

МНОГОКАНАЛЬНЫЕ СИСТЕМЫ СБОРА ДАННЫХ

БИБЛИОТЕКА LTRAP1
РУКОВОДСТВО ПРОГРАММИСТА

*Ревизия 2.0.2
Январь 2019*

Авторы руководства:

Кодоркин А.В., Емельянов А.С., [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	23.01.2006	Первая доступная для пользователя ревизия.
1.0.1	23.04.2006	Добавлено описание меток времени.
1.0.2	25.07.2006	Добавлена функция LTR_SetServerProcessPriority
1.0.3	02.04.2007	Отмечены особенности многопоточковой работы
1.0.4	23.04.2007	Добавлено более подробное описание предупреждения LTR_WARNING_MODULE_IN_USE, и изменены примеры
1.0.5	04.05.2008	Добавлено описание функции LTR_GetCrateRawData
1.0.6	04.09.2008	Добавлены примеры для функции LTR_GetCrateRawData
1.0.7	05.09.2008	Обновлено описание LTR_GetCrateRawData и флагов структуры TLTR
1.0.8	03.09.2009	Проведена корректура текста, добавлено описание функции LTR_GetCrateInfo, дополнен список констант и структур, добавлено описание функций синхронизации и коммутации служебных сигналов крейт-контроллеров LTR-EU
1.0.9	11.01.2010	Добавлены описания новых функций API — управление сервером и соединениями с IP-крейтами из прикладной программы. Добавлен параграф "сообщения сетевой безопасности".
1.0.10	31.03.2010	Добавлено описание функции LTR_GetServerVersion
2.0.0	14.10.2016	Переработано описание. Изменения описаны в разделе "Отличия второй версии документа"
2.0.1	07.11.2016	Добавлено описание флага LTR_CRATE_IP_FLAG_RECONNECT для IP-записи и флага LTR_GETCRATES_FLAGS_WORKMODE_ONLY функции получения информации о подключенных крейтах
2.0.2	09.01.2019	Добавлено описание настраиваемых параметров службы LTRD_PARAM_MODULE_SEND_BUF_SIZE, LTRD_PARAM_MODULE_RECV_BUF_SIZE и LTRD_PARAM_ETH_SEND_NODELAY. Дополнено описание новых кодов ошибок.

Оглавление

1	О чем этот документ	6
2	Отличия второй версии документа	7
3	Установка и подключение библиотеки к проекту	9
4	Общий подход к работе с библиотекой	10
4.1	Последовательность вызова функций	10
4.2	Типы клиентских соединений	11
4.2.1	Управляющее соединение со службой ltrd	11
4.2.2	Управляющее соединение с крейтом	11
4.2.3	Соединение с конкретным модулем	12
4.3	Множественное подключение	12
4.4	Получение списка подключенных крейтов и модулей	13
4.5	Формат задания IP-адресов	14
4.6	Синхрометки	14
4.6.1	Прием и сопоставление синхрометок с данными	15
4.6.2	Синхронизация нескольких крейтов	16
5	Константы, типы данных и функции библиотеки	18
5.1	Константы и перечисления.	18
5.1.1	Константы и макроопределения	18
5.1.2	Коды ошибок.	20
5.1.3	Режим подключения пользовательских выводов процессора крейта.	23
5.1.4	Режим работы выходов DIGOUTx крейта.	23
5.1.5	Режим генерации синхрометок.	24
5.1.6	Уровень вывода журнала службой ltrd.	24
5.1.7	Флаги функции получения информации о подключенных крейтах.	25
5.1.8	Настраиваемые параметры службы ltrd.	26
5.1.9	Номера каналов для соединения со службой ltrd	28
5.1.10	Флаги канала связи ltrd для явного задания интерфейса крейта	29
5.1.11	Дополнительные флаги канала связи с ltrd	29
5.1.12	Флаги состояния соединения	29
5.1.13	Идентификаторы модулей	29
5.1.14	Типы крейтов	30
5.1.15	Интерфейс подключения крейта	30
5.1.16	Состояние соединения с крейтом, соответствующим записи с IP-адресом	31
5.1.17	Флаги, соответствующие записи с IP-адресом крейта	32
5.1.18	Флаги из описания модуля	32

5.1.19	Режим работы крейта	33
5.1.20	Состояние ПЛИС	33
5.2	Типы данных	34
5.2.1	Описатель соединения	34
5.2.2	Конфигурация линий разъема синхронизации	36
5.2.3	Информация о типе и интерфейсе подключения крейта	36
5.2.4	Запись с IP-адресом крейта	37
5.2.5	Статистика крейта	37
5.2.6	Статистика модуля	39
5.2.7	Информация о крейте и его прошивке	41
5.3	Функции	42
5.3.1	Функции инициализации и работы с соединением	42
5.3.1.1	Инициализация описателя соединения	42
5.3.1.2	Открытие соединения	42
5.3.1.3	Открытие управляющего соединения со службой ltrd	43
5.3.1.4	Открытие управляющего соединения с крейтом	44
5.3.1.5	Открытие соединения с заданным таймаутом	45
5.3.1.6	Закрытие соединения	45
5.3.1.7	Проверка, открыто ли соединение	45
5.3.2	Функции информационного характера	46
5.3.2.1	Получение версии службы ltrd	46
5.3.2.2	Получение серийных номеров подключенных крейтов	47
5.3.2.3	Получение информации о подключенных крейтах	48
5.3.2.4	Получение описания крейта	49
5.3.2.5	Получение статистики по крейту	49
5.3.2.6	Получение статистики по модулю	50
5.3.3	Функции управления крейтами	51
5.3.3.1	Получение списка модулей в крейте	51
5.3.3.2	Получение информации о типе и интерфейсе подключения крейта	51
5.3.3.3	Конфигурация линий разъема синхронизации крейта	52
5.3.3.4	Настройка генерации метки "СТАРТ"	52
5.3.3.5	Запуск генерации меток "СЕКУНДА"	53
5.3.3.6	Останов генерации меток "СЕКУНДА"	53
5.3.4	Функции управления службой ltrd	54
5.3.4.1	Сброс указанного модуля	54
5.3.4.2	Установка уровня журнала	55
5.3.4.3	Получение текущего уровня журнала	55
5.3.4.4	Установка параметров работы службы ltrd	56
5.3.4.5	Чтение параметров работы службы ltrd	56
5.3.4.6	Перезапуск службы ltrd	57
5.3.4.7	Останов службы ltrd	57
5.3.5	Функции управления подключением крейтов по Ethernet	58
5.3.5.1	Получение списка записей с IP-адресами крейтов	59
5.3.5.2	Добавление записи с IP-адресом крейта	60
5.3.5.3	Удаление записи с IP-адресом крейта	60
5.3.5.4	Установление соединения с крейтом по IP-адресу	61
5.3.5.5	Разрыв соединения с крейтом по IP-адресу	62

5.3.5.6	Установление соединения со всеми крейтами с признаком автоподключения	62
5.3.5.7	Разрыв соединения со всеми крейтами, подключенными по Ethernet.	63
5.3.5.8	Установка флагов для записи с IP-адресом крейта	63
5.3.6	Функции для обмена данными с модулями	64
5.3.6.1	Прием данных от модуля	64
5.3.6.2	Передача данных в модуль	65
5.3.6.3	Чтение времени последней секундной метки	65
5.3.7	Вспомогательные функции	66
5.3.7.1	Получение текстового сообщения об ошибке	66
5.3.7.2	Установка таймаута по умолчанию для соединения	66

Глава 1

О чем этот документ

Данный документ предназначен в первую очередь для программистов, которые собираются писать свои программы для ПК для работы с крейтами LTR с использованием библиотек, предоставляемых фирмой “Л Кард”.

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. Также документ предполагает знакомство пользователя с документом [“Крейтовая система LTR. Руководство пользователя”](#).

Данный документ содержит описание только одной библиотеки из набора библиотек для работы с LTR — базовой библиотеки для работы с крейтами LTR, файл которой имеет название `ltrapi` (как и общий набор библиотек). В данной библиотеке содержатся общие управляющие функции для работы со службой `ltrd`, функции управления крейтами и получения информации о подключенных крейтах и модулях. Также в данном документе описаны некоторые общие концепции работы с крейтами LTR, общие для всех модулей. При этом в данном документе не содержится описания функций для работы с конкретными модулями, так как для каждого модуля предназначена своя специализированная библиотека с именем `ltrXXXapi` (где XXX — номер модуля), использующая функции базовой, и для каждой такой библиотеки существует свой документ `ltrXXXapi.pdf` с подробным описанием всех функций и типов данной библиотеки.

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 свое значения и для других языков программирования. Поэтому этот документ полезен и пользователям, пишущим на других языках программирования.

Глава 2

Отличия второй версии документа

Данный документ является второй, значительно переработанной версией описания библиотеки `ltrapi`. Данная версия содержит обновленную информацию с учетом замены ранее используемой программы LTR Server на службу `ltrd`, возможности использования библиотек не только под ОС Windows, а также включает дополнения и изменения (совместимые с библиотеками более старых версий), сделанные в версии библиотек 1.31.x по сравнению с библиотеками версии 1.27.x, к которой относится [первая версия документа](#).

Также переработано описание общего подхода и местами более подробно описана работа некоторых функций. В общем описании рассмотрены некоторые дополнительные вопросы, включая описание [типов клиентских соединений](#) с указанием, какие функции применимы для какого типа, описание принципа работы с [синхрометками](#) и др.

Из изменений, связанных с более новыми версиями библиотеки, можно отметить:

- Все константы и типы, используемые в данном документе, для избежания пересечения названий с другими библиотеками имеют префикс `LTR_` или `TLTR_`, соответственно. Старые константы и типы без префиксов оставлены в библиотеке для обратной совместимости и вынесены в отдельный заголовочный файл `ltrapi_compat_defs.h`, который включается по умолчанию, но может быть при желании явно запрещен через определение `LTRAPI_DISABLE_COMPAT_DEFS`.
- Добавлены вспомогательные функции для более удобного открытия управляющих соединений нужного типа — `LTR_OpenSvcControl()` и `LTR_OpenCrate()`.
- Добавлено описание новых функций, работающих только с `ltrd`, которые не поддерживались программой LTR Server (указано в описании функций).
- Добавлены новые коды ошибок, включая обобщенные коды, одинаковые для всех модулей. Библиотеки модулей используют в первую очередь данные общие коды из `ltrapi` и возвращают одинаковую ошибку в одинаковых ситуациях вместо того, чтобы определять свой код в каждом модуле для одних и тех же случаев. Кроме того, часть функций возвращают код ошибки, более подробно указывающий на причину ошибки, а не на саму функцию, где ошибка произошла (так как приложению верхнего уровня уже известно, какая функция вернула ошибку).
- Изменено поведение библиотеки при попытке открыть соединение с модулем, с которым уже было открыто соединение. Теперь это не приводит к нарушению первого соединения, однако повторно открытое соединение имеет ограниченные возможности и может не позволять полноценно работать с модулем. Для

возможности принудительного сброса первого соединения введена функция [LTR_ResetModule\(\)](#). Более подробное описание представлено в разделе [Множественное подключение](#).

- Убраны из описания редко используемые функции, которые не были реализованы в ltrd ввиду отсутствия запросов в их потребности от клиентов. При этом функции оставлены в библиотеке для обратной совместимости. Это функции:
 - LTR_GetCrateRawData()
 - LTR_SetServerProcessPriority()
 - LTR_GetServerProcessPriority()
- Убраны из описания также функции, которые были введены, но реальная работа которых не была поддержана крейтами LTR (при этом функции оставлены в библиотеке для обратной совместимости):
 - LTR_GetIPCrateDiscoveryMode()
 - LTR_SetIPCrateDiscoveryMode()

Глава 3

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 4

Общий подход к работе с библиотекой

4.1 Последовательность вызова функций

При работе с крейтовой системой LTR через библиотеку `ltrapi` функции библиотеки не работают напрямую с крейтами или модулями. Вместо этого они используют службу `ltrd`, которая берет на себя взаимодействие с крейтами, устанавливая клиентское соединение со службой, которое уже может быть связано с определенным модулем или крейтом.

Для описания клиентского соединения служит структура `TLTR`, далее называемая описателем соединения, с использованием которой выполняются практически все функции данной библиотеки. Программа может иметь несколько соединений с `ltrd` и для каждого параллельного соединения необходимо создавать свой описатель соединения (экземпляр структуры `TLTR`).

Таким образом, перед выполнением любых операций необходимо сперва установить клиентское соединение, а после их выполнения закрыть его.

Соответственно, типичная последовательность вызовов выглядит следующим образом:

1. Создать экземпляр структуры `TLTR` и проинициализировать его, вызвав функцию `LTR_Init()`.
2. Установить соединения с `ltrd` одним из двух способов
 - вызвав функцию `LTR_OpenSvcControl()` или `LTR_OpenCrate()` в зависимости от типа соединения.
 - или заполнить необходимые поля структуры `TLTR` вручную и вызвать `LTR_Open()`
3. Вызов нужных управляющих функций или обмен данными с модулем
4. По завершению работы закрыть соединение через `LTR_Close()`

Возможные действия после установления соединения зависят от типа соединения. Типы соединений описаны подробно в [следующем разделе](#).

Клиентское соединение между пользовательской программой и службой `ltrd` выполняется через сокеты. Это позволяет при необходимости подключаться из приложения к службе `ltrd`, запущенной на другой машине. В связи с этим в функциях открытия соединения участвуют параметры, указывающие адрес машины со службой `ltrd` и порт TCP для подключения (при использовании `LTR_Open()` эти данные задаются через

поля [описателя соединения saddr](#) и [sport](#)). В обычном режиме, когда служба и программа запущены на одной машине и используются TCP порт по умолчанию, в качестве адреса используется константа [LTRD_ADDR_DEFAULT](#), а в качестве порта — [LTRD_PORT_DEFAULT](#).

4.2 Типы клиентских соединений

Можно выделить три типа клиентских соединений:

- управляющее соединение со службой [ltrd](#)
- управляющее соединение с крейтом
- соединение с конкретным модулем

4.2.1 Управляющее соединение со службой [ltrd](#)

Для открытия соединения данного типа служит функция [LTR_OpenSvcControl\(\)](#). Также для этой цели можно использовать функцию [LTR_Open\(\)](#), предварительно установив номер канала в поле [cc](#) равным [LTR_CC_CHNUM_CONTROL](#), а поле [csn](#) заполнить специальной строкой [LTR_CSN_SERVER_CONTROL](#).

Соединение данного типа может быть выполнено даже если нет ни одного подключенного крейта.

С использованием данного соединения можно выполнять управляющие команды, описанные в разделах:

- [Функции информационного характера](#)
- [Функции управления службой \[ltrd\]\(#\)](#)
- [Функции управления подключением крейтов по Ethernet](#)

4.2.2 Управляющее соединение с крейтом

Для открытия соединения данного типа служит функция [LTR_OpenCrate\(\)](#). Также для этой цели можно использовать функцию [LTR_Open\(\)](#), предварительно установив номер канала (поле [cc](#)) равным [LTR_CC_CHNUM_CONTROL](#) и заполнить поле [csn](#) серийным номером крейта (или пустой строкой, если нужно установить соединение с первым крейтом).

Для того, чтобы установить данный тип соединения, должен быть хотя бы один подключенный крейт (хотя бы один крейт в списке активных крейтов службы [ltrd](#)).

Данное соединение позволяет выполнять команды, описанные в разделе [Функции управления крейтами](#). Технически через данное соединение можно выполнять и все команды управляющего соединения с [ltrd](#), однако для этого больше подходит соответствующее соединение, т.к. не требует обязательного наличия подключенного крейта.

Для определения, с каким крейтом будет связано данное соединение, как правило достаточно указать серийный номера крейта (или пустую строку, если крейт всегда один). Однако один крейт может быть подключен одновременно по двум интерфейсам, например в случае, если крейт настроен на работу по Ethernet и подключен по этому интерфейсу, но также и подключен по USB в режиме настройки. В связи с этим в общем

случае подключение к крейту определяется двумя параметрами — серийным номером крейта и интерфейсом подключения крейта. Без явного указания интерфейса соединение будет связано с крейтом с использованием интерфейса, который является рабочим для крейта, т.е. через который можно обмениваться данными с модулями крейта, что и необходимо для большинства программ. Однако при необходимости можно явно указать интерфейс подключения крейта через параметр функции `LTR_OpenCrate()` или через флаги `en_LTR_CC_Iface` поля `cc` при использовании `LTR_Open()`.

4.2.3 Соединение с конкретным модулем

Соединение данного типа выполняется с помощью функции `LTR_Open()`, с предварительно заполненной в структуре `TLTR` информацией о крейте, в котором находится нужный модуль, аналогично управляющему соединению с крейтом, но с указанным в поле `cc` номером слота крейта (вместо `LTR_CC_CHNUM_CONTROL`), в который вставлен нужный модуль, с помощью значений от `LTR_CC_CHNUM_MODULE1` до `LTR_CC_CHNUM_MODULE16`.

По данному соединению клиент обменивается данными с модулем по специализированному протоколу данного модуля с использованием функций из раздела [Функции для обмена данными с модулями](#).

Однако пользователю обычно не требуется явно открывать данное соединение через `LTR_Open()` и работать через функции `ltrapi`, так как для работы с модулями используются специализированные библиотеки со своими функциями, которые уже внутри себя используют функции `ltrapi`.

4.3 Множественное подключение

В случае управляющих соединений, клиент может параллельно открывать несколько соединений как со службой, так и с крейтами (как с одним крейтом, так и с несколькими). Каждое новое соединение не зависит от предыдущего и позволяет выполнять любой доступный для данного типа соединения набор функций.

Иначе обстоит дело с подключениями для работы с конкретными модулями. Хотя служба `ltrd` позволяет устанавливать одновременно несколько соединений с одним модулем, но протокол работы с модулями таков, что для корректной работы с конкретным модулем должно быть одновременно установлено только одно соединение.

При попытке открыть соединение с модулем, с которым уже есть установленное клиентское соединение (из этой или другой программы), функция открытия соединения возвращает ошибку `LTR_WARNING_MODULE_IN_USE`.

Обработка библиотеками данной ситуации несколько изменилось в последних версиях `ltrapi` по сравнению с версией 1.27.

В версии 1.27 и ниже функции открытия соединения возвращали данную ошибку, однако выполнили все действия, включая полный сброс и начальную инициализацию модуля. Это позволяло работать с модулем также, как если бы не было первого подключения, но только при условии, если по первому подключению не будут передаваться команды, которые могут нарушить последовательность работы с модулем. В такой ситуации данное предупреждение можно было полностью игнорировать и работать с модулем, как если бы открытие завершилось успешно. Однако проблемой в таком варианте является то, что если первое соединение действительно используется и идет сбор данных, то

попытка повторного установления соединения (возможно из совсем другой программы, которая ничего не знает о первом соединении), приведет к нарушению идущего сбора данных для первого соединения и более того, не гарантирует нормальной работы для второго, если первое не будет сразу закрыто. В частности при запуске программы, которая в первую очередь находит все модули и получает от них информацию, приводит к нарушению работающих в это время всех других программ.

В версиях выше 1.27 при открытии соединения с модулем, если с этим модулем уже существует открытое соединение, сброс модуля, как и посылка любых других команд модулю не выполняется. То есть это не нарушает работу первого соединения, однако модуль для второго соединения находится в неизвестном состоянии и, кроме того, оказываются невыполненными операции из функции открытия, специфичные для модуля (например, для модулей, в которых чтение информации из flash-памяти выполняется при открытии соединения эта информация прочитана не будет). Таким образом, в текущей версии ltrapi, если функция открытия соединения вернула ошибку `LTR_WARNING_MODULE_IN_USE`, то такое соединение может использоваться лишь для ограниченного круга задач, а при штатной работе рекомендуется считать такое предупреждение равносильным ошибке и, как и при любой другой ошибке, закрыть соединение. В этом случае запуск программы, описанной в предыдущем примере, приведет к тому, что она не сможет прочитать информацию о модулях, с которыми уже открыты работающие соединения, но работа ранее запущенных программ не будет нарушена. В случае же, если необходимо открыть новое соединение, сбросив предыдущее, то можно воспользоваться функцией `LTR_ResetModule()` через управляющее соединение со службой ltrd.

4.4 Получение списка подключенных крейтов и модулей

Функции ltrapi позволяют получить список крейтов, с которыми служба ltrd установила соединение, который называется списком активных крейтов службы ltrd. При этом служба сама добавляет в этот список все найденные и успешно проинициализированные крейты, подключенные по интерфейсу USB, в то время как подключение к крейтом по интерфейсу Ethernet выполняется посредством настроенных записей с IP-адресами крейтов, функции для работы с которыми описаны в разделе [Функции управления подключением крейтов по Ethernet](#).

Список активных крейтов можно получить через управляющее соединение со службой ltrd, в то время как для получения списка модулей крейта необходимо устанавливать управляющее соединение с соответствующим крейтом. Таким образом, последовательность вызовов для получения полного списка активных крейтов и модулей может выглядеть следующим образом:

1. Создать экземпляр структуры `TLTR` и проинициализировать его, вызвав функцию `LTR_Init()`.
2. Установить управляющее соединение со службой ltrd с помощью функции `LTR_OpenSvcControl()`.
3. Получить список серийных номеров всех активных крейтов с помощью функции `LTR_GetCrates()` или `LTR_GetCratesEx()`.

4. Для каждого действительного серийного номера крейта из полученного списка выполнить следующие действия:
 - Создать новый экземпляр структуры `TLTR` и проинициализировать его, вызвав функцию `LTR_Init()`.
 - Установить управляющее соединение с крейтом, с текущим серийным номером из списка, вызвав `LTR_OpenCrate()`
 - Получить список модулей в крейте, вызвав `LTR_GetCrateModules()`
 - Закрыть управляющее соединение с крейтом с помощью `LTR_Close()`
5. Закрыть управляющее соединение со службой `ltrd`, вызвав `LTR_Close()` для описателя соединения, созданного на шаге 1.

4.5 Формат задания IP-адресов

IP-адреса (здесь и далее под IP-адресом подразумевается адрес для протокола IP версии 4, которая поддерживается крейтами и библиотекой) используются как для подключения крейтов по интерфейсу Ethernet (IP-адрес крейта), так и при установлении клиентского подключения к службе `ltrd`, запущенной на удаленной машине (IP-адрес машины со службой `ltrd`).

IP адрес в текстовом виде представляет собой четыре цифры от 0 до 255, разделенные точками, например “192.168.1.12”. Для задания IP-адресов в функциях и структурах данной библиотеки (как и в других библиотеках, связанных с `ltrapi`) как правило используется 32-битное число (типа `DWORD`). Каждый байт данного числа задает, соответственно, одну цифру из текстовой записи. При этом первой цифре соответствует старший байт числа, а последней — младший.

Таким образом, для адреса “a.b.c.d” 32-битное число будет определяться как $(a \ll 24) | (b \ll 16) | (c \ll 8) | d$. Например для задания адреса 192.168.1.12 значение параметра в шестнадцатеричном виде будет равно `0xC0A8010C`.

4.6 Синхрометки

Синхрометки — это общий механизм синхронизации крейтов LTR, который описан в главе “Принцип синхронизации сбора данных в системе LTR” [руководства пользователя](#). Общий принцип заключается в том, что в общий поток данных от крейта могут вставляться специальные метки. Благодаря тому, что данные от всех модулей идут общим потоком в том порядке, в котором они попали в крейт, то на приемной стороне по положению метки для каждого модуля можно определить, между какими отсчетами модуля возникло событие, соответствующее этой метки.

Например, этот механизм позволяет привязать с точностью до одного периода дискретизации модуля следующие данные:

- данные от разных модулей одного крейта относительно друг друга
- данные от разных модулей разных крейтов, если крейты соединены через разъем синхронизации и используется генерация меток по принципу “ведущий-ведомые”.

- данные от разных модулей одного или нескольких крейтов к внешним событиям, если используется генерация метки от общего внешнего сигнала, подключенного к разъему синхронизации крейтов.

В системе LTR используется два типа синхрометок — метка “СТАРТ” и метка “СЕКУНДА”. Названия меток отражают их изначальное назначение при внутренней генерации, однако при генерации от внешнего сигнала их назначение может быть произвольным и их можно рассматривать как привязку к двум разным типам синхрособытий.

Метки могут вставляться в поток либо специальными модулями (LTR41, LTR42, LTR43), либо самим крейтом, если он поддерживает данный механизм (поддерживается в крейтах LTR-EU, LTR-CEU, LTR-CU). Также в урезанном виде синхрометки поддерживаются крейтом LTR-U-1-4, однако возможности генерации меток в этом крейте и точность привязки их к данным ограничены и при необходимости использования синхрометок в одноместном крейте рекомендуется применять крейты LTR-CEU-1 или LTR-CU-1.

Функции для настройки меток, генерируемых модулями, описаны в руководствах программистов для этих модулей. Функции же для настройки меток, генерируемых крейтами, реализованы в `ltrapi` и описаны в данном документе. Исключением является крейт LTR-U-1-4, функции настроек меток которого реализованы в библиотеке `ltr021api` и в данном документе не рассматриваются.

Настройка генерации меток крейтов выполняется через [управляющее соединение с крейтом](#) с помощью функций `LTR_MakeStartMark()`, `LTR_StartSecondMark()` и `LTR_StopSecondMark()`. Для трансляции метки на выходы разъема синхронизации необходимо также предварительно настроить линии разъема синхронизации через `LTR_Config()`.

4.6.1 Прием и сопоставление синхрометок с данными

Служба `ltrd` принимает синхрометки в потоке данных от крейта и считает общее их количество, начиная с момента подключения крейта службой `ltrd`. При приеме данных от модуля с помощью функции `LTR_Recv()` принимается также и информация о моментах возникновения синхрометок с их счетчиком (в библиотеках конкретных модулей используется своя функция приема данных, однако ее принцип работы и параметры повторяют параметры функции `LTR_Recv()`). В результате функция `LTR_Recv()` на основе этой информации при необходимости формирует отдельный массив `tmark` из 32-битных слов. Каждое слово этого массива соответствует принятому слову от модуля с тем же номером элемента в массиве и содержит в себе информацию о счетчиках синхрометок на момент времени, соответствующему данному слову модуля. Каждое 32-битное слово `tmark` содержит в старшей половине счетчик меток “СТАРТ”, а в младшей половине счетчик сеток “СЕКУНДА”, как показано в таблице:

Номер бита	Значение
Биты 31-16	Количество меток “СТАРТ”, возникших до прихода соответствующего слова в крейт
Биты 15-0	Количество меток “СЕКУНДА”, возникших до прихода соответствующего слова в крейт

Соответственно, по моменту изменения счетчика можно определить между какими словами от модуля возникло синхрособытие. Например, если принято 10 слов от модуля, а время возникновения синхрометки соответствует времени между 4 и 5 словом, а N — количество синхрометок данного типа, пришедших до первого слова, то в первых 4-х элементах массива `tmakr` будет указан счетчик синхрометок данного типа равный N , а в оставшихся шести — $N + 1$.

Вставка синхрометки в поток данных, как и, соответственно, привязка синхрометки к данным модуля происходит на уровне крейта при приеме слов от модулей (до буферизации), т.е. если говорить более строго, изменение счетчика синхрометки между двумя словами модуля означает, что синхрособытие произошло после того, как в крейт пришло первое слово, но до того, как пришло второе. Для более точной привязки данных от разных типов модулей нужно также учесть задержки, вносимые самим модулем, например, если используется АЦП с фильтром, то следует учесть задержку этого фильтра. Эти задержки фиксированы для определенных настроек определенного модуля и могут быть получены, например, экспериментально для нужного случая.

Следует также отметить, что так как счетчик синхрометок идет от начала установления соединения службы `ltrd` с крейтом и не связан с клиентскими соединениями с модулями, то на момент начала сбора значения счетчиков синхрометки могут быть отличны от нуля, если до работы с модулем от крейта приходили метки. Также следует учитывать, что само значение счетчиков для слов, соответствующих одному времени, одинаково для разных модулей одного крейта, но может отличаться для модулей из разных крейтов. Соответственно, нужно учитывать разные начальные значения счетчиков при старте и привязываться к их изменению, а не к абсолютному значению.

4.6.2 Синхронизация нескольких крейтов

Синхронизация данных модулей от нескольких крейтов с помощью синхрометок возможна для крейтов с разъемом SYNC (LTR-EU, LTR-CEU, LTR-CU), либо при наличии в крейте модулей, способных генерировать метки. В данном документе рассматривается только первый вариант. Принцип при использовании специальных модулей для генерации меток аналогичен, но для настройки использует функции модулей, которые не относятся к данному документу.

При этом возможна как синхронизация нескольких крейтов между собой по принципу “Ведущий --- Ведомые”, так и синхронизация от внешнего сигнала (в этом случае все крейты настраиваются как “ведомые”). Для этого нужно соединить одну из линий DIGOUT разъема синхронизации ведущего крейта с одной из линий DIGIN разъема синхронизации каждого ведомого (при синхронизации от внешнего сигнала — подключить этот сигнал к одной из линий DIGIN всех используемых крейтов). Если используются обе метки, то для каждой метки должен использоваться свой отдельный вход у ведущего крейта и свой вход у ведомых. Электрические вопросы такого подключения в данном руководстве не рассматриваются. Ниже рассматривается именно вопрос правильного использования функций данной библиотеки при выполнении такой синхронизации.

Для примера настройки возьмем вариант синхронизации по принципу “Ведущий --- Ведомые”. При этом будем считать, что выход DIGOUT1 ведущего крейта соединен с DIGIN1 всех ведомых и используется для метки “СТАРТ”, а выход DIGOUT2 ведущего — с DIGIN2 ведомых и используется для метки “СЕКУНДА”.

Общая последовательность настройки выглядит следующим образом:

1. В начальный момент генерация синхрометок во всех участвующих крейтах должна быть отключена. Если это может быть не так, то необходимо для каждого участвующего крейта отключить генерацию синхрометок, вызвав через управляющее соединение с каждым крейтом:
 - `LTR_MakeStartMark()` с указанием режима `LTR_MARK_OFF` для отключения генерации метки “СТАРТ”
 - `LTR_StopSecondMark()` для отключения генерации метки “СЕКUNДА”
2. Настройка выходов DIGOUT разъема синхронизации ведущего крейта. Для этого нужно вызвать функцию `LTR_Config()` для управляющего соединения с ведущим крейтом, передав структуру `TLTR_CONFIG`, заполненную следующим образом:
 - в поле `digout_en` установлено значение 1
 - в поле `digout[0]` установлено значение `LTR_DIGOUT_START`
 - в поле `digout[1]` установлено значение `LTR_DIGOUT_SECOND`
3. Запуск сбора для всех модулей, данные от которых следует синхронизировать. Этот пункт можно выполнить и после настройки синхрометок ведомых (пункт 4), если есть гарантия, что синхрособытие не возникнет в момент запуска сбора, когда часть модулей запущена, а часть нет. В случае использования меток от ведущего крейта это гарантируется, т.к. метки будут генерироваться только после вызова соответствующих функций для ведущего крейта.
4. Настройка генерации синхрометок от сигналов на соответствующих входах DIGIN разъема синхронизации для всех ведомых крейтов. Для этого для каждого ведомого крейта через управляющее соединения с соответствующим крейтом нужно:
 - вызвать функцию `LTR_MakeStartMark()` с указанием режима `LTR_MARK_EXT_DIGIN1_RISE` для настройки генерации метки “СТАРТ” по фронту сигнала на входе DIGIN1
 - вызвать функцию `LTR_StartSecondMark()` с указанием режима `LTR_MARK_EXT_DIGIN2_RISE` для настройки генерации метки “СЕКUNДА” по фронту сигнала на входе DIGIN2
5. Запуск секундных меток для ведущего крейта с помощью `LTR_MakeStartMark()` с указанием режима `LTR_MARK_INTERNAL` через управляющее соединение ведущего крейта
6. Генерация метки “СТАРТ” ведущим крейтом с помощью `LTR_MakeStartMark()` с указанием режима `LTR_MARK_INTERNAL` через управляющее соединение ведущего крейта

Настройка при генерации метки от внешнего сигнала аналогична, за исключением того, что нет ведущего крейта и соответствующие пункты можно опустить.

Глава 5

Константы, типы данных и функции библиотеки

5.1 Константы и перечисления.

5.1.1 Константы и макроопределения

Константа	Значение	Описание
LTRD_ADDR_LOCAL	(0x7F0000011)	IP-адрес для подключения к службе ltrd, соответствующий случаю, когда служба запущена на локальной машине (той же, откуда устанавливается соединение)
LTRD_ADDR_DEFAULT	(LTRD_ADDR_LOCAL)	IP-адрес по умолчанию для подключения к службе ltrd
LTRD_PORT_DEFAULT	(11111)	TCP-порт, используемый по умолчанию, для подключения к службе ltrd
LTR_CRATES_MAX	161	Максимальное количество крейтов, которое можно получить с помощью функции LTR_GetCrates() . В случае, если крейтов может быть больше, то можно воспользоваться функцией LTR_GetCratesEx() , которая не имеет ограничения на количество крейтов
LTR_MODULES_PER_CRATE_MAX	161	Максимальное количество модулей в одном крейте

LTR_CSN_SERVER_CONTROL	“#SERVER_CONTROL”	Если данная строка используется вместо серийного номера крейта при установлении подключения, то будет установлено управляющее соединение с ltrd, не связанное ни с одним крейтом
LTR_MID_MODULE	$((x) \& 0xFF) ((x) \& 0xFF) \ll 8)$	Макрос задания идентификатора модуля с указанным номером
LTR_MODULE_NAME_SIZE	16	Размер строки с названием модуля в описании модуля
LTR_CRATE_DEVNAME_SIZE	32	Размер строки с названием устройства в описании крейта
LTR_CRATE_SERIAL_SIZE	16	Размер строки с серийным номером крейта
LTR_CRATE_SOFTVER_SIZE	32	Размер строки с версией прошивки крейта в его описании
LTR_CRATE_REVISION_SIZE	16	Размер строки с ревизией крейта в его описании
LTR_CRATE_BOARD_OPTIONS_SIZE	16	Размер строки с описанием опций платы в описании крейта
LTR_CRATE_BOOTVER_SIZE	16	Размер строки с версией загрузчика в описании крейта
LTR_CRATE_CPUTYPE_SIZE	16	Размер строки с описанием процессора в описании крейта
LTR_CRATE_TYPE_NAME	16	Размер строки с описанием типа крейта
LTR_CRATE_SPECINFO_SIZE	48	Размер дополнительной информации о крейте
LTR_CRATE_FPGA_NAME_SIZE	32	Размер строки с описанием типа FPGA в описании крейта
LTR_CRATE_FPGA_VERSION_SIZE	32	Размер строки с версией прошивки FPGA в описании крейта
LTR_CRATE_THERM_MAX_CNT	8	Максимальное кол-во термометров в крейте, показания которых отображаются в статистике

LTR_DEFAULT_SEND_RECV_TIMEOUT	10000UL	Таймаут по умолчанию в мс на выполнение запросов к ltrd
-------------------------------	---------	---

5.1.2 Коды ошибок.

Тип: en_LTR_ERRORS		
Описание: Коды ошибок, которые могут возвращать функции библиотеки ltrapi. Также включают общие коды ошибок для модулей.		
Константа	Значение	Описание
LTR_OK	0	Операция выполнена без ошибок
LTR_ERROR_UNKNOWN	-1	Неизвестная ошибка.
LTR_ERROR_PARAMETERS	-2	Неверно задан один из параметров функции
LTR_ERROR_MEMORY_ALLOC	-3	Ошибка динамического выделения памяти.
LTR_ERROR_OPEN_CHANNEL	-4	Ошибка инициализации канала обмена с ltrd
LTR_ERROR_OPEN_SOCKET	-5	Ошибка подключения к ltrd
LTR_ERROR_CHANNEL_CLOSED	-6	Канал обмена с ltrd не был создан или закрыт
LTR_ERROR_SEND	-7	Ошибка передачи данных в ltrd
LTR_ERROR_RECV	-8	Ошибка приема данных от ltrd
LTR_ERROR_EXECUTE	-9	Ошибка обмена с крейт-контроллером
LTR_WARNING_MODULE_IN_USE	-10	Предупреждение: уже создано активное соединение с данным модулем
LTR_ERROR_NOT_CTRL_CHANNEL	-11	Данная операция доступна только для управляющего соединения
LTR_ERROR_SRV_INVALID_CMD	-12	Команда не поддерживается ltrd
LTR_ERROR_SRV_INVALID_CMD_PARAMS	-13	ltrd не поддерживает указанные параметры команды
LTR_ERROR_INVALID_CRATE	-14	Указанный крейт не найден
LTR_ERROR_EMPTY_SLOT	-15	В указанном слоте отсутствует модуль
LTR_ERROR_UNSUP_CMD_FOR_SRV_CTL	-16	Команда не поддерживается управляющим соединением
LTR_ERROR_INVALID_IP_ENTRY	-17	Неверная запись сетевого адреса крейта
LTR_ERROR_NOT_IMPLEMENTED	-18	Данная возможность не реализована
LTR_ERROR_CONNECTION_CLOSED	-19	Соединение было закрыто службой ltrd

LTR_ERROR_LTRD_UNKNOWN_RETCODE	-20	Неизвестный код ошибки службы ltrd
LTR_ERROR_LTRD_CMD_FAILED	-21	Ошибка выполнения управляющей команды ltrd
LTR_ERROR_INVALID_CON_SLOT_NUM	-22	Указан неверный номер слота при открытии соединения
LTR_ERROR_INVALID_MODULE_DESCR	-40	Неверный описатель модуля
LTR_ERROR_INVALID_MODULE_SLOT	-41	Указан неверный слот для модуля
LTR_ERROR_INVALID_MODULE_ID	-42	Неверный ID-модуля в ответе на сброс
LTR_ERROR_NO_RESET_RESPONSE	-43	Нет ответа на сброс модуля
LTR_ERROR_SEND_INSUFFICIENT_DATA	-44	Передано меньше слов в модуль, чем запрашивалось
LTR_ERROR_RECV_INSUFFICIENT_DATA	-45	Принято меньше слов от модуля, чем запрашивалось
LTR_ERROR_NO_CMD_RESPONSE	-46	Нет ответа на переданную команду
LTR_ERROR_INVALID_CMD_RESPONSE	-47	Пришел неверный ответ на команду
LTR_ERROR_INVALID_RESP_PARITY	-48	Ошибка четности в принятом ответе на команду
LTR_ERROR_INVALID_CMD_PARITY	-49	Ошибка четности в переданной команде
LTR_ERROR_UNSUP_BY_FIRM_VER	-50	Возможность не поддерживается данной версией прошивки
LTR_ERROR_MODULE_STARTED	-51	Операция не допустима при запущенном сборе данных
LTR_ERROR_MODULE_STOPPED	-52	Сбор данных остановлен
LTR_ERROR_RECV_OVERFLOW	-53	Произошло переполнение буфера службы ltrd при приеме данных
LTR_ERROR_FIRM_FILE_OPEN	-54	Ошибка открытия файла прошивки
LTR_ERROR_FIRM_FILE_READ	-55	Ошибка чтения файла прошивки
LTR_ERROR_FIRM_FILE_FORMAT	-56	Ошибка формата файла прошивки
LTR_ERROR_FPGA_LOAD_READY_TOUT	-57	Превышен таймаут ожидания готовности ПЛИС к загрузке
LTR_ERROR_FPGA_LOAD_DONE_TOUT	-58	Превышен таймаут ожидания перехода ПЛИС в рабочий режим
LTR_ERROR_FPGA_IS_NOT_LOADED	-59	Прошивка ПЛИС не загружена
LTR_ERROR_FLASH_INVALID_ADDR	-60	Неверный адрес flash-памяти
LTR_ERROR_FLASH_WAIT_RDY_TOUT	-61	Превышен таймаут ожидания завершения записи или стирания Flash-памяти

LTR_ERROR_FIRSTFRAME_NOTFOUND	-62	Не найдено начало кадра данных в потоке от модуля
LTR_ERROR_CARDSCONFIG_UNSUPPORTED	-63	Крейт не поддерживает сохранение конфигурации модулей
LTR_ERROR_FLASH_OP_FAILED	-64	Ошибка выполнения операции с flash-памятью
LTR_ERROR_FLASH_NOT_PRESENT	-65	Flash-память не обнаружена
LTR_ERROR_FLASH_UNSUPPORTED_ID	-66	Обнаружен неподдерживаемый тип flash-памяти
LTR_ERROR_FLASH_UNALIGNED_ADDR	-67	Невыровненный адрес flash-памяти
LTR_ERROR_FLASH_VERIFY	-68	Ошибка при проверке записанных данных во flash-память
LTR_ERROR_FLASH_UNSUP_PAGE_SIZE	-69	Установлен неподдерживаемый размер страницы flash-памяти
LTR_ERROR_FLASH_INFO_NOT_PRESENT	-70	Отсутствует информация о модуле во flash-памяти
LTR_ERROR_FLASH_INFO_UNSUP_FORMAT	-71	Неподдерживаемый формат информации о модуле во flash-памяти
LTR_ERROR_FLASH_SET_PROTECTION	-72	Не удалось установить защиту flash-памяти
LTR_ERROR_FPGA_NO_POWER	-73	Нет питания микросхемы ПЛИС
LTR_ERROR_FPGA_INVALID_STATE	-74	Недействительное состояние загрузки ПЛИС
LTR_ERROR_FPGA_ENABLE	-75	Не удалось перевести ПЛИС в разрешенное состояние
LTR_ERROR_FPGA_AUTOLOAD_TOUT	-76	Истекло время ожидания автоматической загрузки ПЛИС
LTR_ERROR_PROCDATA_UNALIGNED	-77	Обрабатываемые данные не выравнены на границу кадра
LTR_ERROR_PROCDATA_CNTR	-78	Ошибка счетчика в обрабатываемых данных
LTR_ERROR_PROCDATA_CHNUM	-79	Неверный номер канала в обрабатываемых данных
LTR_ERROR_PROCDATA_WORD_SEQ	-80	Неверная последовательность слов в обрабатываемых данных
LTR_ERROR_FLASH_INFO_CRC	-81	Неверная контрольная сумма в записанной информации о модуле
LTR_ERROR_PROCDATA_UNEXP_CMD	-82	Обнаружена неожиданная команда в потоке данных
LTR_ERROR_UNSUP_BY_BOARD_REV	-83	Возможность не поддерживается данной ревизией платы
LTR_ERROR_MODULE_NOT_CONFIGURED	-84	Не выполнена конфигурация модуля

5.1.3 Режим подключения пользовательских выводов процессора крейта.

Тип: en_LTR_UserIoCfg		
Описание: Данные константы задают режим подключения конкретной ножки процессора, которые могут использоваться в некоторых крейтах при написании пользовательской прошивки процессора. Для штатной прошивки не имеют значения.		
Константа	Значение	Описание
LTR_USERIO_DIGIN1	1	Ножка является входом и подключена к DIGIN1
LTR_USERIO_DIGIN2	2	Ножка является входом и подключена к DIGIN2
LTR_USERIO_DIGOUT	0	Ножка является выходом
LTR_USERIO_DEFAULT	LTR_USERIO_DIGOUT	Значение по умолчанию

5.1.4 Режим работы выходов DIGOUTx крейта.

Тип: en_LTR_DigOutCfg		
Описание: Данные значения определяют, какой сигнал будет выставлен на определенном выходе DIGOUT разъема синхронизации крейта.		
Константа	Значение	Описание
LTR_DIGOUT_CONST0	0	Постоянный уровень логического "0"
LTR_DIGOUT_CONST1	1	Постоянный уровень логической "1"
LTR_DIGOUT_USERIO0	2	Выход подключен к ножке userio0
LTR_DIGOUT_USERIO1	3	Выход подключен к ножке userio1
LTR_DIGOUT_DIGIN1	4	Выход подключен ко входу DIGIN1
LTR_DIGOUT_DIGIN2	5	Выход подключен ко входу DIGIN2
LTR_DIGOUT_START	6	На выход подаются импульсы, соответствующие моментам генерации метки "СТАРТ"
LTR_DIGOUT_SECOND	7	На выход подаются импульсы, соответствующие моментам генерации метки "СЕКUNДА"
LTR_DIGOUT_IRIG	8	Контроль сигналов точного времени IRIG (digout1: готовность, digout2: секунда)
LTR_DIGOUT_DEFAULT	LTR_DIGOUT_CONST0	Значение по умолчанию

5.1.5 Режим генерации синхрометок.

Тип: en_LTR_MarkMode		
Описание: Данные значения используются для указания условия, по которому будут генерироваться метки “СТАРТ” или “СЕКУНДА”.		
Константа	Значение	Описание
LTR_MARK_OFF	0	Генерация метки отключена
LTR_MARK_EXT_DIGIN1_RISE	1	Метка генерируется по фронту сигнала на входе DIGIN1
LTR_MARK_EXT_DIGIN1_FALL	2	Метка генерируется по спаду сигнала на входе DIGIN1
LTR_MARK_EXT_DIGIN2_RISE	3	Метка генерируется по фронту сигнала на входе DIGIN2
LTR_MARK_EXT_DIGIN2_FALL	4	Метка генерируется по спаду сигнала на входе DIGIN2
LTR_MARK_INTERNAL	5	Внутренняя генерация метки крейт-контроллером. Для метки “СТАРТ” генерация выполняется однократно при вызове LTR_MakeStartMark() . Для метки “СЕКУНДА” — по таймеру крейта (раз в секунду), начиная с вызова LTR_StartSecondMark() .
LTR_MARK_SEC_IRIGB_DIGIN1	16	Источником метки является декодер сигналов точного времени IRIG-B006. Может использоваться только для секундных меток. В качестве входного сигнала используется сигнал со входа DIGIN1
LTR_MARK_SEC_IRIGB_nDIGIN1	17	Аналогично LTR_MARK_SEC_IRIGB_DIGIN1 , но в качестве входного сигнала используется инвертированный сигнал со входа DIGIN1
LTR_MARK_SEC_IRIGB_DIGIN2	18	Аналогично LTR_MARK_SEC_IRIGB_DIGIN1 , но в качестве входного сигнала используется сигнал со входа DIGIN2
LTR_MARK_SEC_IRIGB_nDIGIN2	19	Аналогично LTR_MARK_SEC_IRIGB_DIGIN1 , но в качестве входного сигнала используется инвертированный сигнал со входа DIGIN2

5.1.6 Уровень вывода журнала службой ltrd.

Тип: en_LTR_LogLevel		
Описание: Уровень вывода журнала службой ltrd.		
Константа	Значение	Описание
LTR_LOGLVL_ERR_FATAL	0	Фатальные ошибки
LTR_LOGLVL_ERR	1	Ошибки
LTR_LOGLVL_WARN	2	Предупреждения

LTR_LOGLVL_INFO	3	Информационные сообщения
LTR_LOGLVL_DETAIL	4	Детали
LTR_LOGLVL_DBG_HIGH	5	Отладочные сообщения повышенного уровня важности
LTR_LOGLVL_DBG_MED	6	Отладочные сообщения среднего уровня важности
LTR_LOGLVL_DBG_LOW	7	Отладочные сообщения низкого уровня важности

5.1.7 Флаги функции получения информации о подключенных крейтах.

Тип: en_LTR_GetCratesFlags		
Описание: Данные флаги могут управлять работой функции LTR_GetCratesEx() . В функцию в качестве параметра flags передается набор данных флагов, объединенных через битовое “ИЛИ”.		
Константа	Значение	Описание
LTR_GETCRATES_FLAGS_WORKMODE_ONLY	0x1	Флаг указывает, что функция должна вернуть только список крейтов, с которыми службой установлено рабочее соединение, в котором возможно управление модулями крейта. Так как на текущий момент у крейта в один момент времени только один интерфейс (заданный в настройках) позволяет работать с модулями, то при данном флаге не будет возвращаться несколько записей, связанных с одним крейтом, при одновременном подключении крейта по нескольким интерфейсам. Данный флаг доступен начиная с версии ltrd 2.1.5.0 и ltrapi 1.31.1.

5.1.8 Настраиваемые параметры службы ltrd.

Тип: en_LTRD_Params		
Описание: Список параметров настройки, управляющих работой службы ltrd. Данные параметры настраиваются через конфигурационный файл или через LTR_SetServerParameter() . В описании каждого кода указывается, каким образом задается значение параметра.		
Константа	Значение	Описание
LTRD_PARAM_ETH_CRATE_POLL_TIME	0x100	Интервал опроса крейта, для проверки, рабочее ли соединение с крейтом по интерфейсу Ethernet и обнаружения отключения крейта. Параметр типа DWORD задает время в мс.
LTRD_PARAM_ETH_CRATE_CON_TOUT	0x101	Таймаут на установление соединения службы с крейтом по Ethernet. Параметр типа DWORD задает время в мс.
LTRD_PARAM_ETH_CRATE_CTLCMD_TOUT	0x102	Таймаут на ответ на управляющую команду крейту по Ethernet. Параметр типа DWORD задает время в мс.
LTRD_PARAM_ETH_INTF_CHECK_TIME	0x103	Интервал проверки адресов хоста для запуска автоподключения. Параметр типа DWORD задает время в мс.
LTRD_PARAM_ETH_CRATE_RECONNECT_TIME	0x104	Время, через которое будет выполнено повторное подключения к крейту по Ethernet в случае возникновения ошибки, если для данной IP записи установлен флаг LTR_CRATE_IP_FLAG_RECONNECT . Параметр доступен начиная с версии ltrd 2.1.5.0 и ltrapi 1.31.1.

LTRD_PARAM_ETH_SEND_NODELAY	0x105	<p>Если параметр равен 1, то данные при передаче в крейт по TCP/IP посылаются немедленно без использования алгоритмов для объединения нескольких посылок в один пакет. Это позволяет в некоторых случаях (которые могут зависеть от версии ОС, ее настроек, конфигурации сети и т.п.) уменьшить время задержки при передаче небольших порций данных на вывод, но приводит к увеличению нагрузки на сеть из-за передачи малыми пакетами. По умолчанию данная опция отключена (параметр имеет значение 0).</p> <p>Параметр доступен начиная с версии ltrd 2.1.8.0 и ltrapi 1.31.3.</p>
LTRD_PARAM_MODULE_SEND_BUF_SIZE	0x200	<p>Размер буфера в 32-битных словах на передачу данных и команд в модуль. Выделяется службой ltrd для каждого модуля для промежуточного хранения слов, принятых от клиента, перед непосредственно передачей. Влияет на максимальное время задержки крейта при чтении данных модулей на вывод и соответственно на размер памяти, используемой службой. По умолчанию данный размер равен 512 Кслов и при отсутствии явной необходимости не требует изменения.</p> <p>Параметр доступен начиная с версии ltrd 2.1.6.0 и ltrapi 1.32.2.</p>

LTRD_PARAM_MODULE_RECV_BUF_SIZE	0x201	<p>Размер буфера в 32-битных словах на прием данных и команд из модуля. Выделяется службой ltrd для каждого модуля для промежуточного хранения слов, принятых от модуля, перед передачей клиенту. Влияет на максимальное время задержки клиента при чтении данных модулей на ввод и соответственно на размер памяти, используемой службой. По умолчанию данный размер равен 1024 Кслов и при отсутствии явной необходимости не требует изменения.</p> <p>Параметр доступен начиная с версии ltrd 2.1.6.0 и ltrapi 1.32.2.</p>
---------------------------------	-------	---

5.1.9 Номера каналов для соединения со службой ltrd

Тип: en_LTR_CC_ChNum		
Описание: Номера каналов для соединения со службой ltrd		
Константа	Значение	Описание
LTR_CC_CHNUM_CONTROL	0	Канал для передачи управляющих запросов крейту или службе ltrd
LTR_CC_CHNUM_MODULE1	1	Канал для работы с модулем в слоте 1
LTR_CC_CHNUM_MODULE2	2	Канал для работы с модулем в слоте 2
LTR_CC_CHNUM_MODULE3	3	Канал для работы с модулем в слоте 3
LTR_CC_CHNUM_MODULE4	4	Канал для работы с модулем в слоте 4
LTR_CC_CHNUM_MODULE5	5	Канал для работы с модулем в слоте 5
LTR_CC_CHNUM_MODULE6	6	Канал для работы с модулем в слоте 6
LTR_CC_CHNUM_MODULE7	7	Канал для работы с модулем в слоте 7
LTR_CC_CHNUM_MODULE8	8	Канал для работы с модулем в слоте 8
LTR_CC_CHNUM_MODULE9	9	Канал для работы с модулем в слоте 9
LTR_CC_CHNUM_MODULE10	10	Канал для работы с модулем в слоте 10
LTR_CC_CHNUM_MODULE11	11	Канал для работы с модулем в слоте 11
LTR_CC_CHNUM_MODULE12	12	Канал для работы с модулем в слоте 12
LTR_CC_CHNUM_MODULE13	13	Канал для работы с модулем в слоте 13
LTR_CC_CHNUM_MODULE14	14	Канал для работы с модулем в слоте 14
LTR_CC_CHNUM_MODULE15	15	Канал для работы с модулем в слоте 15
LTR_CC_CHNUM_MODULE16	16	Канал для работы с модулем в слоте 16

5.1.10 Фалаги канала связи ltrd для явного задания интерфейса крейта

Тип: en_LTR_CC_Iface		
Описание: Фалаги канала связи ltrd для явного задания интерфейса крейта		
Константа	Значение	Описание
LTR_CC_IFACE_USB	0x0100	Явное указание, что соединение должно быть с крейтом, подключенным по USB-интерфейсу
LTR_CC_IFACE_ETH	0x0200	Явное указание, что соединение должно быть с крейтом, подключенным по Ethernet (TCP/IP)

5.1.11 Дополнительные флаги канала связи с ltrd

Тип: en_LTR_CC_Flags		
Описание: Дополнительные флаги канала связи с ltrd		
Константа	Значение	Описание

5.1.12 Флаги состояния соединения

Тип: en_LTR_ChStateFlags		
Описание: Флаги состояния соединения		
Константа	Значение	Описание
LTR_FLAG_RBUF_OVF	(1u<<0)	Флаг переполнения буфера. Указывает что из-за того, что клиент не считывал данные, в ltrd произошло переполнение очереди клиента. соответственно в принятых данных есть разрыв
LTR_FLAG_RFULL_DATA	(1u<<1)	Флаг получения данных в полном формате в функции LTR_GetCrateRawData()

5.1.13 Идентификаторы модулей

Тип: en_LTR_MIDs		
Описание: Идентификаторы модулей		
Константа	Значение	Описание
LTR_MID_EMPTY	0	Пустой слот
LTR_MID_IDENTIFYING	0xFFFF	Модуль в процессе определения типа
LTR_MID_LTR01	LTR_MID_MODULE(1)	Идентификатор модуля LTR01
LTR_MID_LTR11	LTR_MID_MODULE(11)	Идентификатор модуля LTR11
LTR_MID_LTR22	LTR_MID_MODULE(22)	Идентификатор модуля LTR22
LTR_MID_LTR24	LTR_MID_MODULE(24)	Идентификатор модуля LTR24
LTR_MID_LTR25	LTR_MID_MODULE(25)	Идентификатор модуля LTR25
LTR_MID_LTR27	LTR_MID_MODULE(27)	Идентификатор модуля LTR27
LTR_MID_LTR34	LTR_MID_MODULE(34)	Идентификатор модуля LTR34
LTR_MID_LTR35	LTR_MID_MODULE(35)	Идентификатор модуля LTR35
LTR_MID_LTR41	LTR_MID_MODULE(41)	Идентификатор модуля LTR41

LTR_MID_LTR42	LTR_MID_MODULE(42)	Идентификатор модуля LTR42
LTR_MID_LTR43	LTR_MID_MODULE(43)	Идентификатор модуля LTR43
LTR_MID_LTR51	LTR_MID_MODULE(51)	Идентификатор модуля LTR51
LTR_MID_LTR114	LTR_MID_MODULE(114)	Идентификатор модуля LTR114
LTR_MID_LTR210	LTR_MID_MODULE(210)	Идентификатор модуля LTR210
LTR_MID_LTR212	LTR_MID_MODULE(212)	Идентификатор модуля LTR212
LTR_MID_LTR216	LTR_MID_MODULE(216)	Идентификатор модуля LTR216

5.1.14 Типы крейтов

Тип: en_LTR_CrateTypes		
Описание: Типы крейтов		
Константа	Значение	Описание
LTR_CRATE_TYPE_UNKNOWN	0	Неизвестный тип крейта
LTR_CRATE_TYPE_LTR010	10	Крейт LTR-U-8 или LTR-U-16
LTR_CRATE_TYPE_LTR021	21	Крейт LTR-U-1
LTR_CRATE_TYPE_LTR030	30	Крейт LTR-EU-8 или LTR-EU-16
LTR_CRATE_TYPE_LTR031	31	Крейт LTR-EU-2
LTR_CRATE_TYPE_LTR_CU_1	40	Крейт LTR-CU-1
LTR_CRATE_TYPE_LTR_CEU_1	41	Крейт LTR-CEU-1
LTR_CRATE_TYPE_BOOTLOADER	99	Крейт в режиме загрузчика (если в этом режиме нельзя узнать тип)

5.1.15 Интерфейс подключения крейта

Тип: en_LTR_CrateIface		
Описание: Интерфейс подключения крейта		
Константа	Значение	Описание
LTR_CRATE_IFACE_UNKNOWN	0	Неизвестный код интерфейса крейта. При передаче в функции данное значение может указывать, что интерфейс подключения крейта не имеет значения
LTR_CRATE_IFACE_USB	1	Крейт подключен по интерфейсу USB
LTR_CRATE_IFACE_TCP/IP	2	Крейт подключен по Ethernet (TCP/IP)

5.1.16 Состояние соединения с крейтом, соответствующим записи с IP-адресом

Тип: en_LTR_CrateIpStatus		
Описание: Состояние соединения с крейтом, соответствующим записи с IP-адресом		
Константа	Значение	Описание
LTR_CRATE_IP_STATUS_OFFLINE	0	Крейт не подключен
LTR_CRATE_IP_STATUS_CONNECTING	1	Идет процесс установки соединения с крейтом (операция запущена, но еще не завершена)
LTR_CRATE_IP_STATUS_ONLINE	2	Крейт подключен
LTR_CRATE_IP_STATUS_ERROR	3	Ошибка подключения крейта. Установить соединение не удалось.

5.1.17 Флаги, соответствующие записи с IP-адресом крейта

Тип: en_LTR_CrateIpFlags		
Описание: Флаги, соответствующие записи с IP-адресом крейта		
Константа	Значение	Описание
LTR_CRATE_IP_FLAG_AUTOCONNECT	0x1	Флаг указывает, что служба ltrd должна при старте или при обнаружении новой сети должна выполнить подключение к крейту с IP-адресом, соответствующим данной записи
LTR_CRATE_IP_FLAG_RECONNECT	0x2	Флаг указывает, что при ошибке установки соединения с крейтом или при разрыве связи по ошибке с работающим крейтом служба ltrd должна выполнить новую попытку подключения. Если флаг не установлен, то в этих случаях запись переходит в состояние LTR_CRATE_IP_STATUS_ERROR и никаких действий дальше не выполняется. Если флаг установлен, то через заданный интервал времени будет осуществлена новая попытка соединения и эти попытки будут повторяться до тех пор, пока соединение не будет установлено, либо пока не будет выполнена явная команда разрыва соединения, снят этот флаг или удалена соответствующая запись с IP-адресом. Данная возможность доступна начиная с версии ltrd 2.1.5.0 и ltrapi 1.31.1.

5.1.18 Флаги из описания модуля

Тип: en_LTR_ModuleDescrFlags		
Описание: Флаги из описания модуля		
Константа	Значение	Описание
LTR_MODULE_FLAGS_HIGH_BAUD	0x0001	Признак, что модуль использует высокую скорость интерфейса передачи слов в модуль
LTR_MODULE_FLAGS_USE_HARD_SEND_FIFO	0x0100	Признак, что модуль использует статистику внутреннего аппаратного FIFO на передачу данных
LTR_MODULE_FLAGS_USE_SYNC_MARK	0x0200	Признак, что модуль поддерживает генерирование синхромарок

5.1.19 Режим работы крейта

Тип: en_LTR_CrateMode		
Описание: Режим работы крейта		
Константа	Значение	Описание
LTR_CRATE_MODE_BOOTLOADER	1	Крейт находится в состоянии загрузчика
LTR_CRATE_MODE_WORK	2	Крейт находится в рабочем состоянии
LTR_CRATE_MODE_CONTROL	3	Крейт находится в состоянии, когда принимает только управляющие запросы (например, если крейт подключен не по тому интерфейсу, на который настроен)

5.1.20 Состояние ПЛИС

Тип: e_LTR_FPGA_STATE		
Описание: Константы, определяющие текущее состояние ПЛИС модуля. Данные константы используются в функциях модуля и внесены в ltrapi, т.к. являются общими для группы модулей, но они явно не используются функциями данной библиотеки		
Константа	Значение	Описание
LTR_FPGA_STATE_NO_POWER	0x0	Нет сигнала питания ПЛИС
LTR_FPGA_STATE_NSTATUS_TOUT	0x1	Истекло время ожидания готовности ПЛИС к загрузке
LTR_FPGA_STATE_CONF_DONE_TOUT	0x2	Истекло время ожидания завершения загрузки ПЛИС (обычно означает, что во Flash нет действительной прошивки)
LTR_FPGA_STATE_LOAD_PROGRESS	0x3	Идет загрузка ПЛИС
LTR_FPGA_STATE_POWER_ON	0x4	Состояние после POWER_ON. Свидетельствует о том, что было неожиданное снятие питания
LTR_FPGA_STATE_LOAD_DONE	0x7	Загрузка ПЛИС завершена, но работа ПЛИС еще не разрешена
LTR_FPGA_STATE_WORK	0xF	Нормальное рабочее состояние ПЛИС

5.2 Типы данных.

5.2.1 Описатель соединения.

Тип: TLTR		
Описание: Данная структура содержит всю информацию о соединении со службой ltrd. Данная структура используется в большинстве функций в качестве первого параметра. Часть полей (saddr, sport, csn, cc) предназначены для заполнения пользователем перед установкой соединения для задания параметров соединения и после этого уже не изменяются. Часть параметров (flags, tmark) предназначены только для чтения и отображают дополнительную информацию, которая обновляется при приеме данных через LTR_Recv() или в других случаях.		
Поле	Тип	Описание поля
saddr	DWORD	IP-адрес машины, на которой запущена служба ltrd, в 32-битном формате . По умолчанию устанавливается в LTRD_ADDR_DEFAULT равный LTRD_ADDR_LOCAL , что соответствует случаю, когда служба запущена на той же машине, что и пользовательская программа.
sport	WORD	Номер TCP порта, который будет использоваться при подключении к службе ltrd. По умолчанию используется LTRD_PORT_DEFAULT .
csn	CHAR [LTR_CRATE_SERIAL_SIZE]	Серийный номер крейта, которому должно соответствовать соединение. Если задана пустая строка (как по умолчанию), то будет установлено соединение с первым крейтом из текущего списка активных крейтов ltrd. В таком случае после установки соединения это поле будет изменено на реальный серийный номер крейта, с которым установлено соединение. Если записана строка LTR_CSN_SERVER_CONTROL , то будет установлено управляющее соединение со службой ltrd, которое не связано ни с одним крейтом и может быть установлено при отсутствии активных крейтов.

cc	WORD	Тип канала для соединения с ltrd. Задается одной из констант из en_LTR_CC_ChNum . Указывает, является ли данное соединение управляющим (LTR_CC_CHNUM_CONTROL) или это соединение с модулем в заданном слоте (LTR_CC_CHNUM_MODULE1 .. LTR_CC_CHNUM_MODULE16). Может также в редких случаях при необходимости объединяться с флагами en_LTR_CC_Iface и en_LTR_CC_Flags для указания дополнительных параметров соединения.
flags	DWORD	Флаги состояния соединения. Объединенный через логическое ИЛИ набор значений из en_LTR_ChStateFlags . Данное поле предназначено только для чтения и не должно явно изменяться пользователем.
tmark	DWORD	Последнее значение синхрометок, принятое для данного соединения. Данное поле обновляется при выполнении LTR_Recv() , если во время приема были обнаружены синхрометки. Данное поле предназначено только для чтения пользователем и не должно явно изменяться пользователем.
Internal	LPVOID	Непрозрачный указатель на структуру с параметрами, необходимыми для обеспечения обмена. Пользователь никак не должен использовать данное поле.

5.2.2 Конфигурация линий разъема синхронизации.

Тип: TLTR_CONFIG		
Описание: Данная структура используется для настройки функций выходов DIGOUT на разъеме синхронизации SYNC крейтов LTR-EU, LTR-CEU и LTR-CU. Также данная структура позволяет настроить функции специальных пользовательских ножек процессора BlackFin для крейтов LTR-EU, однако последняя функция нужна только при написании пользовательской прошивки.		
Поле	Тип	Описание поля
userio	WORD [4]	Настройка функций пользовательских ножек процессора. Каждый элемент соответствует своей ножке userio0 - userio3 и задает ее режим одним из значений en_LTR_UserIoCfg . (для LTR-EU userio0 — PF1 (ревизия платы 0) или PF0 (в ревизии 1+), userio1 — PG13, userio2 — PF3 (только ревизия 1+), userio3 — резерв). В штатной прошивке крейта данные настройки не оказывают влияние на работу и должны быть установлены в значение LTR_USERIO_DEFAULT .
digout	WORD [2]	Настройка функций выходов DIGOUT на разъеме синхронизации крейта. Каждый элемент массива задает режим соответствующего выхода (элемент 0 — DIGOUT1, а 1 — DIGOUT2) одним из значений из en_LTR_DigOutCfg . Кроме того, для использования этих функций выходы должны быть разрешены через digout_en.
digout_en	WORD	Разрешение выходов DIGOUT1 и DIGOUT2 на разъеме синхронизации крейта (0 — запрещены, 1 — разрешены). Все выходные линии разрешаются или запрещаются одновременно. Если выходы разрешены, то на них будет сформирован сигнал в соответствии с настроенной функцией, указанной в соответствующем элементе массива digout. Если запрещены, то выходы в третьем состоянии, независимо от настроенных функций в массиве digout.

5.2.3 Информация о типе и интерфейсе подключения крейта.

Тип: TLTR_CRATE_INFO		
Описание: Данная структура заполняется функцией LTR_GetCrateInfo() и содержит информацию о типе крейта и интерфейсе его подключения.		
Поле	Тип	Описание поля
CrateType	BYTE	Тип крейта — значение из en_LTR_CrateTypes
CrateInterface	BYTE	Интерфейс подключения крейта — значение из en_LTR_CrateIface

5.2.4 Запись с IP-адресом крейта

Тип: TLTR_CRATE_IP_ENTRY		
Описание: Данная структура содержит информацию о записи об IP-адресе крейта, сохраненную в настройках службы ltrd, а также состояние подключения крейта по интерфейсу Ethernet (TCP/IP), соответствующего данной записи		
Поле	Тип	Описание поля
ip_addr	DWORD	IP-адрес крейта. Формат аналогичен полю saddr в TLTR
flags	DWORD	Набор флагов, связанных с данной записью, из en_LTR_CrateIpFlags
serial_number	CHAR [LTR_CRATE_SERIAL_SIZE]	Если крейт подключен, то в данном поле содержится серийный номер подключенного крейта. Этот номер можно использовать для открытия соединения с крейтом. Для остальных значений status данное поле содержит пустую строку, т.к. серийный номер неизвестен
is_dynamic	BYTE	Резервное поле. Всегда равно 0
status	BYTE	Состояние подключения крейта, соответствующего данной записи. Одно значение из en_LTR_CrateIpStatus

5.2.5 Статистика крейта

Тип: TLTR_CRATE_STATISTIC		
Описание: Структура содержит информацию о состоянии крейта и параметры статистики работы с крейтом, которую ведет служба ltrd. Данная статистика может быть получена через управляющее соединение с помощью функции LTR_GetCrateStatistic(). Статистика ведется с момента установления связи между ltrd и крейтом.		
Поле	Тип	Описание поля
size	DWORD	Размер всех действительных полей структуры, включая само поле size
flags	DWORD	Флаги — резерв
crate_type	WORD	Тип крейта из en_LTR_CrateTypes
crate_intf	WORD	Интерфейс, по которому подключен крейт из en_LTR_CrateIface
crate_state	WORD	Резерв
crate_mode	WORD	Режим работы крейта из en_LTR_CrateMode

con_time	ULONGLONG	Время установления соединения службы с крейтом (формат unixtime)
res	WORD [11]	Резерв
modules_cnt	WORD	Количество слотов в данном типе крейта
mids	WORD [LTR_MODULES_PER_CRATE_MAX]	ID-модулей для всех слотов крейта
res2	WORD [3 *LTR_MODULES_PER_CRATE_MAX]	Резерв
ctl_clients_cnt	WORD	Количество клиентов, подключенных по управляющему каналу к крейту
total_mod_clients_cnt	WORD	Количество клиентов, подключенных ко всем модулям крейта
res3	DWORD [11]	Резерв
wrд_sent	ULONGLONG	Общее количество слов, переданных в крейт (крейту и всем его модулям)
wrд_recv	ULONGLONG	Общее количество слов, принятое от крейта (от самого крейта и всех его модулей)
bw_send	double	Текущая скорость передачи слов в крейт (слов/с)
bw_recv	double	Текущая скорость приема слов из крейта (слов/с)
crate_wrd_recv	ULONGLONG	Количество принятых слов непосредственно от крейта
internal_rbuf_miss	ULONGLONG	Количество потерянных буферов в крейте из-за внутреннего переполнения
internal_rbuf_ovfls	DWORD	Количество переполнений внутреннего буфера крейта
rbuf_ovfls	DWORD	Количество переполнений буфера прима данных от модулей в службе ltrd для модулей крейта (суммарное по всем модулям)
total_start_marks	DWORD	Количество принятых меток "Старт" как от крейта, так и от модулей
total_sec_marks	DWORD	Количество принятых секундных меток как от крейта, так и от модулей

crate_start_marks	DWORD	Количество принятых меток “Старт” непосредственно от крейта
crate_sec_marks	DWORD	Количество принятых секундных меток непосредственно от крейта
crate_unixtime	ULONGLONG	Последнее значение расширенной секундной метки (формат unixtime), если поддерживается крейтом
therm_mask	DWORD	Маска действительных показаний термометров (если не поддерживается — 0)
therm_vals	float [LTR_CRATE_THERM_MAX_CNT]	значение показаний термометров крейта. действительны только если соответствующий бит в therm_mask в 1
res4	DWORD [19]	Резерв

5.2.6 Статистика модуля

Тип: TLTR_MODULE_STATISTIC		
Описание: Структура содержит информацию о состоянии модуля и параметры статистики работы с данным модулем, которую ведет служба ltrd. Данная статистика может быть получена через управляющее соединение с помощью функции LTR_GetModuleStatistic() . Статистика ведется с момента обнаружения модуля и сбрасывается вместе со сбросом модуля через LTR_ResetModule()		
Поле	Тип	Описание поля
size	DWORD	Размер всех действительных полей структуры, включая само поле size
client_cnt	WORD	Количество клиентов, установивших соединение с модулем
mid	WORD	Идентификатор модуля из en_LTR_MIDs
flags	DWORD	Набор флагов, описывающих особенности модуля, из en_LTR_ModuleDescrFlags
name	CHAR [LTR_MODULE_NAME_SIZE]	Строка с названием модуля (возможно с модификациями, если ltrd способен их определить)
res	DWORD [5]	Резерв
wrд_sent	ULONGLONG	Количество слов, переданных модулю

wrd_recv	ULONGLONG	Количество слов, принятых от модуля
bw_send	double	Текущая скорость передачи слов в модуль (слов/с)
bw_recv	double	Текущая скорость приема слов из модуля (слов/с)
wrd_sent_to_client	ULONGLONG	Количество слов, переданных клиенту
wrd_recv_from_client	ULONGLONG	Количество слов, принятых от клиента
wrd_recv_drop	ULONGLONG	Количество отброшенных слов из-за переполнения буфера на прием в службе ltrd
rbuf_ovfls	DWORD	Количество переполнений буфера на прием в службе ltrd
send_srvbuf_size	DWORD	Размер буфера в ltrd для модуля на передачу
recv_srvbuf_size	DWORD	Размер буфера в ltrd для модуля на прием
send_srvbuf_full	DWORD	На сколько слов заполнен буфер на передачу
recv_srvbuf_full	DWORD	На сколько слов заполнен буфер на прием
send_srvbuf_full_max	DWORD	На сколько слов был максимально заполнен буфер на передачу
recv_srvbuf_full_max	DWORD	На сколько слов был максимально заполнен буфер на прием
res2	DWORD [17]	Резерв
start_mark	DWORD	Количество принятых меток "СТАРТ" от модуля
sec_mark	DWORD	Количество принятых секундных меток от модуля
hard_send_fifo_size	DWORD	Размер аппаратной очереди внутри модуля. Данное поле, а также все остальные поля hard_send_... действительны только для модулей вывода с наличием контролируемой ltrd очереди в модуле (для этих модулей также установлен соответствующий флаг в поле flags)

hard_send_fifo_unack_words	DWORD	Заполненность аппаратной очереди (количество отправленных, но не подтвержденных слов)
hard_send_fifo_underrun	DWORD	Количество “голоданий” очереди (очередь пуста при попытке вывода значения из нее) с момента последнего сброса модуля
hard_send_fifo_overrun	DWORD	Количество переполнений очереди с момента последнего сброса модуля
hard_send_fifo_internal	DWORD	Внутренне состояние аппаратной очереди
res3	DWORD [25]	Резерв

5.2.7 Информация о крейте и его прошивке

Тип: TLTR_CRATE_DESCR		
Описание: Структура содержит информацию о крейте, включая все версии, связанные с крейтом. Большинство полей представлены в виде оканчивающейся нулевым символом строки. Данная информация может быть получена Данная статистика может быть получена через управляющее соединение с помощью LTR_GetCrateDescr() .		
Поле	Тип	Описание поля
size	DWORD	Размер всех действительных полей структуры, включая само поле size
devname	char [LTR_CRATE_DEVNAME_SIZE]	Название крейта
serial	char [LTR_CRATE_SERIAL_SIZE]	Серийный номер
soft_ver	char [LTR_CRATE_SOFTVER_SIZE]	Версия прошивки
brd_revision	char [LTR_CRATE_REVISION_SIZE]	Ревизия платы
brd_opts	char [LTR_CRATE_BOARD_OPTIONS_SIZE]	Опции платы
bootloader_ver	char [LTR_CRATE_BOOTVER_SIZE]	Версия загрузчика
cpu_type	char [LTR_CRATE_CPU TYPE_SIZE]	Тип микроконтроллера
fpga_name	char [LTR_CRATE_FPGA_NAME_SIZE]	Название ПЛИС в крейте
fpga_version	char [LTR_CRATE_FPGA_VERSION_SIZE]	Версия прошивки ПЛИС
crate_type_name	char [LTR_CRATE_TYPE_NAME]	Строка с типом крейта

spec_info	char [LTR_CRATE_SPECINFO_SIZE]	Резерв
protocol_ver_major	BYTE	Версия протокола между ltrd и крейтом (мажорная)
protocol_ver_minor	BYTE	Версия протокола между ltrd и крейтом (минорная)

5.3 Функции

5.3.1 Функции инициализации и работы с соединением

5.3.1.1 Инициализация описателя соединения

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

5.3.1.2 Открытие соединения

Формат: INT LTR_Open (TLTR *hnd)
Описание: Функция устанавливает клиентское соединение в соответствии с установленными полями saddr , sport , csn и cc описателя модуля. О типах соединений подробнее описано в главе Типы клиентских соединений После завершения работы необходимо закрыть соединение с помощью LTR_Close() . Если функция возвращает ошибку, то в некоторых случаях соединение может оставаться открытым (например, для ошибки LTR_WARNING_MODULE_IN_USE), поэтому даже в случае возвращения данной функцией ошибки, все равно следует вызывать LTR_Close() . Все возвращенные ошибки (включая LTR_WARNING_MODULE_IN_USE) рекомендуется интерпретировать как признак того, что с соединением нельзя работать и единственной допустимой функцией, которую можно и нужно вызвать следующей, является LTR_Close() .
Параметры: hnd — Описатель соединения.
Возвращаемое значение: Код ошибки .

5.3.1.3 Открытие управляющего соединения со службой ltrd.

Формат: INT LTR_OpenSvcControl (TLTR *hsrv, DWORD ltrd_addr, WORD ltrd_port)
Описание: <p>Данная функция устанавливает управляющее соединение со службой ltrd. Данное соединение может быть установлено даже когда нет ни одного подключенного крейта и позволяет выполнять команды, управляющие работой службы (за исключением команд управления крейтами, которые требуют открытия управляющего соединения с конкретным крейтом).</p> <p>По завершению работы с соединением необходимо его закрыть, вызвав LTR_Close().</p> <p>Действие функции аналогично правильному заполнению полей структуры TLTR с использованием в качестве серийного номера строки LTR_CSN_SERVER_CONTROL и вызову LTR_Open() и служит для удобства, чтобы не заполнять поля вручную.</p> <p>Тоже можно выполнить и заполнив правильно поля структуры TLTR и вызвав LTR_Open(). Данная функция введена для удобства, чтобы не заполнять поля вручную.</p> <p>Функция доступна в ltrapi версии 1.31.0 или выше.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>ltrd_addr — IP-адрес машины, на которой запущена служба ltrd, в 32-битном формате. Для использования значения по умолчанию может быть передано значение LTRD_ADDR_DEFAULT.</p> <p>ltrd_port — Номер TCP порта, который будет использоваться при подключении к службе ltrd. Для использования порта по умолчанию может быть передано значение LTRD_PORT_DEFAULT.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.1.4 Открытие управляющего соединения с крейтом

Формат: INT LTR_OpenCrate (TLTR *hcrate, DWORD ltrd_addr, WORD ltrd_port, INT crate_iface, const char *crate_sn)
Описание: <p>Данная функция устанавливает управляющее соединение с крейтом по его серийному номеру (или с первым крейтом, если серийный номер не задан).</p> <p>Функция позволяет явно указать интерфейс, по которому установлено соединение между службой ltrd и крейтом на случай, если крейт подключен одновременно по двум интерфейсам (например в случае, если крейт настроен и подключен по Ethernet, но также и подключен по USB в режиме настройки). В большинстве случаев достаточно указать в качестве интерфейса <code>LTR_CRATE_IFACE_UNKNOWN</code>, что означает, что крейт может быть подключен по любому интерфейсу. При этом, если крейт подключен по двум интерфейсам, то открытое соединение будет связано с крейтом с использованием интерфейса, который является рабочим для крейта, т.е. через который можно обмениваться данными с модулями крейта, что и необходимо для большинства программ.</p> <p>По завершению работы с соединением необходимо его закрыть, вызвав <code>LTR_Close()</code>.</p> <p>Действие функции аналогично правильному заполнению полей структуры <code>TLTR</code> и вызову <code>LTR_Open()</code>. Данная функция введена для удобства, чтобы не заполнять поля вручную.</p> <p>Функция доступна в ltrapi версии 1.31.0 или выше.</p>
Параметры: <p>hcrate — Описатель соединения.</p> <p>ltrd_addr — IP-адрес машины, на которой запущена служба ltrd, в 32-битном формате. Для использования значения по умолчанию может быть передано значение <code>LTRD_ADDR_DEFAULT</code>.</p> <p>ltrd_port — Номер TCP порта, который будет использоваться при подключении к службе ltrd. Для использования порта по умолчанию может быть передано значение <code>LTRD_PORT_DEFAULT</code>.</p> <p>crate_iface — Значение из <code>en_LTR_CrateIface</code>, указывающая интерфейс, по которому должен быть подключен крейт. Если задано значение <code>LTR_CRATE_IFACE_UNKNOWN</code>, то крейт может быть подключен по любому интерфейсу.</p> <p>crate_sn — Строка с серийным номером крейта, с которым нужно установить соединение. Если передана пустая строка, то будет установлено соединение с первым подключенным крейтом</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.1.5 Открытие соединения с заданным таймаутом

Формат: INT LTR_OpenEx (TLTR *hnd, DWORD timeout)
Описание: Данная функция полностью аналогична LTR_Open() , но позволяет явно указать максимальное время установления соединения с ltrd.
Параметры: hnd — Описатель соединения. timeout — Время в мс на открытие соединения. Если за заданное время не завершится соединение с ltrd, то функция завершится с ошибкой.
Возвращаемое значение: Код ошибки .

5.3.1.6 Закрытие соединения

Формат: INT LTR_Close (TLTR *hnd)
Описание: Функция закрывает ранее открытое с помощью функций LTR_Open() , LTR_OpenEx() , LTR_OpenCrate() или LTR_OpenSvcControl() соединение. При любом возвращенном значении после вызова этой функции соответствующий описатель соединения уже нельзя использовать без открытия нового соединения.
Параметры: hnd — Описатель соединения.
Возвращаемое значение: Код ошибки .

5.3.1.7 Проверка, открыто ли соединение

Формат: INT LTR_IsOpened (TLTR *hnd)
Описание: Функция проверяет, открыто ли в данный момент клиентом соединение, соответствующее указанному описателю. Если соединение открыто, функция возвращает LTR_OK , если закрыто — код ошибки LTR_ERROR_CHANNEL_CLOSED . Данная функция не проверяет каким-либо образом, действительно ли сейчас это соединение, а результат ее определяется только последовательностью вызова функций — было ли успешно открыто и не закрыто данное соединение клиентом.
Параметры: hnd — Описатель соединения.
Возвращаемое значение: Код ошибки (LTR_OK , если соединение установлено).

5.3.2 Функции информационного характера

Набор функций, позволяющих получить общую информацию о службе ltrd, подключенных крестях и статистике обмена. Эти функции могут быть использованы с любым управляющим соединением.

5.3.2.1 Получение версии службы ltrd.

Формат: INT LTR_GetServerVersion (TLTR *hsrv, DWORD *version)
Описание: Функция возвращает номер версии службы ltrd, с которой установлено управляющее соединение. В текстовом виде версия ltrd состоит из четырех чисел, разделенных точками. Возвращаемое данной функцией число представляет собой 32-битное значение, каждый байт которого соответствует одной части номера версии, отделяемой в текстовой записи точкой, причем старший байт соответствует главному номеру (первому числу). Например, 0x02010403 соответствует версии 2.1.4.3. Версии службы ltrd начинаются с 2.0.0.0 и выше, в то время как версии со старшим числом 1 возвращались только ранее используемой программой LTR Server.
Параметры: hsrv — Описатель управляющего соединения. version — В данной переменной возвращается версия службы ltrd в формате, описанном выше.
Возвращаемое значение: Код ошибки .

5.3.2.2 Получение серийных номеров подключенных крейтов

Формат: INT LTR_GetCrates (TLTR *hsrv, BYTE *csn)
Описание: <p>Функция возвращает список серийных номеров крейтов, с которыми установлено соединение службой ltrd (список активных крейтов). Полученные серийные номера можно использовать для установления управляющего соединения с крейтами (для их конфигурации, получения списка модулей и т.п.) или соединений с модулями.</p> <p>Данная функция может вернуть максимум LTR_CRATES_MAX серийных номеров (если подключено крейтов больше, то будет возвращено только первые LTR_CRATES_MAX). Если крейтов может быть больше LTR_CRATES_MAX, то можно использовать функцию LTR_GetCratesEx(), в которой нет ограничения на количество возвращаемых номеров крейтов.</p> <p>Функция возвращает всегда разные серийные номера, т.е. если крейт подключен и по USB (в режиме настройки) и по Ethernet (в рабочем режиме), то функция заполнит только один элемент в выходном массиве.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>csn — Двумерный массив размером [LTR_CRATES_MAX] [LTR_CRATE_SERIAL_SIZE] байт. В данный массив будут сохранены серийные номера подключенных крейтов (каждая строка соответствует своему серийному номеру крейта). Всегда заполняются все LTR_CRATES_MAX серийных номеров. Если крейтов меньше LTR_CRATES_MAX, то на месте серийных номеров после последнего действительного серийного номера будет записана пустая строка (один символ с нулевым кодом).</p>
Возвращаемое значение: Код ошибки

5.3.2.3 Получение информации о подключенных крейтах

Формат: INT LTR_GetCratesEx (TLTR *hsrv, DWORD max_crates, DWORD flags, DWORD *crates_found, DWORD *crates_returned, CHAR serials[] [LTR_CRATE_SERIAL_SIZE], TLTR_CRATE_INFO *info_list)
Описание: <p>Данная функция позволяет получить список серийных номеров крейтов, с которыми установлено соединение службой ltrd (список активных крейтов) с дополнительной информацией о этих крейтах.</p> <p>В отличие от LTR_GetCrates() функция не ограничивает количество подключенных крейтов, информация о которых может быть возвращена.</p> <p>Также в отличие от LTR_GetCrates(), если один крейт подключен сразу по нескольким интерфейсам, то данная функция вернет две записи о крейте, т.е. его серийный номер будет указан дважды в массиве serials, но при этом в соответствующих записях с информацией о крейте будет указан разный интерфейс подключения (может быть изменено флагами).</p> <p>Функцию можно использовать с нулевым значением max_crates для получения количества подключенных крейтов, чтобы затем создать массивы нужных размеров и вызвать функцию второй раз уже для получения информации о всех крейтах.</p> <p>Данная функция поддерживается только службой ltrd. Функция доступна в ltrapi версии 1.31.0 или выше.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>max_crates — Максимальное количество крейтов, информация о которых может быть возвращена функцией.</p> <p>flags — Флаги из en_LTR_GetCratesFlags, управляющие работой функции.</p> <p>crates_found — В данном параметре возвращается общее количество подключенных крейтов. Это значение может быть больше max_crates. Если данная информация не нужна, то может быть передан нулевой указатель.</p> <p>crates_returned — В данной переменной возвращается количество действительных серийных номеров, записанных в массив serials. Данное возвращенное значение не может быть больше значения, переданного в параметре max_crates. Если max_crates равно нулю, то в качестве данного параметра можно передать нулевой указатель.</p> <p>serials — Массив для сохранения серийных номеров подключенных крейтов. Должен быть достаточного размера для сохранения max_crates серийных номеров (каждый серийный номер — строка из LTR_CRATE_SERIAL_SIZE символов). По завершению функции будут заполнены первые crates_returned серийных номеров. Если max_crates равно нулю, то в качестве данного параметра может быть передан нулевой указатель.</p> <p>info_list — Массив для сохранения информации о подключенных крейтах. Должен быть достаточного размера для сохранения max_crates структур TLTR_CRATE_INFO. Каждый элемент соответствует серийному номеру из serials с тем же номером. Если информация о крейтах не нужна, то может быть передан нулевой указатель.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.2.4 Получение описания крейта

Формат: INT LTR_GetCrateDescr (TLTR *hsrv, INT crate_iface, const char *crate_sn, TLTR_CRATE_DESCR *descr, DWORD size)
Описание: <p>Функция позволяет получить структуру типа TLTR_CRATE_DESCR с подробным описанием заданного крейта. Может быть получено описание любого подключенного крейта через одно управляющее соединение — отдельное управляющее соединение с крейтом устанавливать не нужно.</p> <p>Данная функция поддерживается только службой ltrd (не поддерживается ранее используемой программой LTR Server).</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>crate_iface — Интерфейс подключения крейта (аналогично одноименному параметру LTR_OpenCrate()).</p> <p>crate_sn — Серийный номер крейта (аналогично одноименному параметру LTR_OpenCrate()).</p> <p>descr — Структура, в которую будет сохранено описание крейта</p> <p>size — Размер структуры, переданной в качестве параметра descr.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.2.5 Получение статистики по крейту

Формат: INT LTR_GetCrateStatistic (TLTR *hsrv, INT crate_iface, const char *crate_sn, TLTR_CRATE_STATISTIC *stat, DWORD size)
Описание: <p>Функция возвращает дополнительные параметры статистики, которую ведет служба ltrd, связанные с заданным крейтом, в виде структуры TLTR_CRATE_STATISTIC.</p> <p>Данная функция поддерживается только службой ltrd (не поддерживается ранее используемой программой LTR Server).</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>crate_iface — Интерфейс подключения крейта (аналогично одноименному параметру LTR_OpenCrate()).</p> <p>crate_sn — Серийный номер крейта (аналогично одноименному параметру LTR_OpenCrate()).</p> <p>stat — Структура, в которую будут сохранены параметры статистики по крейту.</p> <p>size — Размер структуры, переданной в качестве параметра stat.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.2.6 Получение статистики по модулю

Формат: INT LTR_GetModuleStatistic (TLTR *hsrv, INT crate_iface, const char *crate_sn, INT module_slot, TLTR_MODULE_STATISTIC *stat, DWORD size)
Описание: Функция возвращает дополнительные параметры статистики, которую ведет служба ltrd, связанные с заданным модулем, в виде структуры TLTR_MODULE_STATISTIC . Данная функция поддерживается только службой ltrd (не поддерживается ранее используемой программой LTR Server).
Параметры: hsrv — Описатель управляющего соединения. crate_iface — Интерфейс подключения крейта (аналогично одноименному параметру LTR_OpenCrate()). crate_sn — Серийный номер крейта (аналогично одноименному параметру LTR_OpenCrate()). module_slot — Номер слота модуля (от LTR_CC_CHNUM_MODULE1 до LTR_CC_CHNUM_MODULE16). stat — Структура, в которую будут сохранены параметры статистики по крейту. size — Размер структуры, переданной в качестве параметра stat.
Возвращаемое значение: Код ошибки .

5.3.3 Функции управления крейтами

Набор функций, реализующих общие команды управления крейтами LTR. Эти функции могут использоваться только с [управляющим соединением с крейтом](#).

5.3.3.1 Получение списка модулей в крейте

Формат: INT LTR_GetCrateModules (TLTR *hcrate, WORD *mid)
Описание: Функция позволяет получить список идентификаторов модулей, которые установлены в крейте. На вход функции необходимо подать массив из LTR_MODULES_PER_CRATE_MAX элементов, который функция заполнит нужными значениями. Каждый элемент соответствует своему слоту в крейте (mid[0] — идентификатор модуля, вставленного в первый слот, mid[15] — в 16-ый) и при успешном выполнении функции устанавливается в одно из значений из en_LTR_MIDs . Если в данном слоте не вставлен модуль или слот вообще отсутствует в крейте, то будет установлено значение LTR_MID_EMPTY . Если модуль был обнаружен, но его тип еще не определен, то будет возвращено значение LTR_MID_IDENTIFYING .
Параметры: hcrate — Описатель управляющего соединения с крейтом. mid — Указатель на массив из LTR_MODULES_PER_CRATE_MAX элементов, в котором при успешном выполнении будут возвращены идентификаторы установленных модулей.
Возвращаемое значение: Код ошибки .

5.3.3.2 Получение информации о типе и интерфейсе подключения крейта

Формат: INT LTR_GetCrateInfo (TLTR *hcrate, TLTR_CRATE_INFO *CrateInfo)
Описание: Функция заполняет структуру TLTR_CRATE_INFO информацией о крейте, с которым установлено управляющее соединение.
Параметры: hcrate — Описатель управляющего соединения с крейтом. CrateInfo — В данной структуре при успехе возвращается информация о крейте.
Возвращаемое значение: Код ошибки .

5.3.3.3 Конфигурация линий разъема синхронизации крейта

Формат: INT LTR_Config (TLTR *hcrate, const TLTR_CONFIG *conf)
Описание: Функция устанавливает конфигурацию линий разъема синхронизации SYNC в соответствии с параметрами, заданными в структуре TLTR_CONFIG . Данная функция применима только для крейтов с наличием данного разъема (LTR-EU, LTR-CU, LTR-CEU).
Параметры: hcrate — Описатель управляющего соединения с крейтом. conf — Конфигурация линий разъема синхронизации крейта.
Возвращаемое значение: Код ошибки .

5.3.3.4 Настройка генерации метки “СТАРТ”.

Формат: INT LTR_MakeStartMark (TLTR *hcrate, INT mode)
Описание: Функция устанавливает режим генерации метки “СТАРТ” крейтом. Данная функция работает только с крейтами, поддерживающими стандартный механизм генерации синхрометок (LTR-EU, LTR-CEU, LTR-CU). Крейт может генерировать метку как от внешнего события, так и внутреннего — по команде с ПК. В зависимости от значения режима поведение функции несколько отличается. Если задан режим LTR_MARK_INTERNAL , то при выполнении этой функции крейт генерирует одиночную метку “СТАРТ”, после чего не генерирует метки до следующего вызова данной функции. Т.е. при внутренней генерации метки “СТАРТ” необходимо вызывать данную функцию каждый раз, когда метка должна быть сгенерирована. Остальные режимы соответствуют внешней генерации метки. В них данная функция просто настраивает крейт, чтобы он ожидал заданного в режиме события и по каждому такому событию генерировал метку. Чтобы отключить внешнюю генерацию меток, можно вызвать данную функцию с указанием режима LTR_MARK_OFF . В случае закрытия соединения с крейтом без отключения генерации меток, крейт все равно будет продолжать генерацию до явного отключения.
Параметры: hcrate — Описатель управляющего соединения с крейтом. mode — Режим генерации метки “СТАРТ” — значение из en_LTR_MarkMode .
Возвращаемое значение: Код ошибки .

5.3.3.5 Запуск генерации меток “СЕКУНДА”.

Формат: INT LTR_StartSecondMark (TLTR *hcrate, INT mode)
Описание: <p>Функция запускает генерацию секундных меток в указанном режиме. Данная функция работает только с крейтами, поддерживающими стандартный механизм генерации синхрометок (LTR-EU, LTR-CEU, LTR-CU).</p> <p>Если задан режим LTR_MARK_INTERNAL, то после вызова данной функции крейт начинает генерировать секундную метку раз в секунду (от внутреннего таймера). В остальных режимах крейт ожидает внешнего события и генерирует секундную метку по каждому событию.</p> <p>Останов генерации выполняется с помощью функции LTR_StopSecondMark().</p> <p>В случае закрытия соединения с крейтом без отключения генерации секундных меток, крейт все равно будет продолжать генерацию до явного отключения.</p>
Параметры: <p>hcrate — Описатель управляющего соединения с крейтом. mode — Режим генерации меток “СЕКУНДА”.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.3.6 Останов генерации меток “СЕКУНДА”.

Формат: INT LTR_StopSecondMark (TLTR *hcrate)
Описание: <p>При вызове данной функции крейт прекращает генерацию меток секунда, которая была запущена ранее с помощью LTR_StartSecondMark()</p>
Параметры: <p>hcrate — Описатель управляющего соединения с крейтом.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.4 Функции управления службой ltrd

Набор дополнительных функций управляющего соединения, связанные с управлением работой службы ltrd.

5.3.4.1 Сброс указанного модуля.

Формат: INT LTR_ResetModule (TLTR *hsrv, INT crate_iface, const char *crate_sn, INT module_slot, DWORD flags)
Описание: <p>Данная функция позволяет сбросить любой модуль через управляющее соединение. По выполнению этой функции в службе выполняются следующие действия:</p> <ul style="list-style-type: none">• разрываются все клиентские соединения, установленные с данным модулем• сбрасывается вся статистика, накопленная ltrd по данному модулю• посылается команда сброса модуля для возвращения модуля в исходное состояние <p>Данная функция может быть полезна в случае, если осталось незакрытое клиентское соединение с модулем, которое не дает установить новое рабочее соединение (возвращается ошибка LTR_WARNING_MODULE_IN_USE). Вызов этой функции позволяет сбросить не закрытое соединение.</p> <p>Также функция может быть полезна, если программа закрылась не завершив корректно работу с модулем, например, оставив запущенным сбор данных. Сброс модуля позволяет остановить передачу от модуля ненужных данных, переведя его в начальное состояние.</p> <p>Данная функция поддерживается только службой ltrd (не поддерживается ранее используемой программой LTR Server).</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>crate_iface — Интерфейс подключения крейта (аналогично одноименному параметру LTR_OpenCrate()).</p> <p>crate_sn — Серийный номер крейта (аналогично одноименному параметру LTR_OpenCrate()).</p> <p>module_slot — Номер слота модуля (от LTR_CC_CHNUM_MODULE1 до LTR_CC_CHNUM_MODULE16).</p> <p>flags — Дополнительные флаги, управляющие работой функции. Сейчас не используются, поэтому поле всегда должно быть равно нулю.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.4.2 Установка уровня журнала

Формат: INT LTR_SetLogLevel (TLTR *hsrv, INT level, BOOL permanent)
Описание: Функция устанавливает, какого уровня сообщения будут выводиться в журнал службы ltrd. Все сообщения с менее важным уровнем выводиться не будут.
Параметры: hsrv — Описатель управляющего соединения. level — Устанавливаемый уровень вывода в журнал — значение из en_LTR_LogLevel . permanent — Если FALSE, то изменения касаются только текущего запуска службы ltrd. Если TRUE — изменения сохраняются в настройках и будут учитываться после перезапуска службы.
Возвращаемое значение: Код ошибки .

5.3.4.3 Получение текущего уровня журнала

Формат: INT LTR_GetLogLevel (TLTR *hsrv, INT *level)
Описание: Функция возвращает установленный уровень вывода в журнал службы ltrd.
Параметры: hsrv — Описатель управляющего соединения. level — В данной переменной возвращается установленный уровень журнала — значение из en_LTR_LogLevel .
Возвращаемое значение: Код ошибки .

5.3.4.4 Установка параметров работы службы ltrd.

Формат: INT LTR_SetServerParameter (TLTR *hsrv, DWORD param, void *val, DWORD size)
Описание: <p>Функция выполняет установку одного параметра из настроек работы службы ltrd. Формат и смысл передаваемого значения определяется кодом параметра и описан в описании каждого параметра. Устанавливаемый параметр немедленно применяется и сохраняется в настройках службы ltrd.</p> <p>Данная функция поддерживается только службой ltrd (не поддерживается ранее используемой программой LTR Server).</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>param — Код параметра — значение из en_LTRD_Params.</p> <p>val — Указатель на данные с устанавливаемым значением. Формат данных может зависеть от параметра и указан в описании параметра.</p> <p>size — Размер данных, передаваемых в качестве значения параметра (на которые указывает val).</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.4.5 Чтение параметров работы службы ltrd.

Формат: INT LTR_GetServerParameter (TLTR *hsrv, DWORD param, void *val, DWORD *size)
Описание: <p>Функция выполняет чтение одного параметра из настроек работы службы ltrd. Формат и смысл передаваемого значения определяется кодом параметра и описан в описании каждого параметра.</p> <p>Данная функция поддерживается только службой ltrd (не поддерживается ранее используемой программой LTR Server).</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>param — Код параметра — значение из en_LTRD_Params</p> <p>val — Указатель на данные, куда будет сохранено значение параметра. Формат данных может зависеть от параметра и указан в описании параметра.</p> <p>size — Размер массива, на который указывает переменная val. Разные параметры могут требовать разного размера для сохранения значения.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.4.6 Перезапуск службы ltrd.

Формат: INT LTR_ServerRestart (TLTR *hsrv)
Описание: При успешном выполнении данной команды служба ltrd закрывает все клиентские соединения и все соединения с крейтами, и начинает работу сначала, заново считывая свои настройки. Соответственно, и текущее управляющее соединение, через которое передана команда, становится недействительным — единственной функцией, которая может и должна быть вызвана следующей, является LTR_Close() .
Параметры: hsrv — Описатель управляющего соединения.
Возвращаемое значение: Код ошибки .

5.3.4.7 Останов службы ltrd.

Формат: INT LTR_ServerShutdown (TLTR *hsrv)
Описание: При успешном выполнении данной команды служба ltrd закрывает все клиентские соединения и все соединения с крейтами и завершает свою работу. Соответственно, и текущее управляющее соединение, через которое передана команда, становится недействительным — единственной функцией, которая может и должна быть вызвана следующей, является LTR_Close() .
Параметры: hsrv — Описатель управляющего соединения.
Возвращаемое значение: Код ошибки .

5.3.5 Функции управления подключением крейтов по Ethernet

Данные функции используются с управляющим соединением. Они позволяют управлять списком записей с IP-адресами крейтов службы ltrd, а также выполнять команды для установления и закрытия соединения между ltrd и крейтами по IP-адресам с использованием записей из этого списка.

5.3.5.1 Получение списка записей с IP-адресами крейтов

<p>Формат: INT LTR_GetListOfIPCrates (TLTR *hsrv, DWORD max_entries, DWORD ip_net, DWORD ip_mask, DWORD *entries_found, DWORD *entries_returned, TLTR_CRATE_IP_ENTRY *info_array)</p>
<p>Описание:</p> <p>Функция возвращает список записей с IP-адресами крейтов из настроек службы ltrd (как с установленным соединением с крейтом, так и не установленным). Также вместе с записью возвращается и состояние подключения крейта, соответствующего данной записи.</p> <p>Функция позволяет вернуть не весь список записей, а только тех, адреса которых удовлетворяют заданному фильтру — возвращаются только те IP адреса, которые относятся к подсети, заданной стандартным образом с помощью IP-адреса (ip_net) и маски подсети (ip_mask). Т.е. если задан адрес “192.168.1.0” и маска “255.255.255.0”, то будут возвращены записи с IP-адресами вида “192.168.1.x”, где x - любое значение от 0 до 255.</p> <p>Если нужно получить все записи, то в качестве адреса и маски можно передать нулевые значения.</p> <p>Для получения информации о состоянии конкретной записи можно задать полностью требуемый IP-адрес, а маску задать равной “255.255.255.255”</p> <p>Для получения произвольного количества IP-записей функцию можно вызвать сперва с нулевым значением max_entries, чтобы получить число записей в entries_found, а затем уже выделить массив для приема нужного количества записей и вызвать функцию повторно уже для получения информации по этим записям.</p>
<p>Параметры:</p> <p>hsrv — Описатель управляющего соединения.</p> <p>max_entries — Максимальное количество записей, которое может принять массив info_array</p> <p>ip_net — IP-адрес сети, использующийся для фильтрации возвращаемых записей, в 32-битном формате.</p> <p>ip_mask — Маска подсети для фильтрации возвращаемых записей в 32-битном формате.</p> <p>entries_found — В данной переменной возвращается общее количество найденных записей, удовлетворяющих условию фильтра. Это значение может быть больше max_entries.</p> <p>entries_returned — Количество записей, которое было возвращено в массиве info_array. При успешном выполнении это значение равно наименьшему из max_entries и entries_found.</p> <p>info_array — Массив структур TLTR_CRATE_IP_ENTRY, в котором будут возвращены найденные записи с IP-адресами. Данный массив должен быть достаточного размера для сохранения max_entries записей. Если в max_entries передано нулевое значение, то в качестве данного параметра может быть передан нулевой указатель.</p>
<p>Возвращаемое значение:</p> <p>Код ошибки.</p>

5.3.5.2 Добавление записи с IP-адресом крейта

Формат: INT LTR_AddIPCrates (TLTR *hsrv, DWORD ip_addr, DWORD flags, BOOL permanent)
Описание: <p>Функция добавляет запись с указанным параметром в список записей IP-адресов крейтов службы ltrd.</p> <p>Чтобы установить соединение с крейтом по Ethernet (с использованием LTR_ConnectIPCrates()) его IP-адрес должен быть уже в списке записей.</p> <p>Если запись уже есть в списке, то данная функция только изменяет флаги записи (аналогично LTR_SetIPCratesFlags()).</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>ip_addr — IP-адрес крейта в 32-битном формате.</p> <p>flags — Набор флагов, связанных с добавляемой записью. Объединение по ИЛИ значений из en_LTR_CrateIpFlags.</p> <p>permanent — Если FALSE, то изменения касаются только текущего запуска службы ltrd. Если TRUE — изменения сохраняются в настройках и будут учитываться после перезапуска службы.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.5.3 Удаление записи с IP-адресом крейта

Формат: INT LTR_DeleteIPCrates (TLTR *hsrv, DWORD ip_addr, BOOL permanent)
Описание: <p>Функция удаляет запись с указанным IP-адресом из списка IP-адресов крейтов службы ltrd.</p> <p>При этом с соответствующим крейтом не должно быть установлено соединение (состояние записи должно быть отлично от LTR_CRATE_IP_STATUS_ONLINE или LTR_CRATE_IP_STATUS_CONNECTING), в противном случае функция вернет ошибку. Для удаления записи с подключенным крейтом сперва нужно разорвать соединение с крейтом через LTR_DisconnectIPCrates().</p> <p>Если указанная запись отсутствует в списке, то функция никак не повлияет на список адресов и сразу завершится без ошибки.</p>
Параметры: <p>hsrv — Описатель управляющего соединения</p> <p>ip_addr — IP-адрес крейта в 32-битном формате, запись с которым должна быть удалена.</p> <p>permanent — Если FALSE, то изменения касаются только текущего запуска службы ltrd. Если TRUE — изменения сохраняются в настройках и будут учитываться после перезапуска службы.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.5.4 Установление соединения с крейтом по IP-адресу

Формат: INT LTR_ConnectIPCrates (TLTR *hsrv, DWORD ip_addr)
Описание: <p>Функция является командой службе ltrd, указывающей, что служба должна установить соединение с крейтом с заданным IP-адресом по Ethernet интерфейсу. Запись с заданным адресом должна быть в списке записей с IP-адресами крейтов службы (этот список можно получить с помощью LTR_GetListOfIPCrates()), иначе функция вернет ошибку.</p> <p>Завершение этой функции говорит о том, что служба приняла команду и начала подключение к крейту, но подключение еще может быть не завершено. При этом состояние соединения с крейтом для записи изменяется на LTR_CRATE_IP_STATUS_CONNECTING.</p> <p>По завершению подключения это состояние изменится на LTR_CRATE_IP_STATUS_ONLINE при успешном подключении или на LTR_CRATE_IP_STATUS_ERROR при ошибке, что и является признаком завершения операции. Получить текущее состояние подключения можно получив информацию о записи с помощью LTR_GetListOfIPCrates(),</p> <p>Также при успешном завершении подключения крейт появится в списке активных крейтов, который можно получить с помощью LTR_GetCrates() или LTR_GetCratesEx().</p> <p>Если при вызове данной функции крейт уже подключен (состояние подключения LTR_CRATE_IP_STATUS_ONLINE) или идет процесс подключения (состояние LTR_CRATE_IP_STATUS_CONNECTING), то функция просто завершится успешно, не выполняя никаких действий.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>ip_addr — IP-адрес крейта в 32-битном формате, с которым нужно установить соединение.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.5.5 Разрыв соединения с крейтом по IP-адресу

Формат: INT LTR_DisconnectIPCrates (TLTR *hsrv, DWORD ip_addr)
Описание: <p>При выполнении данной функции служба разрывает соединение с крейтом, подключенным по Ethernet, с указанным адресом. Крейт исчезает из списка активных крейтов и состояние подключения для записи с данным IP-адресом изменяется на LTR_CRATE_IP_STATUS_OFFLINE.</p> <p>Запись с таким адресом должна быть среди списка записей с IP-адресами крейтов службы ltrd, иначе функция вернет ошибку.</p> <p>Если запись с указанным адресом присутствует, но нет активного соединения (состояние подключение отлично от LTR_CRATE_IP_STATUS_ONLINE или LTR_CRATE_IP_STATUS_CONNECTING), то функция завершается успешно, не выполняя каких-либо действий.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>ip_addr — IP-адрес крейта в 32-битном формате, соединение с которым нужно завершить.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.5.6 Установление соединения со всеми крейтами с признаком автоподключения

Формат: INT LTR_ConnectAllAutoIPCrates (TLTR *hsrv)
Описание: <p>Функция дает команду службе ltrd установить соединение через интерфейс Ethernet со всеми крейтами, для записей с IP-адресами которых установлен флаг LTR_CRATE_IP_FLAG_AUTOCONNECT.</p> <p>При этом запуск подключения осуществляется только для тех записей, для которых нет активного соединения, т.е. на записи с текущим состоянием LTR_CRATE_IP_STATUS_ONLINE или LTR_CRATE_IP_STATUS_CONNECTING вызов функции влияния не оказывает.</p> <p>Как и в случае с LTR_ConnectIPCrates() завершение работы функции означает только то, что процесс подключения запущен, а о завершении подключения можно узнать при необходимости по изменению состояния подключения соответствующих записей (которое можно узнать с помощью LTR_GetListOfIPCrates()).</p> <p>Если нет ни одной записи с установленным флагом LTR_CRATE_IP_FLAG_AUTOCONNECT, по которой не установлено соединение с крейтом, то функция завершается успешно, не выполняя каких-либо действий.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.5.7 Разрыв соединения со всеми крейтами, подключенными по Ethernet.

Формат: INT LTR_DisconnectAllIPCrates (TLTR *hsrv)
Описание: <p>По данной команде служба ltrd закрывает все активные соединения с крейтами, подключенными через интерфейс Ethernet. Все крейты будут удалены из списка активных крейтов и все IP-записи изменят состояние подключения на LTR_CRATE_IP_STATUS_OFFLINE.</p> <p>Если на момент вызова функции не было ни одного крейта, подключенного по Ethernet, то функция завершается успешно, не выполняя каких-либо действий.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.5.8 Установка флагов для записи с IP-адресом крейта

Формат: INT LTR_SetIPCratesFlags (TLTR *hsrv, DWORD ip_addr, DWORD flags, BOOL permanent)
Описание: <p>Функция изменяет значение флагов, связанных с уже присутствующей записью с IP-адресом, на указанное значение. Запись с заданным IP-адресом должна присутствовать в списке записей службы ltrd, в противном случае функция завершится с ошибкой.</p>
Параметры: <p>hsrv — Описатель управляющего соединения.</p> <p>ip_addr — IP-адрес крейта в 32-битном формате, для записи с которым следует изменить флаги.</p> <p>flags — Новый набор флагов, связанных с записью. Объединение по ИЛИ значений из en_LTR_CrateIpFlags.</p> <p>permanent — Если FALSE, то изменения касаются только текущего запуска службы ltrd. Если TRUE — изменения сохраняются в настройках и будут учитываться после перезапуска службы.</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.6 Функции для обмена данными с модулями

Данные функции используются для передачи и приема данных модулям. Как правило пользователю нет необходимости использовать данные функции напрямую, т.к. для работы с модулями используются функции из специализированных библиотек для работы с конкретным модулем.

Эти функции могут использоваться только для [соединений с модулями](#) и неприменимы для управляющих соединений.

5.3.6.1 Прием данных от модуля

Формат: INT LTR_Recv (TLTR *hmodule, DWORD *data, DWORD *tmark, DWORD size, DWORD timeout)
Описание: <p>Функция принимает данные от модуля в 32-битных словах в специальном формате слов LTR. Кроме того, функция анализирует информацию о пришедших синхрометках и о переполнении буфера в службе ltrd, обновляя значения поля TLTR::flags описателя модуля и заполняя массив tmark.</p> <p>Функция возвращает управление либо когда примет запрошенное количество слов, либо по истечении таймаута. При этом реально принятое количество слов можно узнать по возвращенному значению.</p>
Параметры: <p>hmodule — Описатель соединения с модулем.</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 синхрометок, соответствующие принятым данным. Т.е. принятому слову data[i] соответствует элемент tmark[i], указывающий количество меток “СТАРТ” и “СЕКUNДА”, соответствующих этому слову. Формат данных слов описан в разделе, посвященном синхрометкам. Если синхрометки не используются, то можно передать в качестве параметра NULL.</p> <p>size — Запрашиваемое количество 32-битных слов, которое нужно принять от модуля.</p> <p>timeout — Таймаут на выполнение операции в миллисекундах. Значение 0 означает значение таймаута по умолчанию. 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>

5.3.6.2 Передача данных в модуль

Формат: INT LTR_Send (TLTR *hmodule, const DWORD *data, DWORD size, DWORD timeout)
Описание: <p>Функция посылает данные модулю в 32-битных словах в специальном формате слов LTR.</p> <p>Функция возвращает управление либо когда все данные будут записаны в буфер на передачу, либо по истечению таймаута. При этом реальное количество слов, записанное в буфер на передачу, можно узнать по возвращенному значению. Т.е. возвращенное значение гарантирует, что данное количество слов было поставлено на передачу, но эти данные еще могут не дойти до модуля на момент завершения функции.</p>
Параметры: <p>hmodule — Описатель соединения с модулем.</p> <p>data — Массив, содержащий данные в виде size 32-битных слов, которые нужно передать в модуль.</p> <p>size — Количество слов, которое нужно послать модулю.</p> <p>timeout — Таймаут на выполнение операции в миллисекундах. Значение 0 означает значение таймаута по умолчанию. Если в течение заданного времени в буфере на передачу не будет места для записи запрашиваемого количества слов, то функция все равно вернет управление, возвратив в качестве результата реально количество слов, записанных в буфер.</p>
Возвращаемое значение: <p>Значение меньше нуля соответствует коду ошибки. Значение больше или равное нулю соответствует количеству реально записанных в буфер на передачу слов, которые поставлены на отправку в модуль.</p>

5.3.6.3 Чтение времени последней секундной метки

Формат: INT LTR_GetLastUnixTimeMark (TLTR *hmodule, LONGLONG *unixtime)
Описание: <p>Данная функция возвращает значение времени, соответствующее последнему обнаруженному приходу расширенной метки “СЕКУНДА” с абсолютным временем при приеме данных по данному соединению посредством LTR_Recv().</p> <p>Данная функция работает только в случае, если крейт поддерживает генерацию расширенной метки “СЕКУНДА”, с указанием абсолютного времени, каким-либо способом. Например, с использованием сервера времени, который передает время по протоколу IRIG-B.</p> <p>Если метки “СЕКУНДА” с абсолютным временем принято не было, то будет возвращено значение 0.</p>
Параметры: <p>hmodule — Описатель соединения с модулем</p> <p>unixtime — Абсолютное значение времени в секундах с 1 января 1970 года (unixtime).</p>
Возвращаемое значение: <p>Код ошибки.</p>

5.3.7 Вспомогательные функции

5.3.7.1 Получение текстового сообщения об ошибке

Формат: LPCSTR LTR_GetErrorString (INT err)
Описание: Функция возвращает строку, соответствующую переданному коду ошибки, в кодировке CP1251 для ОС Windows или UTF-8 для ОС Linux. Функция поддерживает только коды ошибок, определенные в данной библиотеке и возвращаемые функциями данной библиотеки. Библиотеки для конкретных модулей могут иметь дополнительные коды ошибок и свои функции получения текстового описания ошибки, которые поддерживают эти дополнительные коды ошибок.
Параметры: err — Код ошибки.
Возвращаемое значение: Указатель на строку, содержащую сообщение об ошибке.

5.3.7.2 Установка таймаута по умолчанию для соединения

Формат: INT LTR_SetTimeout (TLTR *hnd, DWORD tout)
Описание: Функция устанавливает таймаут по умолчанию на выполнения операций для указанного соединения. При открытии соединения этот таймаут равен LTR_DEFAULT_SEND_RECV_TIMEOUT . Для управляющего соединения этот таймаут определяет время на выполнения любой команды (от передачи запроса службе до приема ответа). Для соединения с модулем — это таймаут, который используется в LTR_Recv() или LTR_Send() , если в них передано нулевое значение таймаута
Параметры: hnd — Описатель соединения. tout — Таймаут по умолчанию в мс.
Возвращаемое значение: Код ошибки .