

МНОГОКАНАЛЬНЫЕ СИСТЕМЫ СБОРА ДАННЫХ

LTR25

РУКОВОДСТВО ПРОГРАММИСТА

*Ревизия 1.0.4
Октябрь 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	02.06.2014	Первая ревизия данного документа.
1.0.1	10.09.2014	Вопрос подключения библиотеки к проектам вынесен в отдельных документ.
1.0.2	17.01.2019	Добавлено описание функций для работы с TEDS. Описан механизм коррекции полярности сигнала, фазы, а также учета выходного сопротивления датчиков.
1.0.3	25.12.2019	Добавлено описание поля SupportedFeatures структуры с информацией о модуле и флага LTR25_FEATURE_EXT_BANDWIDTH_LF
1.0.4	29.10.2020	Добавлено описание функции LTR25_TEDSMemoryRead()

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 каналов АЦП	8
3.3	Прием и обработка данных	9
3.3.1	Проверка на обрыв или короткое замыкание	9
3.3.2	Особенности калибровки данных модулем	9
3.3.3	Инверсия входного сигнала	10
3.3.4	Учет выходного сопротивления датчика	10
3.3.5	Коррекция фазы	10
3.4	Работа с TEDS	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	Специфичные для модуля LTR25 коды ошибок	14
4.1.3	Коды частот сбора АЦП	15
4.1.4	Форматы данных от модуля	16
4.1.5	Значения источника тока	16
4.1.6	Флаги обработки данных	16
4.1.7	Состояние входного канала	17
4.1.8	Режим питания датчиков	17
4.1.9	Дополнительные возможности, поддерживаемые модулем	18
4.1.10	Коды семейств устройств узла TEDS.	18
4.1.11	Флаги для чтения данных из памяти узла TEDS.	19
4.2	Типы данных	19
4.2.1	Калибровочные коэффициенты	19
4.2.2	Набор коэффициентов для коррекции АЧХ модуля	19
4.2.3	Набор коэффициентов для коррекции ФЧХ модуля	20
4.2.4	Информация о модуле	20
4.2.5	Настройки канала АЦП	21
4.2.6	Настройки модуля	21
4.2.7	Параметры текущего состояния модуля	22
4.2.8	Управляющая структура модуля	22
4.2.9	Информация о устройстве узла TEDS.	23
4.3	Функции	24

4.3.1	Функции инициализации и работы с соединением с модулем	24
4.3.1.1	Инициализация описателя модуля	24
4.3.1.2	Открытие соединения с модулем	24
4.3.1.3	Закрытие соединения с модулем	25
4.3.1.4	Проверка, открыто ли соединение с модулем	25
4.3.2	Функции для изменения настроек модуля	25
4.3.2.1	Запись настроек в модуль.	25
4.3.3	Функции для управления сбором данных	26
4.3.3.1	Запуск сбора данных	26
4.3.3.2	Останов сбора данных	26
4.3.3.3	Прием данных от модуля	27
4.3.3.4	Обработка принятых от модуля слов	28
4.3.3.5	Поиск начала первого кадра	29
4.3.4	Функции для работы с памятью TEDS-датчиков	29
4.3.4.1	Проверка поддержки модуля работы с TEDS.	29
4.3.4.2	Установка режима питания датчиков	30
4.3.4.3	Обнаружение узла TEDS.	31
4.3.4.4	Чтение TEDS данных из энергонезависимой памяти дат- чика	32
4.3.4.5	Прямое чтение данных из энергонезависимой памяти TEDS-датчика	33
4.3.5	Функции для работы с flash-памятью модуля	34
4.3.5.1	Чтение данных из flash-памяти модуля	34
4.3.5.2	Запись данных во flash-память модуля	34
4.3.5.3	Стирание области flash-память модуля	35
4.3.6	Функции вспомогательного характера	35
4.3.6.1	Получение сообщения об ошибке	35
4.3.6.2	Чтение информации и калибровочных коэффициентов	35
4.3.6.3	Перевод модуля в режим низкого потребления	36
4.3.6.4	Проверка, разрешена ли работа ПЛИС модуля	36
4.3.6.5	Разрешение работы ПЛИС модуля	36

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 LTR25 с использованием предоставляемой фирмой “Л Кард” библиотеки `ltr25api`.

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 LTR25. Более подробно каждый шаг будет описан в последующих разделах.

Типичная последовательность действий имеет следующий вид:

1. Создать экземпляр структуры [TLTR25](#), представляющей собой описатель модуля. Описатель модуля содержит всю информацию о модуле и используется при вызове всех остальных функций.
2. Проинициализировать поля описателя с помощью [LTR25_Init\(\)](#)
3. Установить соединение с интересующим модулем с помощью функции [LTR25_Open\(\)](#).
4. При наличии датчиков с TEDS выполнить чтение информации из памяти датчиков как описано в главе [Работа с TEDS](#) (порядок по отношению к следующему шагу не важен, т.е. может выполняться как до, так и после записи настроек).
5. Заполнить необходимые поля с настройками модуля подструктуры [Cfg описателя модуля](#) и вызвать [LTR25_SetADC\(\)](#) для записи настроек в модуль.
6. Запустить сбор данных с помощью [LTR25_Start\(\)](#).
7. Прием и обработка данных, как описано ниже
8. По завершению работы выполнить останов сбора данных с помощью [LTR25_Stop\(\)](#).
9. Закрыть соединение с модулем, вызвав функцию [LTR25_Close\(\)](#).

Типичный цикл приема и обработки данных аналогичен большинству модулей АЦП системы LTR и выглядит следующим образом:

1. Прием заданного количества отсчетов с помощью [LTR25_Recv\(\)](#).
2. Обработка принятых отсчетов с помощью [LTR25_ProcessData\(\)](#).

3.2 Настройка модуля

Настройка модуля выполняется аналогично большинству других модулей LTR: в начале значения всех параметров модуля записываются в соответствующие поля [структуры описателя модуля](#), затем вызывается функция `LTR25_SetADC()`, которая значения этих полей передает модулю. Модуль должен быть настроен перед первым запуском сбора данных. Во время запущенного сбора данных изменение настроек не допускается.

Следует отметить, что все поля, относящиеся к настройке модуля, объединены в структуру типа `TLTR25_CONFIG` (поле [Cfg описателя модуля](#)). Только эти поля пользователь должен изменять вручную в [описателе модуля](#) при штатной работе и только эти поля влияют на параметры, записываемые `LTR25_SetADC()`.

При настройке модуля задаются следующие параметры:

- Частота сбора АЦП. Может быть выбрана одна из 8 predetermined частот сбора с помощью установки соответствующего кода в поле [FreqCode](#).
- Настройки каналов АЦП (раздел [Настройка каналов АЦП](#))
- Формат передаваемых данных задается полем [DataFmt](#). От формата зависит разрядность передаваемых отсчетов и количество передаваемых слов на один отсчет АЦП.
- Значение источника тока (10 или 2.86 мА) задается на все каналы сразу с помощью поля [ISrcValue](#).

После выполнения `LTR25_SetADC()` рассчитываются некоторые параметры, которые являются производными от настроек из `TLTR25_CONFIG`. Эти параметры записываются в соответствующие поля структуры состояния модуля типа `TLTR25_STATE` (поле [State описателя модуля](#)). Например, на основе установленного кода частоты в поле [AdcFreq](#) записывается соответствующая частота сбора АЦП в Гц.

3.2.1 Настройка каналов АЦП

Каждый модуль LTR25 имеет восемь каналов, которые выполняют преобразования параллельно. Можно разрешить прием данных по любому набору из этих восьми каналов. Однако следует учитывать, что максимальное количество одновременно разрешенных каналов зависит от настроенной частоты сбора и формата отсчетов (подробнее см. в документе [‘Крейтовая система LTR. Руководство пользователя’](#)).

Настройки каждого канала объединены в структуре `TLTR25_CHANNEL_CONFIG`. Массив структур из `LTR25_CHANNEL_CNT` элементов, каждый элемент которого соответствует нужному каналу, является полем `Ch` структуры с [настройками модуля](#).

Для каждого канала можно независимо задать:

- Разрешен ли сбор данных по этому каналу. Задается полем [Enabled](#).
- Сопротивление подключенного к каналу датчика, если необходима дополнительная коррекция данных, как описано в разделе [Учет выходного сопротивления датчика](#). Задается полем [SensorROut](#).

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 `LTR25_Recv()`, после чего эти данные передаются в `LTR25_ProcessData()` для проверки их корректности и перевода в Вольты. При этом следует учесть, что в зависимости от выбранного формата одному отсчету соответствует либо одно, либо два принятых слова.

Ниже описаны некоторые особенности обработки данных с помощью `LTR25_ProcessData()`, которые специфичны для данного модуля.

3.3.1 Проверка на обрыв или короткое замыкание

Модуль LTR25 позволяет при запущенном сборе определять ситуации обрыва или короткого замыкания для каждого разрешенного канала. При определении одной из этих ситуаций в передаваемом слове вместо самого отсчета АЦП вставляется специальный код, сигнализирующий о возникновении события. Функция `LTR25_ProcessData()` анализирует эти коды и формирует для каждого разрешенного канала статус канала. Статус рассчитывается по всему блоку (но отдельно для каждого канала), то есть если хотя бы для одного отсчета из обрабатываемого блока был установлен признак обрыва, то будет возвращен статус этого канала `LTR25_CH_STATUS_OPEN`. Порядок статусов соответствует порядку разрешенных каналов.

Следует иметь в виду, что определение состояний обрыва и КЗ — процесс инерционный, и установка статусов выполняется с задержкой относительно реального события. Подробнее см. в документе ‘Крейтовая система LTR. Руководство пользователя’.

3.3.2 Особенности калибровки данных модулем

В отличие от большинства остальных модулей LTR (например от LTR24), калибровка данных происходит аппаратно внутри модуля, а не программно. В связи с этим в `LTR25_ProcessData()` нет никаких указаний о выполнении калибровки. Самой библиотекой выполняется чтение калибровочных коэффициентов из Flash-памяти модуля, сохранение их в полях массива `CbrCoef` в структуре с информацией о модуле и запись этих коэффициентов в ПЛИС.

Следует отметить, что калибровочные коэффициенты определяются и сохраняются отдельно только для первых `LTR25_CBR_FREQ_CNT` частот сбора, так как остальные частоты получаются из частоты `LTR25_FREQ_39K` фильтрацией и прореживанием в ПЛИС и для них используются те же коэффициенты, что и для исходной частоты. Для каждого из восьми каналов используются свои калибровочные коэффициенты.

ПЛИС налету выполняет калибровку по формуле $Y = (X + Offset) * Gain$, где X — отсчет с АЦП расширенный до 32-бит, Y — калиброванные данные, $Offset$ — смещение шкалы (32-битный код), а $Gain$ — коэффициент шкалы. При этом перед выполнением калибровки, входное значение АЦП, расширяется до 32-бит. На выходе получаются уже откалиброванные 32-битные отсчеты. В случае 32-битного формата эти отсчеты и передаются в крейт, а в случае 20-битного передаются только старшие 20-бит.

При этом код `LTR25_ADC_SCALE_CODE_MAX` соответствует напряжению, равному максимальному напряжению для заданного диапазона.

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

вызовом `LTR25_SetADC()`, в которой и происходит загрузка коэффициентов в ПЛИС на основе установленной частоты АЦП.

3.3.3 Инверсия входного сигнала

Модуль LTR25 передает инвертированный (с обратным знаком) сигнал и изначально в этом виде он возвращался после вызова `LTR25_ProcessData()`. Для многих задач, где знак сигнала не имеет значения, это не является проблемой, однако в случаях, где это важно, следует учитывать данную особенность. Библиотека не корректирует полярность по умолчанию для сохранения совместимости с возможным клиентским ПО, которое могло бы учитывать эту особенность модуля. В новых версиях библиотеки для автоматической коррекции полярности сигнала был введен отдельный флаг `LTR25_PROC_FLAG_SIGN_COR`, при передаче которого в `LTR25_ProcessData()` будет автоматически скорректирована полярность сигнала.

3.3.4 Учет выходного сопротивления датчика

За счет взаимодействия выходного сопротивления ИСР-датчика и входного сопротивления ИСР-входа модуля образуется делитель, на котором происходит падение измеряемого напряжения с датчика.

Коэффициент передачи этого делителя составляет $K = \frac{R_{in}}{R_{in} + R_{out}}$, где R_{in} — входное сопротивление ИСР канала модуля LTR25, а R_{out} — выходное сопротивление подключенного ко входу датчика, которое также может учитывать при необходимости сопротивление проводов подключения датчика. Для модулей с расширенной полосой пропускания в области нижних частот (установлен флаг `LTR25_FEATURE_EXT_BANDWIDTH_LF` в информации о модуле) входное сопротивление составляет 110 кОм, в то время как для старых модулей это сопротивление составляет 31,6 кОм.

В большинстве случаев это падение значительно меньше точности используемого датчика и им можно пренебречь, но если используется датчик, точность которого и значение выходного сопротивления таковы, что указанное падение напряжения может видимым образом влиять на результат и если для подключаемого датчика явно указано выходное сопротивление, то его можно явно учесть средствами библиотеки. Для этого необходимо во время конфигурации модуля (до вызова `LTR25_SetADC()`) в структуре конфигурации соответствующего канала установить в поле `SensorROut` значение выходного сопротивления датчика. Если значение этого поля больше нуля, то функция `LTR25_ProcessData()` корректирует результат для компенсации указанного падения.

В случае подключения к ИСР-входу преобразователь `LPW25` способ учета его выходного сопротивления описан в его [Руководстве программиста](#).

3.3.5 Коррекция фазы

RC цепь ИСР-входа модуля LTR25 приводит к сдвигу фаз на малых частотах входного сигнала. Для задач, где важны фазы сигнала, этот сдвиг может быть неприемлем. В библиотеке реализован фазовый фильтр для коррекции этого сдвига, описанный в статье [‘Фазовый корректор дифференцирующей RC-цепи’](#).

По умолчанию данный фильтр не применяется для обеспечения совместимости со старыми программами, которые могли учитывать сдвиг фаз модуля самостоятельно. Для применения этого фильтра к выходным данным необходимо передать флаг `LTR25_PROC_FLAG_PHASE_COR`.

Для расчета фильтра по умолчанию используются коэффициенты, рассчитанные исходя из фазового сдвига при номинальных значениях RC-цепи. В случае, если была проведена калибровка фазового сдвига, то коэффициенты фильтра рассчитываются на основании записанного во Flash-память модуля смещения фазы, что позволяет дополнительно учесть разброс элементов (в первую очередь емкости) цепи соответствующего канала.

Важно отметить, что для задач, где важны фазовые соотношения между каналами, необходимо, чтобы версия прошивки ПЛИС была не ниже 9 (может быть обновлена клиентом) и версия библиотек не ниже 1.32.3.

3.4 Работа с TEDS

Начиная с ревизии 2, модуль LTR25 поддерживает цифровой режим работы канала при подключении [TEDS-датчиков](#) класса 1, который в частности может быть в использован для чтения информации о датчике (электронный паспорт датчика — TEDS).

Данные датчики помимо основного аналогового режима, совместимого с ICP-датчиками, имеют специальный цифровой режим, в котором по тем же линиям может выполняться взаимодействия со специальными устройствами на датчике, называемыми узлами (node). Как правило в качестве данного устройства используется энергонезависимая память, хранящая в себе информацию о датчике. Однако в более общем случае узел TEDS может реализовать и другие функциональные возможности (например настройка режимов датчика). Более того, стандартом допускается наличие более одного TEDS узла в датчике, подключенного к одному каналу.

Каждое узел TEDS имеет свой 64-битный уникальный регистрационный номер (URN), включающий в себя код семейства, которому принадлежит микросхема, серийный номер микросхемы (не датчика), а также CRC для проверки целостности предыдущих двух полей. Код семейства определяет, какую функциональность поддерживает узел, а также набор команд для работы с ним. Также в случае наличия нескольких узлов URN используется для адресации узла.

В памяти узла, помимо идентификатора производителя датчика, его модели и серийного номера, также может храниться информация и о характеристиках датчика, необходимых для обработки измерений, полученных с его выхода. В частности там может храниться информация для преобразования измеренного напряжения в физические величины, калибровочные коэффициенты датчика и т.д.

Так как формат данных TEDS в памяти определяется стандартом [IEEE 1451.4](#), то даже программа, не знающая точной модели подключенного датчика, может определить, каким образом следует преобразовать полученные данные и какие выходные физические величины будут получены, а также учесть калибровочные коэффициенты датчиков. Это справедливо, если эта информация закодирована с помощью стандартных шаблонов или шаблонов производителя, известных программе разбора данных TEDS (подробнее о формате данных TEDS описано в [Руководстве программиста библиотеки ltedsapi](#)).

Так как для чтения информации из памяти датчика используется специальный цифровой режим, то оно не может выполняться одновременно с получением аналоговых данных, т.е. необходимо прочитать эту информацию перед началом измерений.

Чтение информации выполняется следующим образом (при этом связь с модулем должна быть уже установлена, но сбор данных не запущен):

1. Перевести все подключенные к каналам модуля датчики в цифровой ре-

жим TEDS, вызвав функцию `LTR25_SetSensorsPowerMode()`, указав режим `LTR25_SENSORS_POWER_MODE_TEDS`

2. Для каждого интересующего канала:

- проверить наличие узла, вызвав функцию `LTR25_TEDSNodeDetect()`
- прочитать информацию о датчике из памяти узла с помощью функции `LTR25_TEDSReadData()`
- разобрать прочитанную информацию

3. По завершении чтения нужной информации перевести датчики модуля в штатный рабочий режим, вызвав функцию `LTR25_SetSensorsPowerMode()`, указав режим `LTR25_SENSORS_POWER_MODE_ICP`.

Следует отметить, что в памяти узла данные хранятся вместе с байтами контрольной суммы и при чтении данных из памяти происходит проверка контрольной суммы и байты с контрольной суммой отбрасываются из прочитанных данных. Таким образом, возвращаемый массив содержит только данные TEDS.

Для разбора прочитанной информации можно воспользоваться библиотекой `ltedsapi`, входящей в состав библиотек LTR, либо воспользоваться любым сторонним ПО разбора данных TEDS в соответствии со стандартом.

Если ко входу подключен преобразователь LPW25, то все функции разбора информации для данного преобразователя реализованы в библиотеке `lpw25api`.

Текущая реализация поддержки TEDS имеет некоторые ограничения, которые необходимо отметить:

1. Модуль поддерживает чтение данных по цифровому интерфейсу только для датчиков с одним узлом TEDS, что как правило справедливо. Тем не менее стандарт предусматривает подключение нескольких узлов к одной линии в цифровом режиме, как для хранения данных одного преобразователя на нескольких микросхемах памяти, так и для хранения данных по нескольким датчикам. В случае наличия нескольких микросхем на одной линии, функция `LTR25_TEDSNodeDetect()` вернет ошибку `LTR25_ERR_TEDS_NODE_URN_CRC` в связи с нарушением данных из-за конфликта при одновременной передаче URN разными узлами. Поддержка данного режима требует доработку обновляемой прошивки ПЛИС модуля LTR25 и функций данной библиотеки, что может быть рассмотрено при обращении клиентов.
2. Библиотека поддерживает чтение данных из наиболее часто используемых микросхем памяти для TEDS-датчиков, таких как DS2430A, DS2431, DS2433, DS28EC20. Тем не менее стандарт позволяет использовать в качестве узлов любые микросхемы памяти с поддержкой интерфейса “1-wire”, которые кроме того могут иметь дополнительные функции помимо памяти. В случае обнаружения не поддерживаемого устройства `LTR25_TEDSNodeDetect()` вернет ошибку `LTR25_ERR_TEDS_UNSUP_NODE_FAMILY`, но вернет действительный код семейства, которому принадлежит устройство. Поддержка других типов устройств или API для работы с иными функциями узлов TEDS помимо памяти может быть добавлена в данную библиотеку по запросу клиентов.

Глава 4

Константы, типы данных и функции библиотеки

4.1 Константы и перечисления

4.1.1 Константы и макроопределения

Константа	Значение	Описание
LTR25_CHANNEL_CNT	8	Количество каналов АЦП в одном модуле LTR25
LTR25_FREQ_CNT	8	Количество частот дискретизации.
LTR25_CBR_FREQ_CNT	2	Количество частот, для которых сохраняются калибровочные коэффициенты
LTR25_I_SRC_VALUE_CNT	2	Количество значений источника тока
LTR25_NAME_SIZE	8	Размер поля с названием модуля.
LTR25_SERIAL_SIZE	16	Размер поля с серийным номером модуля.
LTR25_ADC_RANGE_PEAK	10	Максимальное пиковое значение в Вольтах для диапазона измерения модуля
LTR25_ADC_SCALE_CODE_MAX	2000000000	Код АЦП, соответствующее максимальному пиковому значению
LTR25_FLASH_USERDATA_ADDR	0x0	Адрес, с которого начинается пользовательская область flash-памяти
LTR25_FLASH_USERDATA_SIZE	0x100000	Размер пользовательской области flash-памяти
LTR25_FLASH_ERASE_BLOCK_SIZE	4096	Минимальный размер блока для стирания flash-памяти. Все операции стирания должны быть кратны данному размеру
LTR25_TEDS_NODE_SERIAL_SIZE	6	Размер серийного номера узла TEDS в байтах

4.1.2 Специфичные для модуля LTR25 коды ошибок

Тип: e_LTR25_ERROR_CODES		
Описание: Коды ошибок, которые определены и используются только в ltr25api. Остальные коды ошибок, которые используются разными модулями, определены в ltrapi.h		
Константа	Значение	Описание
LTR25_ERR_FPGA_FIRM_TEMP_RANGE	-10600	Загружена прошивка ПЛИС для неверного температурного диапазона
LTR25_ERR_I2C_ACK_STATUS	-10601	Ошибка обмена при обращении к регистрам АЦП по интерфейсу I2C
LTR25_ERR_I2C_INVALID_RESP	-10602	Неверный ответ на команду при обращении к регистрам АЦП по интерфейсу I2C
LTR25_ERR_INVALID_FREQ_CODE	-10603	Неверно задан код частоты АЦП
LTR25_ERR_INVALID_DATA_FORMAT	-10604	Неверно задан формат данных АЦП
LTR25_ERR_INVALID_I_SRC_VALUE	-10605	Неверно задано значение источника тока
LTR25_ERR_CFG_UNSUP_CH_CNT	-10606	Для заданной частоты и формата не поддерживается заданное количество каналов АЦП
LTR25_ERR_NO_ENABLED_CH	-10607	Не был разрешен ни один канал АЦП
LTR25_ERR_ADC_PLL_NOT_LOCKED	-10608	Ошибка захвата PLL АЦП
LTR25_ERR_ADC_REG_CHECK	-10609	Ошибка проверки значения записанных регистров АЦП
LTR25_ERR_LOW_POW_MODE_NOT_CHANGED	-10610	Не удалось перевести АЦП из/в низкопотребляющее состояние
LTR25_ERR_LOW_POW_MODE	-10611	Модуль находится в низкопотребляющем режиме
LTR25_ERR_INVALID_SENSOR_POWER_MODE	-10612	Неверное значение режима питания датчиков
LTR25_ERR_CHANGE_SENSOR_POWER_MODE	-10613	Не удалось изменить режим питания датчиков
LTR25_ERR_INVALID_CHANNEL_NUMBER	-10614	Указан неверный номер канала модуля
LTR25_ERR_ICP_MODE_REQUIRED	-10615	Модуль не переведен в ICP-режим питания датчиков, необходимый для данной операции
LTR25_ERR_TEDS_MODE_REQUIRED	-10616	Модуль не переведен в TEDS режим питания датчиков, необходимый для данной операции

LTR25_ERR_TEDS_UNSUP_NODE_FAMILY	-10617	Данное семейство устройств TEDS узла не поддерживается библиотекой
LTR25_ERR_TEDS_UNSUP_NODE_OP	-10618	Данная операция не поддерживается библиотекой для обнаруженного типа узла TEDS
LTR25_ERR_TEDS_DATA_CRC	-10619	Неверное значение контрольной суммы в прочитанных данных TEDS
LTR25_ERR_TEDS_1W_NO_PRESENSE_PULSE	-10620	Не обнаружено сигнала присутствия TEDS узла на однопроводной шине
LTR25_ERR_TEDS_1W_NOT_IDLE	-10621	Однопроводная шина не была в незанятом состоянии на момент начала обмена
LTR25_ERR_TEDS_1W_UNKNOWN_ERR	-10622	Неизвестная ошибка при обмене по однопроводной шине с узлом TEDS
LTR25_ERR_TEDS_MEM_STATUS	-10623	Неверное состояние памяти TEDS узла
LTR25_ERR_TEDS_NODE_URN_CRC	-10624	Неверное значение контрольной суммы в URN узла TEDS
LTR25_ERR_TEDS_1W_BAD_OPCODE	-10625	Неверный код операции однопроводной шины
LTR25_ERR_TEDS_1W_PREV_CYCLE_ERR	-10626	Команда откинута из-за ошибки в предыдущем цикле обмена по однопроводной шине

4.1.3 Коды частот сбора АЦП

Тип: e_LTR25_FREQS		
Описание: Коды частот сбора АЦП		
Константа	Значение	Описание
LTR25_FREQ_78K	0	78.125 кГц
LTR25_FREQ_39K	1	39.0625 кГц
LTR25_FREQ_19K	2	19.53125 кГц
LTR25_FREQ_9K7	3	9.765625 кГц
LTR25_FREQ_4K8	4	4.8828125 кГц
LTR25_FREQ_2K4	5	2.44140625 кГц
LTR25_FREQ_1K2	6	1.220703125 кГц
LTR25_FREQ_610	7	610.3515625 Гц

4.1.4 Форматы данных от модуля

Тип: e_LTR25_FORMATS		
Описание: Форматы данных от модуля		
Константа	Значение	Описание
LTR25_FORMAT_20	0	20-битный целочисленный (1 слово на отсчет)
LTR25_FORMAT_32	1	32-битный целочисленный (2 слова на отсчет)

4.1.5 Значения источника тока

Тип: e_LTR25_I_SOURCES		
Описание: Значения источника тока		
Константа	Значение	Описание
LTR25_I_SRC_VALUE_2_86	0	2.86 мА.
LTR25_I_SRC_VALUE_10	1	10 мА.

4.1.6 Флаги обработки данных

Тип: e_LTR25_PROC_FLAGS		
Описание: Флаги, управляющие работой функции LTR25_ProcessData()		
Константа	Значение	Описание
LTR25_PROC_FLAG_VOLT	0x0001	Признак, что нужно перевести коды АЦП в Вольты. Если данный флаг не указан, то будут возвращены коды АЦП. При этом код LTR25_ADC_SCALE_CODE_MAX соответствует максимальному напряжению для установленного диапазона.
LTR25_PROC_FLAG_PHASE_COR	0x010	Признак, что необходимо выполнить коррекцию фазы модуля с помощью специального фильтра, как описано в разделе Коррекция фазы
LTR25_PROC_FLAG_SIGN_COR	0x080	При использовании этого флага будет исправлена инверсия сигнала модулем, как описано в разделе Инверсия входного сигнала .
LTR25_PROC_FLAG_NONCONT_DATA	0x0100	По умолчанию LTR25_ProcessData() предполагает, что ей на обработку передаются все принятые данные и проверяет непрерывность счетчика не только внутри переданного блока данных, но и между вызовами, а также использует состояние фильтров с прошлого вызова. Если обрабатываются не все данные или одни и те же данные обрабатываются повторно, то нужно указать данный флаг.

4.1.7 Состояние входного канала

Тип: e_LTR25_CH_STATUS		
Описание: Возвращается LTR25_ProcessData() для каждого разрешенного канала и определяет, был ли обнаружен обрыв или КЗ для данного канала в обрабатываемом с помощью LTR25_ProcessData() блоке		
Константа	Значение	Описание
LTR25_CH_STATUS_OK	0	Канал в рабочем состоянии
LTR25_CH_STATUS_SHORT	1	Было обнаружено короткое замыкание
LTR25_CH_STATUS_OPEN	2	Был обнаружен разрыв цепи

4.1.8 Режим питания датчиков

Тип: e_LTR25_SENSORS_POWER_MODE		
Описание: Режим питания всех датчиков для всех восьми каналов. Устанавливается функцией LTR25_SetSensorsPowerMode() .		
Константа	Значение	Описание
LTR25_SENSORS_POWER_MODE_ICP	0	Питание датчиков включено. Штатный режим работы, в котором могут проводиться измерения. Соответствуют аналоговому режиму работы TEDS датчиков, совместимому с рабочим режимом ICP-датчиков.
LTR25_SENSORS_POWER_MODE_OFF	1	В данном режиме питание датчиков отключено.
LTR25_SENSORS_POWER_MODE_TEDS	2	Подано питание обратной полярности. Это специальный цифровой режим работы для датчиков с поддержкой TEDS, который может использоваться в частности для чтения данных TEDS с информацией о датчике и его характеристиках

4.1.9 Дополнительные возможности, поддерживаемые модулем

Тип: e_LTR25_FEATURES		
Описание: Флаги, указывающие, какие дополнительные возможности реализованы в данной версии модуля.		
Константа	Значение	Описание
LTR25_FEATURE_EXT_BANDWIDTH_LF	(1 << 0)	Расширенная полоса пропускания в области низких частот. Относится к модулям LTR25, выпущенным в 2020 году и позже. Флаг указывает, что в RC-цепи, используемой для удаления постоянной составляющей сигнала, установлено сопротивление 110 кОм вместо 31.6 кОм, что приводит к расширению нижней границы полосы частот пропускания, а также к меньшему влиянию корректирующего фильтра фазы на АЧХ в области нижних частот и меньшей зависимости падения напряжения от выходного сопротивления датчика.

4.1.10 Коды семейств устройств узла TEDS.

Тип: e_LTR25_TEDS_NODE_FAMILY_CODE		
Описание: Данное перечисление содержит значения кодов семейств устройств с протоколом “1-Wire”, которые могут быть использованы при реализации узла TEDS как для хранения данных TEDS, так и для вспомогательных функций. Этот код определяет возможности, которые предоставляет устройство, и набор команд для работы с этим устройством. Данное определение содержит только коды семейств устройств, поддерживаемых библиотекой. По запросу этот список может быть увеличен. Код обнаруженного устройства узла TEDS возвращается в информации о устройстве при вызове <code>LTR25_TEDSNodeDetect()</code> .		
Константа	Значение	Описание
LTR25_TEDS_NODE_FAMILY_EEPROM_256_OTP	0x14	256 бит EEPROM с 64-битным регистром для одноразовой записи базовой информации (DS2430A)
LTR25_TEDS_NODE_FAMILY_EEPROM_4K	0x23	4 КБит EEPROM (DS2433)
LTR25_TEDS_NODE_FAMILY_EEPROM_1K	0x2D	1 КБит EEPROM (DS2431)
LTR25_TEDS_NODE_FAMILY_EEPROM_20K	0x43	20 КБит EEPROM (DS28EC20)

4.1.11 Флаги для чтения данных из памяти узла TEDS.

Тип: e_LTR25_TEDS_NONE_MEMRD_FLAGS		
Описание: Набор флагов, которые могут быть переданы в функцию LTR25_TEDSMemoryRead() для управления ее поведением		
Константа	Значение	Описание
LTR25_TEDS_NODE_MEMRD_FLAG_OTP	0x0001	Данный флаг указывает, что чтение данных должно быть выполнено из OTP блока памяти (также называемого Application Register), а не из основной области памяти. Имеет значения только для тех микросхем памяти, в которых присутствует OTP блок (LTR25_TEDS_NODE_FAMILY_EEPROM_256_OTP).

4.2 Типы данных

4.2.1 Калибровочные коэффициенты

Тип: TLTR25_CBR_COEF		
Описание: Структура, хранящая калибровочные коэффициенты для одного канала и диапазона.		
Поле	Тип	Описание поля
Offset	float	Код смещения
Scale	float	Коэффициент шкалы

4.2.2 Набор коэффициентов для коррекции АЧХ модуля

Тип: TLTR25_AFC_COEFS		
Описание:		
Поле	Тип	Описание поля
AfcFreq	double	Частота сигнала, для которой снято отношение амплитуд из FirCoef
FirCoef	double [LTR25_CHANNEL_CNT]	Набор отношений измеренной амплитуды синусоидального сигнала к реальной амплитуде для макс. частоты дискретизации и частоты сигнала из AfcFreq для каждого канала

4.2.3 Набор коэффициентов для коррекции ФЧХ модуля

Тип: TLTR25_PHASE_SHIFT_COEFS		
Описание:		
Поле	Тип	Описание поля
PhaseShiftRefFreq	double	Частота, на которой измерен сдвиг фаз каналов модуля
PhaseShift	double [LTR25_CHANNEL_CNT]	Сдвиг фазы для каждого канала модуля в градусах

4.2.4 Информация о модуле

Тип: TLTR25_MODULE_INFO		
Описание: Структура, содержащая информацию о версиях прошивок микросхем модуля и информацию, считанную из Flash-памяти модуля (серийный номер, калибровочные коэффициенты). Все поля заполняются при вызове LTR25_Open() , а также могут быть обновлены при явном вызове LTR25_GetConfig() .		
Поле	Тип	Описание поля
Name	CHAR [LTR25_NAME_SIZE]	Название модуля ("LTR25").
Serial	CHAR [LTR25_SERIAL_SIZE]	Серийный номер модуля.
VerFPGA	WORD	Версия прошивки ПЛИС
VerPLD	BYTE	Версия прошивки PLD
BoardRev	BYTE	Ревизия платы
Industrial	BOOL	Признак, это промышленный вариант модуля или нет
SupportedFeatures	DWORD	Флаги из e_LTR25_FEATURES , указывающие, какие дополнительные возможности поддерживает данная версия модуля
Reserved	DWORD [7]	Зарезервированные поля. Всегда равны 0
CbrCoef	TLTR25_CBR_COEF [LTR25_CHANNEL_CNT] [LTR25_CBR_FREQ_CNT]	Калибровочные коэффициенты модуля. Считываются из Flash-памяти модуля при вызове LTR25_Open() или LTR25_GetConfig() и загружаются в ПЛИС для применения во время вызова LTR25_SetADC()
AfcCoef	TLTR25_AFC_COEFS	Коэффициенты для коррекции АЧХ модуля
PhaseCoef	TLTR25_PHASE_SHIFT_COEFS	Коэффициенты для коррекции ФЧХ модуля
Reserved2	double [32 *LTR25_CHANNEL_CNT - 9 - 9]	Резервные поля

4.2.5 Настройки канала АЦП

Тип: TLTR25_CHANNEL_CONFIG		
Описание: Структура, содержащая настройки одного канала АЦП.		
Поле	Тип	Описание поля
Enabled	BOOL	Признак, разрешен ли сбор по данному каналу
SensorROut	float	Выходное сопротивление датчика, подключенного к данному входу ИСР, с возможным учетом сопротивления проводов подключения датчика, которое будет использовано для коррекции падения напряжения из-за его взаимодействия с входным сопротивлением ИСР-входа. Нулевое значение (по умолчанию) или значение меньше нуля указывает, что коррекция падения напряжения не требуется.
Reserved	DWORD [10]	Резервные поля (не должны изменяться пользователем)

4.2.6 Настройки модуля

Тип: TLTR25_CONFIG		
Описание: Структура содержит все настройки модуля, которые должен заполнить пользователь перед вызовом LTR25_SetADC() .		
Поле	Тип	Описание поля
Ch	TLTR25_CHANNEL_CONFIG [LTR25_CHANNEL_CNT]	Настройки каналов АЦП
FreqCode	BYTE	Код, задающий требуемую частоту сбора АЦП. Одно из значений e_LTR25_FREQS
DataFmt	BYTE	Формат, в котором будут передаваться отсчеты АЦП от модуля. Одно из значений e_LTR25_FORMATS . Формат определяет также количество передаваемых слов на один отсчет и влияет на максимальное количество разрешенных каналов
ISrcValue	BYTE	Используемое значение источника тока. Одно из значений e_LTR25_I_SOURCES
Reserved	DWORD [50]	Резервные поля (не должны изменяться пользователем)

4.2.7 Параметры текущего состояния модуля

Тип: TLTR25_STATE		
Описание: Структура, содержащая параметры модуля, которые пользователь должен использовать только для чтения, так как они изменяются только внутри функций ltr25api.		
Поле	Тип	Описание поля
FpgaState	BYTE	Текущее состояние ПЛИС. Одно из значений из e_LTR_FPGA_STATE
EnabledChCnt	BYTE	Количество разрешенных каналов. Устанавливается после вызова LTR25_SetADC()
Run	BOOL	Признак, запущен ли сбор данных
AdcFreq	double	Установленная частота АЦП. Обновляется после вызова LTR25_SetADC()
LowPowMode	BOOL	Признак, находится ли модуль в состоянии низкого потребления. В этом состоянии нельзя выполнять настройку АЦП или запуск сбора данных. Управление данным режимом выполняется с помощью LTR25_SetLowPowMode()
SensorsPowerMode	DWORD	Текущий режим питания датчиков для всех восьми каналов. Значение из e_LTR25_SENSORS_POWER_MODE. Управление данным режимом выполняется с помощью функции LTR25_SetSensorsPowerMode() .
Reserved	DWORD [30]	Резервные поля

4.2.8 Управляющая структура модуля

Тип: TLTR25		
Описание: Хранит текущие настройки модуля, информацию о его состоянии, структуру канала связи. Передается в большинство функций библиотеки. Некоторые поля структуры доступны для изменения пользователем для настройки параметров модуля. Перед использованием требует инициализации с помощью функции LTR25_Init() .		
Поле	Тип	Описание поля
Size	INT	Размер структуры. Заполняется при вызове LTR25_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	TLTR25_CONFIG	Настройки модуля. Заполняются пользователем перед вызовом LTR25_SetADC() .

State	TLTR25_STATE	Состояние модуля и рассчитанные параметры. Поля изменяются функциями библиотеки. Пользовательской программой могут использоваться только для чтения.
ModuleInfo	TLTR25_MODULE_INFO	Информация о модуле

4.2.9 Информация о устройстве узла TEDS.

Тип: TLTR25_TEDS_NODE_INFO		
Описание: В данной структуре содержится информация о микросхеме, которая используется для реализации узла TEDS и с которой взаимодействует модуль в цифровом режиме работы канала. Эта микросхема как правило содержит энергонезависимую память, которая хранит информацию о датчике (TEDS), но также может содержать и другие функциональные возможности. Данная информация возвращается при обнаружении устройства при вызове LTR25_TEDSNodeDetect() .		
Поле	Тип	Описание поля
Valid	BOOL	Признак действительности информации
DevFamilyCode	BYTE	Идентификатор семейства устройств, к которому принадлежит данное устройство, реализующее узел TEDS. Может быть равен одному из значений e_LTR25_TEDS_NODE_FAMILY_CODE или другому коду, если обнаружено устройство не из списка поддерживаемых библиотекой
DevSerial	BYTE [LTR25_TEDS_NODE_SERIAL_SIZE]	Серийный номер устройства узла TEDS. 48-битное число, представленное в виде 6 байт.
TEDSDataSize	DWORD	Размер TEDS данных в байтах, который может храниться в памяти в памяти узла.
MemorySize	DWORD	Размер памяти узла TEDS для прямого доступа.
Reserved	DWORD [6]	Резерв

4.3 Функции

4.3.1 Функции инициализации и работы с соединением с модулем

4.3.1.1 Инициализация описателя модуля

Формат: INT LTR25_Init (TLTR25 *hnd)
Описание: Функция инициализирует поля структуры описателя модуля значениями по умолчанию. Эта функция должна вызываться для каждой структуры TLTR25 перед вызовом остальных функций.
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.1.2 Открытие соединения с модулем

Формат: INT LTR25_Open (TLTR25 *hnd, DWORD ltrd_addr, WORD ltrd_port, const CHAR *csn, INT slot)
Описание: 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 LTR25_Close() .
Параметры: hnd — Описатель модуля. 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. ltrd_port — TCP-порт для подключения к службе ltrd. По умолчанию используется LTRD_PORT_DEFAULT. 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. slot — Номер слота крейта, в котором установлен интересующий модуль. Значение от LTR_CC_CHNUM_MODULE1 до LTR_CC_CHNUM_MODULE16.
Возвращаемое значение: Код ошибки .

4.3.1.3 Закрытие соединения с модулем

Формат: INT LTR25_Close (TLTR25 *hnd)
Описание: Функция закрывает ранее открытое с помощью LTR25_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.
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.1.4 Проверка, открыто ли соединение с модулем

Формат: INT LTR25_IsOpened (TLTR25 *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 LTR25_SetADC (TLTR25 *hnd)
Описание: Функция передает настройки, соответствующие значениям полей поля Cfg описателя модуля, в модуль. Должна вызываться перед первым запуском сбора данных с помощью LTR25_Start() .
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.3 Функции для управления сбором данных

4.3.3.1 Запуск сбора данных

Формат: INT LTR25_Start (TLTR25 *hnd)
Описание: При вызове данной функции запускается сбор данных с АЦП модуля. После успешного завершения этой функции запускается АЦП и модуль начинает передавать в ПК полученные отсчеты, которые необходимо вычитывать с помощью LTR25_Recv() . При завершении измерения для остановки сбора данных необходимо вызвать LTR25_Stop() . Хотя бы один из каналов АЦП должен быть до этого разрешен и модуль должен быть хотя бы раз сконфигурирован с помощью LTR25_SetADC() .
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.3.2 Останов сбора данных

Формат: INT LTR25_Stop (TLTR25 *hnd)
Описание: При вызове данной функции модуль останавливает сбор и выдачу данных АЦП. При этом вычитываются и выбрасываются все переданные, но непрочитанные данные от модуля.
Параметры: hnd — Описатель модуля.
Возвращаемое значение: Код ошибки .

4.3.3.3 Прием данных от модуля

Формат: INT LTR25_Recv (TLTR25 *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. Формат и количество слов на один отсчет определяется настройкой DataFmt.</p> <p>Для обработки принятых слов и получения значений АЦП используется функция LTR25_ProcessData().</p> <p>Функция 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>

4.3.3.4 Обработка принятых от модуля слов

Формат: INT LTR25_ProcessData (TLTR25 *hnd, const DWORD *src, double *dest, INT *size, DWORD flags, DWORD *ch_status)
Описание: <p>Функция используется для обработки слов, принятых от модуля с помощью LTR25_Recv(). Функция проверяет служебные поля принятых слов, извлекает полезную информацию с отсчетами и, при указании флага LTR25_PROC_FLAG_VOLT, переводит отсчеты в Вольты.</p> <p>Функция предполагает, что передаваемые слова выровнены на начало кадра (первое слово первого разрешенного канала). Если это не так, то неполный кадр в начале будет отброшен и функция вернет ошибку LTR_ERROR_PROCDATA_UNALIGNED.</p> <p>Также функция анализирует признаки обрыва или КЗ в разрешенных каналах. При наличии подобного признака для хотя бы одного отсчета соответствующего канала устанавливается соответствующий статус в элементе массива <code>ch_status</code>.</p> <p>В отличие от модуля LTR25 в модуле LTR25 калибровка отсчетов и корректировка АЧХ выполняется внутри модуля аппаратно, поэтому в LTR25_ProcessData() нет подобных флагов.</p> <p>По умолчанию функция предполагает, что все принятые данные обрабатываются и только один раз, проверяя непрерывность счетчика и между вызовами функции, а также сохраняя состояния фильтров между вызовами. Если это условие не выполняется, то нужно передать флаг LTR25_PROC_FLAG_NONCONT_DATA.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>src — Указатель на массив, содержащий слова, принятые от модуля с помощью LTR25_Recv(), которые нужно обработать.</p> <p>dest — Указатель на массив, в который будут сохранены обработанные данные. Порядок следования соответствует порядку во входном массиве (т.е. сперва первый отсчет первого разрешенного канала, затем первый отсчет второго канала и т.д.). Массив должен иметь достаточно места для сохранения как минимум <code>size</code> элементов для 20-битного режима или <code>size/2</code> для 32-битного.</p> <p>size — На входе принимает размер массива <code>src</code> для обработки. На выходе, при успешном завершении, возвращает количество сохраненных отсчетов в массиве <code>dest</code>.</p> <p>flags — Флаги из e_LTR25_PROC_FLAGS, управляющие работой функции. Может быть объединено несколько флагов через логическое ИЛИ.</p> <p>ch_status — Массив размером на количество элементов, соответствующее количеству разрешенных каналов. В каждом элементе сохраняется статус канала (одно из значений e_LTR25_CH_STATUS), определяющий, были ли признаки КЗ или обрыва в соответствующем канале. Может быть передан нулевой указатель, если эта информация не нужна.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.3.5 Поиск начала первого кадра

Формат: INT LTR25_SearchFirstFrame (TLTR25 *hnd, const DWORD *data, DWORD size, DWORD *index)
Описание: <p>Функция находит в принятом массиве сырых данных от модуля индекс первого слова первого начала кадра. 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 LTR25_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 памятью TEDS-датчиков

Данные функции доступны только для ревизии модуля 2 или выше и для прошивки ПЛИС версии 8 или выше.

4.3.4.1 Проверка поддержки модуля работы с TEDS.

Формат: INT LTR25_CheckSupportTEDS (TLTR25 *hnd)
Описание: <p>Данная функция служит для проверки, поддерживает ли данный экземпляр модуля LTR25 работу с TEDS-датчиками в цифровом режиме, что определяет возможность использования остальных функций из данной главы.</p> <p>Если поддерживает, то функция возвращает LTR_OK, иначе — код ошибки.</p>
Параметры: <p>hnd — Описатель модуля.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.4.2 Установка режима питания датчиков

Формат: INT LTR25_SetSensorsPowerMode (TLTR25 *hnd, DWORD mode)
Описание: <p>Функция устанавливает режим питания датчиков для всех каналов модуля. Основным рабочим режимом является LTR25_SENSORS_POWER_MODE_ICP (единственный доступный для ревизии модуля 0 или 1).</p> <p>Для доступа к энергонезависимой памяти TEDS-датчиков (или другим функциям узлов TEDS) необходимо сперва установить режим LTR25_SENSORS_POWER_MODE_TEDS, в котором на датчики подается питание противоположной полярности.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>mode — Устанавливаемый режим. Значение из e_LTR25_SENSORS_POWER_MODE.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.4.3 Обнаружение узла TEDS.

Формат: INT LTR25_TEDSNodeDetect (TLTR25 *hnd, INT ch, TLTR25_TEDS_NODE_INFO *devinfo)
Описание: <p>Функция выполняет определение наличия устройства (как правило памяти), реализующего узел TEDS, подключенного к указанному каналу модуля LTR25.</p> <p>Функция также считывает 64-битный URN (уникальный регистрационный номер) устройства, содержащий код семейства, которому принадлежит устройство, серийный номер и CRC.</p> <p>При успешном считывании и проверке CRC функция анализирует считанный код семейства устройств, и, если он принадлежит одному из поддерживаемых устройств, определяет по этому коду основные параметры устройства.</p> <p>Считанная и полученная по коду семейства информация, описывающая устройство, возвращается в виде структуры TLTR25_TEDS_NODE_INFO.</p> <p>Список поддерживаемых библиотекой семейств устройств определен в перечислении e_LTR25_TEDS_NODE_FAMILY_CODE.</p> <p>Если устройство обнаружено успешно, но тип не поддерживается библиотекой, то функция вернет ошибку LTR25_ERR_TEDS_UNSUP_NODE_FAMILY, но считанная часть информации о устройстве будет возвращена (идентификатор и серийный номер). Действительность возвращенной информации о памяти можно проверить по полю Valid.</p> <p>Для данной функции предварительно должен быть установлен режим питания датчиков LTR25_SENSORS_POWER_MODE_TEDS.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>ch — Номер канала модуля (от 0 до LTR25_CHANNEL_CNT-1).</p> <p>devinfo — Указатель на структуру, в которую будет возвращена информация об обнаруженном узле. Если информацию получить не удалось, то будет сброшено поле Valid структуры. Может быть передан нулевой указатель, если данная информация не требуется.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.4.4 Чтение TEDS данных из энергонезависимой памяти датчика

Формат: INT LTR25_TEDSReadData (TLTR25 *hnd, INT ch, BYTE *data, DWORD size, DWORD *read_size)
Описание: Функция выполняет чтение данных TEDS из энергонезависимой памяти узла TEDS. Функция выполняет чтение всех данных последовательно, как они хранятся в памяти в нужном порядке, проверяя и удаляя байты с контрольной суммой. Предварительно должен быть установлен режим питания датчиков LTR25_SENSORS_POWER_MODE_TEDS .
Параметры: hnd — Описатель модуля. ch — Номер канала модуля (от 0 до LTR25_CHANNEL_CNT-1). data — Буфер, в котором при успешном выполнении функции будут сохранены прочитанные данные. В нем должно быть достаточно места для сохранения size байт. size — Размер буфера для сохранения данных в байтах read_size — Размер сохраненных в буфер байт с данными TEDS
Возвращаемое значение: Код ошибки .

4.3.4.5 Прямое чтение данных из энергонезависимой памяти TEDS-датчика

Формат: INT LTR25_TEDSMemoryRead (TLTR25 *hnd, INT ch, DWORD addr, BYTE *data, DWORD size, DWORD flags)
Описание: <p>В отличие от LTR25_TEDSReadData() данная функция предназначена для чтения непосредственно содержимого энергонезависимой памяти узла TEDS без какой-либо их интерпретации, что может быть полезно, если датчик поддерживает интерфейс для чтения данных, совместимый с TEDS, но сам формат хранимых данных отличается от спецификации.</p> <p>Данная функция не выполняет проверку контрольной суммы блоков данных и не удаляет слова с контрольной суммой из результирующего массива. Также данная функция требует явно указать место чтения данных, а не полагается на стандартное их последовательное расположение. Место в памяти задается адресом, а также может зависеть от флагов для некоторых микросхем памяти.</p> <p>Предварительно должен быть установлен режим питания датчиков LTR25_SENSORS_POWER_MODE_TEDS.</p>
Параметры: <p>hnd — Описатель модуля.</p> <p>ch — Номер канала модуля (от 0 до LTR25_CHANNEL_CNT-1).</p> <p>addr — Адрес памяти, начиная с которого необходимо прочитать данные. Его интерпретация может зависеть от флагов.</p> <p>data — Буфер, в котором при успешном выполнении функции будут сохранены прочитанные данные. В нем должно быть достаточно места для сохранения size байт.</p> <p>size — Размер буфера для сохранения данных в байтах</p> <p>flags — Флаги из e_LTR25_TEDS_NONE_MEMRD_FLAGS, которые могут быть объединены битовому “ИЛИ”.</p>
Возвращаемое значение: <p>Код ошибки.</p>

4.3.5 Функции для работы с flash-памятью модуля

4.3.5.1 Чтение данных из flash-памяти модуля

Формат: INT LTR25_FlashRead (TLTR25 *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 LTR25_FLASH_USERDATA_ADDR размером LTR25_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 .

4.3.5.2 Запись данных во flash-память модуля

Формат: INT LTR25_FlashWrite (TLTR25 *hnd, DWORD addr, const BYTE *data, DWORD size)
Описание: Функция записывает данные во flash-память модуля по заданному адресу. Записываемая область должна быть предварительно стерта с помощью LTR25_FlashErase() . Пользователю выделяется область памяти с адреса LTR25_FLASH_USERDATA_ADDR размером LTR25_FLASH_USERDATA_SIZE байт.
Параметры: 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
Возвращаемое значение: Код ошибки .

4.3.5.3 Стирание области flash-память модуля

Формат: INT LTR25_FlashErase (TLTR25 *hnd, DWORD addr, DWORD size)
Описание: Функция стирает область во flash-памяти модуля по заданному адресу. Стирание необходимо выполнять перед записью данных. Стирание возможно только блоками, кратными LTR25_FLASH_ERASE_BLOCK_SIZE байт. Пользователю выделяется область памяти с адреса LTR25_FLASH_USERDATA_ADDR размером LTR25_FLASH_USERDATA_SIZE байт.
Параметры: hnd — Описатель модуля. addr — Адрес памяти, начиная с которого необходимо выполнить стирание size — Размер стираемой области в байтах. Должен быть кратен LTR25_FLASH_ERASE_BLOCK_SIZE .
Возвращаемое значение: Код ошибки .

4.3.6 Функции вспомогательного характера

4.3.6.1 Получение сообщения об ошибке

Формат: LPCSTR LTR25_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 ltr25api, так и общие коды ошибок из ltrapi.
Параметры: err — Код ошибки .
Возвращаемое значение: Указатель на строку, содержащую сообщение об ошибке.

4.3.6.2 Чтение информации и калибровочных коэффициентов

Формат: INT LTR25_GetConfig (TLTR25 *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 LTR25_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 .

4.3.6.3 Перевод модуля в режим низкого потребления

Формат: INT LTR25_SetLowPowMode (TLTR25 *hnd, BOOL lowPowMode)
Описание: Функция переводит модуль в режим низкого потребления или переводит его из этого режима снова в рабочий. В режиме низкого потребления снято питание с АЦП и источники тока установлены 2.86 мА. Доступ к регистрам АЦП невозможен. Этот режим может быть использован для полного сброса АЦП, для чего в этом режиме модуль должен находиться не менее 5 с.
Параметры: hnd — Описатель модуля. lowPowMode — Если FALSE — перевод модуля в рабочий режим, иначе — в режим низкого потребления.
Возвращаемое значение: Код ошибки.

4.3.6.4 Проверка, разрешена ли работа ПЛИС модуля

Формат: INT LTR25_FPGAIsEnabled (TLTR25 *hnd, BOOL *enabled)
Описание: Функция проверяет, разрешена ли работа ПЛИС модуля. Для настройки и сбора 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.

4.3.6.5 Разрешение работы ПЛИС модуля

Формат: INT LTR25_FPGAEnable (TLTR25 *hnd, BOOL enable)
Описание: Функция разрешает или запрещает работу ПЛИС модуля. Для настройки и сбора данных ПЛИС должен быть всегда разрешен. В LTR25_Open() выполняется разрешение работы ПЛИС, если была найдена прошивка ПЛИС в памяти модуля и она была успешно загружена, поэтому в штатной работе данная функция не используется.
Параметры: hnd — Описатель модуля. enable — Если FALSE — запрет работы ПЛИС, иначе — разрешение
Возвращаемое значение: Код ошибки.