Многоканальные системы сбора данных

Крейтовая система LTR

Начиная работать с крейтовой системой LTR. Вопросы по программному обеспечению

Ревизия 1.0.6

Февраль 2021

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	29.05.2016	Первая ревизия данного документа
1.0.1	27.07.2016	Добавлена информация о `FreeBSD` и в нужных местах дополнительные ссылки на ltr_cross_sdk.
1.0.2	04.08.2016	В главе про **LabView** сделана ссылка на раздел *C#*, в котором находится информации по используемым классам.
1.0.3	10.04.2017	Добавлена ссылка на OPC-сервер в перечне законченного ПО.
1.0.4	19.05.2017	Добавлена информация о использовании пакета **NuGet** при написании программ на *C/C++* в **Microsoft Visual Studio** (раздел 4.3).
1.0.5	21.09.2020	Исправление ссылки на исходные коды.
1.0.6	10.02.2021	Добавлена ссылка на программу **L Card Measurement Studio** в перечне законченного ПО.

Содержание

1 О чем этот документ

В данном документе рассматриваются общие вопросы по программному обеспечению (ПО) с ссылками на более подробные документы. Этот документ предназначен, чтобы дать общее представление о необходимом и существующем ПО как для конечных пользователей системы, так и для программистов. Для более детальных сведений после прочтения данного документа можно обратится к документу ltr_cross_sdk.pdf. Также для успешной работы с крейтами и модулями LTR необходимо изучить руководство пользователя LTR (http://www.lcard.ru/download/ltr.pdf).

2 Предоставляемое программное обеспечение

Штатная работа крейтов LTR предполагает, что управление крейтами, модулями, сбором и обработкой данных осуществляет ПО верхнего уровня, запущенное на персональном компьютере (ПК). О данном ПО главным образом и пойдет речь в данном документе. Штатный вариант ПО крейта не предполагает автономной работы крейта (без ПК). В случае необходимости автономной работы для крейтов LTR-EU и LTR-CEU может быть создана специализированная прошивка крейта, о чем отдельно рассказано в разделе 4.9. ПО, предоставляемое "Л Кард", бесплатное и в большинстве случаев полностью открытое, т.е. предоставляются исходные коды программ (см. раздел 4.8).

2.1 Какие операционные системы поддерживаются

Для работы ПО, управляющего крейтами, необходим ПК с одной из поддерживаемых операционных систем (ОС). Поддерживаются следующие ОС:

1. Все версии `Windows`, начиная с `Windows XP`.
2. Современные дистрибутивы `Linux`. Для некоторых дистрибутивов распространяются собранные пакеты.
3. `FreeBSD`, начиная с версии 9.3 или выше
4. ОС реального времени `QNX4`, `QNX6`

   Установка ПО под Windows описана в разделе 2.2. Установка для остальных ОС описана в документе ltr_cross_sdk.pdf.

   Примечание: Системные требования в данном случае не указаны, так как они определяются в первую самим ПО верхнего уровня, обрабатываемыми потоками данных и т.п. В простейшем случае системные требования соответствуют требованиям используемой ОС, а также конечно необходимо наличие у ПК требуемого интерфейса (USB или Ethernet).

2.2 Какое ПО необходимо установить для работы под ОС Windows

1. Драйвер USB. Драйвер входит в состав библиотеки **lcomp** (http://www.lcard.ru/download/lcomp.exe), поэтому для работы с крейтами по USB необходимо установить данную библиотеку.

2. Служба (service) **ltrd** (http://www.lcard.ru/download/ltrd-setup.exe). После установки данная служба автоматически запускается вместе со стартом ПК и работает в фоне "незаметно" для пользователя. Она является необходимой для работы с крейтами, так как все штатное ПО работает через нее. Зачем необходима такая служба описано в разделе 2.4.

3. Программа **LTR Manager** (http://www.lcard.ru/download/ltrmanager_setup.exe) — вспомогательная программа с графическим интерфейсом, которая отображает информацию о подключенных крейтах и статистику работы с ними, отображает журнал работы службы **ltrd**, позволяет выполнять настройку крейтов для работы по Ethernet, управлять подключениями крейтов по Ethernet, обновлять прошивку крейтов и выполнять другие служебные действия.

4. Набор библиотек для работы с модулями LTR (http://www.lcard.ru/download/ltrdll.exe). Установка этого набора библиотек необходима для программ, которые используют установленную системную версию библиотек. Часть программ (например **LGraph2**) во время установки устанавливают локальную копию данных библиотек, поэтому для них отдельно устанавливать библиотеки может быть не нужно. Также с библиотеками устанавливаются файлы, необходимые для разработки пользователями своих программ на ПК.

Подробное описание о данном программном обеспечении (включая службу **ltrd**, программу **LTR Manager**, а также дополнительные служебные программы) можно найти в документе ltr_cross_sdk.pdf.

Примечание: Ранее роль программ **ltrd** и **LTR Manager** выполняла программа **LtrServer**, особенности которой описаны в разделе 2.5). В настоящее время данная программа не поддерживается. Пользователям, использующим **LtrServer**, рекомендуется перейти на использование службы **ltrd**, которая совместима по программному интерфейсу с **LtrServer**, в связи с чем программы, работающие с **LtrServer**, должны работать и с **ltrd**, если не используют какие-то редкие специфические особенности **LtrServer**. В случае возникновения проблем при переходе, необходимо обратиться в техподдержку (см. раздел 5)

2.3 Какое существует законченное ПО для работы с модулями LTR

"Л Кард" предоставляет следующие варианты законченного ПО:

1. **L Card Measurement Studio** (https://www.lcard.ru/products/software/lms). Программный комплекс, предназначенный для проведения измерительных экспериментов и решения задач автоматизации на базе оборудования компании "Л Кард". Требует приобретения лицензионного ключа. Доступна бесплатная демонстрационная версия.

2. **OPC-Сервер "Л Кард"** (https://www.lcard.ru/products/software/opc). Программа позволяет использовать крейтовую систему LTR в SCADA-системах и других программных комплексах с поддержкой OPC. Требует приобретения лицензии. Доступна бесплатная демонстрационная версия.

3. **LGraph2** (https://www.lcard.ru/products/software/lgraph) — бесплатная программа самописец-визуализатор для большинства устройств "Л Кард". Следует учитывать, что программа поддерживает не все модули LTR и может поддерживать не все их возможности. Подробную информацию о данной программе можно найти в руководстве пользователя для **LGraph2** (http://www.lcard.ru/download/lgraph2_help.pdf)

4. Программа **UTS** (http://www.lcard.ru/download/uts.zip) — тестовая программа, предназначена для ознакомления и проверки возможностей модуля. Для большинства модулей поддерживает все возможные программные настройки.

5. Специализированные программы для модулей — для некоторых модулей вместо (или в дополнение) **UTS** могут использоваться свои специализированные демонстрационные программы, например:

   - **ltr210-osc** (http://www.lcard.ru/download/ltr210-osc-install.exe) — демонстрационная программа осциллограф для модуля LTR210.
   - **LTR35Gen** (http://www.lcard.ru/download/ltr35gen.zip) — программа для генерации сигналов стандартной формы для модуля LTR35.

   Полный список ПО можно всегда посмотреть в разделе "Программное обеспечение" на странице каждого модуля.

6. Метрологическое программное обеспечение для выполнения поверки модулей LTR (http://www.lcard.ru/download/ltr_metr_setup.exe). Это мультиметры для модулей АЦП и программы для выставления сигналов стандартной формы для модулей ЦАП.

7. Клиент может заказать разработку специализированного законченного ПО для конкретной его задачи в "Л Кард", для чего может сделать запрос через отдел продаж "Л Кард" (см. раздел "Контакты" сайта: http://www.lcard.ru/contakt)

2.4 Для чего нужна служба ltrd

В отличие от других модулей и плат "Л Кард", в одном крейте LTR может находится до 16 модулей. При этом данные от всех модулей одного крейта передаются общим потоком по одному каналу связи. Для того, чтобы пользовательские программы могли работать с каждым модулем независимо (с разными модулями в одном крейте можно работать из разных программ), необходима программа, которая бы разбирала поток данных от крейта по модулям и распределяла их между клиентами. Именно эта задача и возложена на службу **ltrd**. Таким образом, **ltrd** устанавливает связь с самим крейтом, а прикладное ПО (клиент) устанавливает связь уже с **ltrd**, указывая крейт и номер слота в крейте, в котором находится модуль, с которым необходимо работать. **ltrd** принимает данные от крейта, определяет какому модулю они соответствуют и передает их нужным клиентам, а также принимает данные от клиентов, объединяет в общий поток для передачи и передает в крейт.

Важно!: Служба **ltrd** необходима как при работе с крейтом по интерфейсу USB, так и Ethernet.

Важно!: Несмотря на то, что в одноместном крейте может быть только один модуль, протокол работы с этим модулем аналогичен протоколу для многоместных крейтов.

Поэтому даже при работе с одноместным крейтом все равно необходима запущенная служба **ltrd**. Одноместный крейт это все равно крейт, а не отдельный модуль!

Дополнительно введение фоновой программы **ltrd** дает следующие преимущества:

1. Так как связь с крейтом обеспечивает **ltrd**, а прикладное ПО не работает с крейтом напрямую, то прикладное ПО работает одинаково как с разными типами крейтов, так и с крейтами, подключенными по разным интерфейсам. Т.е. ПО, которое было написано при работе с определенным типом крейта по определенному интерфейсу, автоматически может работать и с другими типами крейтов, подключенными по другому интерфейсу (если не делать искусственных ограничений в ПО).

2. Так как соединение с **ltrd** прикладным ПО выполняется через сокеты, то при желании (если ПО написано соответствующим образом) прикладное ПО и **ltrd** могут работать на разных машинах, соединенных по сети. Это можно, например, использовать для управления по сети крейтами, подключенными по USB к удаленной машине.

3. В **ltrd** включены команды управления IP-адресами для подключения крейтов по Ethernet.

4. В **ltrd** реализован сбор статистики для мониторинга за состоянием крейтов.

5. **ltrd** ведет журнал, в котором фиксируется информация о работе службы, изменении состояния подключения крейтов, возникновении ошибок при работе с крейтом и т.д.

2.5 Программа LtrServer и ее различия с ltrd

Программа **LtrServer** использовалась ранее до появления **ltrd** для выполнения тех же функций, но была выполнена как пользовательская программа с графическим интерфейсом (функции которого перенесены в **LTR Manager**), которую необходимо было вручную запускать для работы с крейтами LTR.

Важно!: В настоящее время программа может быть использована, но более не поддерживается “Л Кард”. Пользователям рекомендуется переход на использование службы **ltrd**.

Важно!: При использовании программы **LtrServer** для `Windows 8` и выше необходимо установить в свойствах программы режим совместимости с `Windows XP`

В данном разделе далее для пользователей, которые использовали **LtrServer**, описаны основные различия между **LtrServer** и **ltrd**.

ltrd был создан как кроссплатформенная (работающая не только под `Windows`, но также на текущий момент и под `Linux` и `QNX`) замена программы **LtrServer**. Помимо возможности запуска под разными операционными системами, существуют следующие ключевые различия в работе этих программ:

1. **ltrd** под `Windows` изначально создан как служба, что позволяет запускать его вместе с системой (на что и настраивает систему установщик), причем даже на серверных системах без входа пользователя. Это позволяет работать без ручного запуска каких-либо дополнительных программ. **LtrServer** сделан как обычная графическая программа Windows, которая может сворачиваться в трей.

2. Сама программа **ltrd** не предоставляет графический интерфейс. Для реализации возможностей интерфейса (аналогичных **LtrServer**) используются программа **LTR Manager**, которая подключается к **ltrd** и предоставляет информацию о подключенных крейтов и возможность управлять ими, аналогично **LtrServer**. При этом для работы пользовательского ПО запуск **LTR Manager** не требуется. Есть также утилита командной строки **ltrctl** для выполнения сходных действий.

3. **ltrd** вместе с **LTR Manager** предоставляют удобный способ смены настроек интерфейса крейта (USB или Ethernet, а также настройка IP-адреса для Ethernet) через графический интерфейс. При этом **LTR Manager** позволяет также посмотреть текущие настройки, не изменяя их. При работе через **LtrServer** для смены настроек необходимо закрыть программу, после чего необходимо воспользоваться специальными программами командной строки для настройки, которые можно скачать с сайта по адресу: `http://lcard.ru/download/ltr030config.zip`. Подробнее об использовании этих программ написано в `readme.txt` из указанного архива.

4. Программа **LtrServer** не различает режим работы крейта, когда он подключен по USB, но настроен на работу по Ethernet. При такой ситуации в зависимости от версии программы крейт может быть виден как пустой без каких-либо явных указаний, либо вообще не быть видимым в **LtrServer**. В то время как при работе через **ltrd** и **LTR Manager** он всегда виден в режиме “Только настройка” и предоставляется возможность изменить настройки крейта.

5. **ltrd** вместе с **LTR Manager** предоставляют более подробную статистку и информацию по подключенным крейтам, нежели **LtrServer**.

6. **ltrd** не поддерживает часть редко используемых возможностей **LtrServer**, подробнее см. в документе ltr_cross_sdk.pdf.

3 Работа с крейтом по интерфейсу Ethernet

Крейты LTR-EU и LTR-CEU предоставляют возможность подключения не только по интерфейсу USB, но и по интерфейсу Ethernet с использованием протокола TCP/IP. В отличие от USB работа по сети требует дополнительных настроек, чему посвящена данная глава.

3.1 Настройка крейта на работу по Ethernet

Одновременно крейт LTR позволяет работать с модулями только по одному интерфейсу. Соответственно, крейт всегда настроен на работу либо по USB, либо по Ethernet. При этом изменение настроек крейта LTR-EU всегда выполняется по интерфейсу USB (крейт LTR-CEU позволяет изменить настройки и по Ethernet).

Если крейт настроен на работу по Ethernet, то он все равно работает и как USB-устройство, однако по USB доступны только функции по настройке интерфейса и обновлению прошивки, но невозможна работа с модулями. В таком случае данный крейт будет виден в **LTR Manager** без модулей с режимом работы “Только настройка” в параметрах крейта.

Если при этом связь с крейтом одновременно установлена и по Ethernet, то крейт будет отображаться два раза — одна запись соответствует рабочему подключению по Ethernet, а другая — подключению по USB для настройки.

Настройки крейта можно изменить в программе **LTR Manager**. Для этого достаточно нажать правой кнопкой мыши по крейту и выбрать пункт меню "Настройки крейта" (пункт доступен только для записей, соответствующих крейтам LTR-EU, подключенным по USB, либо крейтам LTR-CEU). Появится окно, в котором будут отображены текущие настройки крейта, которые можно изменить на требуемые.

При описанной выше настройке интерфейса Ethernet должны быть указаны следующие параметры, которые зависят от сети, в которую подключен крейт:

- IP-адрес крейта. 4 числа от 0 до 255 (например, 192.168.1.2). Выбор адреса описан в разделе 3.2.
- Маска подсети. Определяет, какая часть IP-адреса отвечает за сеть, а какая за адрес внутри сети. Для локальных сетей чаще всего используется маска 255.255.255.0, что означает, что первые 3 цифры отвечают за адрес сети, а последняя — за адрес устройства внутри сети.
- IP-адрес шлюза (gateway). Если крейт подключен в локальную сеть (крейт и ПК в одной сети), то не используется. Вопрос работы с крейтом через Интернет — отдельный вопрос, требующий рассмотрения большого количества случаев, и в данном документе не рассматривается.

Важно!: Новые настройки вступают в силу только после перезагрузки крейта. Крейты LTR-CEU поддерживают программную команду перезагрузки (что может сразу и выполнить программа **LTR Manager**). Для крейтов LTR-EU необходимо выполнить перезагрузку вручную снятием питания крейта или кнопкой сброса.

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.

3.2 Выбор IP-адреса для крейта

При назначении IP-адреса нужно учитывать следующие требования к нему:

- Он должен быть уникальный внутри сети (т.е. не должно быть другого устройства с таким же адресом, в частности адрес крейта и ПК должны быть разные).
- Он должен принадлежать той же сети, что и адрес ПК, на котором запущена служба **ltrd** (Т.е. числа в начале, соответствующие адресу сети, должны совпадать).

При подключении в уже готовую сеть надо учитывать, что крейт не поддерживает протоколы для автоматического получения адреса (DHCP, link-local). Соответственно, если в Вашей сети адреса назначаются автоматически, то необходимо, чтобы администратор сети выделил некоторый диапазон адресов для ручного (статического) назначения и назначить крейту адрес из данного диапазона.

При подключении к ПК напрямую необходимо настроить как адрес крейта, так и адрес ПК (или конкретного интерфейса ПК), к которому подключен крейт. Как упоминалось выше, адрес крейта и ПК должны быть разные, но внутри одной сети. Например, если адрес крейта 192.168.1.2, то адрес ПК может быть 192.168.1.1 (маска и у ПК, и у крейта — 255.255.255.0).

3.3 Установление соединения с крейтом

В отличие от крейтов, подключенных по USB, **ltrd** не устанавливает связь с крейтами, подключенными по Ethernet, без ручной настройки. Ниже описаны общие шаги, которые нужно выполнить в программе **LTR Manager** для подключения крейта. Более подробно Вы может прочитать в руководстве ltr_cross_sdk.pdf.

Для установления связи необходимо:

- Добавить IP-адрес крейта в список адресов в программе **LTR Manager**.
- Двойным нажатием на добавленной записи (или через контекстное меню по нажатию на записи правой кнопкой мыши) запустить процедуру установления соединения с крейтом по сети (статус записи должен измениться на "Подключение...").
- При завершении подключения статус должен измениться на "Подключен", а крейт появиться в списке подключенных крейтов.
- После этого с крейтом можно работать также, как с крейтом, подключенным по интерфейсу USB.

Примечание: Если у добавленной записи с IP-адресом крейта установить флажок "Авто", то при запуске **ltrd** будет выполнена попытка автоматически установить соединения с крейтом по данному адресу.

4 Разработка пользовательского программного обеспечения

4.1 Разработка ПО для ПК

Для разработки пользовательского ПО верхнего уровнял "Л Кард" предоставляет набор библиотек, написанных на языке *C* (.dll для `Windows` или .so для `Linux` или `FreeBSD`). Для каждого модуля предоставляется своя библиотека `ltrXXXapi` (где XXX — номер модуля) со своим набором функций. Для управления работой **ltrd**, а также для управления работой крейтов, предоставляется специальная библиотека `ltrapi`.

Для каждой библиотеки предоставляется свое руководство программиста (`ltrXXXapi.pdf`), которое можно скачать с сайта в разделе "Документация" на странице каждого модуля.

Помимо использования в программах на *C*, "Л Кард" предоставляет также "обертки" для возможности работы с модулями с использованием других языков и сред. На данный момент поддерживаются *Delphi*, *C#* и *LabView*.

Примеры программирования пользователь может скачать со странички "ПО для разработчика", а также часть примеров идет вместе с исходными кодами ltr_cross_sdk (`https://bitbucket.org/lcard/ltr_cross_sdk/downloads/ltr_cross_sdk_src.zip`).

4.2 Использование библиотек при написании программ на языках C/C++

Для работы с модулями LTR из своей программы необходимо подключить к проекту динамическую библиотеку `ltrXXXapi` (где XXX - номер модуля) для каждого модуля, а также общую библиотеку `ltrapi`, если используются ее функции.

При написании программы на на языке *C/C++* в среде **Microsoft Visual Studio** можно использовать пакет **NuGet**, как описано в разделе 4.3, который выполняет все действия по подключению библиотек автоматически.

При использовании другой среды (или без среды) для подключения библиотеки нужно выполнить следующие действия:

- Включить заголовочный файл `ltr/include/ltrXXXapi.h` для каждого используемого модуля, в котором определены все доступные константы, типы и функции для работы с данным модулем.
- Для ОС `Windows`:
  - Убедиться, что все нужные файлы `ltrXXXapi.dll` и `ltrapi.dll` установлены либо в директорию с программой, либо в директорию из переменной окружения `PATH` (установщик устанавливает в `%WINDIR%/system32`)
  - Подключить к проекту файлы `ltrXXXapi.lib` или `libltrXXXapi.a` (а также `ltrapi.lib` или `libltrapi.a` при использовании функций из `ltrapi`) для нужного компилятора:
    - *Microsoft Visual C++* 32-битный компилятор — из `ltr/lib/msvc`
    - *Microsoft Visual C++* 64-битный компилятор — из `ltr/lib/msvc64`
    - *Borland C++/Borland C++ Builder/Embarcadero C++ Builder* 32-битный компилятор — из `ltr/lib/borland`
    - *Embarcadero C++ Builder* 64-битный компилятор — из `ltr/lib/borland64`
    - *MinGW* 32-битный компилятор — из `ltr/lib/mingw`
    - *MinGW* 64-битный компилятор — из `ltr/lib/mingw64`
- Для ОС `Linux`
  - Убедиться, что файлы `libltrXXXapi.so`, `libltrXXXapi.so.*` (где * - версия), `libltrapi.so`, `libltrapi.so.*` установлены либо в одну из системных директорий для библиотек, либо в директорию, заданную через переменную окружения `LD_LIBRARY_PATH` или другим способом (собранные пакеты устанавливают библиотеки в стандартную директорию `/usr/lib`)
  - Подключить библиотеку к проекту. При вызове GCC из командной строки это можно сделать с помощью ключа -lltrXXXapi.

4.3 Использование пакета NuGet при написании программ на языках C/C++ в Microsoft Visual Studio

NuGet (`https://www.nuget.org`) — это менеджер пакетов для среды **Microsoft Visual Studio**. Начиная с **Visual Studio 2012** он по умолчанию уже идет предустановленным вместе со средой. Для использования библиотек **ltrapi** в проекте на языке

C/C++ в репозитории **NuGet** доступен пакет **lcard.ltr.ltrapi** (https://www.nuget.org/packages/lcard.ltr.ltrapi).

При использовании пакета **NuGet** установка **ltrdll** через установщик не требуется, так как все необходимые файлы включены в пакет.

При включении этого пакета в проект будут автоматически загружены все необходимые файлы (.h, .lib, .dll), и проект будет настроен так, что будут включены нужные пути до заголовочных файлов, подключены нужные .lib файлы (32-битные или 64-битные в зависимости от используемой конфигурации) и все .dll файлы нужной разрядности будут скопированы в выходную директорию проекта.

В коде проекта достаточно только включить нужные заголовочные файлы (`ltr/include/ltrXXXapi.h`), где объявлены все определения, типы и функции.

Также **NuGet** позволяет отслеживать появление новых версий пакетов и выполнять их обновление.

В последних версиях **Visual Studio** все эти функции встроены в среду и выполняются из нее. Например, для подключения пакета в **Visual Studio 2015** достаточно нажать правой кнопкой на проекте в "Обозревателе решений" и выбрать в меню "Управление пакетами NuGet...". В открывшемся окне выбрать обзор и в строке поиска ввести ltrapi, выбрать пакет **lcard.ltr.ltrapi** и нажать "Установить".

4.4 Использование библиотек при написании программ на Delphi

Для написания программ на *Delphi* "Л Кард" предоставляет набор .pas файлов, в которых объявлены те же структуры и функции, что используются в библиотеках на *C*. В программе нужно включить через uses нужные модули: ltrapi, ltrapitypes, ltrapidefine и ltrXXXapi для каждого используемого модуля. При этом программа будет использовать те же .dll файлы, что и программа на *C*. При этом есть две версии .pas файлов:

- В `ltr/include/pascal` устанавливается старая версия файлов, которые оставлены только для совместимости со старыми проектами. Эти файлы требуют ручной настройки выравнивания структур в среде, их нельзя использовать в современных версиях *Delphi* и могут содержать некоторые неточности.

- В `ltr/include/pascal2` устанавливается новая версия файлов. Эти файлы можно использовать как в *Delphi 7*, так и в современных версиях *Embarcadero Delphi XE*, как для 32-битных, так и 64-битных приложениях. Эти файлы имеют небольшие несовместимости со старой версией (например для символов используется тип AnsiChar, а не Char, т.к. в современных версиях Char — юникодный символ, занимающий два байта, а не один). Эта версия требует лишь небольших изменений и рекомендуются для использования.

Следует иметь ввиду, что также ведется изменение описания функций для старых модулей (с сохранением старых вариантов). Для большего соответствия языку выполняются следующие изменения:

- Отказ от указателей (Pointer) в пользу типизированных переменных var или out.

- Использование типа string (стандартный тип для строк в *Delphi*) для передачи строк для избавления пользователя от ручных преобразований

- Использование для передачи массивов в качестве параметра открытого массива (open array parameter), что означает, что в эти функции можно передать как статический массив, так и динамический (установив предварительно его длину с помощью `SetLength()`)

Новые модули содержат описания с учетом указанных выше изменений, для старых новый вариант функций будет добавляться по мере возможности и обновляться примеры.

4.5 Использование библиотек при написании программ на языке C#

Для написания программ на *C#* предоставляется специальная библиотека-обертка `ltrModulesNet.dll` (устанавливается в `ltr/bin/ltrModulesNet`), которую надо подключить к создаваемому проекту. Она использует уже функции из .dll на языке *C*. Т.е. для работы программы нужна как `ltrModulesNet.dll`, так и остальные используемые библиотеки `ltrXXXapi.dll`.

Все классы в данной библиотеке объявлены внутри пространства ltrModulesNet (можно включить через using ltrModulesNet).

Все функции, структуры и константы объединены внутри соответствующих классов:

- Определения модулей для нового варианта api содержатся в классах ltrXXXapi (без подчеркивания). Эти классы предназначены как для использования в программах на *C#*, так и в программах на *LabView*. При этом в них доступ к полям описателя сделан через свойства класса. Для некоторых функций сделан вариант с более удобными типами параметров. В настоящее время данные классы доступны не для всех модулей.

- Выделен класс ltrcrate для доступа к функциям ltrapi, относящимся к одному крейту.

- Выделен класс ltrsrvcon для доступа к функциям ltrapi, относящимся к управляющему соединению с **ltrd**

- Все определения из ltrapi находятся в классе _LTRNative

- Для старого варианта api используются два класса, класс _ltrXXXapi и отдельный класс ltrXXXapiLabView, предназначенный для использования в *LabView*, отличающийся только тем, что все поля описателя модуля доступны через члены самого класса. В этих некоторые параметры описаны так, что требуют ручного преобразования. При наличии рекомендуется использовать новые классы ltrXXXapi, если они существуют для данного модуля.

Существует два варианта работы с функциями:

- Через статические методы класса. Они в точности повторяют прототип и название функций на *C* и также принимают первым параметром структуру, соответствующую описателю модуля.

- Через методы самого класса. В этом случае создается объект данного класса, который содержит внутри себя структуру описателя модуля. Соответственно, в

методы класса эта структура не передается (т.к. в методе доступны все поля экземпляра класса). Названия методов аналогичны функциям на *C*, но без префикса LTRXXX_ (так как к какому модулю относится функция, понятно по классу, к которому относится этот модуль). Поля описателя модуля должны меняться либо в структуре внутри класса (для классов с подчеркиванием), либо через свойства (для новых классов без подчеркивания). Для новых классов этот вариант более предпочтителен, так как позволяет сделать дополнительно нужные преобразования в стандартные типы *C#* (например для строк или массивов структур).

Новый вариант классов с примерами их использования есть только для ограниченного набора модулей. По возможности список таких модулей будет расширяться.

4.6 Использование библиотек при написании программ на LabView

Так как *LabView* поддерживает работу с управляемыми библиотеками NetFramework, то для создания программ на *LabView* можно использовать библиотеку `ltrModulesNet.dll`.

Описания классов в `ltrModulesNet.dll`, которые можно использовать в *LabView*, приведено в разделе 4.5.

Важно!: Поддерживаются только версии *LabView* начиная с 8.0 (и выше), так как в предыдущих версиях, хотя и была поддержка библиотек NetFramework, но в ней были существенные проблемы с производительностью.

Для работы с классами .Net в *LabView* есть специальная панель **Connectivity ->** **.Net**.

При этом необходимо использовать следующие блоки:

- **Constructor Node** — создает объект. Должен быть создан для каждого модуля LTR, с которым будет идти работа, а также для каждого управляющего соединения с крейтом или **ltrd**. При создании *LabView* предложит выбрать библиотеку и класс (нужно выбрать `ltrModulesNet.dll` и нужный класс: ltrXXXapi для данного модуля, если есть, или ltrXXXapiLabView, если нету). Одним из выходов этого блока является ссылка на объект, которая используются как вход для остальных блоков для работы с модулем. Также с помощью конструктора создается объекты структур, необходимых при работе с модулем.

- **Close Reference** — закрывает и удаляет объект. Должен вызываться для каждого созданного объекта по завершению работы.

- **Invoke Node** — вызов функции (метода класса). При работе с объектом на вход подается ссылка и входные параметры, а на выходе — выходные параметры и обновленная ссылка (которая должна использоваться для блоков, которые будут вызваны после текущего). Входная ссылка и определяет, методы какого объекта используются (после заведения ссылки на вход имя класса появится в верхней строке, а при нажатии на вторую будет предложен на выбор его метод). Для функций, которые не работают с конкретным объектом (эти функции статические и при выборе помечены [S] в начале) ссылку на вход подавать не обязательно. Однако они все равно принадлежат классу, который надо выбрать нажав правой кнопкой на блок и далее **Select Class/.Net**.

- **Property Node** — используется для установки или получения свойств объекта. Через свойства устанавливаются или считываются поля структур (например, настройки модуля или информация о модуле). Можно устанавливать несколько свойств одним блоком, расширив его вниз. Также с помощью свойств можно задавать константы из перечислений (что может быть более понятно, чем просто подавать на вход числа) — в этом случае надо выбрать класс перечисление, а каждому значению соответствует свое свойство.

Следует отметить особенность передачи массивов в качестве выходных параметров. Сами функции библиотеки для эффективности не выделяют массивов данных внутри себя, а используют переданные массивы для сохранения результатов. Поэтому такие параметры в *LabView* являются входными и выходными одновременно. На вход необходимо подать массив размера, достаточного для сохранения результатов (при этом сами данные не имеют значения), а в качестве выходного параметра возвращается тот же массив, но уже заполненный результатами выполнения функции.

4.7 Создание 64-битных программ под ОС Windows

На 64-биной версии ОС `Windows` могут выполняться программы, как собранные 32-битным, так и 64-битным компилятором (последние называют нативными 64-битными приложениями), поэтому многие программы для `Windows` существует только в 32-битном варианте. 64-битный компилятор используют как правило для программ, работающих с большим количеством данных, так как процесс 64-битной программы имеет виртуальное пространство больше 4 Гбайт.

Так как эта возможность может быть очень нужна для систем сбора данных при большом количестве обрабатываемой информации, то начиная с версии 1.28.0 установщик **ltrdll.exe** устанавливает дополнительно 64-битные версии библиотек, т.е. предоставляет возможность создания нативных 64-битных приложений. При этом изменено расположение файлов по сравнению с версиями 1.27.x и ниже (подробнее см. файл `readme.txt` среди устанавливаемых файлов).

Важно!: Для установки 64-битных библиотек была изменена программа для создания установщика. Поэтому, если до установки была установлена версия 1.27.x или ниже, то необходимо сперва ее удалить, а потом уже поставить новую.

При установке на 64-битной версии `Windows` установщик **ltrdll** в соответствующие системные директории ставит и 32-битную версию библиотек, так и 64-битную. При этом директория `Windows/system32` указывает на одну из этих директорий в зависимости от разрядности самого приложения, которое обращается по указанному пути. Для 32-битного приложения в `Windows/system32` находятся 32-битные библиотеки, а 64-битные в `Windows/Sysnative`, а для 64-битного в `Windows/system32` находятся 64-битные, а `Windows/Sysnative` не существует. При этом в `Windows/SysWOW64` всегда лежат 32-битные библиотеки и она всегда существует независимо от разрядности приложения.

При загрузке приложения, если используются системные библиотеки, то они ищутся по путям из переменной окружения `PATH`, среди которых есть `Windows/system32`. Так как эта директория ссылается на разные места в зависимости от разрядности запускаемого приложения, то выбор библиотеки нужной разрядности из `Windows/system32` происходит автоматически. Если библиотеки распространяются вместе с программой, то нужно следить, чтобы разрядность собранного приложения и библиотек в одной директории совпадала.

Единственным отличием при написании программ на *C/C++* является необходимость подключить lib-файл (или .a) в соответствии с разрядностью используемого компилятора.

Для программ на *Delphi* необходимо только указать, для какой платформы будет собираться проект (win32 или win64) и собранная программа будет использовать библиотеку той разрядности, для которой программа была скомпилирована.

Программы на языке *C#* (или на любом другом, использующем NetFramework) компилируются в машинный код при выполнении. При этом один раз созданная программа может выполняться как на 32-битной версии, так и на 64-битной версии виртуальной машины NetFramework (в проекте при необходимости можно указать явно для какой разрядности NetFramework предназначена программа). Таким образом, если разрядность не указана явно, то одна и та же программа в 32-битной версии `Windows` будет выполняться с использованием 32-битной версии библиотек, а в 64-битной — с использованием 64-битной. Для библиотеки `ltrModulesNet.dll` разрядность определяется разрядностью использующего его приложения.

Соответственно, в проекте на *LabView*, который использует .Net библиотеку `ltrModulesNet.dll`, разрядность используемой библиотеки определяется разрядностью используемой среды *LabView*.

4.8 Где взять исходные коды ПО для ПК

- Архив с исходными кодами библиотек **ltrapi**, службы (демона) **ltrd**, программы **LTR Manager**, а также всех остальных утилит, входящих в состав **ltr_cross_sdk**, можно скачать по ссылке https://lcard.ru/download/ltr_cross_sdk_src.zip. Также можно воспользоваться открытым **git** репозиторием https://gitlab.com/l-card/acq/devices/ltr/shared/sdk.

4.9 Создание пользовательской версии прошивки крейтов LTR-EU

Для некоторых задач пользователя может не устраивать штатная прошивка крейта. В качестве примера, можно привести следующие причины:

- Необходимость иметь автономное устройство, запускающее сбор и выполняющее обработку без управления с ПК
- Необходимость обеспечить минимальные и определенные задержки от изменения сигналов на входах системы до реакции на ее выходах, для чего необходимо выполнять принятие решений внутри крейта
- Необходимость реализации своих протоколов для обмена с ПК
- Необходимость управления другими устройствами по сети или RS-485 непосредственно из крейта
- и др.

Для подобных случаев пользователю предоставляется возможность изменения прошивки сигнального процессора Blackfin, установленного в крейтах LTR-EU.

Также пользователь может заказать в “Л Кард” разработку подобной прошивки, обратившись в техническую поддержку.

Важно!: Следует учитывать, что разработка прошивки для встраиваемых систем сильно отличается от разработки ПО для ПК и требует соответствующих навыков.

Важно!: На данный момент “Л Кард” не предоставляет руководства по модификации прошивки процессора крейта, поэтому необходимую информацию программист должен быть готов получить, анализируя предоставляемые исходные коды прошивок. При этом наши техподдержка может ответить на общие вопросы.

Существует две версии прошивки крейтов LTR-EU, для которых “Л Кард” предоставляет пользователю исходные коды для дальнейшей модификации:

1. Прошивка на основе ОСРВ FreeRTOS с использованием стека TCP/IP lwip. Данная прошивка собирается с помощью среды VisualDSP от Analog Devices. Исходные коды данной прошивки можно скачать с сайта “Л Кард” по адресу: `http://www.lcard.ru/download/ltr_source_1_0_0_1.zip`. Это законченная прошивка, но она не будет развиваться в дальнейшем.

2. В настоящее время разрабатывается прошивка на основе ОСРВ RTEMS с использованием стека TCP/IP из данной ОС, перенесенного из `FreeBSD`. Данная версия призвана решить некоторые проблемы старой версии при работе по TCP/IP. Для получения исходных кодов прошивки можно обратиться в службу технической поддержки (см. раздел “Контакты” сайта: `http://www.lcard.ru/contakt`).

5 При возникновении проблем

При возникновении проблем при работе с крейтами и модулями LTR, необходимо сделать следующее:

1. Убедиться, что были прочитаны и осмысленны: данный документ, необходимые разделы руководства пользователя, а также документация на используемое ПО (при наличии) или соответствующие руководства программиста при написании собственных программ.

2. Проверить, последние ли версии ПО установлены. Последние версии Вы можете скачать с сайта в разделе “Программное обеспечение” страницы сайта “Л Кард”, посвященной интересующему Вас модулю.

3. Если Вы используете свое ПО или ПО, разработанное не “Л Кард”, а другими фирмами, то необходимо проверить работу модуля на предоставляемом “Л Кард” ПО (см. раздел 2.3).

4. Посмотреть на форуме технической поддержки (`http://www.lcard.ru/forums/viewforum.php?id=1`), не обсуждалась ли уже Ваша проблема.

5. Если проблема остается, то необходимо **максимально(!)** подробно описать ее, создав тему на форме технической поддержки, либо написав на почту технической поддержки (см. раздел “Контакты” сайта: `http://www.lcard.ru/contakt`).