

МНОГОКАНАЛЬНЫЕ СИСТЕМЫ СБОРА ДАННЫХ

КРЕЙТОВАЯ СИСТЕМА LTR

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ СИСТЕМЫ LTR ПОД
ОС WINDOWS, LINUX, FREEBSD И QNX

*Ревизия 1.0.5
Ноябрь 2016*

Автор руководства:

Борисов Алексей

ООО “Л Кард”

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

тел.: +7 (495) 785-95-25

факс: +7 (495) 785-95-14

Адреса в Интернет:

<http://www.lcard.ru>

E-Mail:

Отдел продаж: sale@lcard.ru

Техническая поддержка: support@lcard.ru

Отдел кадров: job@lcard.ru

Таблица 1: Ревизии текущего документа

Ревизия	Дата	Описание
1.0.0	12.02.2013	Первая ревизия данного документа
1.0.1	22.04.2013	Добавлено описание утилит для настройки интерфейса связи с ПК и обновления прошивки для крейтов LTR-EU
1.0.2	16.04.2014	Изменено название программы Qltrd на LTR Manager . Добавлено описание возможности программного получения журнала ltrd , добавлено описание возможности запуска ltrd как службы под ОС Windows. Добавлено описание настройки интерфейса крейтов LTR-EU из LTR Manager .
1.0.3	27.07.2016	Обновлено описание с учетом факта, что кроссплатформенное ПО является основной версией и под ОС Windows, а также с учетом появления документа “Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению” . Добавлена информация о возможности работы под FreeBSD. Выделен раздел установки ПО и переработан раздел о сравнении возможностей. Учтено появление крейтов LTR-CEU и описана возможность обновления прошивки из LTR Manager . Также внесены общие правки в текст
1.0.4	12.10.2016	Часть информации по ltrap1 вынесена в обновленное руководство по данной библиотеке. Добавлено описание управления настройками синхронизации крейта в программе LTR Manager
1.0.5	08.11.2016	Добавлена информация по флагу повторного подключения при ошибке соединения для записей с IP-адресами крейтов и соответствующей информацией в конфигурации ltrd

Оглавление

1	О чем этот документ	5
2	Установка	6
2.1	Windows	6
2.2	Linux	6
2.3	FreeBSD	7
2.3.1	Установка собранных бинарных пакетов	7
2.3.2	Установка через Ports Collection	7
2.4	QNX6.5	8
2.5	QNX4.25	8
2.6	Сборка из исходных кодов	9
3	Различия в поддерживаемых возможностях	10
3.1	Различия поддержки для различных ОС	10
3.2	Ограничения новой версии по сравнению со старым вариантом работы через LTR Server	10
4	Состав программного обеспечения	12
4.1	ltrd	13
4.1.1	Конфигурационный файл	13
4.1.2	Запуск ltrd	14
4.1.2.1	Запуск ltrd как демона	15
4.1.2.2	Запуск ltrd с использованием systemd	16
4.1.2.3	Запуск ltrd как демона FreeBSD	17
4.1.2.4	Запуск ltrd в качестве службы Windows	17
4.1.2.5	Запуск ltrd в качестве обычной консольной фоновой программы	18
4.1.3	Журнал ltrd	18
4.1.4	Управление соединениями с крейтами по интерфейсу Ethernet	19
4.1.4.1	Список адресов крейтов	19
4.1.4.2	Установление соединения с крейтом	20
4.1.4.3	Автоматическое соединение с крейтом при старте	21
4.1.4.4	Повторное соединение при ошибке	23
4.1.5	Установка соединения с крейтами по интерфейсу USB	24
4.1.6	Список активных крейтов	25
4.1.7	Клиентские подключения	26
4.2	ltrapi	27
4.3	ltrctl	27
4.3.1	Команды управления ip-адресами	28

4.3.2	Команды управления подключениями по ip-адресам	30
4.3.3	Получение информации по активным крейтам и модулям	31
4.3.4	Управление параметрами ltrd	31
4.3.5	Дополнительные команды	32
4.4	LTR Manager	32
4.4.1	Настройка интерфейса крейтов LTR-EU/LTR-CEU	34
4.4.2	Управление настройками синхронизации крейтов	35
4.4.3	Обновление прошивки процессора	37
4.4.4	Работа с удаленным ltrd	37
4.5	Дополнительные утилиты	38
4.5.1	Утилиты настройки и обновления прошивок крейтов LTR-EU	38
4.5.1.1	Установка настроек интерфейса	38
4.5.1.2	Чтение текущих настроек интерфейса	39
4.5.1.3	Обновление прошивок крейта	39

Глава 1

О чем этот документ

Данный документ представляет собой описание кроссплатформенного программного обеспечения для работы с крейтами и модулями LTR и руководство по его использованию. Данная версия ПО является основной поддерживаемой версией под все операционные системы, включая ОС Windows, для которой новая версия является совместимой заменой ранее используемого ПО под Windows (основанного на программе **LTR Server** и библиотеках **ltrapi** версии 1.27 и ниже). Описываемую в данном документе версию ПО рекомендуется использовать во всех проектах, однако для старых проектов следует ознакомиться с [разделом 3.2](#), описывающим возможные несовместимости в некоторых редких случаях (при этом в случае их возникновения рекомендуется обратиться в службу поддержки).

Документ предполагает, что пользователем был прочитан документ [“Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению”](#), в котором дается общий обзор программного обеспечения.

В данном документе описываются подробно описаны программы, входящие в состав базового ПО, особенности их работы для разных поддерживаемых операционных систем и т.п.

В данном документе не рассматриваются аппаратные возможности крейтов и модулей LTR, так как эта информация содержится в [“Руководстве пользователя LTR”](#).

Глава 2

Установка

Для части поддерживаемых систем предлагаются готовые средства для установки ПО. Способ установки готового ПО зависит от операционной системы и описан в соответствующих разделах ниже.

Также для любой операционной системы доступна возможность сборки ПО из исходных кодов, о чем описано в [разделе 2.6](#).

2.1 Windows

Созданы готовые установщики под ОС Windows (поддерживаются версии с Windows XP и выше). Порядок установки описан в документе [“Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению”](#).

2.2 Linux

Для части дистрибутивов Linux “Л Кард” предоставляет репозитории с готовыми пакетами. О том как подключить репозиторий “Л Кард” как внешний репозиторий в поддерживаемых дистрибутивах Linux или установить пакеты вручную описано в документе [“Использование внешних репозиторияев “Л Кард” для дистрибутивов Linux”](#).

Здесь же приведен конкретный список пакетов, соответствующий программному обеспечению для работы с LTR.

- **ltrd** — устанавливает демон, запускает его и также разрешает его запуск при старте системы. Может зависеть от пакетов с библиотеками **libxml2**, **libusb-1.0**, а также на некоторых системах от **libdaemon** или **systemd**.
- Пакеты для установки библиотек **ltrapi**. В .rpm и .deb дистрибутивах принято разделять пакеты для библиотек на пакеты с самими библиотеками (runtime), которые используются собранными программами, и пакеты для разработчиков (с заголовками и символьными ссылками), которые используются для написания своих программ¹. Кроме того, для каждой библиотеки должен быть свой пакет с именем, соответствующим имени библиотеки, что позволяет автоматически отслеживать зависимости приложения от пакетов используемых библиотек. Таким образом файлы **ltrapi** разбиты на следующие пакеты:

¹Для Arch Linux такой практики нет, и все включено в один пакет с именем **ltrapi**

- Пакет **libltrapi1** с общей библиотекой для доступа к **ltrd** и управления крейтами.
- По пакету на каждую библиотеку для работы с модулем с именем **libltrXXXapi1** (**libltr11api1**, **libltr22api1** и т.д.). Эти пакеты зависят от **libltrapi1**. Пакеты для модулей, в которые нужно записывать каждый раз прошивку контроллера или ПЛИС, помимо **.so** файла включен также файл прошивки, который устанавливается как правило в **/usr/share/ltrapi/ltrXXX**.
- **libltrapi1-dev** или **libltrapi1-devel** — общий пакет с заголовками для всех библиотек, зависящий от всех пакетов библиотек, упомянутых выше.

В случае установки из подключенного репозитория зависимости должны разрешаться автоматически и можно для разработки приложений ставить сразу пакет для разработки. В случае установки пакетов вручную необходимо устанавливать пакеты в указанном порядке.

- **ltrctl** — устанавливает соответствующую утилиту. Зависит от **libltrapi1**
- **ltrmanager** — устанавливает программу **LTR Manger**. Зависит от Qt4 или Qt5. Использует библиотеки из **ltrapi**, а также программу **lboot** для обновления прошивки крейтов LTR-CEU.
- **ltreu-config** — пакет с утилитой командной строки для настройки интерфейса крейтов LTR-EU (см. [раздел 4.5.1](#)).
- **ltreu-firm-update** — пакет с утилитой командной строки для обновления прошивки процессора крейтов LTR-EU (см. [раздел 4.5.1](#)).

2.3 FreeBSD

Для FreeBSD портирован демон **ltrd**, библиотеки **ltrapi** и утилита **ltrctl**. Их можно установить двумя разными способами, как описано ниже.

2.3.1 Установка собранных бинарных пакетов

Для поддерживаемых версий FreeBSD и поддерживаемых архитектур предоставляются уже собранные пакеты. Их можно скачать по ссылке https://bitbucket.org/lcard/ltr_cross_sdk/downloads/ltr-freebsd-packages.tar.gz.

После разархивирования можно установить пакеты с помощью утилиты **pkg** (см. описание использования утилиты на странице https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/pkgng-intro.html), взяв пакеты из нужной директории, соответствующей нужной версии и архитектуре ОС (обозначена как **<ltr_pkg_dir>**):

```
pkg install <ltr_pkg_dir>/*
```

2.3.2 Установка через Ports Collection

Данный вариант, в отличие от не устанавливает уже собранные бинарные пакеты, а производит установку путем сборки из исходников с использованием [Ports Collection](#).

При этом сохраняется отслеживание зависимостей пакета, установленных файлов и остальные возможности, что и при установке пакетов.

Если коллекция портов FreeBSD не установлена, то в первую очередь необходимо ее установить:

```
portsnap fetch extract
```

А если установлена, то обновить до последней версии:

```
portsnap fetch update
```

Далее необходимо скачать версию портов для LTR из архива по ссылке: https://bitbucket.org/lcard/ltr_cross_sdk/downloads/ltr-freebsd-ports.tar.gz.

Содержимое директории ports архива нужно скопировать в директорию с установленной коллекцией портов (/usr/ports, может быть переопределена через \$(PORTSDIR)).

Далее из директории каждого порта (например, /usr/ports/science/ltrd) выполнить установку порта:

```
make install
```

Порт затем можно всегда удалить, выполнив из его директории:

```
make deinstall
```

Исходные коды самих утилит и библиотек не входят в состав порта, а архив с исходными кодами скачивается с сайта во время установки. Если нужно отдельным этапом скачать архивы с исходниками, чтобы дальнейшая сборка выполнялась без необходимости выхода в интернет, то можно выполнить из директории порта

```
make fetch
```

Либо можно вывести список скачиваемых файлов через (или точнее команд, выполняемых на этапе fetch):

```
make fetch-list
```

и вручную скопировать файлы и перенести в директорию /usr/ports/distfiles.

2.4 QNX6.5

Бинарную сборку утилит и библиотек для архитектуры x86 можно скачать по ссылке: https://bitbucket.org/lcard/ltr_cross_sdk/downloads/ltr_qnx6.tar.gz. После этого достаточно скопировать содержимое архива на целевую систему. Запуск **ltrd** или добавление в автозапуск необходимо выполнить вручную.

2.5 QNX4.25

Бинарную сборку утилит и библиотек для архитектуры x86 можно скачать по ссылке: https://bitbucket.org/lcard/ltr_cross_sdk/downloads/ltr_qnx4.zip. После этого достаточно скопировать содержимое архива на целевую систему. Запуск **ltrd** или добавление в автозапуск необходимо выполнить вручную.

2.6 Сборка из исходных кодов

Архив с исходниками можно скачать со [странички репозитория](#) данного проекта. Однако для получения самой последней версии исходных кодов рекомендуется клонировать репозиторий программой **mercurial**. Подробнее об использовании репозитория описано в документе [“Использование открытых репозиториях исходных кодов “Л Кард” на bitbucket.org](#).

Для сборки необходима утилита **cmake**, которая выполняет конфигурацию проекта в зависимости от системы, а также поддерживаемый компилятор C/C++.

Процесс сборки описан в файле `INSTALL.txt` архива с исходными кодами.

Глава 3

Различия в поддерживаемых ВОЗМОЖНОСТЯХ

3.1 Различия поддержки для различных ОС

- Поддержка интерфейса USB доступна только для ОС Windows и для ОС Linux с наличием библиотеки **libusb-1.0**. Для остальных ОС поддерживается только работа с крейтом по интерфейсу Ethernet через протокол TCP/IP (т.е. можно использовать только крейты с наличием этого интерфейса). Соответственно только на ОС с поддержкой USB доступны утилиты для операций, которые возможны только по интерфейсу USB (настройка интерфейса и обновление прошивок для крейтов LTR-EU).

Важно!: Это означает в частности, что настройку крейта на работу по Ethernet интерфейсу и назначение нужного IPv4-адреса необходимо выполнить предварительно под ОС Windows или Linux

- Графическая утилита **LTR Manager** доступна только для ОС Windows и Linux.
- Для ОС QNX **ltrd** работает как обычная консольная программа, которую нужно вручную запускать в фоне или самостоятельно добавить в скрипты автозапуска (для остальных ОС реализован запуск **ltrd** как службы/демона штатным для системы образом).
- Для ОС QNX6/QNX4 следует учитывать, что версия использует тот же исходный код и ту же идеологию взаимодействия с модулями, что и версия под остальные ОС общего назначения, а не является специализированным ПО, учитывающим все особенности операционной системы реального времени. Пользователю следует оценить применимость данного ПО в зависимости от решаемой им задачи.

3.2 Ограничения новой версии по сравнению со старым вариантом работы через LTR Server

Кроссплатформенный вариант программного обеспечения для LTR предоставляет схожие возможности с теми, что предоставляет и ранее используемая версия для ОС Windows, за исключением следующих пунктов:

- Не поддерживается возможность получения “сырых” необработанных **ltrd** данных с помощью `LTR_GetCrateRawData()`.
- Не поддерживается возможность создания специального пользовательского виртуального слота для получения пользовательских данных от крейта
- Не поддерживается смена приоритета работы **ltrd** с помощью `LTR_SetServerProcessPriority()`
- Не включены некоторые отладочные режимы
- Может не быть совместимости с ПО, которое явно требует наличие запущенной программы с именем **LTR Server** (например, проявляет окно или процесс с таким названием или явно само запускает программу **LTR Server**)
- Отличия новой версии `ltrapi` описаны в [“Руководстве программиста библиотеки `ltrapi`”](#).

Глава 4

Состав программного обеспечения

Для работы с LTR и написания собственных программ используются следующие программные компоненты:

- **ltrd** — выполняемая в фоне сервисная программа (служба или демон), обслуживающая подключения к крейтам и выполняющая разбор потока данных от крейта и подготовку потока данных для крейта. Все взаимодействие с крейтами выполняется через **ltrd**.
- **ltrapi** — набор библиотек, предоставляющих функции для работы с модулями, для управления крейтами и управления работой самого **ltrd**. Для выполнения этих действий в библиотечных функциях устанавливается клиентское соединение с уже запущенным **ltrd**.
- **LTR Manager** — утилита с графическим интерфейсом, служащая для получения информации о подключенных крейтах, статистики, журнала **ltrd**, а также для подключения крейтов по Ethernet, настройки крейтов, обновления прошивок и других дополнительных действий.
- **ltrctl** — утилита с интерфейсом командной строки, выполняющая сходные функции с **LTR Manager**
- дополнительные утилиты для работы с определенными крейтами или модулями

В общем случае для работы с модулями необходима запущенная программа **ltrd**. На компьютере с пользовательской программой (это может быть не обязательно тот же компьютер, на котором запущен **ltrd**) должны быть установлены библиотеки **ltrapi**, с помощью которых программа устанавливает соединение с **ltrd** и работает с нужными модулями (на каждый модуль свое соединение).

Следует отметить, что один экземпляр **ltrd** может установить соединение с несколькими крейтами, в то время как с каждым крейтом может одновременно работать только одна служба **ltrd**. Если необходимо работать с разными модулями одного крейта с разных компьютеров, то **ltrd** должен быть запущен только на одной машине, в то время как программы работающие на втором компьютере должны устанавливать соединение с **ltrd**, запущенном на первом.

Разница при работе с удаленным (в отличие от работы с локальным) **ltrd** заключается лишь в том, что при вызове соответствующей функции **LTR_Open()** нужно задать IP-адрес компьютера, на котором запущен **ltrd**, а не **LTRD_ADDR_DEFAULT**, как при локальном соединении.

Следует также отметить, что интерфейс между **ltrapi** и **ltrd** совместим с соответствующим интерфейсом ранее используемой программы **LTR Server** для ОС Windows. Поэтому одна и та же программа может работать как с программой **ltrd**, так и **LTR Server**. Следует только учесть, что некоторые команды введены только в **ltrd** (получение статистики, сброс модуля, настройка параметров **ltrd**), в то время как некоторые редко используемые возможности **LTR Server** не реализованы в **ltrd** (см. [раздел 3.2](#)).

4.1 ltrd

Служба **ltrd** необходима для работы всего остального программного обеспечения для взаимодействия с LTR-модулями и крейтами. **ltrd** устанавливает соединение с крейтами, а уже остальные программы через клиентские соединения с **ltrd** работают с нужными модулями и крейтами. Программа **ltrd** выполняется в фоне, запускается как правило вместе с системой и не содержит пользовательского интерфейса.

Для предоставления интерфейса для управления **ltrd** служат дополнительные программы (frontends): **ltrctl** (предоставляет интерфейс командной строки) и **LTR Manager** (предоставляет графический интерфейс с использованием библиотеки Qt4/Qt5). Эти программы используют лишь функции **ltrapi**, поэтому при необходимости могут быть написаны свои варианты интерфейсов в дополнение к предоставляемым.

4.1.1 Конфигурационный файл

ltrd хранит свои настройки в конфигурационном файле формата XML. Все эти настройки могут изменяться и программным образом через функции **ltrapi** или через **LTR Manger**, так и напрямую редактируя файл конфигурации, как будет описано ниже. Если пользователь не планирует изменять файл конфигурации вручную, то данный раздел может быть пропущен.

В стандартных пакетах для ОС Linux файл конфигурации устанавливается в `/etc/ltrd/config.xml`, а в ОС Windows в директорию вместе с **ltrd**. Путь к файлу задается с помощью опции `--config-file` при запуске **ltrd**. Пример этого файла приведен в исходном коде с программой.

Ниже приведен соответствующий иерархический список тегов с описанием задаваемых параметров.

Тегом документа всегда должен быть `<ltrsrv_config>`. Он содержит следующие теги:

- `<log>`. Настройки журнала
 - `<level>`. Задаёт уровень выводимых сообщений журнала от 0 до 7 (см. [раздел 4.1.3](#)).
- `<crate_ip_entries>`. Содержит записи с IP-адресами крейтов (см. [раздел 4.1.4.1](#))
 - `<ip>`. Данных тегов может быть произвольное количество. Каждый описывает свою запись. Значение содержит строку с IPv4-адресом LTR-крейта. У каждого тега могут быть следующие атрибуты:
 - * `“autocon”` — задает, нужно ли выполнять автоматическое подключение к данному крейту при запуске **ltrd** (см. [раздел 4.1.4.3](#)). Может принимать значения “1” или “0”.

* “reconnect” — задает, нужно ли выполнять повторную попытку подключения при ошибке соединения с данным крейтом (см. [раздел 4.1.4.4](#)). Может принимать значения “1” или “0”.

- <intf_eth_params>. Параметры для дополнительной настройки интерфейса между **ltrd** и крейтом при работе по Ethernet. При отсутствии используются значения по умолчанию. В большинстве случаев пользователю не требуется изменять эти параметры.
 - <intf_check_time>. Интервал (в мс) между проверками адресов интерфейсов локальной машины. Используются при определении момента для начала автоподключения крейта.
 - <crate_poll_time>. Интервал (в мс) опроса крейтов. Если с крейтом не было обмена в течении этого интервала, то ему будет послана команда, чтобы проверить, что крейт еще присутствует в сети. Если это значение установить в 0, то опрос крейтов осуществляться не будет.
 - <crate_connection_timeout>. Таймаут (в мс) на время установления TCP-соединения с крейтом
 - <crate_ctlcmd_timeout>. Таймаут (в мс) на время выполнения управляющих команд, посылаемых крейту (предельное время ожидания ответа на команду).
 - <crate_reconnect_time>. Время (в мс), через которое после возникновения ошибки соединения с крейтом будет выполнена повторная попытка подключения (если для данной записи установлен соответствующий признак).

При изменении этих настроек с использованием API-функций **ltrapi**, **ltrctl** или **LTR Manager**, **ltrd** сразу сохраняет измененный файл. Для сохранения изменений нужно убедиться, что у пользователя, от имени которого выполняется **ltrd**, есть право на запись файла конфигурации (в противном случае изменения не будут сохраняться между запусками **ltrd**).

Файл конфигурации при необходимости можно отредактировать вручную при остановленном **ltrd** (в противном случае **ltrd** перепишет файл при завершении работы). Также для ОС, поддерживающих посылку POSIX-сигналов (все поддерживаемые, кроме Windows), возможно изменить файл настройки вручную и при работающем **ltrd**. Для этого после изменения файла конфигурации нужно послать **ltrd** сигнал SIG_HUP, по которому **ltrd** заново прочитает настройки из файла.

4.1.2 Запуск **ltrd**

При установке через установщик на ОС Windows, установке пакетов для поддерживаемых дистрибутивов Linux или пакета/порта на FreeBSD все необходимые настройки для автоматического запуска службы при старте системы выполняются автоматически при установке. В этом случае пользователю нет необходимости знать подробности запуска службы, описанные в данном разделе.

Ниже приведена информация о том, как выполняется запуск скорее для ознакомительных целей, для случая, если нужно вручную останавливать/запускать службу, а также для запуска **ltrd** в ОС, для которых нет готовых пакетов.

4.1.2.1 Запуск `ltrd` как демона

В ОС Linux, FreeBSD и др. для выполнения фоновых задач используются как правило демоны (`daemon`). Демоны запускаются вместе с системой, не связаны с какой-либо сессией и пользовательским терминалом и их родительским процессом является процесс `init`, отвечающий за запуск системы.

Для запуска в режиме демона, используются опция `--daemon`.

При запуске в режиме демона `ltrd` выполняет следующие стандартные для демона шаги:

- если были указаны опции `--user` и/или `--group`, то `ltrd` изменяет пользователя и группу, от имени которых он выполняется, на указанные значения. Это используется, чтобы освободиться от привилегий суперпользователя (`root`), в случае, когда `ltrd` запускается от имени суперпользователя при старте системы. При установке пакетов из репозитория создается специальный пользователь и группа с именем `ltrd`, от лица которого и выполняется демон
- Сброс всех обработчиков сигналов
- Проверка не запущена ли другая копия демона по `pid`-файлу, указанному с помощью опции `--pid-file`. При установке из пакетов используется файл `/var/run/ltrd/pid`
- Создание дочернего процесса, независимого от терминала
- Дочерний процесс делается лидером сессии
- Дочерний процесс устанавливает в качестве текущей директории корневую директорию (`/`)
- Дочерний процесс закрывает все дескрипторы файлов
- Дочерний процесс создает `pid`-файл, указанный опцией `--pid-file`, чтобы демон не мог быть запущен несколько раз. В связи с этим пользователь, от имени которого выполняется демон, должен иметь соответствующие права
- Дочерний процесс сообщает родительскому о том, что инициализация прошла успешно.
- Родительский процесс завершается и остается только дочерний, который и выполняет всю работу
- По завершению работы дочерний процесс удаляет `pid`-файл

Для запуска демонов используются специальные `init`-скрипты. Их организация зависит от конкретного дистрибутива.

В используемых пакетах `init`-скрипты используются в основанных на Debian дистрибутивах. В этих дистрибутивах используются скрипты в стиле System V ([SysV-style init scripts](#)). В них `init`-скрипт `ltrd` устанавливается в `/etc/init.d`. Для автоматического запуска и останова создаются символичные ссылки на `/etc/init.d/ltrd` в директориях `/etc/rcN.d` с именами `S??ltrd` (для запуска) или `K??ltrd` (для останова), где `N` — соответствующий уровень запуска (`runlevel`), а `??` — двузначный номер, определяющий

последовательность запуска скриптов. Создание символических ссылок выполняется с помощью системного скрипта **update-rc.d**, который может определять уровни запуска и последовательность запуска скриптов по заголовку в самом `init`-скрипте.

Ручной запуск демона выполняется с помощью:

```
sudo invoke-rc.d ltrd start
```

Ручной останов демона:

```
sudo invoke-rc.d ltrd stop
```

4.1.2.2 Запуск `ltrd` с использованием `systemd`

Такие дистрибутивы как `Fedora`, `OpenSuse` и `Arch Linux` перешли в современных версиях с традиционных `init`-скриптов на использование программы **systemd** для управления службами и запуском системы.

В отличие от традиционных демонов, `systemd`-демоны не выполняют все описанные выше шаги сами, так как за них эти действия уже выполняет **systemd**. Вместо скриптов используется декларативный файл с описанием службы `ltrd.service` (устанавливается обычно в `/lib/systemd/system` или `/usr/lib/systemd/system` при установке из репозитория, а `/etc/systemd/system` используется как правило при ручной установке).

При этом для поддержки дополнительных возможностей **systemd ltrd** запускается с ключем `--systemd-mode`.

При работе с **systemd** дополнительно используются возможности:

- Рестарт **ltrd** в случае, если он был завершен с ошибкой
- Watchdog-таймер: **ltrd** посылает периодически **systemd** уведомление. Если оно не будет послано в течение заданного интервала (что может произойти только при зависании демона), то **systemd** перезапустит демон.
- Сообщения **ltrd** передаются в журнал **systemd**, откуда уже перенаправляются обычно в системный журнал через **syslog**. Журнал **systemd** может быть просмотрен с помощью **journalctl** (см. пример ниже)

Ручной запуск службы выполняется с помощью:

```
sudo systemctl start ltrd.service
```

Ручной останов:

```
sudo systemctl stop ltrd.service
```

Разрешения запуска с запуском системы:

```
sudo systemctl enable ltrd.service
```

Запрещение запуска с запуском системы:

```
sudo systemctl disable ltrd.service
```

Проверить состояние работы:

```
sudo systemctl status ltrd.service
```

Вывод последних 20 записей из лога **systemd**, относящихся к **ltrd** (клавишей `q` осуществляется выход из просмотра журнала):

```
sudo journalctl --lines=20 _SYSTEMD_UNIT=ltrd.service
```

4.1.2.3 Запуск `ltrd` как демона FreeBSD

В ОС FreeBSD `ltrd` запускается как демон также, как было описано в [разделе 4.1.2.1](#), с использованием `libdaemon` и аналогичными действиями при запуске. Отличием является расположение и состав скриптов запуска, принятые в FreeBSD для управления демонами.

В FreeBSD скрипты запуска располагаются в директориях `/etc/rc.d` или `/usr/local/etc/rc.d`. Последние используются портами и именно в нее устанавливается скрипт запуска `ltrd`. После установки демон разрешен и запущен. Запрещен он может быть при необходимости через `rc.conf`, путем добавления переменной

```
ltrd_enabled='NO'
```

Управление возможно с помощью утилиты `service`, как и для остальных служб (как описано в соответствующем разделе [документации FreeBSD](#)):

Проверка, разрешена ли служба:

```
service ltrd rcvar
```

Проверка, запущена ли служба:

```
service ltrd status
```

Ручной останов службы:

```
service ltrd onestop
```

Ручной запуск службы:

```
service ltrd onestart
```

4.1.2.4 Запуск `ltrd` в качестве службы Windows

Под ОС Windows `ltrd` с ключом `--daemon` запускается как служба Windows. Все необходимые действия могут быть выполнены с помощью стандартных средств Windows. Однако часть действий для удобства реализовано и внутри программы `ltrd` и выполняется с помощью запуска `ltrd` со специальными ключами:

- `--install` — регистрация службы `ltrd` в системе. Перед запуском службы она должна быть зарегистрирована.
- `--remove` — удаление информации о зарегистрированной службе `ltrd`
- `--start` — ручной запуск службы `ltrd`
- `--stop` — ручной останов службы `ltrd`
- `--autostart-enable` — разрешение автоматического запуска службы `ltrd` вместе с системой
- `--autostart-disable` — запрет автоматического запуска службы `ltrd` вместе с системой

При установке стандартным установщиком, служба сразу запускается и разрешается ее запуск вместе с системой, таким образом, не требуется в дальнейшем никаких дополнительных действий. Также могут быть созданы ярлыки для ручного запуска/останова службы и разрешения/запрета ее запуска вместе с системой.

Важно!: Служба **ltrd** и программа **LTR Server** не могут быть запущены одновременно на одной машине. Для корректной работы **ltrd** должна быть закрыта программа **LTR Server**, а для запуска **LTR Server** должен быть остановлена служба **ltrd**

4.1.2.5 Запуск **ltrd** в качестве обычной консольной фоновой программы

ltrd может быть запущен как обычная консольная программа для упрощения запуска на системах, где все нужные настройки не выполняются установкой пакетов автоматически (не упомянутые дистрибутивы Linux и в QNX). Для этого достаточно запустить **ltrd**, указав конфигурационный файл (для случая, когда конфигурационный файл находится в `/etc/ltrd/config.xml`):

```
ltrd --config-file=/etc/ltrd/config.xml &
```

При желании смены пользователя (для снижения привилегий, если запуск идет от суперпользователя) можно использовать опции `--user=<имя пользователя>` и `--group=<группа>`.

4.1.3 Журнал **ltrd**

ltrd выводит сообщения о процессе своей работы (журнал) одним из следующих способов:

- Вывод на `stdout` — этот вариант используется при запуске **ltrd** как обычной консольной программы
- Под ОС Windows при работе в качестве службы сообщения журнала сохраняются в журнал системных событий.
- Сохранение в системный лог с помощью вызова `syslog()` для ОС, поддерживающих данную функцию — этот вариант может использоваться как в дочернем процессе демона, так и при запуске **ltrd** как консольной программы. Сообщения журнала сохраняются **syslogd** (или его аналогом), как правило в один из файлов в директории `/var/log` (например `/var/log/messages`) и могут быть просмотрены стандартными средствами системы для просмотра системного журнала.
- При запуске с ключом `--systemd-mode` лог сохраняется в журнал **systemd** (откуда как правило передаются также и в системный лог) и может быть просмотрен через **journalctl** (как было указано в [разделе 3.2](#)).
- Сообщения журнала также сохраняются в кольцевой буфер (если было разрешено при сборке) и могут быть получены клиентской программой через функции библиотеки **ltrlogapi**, устанавливающей специальный тип управляющего соединения с **ltrd**. В частности эту библиотеку использует программа **LTR Manager**

для вывода журнала **ltrd** в специальное окно. В примерах также есть вариант реализации консольной программы, выводящей сообщения **ltrd** на стандартный вывод.

Каждому выводимому сообщению присваивается уровень важности от 0 до 7 (0 – критические ошибки, 7 – самый низкий уровень для отладки). Для **ltrd** имеется настройка уровня выводимых сообщений, что позволяет не выводить сообщения уровней выше (менее критичные) заданного. Следует учитывать, что при обычной работе **ltrd** не следует устанавливать высокий уровень вывода отладочных сообщений (уровни 6,7), чтобы излишне не загружать системный лог. Типичный уровень – 3, при котором выполняется вывод основных сообщений о ходе работы и всех предупреждений/ошибок, или 4 – где к указанному добавляется дополнительная детальная информация.

Уровень журнала может быть настроен в [конфигурационном файле](#) или изменен при работе одним из следующих способов:

- **ltrapi** — с помощью функции `LTR_SetLogLevel()`.
- **ltrctl** — для установки уровня 4:

```
ltrctl setloglvl 4 perm
```

- **LTR Manager** — выбрать нужное соединение с **ltrd** из списка активных. Выбрать из меню “ltrd” или из контекстного меню пункт “Настройки ltrd...”. В появившемся окне установить нужный уровень и нажать “Ok”. В **LTR Manager** также отдельно настраивается отображаемый уровень сообщений в окне программы, т.е. он принимает все сообщения, которые посылает **ltrd**, но может отображать в окне только те, что удовлетворяют настроенному уровню отображения. Сообщение отображается только если оно одновременно не менее критичное чем настроенный уровень в **ltrd** и уровень отображения в окне **LTR Manger**

4.1.4 Управление соединениями с крейтами по интерфейсу Ethernet

Общий принцип работы с крейтами по Ethernet описан в документе [“Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению”](#). Здесь приводится подробное описание каждого этапа.

4.1.4.1 Список адресов крейтов

ltrd хранит список записей с IP-адресами крейтов, с которыми при желании *может быть* установлено соединение. Сам список записей по сути служит только для запоминания этих адресов. Помимо самого адреса запись может содержать дополнительные флаги ([раздел 4.1.4.3](#) и [4.1.4.4](#)). Список записей с адресами крейтов **ltrd** загружает из файла конфигурации при старте и сохраняется при завершении работы (за исключением случая, когда изменения вносились без сохранения).

Во время работы новые записи с адресами крейтов могут быть добавлены или удалены:

- **ltrapi** — с помощью функций `LTR_AddIPCrates()` и `LTR_DeleteIPCrates()` соответственно.

- **ltrctl:**

- добавление адреса 192.168.0.250:

```
ltrctl ipadd 192.168.0.250 perm
```

- удаление адреса 192.168.0.250:

```
ltrctl iprem 192.168.0.250 perm
```

- **LTR Manager**

- Для добавления нужно нажать кнопку “Добавить IP-адрес” на панели “IP-адреса крейтов”, в появившемся окне ввести адрес и нажать “Ок”
- Для удаления выделить нужный адрес на панели “IP-адреса крейтов” и нажать кнопку “Удалить IP-адрес”

4.1.4.2 Установление соединения с крейтом

Вы можете вручную установить соединения между **ltrd** и крейтом по Ethernet-интерфейсу, используя один из IP-адресов из [настроенного списка](#) в **ltrd**. То есть перед выполнением соединения запись с нужным IP-адресом крейта уже должна быть предварительно добавлена в список.

Запустить процесс установки соединения с крейтом можно следующим образом:

- **ltrapi** — с помощью функции `LTR_ConnectIPCratе()`
- **ltrctl**: — подключение по адресу 192.168.0.250:

```
ltrctl ipcon 192.168.0.250
```

- **LTR Manager** Двойной щелчок по строке с адресом на панели “IP-адреса крейтов”, для которого не установлено соединение, или выделение адреса и нажатие кнопки “Установить соединение с крейтом”

Следует учесть, что успешное выполнение вышеописанной операции свидетельствует лишь о успешном выполнении команды на начало подключения, т.е. само подключение в этот момент еще не завершено.

После команды на подключение **ltrd** выполняет следующие действия:

- Пытается установить управляющие соединения с крейтом с использованием TCP-протокола
- Считывает информацию о крейте
- Устанавливает TCP-соединение с крейтом для передачи данных

После успешного завершения всех шагов крейт добавляется в [список активных крейтов](#), после чего с ним или его модулями можно будет устанавливать клиентское соединение.

За состоянием выполнения подключения можно также смотреть по статусу подключения, связанному с соответствующей записью, которое можно получить следующим образом:

- **ltrapi** — с помощью функции `LTR_GetListOfIPCrates()`
- **ltrctl**:

```
ltrctl iplist
```

- **LTR Manager** Список адресов со статусом подключения всегда отображается в таблице на панели “IP-адреса крейтов”

После запуска подключения, статус записи изменяется на “Подключение...” (connecting), после чего переходит либо в состояние “Подключен” (online) при успешном завершении соединения, либо в состояние “Ошибка” (error) при возникновении ошибки подключения.

В любой момент Вы можете разорвать установленное (или находящееся в процессе установки) соединение с помощью:

- **ltrapi** — соединение с заданным адресом можно разорвать с помощью функции `LTR_DisconnectIPCrates()`, а разорвать все соединения с помощью `LTR_DisconnectAllIPCrates()`.

- **ltrctl**:

- разрыв соединения с крейтом с адресом 192.168.0.250:

```
ltrctl ipdisc 192.168.0.250
```

- разрыв соединения со всеми крейтами

```
ltrctl ipdisc all
```

- **LTR Manager** Двойной щелчок по строке с адресом на панели “IP-адреса крейтов”, с которым установлено соединение или выделение адреса и нажатие кнопки “Разорвать соединение с крейтом”. Для разрыва всех соединений использовать кнопку “Разорвать соединение со всеми крейтами”

4.1.4.3 Автоматическое соединение с крейтом при старте

Записи с адресами крейтов могут быть помечены флагом “Автоподключение”. Помимо того, что используя эти записи можно установить соединение с крейтом вручную, **ltrd** также пытается установить для этих записей соединение автоматически в следующих случаях:

- При запуске программы **ltrd**, в случае наличия хотя бы одного действительного IP-адреса хоста, на котором запущен **ltrd**
- При обнаружении нового адреса хоста, на котором запущен **ltrd**, **ltrd** пытается установить соединение по всем адресам с флагом “Автоподключение”, с которыми на этот момент еще не установлено успешное соединение. Для этого **ltrd** периодически (период является настраиваемым параметром) получает список адресов локальной машины.
- По команде подключения всех крейтов, записи которых помечены флагом “Автоподключение” (см. ниже), если соединение с крейтом в момент выполнения команды уже не установлено.

Второй вариант позволяет разрешить ситуацию, когда адрес хоста реально получается после запуска **ltrd**. Типичным примером этого является вариант, когда **ltrd** запускается вместе со стартом системы, а адрес машины получается автоматически по DNSР.

Примечание: Если выполнить подключение к крейту не удалось или соединение с крейтом будет разорвано позже из-за ошибки, то **ltrd** изменяет статус записи на “Ошибка” и по умолчанию больше не пытается установить соединение по этому адресу, даже в случае наличия флага “Автоподключение” (за исключением случаев смены адреса хоста), и установка соединения после этого возможно только вручную. Для возможности автоматического повторного подключения в этом случае используется отдельный флаг, описанный в [разделе 4.1.4.4](#). Таким образом, по умолчанию (без флага повторного подключения) для выполнения автоподключения нужно, чтобы крейт уже был включен в сеть к моменту запуска **ltrd**.

Указать, что требуется выполнять автоподключение, можно при добавлении новой записи:

- **ltrapi** — с помощью флага `LTR_CRATE_IP_FLAG_AUTOCONNECT` при вызове функции `LTR_AddIPcrate()`
- **ltrctl**: — добавление адреса 192.168.0.250 с флагом автоподключения:

```
ltrctl ipadd 192.168.0.250 auto perm
```

- **LTR Manager** — установить галочку “Автоматически подключать при запуске” в диалоге добавления адреса

Также можно изменить состояние флага автоподключения для уже добавленной записи:

- **ltrapi** — функцией `LTR_SetIPcrateFlags()` с флагом `LTR_CRATE_IP_FLAG_AUTOCONNECT`.
- **ltrctl**:

– установка флага автоподключения для адреса 192.168.0.250:

```
ltrctl ipflag 192.168.0.250 auto set perm
```

– сброс флага автоподключения для адреса 192.168.0.250:

```
ltrctl ipflag 192.168.0.250 auto clear perm
```

- **LTR Manager** — установить или снять галочку в столбце “А” в таблице адресов панели “IP-Адреса крейтов”

Примечание: Установка флага автоподключения при добавлении записи или для уже добавленной записи не приводит к подключению по соответствующему адресу, а только указывает **ltrd**, что автоподключение нужно выполнить при следующем запуске программы или при другом событии для автоподключения, описанном ранее.

Также можно использовать флаг автоподключения для ручного запуска подключения одной командой всех крейтов, для которых записи с IP-адресами помечены данным флагом:

- **ltrapi** — с помощью функции `LTR_ConnectAllAutoIPCrates()`
- **ltrctl**:

```
ltrctl ipcon auto
```

- **LTR Manager** — нажать кнопку “Автосоединение с крейтами” на панели “IP-адреса крейтов”,

4.1.4.4 Повторное соединение при ошибке

Для записей с адресами крейтов может быть установлен флаг “Повторное соединение при ошибке”. Данная возможность реализована в **ltrd** начиная с версии 2.1.5.0 и управление этой возможностью добавлено в **ltrapi** с версии 1.31.1, в **LTR Manager** с версии 1.5.1 и в **ltrctl** с версии 1.0.1.

Данная возможность указывает на то, что при ошибке подключения к этому крейту или в случае разрыва соединения уже подключенного крейта из-за ошибки (т.е. во всех случаях, когда IP-запись переходит в состояние “Ошибка”), необходимо запланировать новую попытку подключения. Если повторное подключение не удалось, то будет запланирована следующая попытка и т.д. до тех пор, пока не удастся установить соединение или не будет явно выполнено одно из следующих действий:

- Команда разрыва связи с данным крейтом (состояние “Ошибка” изменится на “Отключен” и повторные попытки подключения предприниматься не будут).
- Снятие флага повторного подключения для данной записи (см. ниже)
- Удаление данной записи

Время, через которое будет запланирована повторная передача, устанавливается отдельным дополнительным параметром конфигурации службы **ltrd** и оно не изменяется при каждой новой попытке (однако может варьироваться от одного до одного с половиной установленного в конфигурации интервала).

Данная возможность позволяет автоматически восстанавливать разорванное соединение с крейтом из-за временной ошибки (перезагрузка крейта, временная потеря связи из-за неисправности соединения и т.п.), а также позволяет запустить процесс соединения в тот момент, когда крейт еще не подключен к сети без необходимости реализации этой логики в приложении.

Этот флаг может использоваться как совместно с флагом “Автоподключение”, так и при ручном подключении.

Указать, что требуется выполнять повторное подключение при ошибке для новой записи можно при добавлении записи:

- **ltrapi** — с помощью флага `LTR_CRATE_IP_FLAG_RECONNECT` при вызове функции `LTR_AddIPCrates()`
- **ltrctl**: — добавление адреса 192.168.0.250 с флагом повторного подключения:

```
ltrctl ipadd 192.168.0.250 reconnect perm
```

- **LTR Manager** — установить галочку “Повторное подключение при ошибке” в диалоге добавления адреса

Также можно изменить состояние данного флага для уже добавленной записи:

- **ltrapi** — функцией `LTR_SetIPCratesFlags()` с флагом `LTR_CRATE_IP_FLAG_RECONNECT`.

- **ltrctl**:

– установка флага повторного подключения для адреса 192.168.0.250:

```
ltrctl ipflag 192.168.0.250 reconnect set perm
```

– сброс флага повторного подключения для адреса 192.168.0.250:

```
ltrctl ipflag 192.168.0.250 reconnect clear perm
```

- **LTR Manager** — установить или снять галочку в столбце “R” в таблице адресов панели “IP-адреса крейтов”

Примечание: Установка флага повторного подключения для записи, для которой состояние подключения уже соответствует ошибке, приводит к тому, что в течение заданного интервала будет запланирована попытка повторного подключения.

4.1.5 Установка соединения с крейтами по интерфейсу USB

USB-интерфейс в настоящее время поддерживается в **ltrd** под ОС Linux и ОС Windows.

В ОС Windows используется драйвер “Л Кард” для USB-устройств [lcomp](#), который должен быть отдельно установлен.

В ОС Linux используется доступная в большинстве дистрибутивов библиотека **libusb** (версии 1.0.8 и выше). В большинстве дистрибутивов по-умолчанию только root имеет доступ к неизвестным USB-устройствам. Для разрешения доступа используется файл `ltr-crates.rules` с правилами для **udev**. Этот файл устанавливается при установке из готовых пакетов в директорию `/usr/lib/udev/rules.d/` (на некоторых дистрибутивах может использоваться `/lib/udev/rules.d/`), а при ручной установке обычно используется директория `/etc/udev/rules.d`. Указанный файл делает пользователя **ltrd** владельцем файлов устройств USB-крейтов и дает именно ему доступ к поддерживаемым устройствам. То есть, если Вы запускаете **ltrd** вручную, то Вам нужно либо запускать его от имени пользователя **ltrd** (можно с помощью опций `--user` и `--group`, если пользователь и группа уже добавлены в систему), либо изменить соответствующим образом файл правил чтобы дать доступ всем. Также при ручной установке файла правил, чтобы не перезагружать систему или не переподключать уже подключенный крейт, Вы можете обновить правила **udev** и сгенерировать события по нужным устройствам:

```
sudo udevadm control --reload-rules
```

```
sudo udevadm trigger --subsystem-match=usb
```

При установке пакетов все необходимые действия уже выполняются во время установки.

При обнаружении нового крейта, подключенного по USB, **ltrd** устанавливает с ним соединение и выполняет его инициализацию автоматически. Дополнительных настроек не требуется. Только после завершения всей инициализации крейт попадает в [список активных крейтов](#).

Следует отметить, что при подключении по USB крейта, сконфигурированного на работу по Ethernet, он увидится по USB, но в специальном режиме без модулей, предназначенном только для настройки крейта. Узнать об этом состоянии можно посмотрев статистику крейта:

- **ltrapi** — после вызова `LTR_GetCrateStatistic()` по полю `crate_mode` (одна из констант `LTR_CRATE_MODE_`).
- **ltrctl** — в строке “Crate mode” при выводе статистики

```
ltrctl cstat
```

- **LTR Manager** — отображается на панели статистики в информации о крейте как “Режим работы”

4.1.6 Список активных крейтов

ltrd содержит список активных крейтов, с которыми **ltrd** успешно установил соединения. Только с крейтами из этого списка и их модулями клиенты могут установить клиентские соединения. В этот список крейты переходят после полного успешного завершения инициализации соединения.

Удаление из списка активных крейтов выполняется в следующих случаях:

- При явной команде отключения крейта по IP-адресу (в случае Ethernet интерфейса)
- При ошибке передачи управляющей команды, переданной крейту или ошибке приема/передачи данных (что свидетельствует о нарушении канала связи)
- При обнаружении исчезновения крейта в системе

При удалении крейта из списка активных, **ltrd** закрывает соединения всех клиентов, соединенных с этим крейтом или его модулями.

Для проверки действительности соединения с крейтом по Ethernet, **ltrd** может производит периодический опрос состава модулей крейта. При отсутствии ответа на команду за заданное время считается, что соединение с крейтом не действительно. Период опроса и таймаут являются настраиваемыми параметрами **ltrd**.

Список активных крейтов и их модулей можно получить следующим образом:

- **ltrapi** — С помощью функции `LTR_GetCratesEx()`, которая сразу возвращает список активных крейтов с информацией о них. Также для этого можно использовать функцию `LTR_GetCrates()` (которая была введена ранее и оставлена для совместимости) можно получить список серийных номеров крейтов, после чего по этим номерам можно установить соединения с крейтами с помощью `LTR_Open()` и получить от каждого список модулей через `LTR_GetCrateModules()`. Однако второй вариант может использоваться только для получения первых `LTR_CRATES_MAX` крейтов и не различает случай одновременного подключения крейта по двум интерфейсам.

- **ltrctl**:

```
ltrctl clist
```

- **LTR Manager** — в виде дерева в основном окне программы для каждого экземпляра **ltrd** показывается список его активных крейтов со списком модулей

4.1.7 Клиентские подключения

Для работы с крейтами или модулями, а также для управления работой **ltrd**, прикладные программы устанавливают клиентское соединение с **ltrd**.

Всего возможны три типа соединений:

- Управляющие соединения с **ltrd**. Это соединение позволяет управлять самим **ltrd** и получать от него информацию. Для него не требуется наличие активных крейтов. Команды по этому каналу **ltrd** обрабатывает сам.
- Управляющие соединения с крейтом. Позволяет управлять соответствующим крейтом (например, получение списка модулей или настройка синхрометок крейтов). Управляющие команды по этому интерфейсу **ltrd** преобразует в соответствующие команды к крейту или обрабатывает сам.
- Соединение с модулем, вставленным в крейт. С помощью данного соединения клиент работает с модулем с использованием протокола конкретного модуля. В отличие от предыдущих соединений, обмен по этому каналу представляет для **ltrd** непрозрачный поток данных на ввод или на вывод. **ltrd** не знает протоколов модулей и выполняет лишь мультиплексирование/демультиплексирование данных. На передачу данные вставляются в общий поток данных к соответствующему крейту, изменяются лишь поля, указывающие для какого слота предназначены эти данные. При приеме данных от крейта по соответствующему полю определяется, от какого модуля эти слова пришли, и эти слова передаются клиенту, установившему с модулем соединение, без изменений (или отбрасываются, если нет клиентов).

Прикладные программы устанавливают клиентские соединения с помощью функций библиотеки **ltrapi**.

В то время как управляющих соединений (как с **ltrd**, так и с крейтом) может быть несколько одновременно, то с каждым модулем в штатном режиме работы может быть только одно соединение клиента. В случае попытки соединиться с модулем, с которым уже установлено соединение, **ltrd** создаст соединение, но вернет предупреждение о том, что модуль уже используется. О том как обрабатывается данное предупреждение в **ltrapi** описано в [руководстве программиста для библиотеки ltrapi](#).

При необходимости сбросить все клиентские подключения, относящиеся к заданному модулю, чтобы можно было открыть новое соединение без данного предупреждения, можно выполнить сброс конкретного модуля:

- **ltrapi** — с помощью функции `LTR_ResetModule()` для установленного управляющего соединения с **ltrd**.

- **ltrctl**:

- Сброс 6 слота в первом найденном крейте:

```
ltrctl mrst 6
```

- Сброс 6 слота в крейте с серийным номером 1R234567

```
ltrctl mrst 6 1R234567
```

- **LTR Manager** — нажать правой кнопкой на нужный модуль и из контекстного меню выбрать “Сброс модуля”. Или выделить нужный модуль и в меню программы выбрать “Модуль -> Сброс модуля”

4.2 ltrapi

Для написания пользовательских программ, взаимодействующих со службой **ltrd**, крейтами и/или модулями предоставляется набор библиотек на языке C. Интерфейс кроссплатформенных библиотек полностью совместим с интерфейсом ранее созданной версии библиотек под ОС Windows. Это позволяет легко переносить имеющиеся программы из Под ОС Windows (конечно, в случае если сама программа не использует специфичное для Windows API для своих целей) и писать кроссплатформенное ПО, которое работает под любой из поддерживаемых ОС.

Старой версии только для Windows соответствуют библиотеки версий 1.27 и ниже (которые можно скачать по адресу http://www.lcard.ru/download/ltrdll_1_27.exe со старым описанием http://www.lcard.ru/download/ltrapi_v1.pdf). Данная версия в настоящее время не поддерживается и под всеми системами используется общие библиотеки более новых версий.

Набор библиотек состоит из базовой библиотеки с аналогичным названием **ltrapi**, в которой реализованы функции для создания клиентских соединений с **ltrd**, а также общие функции управляющих соединений. Подробное описание базовой библиотеки, включая общие принципы работы, описание всех функций и определений, можно прочитать в соответствующем руководстве программиста: <http://www.lcard.ru/download/ltrapi.pdf>.

Для работы с каждым модулем LTR созданы отдельные библиотеки с названиями **ltrXXXapi**, где XXX соответствует номеру модуля. Эти библиотеки используют внутри себя функции из базовой библиотеки для установления клиентского соединения и передачи данных. Поэтому конкретно для задачи работы с определенным модулем явное использование функций базовой библиотеки не требуется. Описание соответствующих библиотек содержится в документах **ltrXXXapi.pdf**, которые можно скачать из “Библиотеки файлов” или на странице соответствующего модуля в разделе “Документация”.

Подключение библиотек для ОС Windows расписано в документе “Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению”.

Для ОС Linux, FreeBSD и QNX6 библиотеки **ltrapi** представляет собой набор разделяемых библиотек (.so-файлы), которые должны устанавливаться вместе с программой. При этом используется разделение на файл самой библиотеки, имя которого включает полную версию и который необходим для работы использующей библиотеку программы, и на символьную ссылку без версии на первый файл, которая используется при компиляции. Это сделано для потенциальной возможности иметь несколько версий библиотеки на одной машине при необходимости.

Для QNX4 предоставляются статические библиотеки (с расширением .lib), которые включаются в код программы при компиляции.

Ниже описаны присутствующие отличия в поведении библиотек.

4.3 ltrctl

Утилита **ltrctl** предоставляет пользователю интерфейс командной строки для управления **ltrd** и получения от него нужной информации. Для выполнения своих действий **ltrctl** использует функции **ltrapi**.

Все действия, которые нужно выполнить, задаются с помощью команд, каждая из которых может иметь свой список аргументов.

ltrctl может работать в двух режимах:

- Можно указать команду с аргументами как параметры при запуске **ltrctl**. В этом случае **ltrctl** подключится к **ltrd**, выполнит действия, соответствующие команде, после чего разорвет соединение и завершит работу. Например для отображения списка ip-адресов, сохраненных в **ltrd**, можно написать в командной строке:

```
ltrctl iplist
```

- Если команда не указана в параметрах **ltrctl**, то **ltrctl** устанавливает соединение с **ltrd**, после чего переходит в режим ввода команд. Пользователь может ввести нужную команду и нажать клавишу “Ввод”. После выполнения команды **ltrd** предложит ввести новую (и так до того, как не будет введена команда `exit`). Все команды при этом используют одно соединение.

Полный список поддерживаемых команд с кратким описанием можно получить, вызвав:

```
ltrctl --help
```

Синтаксис команд описан по следующим правилам:

- Аргументы в треугольных скобках означают, что вместо них подставляются реальные значения параметра, в то время как слова без треугольных скобок используются как есть. Так “`ipcon <ip_addr>`” означает, что `<ip_addr>` должен быть заменен на реальный адрес, например “`ipcon 192.168.1.4`”.
- Аргументы в квадратных скобках являются опциональными. Например, “`ipadd <ip_addr> [auto] [perm]`” означает, что правильными вариантами являются “`ipadd <ip_addr>`”, “`ipadd <ip_addr> auto`”, “`ipadd <ip_addr> perm`” или “`ipadd <ip_addr> auto perm`”.
- Вертикальная черта между двумя словами означает, что может использоваться одно из двух слов. Например “`ipcon <ip_addr> | auto`” означает, что действительным вариантом вызова является либо “`ipcon <ip_addr>`”, либо “`ipcon auto`”.

Часть настроек **ltrd** можно изменить либо только на время текущей сессии (в этом случае изменения сбросятся при следующем запуске **ltrd**), либо могут быть сразу сохранены в [конфигурационный файл](#). Для сохранения в файл необходимо указать аргумент `perm` (от `permanently`).

ltrctl по умолчанию устанавливает соединение с **ltrd**, запущенном на том же компьютере. Однако при желании можно установить соединение и с удаленным **ltrd**, указав при вызове **ltrctl** ключ `--ip-addr`. Например для соединения с **ltrd**, запущенном на машине с адресом 192.168.1.10 нужно вызвать:

```
ltrctl --ip-addr=192.168.1.10
```

4.3.1 Команды управления ip-адресами

- Вывод информации по всем записям IP-адресов:

```
ltrctl iplist
```

По каждой ip-записи выводится ее адрес, состояние (установлено ли соединение с крэйтом с этим адресом), серийный номер соответствующего крэйта (если соединение установлено) и установлен ли флаг [автоподключения](#) и флаг [повторного подключения](#).

- Посмотреть список адресов по указанной маске:

```
ltrctl iplist <ip_addr> <msk>
```

Выводится та же информация, что и по предыдущей команде, но только для адресов, которые при наложении маски <msk> совпадают с адресом <ip_addr>. Например, можно использовать, чтобы вывести адреса только из определенной подсети.

- Добавление IP-адреса только на время текущей сессии:

```
ltrctl ipadd <ip_addr>
```

- Добавление IP-адреса только на время текущей сессии с установленным флагом [автоподключения](#):

```
ltrctl ipadd <ip_addr> auto
```

- Добавление IP-адреса только на время текущей сессии с установленным флагом [повторного подключения](#):

```
ltrctl ipadd <ip_addr> reconnect
```

- Добавление IP-адреса с сохранением в файл конфигурации:

```
ltrctl ipadd <ip_addr> perm
```

- Добавление IP-адреса с установленным флагом [автоподключения](#) с сохранением в файл конфигурации:

```
ltrctl ipadd <ip_addr> auto perm
```

- Добавление IP-адреса с установленным флагом [повторного подключения](#) с сохранением в файл конфигурации:

```
ltrctl ipadd <ip_addr> reconnect perm
```

- Удаление записи с указанным IP-адресом только для текущей сессии:

```
ltrctl iprem <ip_addr>
```

С крэйтом не должно быть установлено клиентское соединение.

- Удаление записи с указанным IP-адресом с сохранением изменения в файле:

```
ltrctl iprem <ip_addr> perm
```

С крейтом не должно быть установлено клиентское соединение.

- Установка флага **автоподключения** для уже добавленной записи с IP-адресом с сохранением в файле конфигурации (без perm – только для сессии):

```
ltrctl ipflag <ip_addr> auto set perm
```

- Сброс флага **автоподключения** для уже добавленной записи с IP-адресом с сохранением в файле конфигурации (без perm – только для сессии):

```
ltrctl ipflag <ip_addr> auto clear perm
```

- Установка флага **повторного подключения** для уже добавленной записи с IP-адресом с сохранением в файле конфигурации (без perm – только для сессии):

```
ltrctl ipflag <ip_addr> reconnect set perm
```

- Сброс флага **повторного подключения** для уже добавленной записи с IP-адресом с сохранением в файле конфигурации (без perm – только для сессии):

```
ltrctl ipflag <ip_addr> reconnect clear perm
```

4.3.2 Команды управления подключениями по ip-адресам

- Запуск подключения по указанному ip-адресу.

```
ltrctl ipcon <ip_addr>
```

Адрес должен быть уже добавлен до этого с помощью **ipadd**. Проверить, когда подключение завершится и с каким результатом по статусу ip-адреса с помощью команды **iplist**

- Установить подключение со всеми адресами, которые помечены флагом автоподключения

```
ltrctl ipcon auto
```

С уже подключенными крейтами соединение сохраняется. Если выполняется подключение, то оно перезапускается.

- Разорвать соединение по указанному ip-адресу.

```
ltrctl ipdisc <ip_addr>
```

- Разорвать соединение со всеми крейтами, с которыми установлено или находится в процессе установления соединение

```
ltrctl ipdisc all
```

4.3.3 Получение информации по активным крейтам и модулям

- Вывод списка всех **активных крейтов** с составом модулей:

```
ltrctl clist
```

- Вывод статистики по первому активному крейту

```
ltrctl cstat
```

- Вывод статистики по активному крейту с указанным серийным номером

```
ltrctl cstat <serial>
```

- Вывод статистики по модулю в указанном слоте (начиная с 1) в первом найденном крейте.

```
ltrctl mstat <slot>
```

<slot> — число от 1 до 16

- Вывод статистики по модулю в указанном слоте (начиная с 1), находящимся в крейте с указанным серийным номером.

```
ltrctl mstat <slot> <serial>
```

<slot> — число от 1 до 16

4.3.4 Управление параметрами ltrd

- Установить уровень журнала с сохранением в файл (без perm — для текущей сессии):

```
ltrctl setloglvl <lvl> perm
```

<lvl> — число от 0 до 7, обозначающее уровень, сообщения с уровнем выше которого (с меньшим приоритетом) не будут выводиться. Уровень 0 соответствует выводу только сообщений о критических ошибках, 7 — вывод всех сообщений, включая отладочные. Типичный уровень 3 — вывод основных сообщений о ходе работы и всех предупреждений/ошибок.

- Получение текущего уровня журнала

```
ltrctl getloglvl <lvl>
```


4.3.5 Дополнительные команды

- Сброс модуля в указанном слоте в первом найденном крейте:

```
ltrctl mrst <slot>
```

Модулю посылается команда сброса и обнуляется вся статистика.

- Сброс модуля в указанном слоте в крейте с указанным серийным номером:

```
ltrctl mrst <slot> <serial>
```

- Перезапуск **ltrd** с разрывом всех соединений

```
ltrctl restart
```

- Запрос на завершение работы **ltrd**.

```
ltrctl shutdown
```

4.4 LTR Manager

LTR Manager представляет собой графическую утилиту для управления **ltrd** и вывода информации о подключенных крейтах.

LTR Manager не доступен под все поддерживаемые ОС, однако при желании можно запустить данную программу с компьютера с ОС **Linux** или **Windows** для настройки и мониторинга статистики удаленного (запущенного на машине с другой ОС) демона **ltrd**.

На [рисунке 4.1](#) приведен скриншот графического интерфейса **LTR Manager**.

Основные отличия **LTR Manager** от ранее используемой программы **LTR Server**:

- **LTR Manager** является лишь интерфейсом к службе **ltrd** и служит для просмотра статистики и управления. При выходе из **LTR Manager** служба **ltrd** продолжает работать и все соединения сохраняются. Так и для работы с крейтами запуск **LTR Manager** не обязателен.
- **LTR Manager** может быть соединен с удаленным **ltrd** и даже с несколькими **ltrd** одновременно.
- **LTR Manager** позволяет выполнять некоторые дополнительные служебные операции (настройка крейта, обновление прошивки, сброс модулей, управление синхронизацией крейта и т.д.)

Интерфейс представляет собой основное окно и набор панелей, которые могут быть размещены в нужном месте основного окна, вынесены в качестве отдельных окон или быть скрыты по желанию пользователя (видимостью можно управлять или из меню “Окна” или из панели инструментов).

Основное окно показывает дерево, в котором для каждого **ltrd**, с которым установлено соединение, показан список активных крейтов с составом модулей. Для каждого крейта есть иконка, соответствующая интерфейсу, по которому установлено соединение

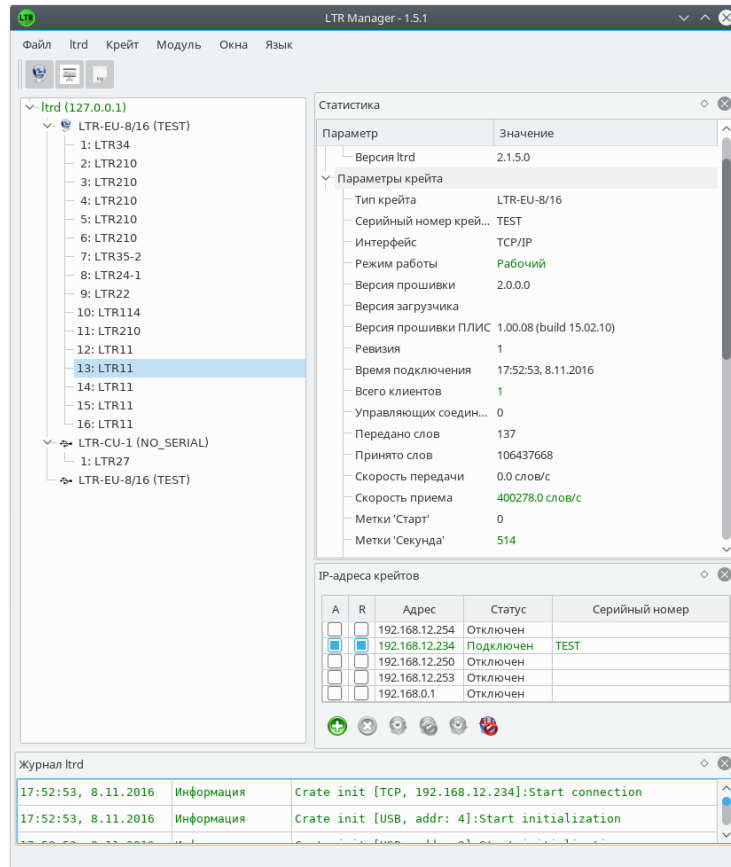


Рис. 4.1: Интерфейс программы **LTR Manager**, запущенной под рабочим столом KDE5

с крейтом. Содержимое остальных панелей зависит от выделенного элемента в этом дереве. Каждый узел можно свернуть по желанию.

Панель статистики показывает информацию по выделенному в основном окне соединению с **ltrd**, крейту и модулю. При наведении курсора на нужный параметр и его удержании появляется подсказка, более подробно описывающая, что означает данный параметр.

Панель “IP-адреса” отображает таблицу со всеми записями IP-адресов крейтов для выбранного соединения с **ltrd** и позволяет выполнять все доступные действия с IP-адресами, описанные в [разделе 4.1.4](#). При наведении мыши на заголовок столбца всплывает подсказка с описанием столбца. Для выполнения действия можно выделить нужную строку с IP-адресом (если действие относится к конкретной записи) и нажать соответствующую нужному действию кнопку под таблицей (название действие выводится в подсказке при наведении мыши на кнопку). Также можно это сделав, нажав правой кнопки мыши по нужной записи для действия, относящегося к конкретной записи, или по любому месту в таблице для общих действий и выбрав нужное действие из контекстного меню. Значение флагов “Автоподключения” и “Повторного подключения по ошибке” можно изменить снимая или устанавливая галочку в столбце “A” или “R” соответственно. Двойное нажатие левой кнопкой мыши на строке с IP-адресом приводит к изменению состояния подключения по данному адресу (началу установки подключения по адресу, если не установлено соединение, и к разрыву, если соединение установлено или идет процесс установки).

Панель “Журнал ltrd” отображает сообщения журнала **ltrd**, соответствующего выбранному **ltrd**. Уровень сообщений журнала отображается цветом сообщения. Через

контекстное меню (по правой кнопке мыши) можно очистить принятые программой **LTR Manager** сообщения журнала, а также поменять уровень отображаемых сообщений.

Важно!: Следует учитывать, что существует две настройки уровня журнала. Одна из них — это настройка **ltrd**, которая определяет, сообщения какого уровня будет выдавать сама служба **ltrd** (уровень учитывается на момент возникновения сообщения, и сообщения более низкого уровня будут отброшены). Вторая — настройка **LTR Manager**, которая определяет, какие сообщения из принятых (уже отфильтрованных по первому уровню) от **ltrd** будут отображаться. При этом изменение уровня приводит изменению состава уже отображенных сообщений. Естественно, установка уровня журнала в **LTR Manager** выше (на разрешение менее критических сообщений), чем настройка **ltrd**, не приведет к появлению дополнительных сообщений.

При закрытии основного окна **LTR Manager** сворачивается и остается в трее (это соответствует привычному режиму работы программ, способных сворачиваться в трей). Для полного закрытия нужно в меню выбрать “Файл -> Закрыть”.

4.4.1 Настройка интерфейса крейтов LTR-EU/LTR-CEU

Программа **LTR Manager** позволяет настроить рабочий интерфейс и его параметры для крейтов, которые поддерживают интерфейс Ethernet (LTR-EU, LTR-CEU). Для LTR-EU крейт должен быть подключен по USB, независимо от используемого интерфейса (если крейт настроен на USB, то будут видны все модули, если на Ethernet — то модули отображаться не будут, а состояние крейта будут “Только настройка”). Для крейтов LTR-CEU изменение настроек возможно и при подключении по Ethernet.

Для изменения настроек необходимо нажать правой кнопкой мыши по нужному крейту и выбрать пункт меню “Настройки крейта...”, после чего появится диалоговое окно, изображенное на [рисунке 4.2](#). В данном окне будут отображены текущие записан-

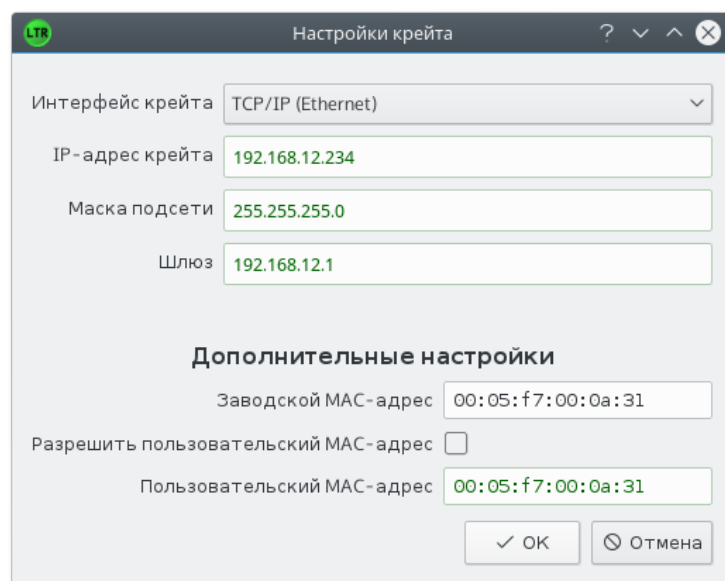


Рис. 4.2: Диалог настройки интерфейса крейтов

ные в энергонезависимую память настройки крейта.

Необходимо выбрать нужный интерфейс и для TCP/IP выбрать IPv4-адрес крейта, маску сети и адрес шлюза, после чего нажать “Ок” и подтвердить запись настроек. Реально настройки вступят только после сброса крейта. Для LTR-EU это необходимо сделать либо нажатием кнопки сброс на его передней панели, либо отключением-включением питания, о чем после записи настроек будет выведено сообщение. Для LTR-CEU существует программная команда сброса и **LTR Manger** может выполнить сам сброс крейта после изменения настроек (что будет предложено).

Для крейтов LTR-CEU изменение настроек может быть также защищено простым паролем. Если пароль не задан, то при изменении настроек можно ничего не вводить в запросе и сразу нажать “Ок”. Это сделано, чтобы можно было защитить крейт от удаленного изменения настроек, т.к. LTR-CEU поддерживает их изменение не только по USB. При этом если пароль забыт, то изменить настройки можно по интерфейсу USB, введя в качестве пароля серийный номер крейта.

Следует отметить, что при входе в диалог настройки, **LTR Manager** вычитывает текущие настройки в памяти крейта (которые могут еще не вступить в силу, если не было сброса крейта с момента последнего изменения) и отображает их в полях диалога, то есть это можно использовать и для просмотра текущих настроек крейта без их изменений.

4.4.2 Управление настройками синхронизации крейтов

Начиная с версии 1.5.0 в программу добавлена возможность управления синхронизацией (генерацией синхрометок) для крейтов LTR-EU, LTR-CEU и LTR-CU.

Данный механизм описан в главе “Принцип синхронизации сбора данных в системе LTR” [руководства пользователя LTR](#).

Для настройки генерации синхрометок необходимо нажать правой кнопкой мыши по нужному крейту и выбрать пункт меню “Управление синхронизацией крейта...”, в результате чего появится диалоговое окно, изображенное на [рисунке 4.3](#).

В данном окне доступны те же настройки управления генерацией синхрометок, что и предоставляют функции `ltrap1`. Эти настройки могут выполняться автоматически законченным ПО, управляющим синхронным сбором данных с нескольких модулей. Однако для тех случаев, когда может быть удобна ручная настройка генерации синхрометок (если эти функции не включены в законченное ПО, необходимо управлять синхронизацией модулей, когда с каждым работает своя программа, при разработке собственных программ и др.), введена дополнительно и такая возможность посредством данного диалога.

Открытие данного диалога не запрещает дальнейшую работу с **LTR Manger** и при необходимости можно открыть несколько таких диалоговых окон, каждое для своего крейта.

Следует учесть, что так как прошивка крейта позволяет только устанавливать настройки генерации синхрометок, но не позволяет считывать их, а данные настройки могут устанавливаться параллельно и из других программ, то нет возможности отобразить текущие настройки крейта. Соответственно, отображаемые значения при открытии диалога не связаны с реальными настройками крейта.

Данный диалог позволяет выполнить следующие настройки:

- Установить, какие сигналы будут транслироваться на выходы DIGOUT1 и DIGOUT2 разъема синхронизации. Для этого нужно заполнить параметры в разделе “Выходы разъема SYNC” и нажать кнопку “Установить” этого раздела.

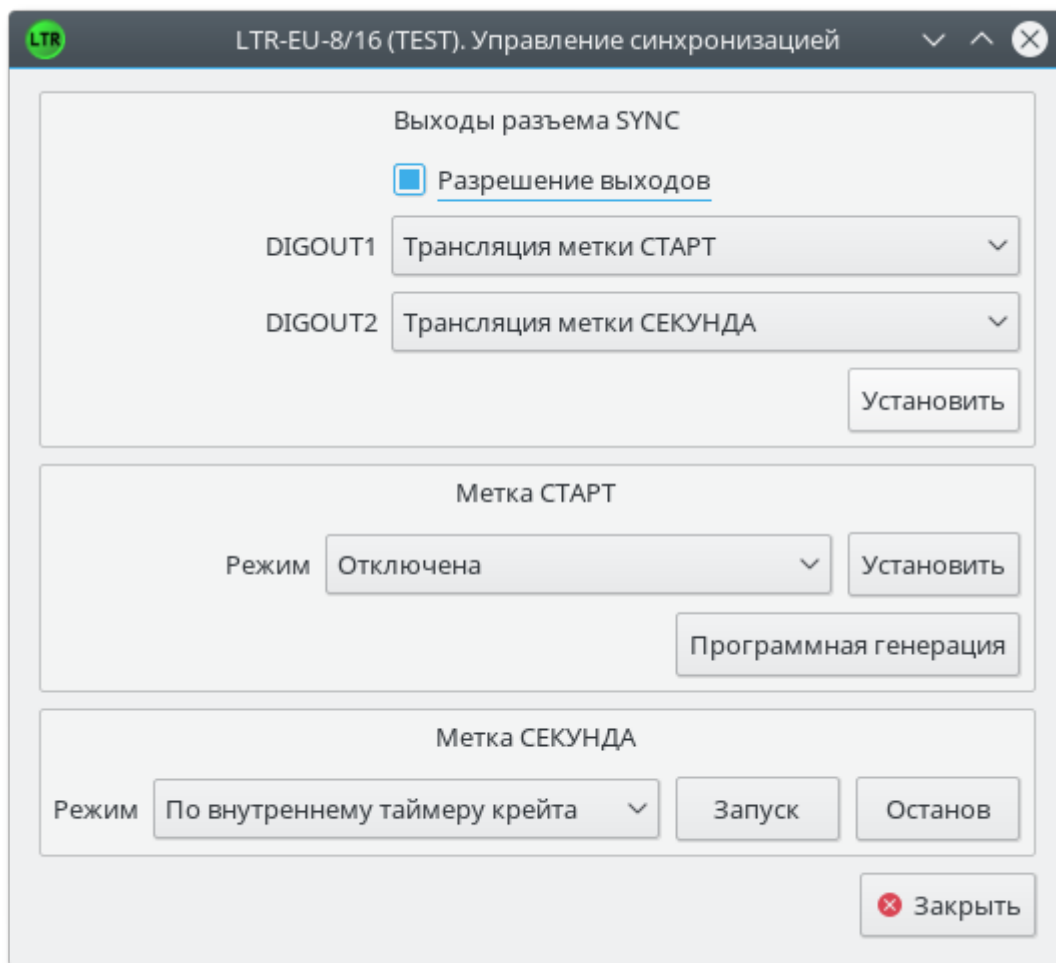


Рис. 4.3: Диалог управления настройками синхронизации крейтов

Если “Разрешение выходов” не отмечено, то выходы будут всегда в третьем состоянии, иначе на них будет транслироваться сигнал в соответствии с выбранной для каждого выхода функцией. В частности это может использоваться для трансляции синхрометок, которые генерирует крейт. Соответственно, сигналы с этих выходов могут использоваться как источники синхрометок для других крейтов.

- Установить режим генерации метки “СТАРТ” от внешнего источника. Для чего нужно выбрать в разделе “Метка СТАРТ” режим генерации метки и нажать кнопку “Установить” справа от режима. После этого крейт будет ждать указанного события и по каждому такому событию генерировать метку “СТАРТ”. Если генерация метки по внешнему событию больше не нужна, то нужно выбрать режим “Отключена” и нажать “Установить”.
- Выполнить ручную генерацию одиночной метки “СТАРТ”. По нажатию кнопки “Программная генерация” раздела “Метка СТАРТ” крейт сгенерирует одну метку СТАРТ (вне зависимости от текущих настроек), после чего крейт перестанет ожидать каких-либо событий генерации метки “СТАРТ”, т.е. перейдет в режим “Отключена” вне зависимости от того, на что был настроен до этого.
- Запустить генерацию секундной метки в одном из режимов. Для этого нужно

установить в разделе “Метка СЕКУНДА” нужный режим и нажать “Запуск”. Если установлен режим “По внутреннему таймеру крейта”, то крейт начнет сразу генерировать метку раз в секунду. Если выбран другой режим, то крейт начинает ожидать соответствующего внешнего события и генерировать метку “СЕКУНДА” по каждому событию выбранного типа.

- Остановить генерацию секундной метки. Для этого нужно нажать кнопку “Останов” в разделе “Метка СЕКУНДА”.

4.4.3 Обновление прошивки процессора

Программа **LTR Manager** позволяет выполнить обновление прошивки процессора для крейтов LTR-EU (только по USB) и LTR-CEU (как по USB, так и Ethernet). Для этого необходимо нажать правой кнопкой мыши по нужному крейту и выбрать пункт меню “Обновление прошивки крейта”. После чего необходимо выбрать файл с прошивкой процессора и он будет загружен. Для крейтов LTR-EU после этого необходимо также вручную выполнить сброс крейта (как и при обновлении настроек), а для LTR-CEU прошивка вступит в силу сразу после обновления.

Скачать последние версии прошивок можно со страницы соответствующего крейта на [сайте](#) в разделе “Программное обеспечение” -> “Firmware и BIOS”.

Важно!: Для крейтов LTR-EU следует явно учитывать, что прошивка для LTR-EU-2 отличается от прошивок LTR-EU-8/16. Необходимо убедиться, что зашивается прошивка, соответствующая типу крейта, т.к. в отличие от LTR-CEU нет явной защиты от записи неверной прошивки крейта

Важно!: При обновлении прошивки LTR-CEU по USB устройство переходит в режим загрузчика (и видится как “Загрузочное устройство”, а не крейт), в котором остается до завершения обновления. Также в случае, если не было обращений за заданное время, крейт также возвращается в рабочий режим. В ОС Windows при первом обновлении может быть запрошен драйвер для загрузочного устройства и его обновление может занять больше времени, чем время ожидания появления устройства в **LTR Manger** и тогда прошивка завершится с ошибкой. В этом случае нужно установить драйвер, дождаться, пока крейт вернется в рабочий режим (появится в **LTR Manger**) и повторить обновление. Под ОС Linux работа с загрузочным устройством от непривилегированного пользователя требует своих правил udev, которые устанавливаются с утилитой **Iboot** автоматически при установке пакета, которая используется в этом случае для обновления.

4.4.4 Работа с удаленным ltrd

По умолчанию при запуске программа **LTR Manager** устанавливает соединение с локальным демоном **ltrd** (помечен адресом “127.0.0.1”), что и требуется во многих случаях. Однако есть возможность из одной графической оболочки управлять **ltrd**, запущенными на других машинах. Можно добавить адреса удаленных экземпляров **ltrd**, подобно тому, как добавляются адреса крейтов в **ltrd**. Для этого служит меню “ltrd->Добавить новое соединение с ltrd”. В появившемся меню следует установить флаг “Удаленное подключение” и ввести адрес. Как и для ip-адресов крейтов можно

указать, нужно ли соединиться с **ltrd** по этому адресу автоматически при последующем запуске **LTR Manger**.

Примечание: Следует иметь ввиду, что список адресов **ltrd** является настройкой самой программы **LTR Manager**, в то время как список адресов крейтов является настройкой каждого **ltrd** и **LTR Manager** только предоставляет интерфейс для их изменения

Новое соединение появится новым узлом верхнего уровня в дереве, отображающим список крейтов и модулей (каждый крейт принадлежит узлу **ltrd**, соответствующему тому, какой экземпляр **ltrd** поддерживает активное соединение с этим крейтом).

При работе с несколькими **ltrd** следует учитывать, что информация, отображаемая на панели “IP Адреса крейтов” и пункты меню “ltrd” относятся именно к выделенному сейчас соединению с **ltrd**.

4.5 Дополнительные утилиты

4.5.1 Утилиты настройки и обновления прошивок крейтов LTR-EU

Для установки настроек интерфейса и обновления прошивок BlackFin или ПЛИС в состав `ltr_cross_sdk` включены утилиты командной строки **ltreu-config** и **ltreu-firm-update** соответственно. Для выполнения данных операций требуется, чтобы крейт был подключен к ПК по USB.

При этом, даже когда крейт настроен на работу по Ethernet, есть возможность подключить его по USB интерфейсу для выполнения операций настройки и обновления прошивки.

Также данные возможности доступны из графического интерфейса программы **LTR Manager**.

Важно отметить, что в отличие от ранее используемых утилит под ОС Windows, данные утилиты работают через **ltrd**, а не напрямую с крейтом, что позволяет не останавливать работу **ltrd**. При необходимости, обе утилиты могут даже работать с удаленно запущенным **ltrd**, для чего нужно передать адрес машины, на которой запущен **ltrd**, с помощью опции `--srv-addr`.

Обе программы по умолчанию устанавливают соединение с первым крейтом LTR-EU, который подключен по USB. Но при этом есть возможность явно указать требуемый крейт по серийному номеру с помощью опции `--csn`.

Обе утилиты требуют после успешного выполнения операций снять и снова подключить питание крейта, чтобы изменения вступили в силу.

Ниже будет описаны основные варианты использования.

4.5.1.1 Установка настроек интерфейса

Для установки настроек интерфейса используется утилита **ltreu-config**.

Для установки следует обязательно указать интерфейс, по которому крейт должен работать, с помощью опции `--intf` или `--iface` (это просто два имени для одной и той же опции). Доступные интерфейсы:

- `usb` — работа с крейтом по USB-интерфейсу

- eth — работа с крейтом по Ethernet-интерфейсу с использованием протокола поверх TCP/IP.

Для USB-интерфейса больше ничего указывать не нужно, в то время как для Ethernet требуется обязательно указать IP-адрес самого крейта (опция `--ip-addr`), маску подсети (опция `--ip-mask`) и адрес шлюза в сети (опция `--gate`).

Таким образом, настройка крейта на работу по USB-интерфейсу выглядит следующим образом:

```
ltreu-config --intf=usb
```

Для настройки крейта на работу по Ethernet интерфейсу с адресом крейта 192.168.1.10, маской 255.255.255.0 и адресом шлюза 192.168.1.1, нужно вызвать:

```
ltreu-config --intf=eth --ip-addr=192.168.1.10 --ip-mask=255.255.255.0
--gate=192.168.1.1
```

Для того, чтобы изменения вступили в силу, требуется снять питание крейта и снова подключить его.

4.5.1.2 Чтение текущих настроек интерфейса

Вызов утилиты `ltreu-config` без параметра `--intf` или `--iface` приводит к чтению текущих настроек модуля и вывода их на экран.

4.5.1.3 Обновление прошивок крейта

Для обновления прошивок используется утилита `ltreu-firm-update`. Она позволяет записать в крейт новую прошивку как для сигнального процессора BlackFin (указывается опцией `--dsp-firm`), так и для ПЛИС (опция `--fpga-firm`). Что будет записываться определяется наличием указанных опций (можно указать обе для обновления и прошивки BlackFin, и ПЛИС).

Важно!: Важно следить, что в крейт записываются прошивки, предназначенные именно для того типа крейта, который обновляется. В противном случае крейт может стать неработоспособен!

Для дополнительной проверки типа крейта необходимо явно указывать тип крейт-контроллера, для которого предназначена прошивка, с помощью опции `--crate-type`. Возможные варианты типа крейт-контроллера:

- ltr030 — для крейта LTR-EU-8/16
- ltr031 — для крейта LTR-EU-2

Имена штатных прошивок всегда начинаются с названия крейт-контроллера, для которого они предназначены.

Следует также отметить наличие опции `--all`, при которой `ltreu-firm-update` выполняет последовательное обновление прошивки для всех крейтов указанного типа, подключенных по USB интерфейсу.

При установке утилит из пакетов можно воспользоваться скриптами `ltr030-firm-update-all.sh` и `ltr031-firm-updata-all.sh` для обновления последними прошивками всех крейтов LTR-EU-8/16 или LTR-EU-2 соответственно.