



## **Руководство (v1.2)**

# **По работе с динамической библиотекой модулей «PCIe – 1553UDx» «mPCIe – 1553UD2» «XMC – 1553UDx»**

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

Для динамических библиотек версии 1.0

**ОС WINDOWS**



**25.01.2017**

**ООО «Новомар» 2017 г.**

## Оглавление

1.	Подключение библиотеки к разрабатываемому проекту. ....	3
1.1	Расшифровка названия библиотеки. ....	4
2.	Функции для работы с драйвером.....	5
3.	Список доступных функций по версиям библиотек .....	6
3.1	Описание функций библиотеки .....	7
3.1.1	UINT32 DEVICE_OPEN.....	7
3.1.2	void DEVICE_CLOSE .....	7
3.1.3	UINT32 INIT_MIL1553.....	8
3.1.4	UINT32 MIL1553_START_STOP .....	10
3.1.5	UINT32 READ_MIL1553.....	11
3.1.6	UINT32 WR_MIL1553_FRM.....	13
3.1.7	UINT32 GetDeviceType.....	15
4.	Обновление библиотеки.....	16
5.	Обновление руководства.....	16

**Данная библиотека поддерживает следующие модули:**

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

**Модуль «mPCIe-1553UD» данной библиотекой не поддерживается. Для «mPCIe-1553UD» используется библиотека:**

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

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

## 1. Подключение библиотеки к разрабатываемому проекту.

Для того, чтобы получить доступ к функциям драйвера, в разрабатываемый проект следует включить динамическую библиотеку “*Mil\_1553UDx.dll*”.

В зависимости данной библиотеки входят системные файлы “*msvcr90.dll*”, “*msvcp90.dll*”, “*kernel32.dll*”. Вы можете найти их в архиве с библиотекой.

Библиотеку можно добавлять в проекты и среды разработок, такие как: Visual C++(2005, 2008), C++ Builder 6, Qt, Borland C++ 5 и т.д.

Пример подключения библиотеки к среде разработки ПО “Microsoft Visual Studio”:

Для этого в настройках Microsoft Visual Studio выберите путь:

Project properties -> Configuration properties -> Linker -> Input -> Additional Dependencies

Укажите имя статической библиотеки-связки (*Mil\_1553UDx.lib*).

Также следует указать компилятору путь к заголовочным файлам библиотеки. Для этого выберите путь:

Project properties -> Configuration properties -> C/C++ -> Additional Include Directories

Укажите путь к папке с заголовочными файлами библиотеки.

По пути Project properties -> Configuration properties -> Linker -> Additional Library Directories следует указать путь к статической библиотеке-связке “*Mil\_1553UDx.lib*”.

Саму динамическую библиотеку “*Mil\_1553UDx.dll*” следует положить в папку с исполняемым файлом вашего разрабатываемого проекта (.exe).

Добавьте в вашем проекте строки:

```
#include "windows.h"
```

```
#include "MIL1553UDxDevice.h"
```

Библиотека подключена к проекту.

Импорт функций см. в примере, приложенном к файлам библиотеки.

Теперь вы можете вызывать функции библиотеки.

## 1.1 Расшифровка названия библиотеки.

<i>Mil_1553UDx.dll</i>		
<i>1</i>	<i>2</i>	<i>3</i>

1	<i>Mil_1553UDx</i>	Название библиотеки
3	<i>dll</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 (переменная NO\_ERR).

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

*TestPCIe1553UDxdll.cpp*

### Пример:

Функция библиотеки:

**UINT32 GetDeviceType()**

Её пример вызова:

```
printf("PCIe-1553UDx has %d channels.\n", lpGET_DEVICE_TYPE(hDevice));
```

Название тестовой функции GetDeviceType (). По этому названию, код данной функции можно найти в файле *TestPCIe1553UDxdll.cpp*

**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, доступных в библиотеке версии до 1.0 и ниже	
<a href="#">DEVICE_OPEN</a>	Открытие устройства
<a href="#">DEVICE_CLOSE</a>	Закрытие устройства
<a href="#">INIT_MIL1553</a>	Инициализация устройства
<a href="#">MIL1553_START_STOP</a>	Запуск и остановка работы устройства
<a href="#">READ_MIL1553</a>	Чтение полученных данных из буфера DMA.
<a href="#">WR_MIL1553_FRM</a>	Запись инструкций, операций и запуск их выполнения
<a href="#">GetDeviceType</a>	Тип устройства (1, 2 и 4 канала)

### 3.1 Описание функций библиотеки

#### 3.1.1 UINT32 DEVICE\_OPEN (UINT32 DEV\_NUM, HANDLE \*m\_hDevice)

##### **ВНИМАНИЕ**

Функция Open открывает модули «mPCIe-1553UD2», «PCIe-1553UDx» и «XMC-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 модуля. Для открытия модуля в функцию следует передать его индекс и указатель на переменную в которой будет храниться идентификатор устройства. Если модуль в системе один, то его индекс равен '0'.

Если нужно работать с несколькими устройствами одновременно, то следует открывать устройство с разными индексами и запоминать полученные идентификаторы в разные переменные:

Входные данные

Переменная	Описание	Диапазон значений
UINT32 DEV_NUM	Индекс устройства	
HANDLE m_hDevice	Указатель на переменную, в которую будет записано значение идентификатора	1-4

Расшифровка кодов ошибок

код	Расшифровка
DEVICE_NOT_OPEN	Устройства с индексом DEV_NUM не найдено.
TYPE_NOT_READ	Ошибка на этапе чтения типа устройства.

Пример:

Открытие устройства с индексом 0  
lpDEVICE\_OPEN(0, &hDevice)

Открытие устройства с индексом 1  
lpDEVICE\_OPEN(1, &hDevice1)

Функцию Open следует вызывать с разными индексами (0, 1, 2, 3), от разных экземпляров:  
m\_Device0.Open(0); m\_Device1.Open(1);

#### 3.1.2 void DEVICE\_CLOSE(HANDLE m\_hDevice)

Закрытие устройства. Для закрытия уст-ва функцию Close следует вызывать от того же экземпляра класса что и функцию Open.

Пример:

lpDEVICE\_OPEN(0, &hDevice)  
lpDEVICE\_CLOSE(hDevice);

### 3.1.3 UINT32 INIT\_MIL1553 (HANDLE m\_hDevice, UINT32 nCh, UINT32 nMode, UINT32 NoResp)

#### Назначение:

Инициализация канала nCh для начала работы.

#### Действие:

Функция записывает выбранный режим работы (**nMode**) в биты 0-1 регистра CTRL\_REG\_PCI\*.

Разрешает работу по обеим шинам, записывая единицы в биты 2-3 этого же регистра.

Записывает значение таймера в биты 19-20 регистра BC\_CONF\_REG\_PCI\*\*.

Разрешает работу DMA, записывая единицу в 0 бит регистра DMA\_DATA\_BASE (работа DMA разрешается для всех каналов одновременно).

\* См. Раздел 5.1.5 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

\*\*См. Раздел 7.2.1 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

#### Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
HANDLE m_hDevice	Идентификатор устройства	
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
UINT32 nMode	Режим работы	<b>BM_A</b> - адресуемый монитор шины <b>BC</b> - контроллер шины <b>RT</b> - оконечное устройство <b>BM_NA</b> - не адресуемый монитор шины
UINT32 NoResp	Выбранное значение таймаута.	17, 60, 85, 110 мкс

Расшифровка кодов ошибок на следующей странице.



## Расшифровка кодов ошибок:

код	Расшифровка
ERR_HANDLER	Не найдено открытого устройства с идентификатором m_hDevice
ERR_CH	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).
ERR_MODE	Значение параметра nMode не попадает в диапазон разрешённых значений
ERR_NORESP	Значение параметра NoResp не входит в диапазон допустимых значений.
ERR_SET_MODE	Ошибка при попытке записать режим работы
ERR_SET_RT_T1_RCVCK	Ошибка при попытке записать таймаут
ERR_BUS_SELECTION	Ошибка при попытке записи выбора шины
ERR_ENABLE_DMA	Ошибка при попытке разрешения работы DMA

## Пример вызова:

```

UINT32 nErr, nCh = 2, NoResp = 110;
nErr = lpINIT_MIL1553(hDevice1, nCh, BC, NoResp);
if (nErr != 1){
    printf("Error initialization device. Err = %d\n");
    scanf("%x");
    return 0;
}
else
    printf("Device has been initialized.\n");

```

**3.1.4 UINT32 MIL1553\_START\_STOP(HANDLE m\_hDevice, UINT32 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» и «mPCIe-1553UD2».

**Примечание:**

-

Входные данные

Переменная	Описание	Диапазон значений
HANDLE m_hDevice	Идентификатор устройства	
UINT32 nCh	Номер канала, размер буфера которого Вы хотите узнать	1-4
BOOL bFlag	Значение TRUE означает разрешение работы, FALSE запрет.	

Расшифровка кодов ошибок

код	Расшифровка
ERR_HANDLER	Не найдено открытого устройства с идентификатором m_hDevice
ERR_CH	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).
ERR_WRK_ENBL_RD	Не удалось прочить данные из регистра CTRL_REG_PCI
ERR_WRK_ENBL_WR	Не удалось записать данные в регистр CTRL_REG_PCI.

Пример вызова:

```

UINT32 nErr, nCh = 2;
nErr = lpSTART_STOP(hDevice1, nCh, TRUE);
if (nErr != 1){
    printf("Error start working device. Err = %d\n");
    scanf("%x");
    return 0;
}
else
    printf("Device has been started.\n");

```

### 3.1.5 UINT32 READ\_MIL1553 (HANDLE m\_hDevice, UINT32 nCh, UINT32 nAddr, UINT32 nSubAddr, UINT32 nRcvTr, UINT32 \*nStatus, void \*pData)

#### Назначение:

Чтение фрейма из буфера DMA канала nCh.

#### Действие:

Функция сравнивает значение регистрового и программного счётчика DMA. Если они не равны, вычитывает один новый фрейм из буфера DMA.

Если фрейм был получен с адреса nAddr, подадреса nSubAddr и имеет признак приёма/передачи nRcvTr, то он будет записан в переменную pData. В противном случае фрейм будет удалён. Указатель буфера DMA сдвигается на соответствующую количеству прочитанных данных позицию.

Также функция проверяет фрейм на наличие ошибок:

- если ошибок нет, то значение переменной nStatus равно нулю;
- если в слове ошибок есть биты ошибок, то значение переменной равно единице;
- если в первом двойном слове есть биты ошибок, то значение переменной равно двум;
- если в первом двойном слове не установлен бит MSG\_GOOD (Сообщение передано успешно), то значение переменной равно трём.

#### Примечание:

-

Входные данные

Переменная	Описание	Диапазон значений
HANDLE m_hDevice	Идентификатор устройства	
UINT32 nCh	Указатель на переменную в которую будет считано значение	1-4
UINT32 nAddr	Номер адреса оконечного устройства	1-31
UINT32 nSubAddr	Номер подадреса	1-31
UINT32 nRcvTr	Признак приёма/передачи	0 - передача 1 - приём
UINT32 *nStatus	Указатель на переменную, в которую будет записан статус ошибок	
void *pData	Указатель на блок памяти размером в 128 байт в который будут считаны данные	

## Расшифровка кодов ошибок

код	Расшифровка
ERR_HANDLER	Не найдено открытого устройства с идентификатором m_hDevice
ERR_CH	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).
ERR_RD_ADDR	Значение параметра nAddr не попадает в диапазон разрешённых значений
ERR_RD_SADDR	Значение параметра nSubAddr не попадает в диапазон разрешённых значений
ERR_RD_RCV_TR	Значение параметра nRcvTr не попадает в диапазон разрешённых значений
RD_MIL_NO_DATA	В буфере DMA нет новых фреймов
ERR_RD_COUNT	Ошибка на этапе сравнения программного и регистрового счётчика фреймов в буфере DMA
ERR_RD_DATA	Ошибка на этапе чтения фрейма из буфера DMA
ERR_RD_NO_DATA	В буфере DMA нет новых фреймов (возможно вычитало другое приложение)
ERR_RD_ADDR_MTCH	Адрес ОУ не совпадает с запрашиваем параметром (nAddr)
ERR_RD_SADDR_MTCH	Подадрес ОУ не совпадает с запрашиваем параметром (nSubAddr)
ERR_RD_RCV_TR_MTCH	Признак приёма/передачи не совпадает с запрашиваемым параметром (nRcvTr)

Пример вызова:

```
UINT32 nAddr=2, nSubAddr=1, nRcvTr=0;
```

```
nErr = lpRD_FRM(hDevice1, nCh, nAddr, nSubAddr, nRcvTr,
&nStatus, pData);
if (nErr != 1){
    printf("Error reading DMA data. Err = %d\n");
    scanf("%x");
    return 0;
}
else{
    if (nStatus)
        printf("Errors in DMA data block. Status = %d \n", nStatus);
    else{
        printf("DMA data block \n");
        for(i=0; i<32; i++)
            printf("0x%x\n", pData[i]);
    }
}
```

**3.1.6 UINT32 WR\_MIL1553\_FRM(HANDLE m\_hDevice, UINT32 nCh,****WR\_MIL1553UDx \*Str, UINT32 \*pData)****Назначение:**

В зависимости от входных параметров, функция составляет инструкции, операции и записывает их. При необходимости также записывает данные для передачи.

Также функция запускает выполнение инструкций.

**Действие:**

Инструкции записываются в область памяти BC\_INSTR\_RAM<sup>1</sup>.

Операции записываются в область памяти BC\_OPERATION\_RAM<sup>2</sup>.

Данные, при необходимости, записываются в область памяти BC\_DATA\_RAM<sup>3</sup>.

Старт выполнения инструкций запускается установкой в единице бита BCSTRT регистра BC\_CONF\_REG\_PCI<sup>4</sup>.

<sup>1</sup>См. Раздел 7.1.1 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

<sup>2</sup>См. Раздел 7.1.2 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

<sup>3</sup>См. Раздел 7.1.3 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

<sup>4</sup>См. Раздел 7.2.1 документа «Руководство по программированию модуля «PCIe-1553UDx», «XMC-1553UDx» и «mPCIe-1553UD2».

**Примечание:** -

Входные данные

Переменная	Описание	Диапазон значений
HANDLE m_hDevice	Идентификатор устройства	
UINT32 nCh	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).	1-4
Ссылка на структуру типа <b>WR_MIL1553UDx</b>		
UINT32 nAddr	Адрес ОУ	1-31
UINT32 nSubAddr	Поадрес ОУ	1-31
UINT32 nRcvTr	Признак приёма/передачи	0 – передача 1 – приём
UINT32 nDWCCount	Количество шестнадцатитбитных слов данных	0-32
UINT32 nRptTime	Период повтора передачи фрейма	0-0xFFFF
UINT32 nRptCount	Количество повторов передачи фрема	0-0xFFFF
UINT32 nFormat	Формат передачи	0-10 1 – на передачу, 2 – на приём.
Конец структуры		
UINT32 *pData	Указатель на блок памяти с данными, которые будут переданы в линию. Размер данных – до 64 байт включительно.	

## Расшифровка кодов ошибок

код	Расшифровка
ERR_HANDLER	Не найдено открытого устройства с идентификатором m_hDevice
ERR_CH	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).
ERR_WR_RPT_COUNT	Значение параметра nRptCount не попадает в диапазон разрешённых значений
ERR_WR_RPT_TIME	Значение параметра nRptTime не попадает в диапазон разрешённых значений
ERR_WR_ADDR	Значение параметра nAddr не попадает в диапазон разрешённых значений
ERR_WR_SADDR	Значение параметра nSubAddr не попадает в диапазон разрешённых значений
ERR_WR_DW_COUNT	Значение параметра nDWCount не попадает в диапазон разрешённых значений
ERR_WR_RCV_TR	Значение параметра nRcvTr не попадает в диапазон разрешённых значений
ERR_WR_FRMT	Значение параметра nFormat не попадает в диапазон разрешённых значений
ERR_WR_INSTR	Ошибка драйвера при попытке записи инстукций.
ERR_WR_OPER	Ошибка драйвера при попытке записи операций.
ERR_WR_DATA	Ошибка драйвера при попытке записи данных.

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;
    UINT32 pData[32], I;

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

    nErr = lpWR_FRM(hDevice1, nCh, Str, pData);
    if (nErr != 1){
        printf("Error sending data. Err = %d\n");
        scanf("%x");
        return 0;
    }
    else
        printf("Data has been send.\n");

```

**3.1.7 UINT32 GetDeviceType (HANDLE m\_hDevice)****Назначение:**

Функция возвращает тип модуля:

- 1 – «PCIe-1553UD»
- 2 – «PCIe-1553UD2»
- 4 – «PCIe-1553UD4»
- 8 – «mPCIe-1553UD2»
- 16 – «XMC-1553UD2»
- 32 – «XMC-1553UD4»

**Действие:**

Функция читает значение типа модуля из регистрового пространства BAR.

**Примечание:**

-

Входные данные

Переменная	Описание	Диапазон значений
HANDLE m_hDevice	Идентификатор устройства	

Расшифровка кодов ошибок

код	Расшифровка
ERR_HANDLER	Не найдено открытого устройства с идентификатором m_hDevice

Пример вызова:

```
printf("Device type is %d .\n", lpGET_DEVICE_TYPE(hDevice));
```

#### 4. Обновление библиотеки

Версия библиотеки	Дата	Изменение
1.0	15.07.2016	- Библиотека создана.

#### 5. Обновление руководства

Версия руководства	Дата	Изменения
1.0	15.07.2016	- Документ создан
1.1	18.11.2016	Добавлено описание поддержки модуля «mPCIe-1553UD2»
1.2	25.01.2017	Добавлено описание поддержки модулей «XMC-1553UD»