

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**  
**Alder SDK**  
**[прием-передача факсов]**



## Уважаемый покупатель!

Вы приобрели СТИ-плату **ОЛЬХА-14**, созданную группой компаний **АГАТ** для решения широкого круга задач компьютерной телефонии.

Платы **ОЛЬХА-14** – идеальное решение для приложений компьютерной телефонии. Системы компьютерной телефонии на базе плат **ОЛЬХА-14** обладают высокой степенью гибкости и масштабируемости. Это открывает широкие возможности как для разработчиков программного обеспечения сетей и систем связи. Новое семейство плат **ОЛЬХА-14** является продолжением ранее разработанного поколения плат **ОЛЬХА-9/10** группы компании **АГАТ**. СТИ-платы **ОЛЬХА-14** полностью программно совместимы с другими платами семейства **ОЛЬХА**.

Ваши отзывы и предложения по данному продукту просим направлять по следующему адресу:

Россия, 129329, г. Москва, ул. Ивовая, дом 1,  
корп. 1, 3 этаж  
Телефон/факс: (495) 799-9069 (многоканальный),  
E-mail: [info@agatrt.ru](mailto:info@agatrt.ru),  
Web-сайт: <http://www.agatrt.ru>

**Copyright © ООО КБ АГАТ-РТ, 2007**

Все права защищены, включая право на полное или частичное воспроизведение, хранение в поисковых системах или передачу, в какой бы то ни было форме, любыми способами - электронными, механическими, с помощью фотокопирования, записи или иными.

Версия документа: 1.1

## Содержание

<b>Введение</b> .....	<b>4</b>
Назначение документа .....	4
Используемые обозначения .....	4
Ссылки на другие документы .....	4
Назначение документов .....	5
О службе технической поддержки .....	5
<b>Общая информация</b> .....	<b>6</b>
Классификация факс-ресурсов .....	6
Факс-ресурсы в платах серии ОЛЬХА-9 .....	6
Факс-ресурсы в платах серии ОЛЬХА-14 .....	7
Факс-ресурсы в AlderSDK .....	7
<b>Прием факса</b> .....	<b>8</b>
Алгоритм приема факса .....	8
Используемые функции .....	9
sdkFaxInit .....	9
sdkFaxReceive .....	10
sdkFaxGetStatus .....	11
sdkFaxCancelReceive .....	12
sdkFaxLeave .....	13
sdkFax2Bmp .....	14
Пример sdkRecFax .....	15
Общие сведения .....	15
Функция Main() .....	15
Вспомогательные функции .....	20
<b>Передача факса</b> .....	<b>21</b>
Алгоритм передачи факса .....	21
Используемые функции .....	22
sdkBmp2Fax .....	22
sdkFaxInit .....	25
sdkFaxSend .....	26
sdkFaxGetStatus .....	27
sdkFaxCancelSend .....	28
sdkFaxLeave .....	29
Пример sdkSendFax .....	30
Общие сведения .....	30
Функция Main() .....	30
Вспомогательные функции .....	36
<b>ПРИЛОЖЕНИЕ</b> .....	<b>37</b>
Коды завершения факс-сеанса .....	37

# Введение

## Назначение документа

Данный документ содержит описание функций **Alder SDK** для работы с факс-ресурсами плат **ОЛЬХА-14**. Документ предназначен для разработчиков приложений на базе компьютерных плат **ОЛЬХА-14** с помощью библиотеки **Alder SDK**.

## Используемые обозначения

### Внимание!



Так помечается информация, на которую следует обратить особое внимание. Это может быть описание какого-либо требования для выполнения описываемой задачи, важная информация по использованию и т.д.

### Полезно!



Так помечается дополнительная информация, которая может быть полезна пользователю. Это может быть ссылка на какой-либо документ или раздел документа, рекомендация по использованию и т.д.

## Ссылки на другие документы

### Полезно!



Последние версии документов, входящих в комплект поставки плат **ОЛЬХА-14**, Вы всегда можете загрузить с официального Web-сайта компании

<http://www.agatrt.ru>

либо запросить по электронной почте в службе технической поддержки

[support@agatrt.ru](mailto:support@agatrt.ru)

При работе с СТИ-платой **ОЛЬХА-14** Вам могут быть полезны следующие документы:

- Паспорт СТИ-плат ОЛЬХА-14;*
- Руководство по эксплуатации СТИ-платы ОЛЬХА-14;*
- Руководство пользователя драйвера Alder14 для плат компьютерной телефонии ОЛЬХА-14 (для MS Windows /XP);*
- Руководство пользователя Alder Software Development Kit – средства разработки Windows-приложений для плат ОЛЬХА-14 (AlderSDK);*

## Назначение документов

<b>Паспорт</b>	- описание технических характеристик; - описание функциональных характеристик
<b>Руководство по эксплуатации</b>	- описание установки платы; - описание платы ОЛЬХА-14.
<b>Руководство пользователя. Драйвер для плат Ольха-14</b>	- описание функций драйвера; - описание работы драйвера с аналоговыми мезонинами / каналами.
<b>Руководство пользователя. AlderSDK</b>	- описание функций библиотеки AlderSDK.dll; - перечень возможных ошибок при работе с библиотекой AlderSDK.dll.

## О службе технической поддержки

Для всех пользователей продукции группы компаний **АГАТ** работает «горячая линия» технической поддержки. Наши специалисты помогут Вам решить все возникающие вопросы на этапах внедрения и эксплуатации.

По всем возникающим вопросам следует обращаться по адресу:

**support@agatrt.ru,**

при этом необходимо указать наименование приобретенного продукта, дату покупки, серийный номер платы, конфигурацию платы.

Кроме этого, можно также заполнить форму-запрос на официальном Web-сайте компании по адресу:

**[http://www.agatrt.ru/support\\_form.html](http://www.agatrt.ru/support_form.html)**

## Общая информация

### Классификация факс-ресурсов

Как известно, факс-аппараты позволяют пересылать графическую информацию, используя обычные телефонные линии общего пользования. Простота использования и снижение стоимости факс-аппаратов в последние десятилетия сделали их необычайно популярными.

Факс-ресурсы в платах компьютерной телефонии позволяют существенно расширить возможности программного обеспечения и предоставить пользователям дополнительные услуги по приему и передаче графической информации.

В компьютерной телефонии факс-ресурсы можно классифицировать по следующему признаку:

1. **Аппаратные факс-ресурсы.** Это факс-ресурсы, реализуемые за счет сторонних вычислительных мощностей, не принадлежащих ПК. Например – это аппаратные факс-модемы, присоединяемые к СОМ-порту или USB-порту. Преимущество их очевидно – экономия ресурсов ПК. К недостаткам можно отнести только более высокую стоимость.
2. **Программные факс-ресурсы.** Это ресурсы, реализуемые за счет вычислительной мощности центрального процессора ПК. Плюсы и минусы данного решения точно противоположны аппаратным решениям. Плюсы – низкая стоимость, минусы – высокие требования к вычислительным ресурсам ПК.

При любой реализации факс-ресурсов их объединяет одно – им необходим модуль аппаратного сопряжения с телефонной линией.

### Факс-ресурсы в платах серии ОЛЬХА-9

В платах серии **ОЛЬХА-9** факс-ресурсы были реализованы программно. Сами платы серии **ОЛЬХА-9** выступали только в роли модуля аппаратного сопряжения с телефонной линией. Собственно факс-ресурсы были реализованы в библиотеке **OlhaFullFax.dll**. Для получения функционала факса необходимо было использовать программный интерфейс этой библиотеки.

Во всех программных решениях в этой области имеется очевидный недостаток – одновременно работающих факс-ресурсов определяется вычислительной мощностью центрального процессора ПК. Для малоканальных решений такой подход является вполне приемлемым, так как мощность процессоров в современных ПК достигла весьма существенных высот.

Для решения типа **OlhaFullFax**, надо признать, характерна универсальность. **OlhaFullFax** работает на всех платах серии **ОЛЬХА-2** и серии **ОЛЬХА-9**. Доступна работа, как на аналоговых линиях, так и на цифровых, например E1.

Но будущее за факс-ресурсами аппаратной реализации.

## Факс-ресурсы в платах серии **ОЛЬХА-14**

В платах серии **ОЛЬХА-14** предложена именно аппаратная реализация. То есть факс-ресурсы интегрированы в саму плату и реализованы исключительно за счет вычислительных мощностей самой платы **ОЛЬХА-14**.

Плата **ОЛЬХА-14** выступает и как модуль сопряжения с телефонной линией, и как носитель факс-ресурсов. Реализация факс-ресурсов на **ОЛЬХА-14** такова, что использование факс-ресурсов возможно как для аналоговых, так и для цифровых линий. Факс-ресурсы в плате **ОЛЬХА-14** не привязаны жестко к отдельно заданной линии, а могут коммутироваться на любую линию, обслуживаемую платой. Впоследствии будет возможна коммутация факс ресурсов и по стандартизированной шине CTBus на другие устройства компьютерной телефонии, поддерживающие эту шину.

## Факс-ресурсы в AlderSDK

Факс-ресурсы, как таковые, от пользователя **AlderSDK** скрыты. Цель **AlderSDK** – скрыть от пользователя многие нюансы реализации ресурсов плат компьютерной телефонии. При реализации управления факс-ресурсами в **AlderSDK** при использовании плат **ОЛЬХА-14** преследовалась единственная цель – это как можно большая совместимость с уже написанными программами для плат **ОЛЬХА-9**. Поэтому, в целом, интерфейс приема-передачи факсов не изменился, но появились определенные нюансы.

Поскольку **AlderSDK** имеет функции уровня драйвера, то возможно управление факс-ресурсами «напрямую».

# Прием факса

## Алгоритм приема факса

Чтобы начать прием факса в процессе сеанса связи, необходимо выполнить следующие действия:

1. Вызвать функцию [sdkFaxInit](#).
2. Вызвать функцию [sdkFaxReceive](#), которая начнет прием факса.
3. Регулярно вызывать функцию [sdkFaxGetStatus](#) для того, чтобы дождаться окончания факс-сессии (поле **RecProcessStatus** структуры **sdkFaxStatus** станет равно **FaxReceiveOff**).
4. Вызвать [sdkFaxCancelReceive](#).
5. Вызвать [sdkFaxLeave](#).
6. Преобразовать hfm-файл в файл (файлы) формата BMP с помощью функции [sdkFax2Bmp](#).

Если в процессе факс-сессии необходимо прервать прием факса, то необходимо сразу перейти к п.4, игнорируя п.3.

## Используемые функции

### sdkFaxInit

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD sdkFaxInit( char *pszFaxKey, DWORD *lpdwMajorVersion, DWORD *lpdwMinorVersion);</code>	
<b>Входные параметры</b>	<code>*pszFaxKey</code>	указатель на строку, содержащую ключ для факс-библиотеки.
	<code>*lpdwMajorVersion</code>	указатель на 4-х байтовое поле для возврата старшей цифры версии факс-библиотеки.
	<code>*lpdwMinorVersion</code>	указатель на 4-х байтовое поле для возврата младшей цифры версии факс-библиотеки.

### Описание

Функция вызывается при совместном использовании плат **ОЛЬХА-14** и других плат серии **ОЛЬХА** (**ОЛЬХА-9** или **ОЛЬХА-2**).

В случае совместного использования плат **ОЛЬХА-14** и плат серии **ОЛЬХА-9** или **ОЛЬХА-2**, входные параметры будут использоваться для инициализации библиотеки **OlhaFullFax**. В этом случае рекомендуется поместить библиотеку **OlhaFullFax** в рабочий каталог программы.

Если же плата **ОЛЬХА-14** используется без плат других серий, то входные параметры игнорируются. В этом случае библиотека **OlhaFullFax** не инициализируется, во внимание не принимается и может отсутствовать в текущем каталоге.

## sdkFaxReceive

---

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD <b>sdkFaxReceive</b>( DWORD <i>dwChannelId</i>, char *<i>pszFile</i>);</code>	
<b>Входные параметры</b>	<i>dwChannelId</i>	идентификатор канал, полученный при помощи функции <b>sdkOpenChannel</b> .
	* <i>pszFile</i>	указатель на имя файла с факс-данными для передачи.

---

### Описание

Функция запускает поток, в котором при помощи функций асинхронного ввода-вывода выполняется прием факса. После завершения приема факса, поток самостоятельно завершает работу.

Если канал принадлежит плате **ОЛЬХА-9**, то прием факса будет происходить при помощи **OlhaFullFax.dll**. Принятый факс библиотека запишет в файл с именем **pszFile**. Формат файла будет соответствовать формату файла библиотеки **OlhaFullFax.dll**.

Если же канал принадлежит плате **ОЛЬХА-14**, то для приема факса будет использоваться встроенный факс-ресурс платы. Принятый факс будет записан в файл с именем **pszFile**.

**sdkFaxGetStatus**

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD <b>sdkFaxGetStatus</b> (DWORD <i>dwChannelId</i>, sdkFaxStatus *<i>lpFaxStatus</i>);</code>	
<b>Входные параметры</b>	<i>dwChannelId</i>	идентификатор канал, полученный при помощи функции <b>sdkOpenChannel</b> .
	* <i>lpFaxStatus</i>	указатель на структуру типа <b>sdkFaxStatus</b> для сохранения состояния факс-сессии по каналу.

**Описание**

При помощи этой функции, можно контролировать завершение факс-сеанса для принятия последующего решения.

Формат структуры **sdkFaxStatus** приведен ниже. Описаны не все поля структуры, а только те, которые используются для работы с факс-ресурсом платы **ОЛЬХА-14**.

Наименование	Тип	Описание
<b>RecProcessStatus</b>	DWORD	Состояние передачи факса. Если факс в процессе приема, то поле содержит значение <b>FaxReceive9600</b> . Если факс-сессия закончена, то в поле будет содержаться значение <b>FaxReceiveOff</b> .
<b>RecFinalStatus</b>	DWORD	Код завершения факс-сессии (см. <a href="#">Коды завершения факс-сеанса</a> ). Поле действительно, если <b>RecProcessStatus=FaxReceiveOff</b> .
<b>SendFinalStatus</b>	DWORD	Счетчик успешно принятых страниц.

## sdkFaxCancelReceive

---

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD</code> <b>sdkFaxCancelReceive</b> ( <code>DWORD dwChannelId</code> );
<b>Входные параметры</b>	<code>dwChannelId</code> идентификатор канал, полученный при помощи функции <b>sdkOpenChannel</b> .
<b>Выходные параметры</b>	Код завершения факс-сессии <b>117 (FAXMODEM_STATUS_CANCELLED)</b> .

---

### Описание

Функция немедленно завершает прием факса. Поток, который принимал факс, немедленно останавливается и прекращает свое существование.

#### Полезно!

---



Рекомендуется в конце каждой факс-сессии после приема факса вызывать функцию *sdkFaxCancelReceive*.

---

#### Внимание!

---



На удаленном факсе, вероятнее всего, факс-сессия завершится с ошибкой.

---

## sdkFaxLeave

---

**Формат  
вызова**                    \_declspec(dllimport) DWORD **sdkFaxLeave**();

**Входные  
параметры**                Отсутствуют

---

### Описание

Функция завершает работу с факс-ресурсами. Для плат серии **ОЛЬХА-9** происходит попытка выгрузить библиотеку **OlhaFullFax.dll** из памяти. Для плат **ОЛЬХА-14** никаких действий не производится.

**sdkFax2Bmp**

<b>Формат вызова</b>	_cdeclspec(dllimport) DWORD <b>sdkFax2Bmp</b> ( char *pszDestFile, char *pszSrcFile);	
<b>Входные параметры</b>	*pszDestFile	указатель на имя результата BMP-файла.
	*pszSrcFile	Указатель на имя исходного hfm-файла.

**Описание**

Функция осуществляет преобразование hfm-файл, полученного в результате приема факса, в файл формата BMP.

Результатом приема факса является hfm-файл. Чтобы получить файл с изображением, с которым работают почти все программы, ориентированные на работу с графикой, необходимо преобразовать hfm-файл в файл (файлы) формата BMP.

**AlderSDK** проверяет формат указанного файла с именем **pszSrcFile**. Если файл имеет формат hfm, то преобразование данных производит библиотека **AFIC.DLL**. Если файл имеет отличный от hfm-формата, то преобразование будет происходить при помощи библиотеки **OlhaFullFax.dll**.

**Внимание!**

Для корректного преобразования файлов необходимо, чтобы библиотека **AFIC.DLL** был доступна в текущем каталоге.

В исходном hfm-файле может содержаться не одна факс-страница, а несколько. Библиотека **AFIC** размещает каждую страницу в отдельном BMP-файле. Для каждого BMP-файла имя формируется следующим образом:

**pszSrcFile -Page-NN.bmp,**

где **pszSrcFile** – имя файла, заданное пользователем в поле **pszSrcFile**; **NN** – номер декодированной страницы.

Таким образом, при приеме трех факс-страниц и последующем декодировании hfm-файла должны получиться три BMP-файла. Если в поле **pszSrcFile** было записано «**test.bmp**», то получившиеся BMP-файлы будут иметь имена **test-Page-1.bmp**, **test-Page-2.bmp**, **test-Page-3.bmp**.

## Пример sdkRecFax

Все вышеизложенное продемонстрировано в примере `sdkRecFax`. Пример принимает факс, записывает его в `hfm`-файл, а затем преобразовывает `hfm`-формат и `BMP`-файлы. Рассмотрим исходные тексты примера подробнее.

### Общие сведения

Пример `sdkRecFax` написан на языке C++ при использовании **MS Visual C++** версии 7.0.

Пример предназначен для демонстрации приема факса только на линиях **FXO**.

Пример оформлен в виде терминальной программы, работающей в текстовом режиме.

#### Внимание!



Исходные тексты программы можно использовать в коммерческих целях. Распространяются исходные тексты свободно и никаких ограничений на использование не имеют. Программа, так же как ее исходные тексты используются пользователем «как есть». Производитель не несет никакой ответственности, если возник ущерб от использования данной программы или ее исходных текстов.

### Функция Main()

Пример оформлен в файле `sdkRecFax.cpp`.

Инициализация библиотеки **AlderSDK**.

```
::printf("Init AlderSDK...\n");
if( sdkInit() != SDK_NOERROR)
    return -1;
::printf("done\n");
```

Получение версии **AlderSDK** и вывод ее на экране.

```
sdkSDKVersion SDKVersion;
sdkGetVersion( &SDKVersion);
::printf("version - %d.%d.%d.%d\n",
        SDKVersion.SDKMajor,
        SDKVersion.SDKMinor,
        SDKVersion.SDKInt1,
        SDKVersion.SDKInt2);
```

## Получение количества установленных плат в ПК.

```
::printf( "Reading boards data...\n");  
if( sdkGetBoardsCount( &BoardsCount)!=SDK_NOERROR)  
    return -1;
```

## Чтение информации об установленных платах в ПК.

```
for( int i= 0; i< ( int)BoardsCount; i++)  
    BoardStatus[i] = new sdkBoardStatus;  
if( sdkUpdateBoards( BoardsCount, BoardStatus)!=SDK_NOERROR)  
    return -1;
```

Проверка наличия установленных каналов с номером 0 на плате. Программа работает только с платой с номером 0 и каналом с номером 0, установленном на этой плате.

```
if( BoardStatus[0]->Lines== 0)  
    return -1;
```

## Резервирование и открытие канала.

```
::printf("Open Channel...\n");  
DWORD ChannelId = BoardStatus[BOARD]->ChannelsId[LINE];  
if( sdkReserveChannel(ChannelId)!=SDK_NOERROR)  
    return -1;  
if( sdkOpenChannel(ChannelId)!=SDK_NOERROR)  
    return -1;  
::printf("done\n");
```

Ожидание звонка на линии **FXO**.

```
if( dwWaitRing( ChannelId)== 0)  
{  
    sdkCloseChannel( ChannelId);  
    sdkFreeChannel( ChannelId);  
    sdkLeave();  
    return -1;  
}
```

Снятие трубки в канале с ожиданием ответа от платы.

```
if( dwSetHook( ChannelId, 1) == 0)
{
    sdkCloseChannel( ChannelId);
    sdkFreeChannel( ChannelId);
    sdkLeave();
    return -1;
}
```

Инициализация факс-ресурсов.

```
DWORD Ver1, Ver2;
if( sdkFaxInit( FAX_KEY, &Ver1, &Ver2) != SDK_NOERROR)
{
    sdkCloseChannel( ChannelId);
    sdkFreeChannel( ChannelId);
    sdkLeave();
    return -1;
}
```

Инициация начала приема факса.

```
if( sdkFaxReceive( ChannelId, FAX_FILE) != SDK_NOERROR)
{
    sdkCloseChannel( ChannelId);
    sdkFreeChannel( ChannelId);
    sdkLeave();
    return -1;
}
```

Организация цикла ожидания завершения факс-сеанса. В цикле проверяется – не захотел ли пользователь немедленно прекратить факс-сессию.

```
sdkFaxStatus FaxStatus;
DWORD dwUserBreak= 0;
::printf( "Press '\ESC\' for exit and '\Space\' for cancel fax session and exit...");
while(1)
{
    switch( dwCheckUserKey())
    {
        case 1:
            dwUserBreak= 1;
            ::printf("User break...\n");
            break;
        case 2:
            ::printf( "User cancel session...\n");
            sdkFaxCancelReceive( ChannelId);
            break;
        default:
            break;
    }
    if( dwUserBreak!= 0)
        break;
    sdkFaxGetStatus( ChannelId, &FaxStatus);
    if( FaxStatus.RecProcessStatus== sdkFaxReceiveOff)
        break;
    ::Sleep(100);
    break;
}
```

Распечатка результатов приема факса.

```
::printf( "FaxFinalStatus=%d PageCount= %d\n",
    FaxStatus.RecFinalStatus,
    FaxStatus.SendFinalStatus);
sdkFaxCancelReceive( ChannelId);
```

### Преобразование hfm-файла в BMP-файл (файлы).

```
::printf( "Decoding Fax...");  
sdkFax2Bmp( BMP_FILE, FAX_FILE);  
::printf( "done\n");
```

### Завершение работы с факс-ресурсами.

```
sdkFaxLeave();
```

### Положить трубку в линии.

```
dwSetHook( ChannelId, 0);
```

### Закрытие канала.

```
printf("Close Channel...\n");  
if( sdkCloseChannel(ChannelId)!=SDK_NOERROR)  
    return -1;  
if( sdkFreeChannel(ChannelId)!=SDK_NOERROR)  
    return -1;  
::printf( "done\n");
```

### Прекращение работы с **AlderSDK**.

```
printf("Leave AlderSDK...\n");  
if( sdkLeave()!=SDK_NOERROR)  
    return -1;  
::printf( "done\n");
```

## Вспомогательные функции

В примере использованы некоторые вспомогательные функции. Вспомогательные функции и их назначение представлены в таблице:

Формат вызова функции	Описание
DWORD <b>dwCheckEscIsPressed();</b>	Функция проверяет, не была ли нажата клавиша <b>ESC</b> . Если <b>ESC</b> была нажата, то возвращает 1, в противном случае 0.
DWORD <b>dwCheckUserKey();</b>	Функция проверяет, какие клавиши нажал пользователь на клавиатуре. Функция возвращает 1, если была нажата клавиша <b>ESC</b> , 2 – если была нажата клавиша <b>Spacebar</b> (пробел), 0 – во всех остальных случаях.
DWORD <b>dwWaitEscKey();</b>	Функция синхронно ожидает нажатия клавиши на клавиатуре. Возвращает 1, если была нажата клавиша <b>ESC</b> , 0 - в противном случае.
DWORD <b>dwWaitRing</b> ( DWORD <i>dwChannelId</i> );	Функция ожидает звонка на линии <b>FXO</b> . Возвращает 1 , если был звонок, 0 – в случае нажатой клавиши <b>ESC</b> .
DWORD <b>dwSetHook</b> ( DWORD <i>dwChannelId</i> , DWORD <i>dwMode</i> );	Функция снимает или кладет трубку в канале с идентификатором <b>dwChannelId</b> , дожидается снятия или положения трубки. Если в процессе ожидания была нажата клавиша <b>ESC</b> , то функция возвращает 0, если ожидание прошло успешно, то возвращает 1. Режим функции – снятие или положение трубки определяется параметром <b>dwMode</b> . Если <b>dwMode</b> равен 0, то трубка кладется, в противном случае трубка поднимается.

# Передача факса

## Алгоритм передачи факса

Чтобы начать передачу факса в процессе сеанса связи, необходимо выполнить следующие действия:

1. Преобразовать файл (файлы) BMP в формат hfm при помощи вызова функции [sdkBmp2Fax](#).
2. Вызвать функцию [sdkFaxInit](#).
3. Вызвать функцию [sdkFaxSend](#), которая начнет передачу факс-страницы.
4. Регулярно вызывать функцию [sdkFaxGetStatus](#) для того, чтобы дождаться окончания факс-сессии (поле **SendProcessStatus** структуры **sdkFaxStatus** станет равно **sdkFaxTransmitOff**).
5. Вызвать [sdkFaxCancelSend](#).
6. Вызвать [sdkFaxLeave](#).

Если в процессе факс-сессии необходимо прервать передачу факса, то необходимо сразу перейти к п.5, игнорируя п.4.

## Используемые функции

### sdkBmp2Fax

---

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD sdkBmp2Fax ( char *pszDestFile, char *pszSrcFile);</code>
<b>Входные параметры</b>	<p><i>*pszDestFile</i>     указатель на имя результата - hfm-файла. Должно обязательно содержать расширение hfm, иначе будет вызван конвертер из библиотеки <b>OlhaFullFax.dll</b> и факс не будет отправлен.</p> <p><i>*pszSrcFile</i>     указатель на имя исходного файла – BMP.</p>

---

### Описание

Функция позволяет преобразовать bmp-файл в формат hfm.

Прежде, чем передать факс, необходимо подготовить hfm-файл, с которым работает факс-ресурс платы ОЛЬХА-14.

Прежде, чем передать факс, необходимо подготовить hfm-файл, с которым работает факс-ресурс платы Преобразование данных производит библиотека **AFIC.DLL**.

#### Внимание!



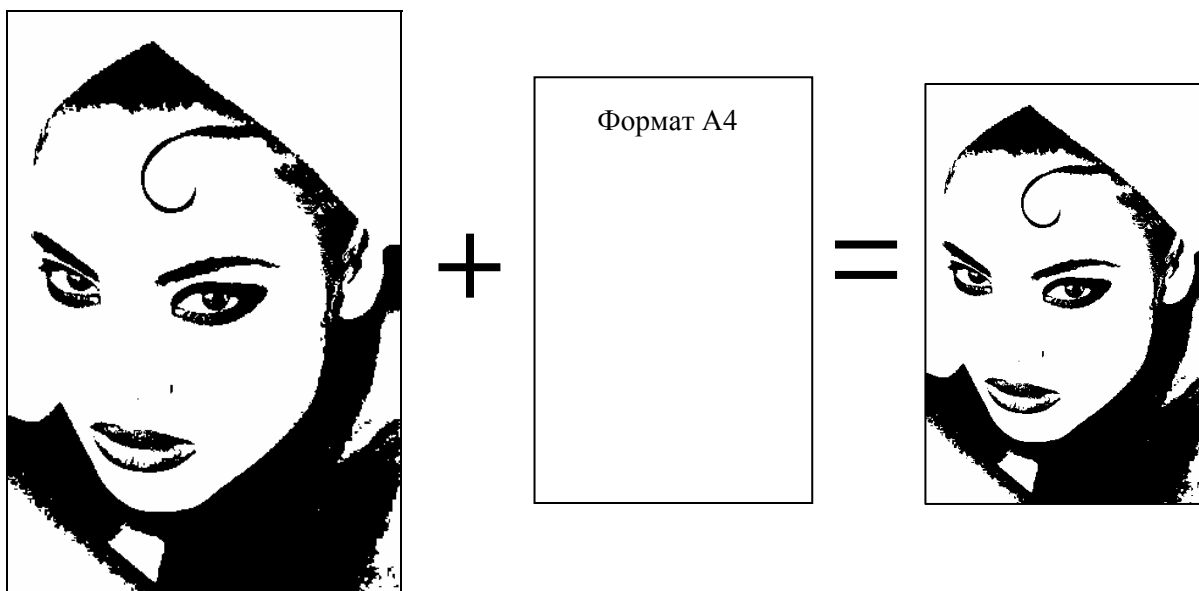
---

Для корректного преобразования файлов необходимо, чтобы библиотека **AFIC.DLL** был доступна в текущем каталоге.  
Не забываете в имени файла-результата указывать расширение **.hfm!**

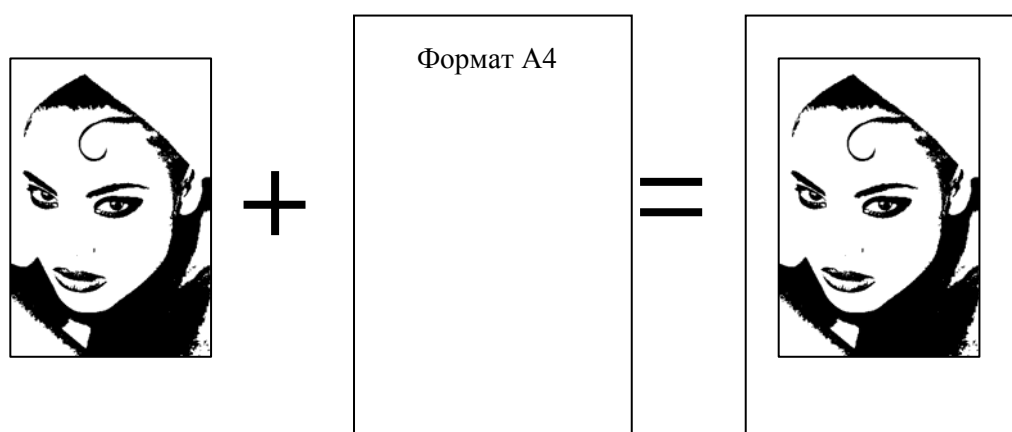
---

Если hfm-файл существует, то BMP-файл будет добавлен в конец hfm-файла в качестве страницы для передачи. Если hfm-файл не существует, то перекодированный BMP-файл будет первой страницей для передачи.

Если картинка больше, чем 1720 в ширину на 2444 точек в высоту, то изображение будет трансформировано до указанных размеров. Можно это проиллюстрировать так:

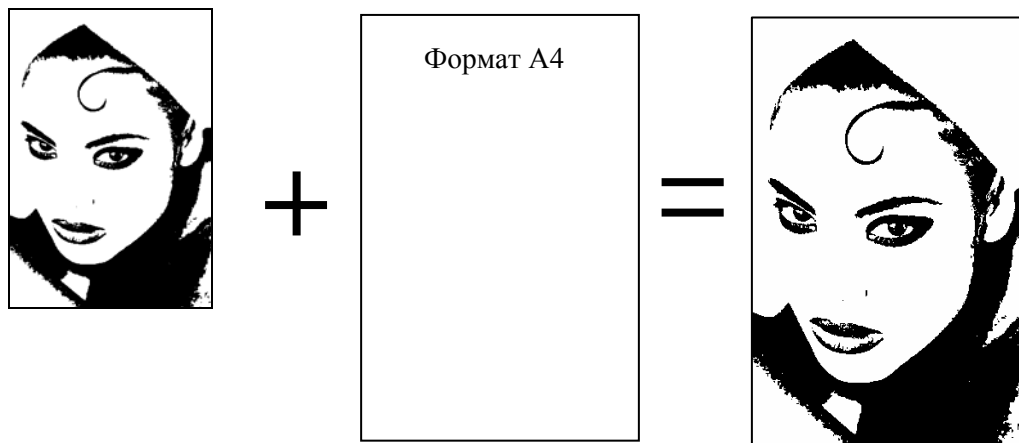


Если изображение меньше, чем 1720 в ширину на 2444 точек в высоту, то результат будет помещен в центр факс-страницы.



Для того чтобы управлять трансформацией исходного изображения, можно использовать букву 's' в расширении файла. То есть если необходимо задать флаг обязательной трансформации, то необходимо указать такое расширение файла **pszDestFile - .hfms**.

Обязательная трансформация предоставляет пользователю возможность «растянуть» картинку на всю страницу факса, если изображение меньше размера 1720 на 2444 точек. То есть результат преобразования всегда будет таким:



Для того чтобы передать факс-страницу в улучшенном качестве необходимо в расширении имени файла **DestFile** указать букву **'f'**. Например – **"test.hfmf"**.

По умолчанию – обычный режим передачи изображения.

Буквы **'s'** и **'f'** возможно использовать совместно. Например – **"test.hfmsf"**. В данном случае будет производиться дополнительная трансформация, и факс-страница будет передаваться в режиме **Fine**.

## sdkFaxInit

---

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD</code>	<code>sdkFaxInit( char *pszFaxKey, DWORD *lpdwMajorVersion, DWORD *lpdwMinorVersion);</code>
<b>Входные параметры</b>	<code>*pszFaxKey</code>	указатель на строку, содержащую ключ для факс-библиотеки.
	<code>*lpdwMajorVersion</code>	указатель на 4-х байтовое поле для возврата старшей цифры версии факс-библиотеки.
	<code>*lpdwMinorVersion</code>	указатель на 4-х байтовое поле для возврата младшей цифры версии факс-библиотеки.

---

### Описание

Функция вызывается при совместном использовании плат **ОЛЬХА-14** и других плат серии **ОЛЬХА** (**ОЛЬХА-9** или **ОЛЬХА-2**).

В случае совместного использования плат **ОЛЬХА-14** и плат серии **ОЛЬХА-9** или **ОЛЬХА-2**, входные параметры будут использоваться для инициализации библиотеки **OlhaFullFax**. В этом случае рекомендуется поместить библиотеку **OlhaFullFax** в рабочий каталог программы.

Если же плата **ОЛЬХА-14** используется без плат других серий, то входные параметры игнорируются. В этом случае библиотека **OlhaFullFax** не инициализируется, во внимание не принимается и может отсутствовать в текущем каталоге.

## sdkFaxSend

---

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD <b>sdkFaxSend</b>(DWORD <i>dwChannelId</i>, char <i>*pszFile</i>);</code>	
<b>Входные параметры</b>	<i>dwChannelId</i>	идентификатор канал, полученный при помощи функции <b>sdkOpenChannel</b> .
	<i>*pszFile</i>	указатель на имя файла с факс-данными для передачи.

---

### Описание

Функция запускает поток, в котором при помощи функций асинхронного ввода-вывода выполняется передача факса после завершения передачи факса, поток самостоятельно завершает работу.

Если канал принадлежит плате **ОЛЬХА-9**, то файл должен быть в формате библиотеки **OlhaFullFax.dll**.

Если же канал принадлежит плате **ОЛЬХА-14**, то формат файла должен соответствовать формату hfm.

**sdkFaxGetStatus**

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD <b>sdkFaxGetStatus</b></code> ( <code>DWORD <i>dwChannelId</i>,</code> <code><i>sdkFaxStatus *lpFaxStatus</i></code> );
<b>Входные параметры</b>	<i>dwChannelId</i> идентификатор канал, полученный при помощи функции <b>sdkOpenChannel</b> .  <i>*lpFaxStatus</i> указатель на структуру типа <b>sdkFaxStatus</b> для сохранения состояния факс-сессии по каналу.

**Описание**

При помощи этой функции, можно контролировать завершение факс-сеанса для принятия последующего решения.

Формат структуры **sdkFaxStatus** приведен ниже. Описаны не все поля структуры, а только те, которые используются для работы с факс-ресурсом платы **ОЛЬХА-14**.

Наименование	Тип	Описание
<b>RecProcessStatus</b>	DWORD	Состояние передачи факса. Если факс в процессе приема, то поле содержит значение <b>FaxReceive9600</b> . Если факс-сессия закончена, то в поле будет содержаться значение <b>FaxReceiveOff</b> .
<b>RecFinalStatus</b>	DWORD	Код завершения факс-сессии (см. <a href="#">Коды завершения факс-сеанса</a> ). Поле действительно, если <code>RecProcessStatus=FaxReceiveOff</code> .
<b>SendFinalStatus</b>	DWORD	Счетчик успешно принятых страниц.

## sdkFaxCancelSend

---

<b>Формат вызова</b>	<code>_declspec(dllimport) DWORD <b>sdkFaxCancelSend</b></code> ( <code>DWORD <i>dwChannelId</i></code> );
<b>Входные параметры</b>	<code><i>dwChannelId</i></code> идентификатор канал, полученный при помощи функции <code>sdkOpenChannel</code> .
<b>Выходные параметры</b>	Код завершения факс-сессии <b>117 (FAXMODEM_STATUS_CANCELLED)</b> .

---

### Описание

Функция немедленно завершает передачу факса. Поток, который передавал факс, немедленно останавливается и прекращает свое существование.

#### Полезно!

---



Рекомендуется в конце каждой факс-сессии после приема факса вызывать функцию `sdkFaxCancelSend`.

---

#### Внимание!

---



На удаленном факсе, вероятнее всего, факс-сессия завершится с ошибкой.

---

**sdkFaxLeave**

---

**Формат  
вызова**                    \_declspec(dllexport) DWORD **sdkFaxLeave**();

**Входные  
параметры**                Отсутствуют

---

**Описание**

Функция завершает работу с факс-ресурсами. Для плат серии **ОЛЬХА-9** происходит попытка выгрузить библиотеку **OlhaFullFax.dll** из памяти. Для плат **ОЛЬХА-14** никаких действий не производится.

## Пример sdkSendFax

Все вышеизложенное продемонстрировано в примере **sdkSendFax**. Пример отправляет указанный ВМР-файл с указанного канала. Предварительно ВМР-файл преобразуется в hfm-файл. Рассмотрим исходные тексты примера подробнее.

### Общие сведения

Пример **sdkRecFax** написан на языке C++ при использовании **MS Visual C++** версии 7.0.

Пример предназначен для демонстрации приема факса только на линиях **FXO**.

Пример оформлен в виде терминальной программы, работающей в текстовом режиме.

---

#### Внимание!

---



---

Исходные тексты программы можно использовать в коммерческих целях. Распространяются исходные тексты свободно и никаких ограничений на использование не имеют. Программа, так же как ее исходные тексты используются пользователем «как есть». Производитель не несет никакой ответственности, если возник ущерб от использования данной программы или ее исходных текстов.

---

### Функция Main()

Пример оформлен в файле **sdkRecFax.cpp**.

Инициализация библиотеки **AlderSDK**.

```
::printf("Init AlderSDK...\n");
if( sdkInit()!= SDK_NOERROR)
    return -1;
::printf("done\n");
```

Инициализация работы с факс-ресурсами.

```
DWORD Ver1,Ver2;
if( sdkFaxInit( FAX_KEY, &Ver1, &Ver2)!=SDK_NOERROR)
{
    sdkLeave();
    return -1;
}
```

Получение версии **AlderSDK** и вывод ее на экране.

```
sdkSDKVersion SDKVersion;  
sdkGetVersion( &SDKVersion);  
::printf("version - %d.%d.%d.%d\n",  
        SDKVersion.SDKMajor,  
        SDKVersion.SDKMinor,  
        SDKVersion.SDKInt1,  
        SDKVersion.SDKInt2);
```

Получение количества установленных плат в ПК.

```
DWORD BoardsCount;  
sdkBoardStatus *BoardStatus[8];  
if( sdkGetBoardsCount( &BoardsCount)!=SDK_NOERROR)  
{  
    sdkLeave();  
    return -1;  
}
```

Чтение информации об установленных платах в ПК.

```
for( int i= 0; i< ( int)BoardsCount; i++)  
    BoardStatus[i] = new sdkBoardStatus;  
if( sdkUpdateBoards( BoardsCount, BoardStatus)!=SDK_NOERROR)  
{  
    sdkLeave();  
    return -1;  
}
```

Проверка наличия установленных каналов с номером 0 на плате. Программа работает только с платой с номером 0 и каналом с номером 0, установленном на этой плате.

```
if( BoardStatus[0]->Lines== 0)  
{  
    sdkLeave();  
    return -1;  
}
```

## Резервирование и открытие канала.

```
::printf("Open Channel...\n");
DWORD ChannelId = BoardStatus[BOARD]->ChannelsId[LINE];
if( sdkReserveChannel(ChannelId)!=SDK_NOERROR)
{
    sdkLeave();
    return -1;
}
if( sdkOpenChannel(ChannelId)!=SDK_NOERROR)
{
    sdkLeave();
    return -1;
}
::printf("done\n");
```

## Снятие трубки в для установки связи.

```
if( dwSetHook( ChannelId, 1)== 0)
{
    sdkLeave();
    return -1;
}
```

Ожидание сигналов **DIALTONE** или **BUSY**. По обнаружению сигнала **BUSY**, прекращаем работу программы.

```
::printf("Wait DIAL TONE...\n");
if( dwWaitDialToneOrBusy( ChannelId)== 0)
{
    sdkLeave();
    return -1;
}
::printf("done\n");
```

### Набор номер в линию.

```
::printf("Dial number...\n");
if( dwWaitDial( ChannelId)== 0)
{
    sdkLeave();
    return -1;
}
::printf("done\n");
```

Ожидание в линии сигнала от удаленного факс-аппарата. Если обнаружится **BUSY**, то прекращаем работу программы.

```
::printf("Wait For Fax...\n");
if( dwWaitFaxOrBusy ( ChannelId)== 0)
{
    sdkLeave();
    return -1;
}
::printf("done\n");
```

### Подготовка hfm-файла.

```
::DeleteFile( FAX_FILE);
if( sdkBmp2Fax( FAX_FILE, BMP_FILE)!= SDK_NOERROR)
{
    sdkLeave();
    return -1;
}
```

### Запуск передачи факса.

```
if( sdkFaxSend( ChannelId, FAX_FILE)!= SDK_NOERROR)
{
    sdkLeave();
    return -1;
}
```

Ожидание завершения факс-сессии. По клавише **ESC** или **Spacebar**– прерываем факс-сессию, если это необходимо.

```
sdkFaxStatus FaxStatus;
DWORD dwUserCancel= 0;
while(1)
{
    switch( dwCheckUserKey())
    {
        case 1:
            ::printf( "User break...\n");
            dwUserCancel= 1;
            break;
        case 2:
            ::printf( "User cancel fax session...\n");
            sdkFaxCancelSend( ChannelId);
            break;
        default:
            break;
    }
    sdkFaxGetStatus( ChannelId, &FaxStatus);
    if( FaxStatus.SendProcessStatus== sdkFaxTransmitOff)
    {
        ::printf( "FaxFinalStatus= %d PageCount= %d\n",
            FaxStatus.SendFinalStatus, FaxStatus.RecFinalStatus);
        break;
    }
    ::Sleep(100);
}
```

Вызов функцию прекращения факс-сессии в любом случае, как рекомендовано выше.

```
sdkFaxCancelSend( ChannelId);  
printf("done\n");
```

Положить трубку в линии.

```
dwSetHook( ChannelId, 0);
```

Закрытие канала.

```
printf("Close Channel...\n");  
if( sdkCloseChannel(ChannelId)!=SDK_NOERROR)  
{  
    sdkLeave();  
    return -1;  
}  
if( sdkFreeChannel(ChannelId)!=SDK_NOERROR)  
{  
    sdkLeave();  
    return -1;  
}  
::printf( "done\n");
```

Завершение работы с факс-ресурсами.

```
sdkFaxLeave();
```

Прекращение работы с **AlderSDK**.

```
printf("Leave AlderSDK...");  
if( sdkLeave()!=SDK_NOERROR)  
    return -1;  
::printf( "done\n");
```

## Вспомогательные функции

В примере использованы некоторые вспомогательные функции. Вспомогательные функции и их назначение представлены в таблице:

Формат вызова функции	Описание
DWORD <b>dwCheckEscIsPressed()</b> ;	Функция проверяет, не была ли нажата клавиша <b>ESC</b> . Если <b>ESC</b> была нажата, то возвращает 1, в противном случае 0.
DWORD <b>dwCheckUserKey()</b> ;	Функция проверяет, какие клавиши нажал пользователь на клавиатуре. Функция возвращает 1, если была нажата клавиша <b>ESC</b> , 2 – если была нажата клавиша <b>Spacebar</b> (пробел), 0 – во всех остальных случаях.
DWORD <b>dwWaitEscKey()</b> ;	Функция синхронно ожидает нажатия клавиши на клавиатуре. Возвращает 1, если была нажата клавиша <b>ESC</b> , 0 - в противном случае.
DWORD <b>dwSetHook</b> ( DWORD <i>dwChannelId</i> , DWORD <i>dwMode</i> );	Функция снимает или кладет трубку в канале с идентификатором <b>dwChannelId</b> , дожидается снятия или положения трубки. Если в процессе ожидания была нажата клавиша <b>ESC</b> , то функция возвращает 0, если ожидание прошло успешно, то возвращает 1. Режим функции – снятие или положение трубки определяется параметром <b>dwMode</b> . Если <b>dwMode</b> равен 0, то трубка кладется, в противном случае трубка поднимается.
DWORD <b>dwWaitFaxOrBusy</b> ( DWORD <i>dwChannelId</i> );	Функция ожидает сигнала готовности факса или сигнала <b>BUSY</b> в линии с идентификатором <b>dwChannelId</b> . Возвращает 1, если был обнаружен сигнал готовности факса, 0 – если был обнаружен сигнал <b>BUSY</b> .
DWORD <b>dwWaitDialToneOrBusy</b> ( DWORD <i>dwChannelId</i> );	Функция ожидает сигнала <b>DIALTONE</b> или сигнала <b>BUSY</b> в линии с идентификатором <b>dwChannelId</b> . Возвращает 1, если был обнаружен сигнал <b>DIALTONE</b> и 0, если был обнаружен сигнал <b>BUSY</b> .
DWORD <b>dwDial</b> ( DWORD <i>dwChannelId</i> );	Функция набирает номер в линию с идентификатором <b>dwChannelId</b> . Номер хранится в глобальной переменной <b>pszNumber</b> . Тип набора номера – тональный. После набора номера функция ожидает реакции от платы с подтверждением окончания набора номера. Возвращает 1, если номер был удачно набран, 0 – если был обнаружен сигнал <b>BUSY</b> .

# ПРИЛОЖЕНИЕ

## Коды завершения факс-сеанса

Коды завершения факс-сеанса определены в файле **Alder14t.h**.

Символьное имя	Код	Комментарии
FAXMODEM_STATUS_RECEIVED_DCN	0	Удаленная сторона прислала сигнал <b>Disconnect</b> . В процессе сеанса ошибок не обнаружено.
FAXMODEM_STATUS_REQUEST_OPERATOR	1	Удаленная сторона вызывает оператора.
FAXMODEM_STATUS_NO_FAX_RESPONSE	2	Нет ответа от удаленного факса.
FAXMODEM_STATUS_FAR_SIDE_IS_RECEIVED	3	Удаленная сторона настроена на прием. Это ошибка – либо удаленная сторона выполняет неверные действия, либо программа пользователя выполняет ошибочные действия.
FAXMODEM_STATUS_NOTHING_TO_SEND	4	Нет страницы для передачи.
FAXMODEM_STATUS_ANSWER_IS_GONE	5	Во время факс-сессии нет ответа от удаленной стороны.
FAXMODEM_STATUS_LAST_PAGE_SENT_ERROR	7	Последняя страница была передана с ошибкой.
FAXMODEM_STATUS_INVALID_PAGE_CONFIRM	8	Неправильное подтверждение получения страницы.
FAXMODEM_STATUS_NO_DIS_OR_DTC	9	Удаленная сторона не готова принимать.
FAXMODEM_STATUS_FARSIDE_NOT_SUPPORT_OUR_MODES	10	Удаленная сторона не поддерживает ни один из режимов передачи.
FAXMODEM_STATUS_RETRAIN_LIMIT	11	Истекло количество попыток установить связь.
FAXMODEM_STATUS_INVALID_ACK	12	Удаленная сторона прислала неправильно подтверждение.
FAXMODEM_STATUS_INVALID_PAGE_ACK	13	Удаленная сторона прислала неправильно подтверждение.
FAXMODEM_STATUS_NO_PAGE_ACK	14	Удаленная сторона прислала неправильное подтверждение.
FAXMODEM_STATUS_PAGE_ACK_RECEIVED	15	Удаленная сторона прислала неправильное подтверждение.
FAXMODEM_STATUS_INVALID_V21	16	Получен неправильный сигнал V.21.
FAXMODEM_STATUS_SENT_OK	17	Передача завершилась успешно.
FAXMODEM_STATUS_CANCELLED	117	Пользователь прекратил работу факс-ресурса, не дождавшись окончания факс-сессии.