

МНОГОКАНАЛЬНЫЕ СИСТЕМЫ СБОРА ДАННЫХ

LTR35

РУКОВОДСТВО ПРОГРАММИСТА

*Ревизия 1.1.1
Ноябрь 2020*

Автор руководства:

Борисов Алексей

ООО “Л Кард”

117105, г. Москва, Варшавское ш., д. 5, корп. 4, стр. 2

тел.: +7 (495) 785-95-25

факс: +7 (495) 785-95-14

Адреса в Интернет:

<http://www.lcard.ru>

E-Mail:

Отдел продаж: sale@lcard.ru

Техническая поддержка: support@lcard.ru

Отдел кадров: job@lcard.ru

Таблица 1: Ревизии текущего документа

Ревизия	Дата	Описание
1.0.0	10.09.2014	Первая ревизия данного документа, описывающая API для прототипа модуля
1.1.0	29.10.2020	Описания обновлено до финального пользовательского API серийного модуля. Описаны новые функции синхронного ввода и многомодульной синхронизации старта.
1.1.1	26.11.2020	Изменен способ задания фазы для арифметических генераторов.

Оглавление

1	О чем этот документ	6
2	Установка и подключение библиотеки к проекту	7
3	Общий подход к работе с библиотекой	8
3.1	Общий алгоритм работы с модулем	8
3.1.1	Вывод сигнала при режиме циклического автогенератора	8
3.1.2	Вывод сигнала при потоковом режиме	9
3.2	Настройка модуля	9
3.2.1	Настройка частоты генерации ЦАП и вывода на цифровые линии	10
3.2.2	Настройка каналов ЦАП	11
3.2.3	Настройка арифметических генераторов	11
3.2.4	Настройка формата передаваемых данных	12
3.2.5	Настройка синхронного ввода	13
3.2.6	Настройка синхронизации нескольких модулей	14
3.3	Режимы вывода	15
3.3.1	Потоковый режим вывода	15
3.3.2	Режим циклического автогенератора	17
3.4	Прием данных при синхронном вводе	18
3.5	Синхронизация запуска нескольких модулей	19
3.6	Особенности калибровки данных	20
4	Константы, типы данных и функции библиотеки	21
4.1	Константы и перечисления	21
4.1.1	Константы и макроопределения	21
4.1.2	Специфичные для LTR35 коды ошибок	23
4.1.3	Модификации модуля LTR35	24
4.1.4	Флаги для управления цифровыми выходами	25
4.1.5	Флаги для подготовки данных	25
4.1.6	Формат отсчетов для передачи данных вывода модулю	25
4.1.7	Режим вывода модуля	25
4.1.8	Используемый выход для канала ЦАП	26
4.1.9	Источники сигнала для каналов ЦАП	26
4.1.10	Скорость выдачи отсчетов ЦАП	27
4.1.11	Режим синхронного ввода	27
4.1.12	Флаги задания цифровых линий для синхронного ввода	28
4.1.13	Режим синхронизации старта генерации модулей	28
4.1.14	Условие запуска генерации для подчиненного модуля	29
4.1.15	Флаги конфигурации модуля	29
4.1.16	Флаги для записи во flash-память модуля	29

4.1.17	Флаги состояния модуля	30
4.2	Типы данных	30
4.2.1	Калибровочные коэффициенты	30
4.2.2	Описание выхода ЦАП	30
4.2.3	Коэффициенты для калибровки АЧХ	31
4.2.4	Информация о модуле	31
4.2.5	Настройки канала ЦАП	32
4.2.6	Значение фазы для арифметических генераторов	32
4.2.7	Настройки арифметического генератора	33
4.2.8	Настройки синтезатора частоты	33
4.2.9	Настройки синхронного ввода	33
4.2.10	Настройки синхронизации старта	34
4.2.11	Настройки модуля	34
4.2.12	Параметры текущего состояния модуля	35
4.2.13	Управляющая структура модуля	37
4.3	Функции	37
4.3.1	Функции инициализации и работы с соединением с модулем	37
4.3.1.1	Инициализация описателя модуля	37
4.3.1.2	Открытие соединения с модулем	38
4.3.1.3	Закрытие соединения с модулем	38
4.3.1.4	Проверка, открыто ли соединение с модулем	39
4.3.2	Функции для изменения настроек модуля	39
4.3.2.1	Подбор коэффициентов для получения заданной частоты генерации ЦАП и цифрового вывода	39
4.3.2.2	Подбор степени делителя частоты ввода с цифровых линий	40
4.3.2.3	Подбор кода арифметической фазы по значению в градусах	40
4.3.2.4	Запись настроек в модуль	41
4.3.3	Функции для управления выдачей данных	41
4.3.3.1	Передача данных ЦАП и цифровых выходов в модуль	41
4.3.3.2	Подготовка данных для передачи в модуль	42
4.3.3.3	Подготовка данных ЦАП для передачи в модуль	43
4.3.3.4	Смена страницы вывода в режиме циклического автогенератора	44
4.3.3.5	Запрос смены страницы вывода в режиме циклического автогенератора	44
4.3.3.6	Ожидание завершения смены страницы в режиме циклического автогенератора	45
4.3.3.7	Запуск ввода-вывода в потоковом режиме	45
4.3.3.8	Запрос на запуск ввода-вывода в потоковом режиме	46
4.3.3.9	Ожидание запуска ввода-вывода в потоковом режиме	46
4.3.3.10	Останов выдачи данных	47
4.3.3.11	Останов выдачи данных с заданным временем ожидания ответа	47
4.3.3.12	Изменение частоты для арифметического генератора	48
4.3.3.13	Изменение амплитуды и смещения арифметического сигнала	48
4.3.3.14	Прием данных входного потока от модуля	49
4.3.4	Функции для работы с flash-памятью модуля	49
4.3.4.1	Чтение данных из flash-памяти модуля	49

4.3.4.2	Запись данных во flash-память модуля	50
4.3.4.3	Стирание области flash-память модуля	50
4.3.5	Функции вспомогательного характера	51
4.3.5.1	Получение сообщения об ошибке	51
4.3.5.2	Проверка, разрешена ли работа ПЛИС модуля	51
4.3.5.3	Разрешение работы ПЛИС модуля	51
4.3.5.4	Получение информации о состоянии модуля	52
4.3.5.5	Чтение информации и калибровочных коэффициентов	52
4.3.5.6	Сброс микросхемы ЦАП	52
4.3.5.7	Асинхронное чтение состояния цифровых входов	53

Глава 1

О чем этот документ

Данный документ предполагает, что пользователь знаком с документами [‘Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению.’](#) и [‘Программное обеспечение для системы LTR’](#), в которых описываются основные принципы работы программного обеспечения для крейтов LTR.

Данный документ предназначен в первую очередь для программистов, которые собираются писать свои программы для работы с модулем LTR35 с использованием предоставляемой фирмой “Л Кард” библиотеки ltr35api.

В данном документе рассматривается вопрос подключения библиотеки к проекту пользователя, дается подробное описание интерфейсных функций, предоставляемых библиотекой, и используемых типов, а также дается описание основных подходов к использованию этих функций.

Сама библиотека написана на языке *C* и все объявления функций и типов приводятся на языке *C*. Однако все привязки к другим языкам программирования являются лишь обертками над библиотекой *C* и все функции, типы и параметры сохраняют свои значения и для других языков программирования. Поэтому этот документ полезен и пользователям, пишущим на других языках программирования.

В настоящем документе не рассматриваются какие-либо вопросы, касающиеся характеристик модуля и подключения сигналов, а также лишь в общем затрагиваются принципы работы самого модуля. Перечисленные вопросы рассматриваются в соответствующем разделе документа [‘Крейтовая система LTR. Руководство пользователя’](#), с которым необходимо ознакомиться перед прочтением данного документа.

Глава 2

Установка и подключение библиотеки к проекту

Использование библиотек для работы с крейтовой системой LTR описано в документе [‘Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению.’](#).

Глава 3

Общий подход к работе с библиотекой

3.1 Общий алгоритм работы с модулем

Данный раздел описывает типичную последовательность действий при работе с модулем LTR35. Более подробно каждый шаг будет описан в последующих разделах.

Типичная последовательность действий имеет следующий вид:

1. Создать экземпляр структуры `TLTR35`, представляющей собой описатель модуля. Описатель модуля содержит всю информацию о модуле и используется при вызове всех остальных функций.
2. Проинициализировать поля описателя с помощью `LTR35_Init()`.
3. Установить соединение с интересующим модулем с помощью функции `LTR35_Open()`.
4. Заполнить все необходимые настройки модуля в поле `Cfg` описателя модуля и вызвать `LTR35_Configure()` для записи настроек в модуль.
5. Подготовка и вывод сигналов в соответствии с установленным режимом вывода модуля, как описано ниже.
6. По завершению работы выполнить останов генерации сигналов с помощью `LTR35_Stop()` или `LTR35_StopWithTout()`.
7. Закрыть соединение с модулем, вызвав функцию `LTR35_Close()`.

3.1.1 Вывод сигнала при режиме циклического автогенератора

Для вывода периодических сигналов в режиме циклического автогенератора нужно выполнить следующие шаги:

1. Подготовить данные для выдачи в нужном формате с помощью `LTR35_PrepareData()` или `LTR35_PrepareDacData()`.
2. Передать подготовленные данные в модуль с помощью `LTR35_Send()`.
3. При необходимости повторить пункты 1 и 2 нужное количество раз, чтобы передать в модуль все отсчеты, соответствующие одному периоду воспроизводимых сигналов.

4. Вызвать функцию `LTR35_SwitchCyclePage()` для начала выдачи периодически повторяемого сигнала на ЦАП/цифровые линии.
5. При необходимости выставить новый сигнал повторить пункты 1-4 для выдачи следующего набора сигналов.

3.1.2 Вывод сигнала при потоковом режиме

Для вывода сигналов в **потоковом режиме** нужно выполнить следующие шаги

1. Подготовить блок данных для выдачи в нужном формате с помощью `LTR35_PrepareData()` или `LTR35_PrepareDacData()` и передать в модуль с помощью `LTR35_Send()`.
2. Запустить выдачу данных в потоковом режиме с помощью `LTR35_StreamStart()`.
3. По мере необходимости подготавливать и подгружать в буфер LTR35 новые данные по аналогии с пунктом 1.

3.2 Настройка модуля

Настройка модуля должна выполняться хотя бы один раз после установления связи с модулем до загрузки данных на выдачу.

Настройка модуля выполняется аналогично большинству других модулей LTR: вначале значения всех параметров модуля записываются в соответствующие поля **структуры описателя модуля**, затем вызывается функция `LTR35_Configure()`, которая значения этих полей передает модулю.

Следует отметить, что все поля, относящиеся к настройке модуля, объединены в структуру типа `TLTR35_CONFIG` (поле **Cfg описателя модуля**). Только эти поля пользователь должен изменять вручную в **описателе модуля** при штатной работе и именно эти поля влияют на параметры, передаваемые функцией `LTR35_Configure()` в модуль.

При настройке модуля задаются следующие параметры:

- Частота генерации ЦАП и вывода на цифровые линии (раздел **Настройка частоты генерации ЦАП и вывода на цифровые линии**).
- Настройки каналов ЦАП (раздел **Настройка каналов ЦАП**).
- Настройка режима вывода. Задается полем `OutMode`. Режимы вывода описаны в разделе **Режимы вывода**.
- Настройка формата передаваемых данных (раздел **Настройка формата передаваемых данных**).
- Настройка арифметических генераторов (раздел **Настройка арифметических генераторов**).
- Настройка параметров синхронного ввода (раздел **Настройка синхронного ввода**).
- Настройка синхронизации старта нескольких модулей (раздел **Настройка синхронизации нескольких модулей**).

- При необходимости настройка дополнительных параметров, задаваемых флагами конфигурации из перечисления `e_LTR35_CFG_FLAGS` в поле `Flags`.

После выполнения `LTR35_Configure()` рассчитываются некоторые параметры, которые являются производными от настроек из `TLTR35_CONFIG`. Эти параметры записываются в соответствующие поля структуры состояния модуля типа `TLTR35_STATE` (поле `State` описателя модуля).

3.2.1 Настройка частоты генерации ЦАП и вывода на цифровые линии

Частота генерации задается общая на все каналы ЦАП, а также с этой частотой производится вывод значений на цифровые линии, т.е. вывод на все разрешенные каналы ЦАП и цифровые линии происходит одновременно. При этом частота преобразования не зависит от количества разрешенных каналов, однако в [поточковом режиме](#) следует учитывать ограничения скорости интерфейса между модулем и крейтом.

Частота генерации ЦАП определяется двумя параметрами:

- Значение входной частота ЦАП. Для гибкой настройки в модуле установлен синтезатор частоты, который из частоты 30 МГц от единого опорного генератора крейта LTR генерирует частоту (далее именуемую частотой синтезатора — f_{synt}), определяемую его настройками, которая уже является входной частотой для ЦАП. Для настроек синтезатора используется поле `FreqSynt` в [настройках модуля](#), которое представляет собой структуру типа `TLTR35_FREQ_SYNT_CONFIG`. В описании данной структуры указана формула для определения частоты синтезатора по значениям полей данной структуры.
- Скоростью выдачи отсчетов самого ЦАП. Данный параметр задается с помощью поля `DacRate` в [настройках модуля](#) и может принимать одно из значений из перечисления `e_LTR35_RATE`. Скорость выдачи отсчетов определяет количество периодов входной частоты ЦАП, соответствующее выдаче одного отсчета. Каждая скорость соответствует своему допустимому интервалу частот генерации ЦАП. Отношение входной частоты и частоты генерации ЦАП, а также допустимый интервал частот генерации для каждой скорости приведены в описании значений перечисления `e_LTR35_RATE`.

Как правило пользователю не обязательно заполнять эти настройки вручную. Для заполнения этих полей предназначена функция `LTR35_FillOutFreq()`, которая сама подбирает подходящие настройки так, чтобы результирующая частота генерации ЦАП (и вывода на цифровые линии) была наиболее близка к заданной пользователем.

В качестве стандартных частот генерации определены следующие три частоты:

- 192 КГц (скорость `LTR35_DAC_RATE_QUAD`)
- 96 КГц (скорость `LTR35_DAC_RATE_DOUBLE`)
- 48 КГц (скорость `LTR35_DAC_RATE_SINGLE`)

При необходимости пользователь может настроить любую частоту генерации ЦАП в диапазоне от 36 КГц (`LTR35_DAC_FREQ_MIN`) до 192 КГц (`LTR35_DAC_FREQ_MAX`), однако только для трех указанных выше стандартных

частот определяются метрологические характеристики модуля, поэтому без крайней необходимости рекомендуются использовать именно стандартные частоты.

Следует отметить, что для всех трех стандартных частот вывода используются одна и та же частота синтезатора, равная 36,864 МГц (`LTR35_SYNT_FREQ_STD`).

После завершения конфигурации с помощью `LTR35_Configure()` полученное результирующее значение частоты синтезатора возвращается в поле `SyntFreq` структуры **состояния модуля**, а результирующее значение частоты генерации ЦАП и цифрового вывода — в поле `OutFreq`.

3.2.2 Настройка каналов ЦАП

В модуле LTR35 может быть 8, 4 или 0 каналов ЦАП в зависимости от модификации модуля. Количество каналов можно узнать по полю `DacChCnt` структуры с **информацией о модуле** после установления связи с модулем.

Настройки каждого канала объединены в структуре `TLTR35_CHANNEL_CONFIG`. Массив структур, каждый элемент которого соответствует определенному каналу, является полем `Ch` структуры с **настройками модуля**. Для каждого канала может быть настроено:

- Разрешена ли выдача данных по этому каналу. Задается полем `Enabled`.
- Какой из выходов используется для данного канала. Задается полем `Output`. Для LTR35-1 каждому каналу соответствует два выхода (1:1 и 1:10) с разным диапазоном амплитуд воспроизводимого сигнала, в то время как для LTR35-2 используется всегда только один выход на канал. При этом максимальное и минимальное значения для каждого выхода доступны в описании выходов (массив `DacOutDescr` структуры с **информацией о модуле**). При этом сигнал воспроизводится всегда на всех выходах используемого канала, но уровень, соответствующий переданным значениям обеспечивается только на настроенном выходе (на другом уровне сигнала отличается на соотношение диапазонов выходов), а также только на настроенном выходе обеспечиваются метрологические характеристики, так как для каждого выхода используются свои калибровочные коэффициенты.
- Источник сигнала для данного канала ЦАП. Задается полем `Source`. Источником могут быть как отсчеты, загруженные в буфер модуля (`LTR35_CH_SRC_SDRAM`), так и отсчеты, генерируемые внутри модуля арифметическими генераторами. Настройки арифметических генераторов описаны в разделе **Настройка арифметических генераторов**.

3.2.3 Настройка арифметических генераторов

В модуле LTR35 присутствует 4 (`LTR35_ARITH_SRC_CNT`) синусоидальных арифметических генератора. У каждого генератора есть два выхода — синус и косинус. Любой выход любого генератора можно подать на любой канал ЦАП, задав в настройках этого канала в поле `Source` соответствующее значение (при этом один выход арифметического генератора может быть подан на несколько каналов ЦАП).

В случае, если на один или несколько каналов ЦАП нужно вывести синусоидальный сигнал заданной частоты, фазы, амплитуды и смещения, то использование арифметических генераторов позволяет сделать это не формируя сигнал в памяти и не загружая его в буфер модуля. Это в частности позволяет:

- Уменьшить время загрузки сигнала. При этом смену частоты, амплитуды и смещения можно выполнять посылкой минимального количества команд (функции [LTR35_SetArithSrcDelta\(\)](#) и [LTR35_SetArithAmp\(\)](#)), не влияя на остальные каналы при запущенной генерации сигналов.
- Уменьшить поток данных в модуль, что позволяет использовать больше каналов в [поточковом режиме](#).
- Избежать занимаемого места в странице памяти в режиме [циклического автогенератора](#). Смена страниц и смена параметров сигнала от арифметического сигнала выполняются независимо, смена страниц не требует загрузки отсчетов для каналов с данными от арифметического генератора, а для смены частоты и амплитуды сигналов арифметического генератора не требуется смена страниц памяти.

Структура [TLTR35_ARITH_SRC_CONFIG](#) содержит настройки одного арифметического генератора. Массив структур, каждый элемент которого соответствует определенному арифметическому генератору, является полем [ArithSrc](#) структуры с [настройками модуля](#). Для каждого арифметического генератора может быть настроена частота и начальная фаза (поле [Phase](#)) сигнала. Частота задается с помощью поля [Delta](#), которое задает значение приращения фазы для каждого последующего отсчета. Для вычисления фазы в ПЛИС модуля LTR35 используются 32-битные целые положительные коды, вес каждого разряда которого равен $2\pi/2^{32}$. Для задания кода фазы используется структура [TLTR35_ARITH_PHASE](#), поле [CodeH](#) которой задает данный 32-битный код (также структура имеет резервное поле для возможности увеличения разрядности кода в будущем). Для заполнения этой структуры значениями, соответствующими заданной фазе в градусах, можно использовать функцию [LTR35_FillArithPhaseDegree\(\)](#).

При этом частота выходного сигнала будет равна (Delta - фаза, соответствующая заданному коду приращения):

$$f_{sig} = \frac{f_{dac} * Delta}{360}$$

Амплитуда и смещение сигнала задаются в Вольтах непосредственно в настройках канала [TLTR35_CHANNEL_CONFIG](#) полями [ArithAmp](#) и [ArithOffs](#) соответственно. Эти настройки являются свойством канала, а не настройками генератора, т.е. для вывода на разные каналы синусоидальных сигналов с одинаковой частотой и фазой, но разными амплитудами и/или смещением можно использовать один и тот же генератор.

Два выхода каждого генератора (синус и косинус) могут быть использованы для получения с помощью одного генератора одинаковых сигналов (возможно с разными амплитудами), сдвинутых по фазе на 90 градусов ($\pi/2$).

Следует также отметить, что в [поточковом режиме](#) генерация сигнала на каналах, выдающих сигнал с одного из арифметических генераторов, не зависит от состояния буфера подгружаемых потоковых данных, т.е. если происходит опустошение очереди, то генерация синусоидальных сигналов все равно выполняется непрерывно. В режиме [циклического автогенератора](#) 20-битный режим позволяет только увеличить скорость загрузки страницы из ПК за счет сокращения передаваемых слов.

3.2.4 Настройка формата передаваемых данных

Поле [OutDataFmt](#) задает формат отсчетов при передаче от ПК в буфер модуля. Использование 20-битного формата ([LTR35_OUTDATA_FORMAT_20](#)) приводит к потере младших 4-х бит отсчетов ЦАП по сравнению с 24-х битным

([LTR35_OUTDATA_FORMAT_24](#)), но позволяет в 2 раза уменьшить объем данных, передаваемых в модуль, так как в этом формате каждый отсчет занимает одно 32-битное слово LTR, а не два. При этом формат влияет только на формат передаваемых слов, а не на формат отсчетов, хранящихся в самом буфере модуля — в буфере модуля может храниться одинаковое количество отсчетов вне зависимости от настроенного формата передаваемых данных.

Таким образом, использование 20-битного формата может быть полезно в первую очередь в [потокном режиме](#), что позволяет задействовать больше каналов на большей частоте.

3.2.5 Настройка синхронного ввода

Помимо генерации сигналов на ЦАП и цифровых выходах модуль LTR35 также может осуществлять синхронный потокный ввод, запуск и останов которого строго привязан к запуску и останову генерации сигналов. На момент написания данного документа эта возможность доступна только для [потокного режима](#).

Все параметры синхронного ввода задаются с помощью поля [InStream](#) в [настройках модуля](#) и описаны с помощью структуры [TLTR35_IN_STREAM_CONFIG](#).

Существует два различных режима синхронного ввода, определяемых полем [InStreamMode](#):

1. Режим эхо-канала ([LTR35_IN_STREAM_MODE_CH_ECHO](#)). В данном режиме после запуска генерации модуль непосредственно в момент вывода нового отсчета передает в ПК слово с кодом, который записывается в заданный канал ЦАП в этот момент. Основные особенности данного режима:
 - Эхо-коды могут возвращаться только для одного канала. Номер этого канала задается с помощью поля [EchoChannel](#).
 - Частота слов в потоке синхронного ввода всегда равна частоте генерации данных на ЦАП вне зависимости от формата данных или количества разрешенных каналов.
 - Возвращаемые в эхо-данных коды ЦАП отличаются от передаваемых в модуль, так как передаются значения кодов, непосредственно записываемые в ЦАП, к которым уже были применены калибровочные коэффициенты и которые были пропущены через фильтр коррекции АЧХ внутри модуля.
 - В [потокном режиме](#), если происходит опустошение буфера, из-за чего один и тот же отсчет выводится повторно, то в ПК будет выслано по эхо-слову на каждый вывод отсчета, включая повторные. Т.е. частота эхо слов всегда соответствует частоте генерации ЦАП и не зависит от наличия данных в буфере модуля. Таким образом, при опустошении буфера количество высланных слов с эхо-данными может не совпадать с количеством переданных отсчетов по данному каналу в модуль и при использовании данного канала для контроля заполненности очереди следует не допускать опустошения.
 - На генерацию эхо-данных может быть настроен не только канал, отсчеты которого соответствуют переданным из ПК данным, но и канал, данные для которого генерируются одним из [арифметических генераторов](#).
 - Количество переданных слов с эхо-данными и их формат не зависят от настроенного [формата посылаемых данных](#). Модуль всегда высылает только

одно слово LTR с записанным 24-битным кодом ЦАП на один период генерации ЦАП.

2. Режим синхронного ввода состояний цифровых входов модуля ([LTR35_IN_STREAM_MODE_DI](#)). В данном режиме реализуется синхронный цифровой ввод с одного или двух цифровых каналов с точной привязкой по времени к одновременно выполняемому выводу данных на ЦАП и цифровые выходы. Основные особенности данного режима:

- Список цифровых входов, которые будут опрашиваться, задается с помощью поля [DIChEnMask](#). LTR35 имеет 2 цифровых входа: DI1 и DI2. Опрос может производиться как одного выбранного цифрового входа, так и сразу обоих, однако при опросе обоих входов максимальная частота ввода каждого входа в два раза меньше, чем при опросе только одного.
- Частота ввода получается на основе частоты синтезатора, также как и частота вывода (раздел [Настройка частоты генерации ЦАП и вывода на цифровые линии](#)), что позволяет обеспечить точное соответствие моментов ввода и вывода данных.
- Частота ввода задается с помощью поля [DIFreqDivPow](#) и определяется по формуле:

$$f_{din_acq} = \frac{f_{synt}}{4 * 2^{DIFreqDivPow}}$$

Фиксированный делитель равный 4 из приведенной формулы определен константой [LTR35_DIN_SYNT_FREQ_DIV](#). При вводе с одного цифрового входа минимальное значение [DIFreqDivPow](#) равно нулю и при стандартных частотах генерации максимальная частота ввода достигает 9,216 МГц. В случае ввода состояний обоих входов минимальное значение [DIFreqDivPow](#) начинается с 1, что соответствует частоте 4,608 МГц на каждый вход для тех же условий. Для подбора значения поля `fieldref{TLTR35_IN_STREAM_CONFIG,DIFreqDivPow}` так, чтобы частота ввода была наиболее близка к заданной, можно также использовать функцию [LTR35_FillDIAcqFreq\(\)](#).

- Из модуля передаются слова, каждое из которых содержит 24 измеренных значений состояний входов (12 с одного входа и 12 с другого при опросе обоих входов). Таким образом, частота слов в потоке ввода соответствует $f_{din_word} = (f_{din_acq} * N_{din_en})/24$, где N_{din_en} — количество опрашиваемых каналов.
- После завершения конфигурации с помощью [LTR35_Configure\(\)](#) полученное результирующее значение частоты опроса цифровых входов возвращается в поле [InStreamDIAcqFreq](#) структуры [состояния модуля](#), а результирующее значение частоты слов в потоке на ввод — в поле [InStreamWordFreq](#).

Прием и интерпретация данных при включенном синхронном вводе описаны в разделе [Прием данных при синхронном вводе](#).

3.2.6 Настройка синхронизации нескольких модулей

В модуле LTR35 реализована возможность запуска генерации (а также возможного ввода данных) по внешнему сигналу, что в частности позволяет выполнять одновременный синхронный запуск для нескольких модулей по принципу ведущий-ведомые.

На момент написания данного документа эта возможность доступна только для [поточкового режима](#). Следует также учитывать общее ограничение скорости вывода на модули из одного крейта в [поточковом режиме](#).

Все параметры синхронного запуска задаются с помощью поля [Sync](#) в [настройках модуля](#) и описаны с помощью структуры [TLTR35_SYNC_CONFIG](#):

- поле [SyncMode](#) задает режим синхронного запуска, что определяет роль модуля при многомодульном синхронном старте. Один модуль как правило настраивается как ведущий ([LTR35_SYNC_MODE_MASTER](#)), который генерирует импульс старта на выходе DO8, а остальные модули настраиваются как ведомые ([LTR35_SYNC_MODE_SLAVE](#)) — на их вход DI2 подается внешний сигнал запуска генерации от мастера. Вместо мастера также может быть использован внешний сигнал запуска модулей, тогда все модули настраиваются как ведомые.
- поле [SlaveSrc](#) задает условие запуска генерации для ведомого модуля. В случае, если источником сигнала запуска является ведущий модуль LTR35 в режиме [LTR35_SYNC_MODE_MASTER](#), данное поле должно быть равным [LTR35_SYNC_SLAVE_SRC_DI2_RISE](#). Остальные значения могут быть полезны при использовании внешним образом генерируемого сигнала общего запуска.

Последовательность вызовов для запуска модулей при многомодульном синхронном старте описана в разделе [Синхронизация запуска нескольких модулей](#).

3.3 Режимы вывода

Режим вывода модуля устанавливается с помощью поля [OutMode](#) структуры с [настройками модуля](#) и влияет на то, как используется внутренний буфер модуля с отсчетами для генерации сигналов. Т.е. режим влияет на каналы, данные для которых загружаются в буфер модуля — то есть, каналы ЦАП с источником [Source](#) = [LTR35_CH_SRC_SDRAM](#) и на вывод на цифровые линии. При этом в любом режиме часть каналов ЦАП может быть также настроена на работу от одного из [арифметических генераторов](#).

3.3.1 Поточковый режим вывода

В потоковом режиме ([LTR35_OUT_MODE_STREAM](#)) буфер модуля LTR35 представляет собой очередь на 8 миллионов отсчетов. Подгружаемые в модуль отсчеты сохраняются в конец очереди. После запуска генерации данных с помощью [LTR35_StreamStart\(\)](#) модуль начинает вычитывать данные из начала очереди, если она не пуста, с установленной частотой генерации и записывать их в каналы ЦАП и/или выводить на цифровые выходы.

В случае опустошения очереди будет повторено предыдущее выведенное значение и установлен флаг опустошения очереди в следующем высланном периодическом статусе. Таким образом, для выдачи синхронного сигнала необходимо выполнять постоянную подкачку данных в модуль так, чтобы в буфере модуля всегда были отсчеты и он не опустошался.

Плюсом данного режима является то, что можно выводить непериодические сигналы произвольной формы и длительности (например из файла или формировать сигналы “на лету”). Минусом же является необходимость поддерживать постоянный поток

слов на вывод, что приводит к ограничению количества одновременно используемых каналов на высоких частотах вывода.

Следует иметь в виду, что максимальная скорость потока вывода не должна превышать значения максимальной скорости интерфейса между крейтом и модулем, равной 500 КСлов/с. Для уменьшения размера потока в данном режиме вывода может быть использован 20-битный формат передаваемых отсчетов (раздел [Настройка формата передаваемых данных](#)), что позволяет увеличить допустимое количество каналов в два раза. Для 20-битного режима для частоты генерации 192 КГц можно одновременно использовать два канала для потокового вывода, для частоты 96 КГц – 5 каналов, для 48 КГц – все каналы, включая вывод на цифровые линии.

Также важно отметить, что при использовании потокового вывода одновременно для нескольких модулей, общая скорость потокового вывода в текущей реализации ПО и крейта также ограничена 500 КСлов/с на крейт, из-за того, что на вывод в крейте используется общая очередь, а слова на вывод в общем случае могут приходиться блоками для одного модуля, что приводит к последовательной передаче блоков.

Запуск синхронного вывода в потоковом режиме осуществляется с помощью функции `LTR35_StreamStart()`. Чтобы обеспечить непрерывный вывод сигнала с самого начала запуска необходимо до вызова `LTR35_StreamStart()` выполнить предзагрузку отсчетов в буфер модуля с помощью `LTR35_Send()`. После запуска необходимо при этом периодически подкачивать потоковые данные со скоростью не меньшей, чем скорость вывода.

При попытке передать слово в модуль, буфер которого полностью заполнен, записываемое слово будет отброшено и будет установлен соответствующий флаг в следующем высланном периодическом статусе. Чтобы гарантировать, что буфер модуля не будет переполнен, независимо от количества передаваемых данных, в службе `ltrd` реализована логика контроля заполненности очереди модуля. Для реализации этой логики модуль высылает специальные статусные слова, которые предназначены для `ltrd` и не видны на уровне `ltr35api` (`ltrd` обрабатывает эти слова сам и не передает клиентам):

- При настройке модуля через `LTR35_Configure()` модуль высылает статус с настройками, необходимыми для отслеживания заполненности буфера. Также этот специальный статус сообщает `ltrd`, что буфер был очищен.
- При чтении каждые `StreamStatusPeriod` (значение 0 означает, что используется значение по-умолчанию) отсчетов из буфера модуль высылает периодический статус, в котором также содержатся признаки, были ли за это время события опустошения буфера или были ли отброшены слова из-за заполненности буфера.

По приему этих слов `ltrd` определяет количество переданных, но не выведенных слов и обновляет статистику по заполненности очереди, возникновению событий опустошения или переполнения очереди.

При вызове `LTR35_Send()` слова передаются через буфер сокета канала обмена программы с `ltrd` в буфер `ltrd`. Из этого буфера, если в очереди модуля есть место, слова с максимальной скоростью интерфейса будут передаваться через буфер крейта в буфер модуля. При этом скорость будет ограничиваться сперва скоростью интерфейса ПК с крейтом, затем по заполнению буфера крейта — скоростью интерфейса между крейтом и модулями. При заполнении буфера `LTR35`, `ltrd` будет ограничивать скорость передачи слов и поток передачи будет соответствовать скорости вывода отсчетов на ЦАП.

Таким образом, если пытаться передать данные в модуль с наибольшей скоростью, постоянно вызывая `LTR35_Send()`, то сперва заполнится буфер модуля, затем будет

заполнен буфер `ltrd`, затем буфер сокета между `ltrd` и клиентом (программой, использующей `ltr35api`). При этом `LTR35_Send()` уже не будет возвращаться немедленно, а будет ожидать в течении заданного таймаута, пока освободится место в буфере сокета (при этом если раньше истечет таймаут, то не все данные могут быть переданы и возвращенное `LTR35_Send()` значение будет отличаться от запрошенного на передачу размера, что потребует повторить посылку не переданных слов). Однако непрерывная выдача на ЦАП не прекратится и переполнения буфера и потерь отсчетов не будет.

Для поддержки данного контроля переполнения необходимо, чтобы версия `ltrd` была не ниже 2.1.3.0.

Если данный контроль не требуется, то он может быть явно запрещен с помощью установки флага `LTR35_CFG_FLAG_DISABLE_PERIODIC_STATUS` при конфигурации модуля, однако в этом случае контроль за заполненностью очереди переносится на пользователя, для чего может быть использован прием данных синхронного ввода (раздел [Прием данных при синхронном вводе](#)). Также ручной контроль заполненности буфера полезен для явного ограничения количества не выведенных слов на уровне, меньшем, чем полный размер буфера, что позволяет уменьшить задержку между изменением данных и сигнала на выходе при изменении сигнала “на лету”, а также уменьшить время останова генерации в потоковом режиме.

Также потоковый режим позволяет эмулировать асинхронный вывод. В этом случае буфер модуля находится почти всегда в пустом состоянии, в результате чего на выходе сохраняются последние выведенные значения, и только при необходимости изменить значения передается одна пачка отсчетов, содержащая по одному отсчету на каждый канал с новым уровнем для вывода.

3.3.2 Режим циклического автогенератора

Данный режим может использоваться для вывода периодического сигнала произвольной формы. В этом режиме набор сигналов (по одному сигналу на каждый канал) сперва загружается в страницу буфер модуля, после чего по вызову функции `LTR35_SwitchCyclePage()` модуль начинает выдавать загруженные данные данной страницы. При этом чтение станицы идет циклически (после вывода всей страницы начинается повторный вывод тех же сигналов с начала страницы) и не требует дальнейшей подкачки данных.

При выводе на несколько каналов сигналов с разными периодами нужно найти общий период всех сигналов как наименьшее общее кратное периодов каждого сигнала. При этом суммарное полученное число отсчетов не должно превышать `LTR35_MAX_POINTS_PER_PAGE`.

В данном режиме память `LTR35` разделена на два циклических буфера (две страницы). Это позволяет записывать новый набор сигналов, не останавливая вывод предыдущего, после чего выполнить смену сигналов на границе старой страницы без остановки генерации.

Загрузка нового набора сигналов выполняется аналогично записи первого набора. Т.е. после того как был записан первый набор сигналов и начата генерация сигналов с помощью `LTR35_SwitchCyclePage()`, то последующие вызовы `LTR35_Send()` будут передавать данные, которые будут записаны уже во вторую страницу. После того как второй набор сигналов будет записан полностью, нужно снова вызвать `LTR35_SwitchCyclePage()` для смены страниц. При этом смена произойдет в момент, когда вывод дойдет до конца первой страницы (т.е. для каждого канала первый отсчет нового сигнала будет выведен после последнего отсчета предыдущего). При этом ЦАП

начнет выдавать сигнал с начала второй страницы, а первая снова будет доступна на запись. `LTR35_SwitchCyclePage()` возвращает управление, когда смена страницы уже будет завершена.

3.4 Прием данных при синхронном вводе

В случае, если в настройках модуля был разрешен синхронный ввод (задан режим, отличный от `LTR35_IN_STREAM_MODE_OFF`), одновременно с запуском генерации данных модуль начинает передавать поток данных в ПК. Скорость и содержимое этих данных зависят от настроек, описанных в разделе [Настройка синхронного ввода](#). Скорость потока данных (частота передаваемых слов) устанавливается в поле `InStreamWordFreq` структуры [состояния модуля](#).

Пользователю необходимо периодически вычитывать эти данные с помощью функции `LTR35_RecvInStreamData()`. При этом необходимо выполнять чтение данные со скоростью не ниже их передачи из модуля, чтобы не допустить переполнения очереди приема в службе `ltrd`.

Формат слов зависит от режима:

- В режиме эхо-канала для каждого слова младшие 24 бита точно равны записываемому в заданный канал ЦАП коду при соответствующем цикле вывода, а старшие 8 бит содержат расширение знака (дублируют старший бит кода ЦАП), что позволяет интерпретировать код как 32-битное целое знаковое число (INT).
- В режиме ввода состояния цифровых линий старшие 8 бит 32-битного слова всегда равны нулю, а младшие 24 бита содержат состояния входов при соответствующем цикле опроса в зависимости от количества опрашиваемых каналов:
 - при опросе одного канала все 24 бита соответствуют последовательным 24-м измерениям состояния входа DI1 или DI2, при этом старший бит соответствует первому измерению, а младший — последнему.
 - при опросе обоих каналов старшие 12 бит соответствуют опросу состояния входа DI2, а младшие 12 бит — DI1. Аналогично в каждой последовательности из 12 бит старший бит соответствует первому измерению, а младший — последнему.

Хотя функции библиотеки `ltr35api` сами по себе не являются потокобезопасными и пользователь должен гарантировать, что что вызовы функций для работы с одним и тем же модулем выполняется последовательно (работа с разными модулями LTR35, как и для других типов модулей, может выполняться из разных потоков). Однако для возможности реализации одновременного ввода и вывода из разных потоков сделано исключение: после запуска генерации возможен одновременный вызов из одного потока функций `LTR35_PrepareData()` / `LTR35_PrepareDacData()` и `LTR35_Send()`, а из другого — вызов `LTR35_RecvInStreamData()`. Таким образом, одновременный ввод-вывод из разных потоков может быть организован следующим образом:

1. Начальная подкачка данных с помощью `LTR35_PrepareData()` / `LTR35_PrepareDacData()` и `LTR35_Send()`.
2. Запуск генерации данных с помощью `LTR35_StreamStart()`.
3. Запуск второго потока.

4. Подкачка новых данных потока с помощью `LTR35_PrepareData()` / `LTR35_PrepareDacData()` из одного и прием данных с помощью `LTR35_RecvInStreamData()` из другого.
5. Повтор пункта 4 до завершения вывода.
6. Останов второго потока.
7. Останов генерации с помощью `LTR35_Stop()` / `LTR35_StopWithTout()`.

Также можно организовать ввод и вывод из одного потока, используя тот факт, что частоты ввода и вывода основаны на одной частоте с выхода синтезатора и их соотношение явно известно для текущей конфигурации (соотношение значений в полях `InStreamWordFreq` и `OutFreq` соответственно). В этом случае запись и чтение должны выполняться блоками, точно соответствующими выбранному интервалу блока (T_b). Одновременный ввод и вывод может быть организован следующим образом:

1. Начальная подкачка данных с помощью `LTR35_PrepareData()` / `LTR35_PrepareDacData()` и `LTR35_Send()`.
2. Запуск генерации данных с помощью `LTR35_StreamStart()`.
3. Подкачка $OutFreq * T_b$ новых точек на каждый разрешенный канал вывода с помощью `LTR35_PrepareData()` / `LTR35_PrepareDacData()`.
4. Прием $InStreamWordFreq * T_b$ слов данных с помощью функции `LTR35_RecvInStreamData()`.
5. Повтор пунктов 3 и 4 до завершения вывода.
6. Останов генерации с помощью `LTR35_Stop()` / `LTR35_StopWithTout()`.

3.5 Синхронизация запуска нескольких модулей

Для организации синхронного запуска одновременно нескольких модулей, модуль LTR35 поддерживает синхронный запуск по принципу ведущий-ведомые.

В этом случае один модуль LTR35 должен быть выбран как ведущий. Его выход DO8 используется для генерации сигнала запуска ведомых модулей. При выводе на цифровые пользователь не управляет состоянием DO8 ведущего модуля и значение этого бита при выводе игнорируется. Выход DO8 ведущего модуля должен быть соединен с входом DI2 каждого ведомого модуля.

Последовательность, в которой модули будут настроены не имеет значения, однако важное значение имеет последовательность запуска — все ведомые модули должны быть запущены раньше ведущего.

Ведомые модули по команде запуска генерации переходят в режим ожидания заданного с помощью поля `SlaveSrc` условия. Ответ на команду запуска возвращается ведомым только после обнаружения этого условия перед непосредственным началом ввода-вывода и может служить признаком успешного запуска по внешнему условию.

Таким образом, от момента передачи команды запуска генерации до получения ответа может пройти значительное время (или ответ может вообще не прийти, если по каким-то причинам сигнал запуска не был подан на модуль), поэтому для этого случая

для отправки команды запуска и ожидания ответа введены две разные функции: `LTR35_StreamStartRequest()` и `LTR35_StreamStartWaitDone()` соответственно. При этом последняя функция может вызываться несколько раз в случае, если не удалось дождаться ответа за предыдущий вызов. Если по какой-то причине нужно отменить ожидание, можно вызвать `LTR35_Stop()`.

Последовательность вызова функций для синхронного запуска выглядит следующим образом:

1. Конфигурация одного модуля ведущим, остальных — ведомыми
2. Предзагрузка данных и запуск генерации без подтверждения для всех ведомых модулей с помощью `LTR35_StreamStartRequest()`.
3. Предзагрузка данных и запуск ведущего модуля с помощью `LTR35_StreamStart()`.
4. Ожидание завершения запуска генерации каждого ведомого модуля с помощью `LTR35_StreamStartWaitDone()`. В случае возвращения функцией кода ошибки `LTR_ERROR_OP_DONE_WAIT_TOUT` возможно продолжение ожидания повторным вызовом `LTR35_StreamStartWaitDone()` или прекращение ожидания с помощью `LTR35_Stop()`.

Пункт 3 может выполняться одновременно с выполнением пункта 4 в случае, если для каждого модуля используется свой поток.

3.6 Особенности калибровки данных

Следует отметить, что калибровка данных выполняется аппаратно внутри модуля, а не программно. В связи с этим в `LTR35_PrepareData()` и `LTR35_PrepareDacData()` нет никаких указаний о выполнении калибровки. Самой библиотекой при открытии связи с модулем через `LTR35_Open()` выполняется чтение калибровочных коэффициентов из Flash-памяти модуля, сохранение их в полях массива `CbrCoef` в структуре с информацией о модуле. При конфигурации модуля с помощью `LTR35_Configure()` выполняется запись нужных коэффициентов (в зависимости от настроенного выхода для каждого канала) в ПЛИС.

ПЛИС налету выполняет калибровку по формуле $Y = (X + Offset) * Scale$, где X — 24-битный отсчет, передаваемый в модуль (или генерируемый одним из арифметических генераторов), Y — калиброванный отсчет, который непосредственно записывается в ЦАП, $Offset$ — смещение шкалы (24-битный код), а $Scale$ — коэффициент шкалы для настроенного выхода соответствующего канала.

Если пользователю необходимо установить свои калибровочные коэффициенты, то он должен изменить значение полей `CbrCoef` в структуре с информацией о модуле перед вызовом `LTR35_Configure()`.

Также в ПЛИС модуля реализована коррекция АЧХ модуля с помощью КИХ-фильтра второго порядка в соответствии с методом, описанным в статье [Метод тонкой коррекции наклона АЧХ с помощью простого цифрового фильтра](#).

Коэффициенты коррекции также загружаются из полей `CbrAfcCoef` в структуре с информацией о модуле во время вызова `LTR35_Configure()`. При необходимости данную коррекцию можно отключить на этапе конфигурации модуля, установив флаг `LTR35_CFG_FLAG_DISABLE_AFC_COR` в поле `Flags` в настройках модуля.

Глава 4

Константы, типы данных и функции библиотеки

4.1 Константы и перечисления

4.1.1 Константы и макроопределения

Константа	Значение	Описание
LTR35_NAME_SIZE	8	Размер строки с именем модуля в структуре TLTR35_MODULE_INFO
LTR35_SERIAL_SIZE	16	Размер строки с серийным номером модуля в структуре TLTR35_MODULE_INFO
LTR35_DAC_CHANNEL_CNT	8	Максимальное количество каналов ЦАП
LTR35_DAC_CH_OUTPUT_CNT	2	Максимальное количество выходов для каждого канала ЦАП
LTR35_DOUT_MAX_CNT	16	Максимальное количество цифровых выходов
LTR35_DIN_CNT	2	Количество цифровых входов модуля
LTR35_MAX_POINTS_PER_PAGE	(4*1024*1024)	Максимальное количество отсчетов в одной странице буфера на вывод в режиме циклического автогенератора
LTR35_ARITH_SRC_CNT	4	Количество арифметических синусоидальных генераторов в модуле
LTR35_DAC_FREQ_MAX	192000	Максимальное значение частоты генерации ЦАП в Гц, которое можно установить

LTR35_DAC_FREQ_MIN	36000	Минимальное значение частоты генерации ЦАП в Гц, которое можно установить
LTR35_DAC_SINGLE_RATE_FREQ_MAX	54000	Максимальное значение частоты генерации ЦАП в Гц для скорости LTR35_DAC_RATE_SINGLE
LTR35_DAC_DOUBLE_RATE_FREQ_MAX	108000	Максимальное значение частоты генерации ЦАП в Гц для скорости LTR35_DAC_RATE_DOUBLE
LTR35_DAC_QUAD_RATE_FREQ_STD	192000	Стандартная частота генерации ЦАП в Гц для скорости LTR35_DAC_RATE_QUAD
LTR35_DAC_DOUBLE_RATE_FREQ_STD	96000	Стандартная частота генерации ЦАП в Гц для скорости LTR35_DAC_RATE_DOUBLE
LTR35_DAC_SINGLE_RATE_FREQ_STD	48000	Стандартная частота генерации ЦАП в Гц для скорости LTR35_DAC_RATE_SINGLE
LTR35_SYNT_FREQ_STD	36864000	Частота синтезатора в Гц, используемая для стандартных частот генерации ЦАП
LTR35_DAC_CODE_MAX	0x7FFFFFFF	Максимальный код ЦАП
LTR35_DAC_SCALE_CODE_MAX	0x600000	Код ЦАП, соответствующий максимальному значению диапазона в Вольтах
LTR35_STREAM_STATUS_PERIOD_DEFAULT	1024	Устанавливаемое по умолчанию количество прочитанных отсчетов для выдачи периодического статуса в потоковом режиме
LTR35_STREAM_STATUS_PERIOD_MAX	1024	Максимальное значение количества отсчетов для выдачи периодического статуса в потоковом режиме
LTR35_STREAM_STATUS_PERIOD_MIN	8	Минимальное значение количества отсчетов для выдачи периодического статуса в потоковом режиме
LTR35_DIN_SYNT_FREQ_DIV	4	Делитель частоты синтезатора для получения базовой частоты синхронного ввода с цифровых линий

LTR35_DI_STREAM_FREQ_DIV_POW_MAX	5	Максимальное значение для степени делителя частоты синхронного ввода с цифровых линий
LTR35_FLASH_USERDATA_ADDR	0x100000	Адрес, с которого начинается пользовательская область Flash-памяти
LTR35_FLASH_USERDATA_SIZE	0x700000	Размер пользовательской области flash-памяти
LTR35_FLASH_ERASE_BLOCK_SIZE	1024	Минимальный размер блока для стирания во Flash-памяти модуля

4.1.2 Специфичные для LTR35 коды ошибок

Тип: e_LTR35_ERROR_CODES		
Описание: Коды ошибок, которые определены и используются только в ltr35api. Остальные коды ошибок, которые используются разными модулями, определены в ltrapi.h.		
Константа	Значение	Описание
LTR35_ERR_INVALID_SYNT_FREQ	-10200	Задана неподдерживаемая частота синтезатора
LTR35_ERR_PLL_NOT_LOCKED	-10201	Ошибка захвата PLL
LTR35_ERR_INVALID_CH_SOURCE	-10202	Задано неверное значение источника данных для канала
LTR35_ERR_INVALID_CH_RANGE	-10203	Задано неверное значение диапазона канала
LTR35_ERR_INVALID_DATA_FORMAT	-10204	Задано неверное значение формата данных
LTR35_ERR_INVALID_OUT_MODE	-10205	Задано неверное значение режима вывода
LTR35_ERR_INVALID_DAC_RATE	-10206	Задано неверное значение скорости выдачи на ЦАП
LTR35_ERR_INVALID_SYNT_CNTRS	-10207	Задано недопустимое значение счетчиков синтезатора
LTR35_ERR_UNSUPPORTED_CONFIG	-10208	Заданная конфигурация ЦАП не поддерживается
LTR35_ERR_INVALID_STREAM_STATUS_PERIOD	-10209	Задано неверное количество отсчетов для выдачи статуса в потоковом режиме
LTR35_ERR_DAC_CH_NOT_PRESENT	-10210	Выбранный канал ЦАП отсутствует в данном модуле
LTR35_ERR_DAC_NO_SDRAM_CH_ENABLED	-10211	Не разрешен ни один канал ЦАП на вывод из SDRAM
LTR35_ERR_DAC_DATA_NOT_ALIGNED	-10212	Данные ЦАП не выравнены на кол-во разрешенных каналов

LTR35_ERR_NO_DATA_LOADED	-10213	Не было загружено ни одного отсчета для циклического вывода
LTR35_ERR_LTRD_UNSUP_STREAM_MODE	-10214	Данная версия службы ltrd не поддерживает контроль переполнения буфера в потоковом режиме
LTR35_ERR_MODE_UNSUP_FUNC	-10215	Данная функция не поддерживается в установленном режиме
LTR35_ERR_INVALID_ARITH_GEN_NUM	-10216	Задано неверное значение номера арифметического генератора
LTR35_ERR_INVALID_IN_STREAM_MODE	-10217	Задан неверный режим потокового ввода
LTR35_ERR_INVALID_IN_DI_FREQ_DIV_POW	-10218	Задано неверное значение степени делителя частоты синхронного ввода с цифровых линий
LTR35_ERR_IN_DI_NO_CHANNELS	-10219	Не разрешен ни один вход для синхронного ввода с цифровых линий
LTR35_ERR_INVALID_IN_DI_CH_LIST	-10220	Задан неверный список каналов для синхронного ввода с цифровых линий
LTR35_ERR_INVALID_SYNC_MODE	-10221	Задано неверное значение режима синхронизации
LTR35_ERR_INVALID_SYNC_SLAVE_SRC	-10222	Задано неверное значение источника синхронизации для подчиненного модуля
LTR35_ERR_SYNC_MODE_NOT_SET	-10223	Не удалось установить заданный режим синхронизации
LTR35_ERR_NO_START_REQUESTED	-10224	Не был выполнен запрос старта, завершение которого бы следовало ожидать
LTR35_ERR_DAC_REG_VALUE_CHECK	-10225	Ошибка контроля записи значений в регистры ЦАП

4.1.3 Модификации модуля LTR35

Тип: e_LTR35_MODIFICATION		
Описание: Модификации модуля LTR35		
Константа	Значение	Описание
LTR35_MOD_UNKNOWN	0	Неизвестная (не поддерживаемая библиотекой) модификация
LTR35_MOD_1	1	LTR35-1
LTR35_MOD_2	2	LTR35-2
LTR35_MOD_3	3	LTR35-3

4.1.4 Флаги для управления цифровыми выходами

Тип: e_LTR35_DOUTS_WORD_FLAGS		
Описание: Флаги управления цифровыми выходами. Могут быть объединены через логическое “ИЛИ” со значениями цифровых выходов при подготовке данных на вывод с помощью LTR35_PrepareData()		
Константа	Значение	Описание
LTR35_DIGOUT_WORD_DIS_H	0x00020000	Запрещение (перевод в третье состояние) старшей половины цифровых выходов. Имеет значение только для LTR35-3.
LTR35_DIGOUT_WORD_DIS_L	0x00010000	Запрещение младшей половины цифровых выходов

4.1.5 Флаги для подготовки данных

Тип: e_LTR35_PREP_FLAGS		
Описание: Флаги, управляющие работой функций LTR35_PrepareData() и LTR35_PrepareDacData()		
Константа	Значение	Описание
LTR35_PREP_FLAGS_VOLT	0x01	Флаг указывает, что данные на входе заданы в Вольтах и их нужно перевести в коды

4.1.6 Формат отсчетов для передачи данных вывода модулю

Тип: e_LTR35_OUTDATA_FORMAT		
Описание: Формат отсчетов для передачи данных вывода модулю		
Константа	Значение	Описание
LTR35_OUTDATA_FORMAT_24	0	24-битный формат. Один отсчет занимает два 32-битных слова LTR
LTR35_OUTDATA_FORMAT_20	1	20-битный формат. Один отсчет занимает одно 32-битное слово LTR

4.1.7 Режим вывода модуля

Тип: e_LTR35_OUT_MODE		
Описание: Режим вывода модуля		
Константа	Значение	Описание
LTR35_OUT_MODE_CYCLE	0	Режим циклического автогенератора. Данные загружаются в буфер перед запуском выдачи, после чего выводятся по кругу без подкачки новых данных. Описан в разделе Режим циклического автогенератора .
LTR35_OUT_MODE_STREAM	1	Потоковый режим. Данные постоянно должны загружаться в буфер модуля. Данные из буфера последовательно вычитываются при их наличии и выводятся модулем. Описан в разделе Потоковый режим вывода .

4.1.8 Используемый выход для канала ЦАП

Тип: e_LTR35_DAC_OUTPUT		
Описание: Используемый выход для канала ЦАП		
Константа	Значение	Описание
LTR35_DAC_OUT_FULL_RANGE	0	Выход 1:1 с диапазоном от -10 до +10 В для LTR35-1 и от 0 до +20 В для LTR35-2.
LTR35_DAC_OUT_DIV_RANGE	1	Выход 1:10 (от -1 до 1 В) для LTR35-1. Для LTR35-2 данный выход не используется.

4.1.9 Источники сигнала для каналов ЦАП

Тип: e_LTR35_CH_SRC		
Описание: Источники сигнала для каналов ЦАП		
Константа	Значение	Описание
LTR35_CH_SRC_SDRAM	0	Сигнал берется из буфера в SDRAM модуля. При этом буфер работает циклически или в виде очереди в зависимости от режима вывода .
LTR35_CH_SRC_SIN1	1	Синус от первого арифметического синусоидального генератора.
LTR35_CH_SRC_COS1	2	Косинус от первого арифметического синусоидального генератора.
LTR35_CH_SRC_SIN2	3	Синус от второго арифметического синусоидального генератора.
LTR35_CH_SRC_COS2	4	Косинус от второго арифметического синусоидального генератора.
LTR35_CH_SRC_SIN3	5	Синус от третьего арифметического синусоидального генератора.
LTR35_CH_SRC_COS3	6	Косинус от третьего арифметического синусоидального генератора.
LTR35_CH_SRC_SIN4	7	Синус от четвертого арифметического синусоидального генератора.
LTR35_CH_SRC_COS4	8	Косинус от четвертого арифметического синусоидального генератора.

4.1.10 Скорость выдачи отсчетов ЦАП

Тип: e_LTR35_RATE		
Описание: Скорость выдачи отсчетов ЦАП определяет количество тактов частоты синтезатора (f_{synt}), которое требуется для выдачи одного отсчета ЦАП, т.е. отношение частоты генерации ЦАП к частоте синтезатора.		
Константа	Значение	Описание
LTR35_DAC_RATE_SINGLE	0	Одиночная скорость. Частота генерации равна $f_{synt}/768$. Используется для частот генерации от 36 КГц (LTR35_DAC_FREQ_MIN) до 54 КГц (LTR35_DAC_SINGLE_RATE_FREQ_MAX) включительно.
LTR35_DAC_RATE_DOUBLE	1	Двойная скорость. Частота генерации равна $f_{synt}/384$. Используется для частот вывода выше 54 КГц и до 108 КГц (LTR35_DAC_DOUBLE_RATE_FREQ_MAX) включительно.
LTR35_DAC_RATE_QUAD	2	Четверная скорость. Частота генерации равна $f_{synt}/192$. Используется для частот генерации выше 108 КГц и до 192 КГц (LTR35_DAC_FREQ_MAX) включительно.

4.1.11 Режим синхронного ввода

Тип: e_LTR35_IN_STREAM_MODE		
Описание: Данный режим определяет, будет ли одновременно с генерацией вывода запущен и синхронный ввод данных, а также какие именно данные будет передаваться в потоке на ввод.		
Константа	Значение	Описание
LTR35_IN_STREAM_MODE_OFF	0	Синхронный ввод не используется.
LTR35_IN_STREAM_MODE_CH_ECHO	1	Режим эхо-канала, в котором от модуля передается поток отсчетов, соответствующих выводимым на заданный с помощью поля EchoChannel канал отсчетам ЦАП. Частота ввода соответствует частоте генерации ЦАП.
LTR35_IN_STREAM_MODE_DI	2	Режим синхронного ввода состояний цифровых входов. Опрашиваемые входы определяются значением поля DIChEnMask , а частота ввода — значением поля DIFreqDivPow .

4.1.12 Флаги задания цифровых линий для синхронного ввода

Тип: e_LTR35_IN_STREAM_DI_CHS		
Описание: С помощью данных флагов определяется, какие входы будут опрашиваться при синхронном вводе с цифровых линий.		
Константа	Значение	Описание
LTR35_IN_STREAM_DI1	0x00001	Ввод значений с линии DI1
LTR35_IN_STREAM_DI2	0x00002	Ввод значений с линии DI2

4.1.13 Режим синхронизации старта генерации модулей

Тип: e_LTR35_SYNC_MODE		
Описание: Данный режим определяет роль модуля при многомодульной синхронизации старта.		
Константа	Значение	Описание
LTR35_SYNC_MODE_INTERNAL	0	Внутренняя синхронизация. Модуль используются отдельно без синхронизации запуска с другими модулями. Запуск выполняется по команде от ПК, функции цифровых входов и выходов не переопределяются.
LTR35_SYNC_MODE_MASTER	1	Модуль является ведущим при синхронизации старта нескольких модулей. Запуск выполняется по команде с ПК. При этом выход DO8 на время вывода используется для выдачи сигнала синхронизации для ведомых модулей.
LTR35_SYNC_MODE_SLAVE	2	Запуск генерации определяется по условию, заданному с помощью поля SlaveSrc . Данный режим используется для ведомых модулей при синхронизации старта нескольких модулей, а также для синхронизации запуска генерации от внешнего источника.

4.1.14 Условие запуска генерации для подчиненного модуля

Тип: e_LTR35_SYNC_SLAVE_SRC		
Описание: Варианты условий запуска генерации для модуля в режиме синхронизации LTR35_SYNC_MODE_SLAVE .		
Константа	Значение	Описание
LTR35_SYNC_SLAVE_SRC_DI2_RISE	0	Запуск по фронту сигнала на входе DI2. Этот режим должен использоваться также в случае, если источником сигнала является другой модуль LTR35 в режиме мастера.
LTR35_SYNC_SLAVE_SRC_DI2_FALL	1	Запуск по спаду сигнала на входе DI2.

4.1.15 Флаги конфигурации модуля

Тип: e_LTR35_CFG_FLAGS		
Описание: Флаги, определяющие дополнительные параметры конфигурации модуля.		
Константа	Значение	Описание
LTR35_CFG_FLAG_DISABLE_AFC_COR	0x00000001	Если данный флаг установлен, то при генерации сигнала не используется фильтр для коррекции АЧХ модуля.
LTR35_CFG_FLAG_DISABLE_PERIODIC_STATUS	0x00000100	Запрет передачи статусных слов для контроля переполнения буфера службой ltrd.

4.1.16 Флаги для записи во flash-память модуля

Тип: e_LTR35_FLASH_WRITE_FLAGS		
Описание: Флаги для записи во flash-память модуля		
Константа	Значение	Описание
LTR35_FLASH_WRITE_ALREADY_ERASED	0x00001	Признак, что записываемая область памяти уже стерта и не требуется дополнительно стирать обновляемые сектора.

4.1.17 Флаги состояния модуля

Тип: e_LTR35_STATUS		
Описание: Набор флагов, определяющих состояние модуля. Возвращаются функцией LTR35_GetStatus()		
Константа	Значение	Описание
LTR35_STATUS_FLAG_PLL_LOCK	0x0001	Признак захвата PLL в момент передачи статуса. Если равен нулю, то модуль неработоспособен.
LTR35_STATUS_FLAG_PLL_LOCK_HOLD	0x0002	Признак, что захват PLL не пропал с момента предыдущей передачи статуса. Должен быть установлен во всех статусах, кроме первого после сброса модуля.
LTR35_STATUS_FLAG_SYNC_MODE	0x0030	2 бита, определяющие текущий настроенный режим синхронизации модуля (коды из перечисления e_LTR35_SYNC_MODE). Данное значение обновляется после вызова LTR35_Configure() в соответствии со значением поля SyncMode .
LTR35_STATUS_FLAG_DI1_STATE	0x0040	Состояние входа DI1
LTR35_STATUS_FLAG_DI2_STATE	0x0080	Состояние входа DI2

4.2 Типы данных

4.2.1 Калибровочные коэффициенты

Тип: TLTR35_CBR_COEF		
Описание: Структура, хранящая калибровочные коэффициенты для одного выхода одного канала ЦАП.		
Поле	Тип	Описание поля
Offset	float	Код смещения
Scale	float	Коэффициент шкалы

4.2.2 Описание выхода ЦАП

Тип: TLTR35_DAC_OUT_DESCR		
Описание: Структура описывает характеристики определенного выхода для данной модификации модуля LTR35.		
Поле	Тип	Описание поля
AmpMax	double	Максимальное пиковое значение амплитуды сигнала для данного выхода.
AmpMin	double	Минимальное пиковое значение амплитуды сигнала для данного выхода.

CodeMax	INT	Код ЦАП, соответствующий максимальной амплитуде.
CodeMin	INT	Код ЦАП, соответствующий минимальной амплитуде.
Reserved	DWORD [3]	Резервные поля.

4.2.3 Коэффициенты для калибровки АЧХ

Тип: TLTR35_AFC_COEF		
Описание: Структура описывает коэффициенты и параметры, используемые для коррекции АЧХ ЦАП		
Поле	Тип	Описание поля
Valid	BOOLEAN	Признак действительности значения остальных полей.
Reserved	DWORD [3]	Резервные поля.
Fd	double	Частота генерации ЦАП, на которой измерено отношение амплитуд.
SigFreq	double	Частота сигнала, для которой измерено отношение амплитуд.
K	double [LTR35_DAC_CHANNEL_CNT]	Набор отношений реально выставленной амплитуды синусоидального сигнала частоты SigFreq к задаваемой амплитуде при частоте генерации Fd для каждого канала ЦАП.

4.2.4 Информация о модуле

Тип: TLTR35_MODULE_INFO		
Описание: Структура, содержащая информацию о версиях прошивок микросхем модуля, информацию модификации модуля и информацию, считанную из Flash-памяти модуля (серийный номер, калибровочные коэффициенты). Все поля доступны после вызова LTR35_Open() .		
Поле	Тип	Описание поля
Name	CHAR [LTR35_NAME_SIZE]	Название модуля (оканчивающаяся нулем ASCII-строка)
Serial	CHAR [LTR35_SERIAL_SIZE]	Серийный номер модуля (оканчивающаяся нулем ASCII-строка)
VerFPGA	WORD	Версия прошивки ПЛИС модуля (действительна только при ее наличии)
VerPLD	BYTE	Версия прошивки PLD
Modification	BYTE	Модификация модуля. Одно из значений из e_LTR35_MODIFICATION .
DacChCnt	BYTE	Количество установленных каналов ЦАП
DoutLineCnt	BYTE	Количество линий цифрового вывода

DacOutDescr	TLTR35_DAC_OUT_DESCR [LTR35_DAC_CH_OUTPUT_CNT]	Описание параметров выходов ЦАП для данной модификации модуля.
Reserved1	DWORD [26]	Резервные поля
CbrCoef	TLTR35_CBR_COEF [LTR35_DAC_CHANNEL_CNT] [LTR35_DAC_CH_OUTPUT_CNT]	Заводские калибровочные коэффициенты.
CbrAfcCoef	TLTR35_AFC_COEF	Заводские коэффициенты для коррекции АЧХ модуля.
Reserved2	DWORD [64 *LTR35_DAC_CHANNEL_CNT *LTR35_DAC_CH_OUTPUT_CNT - 24]	Дополнительные резервные поля

4.2.5 Настройки канала ЦАП

Тип: TLTR35_CHANNEL_CONFIG		
Описание: Структура, содержащая настройки одного канала ЦАП.		
Поле	Тип	Описание поля
Enabled	BOOLEAN	Разрешение выдачи сигнала для данного канала.
Output	BYTE	Используемый выход для данного канала (значение из e_LTR35_DAC_OUTPUT).
Source	BYTE	Источник данных для данного канала (значение из e_LTR35_CH_SRC).
ArithAmp	double	Амплитуда сигнала в режиме арифметического генератора.
ArithOffs	double	Смещение сигнала в режиме арифметического генератора.
Reserved	DWORD [10]	Резервные поля.

4.2.6 Значение фазы для арифметических генераторов

Тип: TLTR35_ARITH_PHASE		
Описание: Структура используется для задания значения фазы в настройках арифметического генератора. Для заполнения данной структуры по требуемому значению фазы в градусах можно использовать функцию LTR35_FillArithPhaseDegree() .		
Поле	Тип	Описание поля
Reserved	DWORD	Резерв
CodeH	DWORD	32-битный код, задающий фазу. Каждый разряд имеет вес $2\pi/2^{32}$.

4.2.7 Настройки арифметического генератора

Тип: TLTR35_ARITH_SRC_CONFIG		
Описание: Структура, содержащая настройки одного арифметического генератора.		
Поле	Тип	Описание поля
Phase	TLTR35_ARITH_PHASE	Начальная фаза сигнала.
Delta	TLTR35_ARITH_PHASE	Приращение фазы сигнала. На данное значение изменяется фаза каждого следующего выводимого отсчета относительно предыдущего.
Reserved	DWORD [14]	Резервные поля.

4.2.8 Настройки синтезатора частоты

Тип: TLTR35_FREQ_SYNT_CONFIG		
Описание: Структура, содержащая коэффициенты, определяющие результирующую частоту на выходе синтезатора частоты. Для получения входной частоты ЦАП из опорной частоты 30 МГц в модуле используется синтезатор. Полученная на выходе синтезатора частота определяется 3-мя коэффициентами, заданными в данной структуре, и рассчитывается по формуле $f_{synt} = \frac{f_{in} * b}{r * 2 * a}$, где $f_{in} = 30$ МГц. Отношение результирующей частоты генерации ЦАП к полученной частоте синтезатора определяется заданной с помощью поля DacRate скорости выдачи на ЦАП. Также частота с выхода синтезатора является базовой частотой для синхронного ввода с цифровых линий, как указано в описании настройки DIFreqDivPow . Как правило пользователю не требуется заполнять эти параметры вручную. Для их заполнения используется функция LTR35_FillOutFreq() .		
Поле	Тип	Описание поля
b	WORD	Коэффициент b в настройках синтезатора
r	WORD	Коэффициент r в настройках синтезатора
a	BYTE	Коэффициент a в настройках синтезатора

4.2.9 Настройки синхронного ввода

Тип: TLTR35_IN_STREAM_CONFIG		
Описание: Данная структура содержит параметры синхронного ввода данных, выполняемого при необходимости одновременно с выводом.		
Поле	Тип	Описание поля
InStreamMode	BYTE	Используемый режим синхронного ввода. Значение из e_LTR35_IN_STREAM_MODE .
EchoChannel	BYTE	Для режима синхронного ввода LTR35_IN_STREAM_MODE_CH_ECHO данное поле задает номер канала (начиная с 0), выводимые значения которого будут содержать эхо-данные. Данный канал должен быть разрешен, чтобы эти данные были корректны.

DlChEnMask	BYTE	Для режима синхронного ввода LTR35_IN_STREAM_MODE_DI данное поле задает, какие именно цифровые входы будут опрашиваться. Данное поле определяется как комбинация флагов из e_LTR35_IN_STREAM_DI_CHS , объединенных по битовому “ИЛИ”. При режиме ввода LTR35_IN_STREAM_MODE_DI должен быть разрешен опрос хотя бы одного входа.
DIFreqDivPow	BYTE	Для режима синхронного ввода LTR35_IN_STREAM_MODE_DI данное поле задает значение степени делителя в формуле определения частоты синхронного ввода. Данное поле может принимать значения от 0 (при опросе одного входа) или 1 (если опрашиваются оба входа) до LTR35_DI_STREAM_FREQ_DIV_POW_MAX . Формула для определения частоты опроса описана в разделе Настройка синхронного ввода . Также для заполнения этого поля можно использовать функцию LTR35_FillDIAcqFreq() .
Reserved	DWORD [7]	Резервные поля.

4.2.10 Настройки синхронизации старта

Тип: TLTR35_SYNC_CONFIG		
Описание: Данная структура содержит параметры синхронизации запуска ввода-вывода модуля с другими модулями или с внешним сигналом.		
Поле	Тип	Описание поля
SyncMode	BYTE	Режим синхронизации старта. Значение из e_LTR35_SYNC_MODE .
SlaveSrc	BYTE	Для режима синхронизации LTR35_SYNC_MODE_SLAVE данное поле задает условие запуска ввода-вывода. Значение из e_LTR35_SYNC_SLAVE_SRC .
Reserved	DWORD [7]	Резервные поля.

4.2.11 Настройки модуля

Тип: TLTR35_CONFIG		
Описание: Структура содержит все настройки модуля, которые должен заполнить пользователь перед вызовом LTR35_Configure() .		
Поле	Тип	Описание поля
Ch	TLTR35_CHANNEL_CONFIG [LTR35_DAC_CHANNEL_CNT]	Настройки каналов ЦАП.
ArithSrc	TLTR35_ARITH_SRC_CONFIG [LTR35_ARITH_SRC_CNT]	Настройки арифметических генераторов.
ArithSrcReserved	TLTR35_ARITH_SRC_CONFIG [LTR35_ARITH_SRC_CNT]	Резервные поля (должны быть установлены в 0)

OutMode	BYTE	Режим вывода модуля (значение из <code>e_LTR35_OUT_MODE</code>).
OutDataFmt	BYTE	Формат данных для вывода (значение из <code>e_LTR35_OUTDATA_FORMAT</code>).
DacRate	BYTE	Скорость выдачи данных на ЦАП (константа из <code>e_LTR35_RATE</code>). Как правило заполняется с помощью функции <code>LTR35_FillOutFreq()</code> .
FreqSynt	<code>TLTR35_FREQ_SYNT_CONFIG</code>	Настройки синтезатора. Как правило заполняются с помощью функции <code>LTR35_FillOutFreq()</code> .
StreamStatusPeriod	WORD	Период передачи статусных слов. В потоковом режиме (<code>OutMode = LTR35_OUT_MODE_STREAM</code>) статусное слово будет передаваться после вывода каждых <code>StreamStatusPeriod</code> слов из буфера. 0 означает выбор значения по-умолчанию.
Flags	DWORD	Дополнительные флаги конфигурации. Объединение по битовому “ИЛИ” флагов из перечисления <code>e_LTR35_CFG_FLAGS</code> .
InStream	<code>TLTR35_IN_STREAM_CONFIG</code>	Настройки синхронного ввода.
Sync	<code>TLTR35_SYNC_CONFIG</code>	Настройки синхронизации запуска нескольких модулей.
Reserved	DWORD [47]	Резервные поля (должны быть установлены в 0).

4.2.12 Параметры текущего состояния модуля

Тип: <code>TLTR35_STATE</code>		
Описание: Структура, содержащая параметры модуля, которые пользователь должен использовать только для чтения, так как они изменяются только внутри функций <code>ltr35api</code> .		
Поле	Тип	Описание поля
FpgaState	BYTE	Текущее состояние ПЛИС. Одно из значений из <code>e_LTR_FPGA_STATE</code> .
Run	BOOLEAN	Признак, запущен ли сейчас ввод-вывод данных.

SwAckWaiting	BOOLEAN	Признак, что модуль находится в состоянии ожидания ответа на запрос начала генерации или переключения страниц. Устанавливается при вызове <code>LTR35_SwitchCyclePageRequest()</code> / <code>LTR35_StreamStartRequest()</code> и сбрасывается после успешного завершения <code>LTR35_SwitchCyclePageWaitDone()</code> / <code>LTR35_StreamStartWaitDone()</code> .
OutFreq	double	Установленная частота генерации ЦАП (Гц) и вывода на цифровые линии. Обновляется после вызова <code>LTR35_Configure()</code> .
EnabledChCnt	BYTE	Количество разрешенных каналов ЦАП. Обновляется после вызова <code>LTR35_Configure()</code> .
SDRAMChCnt	BYTE	Количество разрешенных каналов ЦАП, отсчеты для которых берутся из буфера модуля. Обновляется после вызова <code>LTR35_Configure()</code> .
ArithChCnt	BYTE	Количество разрешенных каналов ЦАП, настроенных на режим арифметического генератора. Обновляется после вызова <code>LTR35_Configure()</code> .
InStreamDIChCnt	BYTE	Количество опрашиваемых цифровых входов в режиме <code>LTR35_IN_STREAM_MODE_DI</code> . Обновляется после вызова <code>LTR35_Configure()</code> .
InStreamDIAcqFreq	double	Частота опроса (Гц) каналов цифровых входов <code>LTR35_IN_STREAM_MODE_DI</code> (соответствует одному биту каждого опрашиваемого канала). Обновляется после вызова <code>LTR35_Configure()</code> .
InStreamWordFreq	double	Установленная частота (Гц) передачи слов из модуля в ПК при разрешенном синхронном вводе. Обновляется после вызова <code>LTR35_Configure()</code> .
SyntFreq	double	Частота синтезатора (Гц). Обновляется после вызова <code>LTR35_Configure()</code> .
Reserved	DWORD [26]	Резервные поля.

4.2.13 Управляющая структура модуля

Тип: TLTR35		
Описание: Структура, которая хранит всю необходимую информацию о модуле и связи с ним: текущие настройки модуля, информацию о его состоянии, структуру канала связи и т.д. При работе с несколькими модулями одновременно для каждого модуля должен быть создан свой описатель. Передается в большинство функций библиотеки. Некоторые поля структуры доступны для изменения пользователем для настройки параметров модуля. Перед использованием требует инициализации с помощью функции LTR35_Init() .		
Поле	Тип	Описание поля
size	INT	Размер структуры. Заполняется в LTR35_Init() .
Channel	TLTR	Структура, содержащая состояние клиентского соединения со службой ltrd. Не используется напрямую пользователем.
Internal	void *	Указатель на непрозрачную структуру с внутренними параметрами, используемыми исключительно библиотекой и недоступными для пользователя.
Cfg	TLTR35_CONFIG	Настройки модуля. Заполняются пользователем перед вызовом LTR35_Configure() .
State	TLTR35_STATE	Состояние модуля и рассчитанные параметры. Поля изменяются функциями библиотеки. Пользовательской программой могут использоваться только для чтения.
ModuleInfo	TLTR35_MODULE_INFO	Информация о модуле.

4.3 Функции

4.3.1 Функции инициализации и работы с соединением с модулем

4.3.1.1 Инициализация описателя модуля

Формат: INT LTR35_Init (TLTR35 *hnd)
Описание: Функция инициализирует поля структуры описателя модуля значениями по умолчанию. Эта функция должна вызываться для каждой структуры TLTR35 перед вызовом остальных функций.
Параметры: hnd — Описатель модуля
Возвращаемое значение: Код ошибки

4.3.1.2 Открытие соединения с модулем

Формат: INT LTR35_Open (TLTR35 *hnd, DWORD ltrd_addr, WORD ltrd_port, const CHAR *csn, WORD slot)
Описание: <p>Функция устанавливает соединение с модулем в соответствии с переданными параметрами, проверяет наличие модуля и считывает информацию о нем. Должна быть вызвана перед работой с модулем. После завершения работы необходимо закрыть соединение с помощью LTR35_Close().</p> <p>В случае возникновения некритических ошибок, функция может вернуть код ошибки, но оставить соединение открытым, но не в полностью рабочем состоянии. Проверить, осталось ли соединение открытым, можно с помощью LTR35_IsOpened().</p>
Параметры: <p>hnd — Описатель модуля</p> <p>ltrd_addr — IP-адрес машины, на которой запущена служба ltrd, в 32-битном формате (описан в разделе “Формат задания IP-адресов” руководства для библиотеки ltrapi). Если служба ltrd запущена на той же машине, что и программа, вызывающая данную функцию, то в качестве адреса можно передать LTRD_ADDR_DEFAULT.</p> <p>ltrd_port — TCP-порт для подключения к службе ltrd. По умолчанию используется LTRD_PORT_DEFAULT.</p> <p>csn — Серийный номер крейта, в котором находится интересующий модуль. Представляет собой оканчивающуюся нулем ASCII-строку. Для соединения с первым найденным крейтом можно передать пустую строку или нулевой указатель.</p> <p>slot — Номер слота крейта, в котором установлен интересующий модуль. Значение от LTR_CC_CHNUM_MODULE1 до LTR_CC_CHNUM_MODULE16.</p>
Возвращаемое значение: Код ошибки .

4.3.1.3 Закрытие соединения с модулем

Формат: INT LTR35_Close (TLTR35 *hnd)
Описание: <p>Функция закрывает ранее открытое с помощью LTR35_Open() соединение. Должна вызываться после завершения работы с модулем. При любом возвращенном значении после вызова этой функции соответствующий описатель уже нельзя использовать для работы с модулем без открытия нового соединения.</p>
Параметры: <p>hnd — Описатель модуля</p>
Возвращаемое значение: Код ошибки

4.3.1.4 Проверка, открыто ли соединение с модулем

Формат: INT LTR35_IsOpened (TLTR35 *hnd)
Описание: Функция проверяет, открыто ли в данный момент соединение с модулем. Если соединение открыто, функция возвращает LTR_OK, если закрыто — код ошибки LTR_ERROR_CHANNEL_CLOSED.
Параметры: hnd — Описатель модуля
Возвращаемое значение: Код ошибки (LTR_OK, если соединение установлено)

4.3.2 Функции для изменения настроек модуля

4.3.2.1 Подбор коэффициентов для получения заданной частоты генерации ЦАП и цифрового вывода

Формат: INT LTR35_FillOutFreq (TLTR35_CONFIG *cfg, double freq, double *fnd_freq)
Описание: Функция подбирает нужные настройки (параметры синтезатора из FreqSynt и значение поля DacRate), чтобы результирующая частота генерации ЦАП и цифрового вывода была наиболее близка к указанной пользователем.
Параметры: cfg — Структура с настройками модуля, поля которой будут изменены, чтобы полученная частота преобразования ЦАП была наиболее близка к заданной. Поля, не влияющие на частоту преобразования ЦАП, будут оставлены без изменений. freq — Требуемая частота преобразования ЦАП в Гц. fnd_freq — В данной переменной возвращается реально полученная частота генерации ЦАП в Гц, соответствующая подобранным параметрам. Может быть передан нулевой указатель, если это значение не интересно.
Возвращаемое значение: Код ошибки .

4.3.2.2 Подбор степени делителя частоты ввода с цифровых линий

Формат: INT LTR35_FillDIAcqFreq (TLTR35_CONFIG *cfg, double freq, double *fnd_freq)
Описание: Функция подбирает значение поля DIFreqDivPow так, чтобы полученная частота опроса цифровых входов была наиболее близкой к заданной пользователю. Перед вызовом данной функции уже должны быть заполнены настройки синтезатора (явным заданием значений в поле FreqSynt или с помощью LTR35_FillOutFreq()), а также значения полей InStreamMode и DIChEnMask .
Параметры: cfg — Структура с настройками модуля, в которой должны быть уже заполнены указанные в описании функции поля и в которую будет записано полученное значение поля DIFreqDivPow в случае успешного завершения функции. freq — Требуемая частота опроса синхронных входов в Гц. fnd_freq — В данной переменной возвращается реально полученная частота опроса цифровых линий, соответствующая подобранным параметрам. Может быть передан нулевой указатель, если это значение не интересует.
Возвращаемое значение: Код ошибки .

4.3.2.3 Подбор кода арифметической фазы по значению в градусах.

Формат: INT LTR35_FillArithPhaseDegree (TLTR35_ARITH_PHASE *phase_code, double degree, double *res_degree)
Описание: Функция принимает значение фазы в градусах и подбирает значение кода, задающего фазу арифметического генератора, чтобы полученное значение фазы было наиболее близко к заданному.
Параметры: phase_code — Структура настроек фазы, которая будет заполнена нужными значениями при успешном выполнении функции. degree — Требуемая фаза в градусах. Может принимать значения от 0 (включая) до 360 (не включая) градусов. res_degree — В данной переменной возвращается реально полученное значение фазы в градусах. Может быть передан нулевой указатель, если это значение не интересует.
Возвращаемое значение: Код ошибки .

4.3.2.4 Запись настроек в модуль

Формат: INT LTR35_Configure (TLTR35 *hnd)
Описание: Функция передает настройки, соответствующие значением полей структуры конфигурации из поля Cfg описателя модуля, в сам модуль. Должна вызываться перед началом передачи данных в модуль.
Параметры: hnd — Описатель модуля
Возвращаемое значение: Код ошибки.

4.3.3 Функции для управления выдачей данных

4.3.3.1 Передача данных ЦАП и цифровых выходов в модуль

Формат: INT LTR35_Send (TLTR35 *hnd, const DWORD *data, DWORD size, DWORD timeout)
Описание: Функция передает данные для вывода в модуль. Данные должны быть в специальном формате. Подготовить данные в нужном формате можно с помощью функций LTR35_PrepareDacData() или LTR35_PrepareData() . Завершение функции означает только то, что данные записаны в буфер канала обмена со службой ltrd. Данные могут дойти до самого модуля и быть выведены позже. Если места в буфере нет, то функция будет ожидать освобождения. Выход из функции осуществляется либо когда все данные будут записаны в буфер, либо когда истечет указанный таймаут.
Параметры: hnd — Описатель модуля. data — Массив со словами, которые необходимо передать модулю. size — Количество передаваемых 32-битных слов. timeout — Таймаут на выполнение функции в мс.
Возвращаемое значение: Значение меньше 0 соответствует коду ошибки. Значение больше или равное нулю — количество записанных слов.

4.3.3.2 Подготовка данных для передачи в модуль

<p>Формат: INT LTR35_PrepareData (TLTR35 *hnd, const double *dac_data, DWORD *dac_size, const DWORD *dout_data, DWORD *dout_size, DWORD flags, DWORD *result, DWORD *snd_size)</p>
<p>Описание:</p> <p>Функция принимает отсчеты ЦАП и цифровых линий и на основе этих данных формирует слова специального формата для дальнейшей передачи в модуль с помощью LTR35_Send().</p> <p>Данные от ЦАП принимаются в следующем порядке: 1-ый отсчет первого разрешенного канала с источником из буфера модуля, затем 1-ый отсчет второго такого канала и т.д., затем по второму отсчету каждого канала и т.д. Количество данных ЦАП на входе должно быть кратно количеству разрешенных каналов с источником LTR35_CH_SRC_SDRAM (поле SDRAMChCnt структуры состояния модуля State). Данные для ЦАП могут быть либо в кодах ЦАП, либо в Вольтах (если передан флаг e_LTR35_PREP_FLAGS). При этом калибровки применяются всегда аппаратно внутри модуля.</p> <p>Также функция принимает данные цифровых линий — младшие биты соответствуют значениям цифровых линий, которые могут быть объединены по битовому “ИЛИ” с флагами из e_LTR35_DOUTS_WORD_FLAGS. На каждые SDRAMChCnt отсчетов ЦАП в выходной буфер помещается одно слово цифровых линий.</p> <p>Формат выходных слов и количество слов на отсчет определяется заданным форматом передаваемых данных. Функция завершает работу как только закончится один из входных массивов или место в выходном буфере.</p>
<p>Параметры:</p> <p>hnd — Описатель модуля.</p> <p>dac_data — Массив отсчетов ЦАП. Может быть передан нулевой указатель, если используются только вывод на цифровые линии.</p> <p>dac_size — На входе задает количество отсчетов в массиве <code>dac_data</code>, на выходе возвращает, сколько отсчетов было реально записано в выходной массив.</p> <p>dout_data — Массив значений для вывода на цифровые линии. Может быть передан нулевой указатель, если используются только данные ЦАП.</p> <p>dout_size — На входе задает количество значений в массиве <code>dout_data</code>, на выходе возвращает, сколько значений было реально записано в выходной массив.</p> <p>flags — Набор флагов из e_LTR35_PREP_FLAGS.</p> <p>result — Результирующий массив в специальном формате для передачи в модуль.</p> <p>snd_size — На входе задает размер массива <code>result</code>, на выходе возвращает, сколько всего 32-битных слов было записано в массив <code>result</code> в ходе выполнения функции.</p>
<p>Возвращаемое значение:</p> <p>Код ошибки.</p>

4.3.3.3 Подготовка данных ЦАП для передачи в модуль

Формат: INT LTR35_PrepareDacData (TLTR35 *hnd, const double *dac_data, DWORD size, DWORD flags, DWORD *result, DWORD *snd_size)
Описание: Данная функция является упрощенным вариантом LTR35_PrepareData() , но подготавливает только данные ЦАП (без поддержки вывода на цифровые выходы). Выходной массив должен быть обязательно достаточного размера для записи слов, соответствующих всем переданным отсчетам ЦАП.
Параметры: hnd — Описатель модуля. dac_data — Массив отсчетов ЦАП. Формат аналогичен LTR35_PrepareData() . size — Количество отсчетов в массиве dac_data. Значение должно быть кратно SDRAMChCnt . flags — Набор флагов из e_LTR35_PREP_FLAGS result — Результирующий массив в специальном формате для передачи в модуль. Массив должен быть размером не меньше $n*size$, где $n = 1$ или 2 в зависимости от формата передаваемых данных . snd_size — Если не нулевой указатель, то в данной переменной возвращается количество подготовленных слов для передачи в модуль.
Возвращаемое значение: Код ошибки .

4.3.3.4 Смена страницы вывода в режиме циклического автогенератора

Формат: INT LTR35_SwitchCyclePage (LTR35 *hnd, DWORD flags, DWORD tout)
Описание: <p>При вызове данной функции начинается циклическая выдача данных с ранее загруженной страницы буфера модуля. Перед вызовом выводимые данные должны быть загружены с помощью LTR35_Send().</p> <p>При успешном завершении функции для записи становится доступна следующая страница.</p> <p>Функция доступна только в режиме циклического автогенератора (<code>OutMode = LTR35_OUT_MODE_CYCLE</code>).</p> <p>Если при вызове этой функции выдача данных еще не была запущена (первый вызов после LTR35_Configure() или LTR35_Stop()), то по вызову функции будет запущена выдача отсчетов, записанных до этого в первую страницу циклического буфера.</p> <p>Если при вызове уже идет вывод циклической страницы, то при выводе последнего отсчета с этой страницы происходит смена страниц, т.е. модуль начинает выводить данные, соответствующие загруженной до этого странице, а ранее выводимая страница становится доступна для записи новых данных.</p> <p>Функция возвращает управления после получения подтверждения смены страницы. Так как для этого все ранее записанные данные должны быть переданы в модуль и кроме того, если уже идет выдача данных, то она должна дойти до последнего отсчета из выдаваемой страницы, то время ответа может быть значительным. Именно поэтому таймаут задается вручную через параметр функции.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>flags — Флаги (резерв — должен передаваться 0).</p> <p>tout — Таймаут в мс на время выполнения функции (на время ожидания ответа о завершении смены страницы).</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.5 Запрос смены страницы вывода в режиме циклического автогенератора

Формат: INT LTR35_SwitchCyclePageRequest (LTR35 *hnd, DWORD flags)
Описание: <p>Функция аналогична LTR35_SwitchCyclePage(), но не дожидается ответа от модуля о том, что смена страниц завершилась. Ожидание завершения смены страниц выполняется с помощью функции LTR35_SwitchCyclePageWaitDone().</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>flags — Флаги (резерв — должен передаваться 0).</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.6 Ожидание завершения смены страницы в режиме циклического автогенератора

Формат: INT LTR35_SwitchCyclePageWaitDone (TLTR35 *hnd, DWORD tout)
Описание: Данная функция ожидает возвращения подтверждения от модуля об успешной смене страниц после вызова LTR35_SwitchCyclePageRequest() . Если подтверждение смены страницы успешно принято, то данная функция вернет LTR_OK. Если же ответа дождаться не удалось, то функция вернет ошибку LTR_ERROR_OP_DONE_WAIT_TOUT. В этом случае данную функции можно вызывать повторно для продолжения ожидания, либо вызвать LTR35_Stop() для отмены смены страницы и останова вывода.
Параметры: hnd — Описатель модуля. tout — Максимальное время ожидания подтверждения.
Возвращаемое значение: Код ошибки .

4.3.3.7 Запуск ввода-вывода в потоковом режиме

Формат: INT LTR35_StreamStart (TLTR35 *hnd, DWORD flags)
Описание: При вызове данной функции начинается выдача данных на ЦАП в потоковом режиме (OutMode = LTR35_OUT_MODE_STREAM). При этом для реализации непрерывного вывода часть данных уже должна быть предзагружена в буфер модуля с помощью LTR35_Send() .
Параметры: hnd — Описатель модуля. flags — Флаги (резерв — должен передаваться 0).
Возвращаемое значение: Код ошибки .

4.3.3.8 Запрос на запуск ввода-вывода в потоковом режиме

Формат: INT LTR35_StreamStartRequest (TLTR35 *hnd, DWORD flags)
Описание: <p>Функция аналогична LTR35_StreamStart(), но не дожидается ответа от модуля о том, что запуск произошел. Ожидание запуска выполняется с помощью функции LTR35_StreamStartWaitDone().</p> <p>Использование такого разделения удобно в частности для запуска ведомого модуля (LTR35_SYNC_MODE_SLAVE), так как в этом случае между посылкой команды запуска с помощью LTR35_StreamStartRequest() и до собственно осуществления запуска по внешнему условию синхронизации (после чего только и вернет модуль ответ) может пройти значительное время.</p>
Параметры: <p>hnd — Описатель модуля. flags — Флаги (резерв — должен передаваться 0).</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.9 Ожидание запуска ввода-вывода в потоковом режиме

Формат: INT LTR35_StreamStartWaitDone (TLTR35 *hnd, DWORD tout)
Описание: <p>Данная функция ожидает возвращения подтверждения от модуля об успешном запуске ввода-вывода после вызова LTR35_StreamStartRequest().</p> <p>Если подтверждение запуска успешно принято, то данная функция вернет LTR_OK.</p> <p>Если же ответа дожидаться не удалось, то функция вернет ошибку LTR_ERROR_OP_DONE_WAIT_TOUT. В этом случае данную функции можно вызывать повторно для продолжения ожидания, либо вызвать LTR35_Stop() для отмены запуска ввода-вывода.</p>
Параметры: <p>hnd — Описатель модуля. tout — Максимальное время ожидания подтверждения.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.10 Останов выдачи данных

Формат: INT LTR35_Stop (TLTR35 *hnd, DWORD flags)
Описание: <p>Функция завершает выдачу данных на ЦАП и/или цифровые выходы как в потоковом режиме, так и в режиме циклического автогенератора, останавливает каналы в режиме арифметического генератора, а также останавливает синхронный ввод данных.</p> <p>При этом следует учесть, что в потоковом режиме, если сильно заполнен буфер ltrd, то команда дойдет до модуля только после того, как все слова данных из буфера будут переданы в модуль, поэтому в этом случае время выполнения может быть значительное.</p> <p>Для этого случая сделана функция LTR35_StopWithTout(), в которой явно можно указать время ожидания ответа на команду останова.</p>
Параметры: <p>hnd — Описатель модуля. flags — Флаги (резерв — должен передаваться 0).</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.11 Останов выдачи данных с заданным временем ожидания ответа

Формат: INT LTR35_StopWithTout (TLTR35 *hnd, DWORD flags, DWORD tout)
Описание: <p>Функция аналогична LTR35_Stop() за исключением того, что время ожидания ответа можно задать явно в мс. Служит для случая, если время ответа может быть существенно больше стандартного времени ответа на команду (например, в потоковом режиме, если буфер ltrd заполнен данными на передачу).</p>
Параметры: <p>hnd — Описатель модуля. flags — Флаги (резерв — должен передаваться 0). tout — Время ожидания ответа в мс</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.12 Изменение частоты для арифметического генератора

Формат: INT LTR35_SetArithSrcDelta (TLTR35 *hnd, BYTE gen_num, const TLTR35_ARITH_PHASE *delta)
Описание: <p>Функция позволяет изменить частоту арифметического генератора при уже запущенной выдаче данных без ее останова и повторного вызова LTR35_Configure(). При этом частота задается также как и при настройке с помощью значения приращения фазы на каждый выводимый отсчет. При успешном выполнении функции соответствующим образом обновляется поле Delta настроек соответствующего арифметического генератора.</p>
Параметры: <p>hnd — Описатель модуля. gen_num — Номер арифметического генератора (от 0 до LTR35_ARITH_SRC_CNT-1). delta — Новое значение приращения фазы на период вывода на ЦАП.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.13 Изменение амплитуды и смещения арифметического сигнала.

Формат: INT LTR35_SetArithAmp (TLTR35 *hnd, BYTE ch_num, double amp, double offset)
Описание: <p>Функция позволяет изменить амплитуду и смещение сигнала канала ЦАП, источником которого является один из арифметических генераторов, при уже запущенной выдаче данных без ее останова и повторного вызова LTR35_Configure(). При успешном выполнении соответствующим образом обновляются поля ArithAmp и ArithOffs настроек соответствующего канала.</p>
Параметры: <p>hnd — Описатель модуля. ch_num — Номер канала ЦАП. amp — Новое значение амплитуды в Вольтах. offset — Новое значение смещения в Вольтах.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.14 Прием данных входного потока от модуля

Формат: INT LTR35_RecvInStreamData (TLTR35 *hnd, INT *data, DWORD *tmark, DWORD size, DWORD timeout)
Описание: Функция позволяет принять слова, передаваемые модулем, при разрешенном синхронном вводе. Формат принятых слов зависит от режима синхронного ввода и описан в разделе Прием данных при синхронном вводе .
Параметры: hnd — Описатель модуля. data — Массив, в который будут сохранены принятые синхронного ввода. Должен быть размером на size 32-битных слов. tmark — Указатель на массив размером на size 32-битных слов, в который будут сохранены значения счетчиков синхрометок, соответствующие принятым данным. Генерация меток настраивается для крейта или специального модуля отдельно. Синхрометки подробнее описаны в разделе “Синхрометки” руководства для библиотеки ltrapi . Если синхрометки не используются, то можно передать в качестве параметра нулевой указатель. size — Запрашиваемое количество 32-битных слов на прием. timeout — Таймаут на выполнение операции в миллисекундах. Если в течение заданного времени не будет принято запрашиваемое количество слов, то функция все равно вернет управление, возвратив в качестве результата реально принятое количество слов
Возвращаемое значение: Значение меньше нуля соответствует коду ошибки . Значение больше или равно нулю соответствует количеству реально принятых и сохраненных в массив data слов.

4.3.4 Функции для работы с flash-памятью модуля

4.3.4.1 Чтение данных из flash-памяти модуля

Формат: INT LTR35_FlashRead (TLTR35 *hnd, DWORD addr, BYTE *data, DWORD size)
Описание: Функция вычитывает данные, записанные во flash-памяти модуля по заданному адресу. Пользователю выделяется область памяти с адреса LTR35_FLASH_USERDATA_ADDR размером LTR35_FLASH_USERDATA_SIZE байт.
Параметры: hnd — Описатель модуля. addr — Адрес памяти, начиная с которого необходимо прочитать данные. data — Массив на size байт, в который будут записаны прочитанные из Flash-памяти данные. size — Количество данных в байтах, которое необходимо прочитать.
Возвращаемое значение: Код ошибки .

4.3.4.2 Запись данных во flash-память модуля

Формат: INT LTR35_FlashWrite (TLTR35 *hnd, DWORD addr, const BYTE *data, DWORD size, DWORD flags)
Описание: <p>Функция записывает данные во flash-памяти модуля по заданному адресу. Запись может выполняться и в не стертые области. Однако в стерты заранее с помощью LTR35_FlashErase() области запись выполняется быстрее, если указан флаг LTR35_FLASH_WRITE_ALREADY_ERASED.</p> <p>Пользователю выделяется область памяти с адреса LTR35_FLASH_USERDATA_ADDR размером LTR35_FLASH_USERDATA_SIZE байт.</p>
Параметры: <p>hnd — Описатель модуля. addr — Адрес памяти, начиная с которого необходимо выполнить запись data — Массив из size байт с данными, которые будут записаны. size — Количество данных в байтах, которое необходимо записать. flags — Набор флагов из e_LTR35_FLASH_WRITE_FLAGS, объединенных по битовому “ИЛИ”.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.4.3 Стирание области flash-память модуля

Формат: INT LTR35_FlashErase (TLTR35 *hnd, DWORD addr, DWORD size)
Описание: <p>Функция стирает область во flash-память модуля по заданному адресу. Стирание необходимо выполнять перед записью данных. Стирание возможно только блоками, кратными LTR35_FLASH_ERASE_BLOCK_SIZE байт. Пользователю выделяется область памяти с адреса LTR35_FLASH_USERDATA_ADDR размером LTR35_FLASH_USERDATA_SIZE байт.</p>
Параметры: <p>hnd — Описатель модуля. addr — Адрес памяти, начиная с которого необходимо выполнить стирание. size — Размер стираемой области в байтах. Должен быть кратен LTR35_FLASH_ERASE_BLOCK_SIZE.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.5 Функции вспомогательного характера

4.3.5.1 Получение сообщения об ошибке

Формат: LPCSTR LTR35_GetErrorString (INT err)
Описание: Функция возвращает строку, соответствующую переданному коду ошибки, в кодировке CP1251 для ОС Windows или UTF-8 для ОС Linux. Функция может обрабатывать как ошибки из ltr35api, так и общие коды ошибок из ltrapi.
Параметры: err — Код ошибки.
Возвращаемое значение: Указатель на строку, содержащую сообщение об ошибке.

4.3.5.2 Проверка, разрешена ли работа ПЛИС модуля

Формат: INT LTR35_FPGAIsEnabled (TLTR35 *hnd, BOOL *enabled)
Описание: Функция проверяет, разрешена ли работа ПЛИС модуля. Для настройки и генерации данных ПЛИС должен быть всегда разрешен.
Параметры: hnd — Описатель модуля. enabled — В случае успешного выполнения функции в этой переменной возвращается FALSE, если ПЛИС запрещен, или TRUE в противном случае.
Возвращаемое значение: Код ошибки.

4.3.5.3 Разрешение работы ПЛИС модуля

Формат: INT LTR35_FPGAEnable (TLTR35 *hnd, BOOL enable)
Описание: Функция разрешает или запрещает работу ПЛИС модуля. Для настройки и генерации данных ПЛИС должен быть всегда разрешен. В LTR35_Open() выполняется разрешение работы ПЛИС, если была найдена прошивка ПЛИС в памяти модуля и она была успешно загружена, поэтому при штатной работе данная функция не используется.
Параметры: hnd — Описатель модуля. enable — Если FALSE — запрет работы ПЛИС, иначе — разрешение.
Возвращаемое значение: Код ошибки.

4.3.5.4 Получение информации о состоянии модуля

Формат: INT LTR35_GetStatus (TLTR35 *hnd, DWORD *status)
Описание: Функция позволяет получить набор флагов, описывающих состояние работы модуля.
Параметры: hnd — Описатель модуля. status — Набор флагов из e_LTR35_STATUS , объединенных по битовому “ИЛИ”, описывающих состояние модуля.
Возвращаемое значение: Код ошибки .

4.3.5.5 Чтение информации и калибровочных коэффициентов

Формат: INT LTR35_GetConfig (TLTR35 *hnd)
Описание: Функция считывает информацию из flash-памяти модуля и обновляет поля ModuleInfo в управляющей структуре модуля. Так как данная операция уже выполняется при вызове LTR35_Open() , то как правило вызов данной функции не требуется. Однако данная функция может быть использована для восстановления измененных коэффициентов в ModuleInfo на заводские.
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.5.6 Сброс микросхемы ЦАП

Формат: INT LTR35_DacReset (TLTR35 *hnd)
Описание: Функция выполняет программный сброс микросхем ЦАП. После этого необходимо выполнить настройку модуля с помощью LTR35_Configure() для настройки ЦАП на нужный режим. Вспомогательная функция, для работы модуля явный сброс ЦАП не требуется.
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.5.7 Асинхронное чтение состояния цифровых входов

Формат: INT LTR35_DIAsyncIn (TLTR35 *hnd, BYTE *di_state)
Описание: Функция выполняет чтения состояния входов DI1 и DI2. Синхронный ввод данных при этом не должен быть запущен.
Параметры: hnd — Описатель модуля. di_state — При успешном выполнении в данной переменной возвращается битовая маска состояний входов DI1 (младший бит) и DI2 (второй бит).
Возвращаемое значение: Код ошибки.