

МНОГОКАНАЛЬНЫЕ СИСТЕМЫ СБОРА ДАННЫХ

LTR12

РУКОВОДСТВО ПРОГРАММИСТА

Ревизия 1.0.0
Март 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	12.03.2020	Первая ревизия данного документа

Оглавление

1	О чем этот документ	5
2	Установка и подключение библиотеки к проекту	6
3	Общий подход к работе с библиотекой	7
3.1	Общий алгоритм работы с модулем	7
3.2	Настройка модуля	8
3.2.1	Настройка последовательности и параметров опроса каналов	8
3.2.2	Источник запуска сбора данных	9
3.2.3	Источник запуска и частота преобразований АЦП	9
3.3	Прием и обработка данных	10
3.3.1	Особенности калибровки данных модулем	10
3.3.2	Особенности обработки измерений собственного нуля	11
3.4	Корректировка смещение нуля	11
4	Константы, типы данных и функции библиотеки	13
4.1	Константы и перечисления	13
4.1.1	Константы и макроопределения	13
4.1.2	Коды ошибок, специфичные для модуля LTR12.	14
4.1.3	Источник запуска сбора данных модулем	15
4.1.4	Источник запуска преобразования АЦП	15
4.1.5	Режим измерения логического канала	16
4.1.6	Флаги, управляющие обработкой данных	16
4.2	Типы данных	18
4.2.1	Калибровочные коэффициенты модуля	18
4.2.2	Информация о модуле	18
4.2.3	Параметры текущего состояния модуля	19
4.2.4	Настройки логического канала	20
4.2.5	Настройки частоты АЦП	20
4.2.6	Настройки модуля	21
4.2.7	Описатель модуля модуля	22
4.3	Функции	22
4.3.1	Функции инициализации и работы с соединением с модулем	22
4.3.1.1	Инициализация описателя модуля	22
4.3.1.2	Открытие соединения с модулем	23
4.3.1.3	Закрытие соединения с модулем	23
4.3.1.4	Проверка, открыто ли соединение с модулем.	24
4.3.2	Функции для изменения настроек модуля	24
4.3.2.1	Запись настроек в модуль	24
4.3.2.2	Вычисление частоты АЦП	24

4.3.2.3	Подбор параметров частоты АЦП	25
4.3.2.4	Заполнение параметров частоты АЦП	25
4.3.2.5	Заполнение параметров канала логической таблицы	26
4.3.3	Функции для управления сбором данных	26
4.3.3.1	Измерение смещения нуля	26
4.3.3.2	Запуск сбора данных	27
4.3.3.3	Останов сбора данных	27
4.3.3.4	Прием данных от модуля	28
4.3.3.5	Обработка принятых от модуля слов	29
4.3.3.6	Сбор одного кадра	30
4.3.3.7	Поиск начала первого кадра	30
4.3.4	Функции для работы с flash-памятью модуля	31
4.3.4.1	Чтение данных из пользовательской области flash-памяти модуля	31
4.3.4.2	Запись данных в пользовательскую область flash-память модуля	31
4.3.5	Функции вспомогательного характера	32
4.3.5.1	Чтение информации и калибровочных коэффициентов	32
4.3.5.2	Получение сообщения об ошибке	32

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 LTR12 с использованием предоставляемой фирмой “Л Кард” библиотеки `ltr12api`.

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 LTR12. Более подробно каждый шаг будет описан в последующих разделах.

Типичная последовательность действий имеет следующий вид:

1. Создать экземпляр структуры [TLTR12](#), представляющей собой описатель модуля. Описатель модуля содержит всю информацию о модуле и используется при вызове всех остальных функций.
2. Проинициализировать поля описателя с помощью [LTR12_Init\(\)](#)
3. Установить соединение с интересующим модулем с помощью функции [LTR12_Open\(\)](#).
4. Заполнить необходимые поля с настройками модуля подструктуры [Cfg описателя модуля](#) и вызвать [LTR12_SetADC\(\)](#) для записи настроек в модуль. Для заполнения части полей доступны вспомогательные функции.
5. При необходимости выполнить измерение начального смещения нуля АЦП с помощью [LTR12_MeasAdcZeroOffset\(\)](#).
6. Запуск сбора данных с помощью [LTR12_Start\(\)](#).
7. Прием заданного количества слов с помощью [LTR12_Recv\(\)](#). Количество слов должно быть кратно размеру кадра (поле [FrameWordsCount](#) подструктуры [State описателя модуля](#))
8. Обработка принятых слов с помощью [LTR12_ProcessData\(\)](#) для получения результатов измерения.
9. Переход к пункту 7, если нужны еще данные, или к пункту 10 при завершении сбора.
10. По завершению работы выполнить останов сбора данных с помощью [LTR12_Stop\(\)](#).
11. Закрыть соединение с модулем, вызвав функцию [LTR12_Close\(\)](#).

3.2 Настройка модуля

После того как связь с модулем установлена, необходимо выполнить настройку модуля. Она выполняется аналогично большинству других модулей LTR: вначале значения всех параметров записываются в соответствующие поля [структуры описателя модуля](#), затем вызывается функция `LTR12_SetADC()`, которая значения этих полей передает модулю.

Все поля, относящиеся к настройке модуля, объединены в структуру типа `TLTR12_CONFIG` (поле `Cfg` описателя модуля). Именно эти поля пользователь должен изменять для настройки модуля и именно эти поля влияют на параметры, записываемые `LTR12_SetADC()`. Часть полей пользователь может заполнять как явно, так и с помощью вспомогательных функций, которые позволяют рассчитать значение поля исходя из значения желаемого результирующего параметра, определяемого полем.

При настройке модуля задаются следующие параметры:

- Последовательность опроса каналов и режимы измерения (раздел [Настройка последовательности и параметров опроса каналов](#)).
- Источник запуска сбора данных (раздел [Источник запуска сбора данных](#)).
- Источник запуска каждого преобразования АЦП (раздел [Источник запуска и частота преобразований АЦП](#)).
- Частота АЦП при внутреннем источнике запуска преобразований (раздел [Источник запуска и частота преобразований АЦП](#)).

При выполнении `LTR12_SetADC()` функция проверяет корректность заполнения полей конфигурации, а также рассчитывает ряд результирующих параметров, соответствующих настроенной конфигурации, которые будут сохранены в поля структуры состояния модуля `TLTR12_STATE` (поле `State` описателя модуля).

3.2.1 Настройка последовательности и параметров опроса каналов

LTR12 является модулем с коммутацией каналов. Это означает, что на модуле присутствует одна микросхема АЦП и в один момент времени выполняется преобразование лишь по одному каналу, а многоканальность достигается за счет последовательного опроса всех требуемых каналов. Как и в других АЦП “Л Кард” с коммутацией каналов, последовательность опроса каналов и параметры измерения при каждом опросе задаются с помощью логической таблицы, которая представляет собой последовательность логических каналов. Количество логических каналов задается полем `LChCnt`, а сама таблица определяется первыми `LChCnt` элементами массива `LChTbl`.

Параметры логического канала задаются структурой `TLTR12_LCHANNEL` и определяют номер физического канала, который опрашивается за этот цикл, а также используемый режим измерения при данном цикле опроса. Соответственно, модуль сперва опрашивает физический канал, заданный в первом элементе логической таблицы, затем канал, определенный вторым элементом, и так до конца таблицы, после чего начинает снова опрос с первого логического канала.

Один цикл опроса всех каналов из логической таблицы именуется кадром АЦП.

Дополнительно для LTR12 можно задать количество фоновых измерений с помощью поля `BgLChCnt`. Если значение этого поля больше нуля, то в конец кадра после опроса всех каналов логической таблицы будет добавлено соответствующее количество измерений собственного нуля. При обработке принятых данных значения этих измерений не будут сохранены в массив с результатами обработки, но они могут использоваться для автоматической коррекции смещения нуля АЦП, как описано в разделах [Особенности обработки измерений собственного нуля](#) и [Корректировка смещение нуля](#).

Таким образом полная длина кадра (количество преобразований АЦП за один кадр) равна `LChCnt + BgLChCnt`. Это значение сохраняется в поле `FrameWordsCount` состояния модуля после выполнения `LTR12_SetADC()`.

3.2.2 Источник запуска сбора данных

По умолчанию сбор данных с АЦП запускается по команде с ПК при выполнении `LTR12_Start()`. Дополнительно LTR12 поддерживает запуск сбора данных по фронту или спаду внешнего сигнала, подаваемого на вход `START` сигнального разъема модуля. Это позволяет запустить синхронно сбор данных нескольких модулей, подав один и тот же внешний сигнал на все модули.

Задать данный режим можно с помощью поля `AcqStartSrc`.

При запуске по внешнему сигналу при выполнении функции `LTR12_Start()` модуль переходит в режим ожидания соответствующего условия и только при его возникновении запускает опрос каналов АЦП и передачу данных с измерениями. После обнаружения условия и запуска сбора модуль непрерывно выполняет сбор данных больше не анализируя сигнал на входе `START`. Останов сбора всегда выполняется по команде с ПК при вызове `LTR12_Stop()`.

При внешнем старте после вызова `LTR12_Start()` даже если модуль не дождался внешнего сигнала запуска сбора, все равно необходимо вызывать останов сбора с помощью `LTR12_Stop()`, чтобы выйти из режима ожидания внешнего сигнала запуска сбора.

3.2.3 Источник запуска и частота преобразований АЦП

Запуск каждого отдельного преобразования АЦП может выполняться как по внутреннему периодическому сигналу модуля, так и по каждому фронту или спаду внешнего сигнала, подаваемого на вход `SYN` сигнального разъема модуля. Источник сигнала запуска преобразования задается полем `ConvStartSrc`.

При использовании внутреннего сигнала запуска преобразований преобразования выполняются с постоянной частотой, определяемой параметрами структуры в поле `AdcFreqParams` конфигурации модуля. Способ задания частоты приведен в описании структуры `TLTR12_ADCFREQ_CFG`. Для подбора значений параметров данной структуры по значению требуемой частотой преобразования можно воспользоваться функциями `LTR12_FillAdcFreqParam()` или `LTR12_FindAdcFreqParams()`.

Также результирующее значение частоты будет сохранено в поле `AdcFreq` структуры состояния модуля после вызова `LTR12_SetADC()`. Как и во всех АЦП с коммутацией каналов, так как опрос разных каналов выполняется последовательно, частота АЦП разделяется между всеми каналами. Частота дискретизации сигнала отдельного логического канала будет соответствовать частоте следования кадров и равна $F_{ch} = F_{adc} / (LChCnt + BgLChCnt)$. Значение этой частоты будет сохранено в поле `FrameFreq` структуры состояния модуля после вызова `LTR12_SetADC()`.

При использовании внешнего сигнала преобразования частота преобразований определяется частотой внешнего сигнала (если этот сигнал в принципе является периодическим). Соответственно она не задается в настройках модуля, а полученные значения `AdcFreq` и `FrameFreq` структуры состояния модуля не соответствуют действительности.

3.3 Прием и обработка данных

Прием и обработка данных осуществляется таким же образом, как и во многих других модулях LTR. Сперва осуществляется прием сырых данных в виде 32-битных слов специального формата с помощью `LTR12_Recv()`, после чего эти данные передаются в `LTR12_ProcessData()` для проверки их корректности и перевода в величины (мА). Обработка данных всегда производится над массивами, кратными размеру кадра АЦП.

Количество 32-битных слов в одном кадре определяется полем `FrameWordsCount`, устанавливаемым библиотекой после настройки модуля с помощью `LTR12_SetADC()`, а количество полученных измерений на один кадр соответствует размеру логической таблицы, который задается полем `LChCnt` настроек модуля.

Ниже описаны некоторые особенности обработки данных с помощью `LTR12_ProcessData()`, которые специфичны для данного модуля.

3.3.1 Особенности калибровки данных модулем

Как и в случае с большинством других модулей АЦП, при производстве производится калибровка модуля LTR12 с записью его индивидуальных заводских калибровочных коэффициентов во flash-память модуля. Эти коэффициенты считываются в структуру в поле `Cbr` информации о модуле при успешном открытии соединения с модулем.

Применение коэффициентов происходит в `LTR12_ProcessData()` в том случае, если в функцию передан флаг `LTR12_PROC_FLAG_CALIBR`. Калибровка выполняется по формуле $Y = (X + Offset) * Scale$, где X — принятый код АЦП, Y — калиброванные данные, $Offset$ — коэффициент смещение нуля в кодах АЦП, а $Scale$ — коэффициент шкалы.

В результате применения калибровки получается калиброванные коды АЦП, для которых код `LTR12_ADC_SCALE_CODE_MAX` соответствует значению диапазона входной величины (`LTR12_ADC_RANGE_MA`). Этот код может быть переведен в результирующую величину в мА, если передан флаг `LTR12_PROC_FLAG_CONV_UNIT`.

В отличие от других модулей с коммутацией каналов в модуле LTR12 для каждого из 32 каналов АЦП используется свой индивидуальный коэффициент шкалы, т.к. в каждом канале установлен свой токовый шунт и коэффициент шкалы каждого канала учитывает в том числе и разброс сопротивления шунтов. В режиме измерения с общей землей (`LTR12_CH_MODE_COMM`) используется непосредственно коэффициент соответствующего канала. В разностном режиме (`LTR12_CH_MODE_DIFF`) точное определение коэффициента шкалы невозможно и используется среднее значение между коэффициентами шкалы соответствующих двух каналов, что обуславливает большее значение погрешности для данного режима измерения. Для режима измерения собственного нуля (`LTR12_CH_MODE_ZERO`) калибровочные коэффициенты и перевод в физические величины вообще не применяется, как описано в разделе [Особенности обработки измерений собственного нуля](#).

Коэффициент смещения нуля используется общий для всех каналов, так как это характеристика самого АЦП. Кроме того, вместо заводского значения этого коэффициента может использоваться явно измеренное значение смещения нуля из поля `AdcZeroOffsetCode`, если в `LTR12_ProcessData()` передан флаг `LTR12_PROC_FLAG_ZERO_OFFSETS_COR`.

3.3.2 Особенности обработки измерений собственного нуля

Модуль `LTR12` позволяет выполнять измерение значения собственно нуля АЦП. Это значение может использоваться для корректировки смещения нуля измерений, как описано в разделе [Корректировка смещение нуля](#). При этом величина смещения нуля измеряется исключительно в некалиброванных кодах АЦП и не может быть приведена к каким-либо физическим величинам, поэтому `LTR12_ProcessData()` к измерениям нуля не применяет калибровку или перевод в величины независимо от переданных флагов.

Существуют два варианта включения измерений собственного нуля в цикл опроса АЦП:

- Фоновые измерения собственного нуля. Если в настройках модуля задано ненулевое значение поля `BgLChCnt`, то после измерений, соответствующих основной логической таблице, модуль выполняет указанное количество измерений смещения нуля. Значения этих измерений не возвращаются в результирующем массиве на выходе `LTR12_ProcessData()`. Этот режим удобен для автокоррекции смещения нуля (с использованием флагов `LTR12_PROC_FLAG_ZERO_OFFSETS_COR` и `LTR12_PROC_FLAG_ZERO_OFFSETS_AUTORECALC`), когда сами измерения нуля не требуются пользователю и вся корректировка выполняется внутри `LTR12_ProcessData()`.
- Явно заданные измерения нуля. Соответствует случаю, когда в логической таблице (`LChTbl`) явно задан логический канал с режимом измерения `LTR12_CH_MODE_ZERO`. Данные от этого канала возвращаются вместе с остальными измерениями в массиве с результатами, но в отличие от других измерений они всегда возвращаются в кодах без применения калибровки, независимо от флагов, переданных в `LTR12_ProcessData()`. Данный режим как правило не требуется при штатной работе и может быть использован для получения всех измерений собственного нуля, например для анализа уровня шумов при измерении нуля самого АЦП и т.п.

В обоих случаях полученные коды смещения нуля могут использоваться для корректировки текущего значения измеренного значения смещения нуля, сохраняемого в поле `AdcZeroOffsetCode`, если в `LTR12_ProcessData()` передан флаг `LTR12_PROC_FLAG_ZERO_OFFSETS_AUTORECALC`.

3.4 Корректировка смещение нуля

Корректировка смещения нуля АЦП при обработки измерений для модуля `LT R12` может выполняться одним из следующих способов:

1. Использование заводского калибровочного значения смещения нуля. Этот режим не требует каких-либо дополнительных специальных измерений. Для его исполь-

зования достаточно передать флаг `LTR12_PROC_FLAG_CALIBR` при обработке данных с помощью `LTR12_ProcessData()`.

2. Однократное измерение смещения нуля перед стартом. В этом случае перед запуском сбора с помощью `LTR12_Start()` необходимо выполнить измерение нуля с помощью `LTR12_MeasAdcZeroOffset()`. После запуска измерений при обработке данных помимо флага `LTR12_PROC_FLAG_CALIBR` необходимо передать флаг `LTR12_PROC_FLAG_ZERO_OFFSETS_COR`, в результате чего вместо заводского калибровочного коэффициента будет использоваться измеренное во время вызов `LTR12_MeasAdcZeroOffset()` значение нуля. Этот способ позволяет компенсировать возможный уход нуля с момента заводской калибровки до начала измерений.
3. Автоматическая корректировка нуля во время сбора. В данном варианте помимо измерения собственного нуля перед стартом с помощью `LTR12_MeasAdcZeroOffset()` также выполняется постоянное измерение собственного смещения нуля непосредственно во время сбора данных. Для этого устанавливается ненулевое значение поля `BgLChCnt`, в результате чего после измерений, соответствующих логической таблице модуля, в каждом кадре будет выполнено дополнительно `BgLChCnt` измерений смещения нуля. При обработке данных в `LTR12_ProcessData()` помимо флагов `LTR12_PROC_FLAG_CALIBR` и `LTR12_PROC_FLAG_ZERO_OFFSETS_COR` необходимо передать флаг `LTR12_PROC_FLAG_ZERO_OFFSETS_AUTORECALC`, в результате чего функция будет использовать измерения собственного нуля для обновления расчетного значения смещения нуля, используемого для корректировки основных измерений. Эти дополнительные измерения нуля будут использоваться только внутри `LTR12_ProcessData()` и не будут включены в результат обработки. Этот способ позволяет учесть уход смещения нуля во время проведения длительного непрерывного эксперимента, однако он также требует дополнительного времени в цикле опроса АЦП во время сбора, что уменьшает максимально возможное значение частоты сбора на канал.

Глава 4

Константы, типы данных и функции библиотеки

4.1 Константы и перечисления

4.1.1 Константы и макроопределения

Константа	Значение	Описание
LTR12_CHANNELS_CNT	32	Максимальное число измерительных каналов в одном модуле
LTR12_DIFF_CHANNELS_CNT	16	Максимальное число измерительных разностных каналов в одном модуле
LTR12_RANGES_CNT	1	Количество диапазонов измерения
LTR12_ADC_RANGE_MA	20	Предел диапазона измерения модуля в мА
LTR12_R_IN_NOM	25	Номинальное значение входного сопротивления в Омах
LTR12_MAX_LCHANNELS_CNT	128	Максимальное количество каналов в логической таблице
LTR12_NAME_SIZE	8	Размер поля с названием модуля
LTR12_SERIAL_SIZE	16	Размер поля с серийным номером модуля
LTR12_FIRMDATE_SIZE	14	Размер поля с информацией о дате создания прошивки контроллера модуля
LTR12_ADC_SCALE_CODE_MAX	8000	Код АЦП, соответствующий номинальному значению диапазона
LTR12_IN_CLOCK_FREQ	15000000	Значение входной тактовой частоты контроллера модуля в Гц
LTR12_MAX_ADC_DIVIDER	65535	Максимальное допустимое значение поля делителя частоты АЦП
LTR12_MIN_ADC_DIVIDER	2	Минимальное допустимое значение поля делителя частоты АЦП
LTR12_MAX_ADC_FREQ	400000	Максимальное значение частоты АЦП в Гц
LTR12_FLASH_USERDATA_SIZE	2048	Размер пользовательской области flash-памяти

4.1.2 Коды ошибок, специфичные для модуля LTR12.

Тип: e_LTR12_ERROR_CODES		
Описание: Коды ошибок, которые определены и используются только в ltr12api. Общие коды ошибок, которые используются разными модулями, определены в ltrapi.h.		
Константа	Значение	Описание
LTR12_ERR_INVALID_MODULEINFO_RESP_FORMAT	-10900	Неверный формат ответов с информацией о модуле.
LTR12_ERR_INVALID_CFG_LCH_CNT	-10901	Задано неверное количество логических каналов в настройках модуля.
LTR12_ERR_INVALID_CFG_PHY_CH_NUM	-10902	Задан неверный номер физического канала в настройках модуля.
LTR12_ERR_INVALID_CFG_CH_MODE	-10903	Задан неверный режим измерения канала в настройках модуля.
LTR12_ERR_INVALID_CFG_ADC_PRESCALER	-10904	Задано неверное значение предварительного делителя частоты АЦП в настройках модуля.
LTR12_ERR_INVALID_CFG_ADC_DIVIDER	-10905	Задано неверное значение делителя частоты АЦП в настройках модуля.
LTR12_ERR_INVALID_CFG_ADC_RATE	-10906	Заданная частота АЦП находится вне допустимого диапазона.
LTR12_ERR_INVALID_CFG_CONV_START_SRC	-10907	Задан неверный источник запуска преобразования АЦП в настройках модуля.
LTR12_ERR_INVALID_CFG_ACQ_START_SRC	-10908	Задан неверный источник запуска сбора данных в настройках модуля.
LTR12_ERR_NO_END_OF_FRAME_RECVD	-10909	Не обнаружено действительного признака конца кадра в принятых данных.

4.1.3 Источник запуска сбора данных модулем

Тип: e_LTR12_ACQ_START_SRC		
Описание: Источник запуска сбора данных модулем		
Константа	Значение	Описание
LTR12_ACQ_START_SRC_INT	0	Внутренний запуск. Сбор запускается непосредственно по команде модулю при вызове LTR12_Start() .
LTR12_ACQ_START_SRC_EXT_RISE	1	Запуск сбора осуществляется по фронту внешнего сигнала на входе START сигнального разъема модуля.
LTR12_ACQ_START_SRC_EXT_FALL	2	Запуск сбора осуществляется по спаду внешнего сигнала на входе START сигнального разъема модуля.

4.1.4 Источник запуска преобразования АЦП

Тип: e_LTR12_CONV_START_SRC		
Описание: Источник запуска преобразования АЦП		
Константа	Значение	Описание
LTR12_CONV_START_SRC_INT	0	Запуск преобразований АЦП осуществляется от внутреннего периодического сигнала модуля, частота которого определяется настройками из AdcFreqParams .
LTR12_CONV_START_SRC_EXT_RISE	1	Запуск преобразований АЦП осуществляется по фронту внешнего сигнала на входе SYN сигнального разъема модуля
LTR12_CONV_START_SRC_EXT_FALL	2	Запуск преобразований АЦП осуществляется по спаду внешнего сигнала на входе SYN сигнального разъема модуля

4.1.5 Режим измерения логического канала

Тип: e_LTR12_CH_MODE		
Описание: Режим измерения логического канала		
Константа	Значение	Описание
LTR12_CH_MODE_COMM	0	Режим измерения с общей землей одного из 32 каналов модуля. Данный режим является основным рабочим режимом модуля LTR12.
LTR12_CH_MODE_DIFF	1	Разностный режим измерения. Выполняется измерение разности между выбранным одним из первых 16 каналов и соответствующим ему каналом из вторых 16.
LTR12_CH_MODE_ZERO	2	Режим измерения собственного нуля модуля.

4.1.6 Флаги, управляющие обработкой данных

Тип: e_LTR12_PROC_FLAGS		
Описание: Набор флагов, который может быть передан в функцию LTR12_ProcessData() для указания, какие преобразования и какую обработку данных требуется выполнить.		
Константа	Значение	Описание
LTR12_PROC_FLAG_CALIBR	0x00000001	Признак, что нужно выполнить калибровку, применив к данным калибровочные коэффициенты, как описано в разделе Особенности калибровки данных модулем .
LTR12_PROC_FLAG_CONV_UNIT	0x00000002	Признак, что нужно перевести коды АЦП в измеряемые величины (мА).
LTR12_PROC_FLAG_ZERO_OFFSETS_COR	0x00010000	Данный флаг указывает, что необходимо выполнять коррекцию нуля с учетом измеренного значения смещения нуля из AdcZeroOffsetCode . Данное значение может быть измерено перед запуском сбора данных с помощью LTR12_MeasAdcZeroOffset() , а также во время сбора данных в фоновом режиме. Если измерение собственного нуля не производилось, то использование данного флага не оказывает влияние на результат.

<p>LTR12_PROC_FLAG_ZERO_OFFS_ AUTORECALC</p>	<p>0x00020000</p>	<p>Данный флаг указывает, что необходимо использовать все обрабатываемые измерения собственного нуля АЦП для перерасчета текущего значения смещения нуля в поле AdcZeroOffsetCode, которое может быть использовано для коррекции нуля измерительных каналов при передаче флага LTR12_PROC_FLAG_ZERO_OFFS_COR.</p> <p>Данное действие относится как к фоновым каналам, задаваемым с помощью поля BgLChCnt, так и к каналам из логической таблицы с режимом измерения LTR12_CH_MODE_ZERO.</p> <p>Если в кадре отсутствуют измерения нуля, то передача данного флага не влияет на работу функции.</p>
<p>LTR12_PROC_FLAG_NONCONT_DATA</p>	<p>0x0100</p>	<p>По умолчанию LTR12_ProcessData() предполагает, что ей на обработку передаются все принятые данные и проверяет непрерывность счетчика в данных не только внутри переданного блока данных, но и между вызовами. Если обрабатываются не все данные или одни и те же данные обрабатываются повторно, то нужно указать данный флаг, чтобы счетчик проверялся только внутри обрабатываемого блока.</p>

4.2 Типы данных

4.2.1 Калибровочные коэффициенты модуля

Тип: TLTR12_CBR_INFO		
Описание: Структура содержит значения заводских калибровочных коэффициентов модуля, считанных из flash-памяти модуля.		
Поле	Тип	Описание поля
Offset	float	Код смещения нуля. Общий для всех каналов модуля.
ChScale	float [LTR12_CHANNELS_CNT]	Индивидуальный коэффициент шкалы для каждого канала модуля
Reserved	float [63]	Резерв

4.2.2 Информация о модуле

Тип: TLTR12_MODULE_INFO		
Описание: Структура, содержащая информацию о версиях прошивок контроллера модуля и информацию, считанную из flash-памяти модуля (серийный номер, калибровочные коэффициенты). Все поля заполняются при вызове LTR12_Open() , а также могут быть повторно вычитаны из flash-памяти при явном вызове LTR12_GetConfig() .		
Поле	Тип	Описание поля
Name	CHAR [LTR12_NAME_SIZE]	Название модуля. Оканчивающаяся нулем ASCII-строка ("LTR12").
Serial	CHAR [LTR12_SERIAL_SIZE]	Серийный номер модуля. Оканчивающаяся нулем ASCII-строка.
FirmwareVersion	WORD	Версия прошивки контроллера модуля (младший байт — минорная, старший - мажорная)
FirmwareDate	CHAR [LTR12_FIRMDATE_SIZE]	Дата создания прошивки контроллера. Оканчивающаяся нулем ASCII-строка.
Reserved	BYTE [8]	Резерв
Cbr	TLTR12_CBR_INFO	Заводские калибровочные коэффициенты, считанные из flash-памяти модуля.

4.2.3 Параметры текущего состояния модуля

Тип: TLTR12_STATE		
Описание: Структура, содержащая параметры модуля, которые пользователь может использовать только для чтения, так как они изменяются только внутри функций <code>ltr12api</code> .		
Поле	Тип	Описание поля
Configured	BOOLEAN	Признак, что была выполнена конфигурация модуля с помощью <code>LTR12_SetADC()</code> с момента установления соединения.
Run	BOOLEAN	Признак, запущен ли сбор данных. Изменяется при вызове функций <code>LTR12_Start()</code> и <code>LTR12_Stop()</code> .
FrameWordsCount	DWORD	Количество слов в одном кадре. В <code>LTR12_ProcessData()</code> нужно передавать количество слов, кратное этому значению. Обновляется после вызова <code>LTR12_SetADC()</code> . Определяется суммой количества каналов в логической таблице (<code>LChCnt</code>) и количеством фоновых измерений (<code>BgLChCnt</code>)
AdcFreq	double	Установленная частота АЦП в Гц. Обновляется после вызова <code>LTR12_SetADC()</code> .
FrameFreq	double	Установленная частота следования кадров в Гц. Соответствует частоте на каждый логический канал. Обновляется после вызова <code>LTR12_SetADC()</code> .
AdcZeroOffsetCode	double	Текущее измеренное значение смещения нуля АЦП в кодах. После установления соединения устанавливается равным калибровочному коэффициенту смещения нуля. Смещение может быть измерено явно до старта сбора данных с помощью функции <code>LTR12_MeasAdcZeroOffset()</code> , а также автоматически корректироваться во время сбора, если включены фоновые измерения нуля и в <code>LTR12_ProcessData()</code> передан флаг <code>LTR12_PROC_FLAG_ZERO_OFFS_AUTORECALC</code> . Применяется для корректировки нуля вместо калибровочного коэффициента при передаче флага <code>LTR12_PROC_FLAG_ZERO_OFFS_COR</code> в <code>LTR12_ProcessData()</code> .
Reserved	DWORD [16]	Резерв.

4.2.4 Настройки логического канала

Тип: TLTR12_LCHANNEL		
Описание: Структура определяет настройки одного опроса АЦП из последовательности опросов, определяемых логической таблицей.		
Поле	Тип	Описание поля
PhyChannel	BYTE	Номер физического канала (входа на разъеме), сигнал на котором должен быть измерен при данном опросе АЦП, начиная от 0.
Mode	BYTE	Используемый режим измерения. Значение из e_LTR12_CH_MODE .
Reserved	WORD	Резерв

4.2.5 Настройки частоты АЦП

Тип: TLTR12_ADCFREQ_CFG		
Описание: Структура содержит набор настроек, определяющих частоту преобразования АЦП в случае, если используется внутренний источник запуска преобразований АЦП (значение ConvStartSrc равно LTR12_CONV_START_SRC_INT). Значение частоты преобразования АЦП определяется по формуле:		
$F_{adc} = \frac{LTR12_IN_CLOCK_FREQ}{Prescaler * (Divider + 1)}$		
Частота 400 КГц является специальным случаем и задается явно с помощью установки поля Is400KHz , либо с помощью задания специальных значений Prescaler = 1, Divider = 36 (по аналогии с LTR11).		
Поля данной структуры могут быть заполнены автоматически с помощью функций LTR12_FillAdcFreqParam() или LTR12_FindAdcFreqParams() . Для расчета полученной частоты АЦП по заполненным параметрам можно использовать функцию LTR12_CalcAdcFreq() .		
Поле	Тип	Описание поля
Is400KHz	BOOLEAN	Признак использования частоты 400КГц. Если истина, то частота преобразования равна 400 КГц, а остальные параметры не имеют значения
Prescaler	WORD	Предварительный делитель тактовой частоты модуля. Должен быть равен одному из следующих значений: 1, 8, 64, 256 или 1024.
Divider	DWORD	Значение делителя тактовой частоты модуля. Может принимать значения в интервале от LTR12_MIN_ADC_DIVIDER до LTR12_MAX_ADC_DIVIDER включительно при условии, что полученная частота АЦП не превышает LTR12_MAX_ADC_FREQ .
Reserved	DWORD [6]	Резерв

4.2.6 Настройки модуля

Тип: TLTR12_CONFIG		
Описание: Структура содержит все настройки модуля, которые должен заполнить пользователь перед вызовом <code>LTR12_SetADC()</code> . Для заполнения параметров частоты АЦП и логических каналов могут использоваться вспомогательные функции.		
Поле	Тип	Описание поля
AdcFreqParams	TLTR12_ADCFREQ_CFG	Параметры, определяющие частоту преобразования АЦП при использовании внутреннего источника запуска преобразования. Если <code>ConvStartSrc</code> отлично от <code>LTR12_CONV_START_SRC_INT</code> , то данные настройки не имеют значения.
AcqStartSrc	DWORD	Источник запуска сбора данных. Одно из значений <code>e_LTR12_ACQ_START_SRC</code> .
ConvStartSrc	DWORD	Источник запуска каждого отдельного преобразования АЦП. Одно из значений <code>e_LTR12_CONV_START_SRC</code> .
Reserved	DWORD [30]	Резерв
BgLChCnt	DWORD	Количество фоновых измерений в одном кадре. Если отлично от нуля, то после опроса АЦП в соответствии с логической таблицей происходит указанное количество измерений собственного нуля, которые не будут возвращены в результатах обработки с помощью <code>LTR12_ProcessData()</code> , а могут только использоваться для корректировки смещения нуля модуля.
LChCnt	DWORD	Количество используемых логических каналов в таблице опроса АЦП <code>LChTbl</code> . Определяет число пользовательских измерений в одном кадре (цикле опроса каналов) АЦП. Сумма <code>LChCnt</code> и <code>BgLChCnt</code> не должна превышать <code>LTR12_MAX_LCHANNELS_CNT</code> .
LChTbl	TLTR12_LCHANNEL [LTR12_MAX_LCHANNELS_CNT]	Логическая таблица. Определяет, какие каналы модуля должны быть опрошены в каком порядке и с какими настройками.

4.2.7 Описатель модуля модуля

Тип: TLTR12		
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 LTR12_Init() .		
Поле	Тип	Описание поля
Size	INT	Размер структуры. Заполняется в LTR12_Init() .
Channel	TLTR	Структура, содержащая состояние клиентского соединения со службой ltrd. Не используется напрямую пользователем.
Internal	PVOID	Указатель на непрозрачную структуру с внутренними параметрами, используемыми исключительно библиотекой и недоступными для пользователя.
Cfg	TLTR12_CONFIG	Настройки модуля. Заполняются пользователем перед вызовом LTR12_SetADC() .
State	TLTR12_STATE	Состояние модуля и рассчитанные параметры. Поля изменяются функциями библиотеки. Пользовательской программой могут использоваться только для чтения.
ModuleInfo	TLTR12_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 LTR12_Init (TLTR12 *hnd)
Описание: Функция инициализирует поля структуры описателя модуля значениями по умолчанию. Эта функция должна вызываться для каждой структуры TLTR12 перед вызовом остальных функций.
Параметры: hnd — Описатель модуля
Возвращаемое значение: Код ошибки .

4.3.1.2 Открытие соединения с модулем

Формат: INT LTR12_Open (TLTR12 *hnd, DWORD ltrd_addr, WORD ltrd_port, const CHAR *csn, INT 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 LTR12_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 LTR12_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: <p>Код ошибки.</p>

4.3.1.3 Закрытие соединения с модулем

Формат: INT LTR12_Close (TLTR12 *hnd)
Описание: <p>Функция закрывает ранее открытое с помощью LTR12_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: <p>Код ошибки.</p>

4.3.1.4 Проверка, открыто ли соединение с модулем.

Формат: INT LTR12_IsOpened (TLTR12 *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 Запись настроек в модуль

Формат: INT LTR12_SetADC (TLTR12 *hnd)
Описание: Функция передает настройки, соответствующие значениям полей структуры конфигурации описателя модуля, в модуль. Должна вызываться хотя бы раз до первого запуска сбора, а также после любых изменений настроек модуля, чтобы они вступили в силу.
Параметры: hnd — Описатель модуля
Возвращаемое значение: Код ошибки.

4.3.2.2 Вычисление частоты АЦП

Формат: INT LTR12_CalcAdcFreq (const TLTR12_ADCFREQ_CFG *adcCfg, double *resultAdcFreq)
Описание: Вспомогательная функция, которая вычисляет значение частоты АЦП, соответствующее переданным параметрам в структуре конфигурации частоты TLTR12_ADCFREQ_CFG .
Параметры: adcCfg — Структура с параметрами, задающими частоту АЦП. resultAdcFreq — В данном параметре возвращается рассчитанное значение частоты АЦП в Гц, соответствующее переданным параметрам, если переданные параметры были корректны.
Возвращаемое значение: Код ошибки.

4.3.2.3 Подбор параметров частоты АЦП

Формат: <code>INT LTR12_FindAdcFreqParams (double adcFreq, TLTR12_ADCFREQ_CFG *adcCfg, double *resultAdcFreq)</code>
Описание: <p>Вспомогательная функция, которая по заданному значению частоты АЦП (суммарной на все каналы) подбирает значения задающих эту частоту параметров, чтобы результирующая частота была наиболее близка к заданной.</p> <p>Полученные параметры возвращаются в виде заполненной структуры TLTR12_ADCFREQ_CFG.</p> <p>Функция аналогична LTR12_FillAdcFreqParams(), но может использоваться без описателя модуля.</p>
Параметры: <p>adcFreq — Требуемая частота АЦП в Гц.</p> <p>adcCfg — Указатель на структуру, поля которой будут заполнены подобранными значениями параметров.</p> <p>resultAdcFreq — Указатель на переменную, в которой будет возвращено реально полученное значение частоты АЦП в Гц. Может быть передан нулевой указатель, если это значение не требуется.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.2.4 Заполнение параметров частоты АЦП

Формат: <code>INT LTR12_FillAdcFreqParams (TLTR12 *hnd, double adcFreq, double *resultAdcFreq)</code>
Описание: <p>Вспомогательная функция, которая по заданному значению частоты АЦП (суммарной на все каналы) заполняет соответствующие параметры конфигурации описателя модуля (структуру поля AdcFreqParams), чтобы результирующая частота была наиболее близка к заданной.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>adcFreq — Требуемая частота АЦП в Гц.</p> <p>resultAdcFreq — Указатель на переменную, в которой будет возвращено реально полученное значение частоты АЦП в Гц. Может быть передан нулевой указатель, если это значение не требуется.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.2.5 Заполнение параметров канала логической таблицы

Формат: INT LTR12_FillLChannel (TLTR12 *hnd, DWORD lchNum, BYTE phyChNum, BYTE mode)
Описание: Вспомогательная функция для заполнения логического канала таблицы опроса АЦП в настройках описателя модуля. Функция принимает номер логического канала и его параметры и заполняет соответствующий элемент массива LChTbl .
Параметры: hnd — Описатель модуля. lchNum — Номер логического канала (номер элемента таблицы LChTbl). Значение от 0 до LTR12_MAX_LCHANNELS_CNT - 1. phyChNum — Номер физического канала, который должен быть опрошен, считая от 0. mode — Режим измерения канала. Значение из e_LTR12_CH_MODE .
Возвращаемое значение: Код ошибки .

4.3.3 Функции для управления сбором данных

4.3.3.1 Измерение смещения нуля

Формат: INT LTR12_MeasAdcZeroOffset (TLTR12 *hnd, DWORD flags)
Описание: Функция выполняет измерение собственного нуля АЦП на заданной в конфигурации модуля частоте преобразования и сохраняет результирующее усредненное значение в поле AdcZeroOffsetCode структуры состояния State описателя модуля. Это измеренное значение может быть использовано для корректировки смещения нуля принятых данных при передаче флага LTR12_PROC_FLAG_ZERO_OFFS_COR в LTR12_ProcessData() , что позволяет учесть уход нуля АЦП на момент запуска измерений. Эта функция вызывается после конфигурации модуля непосредственно перед запуском сбора данных АЦП с помощью LTR12_Start() .
Параметры: hnd — Описатель модуля. flags — Флаги (резерв — должен передаваться 0)
Возвращаемое значение: Код ошибки .

4.3.3.2 Запуск сбора данных

Формат: INT LTR12_Start (TLTR12 *hnd)
Описание: <p>Данная функция переводит модуль в режим сбора данных с АЦП. Если используется запуск сбора по внутреннему сигналу (<code>AcqStartSrc</code> равно <code>LTR12_ACQ_START_SRC_INT</code>), то при вызове данной функции запускается непосредственно опрос АЦП и модуль начинает передавать полученные отсчеты, которые необходимо вычитывать с помощью <code>LTR12_Recv()</code>.</p> <p>В режиме запуска сбора от внешнего сигнала при вызове данной функции модуль переходит в режим ожидания указанного условия запуска, после чего уже будет запущен опрос АЦП и передача отсчетов.</p> <p>При завершении измерений для остановки сбора данных необходимо вызвать <code>LTR12_Stop()</code>.</p>
Параметры: <code>hnd</code> — Описатель модуля
Возвращаемое значение: Код ошибки.

4.3.3.3 Останов сбора данных

Формат: INT LTR12_Stop (TLTR12 *hnd)
Описание: <p>При вызове данной функции модуль останавливает сбор и выдачу данных.</p> <p>При этом вычитываются и отбрасываются все переданные, но непрочитанные данные от модуля.</p>
Параметры: <code>hnd</code> — Описатель модуля
Возвращаемое значение: Код ошибки.

4.3.3.4 Прием данных от модуля

Формат: INT LTR12_Recv (TLTR12 *hnd, DWORD *data, DWORD *tmark, DWORD size, DWORD timeout)
Описание: <p>Функция принимает запрашиваемое число слов от модуля. Возвращаемые слова находятся в специальном формате, который включает в себя служебную информацию. Возвращаемые слова соответствуют как основным пользовательским измерениям, так и фоновым измерениям нуля в соответствии с настройками модуля.</p> <p>Для обработки принятых слов и получения измеренных значений, полученные слова необходимо передать в функцию LTR12_ProcessData().</p> <p>Функция LTR12_Recv() возвращает управление либо когда примет запрошенное количество слов, либо по истечению таймаута. При этом реально принятое количество слов можно узнать по возвращенному значению.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>data — Массив, в который будут сохранены принятые слова. Должен быть размером на size 32-битных слов.</p> <p>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.</p> <p>size — Запрашиваемое количество 32-битных слов на прием.</p> <p>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.</p>
Возвращаемое значение: <p>Значение меньше нуля соответствует коду ошибки. Значение больше или равное нулю соответствует количеству реально принятых и сохраненных в массив data слов.</p>

4.3.3.5 Обработка принятых от модуля слов

Формат: INT LTR12_ProcessData (TLTR12 *hnd, const DWORD *src, double *dest, INT *size, DWORD flags)
Описание: <p>Функция используется для обработки слов, принятых от модуля с помощью LTR12_Recv(). Функция проверяет служебные поля принятых слов, извлекает полезную информацию с отсчетами и выполняет расчет результирующих измерений с применением при необходимости калибровочных коэффициентов и корректировки нуля.</p> <p>Функция возвращает результаты только для измерений, соответствующей пользовательской таблице логических каналов (LChTbl). Если включены фоновые измерения (BgLChCnt > 0), то соответствующие измерения могут быть использованы только для корректировки текущего значения нуля, а сами результаты по этим каналам не сохраняются в выходной массив.</p> <p>В функцию необходимо передавать слова, выравненные на начало кадра и в количестве, кратном количеству слов в кадре (можно узнать по значению поля FrameWordsCount).</p> <p>Функция изначально считает, что все данные, принятые LTR12_Recv(), полностью последовательно обрабатываются с помощью LTR12_ProcessData(), проверяя целостность счетчика и между вызовами функции. Если это не так, то необходимо передать флаг LTR12_PROC_FLAG_NONCONT_DATA.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>src — Указатель на массив, содержащий слова, принятые от модуля с помощью LTR12_Recv(), которые нужно обработать.</p> <p>dest — Указатель на массив, в который будут сохранены обработанные значения измерений, соответствующих логическим каналам таблицы LChTbl. Порядок следования соответствует порядку во входном массиве: сперва первый отсчет первого логического канала, затем первый отсчет второго логического канала и т.д.</p> <p>size — На входе принимает размер массива src для обработки. На выходе, при успешном завершении, возвращает количество сохраненных отсчетов в массиве dest.</p> <p>flags — Флаги из e_LTR12_PROC_FLAGS, управляющие работой функции. Может быть объединено несколько флагов через логическое ИЛИ.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.6 Сбор одного кадра

Формат: INT LTR12_GetFrame (TLTR12 *hnd, DWORD *data, DWORD timeout)
Описание: <p>Функция осуществляет запуск сбора одного кадра АЦП модулем с автоматической остановкой сбора после первого кадра и принимает все слова модуля, соответствующие данному кадру.</p> <p>Принятые слова должны быть далее обработаны с помощью LTR12_ProcessData().</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>data — Массив, в который будут сохранены принятые слова. Размер должен быть достаточен для сохранения слов одного кадра, который можно определить по полю FrameWordsCount.</p> <p>timeout — Таймаут на выполнение операции в миллисекундах. Если в течение заданного времени не будет принято всех слов кадра, то функция завершится с ошибкой.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.7 Поиск начала первого кадра

Формат: INT LTR12_SearchFirstFrame (TLTR12 *hnd, const DWORD *data, DWORD size, DWORD *index)
Описание: <p>Функция находит в принятом с помощью LTR12_Recv() массиве слов от модуля индекс первого слова, соответствующего началу кадра.</p> <p>Может использоваться для выравнивания на начало кадра данных в случае произошедшего сбоя без останова сбора.</p> <p>Если в переданном массиве начало кадра не найдено, то функция вернет ошибку LTR_ERROR_FIRSTFRAME_NOTFOUND.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>data — Указатель на массив, содержащий слова, принятые от модуля с помощью LTR12_Recv(), в которых ищется начало кадра.</p> <p>size — Количество слов в массиве data.</p> <p>index — В этой переменной возвращается индекс элемента, соответствующего началу первого кадра, если функция завершилась успешно.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.4 Функции для работы с flash-памятью модуля

4.3.4.1 Чтение данных из пользовательской области flash-памяти модуля

Формат: INT LTR12_FlashUserDataRead (LTR12 *hnd, DWORD addr, BYTE *data, DWORD size)
Описание: Функция вычитывает данные, записанные в пользовательскую область flash-памяти модуля, начиная с заданного адреса. Размер пользовательской области составляет <code>LTR12_FLASH_USERDATA_SIZE</code> байт.
Параметры: hnd — Описатель модуля. addr — Адрес относительно начала пользовательской области, начиная с которого необходимо прочитать данные. data — Массив на <code>size</code> 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 в пользовательскую область flash-память модуля

Формат: INT LTR12_FlashUserDataWrite (LTR12 *hnd, DWORD addr, const BYTE *data, DWORD size)
Описание: Функция записывает данные в пользовательскую область flash-память модуля, начиная с заданного адреса. При записи автоматически выполняется предварительное стирание данных, поэтому отдельная функция стирания содержимого flash-памяти не требуется.
Параметры: hnd — Описатель модуля. addr — Адрес относительно начала пользовательской области, начиная с которого необходимо выполнить запись. data — Массив из <code>size</code> байт с данными, которые будут записаны. size — Количество данных в байтах, которое необходимо записать.
Возвращаемое значение: Код ошибки.

4.3.5 Функции вспомогательного характера

4.3.5.1 Чтение информации и калибровочных коэффициентов

Формат: INT LTR12_GetConfig (TLTR12 *hnd)
Описание: Функция считывает информацию из flash-памяти модуля и обновляет поля <code>ModuleInfo</code> в управляющей структуре модуля. Так как данная операция уже выполняется при вызове <code>LTR12_Open()</code> , то как правило вызов данной функции не требуется. Однако данная функция может быть использована для восстановления информации в <code>ModuleInfo</code> .
Параметры: <code>hnd</code> — Описатель модуля.
Возвращаемое значение: Код ошибки.

4.3.5.2 Получение сообщения об ошибке

Формат: LPCSTR LTR12_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 <code>ltr12api</code> , так и общие коды ошибок из <code>ltrapi</code> .
Параметры: <code>err</code> — Код ошибки
Возвращаемое значение: Указатель на строку, содержащую сообщение об ошибке.