

МОДУЛИ «ОСПЧ»

БИБЛИОТЕКА ФУНКЦИЙ. МНОГОКАНАЛЬНЫЙ РЕЖИМ.

Руководство программиста

Листов 41

2017

СОДЕРЖАНИЕ

1. Общие сведения	3
2. Общие функции	4
2.1. Инициализация и освобождение ресурсов	4
2.2. Описание последней ошибки	4
2.3. Загрузка ОЗУ демодулятора и декодера	5
2.4. Сброс демодулятора и декодера	6
2.5. Ввод лицензионного ключа	6
3. Функции управления демодулятором	8
3.1. Установка параметров	8
3.2. Настройка тактовой и несущей частот	9
3.3. Загрузка фильтра	10
3.4. Управление схемами регулировки усиления	11
3.5. Считывание параметров демодулятора	13
4. Функции управления декодером	16
4.1. Загрузка декодера Витерби	16
4.2. Загрузка декодера TPC	17
4.3. Загрузка декодера ССК	21
4.4. Загрузка декодера VF LDPC	23

4.5. Загрузка декодера LDPC	24
4.6. Настройка декодера Рида-Соломона	25
4.7. Загрузка дополнительных опций декодеров	26
4.8. Установка маски каналов	28
4.9. Считывание параметров декодера	28
5. Функции управления каналами данных	31
6. Возвращаемые значения	35
7. Рекомендуемый порядок работы	36
7.1. Подключение к устройству	36
7.2. Загрузка демодулятора	36
7.3. Загрузка декодера на примере TPC	37
7.4. Настройка буферов памяти для получения данных	38
7.5. Запись сигнала в непрерывном режиме	38
7.6. Завершение работы	39
8. Перечень сокращений	40

1. ОБЩИЕ СВЕДЕНИЯ

Комплект «Libm SDK» предназначен для разработки ПО управления устройствами «ОСПЧ-Е3», «ОСПЧ-Е2», «ОСПЧ-Е1», «ОСПЧ-Е» и «ОСПЧ-М1» в многоканальном режиме. Для устройств «ОСПЧ-Е» и «ОСПЧ-М1» возможна работа в 16-ти канальном режиме. Устройства «ОСПЧ-Е1», «ОСПЧ-Е2» и «ОСПЧ-Е3» поддерживают работу в 8, 16 и 32 канальном режиме.

В комплект «Libm SDK» включены:

- библиотека управления;
- заголовочные файлы;
- руководство программиста;
- примеры.

В заголовочных файлах представлены описания интерфейса функций, типов параметров функций, значений возврата функций. Все функции библиотеки, за небольшим исключением, возвращают код ошибки. Коды возвратов функций описаны в разделе 6. В функциях считывания значений или статуса некоторые параметры передаются по указателю. При успешном выполнении функции поля структур или параметры будут заполнены считываемыми значениями.

2. ОБЩИЕ ФУНКЦИИ

2.1. Инициализация и освобождение ресурсов

Для инициализации библиотеки необходимо использовать функцию `InitLibrary`. Для освобождения ресурсов библиотеки предназначена функция `ReleaseLibrary`. Если предполагается управление несколькими устройствами из одного процесса, функции `InitLibrary` и `ReleaseLibrary` необходимо вызывать для каждого устройства.

```
int InitLibrary (
    UCHAR    DevNum
);
```

Параметр	Описание
DevNum	номер устройства

```
int ReleaseLibrary (
    UCHAR    DevNum
);
```

Параметр	Описание
DevNum	номер устройства

2.2. Описание последней ошибки

В случае возникновения ошибки возвращаемое значение функции библиотеки будет отличным от нуля. Определения и значения кодов ошибок содержатся в разделе 6. Для получения подробной информации об ошибке следует воспользоваться функцией `ReturnLastError`.

```
char* ReturnLastError (
    UCHAR    DevNum
);
```

Параметр	Описание
DevNum	номер устройства

2.3. Загрузка ОЗУ демодулятора и декодера

Загрузка ОЗУ демодулятора выполняется функцией LoadDEMARAM. Если используется модуль «ОСПЧ-Е» или «ОСПЧ-М1», перед вызовом LoadDEMARAM необходимо выполнить загрузку ПЛИС демодулятора файлом конфигурации "load_ram.bit". Загрузку ПЛИС файлом рабочей конфигурации следует произвести после загрузки ОЗУ демодулятора. Для устройств «ОСПЧ-Е1», «ОСПЧ-Е2», «ОСПЧ-Е3» загрузку ОЗУ функцией LoadDEMARAM следует производить после загрузки файла рабочей конфигурации ПЛИС.

<pre>int LoadDEMARAM (UCHAR DevNum, PCHAR RAM_File);</pre>	
Параметр	Описание
DevNum	номер устройства
RAM_File	путь к файлу, загружаемому в ОЗУ демодулятора

Загрузка ОЗУ декодера выполняется функцией LoadDECARAM. Для мезонинного модуля «Декодер-М1» загрузку ОЗУ необходимо производить до загрузки ПЛИС декодера, при этом в качестве параметра FPGA_File необходимо указать файл "ext_ram.bit". Если используется мезонинный модуль «Декодер-М2» или «Декодер-М3», загрузка ОЗУ декодера выполняется после загрузки ПЛИС декодера, при этом параметр FPGA_File игнорируется (NULL).

<pre>int LoadDECARAM (</pre>

```

    UCHAR      DevNum,
    PCHAR      FPGA_File,
    _DEC_RAM_NUM RamNum
);

```

Параметр	Описание
DevNum	номер устройства в системе
FPGA_File	путь к файлу конфигурации ПЛИС для загрузки ОЗУ
RAM_File	путь к файлу, загружаемому в ОЗУ декодера
RamNum	номер ОЗУ декодера

2.4. Сброс демодулятора и декодера

После загрузки демодулятора и декодера необходимо выполнить сброс схем ФАПЧ. Для этого необходимо вызвать функцию ResetDemDec.

```

int ResetDemDec (
    UCHAR      DevNum
);

```

Параметр	Описание
DevNum	номер устройства

2.5. Ввод лицензионного ключа

Для работы определенных режимов устройств «ОСПЧ», в частности при получении данных с выхода декодера, необходим ввод лицензионного ключа. Ввод лицензии осуществляется функцией InstallLicenseKey и должен быть произведен сразу после загрузки ПЛИС демодулятора.

```

int InstallLicenseKey (
    UCHAR      DevNum,
    PINT64     LicenseKey
);

```

Параметр	Описание
DevNum	номер устройства
LicenseKey	лицензионный ключ

3. ФУНКЦИИ УПРАВЛЕНИЯ ДЕМОДУЛЯТОРОМ

3.1. Установка параметров

Установка базовых параметров демодулятора, таких как вид модуляции, автоподстройка частоты, ширина полосы ФАПЧ выполняется с помощью функции SetDEMParam. Параметры демодулятора настраиваются для каждого канала.

<pre>int SetDEMParam (UCHAR DevNum, UCHAR Channel, PDEMPParameters DEMParam);</pre>	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
DEMParam	указатель на структуру параметров демодулятора

Таблица 1. Структура параметров демодулятора

<pre>DEMPParameters { UCHAR Modulation; bool CarrierTracking; bool ClockTracking; bool APCH; bool PLL; };</pre>	
Modulation	вид модуляции
CarrierTracking	автоподстройка несущей частоты
ClockTracking	автоподстройка тактовой частоты
APCH	АПЧ
PLL	ФАПЧ

3.2. Настройка тактовой и несущей частот

Установка несущей и центральной частоты выполняется функцией SetCarrierFrequency.

```
int SetCarrierFrequency (
    UCHAR    DevNum,
    UCHAR    Channel,
    double   CarrierFreq,
    double   CentralFreq
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
CarrierFreq	несущая частота, кГц
CentralFreq	центральная частота, кГц

Для установки тактовой частоты необходимо использовать функцию SetClockFrequency.

```
int SetClockFrequency (
    UCHAR    DevNum,
    UCHAR    Channel,
    double   ClockFreq
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
ClockFreq	тактовая частота, кГц

Настройка частот выполняется для каждого канала демодулятора. Частоты задаются в килогерцах (кГц).

Считать измеренное значение несущей или тактовой частоты с учетом ФАПЧ можно с помощью функции MeasureFrequency.

```
int MeasureFrequency (
```

```
    UCHAR    DevNum,
```

```
    UCHAR    Channel,
```

```
    double*  CarrierFreq,
```

```
    double*  ClockFreq
```

```
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
CarrierFreq	указатель на переменную, содержащую измеренное значение несущей частоты
ClockFreq	указатель на переменную, содержащую измеренное значение тактовой частоты

3.3. Загрузка фильтра

Загрузка фильтра для каждого канала демодулятора выполняется функцией LoadFilter.

```
int LoadFilter (
```

```
    UCHAR    DevNum,
```

```
    UCHAR    Channel,
```

```
    char*    FilterFile
```

```
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
FilterFile	путь к файлу фильтра

3.4. Управление схемами регулировки усиления

Устройства «ОСПЧ» имеют две схемы регулировки усиления (РУ). Первая схема РУ аналоговая, настраивается для всех каналов. Вторая схема РУ цифровая, реализована в ПЛИС, настраивается для каждого канала. Схемы регулировки усиления могут работать в автоматическом (APУ) и ручном режиме (PPУ). Для настройки первой схемы РУ необходимо использовать функцию SetGC1. Управление второй схемой РУ производится функцией SetGC2.

<pre>int SetGC1 (UCHAR DevNum, PGC1Control GC1Ctrl);</pre>	
Параметр	Описание
DevNum	номер устройства
GC1Ctrl	указатель на структуру параметров первой схемы РУ

Таблица 2. Структура параметров первой схемы РУ

<pre>GC1Control { WORD Gain; UCHAR LHSwitch; UCHAR Manual; UCHAR Att; };</pre>	
Gain	нижняя или верхняя граница "нечувствительности" APУ
LHSwitch	выбор настройки нижней или верхней границы APУ
Manual	включение ручного режима управления схемой РУ
Att	код управления аттенюатором при PПУ

```
int SetGC2 (
```

<pre> UCHAR DevNum, UCHAR Channel, PGC2Control GC2Ctrl); </pre>	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
GC2Ctrl	указатель на структуру параметров второй схемы РУ

Таблица 3. Структура параметров второй схемы РУ

<pre> GC2Control { WORD Threshold; WORD Limit; WORD Gain; UCHAR Manual; UCHAR TimeHigh; UCHAR TimeLow; }; </pre>	
Threshold	значение порога работы АРУ
Limit	ограничитель макс. усиления АРУ
Gain	код усиления при РРУ
Manual	включение ручного режима управления схемой РУ
TimeHigh	управление постоянной времени АРУ
TimeLow	управление постоянной времени АРУ

Считывание значений аттенюаторов первой и второй схемы РУ выполняется функцией `GetValuesGC`.

```

int GetValuesGC (
    UCHAR      DevNum,
    UCHAR      Channel,
    PGCValues  GCVal
    )
    
```

);	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
GCVaI	указатель на структуру параметров РУ

Таблица 4. Структура параметров схем РУ

<pre> GCValues { double ARU1Hard; double ARU1Soft; DWORD ARU1_YE; DWORD ARU2_YE; double ARU2_dB; }; </pre>	
ARU1Hard	значение аттенюатора первой схемы РУ, дБ
ARU1Soft	значение аттенюатора первой схемы РУ, дБ
ARU1_YE	код аттенюатора первой схемы РУ
ARU2_YE	код аттенюатора второй схемы РУ
ARU2_dB	значение аттенюатора второй схемы РУ, дБ

3.5. Считывание параметров демодулятора

При работе с «ОСПЧ» может понадобиться считать состояние демодулятора. К тому же, для настройки второй схемы РУ и канала данных DMD требуются параметры, зависящие от настроек демодулятора. Считать статус демодулятора, а именно, синхронизацию по несущей и тактовой частоте, а также статус перегрузки первой схемы РУ можно с помощью функции ReadDEMStatus.

```

int ReadDEMStatus (
    UCHAR          DevNum,

```

<pre> UCHAR Channel, PDEMStatus DEMStatus); </pre>	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
DEMStatus	указатель на структуру статуса демодулятора

Таблица 5. Структура параметров статуса демодулятора

<pre> DEMStatus { bool VN; bool TH; bool GC1; }; </pre>	
VN	синхронизация по несущей частоте
TH	синхронизация по тактовой частоте
GC1	статус перегрузки АЦП

С помощью функции ReadSNRE1 осуществляется считывание значения ОСШ для «ОСПЧ-Е1», «ОСПЧ-Е2», «ОСПЧ-Е3».

<pre> int ReadSNRE1 (UCHAR DevNum, double* SNR); </pre>	
Параметр	Описание
DevNum	номер устройства
SNR	указатель на переменную, содержащую значение ОСШ

Функция GetModParameters возвращает параметры, зависящие от текущих настроек демодулятора.

<pre>int GetModParameters (PMODSetup Modulation, PMODParameters ModParam);</pre>	
Параметр	Описание
DevNum	номер устройства
Modulation	вид модуляции
ModParam	указатель на структуру параметров демодулятора

Таблица 6. Структура параметров демодулятора

<pre>MODParameters { WORD GC2Threshold; struct { UCHAR Format; UCHAR Kdv; char PackFile[NUM_CHARS]; } DMDCH; };</pre>	
GC2Threshold	коэффициент, необходимый для второй схемы РУ
DMDCH.Format	формат данных DMD канала (см. раздел 5)
DMDCH.Kdv	коэффициент деления частоты
DMDCH.PackFile	имя файла для загрузки в режиме DMDPack

4. ФУНКЦИИ УПРАВЛЕНИЯ ДЕКОДЕРОМ

4.1. Загрузка декодера Витерби

Для загрузки декодера Витерби необходимо загрузить ПЛИС файлом конфигурации "vit_k7_mult.bit" и задать параметры. Для загрузки файла конфигурации ПЛИС используется функция LoadViterbi.

<pre>int LoadViterbi (UCHAR DevNum, char* FPGAFile);</pre>	
Параметр	Описание
DevNum	номер устройства
FPGAFile	путь к файлу конфигурации ПЛИС декодера Витерби

Для настройки параметров используется LoadViterbiReg. В данном режиме реализовано 8 независимых каналов обработки данных.

<pre>int LoadViterbiReg (UCHAR DevNum, UCHAR Channel, PViterbiParameters ViterbiParam);</pre>	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
ViterbiParam	указатель на структуру параметров декодера Витерби

Таблица 7. Структура параметров декодера Витерби

<pre>ViterbiParameters { UCHAR Modulation; VITERBI_TYPE VitType;</pre>	
--	--

	bool	ChannelOrder;
	bool	SpectrInversion;
	};	
Modulation	вид модуляции	
VitType	тип декодера Витерби	
ChannelOrder	порядок следования каналов	
SpectrInversion	инверсия спектра	

4.2. Загрузка декодера TPC

Загрузка декодера TPC производится в несколько этапов:

- загрузка ПЛИС файлом рабочей конфигурации и загрузка ОЗУ;
- настройка общих параметров декодера;
- настройка параметров TPC;
- настройка количества итераций.

Загрузка ПЛИС файлом рабочей конфигурации осуществляется функцией LoadTPCHamming. Для мезонинных модулей «Декодер-М1» используется файл конфигурации "tpc_fpga_mult.bit". Для «Декодер-М2» нужно использовать файл "tpc_aha_mult.bit". Если установлен мезонинный модуль «Декодер-М3», загружается файл "tpc_fpga_mult.bit".

int LoadTPCHamming (UCHAR DevNum, char* FPGAFile);	
Параметр	Описание
DevNum	номер устройства
FPGAFile	путь к файлу конфигурации ПЛИС декодера TPC

Настройка общих параметров декодера включает в себя настройку инверсии спектра и порядка каналов и выполняется функцией LoadTPCHammingReg.

```
int LoadTPCHammingReg (
    UCHAR          DevNum,
    UCHAR          Channel,
    PTPC_HAMMING_REG  TPCRegParam
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
TPCRegParam	указатель на структуру параметров декодера TPC

Таблица 8. Структура общих параметров декодера TPC

```
TPC_HAMMING_REG {
    UCHAR    Modulation;
    bool     ChannelOrder;
    bool     SpectrInversion;
};
```

Modulation	вид модуляции
ChannelOrder	порядок следования каналов
SpectrInversion	инверсия спектра

Настройка параметров TPC производится функцией LoadTPCHammingANA.

```
int LoadTPCHammingANA (
    UCHAR          DevNum,
    PTPC_HAMMING_ANA  TPCANAParam
);
```

Параметр	Описание
DevNum	номер устройства
TPCANAParam	указатель на структуру параметров загрузки микросхемы АНА декодера TPC

Таблица 9. Структура параметров канала декодера TPC

```

TPC_HAMMING_AHA {
    UCHAR    nX, kX, nY, kY, nZ, kZ;
    USHORT   Reduce1;
    USHORT   Reduce2;
    bool     Dopus;
    USHORT   NumBlocks;
    USHORT   UWLength;
    DWORD    UW1, UW2, UW3;
    _ORDER   UWDel;
    bool     Dintrlv;
    USHORT   SyncBitErrEntry;
    USHORT   SyncBitErrExit;
    DWORD    ScrPoly;
    DWORD    ScrSet;
    _ORDER   Scrsw;
    _SYNC    Scrsync;
    USHORT   DelLastBits;
};

```

nX	ПУ код по x
kX	ПУ код по x
nY	ПУ код по y
kY	ПУ код по y
nZ	ПУ код по z
kZ	ПУ код по z
Reduce1	сокращение первой оси
Reduce2	сокращение первой плоскости
Dopus	дополнительная четность
NumBlocks	количество блоков
UWLength	длина уникального слова
UW1	уникальное слово (31:0)
UW2	уникальное слово (63:32)

UW3	уникальное слово (95:64)
UWDel	удаление уникального слова
Dintrlv	деперемежитель
SyncBitErrEntry	количество ошибочных бит в синхропризнаке при поиске
SyncBitErrExit	количество ошибочных бит в синхропризнаке при проверке
ScrPoly	полином аддитивного скремблера
ScrSet	установка аддитивного скремблера
Scrsw	включение аддитивного скремблера
Scrsync	синхронизация аддитивного скремблера
DelLastBits	удаление последних бит

Для настройки количества итераций используется функция LoadTPCHammingIter.

<pre>int LoadTPCHammingIter (UCHAR DevNum, PTPC_HAMMING_ITER TPCANAlterParam);</pre>	
Параметр	Описание
DevNum	номер устройства
TPCANAlterParam	указатель на структуру параметров декодера TPC

Таблица 10. Структура параметров количества итераций декодера TPC

<pre>TPC_HAMMING_ITER { UCHAR* Modulation; double* ClkF; };</pre>	
Modulation	набор идентификаторов вида модуляции для каждого канала
ClkF	набор значений частоты для каждого канала

В зависимости от версии мезонинного модуля количество каналов обработки данных различается. Для «Декодер-М1» доступны 4 канала, имеющие одинаковые параметры ТРС. Для «Декодер-М2» доступны 8 каналов, так же имеющие одинаковые параметры ТРС. Если используется мезонинный модуль «Декодер-М3», обработка данных возможна по 3 каналам, имеющим независимые настройки.

4.3. Загрузка декодера ССК

Для загрузки декодера ССК необходимо:

- загрузить ПЛИС файлом рабочей конфигурации;
- настроить параметры ССК;
- настроить общие параметры декодера.

Загрузка ПЛИС выполняется функцией LoadSSK. При выборе файла конфигурации ПЛИС учитывается требуемый режим, а также версия мезонинного модуля.

Для мезонинных модулей «Декодер-М1» и «Декодер-М2», имя файла конфигурации ПЛИС начинается с "seq_12", "seq_34" или "seq_78" для режима ССК 1/2, 3/4 или 7/8 соответственно. Если используется мезонинный модуль «Декодер-М1», имя файла заканчивается на "_4k.bit", при этом декодер имеет 4 одинаково настраиваемых канала. Для мезонинного модуля «Декодер-М2» имя файла конфигурации ПЛИС имеет окончание "_mult.bit". На данном мезонинном модуле доступно 8 одинаково настраиваемых каналов.

Если установлен мезонинный модуль «Декодер-М3», вне зависимости от режима ССК используется файл конфигурации "seq_a_mult.bit". В данной

конфигурации реализовано 8 независимо настраиваемых каналов обработки данных.

<pre>int LoadSSK (UCHAR DevNum, char* FPGAFile);</pre>	
Параметр	Описание
DevNum	номер устройства
FPGAFile	путь к файлу конфигурации ПЛИС декодера ССК

<pre>int LoadSSKReg (UCHAR DevNum, UCHAR Channel, PSSKParameters SSKParam);</pre>	
Параметр	Описание
DevNum	номер устройства
SSKParam	указатель на структуру параметров декодера ССК

Таблица 11. Структура параметров декодера ССК

<pre>SSKParameters { UCHAR Modulation; SSK_TYPE SSKType; PUNCTURING Punct; bool ChannelOrder; bool SpectrInversion; };</pre>	
Modulation	вид модуляции
SSKType	тип декодера ССК
Punct	выкалывание

ChannelOrder	порядок следования каналов
SpectrInversion	инверсия спектра

4.4. Загрузка декодера VF LDPC

Для загрузки декодера VF LDPC необходимо загрузить ПЛИС и ОЗУ декодера и задать параметры. Загрузка ПЛИС выполняется функцией LoadVFLDPC. Для мезонинного модуля «Декодер-М2» необходимо использовать файл конфигурации "vf_mult_4ch.bit". Если установлен мезонинный модуль «Декодер-М3», используется файл "vf_mult_4ch.bit". Для мезонинного модуля «Декодер-М1» данный режим декодера недоступен. ОЗУ необходимо загружать файлом "p8_q8_q16.iq". Для мезонинного модуля «Декодер-М2» доступно 4 канала обработки данных. Если используется мезонинный модуль «Декодер-М3», доступно 8 каналов.

<pre>int LoadVFLDPC (UCHAR DevNum, char* FPGAFile);</pre>	
Параметр	описание
DevNum	номер устройства
FPGAFile	путь к файлу конфигурации ПЛИС декодера VF LDPC

Для загрузки параметров используется функция LoadVFLDPCReg.

<pre>int LoadVFLDPCReg (UCHAR DevNum, UCHAR Channel, PVFLDPCParameters VFLDPCParam);</pre>	
Параметр	Описание

DevNum	номер устройства
Channel	номер настраиваемого канала
VFLDPCParam	указатель на структуру параметров декодера VF LDPC

Таблица 12. Структура параметров декодера VFLDPC

<pre>VFLDPCParameters { VFLDPC_TYPE VType; bool SpectrInversion; };</pre>	
VType	тип декодера VF LDPC
SpectrInversion	инверсия спектра

4.5. Загрузка декодера LDPC

Загрузка декодера LDPC производится загрузкой ПЛИС и ОЗУ, а также настройкой параметров декодера. Загрузка ПЛИС осуществляется функцией LoadLDPC600. Для загрузки ПЛИС необходимо использовать файл конфигурации "с600_mult_8ch.bit". ОЗУ необходимо загружать файлом "p8_q8_q16.iq". Режим декодера LDPC доступен только на мезонинных модулях «Декодер-М2» и «Декодер-М3». В нем реализовано 8 каналов обработки данных.

<pre>int LoadLDPC600 (UCHAR DevNum, char* FPGAFile);</pre>	
Параметр	Описание
DevNum	номер устройства
FPGAFile	путь к файлу конфигурации ПЛИС декодера LDPC

Настройка параметров для загрузки декодера LDPC выполняется функцией LoadLDPC600Reg.

```
int LoadLDPC600Reg (
    UCHAR          DevNum,
    UCHAR          Channel,
    PLDPC600Parameters LDPC600Param
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер настраиваемого канала
LDPC600Param	указатель на структуру параметров декодера LDPC

Таблица 13. Структура параметров декодера LDPC

```
LDPC600Parameters {
    UCHAR    Speed;
    bool     SpectrInversion;
    UCHAR    Modulation;
};
```

Speed	скорость декодера
SpectrInversion	инверсия спектра
Modulation	вид модуляции

4.6. Настройка декодера Рида-Соломона

В устройствах «ОСПЧ» реализован декодер Рида-Соломона. Для его настройки и включения используется функция SetDECERS.

```
int SetDECERS (
    UCHAR    DevNum,
    PRSOptions RSOpt
);
```

Параметр	Описание
DevNum	номер устройства
RSOpt	указатель на структуру параметров декодера Рида-Соломона

Таблица 14. Структура параметров декодера Рида-Соломона

<pre>RSOptions { WORD BlockLength; WORD InfSymCnt; WORD RowShift; WORD RowCount; bool AddScrEn; };</pre>	
BlockLength	размер блока
InfSymCnt	количество информационных символов
RowShift	сдвиг строк
RowCount	количество строк
AddScrEn	включение аддитивного дескремблера

4.7. Загрузка дополнительных опций декодеров

При декодировании сигнала может понадобиться дополнительная обработка с помощью дифференциального декодера, инвертора или дескремблера. Включить и настроить данные опции можно с помощью функции SetDECOptions.

<pre>int SetDECOptions (UCHAR DevNum, PDECOptions DECOpt);</pre>	
Параметр	Описание
DevNum	номер устройства

DECOpt	указатель на структуру параметров, общих для всех декодеров
--------	---

Таблица 15. Структура дополнительных параметров декодеров

<pre> DECOptions { bool DECDiffEn; bool DECIInvEn; _DESCRAMBLER DECDscrOptions; DWORD DECDscrPolynome; }; </pre>	
DECDiffEn	включение дифференциального декодера
DECIInvEn	включение инвертора декодера
DECDscrOptions	управление дескремблером
DECDscrPolynome	полином нестандартного дескремблера (_DESCRAMBLER::DSCR_NONSTD)

Для управления демультиплексированием сигнала используется функция LoadDMX.

<pre> int LoadDMX (UCHAR DevNum, PDMXParameters DMXParam); </pre>	
Параметр	Описание
DevNum	номер устройства
DMXParam	указатель на структуру параметров демультиплексора

Таблица 16. Структура параметров демультиплексора

<pre> DMXParameters { _DMX DMXMode; WORD FrameLength; bool ADDScramblerEn; }; </pre>	
---	--

DMXMode	режим демультимплексора
FrameLength	размер кадра
ADDScramblerEn	включение аддитивного скремблера

4.8. Установка маски каналов

При использовании декодеров LDPC, VF LDPC или TPC Hamming может потребоваться включение или отключение некоторых каналов обработки данных. Данная настройка производится заданием маски каналов при помощи функции SetChannelMask.

<pre>int SetChannelMask (UCHAR DevNum, DWORD Mask);</pre>	
Параметр	Описание
DevNum	номер устройства
Mask	маска каналов

4.9. Считывание параметров декодера

В процессе работы с устройством может потребоваться информация о состоянии декодера. Узнать версию установленного мезонинного модуля можно с помощью функции GetDecoderVersionOnBoard.

<pre>int GetDecoderVersionOnBoard (UCHAR DevNum, DECODER_VERSION* DecVersion);</pre>	
Параметр	Описание
DevNum	номер устройства

DecVersion	указатель на переменную, содержащую версию мезонинного модуля, установленного в устройство
------------	--

Функция ReadDECTemperature возвращает температуру ПЛИС декодера, а также статус перегрева ПЛИС.

<pre>int ReadDECTemperature (UCHAR DevNum, PDECTemperature DECTemp);</pre>	
Параметр	Описание
DevNum	номер устройства
DECTemp	указатель на структуру значений температуры и статуса перегрева ПЛИС декодера

Таблица 17. Структура параметров температуры декодера

<pre>DECTemperature { int FPGA_DEC_Temperature; bool FPGA_DEC_Overheat; };</pre>	
FPGA_DEC_Temperature	температура ПЛИС декодера
FPGA_DEC_Overheat	статус перегрева ПЛИС декодера

Чтобы получить информацию о синхронизации декодера, а также вероятность битовой ошибки на выходе, необходимо воспользоваться функцией ReadDECStatus.

<pre>int ReadDECStatus (UCHAR DevNum, UCHAR Channel, PDECStatus DECStatus);</pre>	
---	--

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
DECStatus	указатель на структуру статуса декодера

Таблица 18. Структура параметров состояния декодера

<pre>DECStatus { double BER; BYTE DEC; bool DRS; bool RSP; bool DMX; };</pre>	
BER	вероятность битовой ошибки на входе декодера
DEC	синхронизация декодера
DRS	синхронизация деперемежителя декодера Рида-Соломона
RSP	синхронизация декодера Рида-Соломона
DMX	синхронизация демультиплексора

5. ФУНКЦИИ УПРАВЛЕНИЯ КАНАЛАМИ ДАННЫХ

Устройства «ОСПЧ» поддерживают два режима передачи данных в ОЗУ ПЭВМ: режим с захватом шины PCI и прямым доступом к памяти (Busmaster) и режим Slave. Каналы передачи данных DMA можно разделить на 2 типа: DMD и IQ.

Каждый канал обработки данных демодулятора имеет свой канал DMA передачи демодулированных и декодированных данных (DMD). Канал IQ общий для всех каналов обработки данных демодулятора. Данный DMA канал (IQ) предназначен для получения данных АЦП и отсчетов для построения векторной диаграммы выбранного канала обработки данных демодулятора.

Для данных с выхода демодулятора и декодера номер канала DMA соответствует номеру канала обработки сигнала демодулятора. Для данных АЦП или IQ номер DMA канала равен 17 для «ОСПЧ-Е» и «ОСПЧ-М1», и равен 33 для «ОСПЧ-Е1», «ОСПЧ-Е2» и «ОСПЧ-Е3».

Настройка DMD-канала выполняется функцией SetDMDChannel. В зависимости от формата записываемого сигнала, параметры Format и Kdv настраиваются по-разному. Для ввода данных в режиме "Демодулятор - 8 бит", значение параметра Format равно 7, параметр Kdv равен 2. Для данных с выхода декодера значение параметров Format и Kdv следует использовать 8 и 1 соответственно. При вводе данных в режиме "Демодулятор - упаковка" значения Format и Kdv необходимо получить с помощью функции GetModParameters.

```
int SetDMDChannel (
    UCHAR          DevNum,
    UCHAR          Channel,
    PDMDParameters DMDParam
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
DMDParam	указатель на структуру параметров канала DMD

Таблица 19. Структура параметров канала данных DMD

```
DMDParameters {
    UCHAR    Format;
    UCHAR    Master;
    UCHAR    Reset;
    UCHAR    Kdv;
    UCHAR    Slave;
    UCHAR    BufferSize;
};
```

Format	управление форматом сигнала при записи
Master	пересылка данных в режиме Busmaster
Reset	сброс счетчиков
Kdv	коэффициент деления частоты
Slave	пересылка данных в режиме Slave
BufferSize	размер пересылаемого блока данных

Настройка IQ-канала выполняется функцией SetIQChannel. Параметр Format определяет формат записи сигналов. Если установлено значение 0, то передаются 8-разрядные выборки каналов I и Q. Значение 1 включает передачу 16-разрядных выборок каналов I и Q. При записи 2 передаются отсчеты сигнала с АЦП в виде 16-разрядного дополнительного кода.

Младшие 4 разряда принудительно установлены в 0. Младшие 16 разрядов 32-разрядного слова соответствуют более раннему отсчету. Значение 3 означает, что по данному каналу будет передаваться значение инкрементирующегося 32-разрядного счетчика ("пила"), т.е. выборка начинается со значения 0, а каждое следующее на единицу больше предыдущего. Для циклической передачи данных необходимо, чтобы параметр StartStop был равен 0, иначе, будет заполнен один буфер памяти и передача данных будет остановлена.

```
int SetIQChannel (
```

```
    UCHAR          DevNum,
```

```
    UCHAR          Channel,
```

```
    PIQParameters  IQParam
```

```
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
IQParam	указатель на структуру параметров канала IQ

Таблица 20. Структура параметров канала данных IQ

```
IQParameters {
```

```
    UCHAR    Master;
```

```
    UCHAR    StartStop;
```

```
    UCHAR    Reset;
```

```
    UCHAR    BufferSize;
```

```
    UCHAR    Format;
```

```
    UCHAR    Slave;
```

```
};
```

Master	пересылка данных в режиме Busmaster
StartStop	режим однократной пересылки
Reset	сброс счетчиков

BufferSize	размер пересылаемого блока данных
Format	управление форматом сигнала при записи

Для получения данных IQ-канала в режиме Slave можно воспользоваться функцией ReadIQSlave.

<pre>int ReadIQSlave (UCHAR DevNum, char* DataBuff, DWORD BuffSize, int WaitTimeout_ms);</pre>	
Параметр	Описание
DevNum	номер устройства в системе
DataBuff	буфер данных
BuffSize	размер считываемых данных (размер буфера), байт
WaitTimeout_ms	время накопления данных, мс

Загрузка файла упаковки демодулированных данных при записи в память производится функцией PackMode.

<pre>int PackMode (UCHAR DevNum, char* PackFile);</pre>	
Параметр	Описание
DevNum	номер устройства
PackFile	путь к файлу упаковки

6. ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

Идентификатор	Значение	Описание
E_NO_ERROR	0	Выполнено без ошибок
E_BAD_PARAM	-1	В функцию переданы некорректные параметры
E_IO_ERROR	-2	Ошибка ввода-вывода ¹
E_BAD_DEV_TYPE	-3	Тип устройства не поддерживается
E_BAD_DEC_TYPE	-4	Мезонинный модуль декодера не поддерживается
E_PLL_ERROR	-5	Ошибка настройки PLL
E_OPEN_FILE_ERROR	-6	Невозможно открыть файл ²
E_FPGA_LOAD_ERROR	-7	Ошибка загрузки ПЛИС ³
E_OSPCH_LIB_ERROR	-8	Внутренняя ошибка библиотеки
E_DEC_NOT_LOAD	-9	Плис декодера не загружена ⁴
E_NO_LICENSE	-10	Лицензионный ключ для данного режима не введен или введен неверно

Примечание:

1. При обмене данными с устройством произошла ошибка ввода-вывода.
2. Возможно, файл не существует, занят другим приложением или в функцию передан некорректный путь к файлу.
3. Возможно, осуществляется попытка загрузить ПЛИС файлом конфигурации, не предназначенным для нее.
4. Возможно, выполняется настройка декодера без загрузки ПЛИС.

7. РЕКОМЕНДУЕМЫЙ ПОРЯДОК РАБОТЫ

7.1. Подключение к устройству

7.1.1. Определить число поддерживаемых устройств: *DeviceSupportedCount*

7.1.2. Найти устройство нужного типа, используя для получения имени устройства в системе функцию *GetFullDeviceName*

7.1.3. Инициализировать выбранное устройство: *CreateDeviceInstance*

7.1.4. Для устройств «ОСПЧ-Е1», «ОСПЧ-Е2», «ОСПЧ-Е3» выполнить загрузку синтезатора: *rSynthLoad*

7.1.5. Инициализировать библиотеку управления: *InitLibrary*

7.1.6. Получить для дальнейшего использования базовые параметры устройства: *GetDeviceSlotId*, *GetLConvertorType*, *GetLConvertorVersion*, *GetDecoderVersionOnBoard*

7.1.7. Считать из реестра количество буферов памяти на канал и максимальные значения размеров буферов

7.2. Загрузка демодулятора

7.2.1. Загрузить ПЛИС и ОЗУ демодулятора

Для «ОСПЧ-Е» и «ОСПЧ-М1» необходимо сначала загрузить ПЛИС конфигурацией "load_ram.bit", затем загрузить ОЗУ, далее загрузить ПЛИС рабочей конфигурацией:

```
FPGALoad("load_ram.bit");
```

```
LoadDEMRAM("n_om16.dat");
```

```
FPGALoad("work_configuration_filename.bit");
```

Для «ОСПЧ-Е1», «ОСПЧ-Е2» и «ОСПЧ-Е3» необходимо загрузить ПЛИС рабочей конфигурацией и выполнить загрузку ОЗУ:

FPGALoad("work_configuration_filename.bit");

LoadDEMARAM("n_mch.dat");

7.2.2. При необходимости установить лицензионный ключ: *InstallLicenseKey*

7.2.3. Загрузить АЦП: *ADCLoadMulti*

7.2.4. Для устройств «ОСПЧ-Е1», «ОСПЧ-Е2», «ОСПЧ-Е3» выполнить загрузку синтезатора L-конвертора и чтение таблицы управляющих напряжений:

ISynthInit, InitLConvertorVList

7.2.5. Настроить первую схему РУ: *SetGC1*

7.2.6. Установить центральную частоту: *SetCarrierMulti*

7.2.7. Для каждого канала демодулятора:

7.2.7.1. Считать параметры демодулятора в соответствие с видом модуляции: *GetModParameters*

7.2.7.2. Настроить вторую схему РУ: *SetGC2*

7.2.7.3. Установить параметры демодулятора: *SetDemParam*

7.2.7.4. Установить несущую и тактовую частоту: *SetCarrierFrequency, SetClockFrequency*

7.2.7.5. Загрузить фильтр: *LoadFilter*

7.2.7.6. Выключить и сбросить канал DMD: *SetDMDChannel*

7.3. Загрузка декодера на примере ТРС

7.3.1. Загрузить ПЛИС и ОЗУ декодера

Для мезонинного модуля «Декодер-М1» необходимо сначала загрузить ОЗУ, потом загрузить ПЛИС:

LoadDECGRAM(DeviceNumber, "ext_ram.bit", "en300744.iq", DECGRAM0);

LoadTPCHamming(DeviceNumber, "work_configuration_filename.bit");

Для мезонинных модулей «Декодер-М2», «Декодер-М3» необходимо сначала загрузить ПЛИС, потом загрузить ОЗУ, причем при загрузке ОЗУ параметр FPGA_File игнорируется:

LoadTPCHamming(DeviceNumber, "work_configuration_filename.bit");

LoadDECGRAM(DeviceNumber, NULL, "ram_filename.iq", DECGRAM0);

7.3.2. Сбросить ФАПЧ демодулятора и декодера: *ResetDemDec*

7.3.3. Установить параметры декодера: *LoadTPCHammingReg*,
LoadTPCHammingАНА, *LoadTPCHammingItr*

7.3.4. Сбросить ФАПЧ демодулятора и декодера: *ResetDemDec*

7.3.5. Установить маску каналов: *SetChannelMask*

7.4. Настройка буферов памяти для получения данных

7.4.1. Остановить Busmaster: *MasterStop*

7.4.2. Получить адреса буферов: *MasterGetBuffers*

7.4.3. Установить размер буферов

7.5. Запись сигнала в непрерывном режиме

7.5.1. Остановить Busmaster: *MasterStop*

Примечание: функция MasterLoad должна вызываться при остановленной пересылке данных в режиме Busmaster.

7.5.2. Выполнить загрузку Busmaster: *MasterLoad*

7.5.3. Запустить Busmaster: *MasterStart*

7.5.4. Инициализировать чтение данных

Подготовить поток (thread) получения данных. В потоке использовать событие с именем "Global\\dgsprm[deviceslot]evnt[channel]", где [deviceslot] – слот, полученный ранее, [channel] – номер канала DMA.

7.5.5. Включить передачу данных: *SetDMChannel* или *SetIQChannel*

7.5.6. Считывать данные в ранее инициализированном потоке

7.5.7. По окончании ввода данных необходимо выключить передачу данных и остановить Busmaster: *SetDMChannel* или *SetIQChannel*, *MasterStop*

Примечание: перед вводом данных из устройства необходимо убедиться, что драйвер настроен на необходимое количество каналов. Если это не так (как правило, по умолчанию для одноканальной версии используется 4 канала), необходимо выполнить настройку каналов в реестре и перезагрузить компьютер.

7.6. Завершение работы

7.6.1. Деинициализировать библиотеку управления: *ReleaseLibrary*

7.6.2. Отключиться от устройства: *ReleaseDeviceInstance*

8. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

АПЧ	автоматическая подстройка частоты
АРУ	автоматическая регулировка усиления
ОС	операционная система
ОЗУ	оперативное запоминающее устройство
ОСШ	отношение сигнал-шум
ПО	программное обеспечение
ПЭВМ	персональная электронно-вычислительная машина
РРУ	ручная регулировка усиления
РУ	регулировка усиления
ФАПЧ	фазовая автоподстройка частоты