

МОДУЛИ «ОСПЧ»

БИБЛИОТЕКА ФУНКЦИЙ. МНОГОКАНАЛЬНЫЙ РЕЖИМ TDMA

Руководство программиста

Листов 38

2020

## СОДЕРЖАНИЕ

1. Общие сведения	3
2. Общие функции	4
2.1. Инициализация и освобождение ресурсов	4
2.2. Описание последней ошибки	4
2.3. Установка режима TDMA	5
2.4. Сброс демодулятора и декодера	5
2.5. Ввод лицензионного ключа	6
2.6. Установка маски каналов	7
2.7. Считывание служебных параметров	7
3. Функции управления демодулятором	9
3.1. Настройка частот	9
3.2. Управление схемами регулировки усиления	10
3.3. Загрузка фильтра	13
3.4. Настройка ФАПЧ	13
3.5. Управление коррелятором	14
3.6. Считывание параметров демодулятора	17
4. Функции управления декодером	19
4.1. Загрузка параметров декодера режимов TDMA-MCH, TDMA-SW	19
4.2. Загрузка параметров декодера режима TDMA-DW	20
4.3. Загрузка параметров декодера режимов TDMA-ID	20
4.4. Загрузка параметров декодера режима A-TDMA	23
4.5. Загрузка ОЗУ декодера	24
4.6. Установка инверсии спектра	24
4.7. Считывание параметров декодера	25
5. Функции управления каналами данных	27
6. Возвращаемые значения	31

7. Рекомендуемый порядок работы	33
7.1. Подключение к устройству	33
7.2. Загрузка демодулятора	33
7.3. Загрузка декодера на примере TDMA-ID-Inf	35
7.4. Настройка буферов памяти для получения данных	35
7.5. Запись сигнала в непрерывном режиме	35
7.6. Завершение работы	36
8. Перечень сокращений	37

## 1. ОБЩИЕ СВЕДЕНИЯ

Комплект «libtdma SDK» предназначен для разработки ПО управления устройствами «ОСПЧ-Е3», «ОСПЧ-Е2», «ОСПЧ-Е1», «ОСПЧ-Е» и «ОСПЧ-М1» в режимах: TDMA-MCH, TDMA-SW, TDMA-DW, TDMA-ID-Evo, TDMA-ID-Inf, A-TDMA.

В комплект «libtdma 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. Установка режима TDMA

Установка режима TDMA производится функцией SetTDMAMode. Данную функцию необходимо вызвать перед загрузкой ПЛИС демодулятора.

```
int SetTDMAMode (
    unsigned char    DevNum,
    TDMA_MODE       TdmaMode
);
```

Параметр	Описание
DevNum	номер устройства
TdmaMode	режим TDMA

### 2.4. Сброс демодулятора и декодера

После загрузки демодулятора и декодера выполняется сброс схем ФАПЧ. Для этого необходимо использовать функции ResetDEM и ResetDEC. В режимах TDMA-ID-Evo, TDMA-ID-Inf, A-TDMA требуется дополнительный сброс ФАПЧ декодера с проверкой статуса ФАПЧ функцией checkDecPll. При успешном захвате ФАПЧ функция checkDecPll вернет E\_NO\_ERROR (0).

```
int ResetDem (
    UCHAR    DevNum
);
```

Параметр	Описание
DevNum	номер устройства

```
int ResetDec (
```

<pre>         UCHAR    DevNum     );     </pre>	
Параметр	Описание
DevNum	номер устройства

<pre> int checkDecPll (     UCHAR    DevNum );     </pre>	
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. Для устройств «ОСПЧ-Е», «ОСПЧ-М1», и режима TDMA-DW ввод ключа выполняется сразу после загрузки ПЛИС демодулятора. Для всех остальных режимов и устройств ввод ключа осуществляется после загрузки ПЛИС декодера.

<pre> int InstallLicenseKey (     UCHAR    DevNum,     PINT64   LicenseKey );     </pre>	
Параметр	Описание
DevNum	номер устройства
LicenseKey	лицензионный ключ

## 2.6. Установка маски каналов

Включение или отключение обработки данных, а также записи сигнала в файл для каналов производится заданием маски каналов при помощи функции `SetChannelMask`. Порядковый номер канала соответствует выставленному биту 32-разрядного числа `Mask`.

```
int SetChannelMask (
    UCHAR    DevNum,
    DWORD    Mask
);
```

Параметр	Описание
DevNum	номер устройства
Mask	маска каналов

## 2.7. Считывание служебных параметров

В некоторых случаях для идентификации устройства требуется считать его DNA-код. Для этого предназначена функция `getDna`.

```
int getDna (
    UCHAR        DevNum,
    unsigned int* dnaH,
    unsigned int* dnaL
);
```

Параметр	Описание
DevNum	номер устройства
dnaH	указатель на переменную, содержащую старшие 32 бита кода DNA
dnaL	указатель на переменную, содержащую младшие 32 бита кода DNA



Узнать версию установленного мезонинного модуля можно с помощью функции `GetDecoderVersionOnBoard`.

```
int GetDecoderVersionOnBoard (  
    UCHAR          DevNum,  
    DECODER_VERSION* DecVersion  
);
```

Параметр	Описание
DevNum	номер устройства
DecVersion	указатель на переменную, содержащую версию мезонинного модуля, установленного в устройство

### 3. ФУНКЦИИ УПРАВЛЕНИЯ ДЕМОДУЛЯТОРОМ

#### 3.1. Настройка частот

Установка центральной частоты выполняется функцией SetCentralFreq.

<pre>int SetCentralFreq (     unsigned char    DevNum,     double           Frequency );</pre>	
Параметр	Описание
DevNum	номер устройства
Frequency	центральная частота, Гц

Установка несущей частоты выполняется функцией SetCarrierFreq.

<pre>int SetCarrierFreq (     unsigned char    DevNum,     unsigned char    Channel,     double           Frequency );</pre>	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
Frequency	несущая частота, Гц

Для установки тактовой частоты необходимо использовать функцию SetClockFreq.

<pre>int SetClockFreq (     unsigned char    DevNum,     unsigned char    Channel,     double           Frequency );</pre>	
--	--

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

Настройка несущей и тактовой частот выполняется для каждого канала демодулятора. Частоты задаются в герцах (Гц). Установка несущей частоты должна выполняться после установки центральной частоты. Несущие частоты для всех каналов не должны отличаться от тактовой частоты более чем на половину полосы L-конвертора, то есть на 36 МГц или 55МГц (в зависимости от применяемого типа L-конвертора).

### 3.2. 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 (APU) и ручном режиме (PPU). Для настройки первой схемы РУ необходимо использовать функцию SetGC1. Управление второй схемой РУ производится функцией SetGC2.

<pre>int SetGC1 (     UCHAR          DevNum,     GC1Ctrl*       GC1Param );</pre>	
Параметр	Описание
DevNum	номер устройства
GC1Param	указатель на структуру параметров первой схемы РУ

Таблица 1. Структура параметров первой схемы РУ

<pre>GC1Control {     WORD      Gain;     UCHAR     Manual;     UCHAR     Att; };</pre>	
Gain	нижняя или верхняя граница "нечувствительности" АРУ
Manual	включение ручного режима управления схемой РУ
Att	код управления аттенюатором при РРУ

<pre>int SetGC2 (     UCHAR      DevNum,     UCHAR      Channel,     GC2Ctrl*   GC2Param );</pre>	
<b>Параметр</b>	<b>Описание</b>
DevNum	номер устройства
Channel	номер канала обработки данных
GC2Ctrl	указатель на структуру параметров второй схемы РУ

Таблица 2. 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,
    PGCVals  GCVal
);
```

Параметр	Описание
DevNum	номер устройства
Channel	номер канала обработки данных
GCVal	указатель на структуру параметров РУ

Таблица 3. Структура параметров схем РУ

```
GCValues {
    double    ARU1Hard;
    double    ARU1Soft;
    DWORD     ARU1_YE;
    DWORD     ARU2_YE;
    double    ARU2_dB;
    bool      GC1Overload;
};
```

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

GC1Overload	статус перегрузки схемы РУ
-------------	----------------------------

### 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 с коэффициентами фильтра

### 3.4. Настройка ФАПЧ

Настройка ФАПЧ производится функцией SetPLL.

```
int SetPLL(
    UCHAR    DevNum,
    PPLLCtrl PLLParam);
```

Параметр	Описание
DevNum	номер устройства
PLLParam	параметры ФАПЧ

Таблица 4. Структура параметров ФАПЧ

```
PLLCtrl{
    UCHAR    Channel;
```

PLL_BAND	PLLBand;
bool	CarrierTracking;
bool	ClockTracking;
SHORT	PreambleLength;
SHORT	SlotLen;
SIGNAL_TYPE	iDirSignalType;
};	
Channel	номер канала
PLLBand	ширина полосы ФАПЧ
CarrierTracking	слежение за несущей
ClockTracking	слежение за тактовой
PreambleLength	длина преамбулы
SlotLen	длина слота
iDirSignalType	вид модуляции (только для режима TDMA-ID)

### 3.5. Управление коррелятором

Загрузка параметров коррелятора требуется во всех режимах. Для режимов TDMA-MCH, TDMA-SW, A-TDMA используется функция LoadCorrelator. В режимах TDMA-ID настройка осуществляется функцией loadIDirDemCorrelator. Для настройки в режиме TDMA-DW используются функции LoadParam\_DW\_Auto\_ClassB и LoadParam\_DW\_SyncParam. Перед загрузкой параметров коррелятора требуется выполнить его инициализацию функцией InitDemCorrelator.

int InitDemCorrelator ( unsigned char    DevNum );	
<b>Параметр</b>	<b>Описание</b>
DevNum	номер устройства

```
int LoadCorrelator (
    unsigned char    DevNum,
    CorrelatorCtrl* CorrelatorParam
);
```

Параметр	Описание
DevNum	номер устройства
CorrelatorParam	параметры коррелятора

Таблица 5. Параметры коррелятора режимов TDMA-MCH, TDMA-SW, A-TDMA

```
CorrelatorCtrl {
    UCHAR    Channel;
    DWORD    IWord;
    DWORD    QWord;
    DWORD    Mask;
    DEC_SPEED DecSpeed;
    DWORD    PacketLength;
    DWORD    Treshold;
};
```

Channel	номер канала
IWord	уникальное слово I
QWord	уникальное слово Q
Mask	маска уникального слова
DecSpeed	скорость декодера
PacketLength	длина пакета
Treshold	порог коррелятора

```
int loadIDirDemCorrelator (
    unsigned char    DevNum,
```



IDirDemCorrelatorParam* CorrelatorParam );	
Параметр	Описание
DevNum	номер устройства
CorrelatorParam	параметры коррелятора

Таблица 6. Параметры коррелятора режимов TDMA-ID

IDirDemCorrelatorParam {	
char	uw[IDIR_MAX_UW_LEN];
int	uwLength;
unsigned int	uwI;
unsigned int	uwQ;
bool	useIQUW;
int	preambleLength;
int	burstLength;
DWORD	correlatorThreshold;
SIGNAL_TYPE	modulation;
};	
Channel	номер канала
uw	уникальное слово в строковом формате
uwLength	длина уникального слова в строковом формате
uwI	уникальное слово I
uwQ	уникальное слово Q
useIQUW	выбор типа уникального слова (0 - строковый формат, 1- формат IQ)
preambleLength	длина преамбулы
burstLength	длина пакета
correlatorThreshold	порог коррелятора
modulation	вид модуляции

```
int LoadParam_DW_Auto_ClassB (
    unsigned char      DevNum
);
```

Параметр	Описание
DevNum	номер устройства

```
int LoadParam_DW_SyncParam (
    unsigned char      DevNum,
    unsigned char      channel,
    double             dt,
    double             clock
);
```

Параметр	Описание
DevNum	номер устройства
channel	номер канала
dt	время запаздывания сигналов, мс
clock	тактовая частота, Гц

### 3.6. Считывание параметров демодулятора

При работе с «ОСПЧ» может понадобиться считать состояние демодулятора. Считать статус синхронизации в режимах TDMA-MCH, TDMA-SW можно с помощью функции ReadStatus\_MCH\_SW.

```
int ReadDEMStatus (
    UCHAR      DevNum,
    PStatus_MCH_SW      PState
);
```

Параметр	Описание
DevNum	номер устройства

PState	указатель на структуру статуса синхронизации в режимах TDMA-MCH, TDMA-SW
--------	--

Таблица 7. Структура параметров статуса синхронизации в режимах TDMA-MCH, TDMA-SW

<pre>Status_MCH_SW{     bool Sync[CHANNEL_COUNT_MCH_SW]; };</pre>	
Sync	статус синхронизации

В режимах TDMA-DW, TDMA-ID-Evo, TDMA-ID-Inf, A-TDMA для считывания статуса синхронизации предусмотрена функция ReadStatus\_DW.

<pre>int ReadStatus_DW (     UCHAR      DevNum,     PStatus_DW PState );</pre>	
Параметр	Описание
DevNum	номер устройства
PState	указатель на структуру статуса синхронизации в режимах TDMA-DW, TDMA-ID-Evo, TDMA-ID-Inf, A-TDMA

Таблица 8. Структура параметров статуса синхронизации в режимах TDMA-DW, TDMA-ID-Evo, TDMA-ID-Inf, A-TDMA

<pre>PStatus_DW {     struct {         bool CHSync;         bool UWSync[UW_COUNT_DW];     }Channel[MAX_CHANNEL_COUNT]; };</pre>	
CHSync	синхронизация канала
UWSync	синхронизация уникального слова

#### 4. ФУНКЦИИ УПРАВЛЕНИЯ ДЕКОДЕРОМ

Для загрузки декодера необходимо загрузить ПЛИС и задать параметры декодера. Для загрузки файла конфигурации ПЛИС используется функция `FPGALoad` библиотеки `osrch`. Ввод параметров декодера осуществляется функциями библиотеки `libtdma`. Для режимов TDMA-ID-Evo и TDMA-ID-Inf после загрузки ПЛИС требуется выполнить загрузку ОЗУ мезонинного модуля.

##### 4.1. Загрузка параметров декодера режимов TDMA-MCH, TDMA-SW

Ввод параметров декодера режимов TDMA-MCH, TDMA-SW осуществляется функцией `LoadDECReg_MCH_SW`.

<pre>int LoadDECReg_MCH_SW (     UCHAR          DevNum,     PDECtrl_MCH_SW DecParam );</pre>	
Параметр	Описание
DevNum	номер устройства
DecParam	указатель на структуру параметров декодера TDMA-MCH, TDMA-SW

Таблица 9. Структура параметров декодера TDMA-MCH, TDMA-SW

<pre>DECtrl_MCH_SW{     DWORD    Modulation;     DWORD    Scrambler;     {         DEC_SPEED    DecSpeed;         DWORD        PacketLength;     }Channel[MAX_CHANNEL_COUNT]; };</pre>	
Modulation	вид модуляции

Scrambler	значение скремблера
DecSpeed	скорость декодера
PacketLength	длина пакета

#### 4.2. Загрузка параметров декодера режима TDMA-DW

Ввод параметров декодера TDMA-DW производится функцией LoadParam\_DW\_DecParam.

```
int LoadParam_DW_DecParam (
    UCHAR    DevNum,
    int      allPacketLength,
    UCHAR    scramblerEnable,
    UCHAR    header
);
```

Параметр	Описание
DevNum	номер устройства
allPacketLength	длина пакета
scramblerEnable	включение скремблера
header	заголовок

#### 4.3. Загрузка параметров декодера режимов TDMA-ID

Установка параметров декодера режима TDMA-ID-Evo производится функцией loadIDirEvolutionDecParam.

```
int loadIDirEvolutionDecParam (
    UCHAR    DevNum,
    UCHAR    channel,
    IDirEvolutionDecParam* param
);
```

Параметр	Описание
----------	----------

DevNum	номер устройства
channel	номер канала
param	указатель на структуру параметров декодера TDMA-ID-Evo

Таблица 10. Структура параметров декодера TDMA-ID-Evo

<pre> IDirEvolutionDecParam {     int          payloadLengthBytes;     int          yPuncturePeriod;     unsigned int yPuncturePattern;     int          p;     int          q0;     int          q1;     int          q2;     int          q3;     int          pilotPeriod;     int          pilotBlock;     SIGNAL_TYPE  modulation;     int          infPilot1;     int          infPilot2;     int          infPilot3;     int          infPilot4; }; </pre>	
payloadLengthBytes	размер информационной части пакета
yPuncturePeriod	период выкалывания проверочной части
yPuncturePattern	маска выкалывания проверочной части
p	параметр "P" перемежителя
q0	параметр "Q0" перемежителя
q1	параметр "Q1" перемежителя
q2	параметр "Q2" перемежителя
q3	параметр "Q3" перемежителя

pilotPeriod	период повторения пилот-сигналов
pilotBlock	размер блока пилот-сигналов
modulation	вид модуляции
infPilot1	размер информационной части до 1-го пилот-сигнала
infPilot2	размер информационной части до 2-го пилот-сигнала
infPilot3	размер информационной части до 3-го пилот-сигнала
infPilot4	размер информационной части до 4-го пилот-сигнала

Параметры infPilot1, infPilot2, infPilot3, infPilot4 актуальны только для вида модуляции ФМ-8.

При использовании режима TDMA-ID-Inf, настройку параметров следует производить функцией loadIDirInfinityDecParam.

<pre>int loadIDirInfinityDecParam (     UCHAR          DevNum,     UCHAR          channel,     IDirInfinityDecParam* param );</pre>	
Параметр	Описание
DevNum	номер устройства
channel	номер канала
param	указатель на структуру параметров декодера TDMA-ID-Inf

Таблица 11. Структура параметров декодера TDMA-ID-Inf

<pre>IDirInfinityDecParam{     SIGNAL_TYPE modulation;     INFINITY_MODE mode; };</pre>	
modulation	вид модуляции
mode	режим декодера TDMA-ID-Inf

#### 4.4. Загрузка параметров декодера режима A-TDMA

Ввод параметров декодера режима A-TDMA производится функцией loadATdmaDecParam.

<pre>int loadIDirInfinityDecParam (     UCHAR          DevNum,     UCHAR          channel,     SIGNAL_TYPE     modulation,     ATDMADecParam* param );</pre>	
Параметр	Описание
DevNum	номер устройства
channel	номер канала
modulation	вид модуляции
param	указатель на структуру параметров декодера A-TDMA

Таблица 12. Структура параметров декодера A-TDMA

<pre>IDirInfinityDecParam{     ATDMA_PAYLOAD_SIZE    payloadSize;     ATDMA_SPEED           speed;     ATDMA_PILOT_SIZE      pilotSize;     unsigned char         pilotPeriod; };</pre>	
payloadSize	размер информационной части пакета
speed	кодовая скорость
pilotSize	размер блока пилот-сигналов
pilotPeriod	период повторения пилот-сигналов



#### 4.5. Загрузка ОЗУ декодера

В режимах TDMA-ID-Evo, TDMA-ID-Inf, A-TDMA после загрузки ПЛИС декодера требуется выполнить загрузку ОЗУ декодера. Для этого предназначена функция loadDecRam.

<pre>int LoadDECRAM (     UCHAR      DevNum,     UCHAR      ramNum,     char*      ramFile );</pre>	
Параметр	Описание
DevNum	номер устройства в системе
FPGA_File	путь к файлу конфигурации ПЛИС для загрузки ОЗУ
ramNum	номер ОЗУ декодера
ramFile	путь к файлу, загружаемому в ОЗУ декодера

#### 4.6. Установка инверсии спектра

Для всех режимов кроме TDMA-DW доступно включение инверсии спектра. Для этого предназначена функция SpectrInversion\_MCH\_SW.

<pre>int SpectrInversion_MCH_SW (     UCHAR      DevNum,     bool       SpectrInversion );</pre>	
Параметр	Описание
DevNum	номер устройства
SpectrInversion	управление инверсией спектра

#### 4.7. Считывание параметров декодера

В процессе работы с устройством может потребоваться информация о состоянии декодера. Считывание вероятности битовой ошибки на выходе декодера производится функцией ReadBER.

<pre>int ReadBER (     UCHAR    DevNum,     UCHAR    Channel,     double*  BER );</pre>	
Параметр	Описание
DevNum	номер устройства
Channel	номер канала
BER	указатель на переменную, содержащую значение BER

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 13. Структура параметров температуры декодера

<pre>DECTemperature {     int    FPGA_DEC_Temperature;</pre>
--

bool FPGA_DEC_Overheat; };	
FPGA_DEC_Temperature	температура ПЛИС декодера
FPGA_DEC_Overheat	статус перегрева ПЛИС декодера

## 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.

В режимах TDMA-ID, A-TDMA, а также TDMA-MCH, TDMA-SW, TDMA-DW для «ОСПЧ-ЕЗ» с типом L-конвертора, выше или равным 5, каждый канал обработки данных демодулятора имеет свой канал DMA передачи демодулированных и декодированных данных (DMD). При этом номер канала DMA соответствует номеру канала обработки сигнала демодулятора, умноженному на 2. В остальных случаях, данные всех каналов обработки данных передаются по нулевому каналу DMA.

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 АЦП или IQ номер DMA канала равен 1.

Настройка DMD-канала для одноканального DMA режима выполняется функцией SetDMDChannel. В многоканальном DMA режиме используется функция SetIDirDMDChannel.

```
int SetDMDChannel (
    UCHAR    DevNum,
    PDMDCtrl DMDParam
);
```

Параметр	Описание
DevNum	номер устройства

DMDParam	указатель на структуру параметров канала DMD
----------	--

Таблица 14. Структура параметров канала данных DMD

<pre>DMDParameters {     UCHAR    Format;     UCHAR    Master;     UCHAR    BufferSize;     UCHAR    Slave; };</pre>	
Format	управление форматом сигнала при записи (для данных с выхода декодера значение Format = 0)
Master	пересылка данных в режиме Busmaster
BufferSize	размер пересылаемого блока данных
Slave	пересылка данных в режиме Slave

<pre>int SetIDirDMDChannel (     UCHAR    DevNum,     UCHAR    Channel,     PDMDCtrl DMDParam );</pre>	
<b>Параметр</b>	<b>Описание</b>
DevNum	номер устройства
Channel	номер канала
DMDParam	указатель на структуру параметров канала DMD

Таблица 15. Структура параметров канала данных DMD

<pre>IDirDMDCtrl{     SIGNAL_TYPE    modulation;     IDIR_DMD_FORMAT    format;</pre>	
---	--

bool	transferEnable;
bool	reverseBits;
};	
modulation	вид модуляции
format	управление форматом сигнала при записи
transferEnable	включение пересылки данных по DMA каналу
reverseBits	разворот бит в байте

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,
PIQCtrl	IQParam
);	
<b>Параметр</b>	<b>Описание</b>
DevNum	номер устройства
IQParam	указатель на структуру параметров канала IQ

Таблица 16. Структура параметров канала данных IQ

<pre> IQParameters {     bool      Master;     bool      StartStop;     bool      Reset;     UCHAR     BufSize;     UCHAR     Channel;     UCHAR     Format;     bool      SpectrumMode; }; </pre>	
Master	пересылка данных в режиме Busmaster
StartStop	режим однократной пересылки
Reset	сброс счетчиков
BufferSize	размер пересылаемого блока данных
Channel	номер канала
Format	управление форматом сигнала при записи
SpectrumMode	включение режима данных для спектра

## 6. ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

Идентификатор		Описание
E_NO_ERROR	0	Выполнено без ошибок
E_BAD_PARAM	-1	В функцию переданы некорректные параметры
E_IO_ERROR	-2	Ошибка ввода-вывода <sup>1</sup>
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 <sup>2</sup>
E_FPGA_LOAD_ERROR	-7	Ошибка загрузки ПЛИС <sup>3</sup>
E_OSPCH_LIB_ERROR	-8	Внутренняя ошибка библиотеки
E_DEC_NOT_LOAD	-9	Плис декодера не загружена <sup>4</sup>
E_NO_LICENSE	-10	Лицензионный ключ для данного режима не введен или введен неверно
E_UNSUPPORTED_MODE	-11	Режим не поддерживается
E_DEC_PLL_NOT_LOCKED	-12	Нет захвата PLL декодера
E_SYSTEM_ERROR	-13	Внутренняя ошибка
E_UNSUPPORTED_MODULATION	-14	Вид модуляции не поддерживается
E_INCORRECT_UW	-15	Ошибка в уникальном слове
E_DNA_MODE_NOT_SUPPORTED	-16	DNA-код не поддерживается на этом устройстве
E_DNA_NOT_INITED	-17	DNA-код не установлен в устройстве
E_DNA_ERROR	-18	Ошибка считывания DNAкода

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. Установить режим TDMA: *SetTDMAMode*

7.2.2. Остановить DMA мастер: *MasterStop*

7.2.3. Загрузить ПЛИС демодулятора: *FPGALoad*

7.2.4. При необходимости установить лицензионный ключ: *InstallLicenseKey*

7.2.5. Загрузить АЦП: *ADCLoadMulti*

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

7.2.7. Установить центральную частоту: *SetCentralFreq, SetCarrierMulti*

7.2.8. Настроить первую схему РУ: *SetGC1*

7.2.9. Инициализировать коррелятор: *InitDemCorrelator*

7.2.10. Для каждого канала демодулятора:

7.2.10.1. Настроить вторую схему РУ: *SetGC2*

7.2.10.2. Установить несущую частоту: *SetCarrierFreq*

7.2.11. Для каждого канала демодулятора:

7.2.11.1. Установить тактовую частоту: *SetClockFreq*

7.2.11.2. Загрузить фильтр: *LoadFilter*

7.2.11.3. Для режима TDMA-DW на устройствах «ОСПЧ-ЕЗ» - ввести параметры синхронизатора: *LoadParam\_DW\_SyncParam*

7.2.12. Загрузить коррелятор для каждого канала демодулятора:

7.2.12.1. Для режимов TDMA-MCH, TDMA-SW, A-TDMA: *LoadCorrelator*

7.2.12.2. Для режимов TDMA-ID: *loadIDirDemCorrelator*

7.2.12.3. Для режима TDMA-DW - выполнить автоматическую загрузку: *LoadParam\_DW\_Auto\_ClassB*

7.2.13. Для каждого канала демодулятора:

7.2.13.1. Настроить PLL: *SetPLL*

7.2.13.2. Для режима TDMA-DW на устройствах «ОСПЧ-ЕЗ» - ввести параметры синхронизатора: *LoadParam\_DW\_SyncParam*

### **7.3. Загрузка декодера на примере TDMA-ID-Inf**

7.3.1. Загрузить ПЛИС декодера: *FPGALoad*

7.3.2. Ввести лицензионный ключ: *InstallLicenseKey*

7.3.3. Сбросить ФАПЧ декодера: *ResetDEC*

7.3.4. Проверить захват ФАПЧ: *checkDecPll*

7.3.5. Загрузить ОЗУ декодера: *loadDecRam*

7.3.6. Ввести параметры декодера режима IDMA-ID-Inf для каждого канала:  
*loadIDirInfinityDecParam*

7.3.7. Настроить инверсию спектра: *SpectrInversion\_MCH\_SW*

7.3.8. Сбросить ФАПЧ декодера: *ResetDEC*

7.3.9. Сбросить ФАПЧ демодулятора: *ResetDEM*

7.3.10. 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\\dgspm[deviceslot]evnt[channel]", где [deviceslot] – слот, полученный ранее, [channel] – номер канала DMA.

7.5.5. Включить передачу данных: *SetDMDChannel*, *SetIDirDMDChannel*, *SetIQChannel*

7.5.6. Считывать данные в ранее инициализированном потоке

7.5.7. По окончании ввода данных необходимо выключить передачу данных и остановить Busmaster: *SetDMDChannel*, *SetIDirDMDChannel*, *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