UINT32 nData)**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 [3.1.9 WriteBar](#).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nAddr	Адрес начала блока памяти для записи	см. примечание
UINT32 pData	Переменная с записываемыми данными	

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testWriteReg(){
    CMIL1553UdxDevice objEx;
    UINT32 nAddr = 0x2000, nData = 0;
    Open(&objEx,0);
    if (WriteReg(objEx,nAddr, nData))
        //Данные записаны по желаемому адресу в BAR.
    Else
        //Призошла ошибка.
    Close(&objEx);
    return true;
}
```

2.1 Функции конфигурирования устройства

3.1.13 BOOL EnableDMA(CMIL1553UDxDevice mil_device)

Назначение:

Разрешение работы DMA для всех каналов.

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» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

После загрузки операционной системы работа не разрешена.

Входные данные

-

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testEnableDMA(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    if (EnableDMA(objEx))
        //Работа DMA разрешена
    else
        //Призошла ошибка.
    Return true;
}
```

3.1.14 BOOL DisableDMA(CMIL1553UDxDevice mil_device)**Назначение:**

Запрет работы DMA для всех каналов.

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» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

-

Входные данные

-

Расшифровка кодов ошибок

код	Расшифровка
FALSE	Запрос от драйвера вернулся с ошибкой

Пример вызова:

```
bool testDisableDMA(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    if (DisableDMA(objEx))
        //Работа DMA запрещена
    else
        //Призошла ошибка.
    Return true;
}
```

3.1.15 UINт32 EnableSubAddr(CMIL1553UDxDevice mil_device, UINт32 nCh, UINт32 nRT, UINт32 nNum)

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», «CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

*** см. Раздел 6.1.13 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», «CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание: -

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 nRT	Выбор режима работы. Если RECEIVE, то подадрес разрешается на приём, если TRANSMIT, то подадрес разрешается на передачу	RECEIVE TRANSMIT
UINт32 nNum	Номер подадреса	1-30

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nRT не попадает в диапазон разрешённых значений
5	Значение параметра nNum не попадает в диапазон разрешённых значений

Пример вызова:

```
bool testEnableSubAddr(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    int nCh =1, nNum = 1 nRT = 1, //Enable to receive;
    if (EnableSubAddr(objEx,nCh, nRT, nNum) == 1)
        //Подадрес 1 канала 1 разрешён на передачу
    else
        //Произошла ошибка
    Close(&objEx);
    return true;}

```


3.1.16 UINт32 DisableSubAddr(CMIL1553UDxDevice mil_device, UINт32 nCh, UINт32 nNum)**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 модуля «PCIE-1553UDx», «XMC-1553UDx», «CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

*** см. Раздел 6.1.13 документа «Руководство по программированию модуля модуля «PCIE-1553UDx», «XMC-1553UDx», «CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 nNum	Номер подадреса	1-30

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nNum не попадает в диапазон разрешённых значений

Пример вызова:

```
bool testDisableSubAddr(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    int nCh =1, nNum = 1;
    if (DisableSubAddr(objEx,nCh, nNum) == 1)
        //Под адрес 1 канала 1 запрещён на приём и передачу
    else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```

3.1.17 UINT32 ReadFreeTimer(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pData)**Назначение:**

Чтение значения неуправляемого таймера.

Действие:

В переменную pData читается значение регистра FREE_TIMER* (адреса 2038h, 4038h, 6038h, 8038h).

*См. Раздел 5.1.7 документа «Руководство по программированию модуля модуля «PCIE-1553UDx», «XMC-1553UDx», «CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 *pData	Указатель на переменную куда будет считано значение.	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	В параметре 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:

```
bool testReadFreeTimer(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nTimer,
        nCh=1;
    if (ReadFreeTimer(objEx,nCh, &nTimer))
        //В переменной nTimer значение таймера
    else
        //Произошла ошибка
    return true;
}
```

3.1.18 UINт32 ResetFreeTimer(CMIL1553UDxDevice mil_device, UINт32 nCh)**Назначение:**

Сброс значения неуправляемого таймера.

Действие:

В регистр FREE_TIMER* (адреса 2038h, 4038h, 6038h, 8038h) записывается значение 0.

*См. Раздел 5.1.7 документа «Руководство по программированию модуля модуля «PCIe-1553UDx», «XMC-1553UDx», «CPCIS-1553UDx» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	В параметре 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:

```
bool testResetFreeTimer(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINт32 nCh=1;
    if (ResetFreeTimer(objEx,nCh) == 1)
        //Таймер сброшен
    else
        //Произошла ошибка
    return true;
}
```

3.1.19 UINT32 ReadHW_STAT_REG(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *pData1, UINT32 *pData2)

Назначение:

Чтение статусных регистров.

Действие:

В переменную pData1 читается значение регистра HW_STAT_REG1* (адреса 2080h, 4080h, 6080h, 8080h).

В переменную pData2 читается значение регистра HW_STAT_REG2** (адреса 2084h, 4084h, 6084h, 8084h).

*См. Раздел 6.1.9 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

**См. Раздел 6.1.10 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 *pData1	Указатель на переменную в которую будут считаны данные из регистра HW_STAT_REG1.	
UINT32 *pData2	Указатель на переменную в которую будут считаны данные из регистра HW_STAT_REG2.	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	В параметре 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:

```
bool testReadHW_STAT_REG(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nStat1,
           nStat2,
           nCh = 1;
    if (ReadHW_STAT_REG(objEx, nCh, &nStat1, &nStat2) == 1)
        //Регистры прочитаны

    else
        Произошла ошибка
    return true;
}
```

3.1.20 UINT32 SetMode(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nMode)**Назначение:**

Выбор режима работы модуля канала nCh.

Действие:

Функция записывает значение из переменной nMode 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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
UINT32 nMode	Выбранный режим работы устройства.	BM_A – адресуемый монитор шины BC – контроллер шины RT – оконечное устройство BM_NA – не адресуемый монитор шины

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	В параметре 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).
4	Значение параметра nMode не попадает в диапазон разрешённых значений

Пример вызова:

```
bool testSetMode(){
    CMIL1553UdxDevice objEx;
    UINT32 nCh = 3;
    Open(&objEx,0);
    if (SetMode(objEx, nCh, 0) == 1)
        //Режим работы установлен
    else
        //Произошла ошибка
    return true;
}
```

3.1.21 UINт32 WorkEnable(CMIL1553UDxDevice mil_device, UINт32 nCh, bool bFlag)**Назначение:**

Разрешение/запрет работы модуля канала nCh.

Действие:

Если значение переменной bFlag равно TRUE, то функция устанавливает в единицу 23 бит регистра CTRL_REG_PCI* (адреса 2000h-4000h-6000h-8000h).

Если значение переменной bFlag равно FALSE, 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» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала, работу которого требуется разрешить	1-4
bool bFlag	Значение TRUE означает разрешение работы, FALSE запрет.	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testWorkEnable(){
    CMIL1553UdxDevice objEx;
    UINт32 nCh = 1;
    Open(&objEx,0);
    if (WorkEnable(objEx, nCh, true) == 1)
        //Работа первого канала разрешена;
    else
        //Произошла ошибка
    return true;
}
```

3.1.22 UINT32 BusSelection(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nMode)**Назначение:**

Функция выбора шины канала nCh.

Действие:

Функция записывает значение переменной nMode в регистр CTRL_REG_PCI* (адреса 2000h-4000h-6000h-8000h) начиная со второго бита.

*См. Раздел 5.1.5 документа «Руководство по программированию «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

Если значение равно 0, то обе шины устройства отключены.

Если значение равно 1, то разрешена работа с шиной «А».

Если значение равно 2, то разрешена работа с шиной «В».

Если значение равно 3, то разрешена работа с обеими шинами.

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
UINT32 nMode	См. Раздел примечание.	0-3

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение переменной nMode не попадает в диапазон разрешённых значений

Пример вызова:

```
bool testBusSelection(){
    CMIL1553UdxDevice objEx;
    UINT32 nCh = 1;
    Open(&objEx,0);
    if(BusSelection(objEx, nCh, 1) == 1)
        //Разрешена работа с шиной "А"
    else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```

3.1.23 UINT32 SetRTAddress(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 Address)**Назначение:**

Выбор адреса удалённого устройства канала nCh.

Действие:

Функция записывает значение переменной Address 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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
UINT32 Address	Желаемый номер адреса оконечного устройства.	1-30

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testSetRTAddress(){
    CMIL1553DeviceUD objEx;
    UINT32 nAddress = 3;
    Open(&objEx,0);
    if(SetRTAddress(objEx, 0, nAddress))
        //Адрес удалённого устройства записан в регистр
    else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```


3.2 Функции для чтения данных

3.2.1 UINT32 GetNBlockRawDMA(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 *nValue)

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 nValue.

*См. Раздел 5.1.2 документа «Руководство по программированию модуля «PCIE-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 * nValue	Указатель на переменную в которую будут считаны данные	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testGetNBlockRawDMA(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nBlocks, nCh=1;

    if (GetNBlockRawDMA(objEx, nCh, &nBlocks) == 1)
        //В переменной nBlocks появилось количество новых блоков
        данных.
    Else
        //Произошла ошибка
    return true;
}
```

3.2.2 UINт32 ReadDMAChDataBlocks(CMIL1553UDxDevice mil_device, UINт32 nCh, void *pData, UINт32 nBlocks, UINт32 *pBlocks)

Назначение:

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

Действие:

Чтобы воспользоваться функцией, рекомендуется узнать количество новых блоков данных с помощью функции [3.2.1 GetNBlockRawDMA](#). 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:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
void *pData	Указатель на переменную в которую будут считаны данные	
UINт32 nBlocks	Запрашиваемое количество блоков данных для чтения	
UINт32 *pBlocks	Указатель на переменную в которую будет записано считанное количество новых блоков данных. Оно может быть меньше чем запрашиваемое количество.	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

Пример вызова:

```
bool testReadDMADataBlocks(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 buf[64];
    UINT32 nBlocks,
           nSize = 2,
           nCh = 1;
    if (ReadDMAChDataBlocks(objEx, nCh, &buf, nSize, &nBlocks) == 1){
        cout<<"There is "<<nBlocks<<" new data blocks in DMA "<<nCh<<"
ch"<<endl;
        if (nBlocks>0){
            int i = 32*nBlocks;
            for (int j=0; j<i; j++)
                cout<<"0x"<<buf[j]<<endl;
        }
    }
    else
        cout<<"GetNBlockRawDMA: I/O error";
    return true;
}
```

3.2.3 UINт32 ReadDMAChDataBlockCheck(CMIL1553UDxDevice mil_device, UINт32 nCh, void *pData, UINт32 *nCheck)

Назначение:

Чтение данных из буфера DMA канала nCh. Данная функция читает один новый блок данных DMA канала nCh и проверяет его на наличие ошибок.

Действие:

Функция вычитывает один блок новых данных DMA. Если новых данных нет, то в переменную **nCheck** записывается значение **0**.

После этого функция проверяет вычитанный блок данных на наличие ошибок. Если ошибок нет, то в переменную **nCheck** записывается значение **1**. Если ошибки есть, то **nCheck** может принять следующие значения:

- **2** – первое слово служебных данных не равно нулю;
- **3** – в режиме КШ – 28 бит первого слова данных (MSG_GOOD)* не равен единице;
- **4** – в других режимах работы – 26 бит первого слова данных (RCV_ERR)** равен единице;
- **5** – слово ошибок равно нулю.

Вне зависимости от ошибок прочитанный блок данных записывается в переменную pData.

После выполнения функции указатель буфера DMA сдвигается на соответствующую количеству прочитанных данных позицию.

*см. Раздел 7.3 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

*см. Раздел 6.2 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
void *pData	Указатель на переменную в которую будут считаны данные	
UINт32 * nCheck	Указатель на переменную в которую будет записан номер ишибки (см. п. Действие).	0-4

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Ошибка на этапе чтения режима работы платы

Пример вызова на следующей странице:

Пример вызова:

```
bool testReadDMADataBlocks(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 buf[32];
    UINT32 nCheck,
           nCh = 1;
    if (ReadDMAChDataBlockCheck(objEx, nCh, &buf, &nCheck) == 1){
        if (nCheck>0){
            for (int j=0; j<32; j++){
                cout<<"0x"<<buf[j]<<endl;
            }
            if (nCheck>1)
                cout<<"There is errors in DMA data blocok"<<endl;
        }
    }
    else
        cout<<"GetNBlockRawDMA: I/O error";
    return true;
}
```

3.2.4 UINT32 ClearDMA(CMIL1553UDxDevice mil_device, UINT32 nCh)

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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Действие:

-

Примечание:

ВНИМАНИЕ!!!

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

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	0-4 0 – очистить DMA по всем каналам

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testClearDMA(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    int nCh = 2;
    if (ClearDMA(objEx, nCh) == 1)
        //Счётчик уравнен, функция GetNBlockRawDMA
        //будет возвращать 0.
    Else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```

3.3 Функции MIL1553 режима ОУ

3.3.1 UINT32 WriteSubAddrBuf(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nBufNum, UINT32 nNum, UINT32 *pData, UINT32 nSize)

Назначение:

Функция записывает данные из буфера pData в буфер подадреса канала nCh.

Действие:

Функция записывает 64 байта данных из переменной pData 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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

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

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nBufNum	Номер буфера	0 – RT_DATA_BUF0n 1 –RT_DATA_BUF1n
UINT32 nNum	Номер подадреса	1-30
UINT32 *pData	Указатель на область памяти размером 64 байта	
UINT32 nSize	(Проверка) Размер области памяти pData	Только 64

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nBufNum не входит в диапазон допустимых значений.
5	Значение параметра nNum не входит в диапазон допустимых значений.
6	Значение параметра nSize не равно 64.

Пример вызова на следующей странице:

Пример вызова:

```
bool testWriteSubAddrBuf(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh=1,
           nNum=1,
           nBufNum=0,
           nData[16],
           nSize=64;
    for (int i=0; i<16; i++)
        nData[i]=0x1234;

    if (WriteSubAddrBuf(objEx, nCh, nBufNum, nNum, nData, nSize) != 1){
        //Произошла ошибка
        return FALSE;
    }
    else
        //Данные записаны
        return true;
}
```


3.3.2 UINт32 ReadSubAddrBuf(CMIL1553UDxDevice mil_device, UINт32 nCh, UINт32 nBufNum, UINт32 nNum, UINт32 *pData, UINт32 nSize)

Назначение:

Функция читает данные из буфера подадреса канала nCh в буфер pData.

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», «ХМС-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

**См. Раздел 6.1.17 документа «Руководство по программированию модуля «PCIE-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 nBufNum	Номер буфера	0 – RT_DATA_BUF0n 1 –RT_DATA_BUF1n
UINт32 nNum	Номер подадреса	1-30
UINт32 *pData	Указатель на область памяти размером 64 байта	
UINт32 nSize	(Проверка) Размер области памяти pData	Только 64

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nBufNum не входит в диапазон допустимых значений.
5	Значение параметра nNum не входит в диапазон допустимых значений.
6	Значение параметра nSize не равно 64.

Пример вызова на следующей странице:

Пример вызова:

```
bool testReadSubAddrBuf(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh=1,
           nNum=1,
           nBufNum=0,
           nData[16],
           nSize=64;
    if (ReadSubAddrBuf(objEx, nCh, nBufNum, nNum, nData, nSize) != 1){
        //Произошла ошибка
        return FALSE;
    }
    else{
        //Данные прочитаны из буфера подадреса в буфер nData
    }
    return true;
}
```

3.3.3 UINT32 SetTrProgMode(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum)

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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

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

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nNum	Номер подадреса.	1-30

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nNum не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetTrProgMode(){
    CMIL1553UdxDevice objEx;
    UINT32 nCh = 2,
           nSubAddr = 1;
    Open(&objEx,0);
    if(SetTrProgMode(objEx, nCh, nSubAddr) == 1)
        //Режим передачи данных “Программный” включён
    else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```

3.3.4 UINT32 SetTrAutoMode(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum)

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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

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

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nNum	Номер подадреса.	1-30

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nNum не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetTrAutoMode(){
    CMIL1553UdxDevice objEx;
    UINT32 nCh = 2,
           nSubAddr = 10;
    Open(&objEx,0);
    if(SetTrAutoMode(objEx, nCh, nSubAddr) == 1)
        //Режим передачи данных “Автомат” включён
    else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```

3.3.5 UINT32 SetTrBufBitReady(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nNum, UINT32 nBuf)

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.*

Для проверки статуса бита воспользуйтесь функцией [ReadTrBufStatus](#).

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

Примечание:

nBuf = **NONE**:

22 и 23 бит сбрасываются в ноль (оба буфера не готовы к передаче);

nBuf = **BUF0**:

22 бит устанавливается в 1, а 23 сбрасывается в ноль (к передаче готов только буфер 0);

nBuf = **BUF1**:

22 бит сбрасывается в ноль, а 23 устанавливается в 1 (к передаче готов только буфер 1);

nBuf = **BOTH**:

22 и 23 бит устанавливаются в единицу (оба буфера готовы к передаче).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nNum	Номер подадреса.	1-30
UINT32 nBuf	Режим готовности буфера	0-3

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nNum не входит в диапазон допустимых значений.
5	Значение параметра nBuf не входит в диапазон допустимых значений.

Пример вызова на следующей странице:

Пример вызова:

```
bool testSetTrBufBitReady(){
    CMIL1553UdxDevice objEx;
    UINT32 nCh = 2, nSubAddr = 10,
        nBufReady = BOTH;    //both buffers
    Open(&objEx,0);
    if(SetTrBufBitReady(objEx, nCh,nSubAddr,nBufReady) == 1)
        //Бит готовности выставлен
    Else //Произошла ошибка
    Close(&objEx);
    return true; }
```

3.3.6 UINт32 ReadTrBufStatus(CMIL1553UDxDevice mil_device, UINт32 nCh, UINт32 nNum, UINт32 nBuf, BOOL *bReadyFlag)

Назначение:

Чтение бита статуса готовности буфера nBuf канала nCh.

Действие:

nBuf = **BUF0**. Если 22 бит регистра RT_TR_REGn* установлен в 1, то bReadyFlag будет равна true, если же бит сброшен в 0, то bReadyFlag будет равна false.

nBuf = **BUF1**. Если 23 бит регистра RT_TR_REGn* установлен в 1, то bReadyFlag будет равна true, если же бит сброшен в 0, то bReadyFlag будет равна false.

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

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 nNum	Номер подадреса.	1-30
UINт32 nBuf	Номер буфера для передачи данных	BUF0, BUF1
Bool bReadyFlag	Указатель, в который запишется информация о состоянии бита	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nNum не входит в диапазон допустимых значений.
5	Значение параметра nBuf не входит в диапазон допустимых значений.

Пример вызова:

```
bool testReadTrBufStatus(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINт32 nCh = 2, nSubAddr = 10, nBuf = 0;
    BOOL flag=false;
    if(ReadTrBufStatus(objEx, nCh,nSubAddr,nBuf, &flag) != 1)
        //Произошла ошибка
    else{
        if (flag)
            //Буфер готов к отправке
        else
            //Буфер не готов к отправке
        }
    Close(&objEx);
    return true; }
```

3.3.7 UINT32 SetRT_TCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer)

Назначение:

Задание значения инкремента основного таймера канала nCh для всех режимов за исключением КШ.

Действие:

Функция записывает желаемое значение таймера в регистр RTBM_CONF_REG_PCI* (адреса 2004h, 4004h, 6004h, 8004h) начиная с шестнадцатого бита.

*См. Раздел 6.1.1 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nTimer	Выбранное значение инкремента.	0 – выключен Варианты инкрементов: 1, 2, 4, 8, 16, 32, 64 мкс

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nTimer не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetRT_TCK(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetRT_TCK(objEx, nCh, nTimer) == 1)
        //RT_TCK установлен
    else
        //Произошла ошибка
    return true;
}
```


3.3.8 UINT32 SetRT_T1_TRCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer)

Назначение:

Задание таймаута передачи t1 для всех режимов канала nCh за исключением режима КШ.

Действие:

Функция записывает желаемое значение таймаута в регистр RTBM_CONF_REG_PCI* (адреса 2004h, 4004h, 6004h, 8004h) начиная с восьмого бита.

*См. Раздел 6.1.1 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nTimer	Выбранное значение таймаута.	6, 8, 11, 13, 18, 61, 86, 111 мкс

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nTimer не входит в диапазон допустимых значений.

Пример вызова:

```
bool test SetRT_T1_TRCK (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetRT_T1_TRCK(objEx, nCh, nTimer) == 1)
        //RT_T1_TRCK установлен
    else
        //Произошла ошибка
    return true;
}
```

3.3.9 BOOL SetRT_T1_RCVCK(CMIL1553UDxDevice mil_device, UINT32 nTimer)

Назначение:

Задание таймаута приёма t1 канала nCh для всех режимов за исключением режима КШ.

Действие:

Функция записывает желаемое значение таймера в регистр RTBM_CONF_REG_PCI* (адреса 2004h, 4004h, 6004h, 8004h) начиная с нулевого бита.

*См. Раздел 6.1.1 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nTimer	Выбранное значение таймаута.	17, 60, 85, 110 мкс

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nTimer не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetRT_T1_RCVCK (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetRT_T1_RCVCK(objEx, nCh, nTimer) == 1)
        //RT_T1_RCVCKустановлен
    else
        //Произошла ошибка
    return true;
}
```

Важно!

С помощью функций с номера 3.3.6 по номер 3.3.12 можно регулировать значение одиночных битов регистра MIL_STA_REG.

3.3.10 UINT32 SetMC16_CSR(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.3.11 UINT32 SetTX_OAC(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.3.12 UINT32 SetSET_INSTR(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.3.13 UINT32 SetSET_SR(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.3.14 UINT32 SetSET_BSY(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.3.15 UINT32 SetSET_SF(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.3.16 UINT32 SetSET_TF(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

Назначение:

Изменение значения отдельного бита регистра MIL_STAT_REG*(адреса 2088h, 4088h, 6088h, 8088h) канала nCh.

*См. Раздел 6.1.11 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Действие:

Если значение переменной bFlag равно TRUE, то значение бита будет установлено в единицу. Если значение переменной bFlag равно FALSE, то значение бита будет сброшено в ноль.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
BOOL bFlag	Выбранное значение	TRUE, FALSE

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testSetSET_TF (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;
    if (SetSET_TF(objEx, nCh, bFlag) == 1)
        // SetSET_TF установлен
    else
        //Произошла ошибка
    return true;}

```

3.4 Функции MIL1553 режима КШ

3.4.1 UINт32 WriteBCFrame(CMIL1553UDxDevice mil_device, UINт32 nCh, WR_MIL1553Udx *Str, UINт32 *pData)

Назначение:

Создание и запись фреймов(инструкций, операций и данных) для программирования КШ. В зависимости от входных параметров, функция составляет инструкции, операции и записывает их. При необходимости также записывает данные для передачи. Запуск инструкций необходимо производить с помощью функции [SetBCSTRт](#). Для более подробной информации см. раздел «С чего начать данного документа».

Действие:

Инструкции записываются в область памяти BC_INSTR_RAM¹.
Операции записываются в область памяти BC_OPERATION_RAM².
Данные, при необходимости, записываются в область памяти BC_DATA_RAM³.
Старт выполнения инструкций запускается установкой в единице бита BCSTRт регистра BC_CONF_REG_PCI⁴.

¹См. Раздел 7.1.1 документа «Руководство по программированию модуля «PCIE-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «тPCIE-1553UD1», «тPCIE-1553UD2».

²См. Раздел 7.1.2 документа «Руководство по программированию модуля «PCIE-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «тPCIE-1553UD1», «тPCIE-1553UD2».

³См. Раздел 7.1.3 документа «Руководство по программированию модуля «PCIE-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «тPCIE-1553UD1», «тPCIE-1553UD2».

⁴См. Раздел 7.2.1 документа «Руководство по программированию модуля «PCIE-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «тPCIE-1553UD1», «тPCIE-1553UD2».

Примечание: -

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 nCount	Номер фрейма в цикле записи	> 0
Ссылка на структуру типа WR_MIL1553Udx		
UINт32 nAddr	Адрес ОУ	1-31
UINт32 nSubAddr	Подадрес ОУ	1-31
UINт32 nRcvTr	Признак приёма/передачи	0 – передача 1 – приём
UINт32 nDWCCount	Количество шестнадцатитбитных слов данных	0-32
UINт32 nRptTime	Период повтора передачи фрейма	0-0xFFFF
UINт32 nRptCount	Количество повторов передачи фрейма	0-0xFFFF
UINт32 nFormat	Формат передачи	0-10 1 – на передачу, 2 – на приём.
Конец структуры		
UINт16 *pData	Указатель на блок памяти с данными, которые будут переданы в линию. Размер данных – до 64 байт включительно.	

Расшифровка кодов ошибок

код	Расшифровка
9	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).
2	Значение параметра nRptCount не попадает в диапазон разрешённых значений
3	Значение параметра nRptTime не попадает в диапазон разрешённых значений
4	Значение параметра nAddr не попадает в диапазон разрешённых значений
5	Значение параметра nSubAddr не попадает в диапазон разрешённых значений
6	Значение параметра nDWCount не попадает в диапазон разрешённых значений
7	Значение параметра nRcvTr не попадает в диапазон разрешённых значений
8	Значение параметра nFormat не попадает в диапазон разрешённых значений
10	Ошибка драйвера при попытке записи инструкций.
11	Ошибка драйвера при попытке записи операций.
12	Ошибка драйвера при попытке записи данных.
13	Память данных заполнена (BC_DATA_RAM).

Пример вызова:

```

WR_MIL1553Udx Str;
    Str.nAddr = 2, //Адрес ОУ
    Str.nSubAddr = 1, //Поадрес ОУ
    Str.nRcvTr = 0, //Признак приём = 1, передача = 0
    Str.nRptTime = 0xff,
    Str.nRptCount = 0, //0 – бесконечно
    Str.nFormat = 1, //2 – на приём, 1 – на передачу
    Str.nDWCount = 32;
UINT16 pData[32]<;
UNIT32 nCount

for (int i=0; i<32; i++)
    pData[i] = 2;

nErr = lpWR_FRM(objEx, nCh, nCount, &Str, pData);
if (nErr != 1){
    cout<<"Error writing frame. Err = "<<nErr<<endl;
}
else
cout<<"Frame has been written.\n";

```

3.4.2 UINT32 SetBC_TCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer)

Назначение:

Задание значения инкремента основного таймера канала nCh для режима КШ.

Действие:

Функция записывает желаемое значение таймера в регистр BC_CONF_REG_PCI* (адреса 2008h, 4008h, 6008h, 8008h) начиная с двадцать четвертого бита.

*См. Раздел 7.1.5 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nTimer	Выбранное значение инкремента.	0 – выключен Варианты инкрементов: 1, 2, 4, 8, 16, 32, 64 мкс

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nTimer не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetBC_TCK (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetBC_TCK(objEx, nCh, nTimer) == 1)
        // SetBC_TCK установлен
    else
        //Произошла ошибка
        return true;
}
```

3.4.3 UINT32 SetBC_T1_TRCK(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTimer)

Назначение:

Задание таймаута передачи t1 канала nCh для режима КШ.

Действие:

Функция записывает желаемое значение таймаута в регистр BC_CONF_REG_PCI* (адреса 2008h, 4008h, 6008h, 8008h) начиная с двадцать первого бита.

*См. Раздел 7.1.5 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nTimer	Выбранное значение таймаута.	6, 8, 11, 13, 18, 61, 86, 111 мкс

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nTimer не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetBC_T1_TRCK (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetBC_T1_TRCK(objEx, nCh, nTimer) == 1)
        // SetBC_T1_TRCK установлен
    else
        //Произошла ошибка
    return true;
}
```

3.4.4 UINт32 SetBC_T1_RCVCK(CMIL1553UDxDevice mil_device, UINт32 nCh, UINт32 nTimer)

Назначение:

Задание таймаута приёма t1 канала nCh для режима КШ.

Действие:

Функция записывает желаемое значение таймера в регистр BC_CONF_REG_PCI* (адреса 2008h, 4008h, 6008h, 8008h) начиная с девятнадцатого бита.

*См. Раздел 7.1.5 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 nTimer	Выбранное значение таймаута.	17, 60, 85, 110 мкс

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nTimer не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetBC_T1_RCVCK (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINт32 nCh = 1,
        nTimer = 1;

    if (SetBC_T1_RCVCK(objEx, nCh, nTimer) == 1)
        //BC_T1_RCVCKустановлен
    else
        //Произошла ошибка
    return true;
}
```


3.4.5 UINT32 SetBC_TTA(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nMode)

Назначение:

Сброс или загрузка основного таймера КИШ канала nCh.

Действие:

Функция записывает желаемое значение таймера в регистр BC_CONF_REG_PCI* (адреса 2008h, 4008h, 6008h, 8008h) начиная с тридцатого бита.

*См. Раздел 7.1.5 документа «Руководство по программированию модуля «PCIe-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nMode	Выбранное воздействие.	0 – не оказывает воздействия на таймер 1 – загрузка значением из TIME_UTIL_REG_PCI 2 – сброс таймера 3 – не оказывает воздействия на таймер

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nMode не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSetBC_TCK (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetBC_TCK(objEx, nCh, nTimer) == 1)
        //RT_T1_RCVCKустановлен
    else
        //Произошла ошибка
    return true;
}
```

Важно!

С помощью функций с номера 3.4.5 по номер 3.4.17 можно регулировать значение одиночных битов регистра BC_CONF_REG_PCI.

3.4.6 UINT32 SetTIME_SEL(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.7 UINT32 SetBCSTRT(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.8 UINT32 SetBCRE(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.9 UINT32 SetBC2RE(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.10 UINT32 SetBCR1A(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.11 UINT32 SetBCR2A(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.12 UINT32 SetBCRSB(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.13 UINT32 SetMENDV(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.14 UINT32 SetBSYNDV(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.15 UINT32 SetBCTTM(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.16 UINT32 SetTTSYNEN(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.17 UINT32 SetETTSYN(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

3.4.18 UINT32 SetBCRME(CMIL1553UDxDevice mil_device, UINT32 nCh, BOOL bFlag)

Назначение:

Изменение значения отдельного бита регистра BC_CONF_REG_PCI*.

*См. Раздел 7.1.5 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Действие:

Если значение переменной bFlag равно TRUE, то значение бита будет установлено в единицу. Если значение переменной bFlag равно FALSE, то значение бита будет сброшено в ноль.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
BOOL bFlag	Выбранное значение	TRUE, FALSE

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testSetBCRME (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;
    if (SetBCRME (objEx, nCh, bFlag) == 1)
        // SetBCRME установлен
    else
        //Произошла ошибка
return true;}
```

3.4.19 UINT32 SetTIME_UTIL_REG_PCI(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nTime)

Назначение:

Запись значения из переменной nTime в регистр TIME_UTIL_REG_PCI* (адрес 200Ch).

*См. Раздел 7.1.6 документа «Руководство по программированию модуля «PCie-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCie-1553UD1», «mPCie-1553UD2».

Действие:

-

Примечание:

При установке бита 30 регистра управления КШ в логическую 1, данные из TIME_UTIL_REG_PCI переписываются в основной таймер КШ.

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nTime	Выбранное значение	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool test SetTIME_UTIL_REG_PCI(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;

    if (SetTIME_UTIL_REG_PCI (objEx, nCh, nTimer) == 1)
        // SetTIME_UTIL_REG_PCI установлен
    else
        //Произошла ошибка
    return true;
}
```

3.4.20 UINT32 SetGPF2(CMIL1553UDxDevice mil_device, UINT32 nCh, bool bFlag)**Назначение:**

Регулировка цикличного выполнения скриптов КШ. Для более подробной информации см. пункт [«С чего начать»](#) данного документа.

Установка или сброс флага GPF2 регистра GPF/BC_COND_CODE*.

*См. Раздел 7.2.4 документа «Руководство по программированию модуля «PCIe-1553UDx, «ХМС-1553UDx и «mPCIe-1553UD2.

Действие:

Если bFlag равен true, то функция устанавливает в единицу 18-ый бит регистра GPF/BC_COND_CODE, если равен false, то функция устанавливает в единицу 26 бит этого же регистра.

Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
bool bFlag		True, false

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
CMIL1553UDxDevice objEx;
Open(&objEx,0);

if(SetGPF2(objEx, nCh, bFlag) == 1)
    cout<<"Flag GPF2 was set/unset. Ch="<<nCh<<endl;
else{
    Close(&objEx);
    cout<<"SetGPF2: I/O error"<<endl;
    return false;
}

Close(&objEx);
```

3.5 Работа с Flash

3.5.1 UINT32 AutoLoadingAfterStartOn(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nArea)

Назначение:

Включение автоматической загрузки значений регистров выбранной области из флэш канала nCh.

Действие:

Значение переменной nArea записывается в регистр CTRL_REG_PCI* (адреса 2000h, 4000h, 6000h, 8000h) начиная с двадцать девятого бита.

Также двадцать восьмой бит этого же регистра устанавливается в единицу.

*См. Раздел 5.1.5 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

До вызова этой функции, следует выключить разрешение работы устройства. Для этого можете воспользоваться функцией [WorkEnable](#).

После включения необходимо записать новое значение изменённого регистра.

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nArea	Выбранное значение области регистров для восстановления после включения.	(область адресов приведена для первого канала) 22- действия не производятся 1 – область адресов 2000...2028h. 2 – область адресов 2000...2028h; 2100...21FCh. 3 – область адресов 2000...2028h; 2100...21FCh; 200...22FC.

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

```
bool testAutoLoadingAfterStartOn(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;
    if (AutoLoadingAfterStartOn(objEx, nCh, nArea) == 1)
        // AutoLoadingAfterStartOn установлен
    else
        //Произошла ошибка
    return true;}

```

3.5.2 UINт32 AutoLoadingAfterStartOff(CMIL1553UDxDevice mil_device, UINт32 nCh)

Назначение:

Выключение автоматической загрузки значений регистров выбранной области из флэш канала nCh.

Действие:

Биты с двадцать восьмого по тридцатый регистра CTRL_REG_PCI* (адреса 2000h, 4000h, 6000h, 8000h) сбрасываются в ноль.

*См. Раздел 7.1.5 документа «Руководство по программированию модуля «PCIe-1553UDx», «ХМС-1553UDx», CPCIS-1553UDx» и «mPCIe-1553UD1», «mPCIe-1553UD2».

Примечание:

До вызова этой функции, следует выключить разрешение работы устройства. Для этого можете воспользоваться функцией [WorkEnable](#).

Входные данные

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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:

См. Пример SetRT_TCK()

3.5.3 UINT32 WriteRegsValueToEEPROM(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nArea)

Назначение:

Функция записывает во флеш значения регистров выбранного диапазона адресов.

Действие:

Функция записывает значение из переменной nArea в регистр EEPROM_CONFIG* (адреса 207Ch, 407Ch, 607Ch, 807Ch) начиная со второго бита и устанавливает в единицу четвёртый бит этого же регистра.

*См. Раздел 5.1.6 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

До вызова этой функции, следует выключить разрешение работы устройства. Для этого можете воспользоваться функцией [WorkEnable](#).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nArea	Выбранная область регистров для записи их значений.	(область адресов приведена для первого канала) 22- действия не производятся 1 – область адресов 2000...2028h. 2 – область адресов 2000...2028h; 2100...21FCh. 3 – область адресов 2000...2028h; 2100...21FCh; 200...22FC.

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nArea не входит в диапазон допустимых значений.
5	Работа устройства не прекращена. Сбросьте в ноль 23 бит регистра CTRL_REG_PCI (WorkEnable)

Пример вызова:

```
bool testWriteRegsValueToEEPROM (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;
    if (WriteRegsValueToEEPROM(objEx, nCh, nArea) == 1)
        // WriteRegsValueToEEPROM установлен
    else
        //Произошла ошибка
    return true;}

```


3.5.4 UINT32 ReadRegsValueFromEEPROM(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 nArea)

Назначение:

Функция читает из флеш значения регистров выбранного диапазона адресов.

Действие:

Функция записывает значение из переменной nArea в регистр EEPROM_CONFIG (адреса 207Ch, 407Ch, 607Ch, 807Ch) начиная с нулевого бита. Четвёртый бит данного регистра сбрасывается в ноль.

*См. Раздел 5.1.6 документа «Руководство по программированию модуля «PCIE-1553UDx», «XMC-1553UDx», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

До вызова этой функции, следует выключить разрешение работы устройства. Для этого можете воспользоваться функцией [WorkEnable](#).

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 nArea	Выбранная область регистров для чтения их значений.	(область адресов приведена для первого канала) 22- действия не производятся 1 – область адресов 2000...2028h. 2 – область адресов 2000...2028h; 2100...21FCh. 3 – область адресов 2000...2028h; 2100...21FCh; 200...22FC.

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nArea не входит в диапазон допустимых значений.
5	Работа устройства не прекращена. Сбросьте в ноль 23 бит регистра CTRL_REG_PCI (WorkEnable)

Пример вызова:

```
bool testReadRegsValueFromEEPROM (){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1,
           nTimer = 1;
    if (ReadRegsValueFromEEPROM (objEx, nCh, nArea) == 1)
        // ReadRegsValueFromEEPROM установлен
    else
        //Произошла ошибка
    return true;}

```

3.6 Функции прерываний

3.6.1 UINT32 Register_Interrupt(CMIL1553UDxDevice mil_device, UINT32 nCh, UINT32 InterruptName, UINT32 hEvent)

Назначение:

Разрешение и ожидание прерывания InterruptName канала nCh.

Действие:

Функция передаёт драйверу событие (event) hEvent и информацию об ожидаемом прерывании. Драйвер разрешает прерывание InterruptName 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», CPCIS-1553UDx» и «mPCIE-1553UD1», «mPCIE-1553UD2».

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

Примечание: Варианты InterruptName:

- **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. Для разрешения воспользуйтесь функцией [EnableSubAddr](#) и [SubAddrIRQ Receive](#));
- **RTx_INT_TEN** – прерывание от подадреса передачи контроллера nCh (Прерывание не будет срабатывать, если подадрес не разрешён на передачу данных. Для разрешения воспользуйтесь функцией [EnableSubAddr](#) и [SubAddrIRQ Transmit](#));
- **INT_BCx** – Прерывание контроллера nCh режима КШ.

Входные данные

Переменная	Описание	Диапазон значений
UINT32 nCh	Номер канала	1-4
UINT32 InterruptName	Идентификатор прерывания	См. п. примечание
UINT32 hEvent	Event для ожидания прерывания	

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра InterruptName не входит в диапазон допустимых значений.

Пример вызова:

```
CMIL1553UdxDevice objEx;  
Open(&objEx,0);  
wchar_t name[17] = L"MyInterruptEvent";  
UINT32 nCh = 1;  
  
HANDLE Event= CreateEvent(NULL, TRUE, FALSE, name);  
if (!Event){  
    return FALSE;  
}  
  
if (Register_Interrupt(objEx,nCh, INT_HDATx, Event) != 1){  
    //Произошла ошибка (проверьте значения входных параметров)  
    return FALSE;  
}
```

3.6.2 UINт32 DisableInterrupt(CMIL1553UDxDevice mil_device, UINт32 nCh, UINт32 InterruptName)

Назначение:

Запрет прерывания InterruptName канала nCh. (Функции разрешения прерывания отсутствует, т.к. прерывание разрешается в функции Register_Interrupt)

Действие:

Функция запрещает прерывание InterruptName, 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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

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

Примечание:

Варианты InterruptName:

- 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

Переменная	Описание	Диапазон значений
UINт32 nCh	Номер канала	1-4
UINт32 InterruptName	Идентификатор прерывания	См. п. примечание

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра InterruptName не входит в диапазон допустимых значений.

Пример вызова на следующей странице:

Пример вызова:

```
bool testDisable_Interrupt(){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);
    UINT32 nCh = 1;

    if (DisableInterrupt(objEx, nCh, RTx_INT_ERR) != 1){
        //Произошла ошибка
        return FALSE;
    }
    //Прерывание RTx_INT_ERR было запрещено
    return true;
}
```

3.6.3 UINT32 SubAddrIRQ_Receive(UINT32 nCh, UINT32 nAction, UINT32 nNum)

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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные:

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

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nAction не входит в диапазон допустимых значений.
	Значение параметра nNum не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSubAddrIRQ_Receive(int nCh, int nAction){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);

    int nNum = 3;
    if (SubAddrIRQ_Receive(objEx, nCh, nAction, nNum) == 1)
        //Прерывания разрешены
    else
        //Произошла ошибка

    Close(&objEx);
    return true;
}
```

3.6.4 UINT32 SubAddrIRQ_Transmit(UINT32 nCh, UINT32 nAction, UINT32 nNum)

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» и «mPCIE-1553UD1», «mPCIE-1553UD2».

Примечание:

-

Входные данные:

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

Расшифровка кодов ошибок

код	Расшифровка
2	Запрос от драйвера вернулся с ошибкой
3	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).
4	Значение параметра nAction не входит в диапазон допустимых значений.
	Значение параметра nNum не входит в диапазон допустимых значений.

Пример вызова:

```
bool testSubAddrIRQ_Receive(int nCh, int nAction){
    CMIL1553UdxDevice objEx;
    Open(&objEx,0);

    int nNum = 3;
    if (SubAddrIRQ_Receive(objEx, nCh, nAction, nNum) == 1)
        //Прерывания разрешены
    else
        //Произошла ошибка
    Close(&objEx);
    return true;
}
```

4. С чего начать

4.1 Необходимый набор функций для инициализации модуля на приём данных в режиме ОУ

[SetRTAddress\(\);](#)

[SetMode\(\);](#) Выбрать режим ОУ

[EnableSubAddr\(\);](#) Второй входной параметр nRT должен иметь значение 1 либо RECEIVE (подадрес разрешается на приём).

[BusSelection\(\);](#)

[WorkEnable\(\);](#) При инициализации модуля, разрешать работу следует последним действием.

[EnableDMA\(\);](#)

4.2 Необходимый набор функций для инициализации модуля на передачу данных в режиме ОУ

[SetRTAddress\(\);](#)

[SetMode\(\);](#) Выбрать режим ОУ

[EnableSubAddr\(\);](#) Второй входной параметр nRT должен иметь значение 0 либо TRANSMIT (подадрес разрешается на передачу).

[BusSelection\(\);](#)

[WorkEnable\(\);](#) При инициализации модуля, разрешать работу следует последним действием.

[EnableDMA\(\);](#)

4.3 Набор функций для подготовки данных к передаче в режиме ОУ

[WriteSubAddrBuf\(\)](#)

[SetTrProgMode\(\)](#) или [SetTrAutoMode\(\)](#)

[SetTrBufBitReady\(\)](#)

4.4 Набор функций для чтения принятых данных

[GetNBlockRawDMA\(\);](#)

[ReadDMAChDataBlocks\(\);](#)

4.5 Набор функций для инициализации модуля для работы в режиме КШ

SetMode();	Выбрать режим КШ
BusSelection();	
WorkEnable();	При инициализации модуля, разрешать работу следует последним действием.
EnableDMA();	
SetBC_TCK();	

4.6 Набор функций для программирования работы КШ для начинающего пользователя

[WriteBCFrame\(\)](#) запись фреймов в память КШ
[SetBCSTRT\(\)](#) запуск выполнения фреймов из памяти КШ
[SetGPF2\(\)](#) регулировка цикличности выполнения фреймов

Обратите внимание!!!

Для **передачи** данных (КШ - ОУ) во входном параметре типа WR_MIL1553UDx функции WriteBCFrame следующие поля должны иметь значения:

nFormat = 1 - Формат 1, согласно ГОСТ Р 52070-2003;
nRcvTr = 0 - Бит признака приёма передачи сообщений

Для **приёма** данных (ОУ - КШ) во входном параметре типа WR_MIL1553UDx функции WriteBCFrame следующие поля должны иметь значения:

nFormat = 2 - Формат 2, согласно ГОСТ Р 52070-2003;
nRcvTr = 1 - Бит признака приёма передачи сообщений

Не обязательно после каждого вызова WriteBCFrame запускать выполнение работы скриптов. С помощью параметра nCount Вы можете выстроить собственный алгоритм работы КШ.

Например:

1. Вы хотите только циклично передавать данные в ОУ:

- вызов WriteBCFrame с параметром nCount равным нулю;
- вызов SetGPF2 установка флага GPF2
- вызов SetBCSTRT - запуск выполнения фрейма;
- модуль PCIe-1553UDx начинает самостоятельно циклично (с периодом nRptTime) выполнять фрейм, записанный с помощью функции WriteBCFrame (передавать данные в ОУ), пока Вы не сбросите флаг с помощью функции SetGPF2
- вызов SetGPF2 – сброс флага GPF2. Модуль PCIe-1553UDx выполнит все фреймы до конца итерации цикла и остановит выполнение.

2. Вы хотите циклично передавать в ОУ и принимать данные из него или любого другого ОУ:

- вызов WriteBCFrame с параметром nCount равным нулю и остальными параметрами для передачи данных в ОУ;
- вызов WriteBCFrame с параметром nCount равным единице и остальными параметрами для приёма данных из ОУ;
- вызов SetGPF2 установка флага GPF2
- вызов SetBCSTRT - запуск выполнения фреймов;
- модуль PCIe-1553UDx начинает самостоятельно циклично (с периодом nRptTime) выполнять фреймы, записанные с помощью функции WriteBCFrame, пока Вы не сбросите флаг с помощью функции SetGPF2
- вызов SetGPF2 – сброс флага GPF2. Модуль PCIe-1553UDx выполнит все фреймы до конца итерации цикла и остановит выполнение.

4.7 Набор функций программирования работы КШ для продвинутого пользователя

[WriteBar\(\)](#); записать инструкции в область памяти BC_INSTR_RAM* (адреса 0x18000, 0x28000, 0x38000, 0x48000);

[WriteBar\(\)](#); записать операции в область памяти BC_OPERATION_RAM** (адреса 0x1C000, 0x2C000, 0x3C000, 0x4C000);

[WriteBar\(\)](#); записать данные в область памяти BC_DATA_RAM*** (адреса 0x10000, 0x20000, 0x30000, 0x40000);

[SetBCSTRT\(nCh, true\)](#); начать выполнение инструкций.

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

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

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

5. Обновление библиотеки

Версия библиотеки	Дата	Изменение
1.0	18.11.2017	- Библиотека создана.

6. Обновление руководства

Версия руководства	Дата	Изменения
1.0	18.11.2017	- Документ создан