

**Библиотека пользовательского интерфейса**  
**модуля LTR24**

**Крейтовая система LTR**

Руководство программиста

Автор: Ковалев О.А., Борисов А.В.

Последнюю версию руководства можно скачать на сайте ООО “Л-Кард”.

ООО “Л-Кард” оставляет за собой право обновлять документацию без уведомления пользователей об изменениях.

Контактная информация ООО “Л Кард”:

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

Тел.: (495) 785-95-25

Факс: (495) 785-95-14

Сайт: <http://www.lcard.ru>

FTP: <ftp://ftp.lcard.ru/pub>

E-mail:

- Отдел продаж: [sale@lcard.ru](mailto:sale@lcard.ru)
- Тех. поддержка: [support@lcard.ru](mailto:support@lcard.ru)
- Отдел кадров: [job@lcard.ru](mailto:job@lcard.ru)
- Общие вопросы: [lcard@lcard.ru](mailto:lcard@lcard.ru)

Представители в регионах:

- Украина: HOLIT Data Systems, <http://www.holit.com.ua>, (044) 241-6754
- Санкт-Петербург: Autex Spb Ltd., <http://www.autex.spb.ru>, (812) 567-7202
- Новосибирск: Сектор-Т, <http://www.sector-t.ru>, (383-2) 396-592
- Екатеринбург: Аск, <http://www.ask.ru>, 71-4444
- Казань: ООО “Шатл”, [shuttle@kai.ru](mailto:shuttle@kai.ru), (8432) 38-1600

© 2013, ООО “Л Кард”. Все права защищены.

История ревизий настоящего документа.

<b>Ревизия</b>	<b>Дата</b>	<b>Примечания по внесенным изменениям</b>
1.0.0	24.04.2013	Первая доступная для пользователя ревизия
1.0.1	3.09.2013	Добавлено описание возможности коррекции АЧХ модуля, новых режимов для LTR24-2, учтены изменения параметров LTR24_ProcessData() и полей структур.

# Оглавление

1. О чем этот документ.....	6
2. Общая информация .....	6
2.1. Форматы данных .....	6
2.2. Калибровка .....	7
3. Использование .....	8
3.1. Подключение к проекту .....	8
3.2. Работа с библиотекой.....	9
3.3. Настройка модуля.....	11
3.4. Коррекция АЧХ.....	11
4. Описание API.....	13
4.1. Константы .....	13
Коды частот дискретизации .....	13
Коды диапазонов в режиме дифференциальный вход .....	14
Коды диапазонов в режиме ICP-вход .....	14
Значения источника тока .....	14
Коды форматов .....	15
Флаги, управляющие обработкой данных.....	15
Коды ошибок .....	15
4.2. Типы и структуры данных .....	16
TLTR24_AFC_IIR_COEF.....	16
TLTR24_AFC_COEFS .....	16
TINFO_LTR24.....	17
TLTR24 .....	18
Формат данных. Устанавливается равным одной из констант «Коды диапазонов в режиме ICP-вход.....	19
Значения источника тока .....	19
4.3. Функции.....	21
LTR24_GetVersion .....	21
LTR24_GetErrorString.....	21
LTR24_Init .....	21

LTR24_Open .....	21
LTR24_Close .....	22
LTR24_IsOpened.....	22
LTR24_GetConfig .....	23
LTR24_SetADC .....	23
LTR24_Start .....	23
LTR24_Stop .....	24
LTR24_Recv.....	24
LTR24_RecvEx .....	26
LTR24_ProcessData .....	26
LTR24_SetZeroMode .....	28
LTR24_SetACMode .....	28
LTR24_StoreMcs.....	28
LTR24_RestoreMcs.....	29
LTR24_ClearMcsSlot .....	29
LTR24_InvalidateMcsSlot .....	30
LTR24_FindFrameStart.....	30
4.4. Форматы данных .....	31
Кадр.....	31
20-битный отсчет.....	31
24-битный отсчет.....	31

# 1. О чем этот документ...

Данный документ – это руководство программиста. Здесь рассматриваются вопросы прикладного программирования для модуля LTR24 с использованием библиотеки *ltr24api*. Здесь не затронуты вопросы, связанные с подключениями, принципом работы и архитектурой аппаратной части модуля LTR24. Информация по данной тематике доступна в документе «Крейтовая система LTR. Руководство пользователя».

## 2. Общая информация

Библиотека *ltr24api* является интерфейсом прикладного программирования модуля сбора данных LTR24 крейтовой системы LTR. Данный модуль с точки зрения программирования представляет собой 24-битный 4-канальный АЦП.

Основные особенности модуля LTR24:

- 4 канала, которые могут работать либо в режиме «Диф. вход», либо в режиме «ICP-вход»
- 2 диапазона ( $\pm 2\text{В}$  и  $\pm 10\text{В}$  в режиме «Диф. вход» или  $\sim 1\text{В}$  и  $\sim 5\text{В}$  в режиме «ICP-вход»)
- 16 частот дискретизации (от 610,352 Гц до 117,188 кГц)
- 2 формата данных (20- и 24-битный)
- Возможность включения одного из тестовых режимов – измерения собственного нуля или «ICP-тест»
- Режим отсечки постоянной составляющей по каждому каналу

Режимы «ICP-вход» и «ICP-тест» доступны только для модификации LTR24-2, которая имеет дополнительные входы для подключения ICP-датчиков. Программным образом можно проверить, доступны ли данные режимы с помощью поля `SupportICP` структуры с информацией о модуле (типа `TINFO_LTR24`) после открытия связи с модулем.

Модуль имеет определенные ограничения по настройке. В следующих разделах они описываются более подробно.

### 2.1. Форматы данных

Модуль позволяет работать в двух форматах данных: 20-битном и 24-битном. Эти форматы несколько отличаются по возможностям. В

Таблица 2-1 перечислены параметры, по которым они отличаются.

Таблица 2-1. Сравнение возможностей форматов данных

Параметр	Формат данных	
	20-битный	24-битный
Количество сырых данных на отсчет, 32-битное слово	1	2
Контроль непрерывности данных	Бит, устанавливающийся в 1 в каждом 15 слове	Счетчик по модулю 15

Максимальное количество включенных каналов	4	См. Таблица 2-2
Отслеживание перегрузки входного тракта	–	+

Использование 24-битного формата данных увеличивает точность, но в 2 раза увеличивает поток данных от модуля, что может стать проблемой при использовании большого количества модулей в одном крейте. Ограничение на максимальное количество каналов вызвано ограниченной пропускной способностью интерфейса с модулем. Модуль также позволяет отслеживать перегрузку входного тракта, что в определенных ситуациях приходится учитывать. Подробнее о перегрузке входного тракта см. «Крейтовая система LTR. Руководство пользователя».

Таблица 2-2. Макс. кол-во каналов при использовании 24-битного формата данных

Частота дискр., кГц	Макс. кол-во каналов
117,1875	2
78,125	3
58,59375 и ниже	4

## 2.2. Калибровка

Модуль поставляется откалиброванным на предприятии-изготовителе. Калибровочные коэффициенты хранятся в ПЗУ модуля и считываются из него с помощью функции `LTR24_GetConfig`. Считанные заводские коэффициенты сохраняются в поле `ModuleInfo` управляющей структуры модуля. Эти коэффициенты не должны меняться пользователем. Также копия коэффициентов сохраняются в поля `CalibCoef` и `AfcCoef` самой управляющей структуры, и именно они используются при указании соответствующих флагов. Таким образом, у пользователя есть возможность изменить заводские коэффициенты на собственные, не изменяя при этом информацию в `ModuleInfo`. Это может понадобиться, например, если необходимо откалибровать весь аналоговый тракт до модуля сбора данных.

Используется отдельные калибровочные коэффициенты для каждого канала, каждой частоты дискретизации и для каждого диапазона. При этом для калибровки используются два коэффициента: масштабный коэффициент (коэффициент шкалы) и смещение.

Также в ПЗУ модуля хранятся коэффициенты для расчета фильтров для коррекции АЧХ. Подробнее об этом описано в разделе «*Коррекция АЧХ*» данного документа.

Кроме того, для модуля LTR24-2 в ПЗУ модуля хранятся измеренные точные значения источников тока по каждому каналу для подключения ИСР-датчиков. Эти значения могут быть использованы для измерения внешних тензорезисторов.

## 3. Использование

### 3.1. Подключение к проекту

Для подключения библиотеки *ltr24api* к проекту на языке C/C++ необходимо выполнить следующее:

Для **ОС Windows**:

1. Должны быть установлены библиотеки *ltrdll.exe*.
2. Подключить заголовочный файл *ltr24api.h*:

```
#include <ltr/include/ltr24api.h>
```

3. Добавить в список каталогов для поиска директорию, относительно которой в “*ltr/include*” располагаются заголовки. При установке по умолчанию это путь “*C:\Program Files\L-Card*” или, для 64-битных систем, “*C:\Program Files (x86)L-Card*”.
4. Подключить библиотеку импорта *ltr24api.lib* для нужного компилятора  
- *Microsoft Visual C++* – из “*ltr\lib\msvc*”  
- *Borland C++/Borland C++ Builder* – из “*ltr\lib\borland*”
5. Для запуска собранной программы, необходимо, чтобы библиотека *ltr24api.dll* (а также *ltrapi.dll* и *ltrmcs.dll*, от которых она зависит) находились либо в директории с программой, либо в директории из переменной окружения PATH (установщик их устанавливает в “*%WINDIR%/system32*”).

Для **ОС Linux**:

1. Установить библиотеки, либо используя собранные пакеты, либо собрав самостоятельно (подробнее в документе *ltr\_cross\_sdk.pdf*)
2. Подключить заголовочный файл *ltr24api.h*:

```
#include <ltr/include/ltr24api.h>
```

3. Если директория “*ltr/include*” находится не по стандартному пути, то добавить соответствующий путь для поиска заголовочных файлов, например с помощью ключа *-I<путь>* при сборке GCC. При установке пакетов, заголовки ставятся в стандартную директорию “*/usr/include*” и задавать путь не нужно.

Подключить библиотеку *libltr24api.so* к проекту (например, с помощью ключа *-lltr24api* при сборке GCC). Если библиотеки находятся не по стандартному пути, то указать его с помощью ключа *-L<путь>*. При установке пакетов, библиотеки ставятся в */usr/lib* и указывать путь не нужно.

4. Для запуска необходимо наличие *libltr24api.so* и всех библиотек, от которых она зависит, в одной из стандартных директорий или директорий, заданных через переменную *LD\_LIBRARY\_PATH* или иным способом.



## 3.2. Работа с библиотекой

Управление модулем LTR24 осуществляется через управляющую структуру (TLTR24), отражающую текущее состояние модуля, канал связи и т.д. Одна структура TLTR24 используется для управления одним модулем.

При работе с библиотекой *ltr24api* необходимо соблюдать следующую обязательную последовательность действий:

1. Инициализация полей управляющей структуры (LTR24\_Init).
2. Открытие канала связи с модулем (LTR24\_Open).
3. Работа с модулем, вызов других функций.
4. Закрытие канала связи с модулем (LTR24\_Close).

Типичная последовательность вызовов выглядит следующим образом:

1. Инициализация полей управляющей структуры (LTR24\_Init).
2. Открытие канала связи с модулем (LTR24\_Open).
3. Получение информации из ПЗУ, включая калибровочные коэффициенты (LTR24\_GetConfig)
4. Заполнение полей управляющей структуры модуля, отвечающих за конфигурацию модуля
5. Запись установленной конфигурации в модуль (LTR24\_SetADC).
6. Запуск сбора данных (LTR24\_Start)
7. Прием блока данных с помощью LTR24\_Recv
8. Обработка принятого блока данных с помощью
9. LTR24\_ProcessData
10. Если требуется принять еще данные, то переход к пункту 7, иначе к пункту 10.
11. Останов сбора данных (LTR24\_Stop).
12. Закрытие канала связи с модулем (LTR24\_Close).

После открытия канала связи модуль может находиться в двух режимах: конфигурирование и сбор данных. В режиме конфигурирования выполняется настройка параметров сбора данных, модуль работает в режиме “запрос/ответ”. В режиме сбора данных происходит потоковая передача данных со всех включенных каналов АЦП.

В режиме конфигурирования выполняется считывание информации из ПЗУ модуля (LTR24\_GetConfig), настройка параметров модуля через заполнение полей управляющей структуры, запись параметров в модуль (LTR24\_SetADC). По окончании настройки модуля выполняется запуск сбора данных с помощью функции LTR24\_Start, после чего модуль переходит в режим сбора данных.

В режиме сбора данных можно настраивать только режим отсечки постоянной составляющей по каждому каналу отдельно (LTR24\_SetACMode) и режим измерения собственного нуля для всех каналов одновременно (

LTR24\_SetZeroMode). В этом режиме параметры устанавливаются только с помощью специальных функций, а не через поля управляющей структуры. Основное назначение этого режима – прием сырых данных с модуля и их обработка (функции LTR24\_Recv,

LTR24\_RecvEx,

LTR24\_ProcessData). Данные принимаются кадрами (см. Кадр). Возвращение в режим конфигурирования выполняется после остановки сбора данных (LTR24\_Stop).

### 3.3. Настройка модуля

Для настройки модуля необходимо заполнить поля управляющей структуры, отвечающие за параметры модуля, после чего вызвать LTR24\_SetADC. Настройки, которые задаются для всего модуля или для всех каналов одновременно, представлены полями самой управляющей структуры, а настройки, которые задаются индивидуально для каждого канала, – полями массива структур ChannelMode. Сбор данных при настройке не должен быть запущен.

Могут быть настроены следующие параметры:

- Частота дискретизации АЦП (поле ADCFreqCode)
- Режим работы для каждого канала (см. таблицу 3-1)
- Выбор, по каким каналам разрешен прием данных (поле Enable из ChannelMode)
- Диапазон для каждого канала (поле Range из ChannelMode)
- Значение источника тока для ICP-входов – задается сразу для всех входов (поле ISrcValue)
- Формат данных (поле DataFmt)

Таблица 3-1. Определение режима для каждого канала

TestMode	ICPMode	AC	Режим
FALSE	FALSE	FALSE	Дифференциальный вход без отсечки постоянной составляющей
FALSE	FALSE	TRUE	Дифференциальный вход с отсечкой постоянной составляющей
FALSE	TRUE	X	Режим ICP-вход
TRUE	FALSE	X	Режим измерения собственного нуля
TRUE	TRUE	X	Режим «ICP-тест»

### 3.4. Коррекция АЧХ

Библиотечные функции позволяют выполнять коррекцию АЧХ модуля с помощью дополнительных фильтров. Для всех диапазонов выполняется коррекция завала АЧХ входного тракта модуля с помощью КИХ фильтра, как описано в статье “Метод тонкой коррекции наклона АЧХ с помощью простого цифрового фильтра”. В ПЗУ модуля сохраняется значение отношения измеренной модулем амплитуды сигнала заданной частоты при максимальной частоте дискретизации АЦ, к реально поданной амплитуде сигнала, а также значение частоты сигнала в Гц.

Кроме того, для частот АЦП 39,0625 КГц и ниже выполняется дополнительная коррекция АЧХ самого АЦП с помощью БИХ фильтра 2-го порядка, коэффициенты которого также хранятся в ПЗУ модуля.

Для выполнения коррекции АЧХ необходимо передать флаг `LTR24_PROC_FLAG_AFC_COR` в функцию

`LTR24_ProcessData` при обработке данных (коэффициенты до этого уже должны быть прочитаны с помощью `LTR24_GetConfig`). При этом, по умолчанию предполагается, что все принятые данные передаются в

`LTR24_ProcessData` подряд без разрывов и повторов, и фильтры не сбрасываются между вызовами

`LTR24_ProcessData`. Если это не так, и очередная порция данных для обработки идет не сразу за предыдущей, то следует указать это с помощью флага `LTR24_PROC_FLAG_NONCONT_DATA`. Естественно, фильтры всегда сбрасываются при запуске сбора с помощью `LTR24_Start`.

## 4. Описание API

### 4.1. Константы

LTR24_VERSION_CODE	0x02000000UL
Текущая версия библиотеки (2.0.0.0).	
LTR24_CHANNEL_NUM	4
Количество каналов.	
LTR24_RANGE_NUM	2
Количество диапазонов в режиме дифференциального входа.	
LTR24_ICP_RANGE_NUM	2
Количество диапазонов в режиме ICP-входа.	
LTR24_FREQ_NUM	16
Количество частот дискретизации.	
LTR24_I_SRC_VALUE_NUM	16
Количество значений источника тока.	
LTR24_NAME_SIZE	8
Размер поля имени.	
LTR24_SERIAL_SIZE	16
Размер поля серийного номера.	

#### Коды частот дискретизации

LTR24_FREQ_117K	0
117,1875 кГц	
LTR24_FREQ_78K	1
78,125 кГц	
LTR24_FREQ_58K	2
58,59375 кГц	
LTR24_FREQ_39K	3
39,0625 кГц	
LTR24_FREQ_29K	4
29,296875 кГц	
LTR24_FREQ_19K	5
19,53125 кГц	
LTR24_FREQ_14K	6
14,6484375 кГц	
LTR24_FREQ_9K7	7
9,765625 кГц	

LTR24_FREQ_7K3	8
7,32421875 кГц	
LTR24_FREQ_4K8	9
4,8828125 кГц	
LTR24_FREQ_3K6	10
3,662109375 кГц	
LTR24_FREQ_2K4	11
2,44140625 кГц	
LTR24_FREQ_1K8	12
1,8310546875 кГц	
LTR24_FREQ_1K2	13
1,220703125 кГц	
LTR24_FREQ_915	14
915.52734375 Гц	
LTR24_FREQ_610	15
610.3515625 Гц	

### Коды диапазонов в режиме дифференциальный вход

LTR24_RANGE_2	0
Диапазон $\pm 2$ В.	
LTR24_RANGE_10	1
Диапазон $\pm 10$ В.	

### Коды диапазонов в режиме ICP-вход

LTR24_ICP_RANGE_1	0
Диапазон $\sim 1$ В.	
LTR24_ICP_RANGE_5	1
Диапазон $\sim 5$ В.	

### Значения источника тока

LTR24_I_SRC_VALUE_2_86	0
2.86 мА.	
LTR24_I_SRC_VALUE_10	1
10 мА.	

## Коды форматов

LTR24_FORMAT_20	0
20-битный формат данных.	
LTR24_FORMAT_24	1
24-битный формат данных.	

## Флаги, управляющие обработкой данных

LTR24_PROC_FLAG_CALIBR	0x00000001
Признак, что нужно применить к данным калибровочные коэффициенты.	
LTR24_PROC_FLAG_VOLT	0x00000002
Признак, что нужно перевести коды АЦП в Вольты.	
LTR24_PROC_FLAG_AFC_COR	0x00000004
Признак, что необходимо выполнить коррекцию АЧХ.	
LTR24_PROC_FLAG_NONCONT_DATA	0x00000100
Признак, что идет обработка не непрерывных данных.	

## Коды ошибок

LTR24_ERR_INVALID_FREQ	-10100
Задана некорректная частота дискретизации.	
LTR24_ERR_INVALID_FORMAT	-10101
Задан некорректный формат данных.	
LTR24_ERR_CFG_UNSUP_CH_CNT	-10102
Для заданной частоты и 24-битного формата не поддерживается установленное количество каналов.	
LTR24_ERR_INVALID_RANGE	-10103
Некорректный диапазон канала.	
LTR24_ERR_WRONG_CRC	-10104
Неверная контрольная сумма EEPROM.	
LTR24_ERR_VERIFY_FAILED	-10105
Ошибка верификации записи в EEPROM.	
LTR24_ERR_DATA_FORMAT	-10106
Неверный формат данных в обработанных отсчетах.	
LTR24_ERR_UNALIGNED_DATA	-10107
Невыровненные данные.	
LTR24_ERR_DISCONT_DATA	-10108
Сбой счетчика данных в обработанных отсчетах.	
LTR24_ERR_CHANNELS_DISBL	-10109

Ни один канал не был разрешен.

LTR24_ERR_UNSUP_VERS	-10110	Версия формата управляющей структуры не поддерживается.
LTR24_ERR_FRAME_NOT_FOUND	-10111	Начало кадра не найдено.
LTR24_ERR_OPEN_MCS_MOD	-10112	Не удалось открыть модуль для работы с сохранением контекста.
LTR24_ERR_NO_SAVED_MCS	-10113	Нет сохраненного контекста.
LTR24_ERR_MCS_NOT_VALID	-10114	Сохраненный контекст недействителен.
LTR24_ERR_MCS_DIFF_MID	-10115	Сохраненный контекст принадлежит другому модулю.
LTR24_ERR_UNSUP_FLASH_FMT	-10116	Неподдерживаемый формат данных во Flash-памяти модуля.
LTR24_ERR_INVALID_I_SRC_VALUE	-10117	Задано некорректное значение источника тока.
LTR24_ERR_UNSUP_ICP_MODE	-10118	Данная модификация модуля не поддерживает ICP-режим.

## 4.2. Типы и структуры данных

### TLTR24\_AFC\_IIR\_COEF

```
typedef struct {  
    double a0;  
    double a1;  
    double b0;  
} TLTR24_AFC_IIR_COEF;
```

Коэффициенты БИХ-фильтра для коррекции АЧХ

### TLTR24\_AFC\_COEFS

```
typedef struct {  
    double AfcFreq;  
    double FirCoef[LTR24_CHANNEL_NUM][LTR24_RANGE_NUM];  
    TLTR24_AFC_IIR_COEF AfcIirCoef;  
} TLTR24_AFC_COEFS;
```

Набор коэффициентов для коррекции АЧХ модуля.

AfcFreq

Частота сигнала, для которой измерено отношение амплитуд и сохранено в FirCoef



FirCoef

Набор отношений измеренной амплитуды синусоидального сигнала к реальной амплитуде для макс. частоты дискретизации и частоты сигнала из AfcFreq для каждого канала и каждого диапазона

AfcIirCoef

Коэффициенты БИХ-фильтра для коррекции АЧХ АЦП на частотах дискретизации 39,0625 КГц и ниже

## TINFO\_LTR24

```
typedef struct {
    CHAR          Name[LTR24_NAME_SIZE];
    CHAR          Serial[LTR24_SERIAL_SIZE];
    BYTE          VerPLD;
    BOOL          SupportICP;
    DWORD         Reserved[8];
    struct {
        float     Offset;
        float     Scale;
    } CalibCoef[LTR24_CHANNEL_NUM][LTR24_RANGE_NUM][LTR24_FREQ_NUM];
    TLTR24_AFC_COEFS AfcCoef;
    double ISrcVals[LTR24_CHANNEL_NUM][LTR24_I_SRC_VALUE_NUM];
} TINFO_LTR24;
```

Содержит информацию о модуле. Вся информация, кроме значений полей SupportICP и VerPLD, берется из ПЗУ модуля и действительна только после вызова LTR24\_GetConfig.

Name

Название модуля (“LTR24”).

Serial

Серийный номер модуля.

VerPLD

Версия прошивки ПЛИС.

SupportICP

Признак, поддерживает ли модуль режим измерения с ICP-датчиков. Для LTR24-2 это поле равно TRUE, для остальных модификаций – FALSE.

Reserved

Зарезервированные поля. Всегда равны 0.

CalibCoef

Заводские калибровочные коэффициенты по каждому каналу, диапазону и частоте.

Offset

Смещение.

Scale

Коэффициент масштаба.

AfcCoef

Коэффициенты для корректировки АЧХ.

ISrcVals

Измеренные значения источников токов для каждого канала (только для LTR24-2).

## TLTR24

```
struct TLTR24 {
    INT      Size;
    TLTR     Channel;
    BOOL     Run;
    BYTE     ADCFreqCode;
    double   ADCFreq;
    BYTE     DataFmt;
    BYTE     ISrcValue;
    BOOL     TestMode;
    DWORD    Reserved[16];
    struct {
        BOOL     Enable;
        BYTE     Range;
        BOOL     AC;
        BOOL     ICPMode;
        DWORD    Reserved[4];
    } ChannelMode[];
    TINFO_LTR24 ModuleInfo;
    struct {
        float    Offset;
        float    Scale;
    } CalibCoef[LTR24_CHANNEL_NUM][LTR24_RANGE_NUM][LTR24_FREQ_NUM];
    TLTR24_AFC_COEFS AfcCoef;
    PVOID     Internal;
};
```

Управляющая структура модуля. 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 LTR24\_Init.

Size

Размер структуры TLTR24. Заполняется автоматически при вызове функции LTR24\_Init.

Channel

Канал связи с LTR сервером.

Run

Текущее состояние сбора данных (TRUE – сбор данных запущен).

ADCFreqCode

Код частоты дискретизации. Устанавливается равным одной из констант «Коды частот дискретизации». **Устанавливается пользователем.**

ADCFreq

Значение частоты дискретизации в Гц. Заполняется значением частоты дискретизации, соответствующим коду в поле ADCFreqCode, после выполнения функции LTR24\_SetADC.

DataFmt

**Формат данных. Устанавливается равным одной из констант «Коды диапазонов в режиме ICP-вход**

LTR24\_ICP\_RANGE\_1 0

Диапазон ~1 В.

LTR24\_ICP\_RANGE\_5 1

Диапазон ~5 В.

**Значения источника тока**

LTR24\_I\_SRC\_VALUE\_2\_86 0

2.86 мА.

LTR24\_I\_SRC\_VALUE\_10 1

10 мА.

Коды форматов». **Устанавливается пользователем.**

ISrcValue

Значение источника тока для всех каналов подключения ICP-датчиков. Устанавливается равным одной из констант «Значения источника тока». Имеет значение только для LTR24-2. **Устанавливается пользователем.**

TestMode

Включение тестовых режимов ("Измерение нуля" или "ICP-тест" в зависимости от значения поля ICPMode для каждого канала) для всех каналов (TRUE – включен). **Устанавливается пользователем.**

Reserved

Резерв. Поле не должно изменяться пользователем.

ChannelMode

Режимы каналов. **Все поля устанавливается пользователем.**

Enable

Включение канала. Если равно TRUE, то модулем будут передаваться слова, соответствующие отсчетом с данного канала, если FALSE – нет.

Range

Диапазон канала. Устанавливается равным одной из констант «Коды диапазонов в режиме дифференциальный вход» или «Коды диапазонов в режиме ICP-вход» в зависимости от значения поля ICPMode .

AC

Режим отсечки постоянной составляющей (TRUE – включен). Имеет значение только в случае, если поле ICPMode равно FALSE.

ICPMode

Включение режима измерения ICP-датчиков. Если FALSE – используется режим "Диф. вход" или "Измерение нуля" (в зависимости от поля TestMode), если TRUE – режим "ICP вход" или "ICP тест".

Reserved

Резерв. Поле не должно изменяться пользователем.

ModuleInfo

Информация о модуле.

CalibCoef

Применяемые для коррекции данных в функции

LTR24\_ProcessData калибровочные коэффициенты по каждому каналу, диапазону и частоте. При вызове LTR24\_GetConfig в данные поля копируются заводские калибровочные коэффициенты (те же, что и в ModuleInfo). Но, при необходимости, пользователь может записать сюда свои коэффициенты.

Offset

Смещение.

Scale

Коэффициент масштаба.

AfcCoef

Коэффициенты для корректировки АЧХ, применяемые в функции

LTR24\_ProcessData. При вызове LTR24\_GetConfig в данные поля копируются значения из ПЗУ модуля (те же, что и в ModuleInfo).

Internal

Указатель на структуру с параметрами, используемыми только библиотекой и недоступными пользователю.

## 4.3. Функции

### LTR24\_GetVersion

```
DWORD LTR24_GetVersion (void);
```

Используется для определения совместимости программного обеспечения и текущей версии библиотеки по номеру ее версии. Номер версии библиотеки, с которым компилировалась программа, доступен через константу `LTR24_VERSION_CODE`.

Возвращает:

Текущий номер версии библиотеки *ltr24api*.

### LTR24\_GetErrorString

```
LPCSTR LTR24_GetErrorString (INT error);
```

Возвращает текстовое описание ошибки по ее коду. Текстовое описание является строкой, оканчивающейся нулевым символом. Кодировка описания – WINDOWS-1251 для ОС Windows или UTF-8 для ОС Linux.

```
error [in]
```

Код ошибки.

Возвращает:

Текстовое описание кода ошибки.

### LTR24\_Init

```
INT LTR24_Init (TLTR24 *ltr24);
```

Инициализирует поля управляющей структуры модуля. Перед использованием управляющей структуры в других функциях ее необходимо проинициализировать.

```
ltr24 [in]
```

Управляющая структура модуля.

Возвращает:

`LTR_OK`.

### LTR24\_Open

```
INT LTR24_Open (TLTR24 *ltr24,  
                DWORD ip_addr,  
                WORD port,  
                const CHAR *serial,  
                INT slot  
                );
```

Открывает канал связи с модулем. Соединение осуществляется через LTR сервер, запущенный на хосте с IP адресом `addr` и слушающий TCP порт `port`. Конкретный модуль выбирается серийным номером крейта `serial` и номером слота в крейте `slot`.

Если `serial` равен `NULL` или пустой строке (`""`), то выбирается первый крейт в списке LTR сервера. В качестве IP адреса и номера порта могут использоваться соответственно константы `SADDR_DEFAULT` и `SPORT_DEFAULT`, задавая значения по умолчанию (127.0.0.1:11111). Порядок байтов в IP адресе: `1.2.3.4 > 0x01020304`.

После завершения работы с модулем необходимо закрыть канал связи с помощью функции `LTR24_Close`.

`ltr24` [in]

Управляющая структура модуля.

`ip_addr` [in]

IP адрес хоста, на котором запущен LTR сервер.

`port` [in]

Порт, который прослушивается LTR сервером.

`serial` [in]

Серийный номер крейта.

`slot` [in]

Номер слота в крейте. Слоты нумеруются с 1.

Возвращает:

`LTR_OK` или код ошибки.

## LTR24\_Close

```
INT LTR24_Close (TLTR24 *ltr24);
```

Закрывает канал связи с модулем. После завершения работы с модулем необходимо закрыть канал связи с ним.

`ltr24` [in]

Управляющая структура модуля.

Возвращает:

`LTR24_OK` или код ошибки.

## LTR24\_IsOpened

```
INT LTR24_IsOpened (TLTR24 *ltr24);
```

Проверяет, открыт ли канал связи с модулем.

`ltr24` [in]

Управляющая структура модуля.

Возвращает:

LTR\_OK, если модуль открыт, или код ошибки.

## LTR24\_GetConfig

```
INT          LTR24_GetConfig          (TLTR24          *ltr24);
```

Считывает информацию из ПЗУ модуля, обновляет структуру `ModuleInfo`, `CalibCoef` и `AfcCoef` управляющей структуры модуля.

```
ltr24          [in]
```

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_SetADC

```
INT          LTR24_SetADC          (TLTR24          *ltr24);
```

Конфигурирует модуль в соответствии с выбранными настройками. Настройка выполняется путем заполнения полей управляющей структуры, предназначенных для изменения пользователем.

```
ltr24          [in]
```

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_Start

```
INT          LTR24_Start          (TLTR24          *ltr24);
```

Запускает сбор данных с модуля. Перед запуском сбора данных модуль должен быть сконфигурирован с помощью функции `LTR24_SetADC`. Конфигурирование модуля во время сбора данных недоступно, за исключением режима измерения собственного нуля и режима отсечки постоянной составляющей. Изменение этих параметров в режиме сбора данных выполняются с помощью функций `LTR24_SetZeroMode` и `LTR24_SetACMode`.

```
ltr24          [in]
```

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_Stop

```
INT          LTR24_Stop          (TLTR24          *ltr24);
```

Останавливает сбор данных с модуля. После остановки сбора данных модуль может быть переконфигурирован.

```
ltr24          [in]
```

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_Recv

```
INT          LTR24_Recv          (TLTR24          *ltr24,  
                                DWORD          *data,  
                                DWORD          *tmark,  
                                DWORD          size,  
                                DWORD          timeout,  
                                );
```

Принимает сырые данные от модуля и складывает их в массив `data`. Функция возвращает управление либо когда будет принято запрашиваемое число 32-битных слов, либо когда истечет указанный в параметре `timeout` интервал времени. В 24-битном формате каждому отсчету АЦП соответствует два слова данных, а в 20-битном – одно.

Порядок следования слов: сперва первый отсчет первого разрешенного канала, затем первый отсчет со второго канала, ... первый отсчет с n-го канала., потом вторые отсчеты по каждому разрешенному каналу потом и т.д. Отсчеты приходят только для тех каналов, по которым разрешен сбор данных.

В массив `tmark` складываются значения меток СЕКУНДА и СТАРТ. Каждому элементу массива `data` ставится в соответствие элемент `tmark`. При отсутствии необходимости приема секундных меток параметр `tmark` устанавливается равным `NULL`.

Количество запрашиваемых данных измеряется в 32-битных словах.

Принятые сырые данные передаются функции `LTR24_ProcessData` для коррекции и перевода в физические величины.

```
ltr24          [in]
```

Управляющая структура модуля.

```
data          [out]
```

Массив для записи принятых данных.

```
tmark         [out]
```

Массив для записи меток СЕКУНДА и СТАРТ.

```
size          [in]
```



Количество запрашиваемых данных.

timeout [in]

Таймаут ожидания данных в мс.

Возвращает:

Количество принятых слов данных ( $\geq 0$ ) или код ошибки ( $< 0$ ).

## LTR24\_RecvEx

```
INT          LTR24_RecvEx          (TLTR24      *ltr24,  
                                   DWORD           *data,  
                                   DWORD           *tmark,  
                                   DWORD           size,  
                                   DWORD           timeout,  
                                   LARGE_INTEGER *time  
                                   );
```

Принимает сырые данные от модуля и складывает их в массив `data`. Функция аналогична `LTR24_Recv`, только дополнительно записывает по каждому слову данных абсолютное время приема, измеренное по часам в крейт-контроллере. Время имеет формат POSIX, 64 бита. При отсутствии необходимости приема абсолютных меток времени параметр `time` устанавливается равным `NULL`, либо используется функция `LTR24_Recv`.

`ltr24` [in]

Управляющая структура модуля.

`data` [out]

Массив для записи принятых данных.

`tmark` [out]

Массив для записи меток СЕКУНДА и СТАРТ.

`size` [in]

Количество запрашиваемых данных.

`timeout` [in]

Таймаут ожидания данных в мс.

`time` [out]

Массив для записи абсолютного времени приема.

## LTR24\_ProcessData

```
INT          LTR24_ProcessData    (TLTR24      *ltr24,  
                                   const DWORD   *input,  
                                   double         *output,  
                                   INT            *size,  
                                   DWORD          flags,  
                                   BOOL           *overload  
                                   );
```

Выполняет преобразование сырых данных, применяет калибровочные коэффициенты, проверяет отсутствие разрывов в данных. Сырые данные должны передаваться выровненными по границе кадра, и содержать целое число кадров (см. *Кадр*). В случае передачи невыровненных кадров функция вырезает неполные кадры и возвращает ошибку. О наличии разрывов функция сообщает ошибкой.

По умолчанию функция предполагает, что все принятые данные от конкретного модуля обрабатываются функцией `LTR24_ProcessData` и обрабатываются один раз (т.е. блок данных, переданный в функцию, соответствует данным сразу за обработанными до этого данными). Если это не так, то нужно указать это с помощью флага `LTR24_PROC_FLAG_NONCONT_DATA`.

Выходные данные возвращаются либо в кодах АЦП, либо в Вольтах (если указан флаг `LTR24_PROC_FLAG_VOLT`).

Если указан флаг `LTR24_PROC_FLAG_CALIBR`, то применяются калибровочные коэффициенты из массива `CalibCoef` управляющей структуры модуля.

Функция также может выполнять коррекцию АЧХ модуля с использованием коэффициентов из поля `AfcCoef` управляющей структуры модуля. Для этого необходимо передать флаг `LTR24_PROC_FLAG_AFC_COR`.

При работе в 24-битном формате отсчет состоит из двух слов сырых данных, поэтому количество элементов в выходных массивах должно быть в 2 раза меньше. Массив `overload` должен содержать столько же элементов, сколько в массиве `output`.

`ltr24` [in]

Управляющая структура модуля.

`input` [in]

Массив сырых данных.

`output` [out]

Массив для записи обработанных данных.

`size` [in, out]

Количество сырых данных. После выполнения – количество данных в выходном массиве.

`flags` [in]

Набор флагов из «Флаги, управляющие обработкой данных». Может быть передано несколько флагов, объединенных через логическое «ИЛИ».

`overload` [out]

Массив для записи информации о перегрузке входов (данный признак отслеживается только в 24-битном формате данных).

Возвращает:

`LTR_OK` или код ошибки.

## LTR24\_SetZeroMode

```
INT          LTR24_SetZeroMode      (TLTR24      *ltr24,  
                                     BOOL          enable  
                                     );
```

Изменяет состояние режима измерения собственного нуля по всем каналам. Данная функция используется только во время сбора данных. Для настройки в режиме конфигурирования используется поле `TestMode` управляющей структуры модуля.

`ltr24` [in]

Управляющая структура модуля.

`enable` [in]

Состояние режима измерения собственного нуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_SetACMode

```
INT          LTR24_SetACMode        (TLTR24      *ltr24,  
                                     BYTE          chan,  
                                     BOOL          enable  
                                     );
```

Изменяет состояние режима отсечки постоянной составляющей для выбранного канала. Данная функция используется только во время сбора данных. Для настройки в режиме конфигурирования используется поле `AC` для каждого канала в управляющей структуре модуля.

`ltr24` [in]

Управляющая структура модуля.

`chan` [in]

Номер канала.

`enable` [in]

Состояние режима отсечки постоянной составляющей.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_StoreMcs

```
INT          LTR24_StoreMcs         (TLTR24      *ltr24);
```

Сохраняет управляющую структуру модуля в крейт-контроллере. Это позволяет в случае обрыва связи без необходимости остановки сбора данных восстановить весь контекст.

Данная возможность доступна только для крейт-контроллеров с расширением MCS (только в крейт-контроллерах LTR032).

ltr24 [in]

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_RestoreMcs

```
INT          LTR24_RestoreMcs      (TLTR24      *ltr24,  
                                   DWORD          ip_addr,  
                                   WORD           port,  
                                   const CHAR    *serial,  
                                   BYTE          slot  
                                   );
```

Восстанавливает управляющую структуру модуля из крейт-контроллера. Функция аналогична LTR24\_Open, за исключением того, что пытается без сброса модуля восстановить сохраненный контекст. Канал связи с модулем должен быть закрыт.

Данная возможность доступна только для крейт-контроллеров с расширением MCS (только в крейт-контроллерах LTR032).

ltr24 [in]

Управляющая структура модуля.

ip\_addr [in]

IP адрес хоста, на котором запущен LTR сервер.

port [in]

TCP порт, который прослушивается LTR сервером.

serial [in]

Серийный номер крейта.

slot [in]

Номер слота в крейте. Слоты нумеруются с 1.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_ClearMcsSlot

```
INT          LTR24_ClearMcsSlot    (TLTR24      *ltr24);
```

Удаляет сохраненные данные об управляющей структуре.

Данная возможность доступна только для крейт-контроллеров с расширением MCS (только в крейт-контроллерах LTR032).

ltr24 [in]

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_InvalidateMcsSlot

```
INT          LTR24_InvalidateMcsSlot (TLTR24          *ltr24);
```

Делает сохраненные данные в слоте недействительными. Используется для того, чтобы во время переконфигурирования модуля не возникла ситуация, когда сохраненные данные и действительное состояние модуля отличаются. После переконфигурирования необходимо повторно сохранить контекст.

Данная возможность доступна только для крейт-контроллеров с расширением MCS (только в крейт-контроллерах LTR032).

```
ltr24          [in]
```

Управляющая структура модуля.

Возвращает:

LTR\_OK или код ошибки.

## LTR24\_FindFrameStart

```
INT          LTR24_FindFrameStart (TLTR24          *ltr24,  
                                   const DWORD *data,  
                                   INT          size,  
                                   INT          *index  
                                   );
```

Находит номер слова, являющегося началом кадра. Используется для того, чтобы после восстановления управляющей структуры из крейт-контроллера восстановить выравнивание по границе кадра в сбившемся потоке данных.

```
ltr24          [in]
```

Управляющая структура модуля.

```
data          [in]
```

Массив сырых данных.

```
size          [in]
```

Количество сырых данных.

```
index         [out]
```

Индекс начала кадра.

Возвращает:

LTR\_OK или код ошибки.

## 4.4. Форматы данных

### Кадр

Сырые данные от модуля поступают кадрами. Кадр представляет собой последовательность отсчетов по всем включенным каналам в порядке увеличения номера канала. Для 20-битного формата данных отсчет соответствует одному 32-битному слову, для 24-битного формата – два 32-битных слова. Формат слов в отсчете приведены в разделе форматы команд и данных.

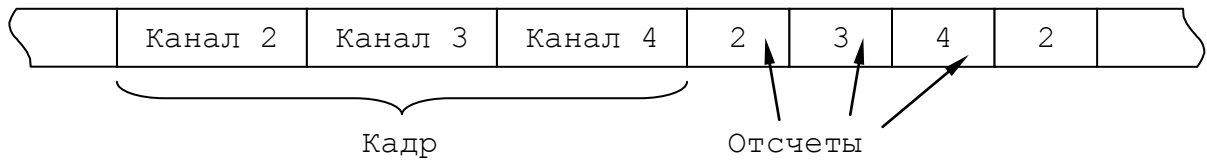


Рисунок 4-2. Последовательность принимаемых данных (включены каналы 2, 3, 4)

### 20-битный отсчет

Передается 32-битным словом.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	-	-	-	-	-	-	0	C	N	D19	D18	D17	D16	

D0 – D19

20-битный код АЦП.

N

Номер канала.

C

Счетчик данных. Устанавливается в 1 для каждого 15-го слова.

### 24-битный отсчет

Отсчет передается двумя расположенными последовательно 32-битными словами в следующем порядке: HIGH, LOW.

HIGH:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	V	D23	D22	D21	D20	D19	D18	D17	D16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	-	-	-	-	-	-	1	0	N	C				

LOW:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	-	-	-	-	-	-	1	1	N	C				

D0 – D23

24-битный код АЦП.

N

Номер канала.

C

Счетчик данных по модулю 15 (счет по кругу от 0 до 14). Значение счетчика одинаково для обеих частей отсчета.

V

Признак перегрузки входного тракта канала.