Современные устройства сбора данных

LTA

Руководство программиста

Ревизия 1.0.0

Июль 2026

Автор руководства:
Мирзиянов Руслан

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

LTA API © Copyrigh 2026, ООО “Л Кард”. Все права защищены.

Таблица 1: Ревизии текущего документа

Ревизия	**Дата**	**Описание**
1.0.0	07.2026	Первая ревизия данного документа

Оглавление

1 О чём этот документ **5**

2 Установка и подключение библиотеки к проекту **6**

3 Общий подход к работе с библиотекой **7**
3.1 Общие и специализированные функции для работы с модулем 7
3.2 Общий алгоритм работы с модулем 7

4 Константы, типы данных и функции библиотеки **8**
4.1 Константы и перечисления 8
4.1.1 Константы и макроопределения 8
4.1.2 Номер посадочного места модуля 9
4.1.3 Коды ошибок библиотеки 9
4.1.4 Коды ошибок самодиагностики модуля 11
4.1.5 Флаги, описывающие особенности устройства 12
4.1.6 Команды для обращения к энергонезависимой памяти модуля . . 12
4.1.7 Флаги сетевых настроек 12
4.1.8 Входные флаги управления логикой функций API модуля 13
4.1.9 Выходные флаги функций API модуля 13
4.1.10 Язык вывода строки с описанием ошибки 13
4.1.11 Номер версии библиотеки 13
4.2 Типы данных 14
4.2.1 Структуры 14
4.2.1.1 Запись о устройстве 14
4.2.1.2 Фильтры для поиска подключенных устройств 15
4.2.1.3 Текущие сетевые настройки модуля 15
4.2.1.4 Информация о модуле 15
4.2.1.5 Запись о сетевых настройках модуля 16
4.2.1.6 Калибровочные коэффициенты диапазона. 16
4.2.1.7 Описатель соединения с модулем 17
4.3 Функции библиотеки lta_api 17
4.3.1 Выполнение процедуры поиска модулей в локальной сети 17
4.3.2 Заполнение записи типа lta_dev_rec_t для её последующего использования в LTA_Open. 18
4.3.3 Открытие соединения с модулем 18
4.3.4 Закрытие соединения с модулем 19
4.3.5 Проверка, открыто ли соединение с модулем 19
4.3.6 Проверка, запущен ли синхронный ввод/вывод 19
4.3.7 Проверка соединения с модулем 20
4.3.8 Сброс модуля 20

4.3.9 Получение информации о модуле 20
4.3.10 Чтение пользовательской сетевой конфигурации интерфейса . . . 21
4.3.11 Установка пользовательской сетевой конфигурации интерфейса . 21
4.3.12 Чтение блока данных из пользовательской энергонезависимой памяти модуля 22
4.3.13 Запись блока данных в пользовательскую энергонезависимую память модуля 22
4.3.14 Стирание блока данных пользовательской энергонезависимой памяти модуля 23
4.3.15 Получить версию библиотеки 23
4.3.16 Получение строки об ошибке 23
4.3.17 Получение строки об ошибке самодиагностики 24
4.3.18 Моргание светодиодом 24

Глава 1

О чём этот документ

Данный документ предназначен в первую очередь для программистов, которые собираются писать свои программы для работы с модулем lta11 с использованием предоставляемой фирмой “Л Кард” библиотеки **lta_api**.

В данном документе рассматривается вопрос подключения библиотеки к проекту пользователя, дается подробное описание интерфейсных функций, предоставляемых библиотекой, и используемых типов, а также дается описание основных подходов к использованию этих функций.

Сама библиотека написана на языке C и все объявления функций и типов приводятся на языке C. Однако все привязки к другим языкам программирования являются лишь обертками над библиотекой C и все функции, типы и параметры сохраняют свои значения и для других языков программирования. Поэтому этот документ полезен и пользователям, пишущим на других языках программирования.

Глава 2

Установка и подключение библиотеки к проекту

Для начала работы с устройством достаточно установить динамическую библиотеку (**lta_api.dll** для Windows или **lta_api.so** для Linux) в директорию, присутствующую в соответствующей переменной окружения, либо в директорию с проектом. В данной динамической библиотеке содержатся все функции для работы с модулями LTA. Для работы с модулем lta11 в вашем проекте на языках C/C++ необходимо добавить заголовочный файл **lta_api.h**

Работа с модулем lta11 не требует установки дополнительных драйверов.

Глава 3

Общий подход к работе с библиотекой

3.1 Общие и специализированные функции для работы с модулем

В связи с тем, что большая часть функций используемая для соединения с модулями LTA совпадают, то большинство функций используемых для обнаружения и подключения реализовано общими для всех модулей.

Все функции реализованы в библиотеке **lta_api**, и могут быть добавлены в проект с помощью соответствующего заголовочного файла

- **lta_api.h** – содержит прототипы функций, для общих для всех устройств LTA.
- **lta11_api.h** – содержит прототипы функций, специфичных для модуля lta11.

3.2 Общий алгоритм работы с модулем

Данный раздел описывает типичную последовательность действий при работе с модулем lta11. Более подробно каждый шаг будет описан в последующих разделах. Типичная последовательность действий имеет следующий вид:

1. С помощью функции `LTA_Search()` найти все подключенные по Ethernet модули.
2. Установить соединение с интересующим модулем и создать экземпляр структуры `lta_hnd_t`, представляющей собой описатель модуля, с помощью функции `lta11_Open()`. Описатель модуля содержит всю информацию о модулях LTA и используется при вызове всех остальных функций.
3. Установка параметров модуля с помощью набора функций для изменения настроек модуля (названия функций начинаются с `lta11_Set`)
4. Передача установленных параметров в модуль с помощью `lta11_Configure()`
5. Работа с модулем в синхронном и/или асинхронном режиме (описана в последующих подразделах).
6. Закрыть соединение с модулем, вызвав функцию `lta11_Close()`.

Глава 4

Константы, типы данных и функции библиотеки

4.1 Константы и перечисления

4.1.1 Константы и макроопределения

Константа	Значение	Описание
LTA_DEVNAME_SIZE	(32U)	Максимальное количество символов в строке с названием устройства
LTA_SERIAL_SIZE	(32U)	Максимальное количество символов в строке с серийным номером
LTA_REVISION_SIZE	(16U)	Максимальное количество символов в строке с названием ревизии платы
LTA_SLOT_NUM_SIZE	(4U)	Максимальное количество символов в строке с названием номера слота
LTA_MCU_VER_SIZE	(32U)	Максимальное количество символов в строке с версией прошивки
LTA_PASSWORD_SIZE	(32U)	Максимальный размер строки с паролем на настройки
LTA_IPV4_ADDR_SIZE	(4U)	Размер поля IP-адреса
LTA_MAC_ADDR_SIZE	(6U)	Размер поля MAC-адреса Ethernet интерфейса
LTA_INSTANCE_NAME_SIZE	(64U)	Размер строки с описанием экземпляра устройства
LTA_USER_DATA_SIZE	(256U)	Размер поля резервной области памяти в байтах
LTA_CMD_PORT_DEFAULT	(11114)	TCP-порт управляющего соединения, использующийся по умолчанию
LTA_DATA_PORT_DEFAULT	(11115)	TCP-порт соединения передачи данных, использующийся по умолчанию
LTA_USER_MEM_SIZE	(0x4000)	Размер области пользовательской энергонезависимой памяти
LTA_LIB_VER	(0x01000000)	Актуальная, последняя версия библиотеки

4.1.2 Номер посадочного места модуля

Тип: `lta_dev_pos_t`		
Описание: Номер посадочного места модуля		
Константа	**Значение**	**Описание**
`LTA_DEV_POS_NUM_CRATE`	`0`	Крейт-контроллер в мультислотовом крейте
`LTA_DEV_POS_NUM_1`	`1`	Модуль установлен в 1-й слот крейта
`LTA_DEV_POS_NUM_2`	`2`	Модуль установлен во 2-й слот крейта
`LTA_DEV_POS_NUM_3`	`3`	Модуль установлен в 3-й слот крейта
`LTA_DEV_POS_NUM_4`	`4`	Модуль установлен в 4-й слот крейта
`LTA_DEV_POS_NUM_5`	`5`	Модуль установлен в 5-й слот крейта
`LTA_DEV_POS_NUM_6`	`6`	Модуль установлен в 6-й слот крейта
`LTA_DEV_POS_NUM_7`	`7`	Модуль установлен в 7-й слот крейта
`LTA_DEV_POS_NUM_8`	`8`	Модуль установлен в 8-й слот крейта
`LTA_DEV_POS_NUM_9`	`9`	Модуль установлен в 9-й слот крейта
`LTA_DEV_POS_NUM_10`	`10`	Модуль установлен в 10-й слот крейта
`LTA_DEV_POS_NUM_11`	`11`	Модуль установлен в 11-й слот крейта
`LTA_DEV_POS_NUM_12`	`12`	Модуль установлен в 12-й слот крейта
`LTA_DEV_POS_NUM_13`	`13`	Модуль установлен в 13-й слот крейта
`LTA_DEV_POS_NUM_14`	`14`	Модуль установлен в 14-й слот крейта
`LTA_DEV_POS_NUM_15`	`15`	Модуль установлен в 15-й слот крейта
`LTA_DEV_POS_NUM_16`	`16`	Модуль установлен в 16-й слот крейта
`LTA_DEV_POS_NUM_UNKNOWN`	`200`	Посадочное место неизвестно (некорректно)
`LTA_DEV_POS_NUM_SINGLE`	`255`	Модуль установлен в одиночный крейт

4.1.3 Коды ошибок библиотеки

Тип: `lta_err_t`		
Описание: Коды ошибок библиотеки		
Константа	**Значение**	**Описание**
`LTA_ERR_OK`	`0`	Функция выполнена без ошибок
`LTA_ERR_INVALID_HANDLE`	`-1`	В функцию передан недействительный описатель модуля
`LTA_ERR_MEMORY_ALLOC`	`-2`	Ошибка выделения памяти
`LTA_ERR_INVALID_ARG`	`-3`	В функцию передан недействительный аргумент
`LTA_ERR_DEV_NOT_FOUND`	`-4`	Указанное устройство не обнаружено
`LTA_ERR_MODULE_IN_USE`	`-5`	Предупреждение: уже создано активное соединение с данным модулем
`LTA_ERR_NOT_IMPLEMENTED`	`-6`	Данная возможность не реализована

LTA_ERR_NOT_CONFIGURED	-7	Не выполнена конфигурация модуля
LTA_ERR_DEVICE_NOT_OPENED	-8	Связь с устройством закрыта или не была установлена
LTA_ERR_INVALID_DEVICE	-9	Неверный тип устройства
LTA_ERR_MUTEX_CREATE	-10	Ошибка создания системного мьютекса
LTA_ERR_MUTEX_INVALID_HANDLE	-11	Неверный описатель мьютекса
LTA_ERR_MUTEX_LOCK_TOUT	-12	Истекло время ожидания освобождения мьютекса
LTA_ERR_MUTEX_RELEASE	-13	Ошибка освобождения мьютекса
LTA_ERR_SOCKET_RECV	-14	Ошибка чтения данных из сокета
LTA_ERR_SOCKET_SEND	-15	Ошибка передачи данных в сокет
LTA_ERR_SOCKET_OPEN	-16	Не удалось открыть сокет для соединения
LTA_ERR_CONNECTION_TOUT	-17	Превышено время подключения
LTA_ERR_CONNECTION_CLOSED_BY_DEV	-18	Соединение закрыто другой устройством
LTA_ERR_SOCKET_SET_BUF_SIZE	-19	Не удалось установить заданный размер буфера сокета
LTA_ERR_CONNECTION_RESET	-20	Соединение было сброшено другой стороной
LTA_ERR_HOST_UNREACHABLE	-21	Не удалось найти хост с указанным адресом
LTA_ERR_TCP_CONNECTION_ERROR	-22	Ошибка установления TCP-соединения
LTA_ERR_DNSSD_NOT_RUNNING	-23	Не запущена программа обнаружения устройств в локальной сети
LTA_ERR_DNSSD_COMMUNICATION	-24	Ошибка при обращении к программе обнаружения устройств в локальной сети
LTA_ERR_DNSSD_INVALID_SVC_REC_HND	-25	Неверный описатель записи о сервисе
LTA_ERR_SELF_DIAG	-26	Предупреждение: ошибка самодиагностики модуля
LTA_ERR_UNKNOWN_DEV	-27	Предупреждение: подключены к неизвестному модулю
LTA_ERR_CMD_FAILED	-50	Ошибка выполнения управляющей команды
LTA_ERR_CMD_RECV_INSUF_WORDS	-51	Принято недостаточно слов от модуля
LTA_ERR_CMD_INVALID_RESP_SIZE	-52	Неверный размер ответа на управляющий запрос

LTA_ERR_CMD_SEND_INSUF_WORDS	-53	Передано недостаточно слов в модуль
LTA_ERR_CMD_NO_RESPONSE	-54	Не пришло ответа на переданную команду
LTA_ERR_CMD_RECV_SIGNATURE	-55	Неверный признак начала ответа
LTA_ERR_NO_DATA_CONNECTION	-60	Соединение для передачи данных не установлено
LTA_ERR_STREAM_IS_RUNNING	-61	Функция не может быть выполнена при запущенном потоке сбора данных
LTA_ERR_STREAM_IS_NOT_RUNNING	-62	Сбор данных уже остановлен
LTA_ERR_STREAM_OVERFLOW	-63	Произошло переполнение внутреннего буфера для потока синхронного ввода
LTA_ERR_UNSUP_STREAM_MSG	-64	Неизвестное сообщение в потоке синхронного ввода
LTA_ERR_DATA_CON_IN_USE	-65	Соединение для передачи данных уже установлено

4.1.4 Коды ошибок самодиагностики модуля

Тип: lta_self_diag_err_t		
Описание: Коды ошибок самодиагностики модуля		
Константа	**Значение**	**Описание**
LTA_SD_ERR_OK	0	Самодиагностика выполнена без ошибок
LTA_SD_ERR_SDRAM_TST	1	Тест SDRAM не пройден
LTA_SD_ERR_ETH_TST	2	Тест Ethernet не пройден
LTA_SD_ERR_PWR_TST	3	Тест питания не пройден
LTA_SD_ERR_FPGA_NOT_LOAD	4	Ошибка загрузки ПЛИС
LTA_SD_ERR_FPGA_CLK	5	Ошибка тактирования ПЛИС
LTA_SD_ERR_FPGA_CMD_INT	6	Ошибка интерфейса команд ПЛИС
LTA_SD_ERR_FPGA_DATA_INT	7	Ошибка интерфейса данных ПЛИС
LTA_SD_ERR_FPGA_MKO_ADDR	8	Ошибка адреса МКО
LTA_SD_ERR_FPGA_PRJ_NAME	9	Ошибка названия проекта ПЛИС

4.1.5 Флаги, описывающие особенности устройства

Тип: `lta_dev_flags_t`		
Описание: Флаги, описывающие особенности устройства		
Константа	**Значение**	**Описание**
`DEV_FLAGS_MCU_GD32F450`	`(uint32_t) 0x00000001`	Признак, что установлен микроконтроллер GD32F450
`DEV_FLAGS_FPGA_GW1N9`	`(uint32_t) 0x00000010`	Признак, что установлена ПЛИС GW1N-UV9EQ144
`DEV_FLAGS_FACTORY_INFO_IS_EMPTY`	`(uint32_t) 0x00000100`	Заводские настройки не записаны
`DEV_FLAGS_NET_INFO_IS_EMPTY`	`(uint32_t) 0x00000200`	Сетевые настройки не записаны
`DEV_FLAGS_ADDR_ESTABLISHED`	`(uint32_t) 0x00000400`	Адрес, выделенный DHCP-сервером, проверен
`DEV_FLAGS_FPGA_LOADED`	`(uint32_t) 0x00000800`	Признак, что присутствует прошивка ПЛИС и она успешно была загружена

4.1.6 Команды для обращения к энергонезависимой памяти модуля

Тип: `lta_fm_cmd_t`		
Описание: Команды для обращения к энергонезависимой памяти модуля		
Константа	**Значение**	**Описание**
`LTA_FM_CMD_WRITE`	`0`	Записать настройки
`LTA_FM_CMD_ERASE`		Стереть настройки

4.1.7 Флаги сетевых настроек

Тип: `lta_ec_flags_t`		
Описание: Флаги сетевых настроек		
Константа	**Значение**	**Описание**
`LTA_EC_FLAGS_AUTO_IP`	`0x0001`	Разрешение автоматического получения IP-адреса (через DHCP или link-local)
`LTA_EC_FLAGS_USER_MAC`	`0x0002`	Использовать пользовательский MAC-адрес

4.1.8 Входные флаги управления логикой функций API модуля

Тип: `lta_inflags_t`		
Описание: Входные флаги управления логикой функций API модуля		
Константа	**Значение**	**Описание**
`LTA_INFLAGS_DONT_INIT`	`0x00000001`	Не выполнять переинициализацию модуля (если она выполнена ранее)
`LTA_INFLAGS_DONT_STOP`	`0x00000002`	Не выполнять останов сбора (выдачи) данных
`LTA_INFLAGS_DONT_RESET`	`0x00000004`	Не выполнять сброс конфигурации модуля
`LTA_INFLAGS_DONT_CLOSE_CON`	`0x00000010`	Не выполнять закрытие других открытых соединений команд и данных

4.1.9 Выходные флаги функций API модуля

Тип: `lta_outflags_t`		
Описание: Выходные флаги функций API модуля		
Константа	**Значение**	**Описание**
`LTA_OUTFLAGS_REOPEN`	`0x00000001`	Модуль открыт повторно (переинициализация не выполялась)
`LTA_OUTFLAGS_IS_RUNNING`	`0x00000002`	Модуль находится в режиме сбора (выдачи) данных

4.1.10 Язык вывода строки с описанием ошибки

Тип: `lta_err_lang_t`		
Описание: Язык вывода строки с описанием ошибки		
Константа	**Значение**	**Описание**
`LTA_ERR_LANG_RU`	`0`	Вывод на русском языке
`LTA_ERR_LANG_EN`		Вывод на английском языке

4.1.11 Номер версии библиотеки

Тип: `lta_lib_ver_t`		
Описание: Номер версии библиотеки		
Константа	**Значение**	**Описание**
`LTA_LIB_VER_100`	`0x01000000`	Версия 1.0.0

4.2 Типы данных

4.2.1 Структуры

4.2.1.1 Запись о устройстве

Тип: `lta_dev_rec_t`		
Описание: Структура, описывающая устройство, по которой с ним можно установить соединение.		
Поле	**Тип**	**Описание поля**
`devname`	`char [LTA_DEVNAME_SIZE]`	Имя устройства (оканчивающаяся нулем ASCII-строка). Это обязательное поле.
`inst_name`	`char [LTA_INSTANCE_NAME_SIZE]`	Имя экземпляра устройства доступное для редактирования пользователю.
`serial_dev`	`char [LTA_SERIAL_SIZE]`	Серийный номер модуля (оканчивающаяся нулем ASCII-строка). Можно передать NULL, если значение неизвестно.
`serial_crate`	`char [LTA_SERIAL_SIZE]`	Серийный номер крейта (оканчивающаяся нулем ASCII-строка). Можно передать NULL, если значение неизвестно.
`slot`	`uint8_t`	Номер слота в крейте, куда установлен модуль. Может принимать значения из `lta_dev_pos_t`. Можно передать `LTA_DEV_POS_NUM_UNKNOWN`, если значение неизвестно, тогда номер слота не будет проверяться.
`ip_addr`	`uint32_t`	IPv4 адрес модуля в виде 32-битного слова. Для адреса “a.b.c.d” ip_addr = (a<<24)|(b<<16)|(c<<8)|d.
`cmd_port`	`uint16_t`	Номер порта для передачи команд. Должен соответствовать значению, записанному в сетевых настройках модуля. По умолчанию используется `LTA_CMD_PORT_DEFAULT`.

4.2.1.2 Фильтры для поиска подключенных устройств

Тип: `lta_dev_fltr_t`		
Описание: Структура содержит параметры по которым выполняется фильтрация модулей при поиске.		
Поле	**Тип**	**Описание поля**
`devname`	`char *`	Фильтр по наименованию устройства (оканчивающаяся нулем ASCII-строка). Можно передать NULL для отключения фильтра.
`serial`	`char *`	Фильтр по серийному номеру устройства (оканчивающаяся нулем ASCII-строка). Можно передать NULL для отключения фильтра.
`slot`	`uint8_t *`	Фильтр по номеру слота в крейте `lta_dev_pos_t`. Можно передать NULL для отключения фильтра.

4.2.1.3 Текущие сетевые настройки модуля

Тип: `lta_eth_info_t`		
Описание: Структура содержит значения текущих сетевых настроек.		
Поле	**Тип**	**Описание поля**
`flags`	`uint32_t`	Флаги настройки `lta_ec_flags_t`
`mac`	`uint8_t [LTA_MAC_ADDR_SIZE]`	MAC адрес
`addr`	`uint8_t [LTA_IPV4_ADDR_SIZE]`	IPv4 адрес
`mask`	`uint8_t [LTA_IPV4_ADDR_SIZE]`	Маска подсети
`gate`	`uint8_t [LTA_IPV4_ADDR_SIZE]`	Адрес шлюза
`cmd_port`	`uint16_t`	Порт передачи команд
`data_port`	`uint16_t`	Порт передачи данных

4.2.1.4 Информация о модуле

Тип: `lta_info_t`		
Описание: Структура, содержащая постоянную информация о модуле, которая как правило не изменяется после открытия		
Поле	**Тип**	**Описание поля**
`devname`	`char [LTA_DEVNAME_SIZE]`	Название устройства
`serial_dev`	`char [LTA_SERIAL_SIZE]`	Серийный номер устройства
`serial_crate`	`char [LTA_SERIAL_SIZE]`	Серийный номер крейта
`brd_rev`	`char [LTA_REVISION_SIZE]`	Ревизия платы
`slot`	`char [LTA_SLOT_NUM_SIZE]`	Номер посадочного слота в крейте

mcu_ver	char [LTA_MCU_VER_SIZE]	Версия прошивки контроллера (из lboot)
fpga_ver	uint32_t	Версия ПЛИС
dev_flags	uint32_t	Флаги lta_dev_flags_t, описывающие особенности устройства
net_settings	lta_eth_info_t	Текущие сетевые настройки
self_diag_err	int32_t	Текущая ошибка самодиагностики модуля lta_self_diag_err_t

4.2.1.5 Запись о сетевых настройках модуля

Тип: lta_eth_cfg_t		
Описание: Структура, содержащая пользовательские сетевые настройки, записанные в память микроконтроллера. Используется для функций записи и чтения настроек.		
Поле	**Тип**	**Описание поля**
flags	uint32_t	Флаги настройки lta_ec_flags_t
mac	uint8_t [LTA_MAC_ADDR_SIZE]	MAC адрес
inst_name	char [LTA_INSTANCE_NAME_SIZE]	строка с именем экземпляра устройства
addr	uint8_t [LTA_IPV4_ADDR_SIZE]	IPv4 адрес
mask	uint8_t [LTA_IPV4_ADDR_SIZE]	Маска подсети
gate	uint8_t [LTA_IPV4_ADDR_SIZE]	Адрес шлюза
cmd_port	uint16_t	Порт передачи команд
data_port	uint16_t	Порт передачи данных
ctrl	uint32_t	Режим работы (резерв)
data	uint8_t [LTA_USER_DATA_SIZE]	Пользовательские данные

4.2.1.6 Калибровочные коэффициенты диапазона.

Тип: lta_cbr_coef_t		
Описание: Структура содержит калибровочные значения смещения нуля и коэффициента шкалы для одного диапазона АЦП или ЦАП.Результирующее значение АЦП вычисляется как (val - offs) * k, где val - неоткалиброванное значение		
Поле	**Тип**	**Описание поля**
offs	double	смещение нуля
k	double	коэффициент шкалы

4.2.1.7 Описатель соединения с модулем

Тип: `lta_hnd_t`
Описание: Непрозрачный указатель на структуру, содержащую информацию о настройках модуля и текущем соединении с ним. Пользовательской программе не доступны поля структуры напрямую, а только через функции библиотеки. Функции принимают описатель соединения с модулем своим первым параметром.

4.3 Функции библиотеки lta_api

4.3.1 Выполнение процедуры поиска модулей в локальной сети

Формат: `int32_t LTA_Search (lta_dev_rec_t *rec_list, uint32_t size, uint32_t *devcount, uint32_t timeout, uint32_t api_version, const lta_dev_fltr_t *filter)`

Описание:

При вызове данной функции выполняется процесс поиска сервисов, соответствующих модулям серии LTA, в локальной сети и сохранение записей о найденных устройствах в массив. Для запуска сеанса необходима запущенная служба (демон) DNS-SD, поддерживаются Bonjour для ОС Windows и Avahi для ОС Linux.

Параметры:

rec_list — Массив для сохранения записей о найденных устройствах. Должен содержать место для сохранения не менее size записей. Если нужно только получить количество модулей в системе, нужно передать rec_list=NULL, size=0 и devcount!=NULL.

size — Максимальное количество записей в массиве rec_list. Если устройств найдено больше, то будут сохранены только первые size записей.

devcount — Указатель на переменную, в которую сохраняется число найденных модулей (может быть больше size). Можно передать NULL, если не используется.

timeout — Время поиска в мс.

api_version — Версия библиотеки `lta_lib_ver_t` для обеспечения совместимости API. Дополнительно, номер актуальной версии библиотеки хранится в константе `LTA_LIB_VER`.

filter — Фильтры для поиска стройств. Можно передать NULL для отключения фильтров.

Возвращаемое значение:

Код ошибки `lta_err_t`.

4.3.2 Заполнение записи типа lta_dev_rec_t для её последующего использования в LTA_Open.

Формат: `int32_t LTA_MakeDevRecordByParams (lta_dev_rec_t *devrec, const char *name, const char *sn, uint8_t slot, uint32_t ip_addr, uint16_t cmd_port)`

Описание:

Параметры:
devrec — Указатель на запись которую заполняем
name — Тип модуля, например, LTA37_DEVNAME
sn — Серийный номер, если надо открыть модуль по серийному номеру или NULL
slot — Номер слота в крейте
ip_addr — Ip адрес модуля
cmd_port — tcp порт соединения, по умолчанию модули настроены на `LTA_CMD_PORT_DEFAULT`

Возвращаемое значение:
Код ошибки `lta_err_t`.

4.3.3 Открытие соединения с модулем

Формат: `int32_t LTA_Open (lta_hnd_t *hnd, const lta_dev_rec_t *dev_rec, uint32_t api_version, uint32_t timeout, uint32_t in_flags, uint32_t *out_flags)`

Описание:

Функция устанавливает соединение с модулем по переданной записи. Должна быть вызвана перед работой с модулем. После завершения работы необходимо закрыть соединение с помощью `LTA_Close()`.

Особенность использования: Пользователю необходимо предварительно инициализировать указатель на описатель модуля нулем (lta_hnd_t hnd = NULL). В противном случае, функция вернет `LTA_ERR_MODULE_IN_USE` (модуль используется или не был закрыт). Для повторного использования описателя, необходимо закрыть предыдущее соединение с помощью `LTA_Close()` и снова вызвать `LTA_Open()`.

Параметры:
hnd — Указатель на описатель модуля.
dev_rec — Структура с параметрами для открытия устройства #lta_open_rec_t.
api_version — Версия библиотеки `lta_lib_ver_t` для обеспечения обратной совместимости API.Дополнительно, номер актуальной версии библиотеки хранится в константе `LTA_LIB_VER`.
timeout — Время на установления подключения в мс.
in_flags — Входные флаги, управляющие логикой работы функции. Может принимать значения из `lta_inflags_t`.
out_flags — Выходные флаги `lta_outflags_t`. Может быть NULL.

Возвращаемое значение:
Код ошибки `lta_err_t`.

4.3.4 Закрытие соединения с модулем

Формат: `int32_t LTA_Close (lta_hnd_t *hnd, uint32_t in_flags)`
Описание: Функция разрывает соединение с модулем, если оно было ранее установлено. Если соединение уже закрыто, то функция возвращает ошибку `LTA_ERR_DEVICE_NOT_OPENED`. Особенность использования: При успешном выполнении функция обнуляет переданный ей указатель.
Параметры: **hnd** — Указатель на описатель модуля. **in_flags** — Входные флаги, управляющие логикой работы функции. Может принимать значения из `lta_inflags_t`.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.5 Проверка, открыто ли соединение с модулем

Формат: `int32_t LTA_IsOpened (lta_hnd_t hnd)`
Описание: Функция проверяет, открыто ли в данный момент соединение с модулем. Если соединение открыто, то функция возвращает `LTA_ERR_OK`. Если же закрыто, то функция возвращает ошибку `LTA_ERR_DEVICE_NOT_OPENED`.
Параметры: **hnd** — Описатель модуля.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.6 Проверка, запущен ли синхронный ввод/вывод

Формат: `int32_t LTA_IsRunning (lta_hnd_t hnd)`
Описание: Функция проверяет запущен ли синхронный ввод вывод. Если сбор данных не запущен, то функция возвращает ошибку `LTA_ERR_STREAM_IS_NOT_RUNNING`, если запущен, то нулевой код ошибки
Параметры: **hnd** — Описатель модуля.
Возвращаемое значение: Код ошибки.

4.3.7 Проверка соединения с модулем

Формат: `int32_t LTA_Ping (lta_hnd_t hnd, uint32_t timeout)`
Описание: Функция отправляет команду модулю запрос, и контролирует его прием в течение заданного времени.
Параметры: **hnd** — Описатель модуля. **timeout** — Время на прием ответа в мс.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.8 Сброс модуля

Формат: `int32_t LTA_ResetModule (lta_hnd_t *hnd)`
Описание: Функция отправляет команду модулю на перезагрузку. Может использоваться для применения сетевых настроек после записи. После выполнения функции автоматически вызывается `LTA_Close()`.
Параметры: **hnd** — Указатель на описатель модуля.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.9 Получение информации о модуле

Формат: `int32_t LTA_GetDevInfo (lta_hnd_t hnd, lta_info_t *info)`
Описание: Функция позволяет получить информацию о модуле с которым установлена связь. Значение копируется из внутренней структуры описателя в переданную структуру.
Параметры: **hnd** — Описатель модуля. **info** — Указатель на структуру lta_info_t, куда будет скопирована информация о модуле.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.10 Чтение пользовательской сетевой конфигурации интерфейса

Формат: `int32_t LTA_EthConfigRead (lta_hnd_t hnd, lta_eth_cfg_t *cfg)`
Описание: Функция читает текущие параметры интерфейса и сохраняет их в переданную структуру.
Параметры: **hnd** — Описатель модуля. **cfg** — Указатель на структуру lta_eth_cfg_t, куда будет скопирована сетевая конфигурация.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.11 Установка пользовательской сетевой конфигурации интерфейса

Формат: `int32_t LTA_EthConfigWrite (lta_hnd_t hnd, uint32_t cmd, const lta_eth_cfg_t *cfg, const char *cur_passwd, const char *new_passwd)`
Описание: Функция позволяет выполнить запись в энергонезависимую память или очистку пользовательской конфигурации сетевого интерфейса. Для выполнения функции нужно передать пароль. Если пароль не был установлен, то нужно передать пустую строку. После выполнения функции необходимо перезагрузить модуль для применения новых настроек. Это можно выполнить с помощью функции `LTA_ResetModule()`. После, необходимо повторно установить соединение с модулем. После выполнения очистки сетевой конфигурации и перезагрузки будут установлены заводские сетевые настройки: IP-адрес: 192.168.0.[номер слота в крейте или 01 в случае одиночного крейта]; Маска: 255.255.255.0; Шлюз: 192.168.0.100; Порт для передачи команд: 11114 (`LTA_CMD_PORT_DEFAULT`). Порт для передачи данных: 11115 (`LTA_DATA_PORT_DEFAULT`).
Параметры: **hnd** — Описатель модуля. **cmd** — Команда `lta_fm_cmd_t` записи или очистки настроек. **cfg** — Указатель на конфигурацию сетевого интерфейса. **cur_passwd** — Строка с текущим паролем. **new_passwd** — Строка с новым паролем.
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.12 Чтение блока данных из пользовательской энергонезависимой памяти модуля

Формат: `int32_t LTA_UserMemRead (lta_hnd_t hnd, uint32_t addr, uint32_t size, uint8_t *dst)`
Описание: Функция считывает данные из выделенного сектора Flash-памяти модуля в переданный массив. Для чтения доступно `LTA_USER_MEM_SIZE` байт памяти.
Параметры: **hnd** — Описатель модуля. **addr** — Адрес начала блока. **size** — Количество байт для чтения. **dst** — Массив, куда будут сохранены считанные данные (должен быть не меньше size байт).
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.13 Запись блока данных в пользовательскую энергонезависимую память модуля

Формат: `int32_t LTA_UserMemWrite (lta_hnd_t hnd, uint32_t addr, uint32_t size, const uint8_t *src)`
Описание: Функция записывает переданный массив данных в выделенный сектор Flash-памяти модуля. Эта область должна быть предварительно стерта с помщью LTA_MemErase(). Для записи доступно `LTA_USER_MEM_SIZE` байт памяти.
Параметры: **hnd** — Описатель модуля. **addr** — Адрес начала блока (должен быть кратен 4). **size** — Количество байт для записи. **src** — Массив с данными (должен быть не меньше size байт).
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.14 Стирание блока данных пользовательской энергонезависимой памяти модуля

Формат: `int32_t LTA_UserMemErase (lta_hnd_t hnd, uint32_t addr, uint32_t size)`
Описание: Функция стирает выделенный сектор Flash-памяти модуля. После, все ячейки будут читаться как 0xFF.
Параметры: **hnd** — Описатель модуля. **addr** — Адрес начала блока (должен быть кратен `LTA_USER_MEM_SIZE`). **size** — Количество байт для стирания (должно быть кратно `LTA_USER_MEM_SIZE`).
Возвращаемое значение: Код ошибки `lta_err_t`.

4.3.15 Получить версию библиотеки

Формат: `uint32_t LTA_GetLibraryVersion (void)`
Описание: Функция возвращает версию библиотеки. Версия возвращается в виде 32-битного числа. Строковое представление возвращенной версии — четыре числа, старшее соответствует старшему байту, младшее — младшему. Старший байт — мажорная версия, второй по старшинству байт — минорная, третий — ревизия, четвертый — номер сборки (не используется — всегда 0)
Параметры: **hnd** — Описатель модуля.
Возвращаемое значение: Версия библиотеки (32-битное число).

4.3.16 Получение строки об ошибке

Формат: `const char * LTA_GetErrorString (int32_t err, lta_err_lang_t lang)`
Описание: Функция возвращает указатель на строку, соответствующую переданному коду ошибки. Следует учесть, что в ОС Windows строка возвращается в стандартной для Windows кодировке CP1251, в то время как в Linux используется кодировка UTF-8.
Параметры: **err** — Код ошибки `lta_err_t`. **lang** — Язык строки `lta_err_lang_t`.
Возвращаемое значение: Указатель на строку, соответствующую коду ошибки.

4.3.17 Получение строки об ошибке самодиагностики

Формат: `const char * LTA_GetSelfDiagErrorString (int32_t err, lta_err_lang_t lang)`
Описание: Функция возвращает строку, соответствующую переданному коду ошибки самодиагностики. Следует учесть, что в ОС Windows строка возвращается в стандартной для Windows кодировке CP1251, в то время как в Linux используется кодировка UTF-8.
Параметры: **err** — Код ошибки `lta_self_diag_err_t`. **lang** — Язык `lta_err_lang_t`.
Возвращаемое значение: Указатель на строку, соответствующую коду ошибки.

4.3.18 Моргание светодиодом

Формат: `int32_t LTA_LedBlink (lta_hnd_t hnd)`
Описание: При вызове этой функции происходит мигание светодиодом на передней панели Может быть использована для визуальной идентификации модуля после его открытия. При запущенном синхронном вводе/выводе данный светодиод всегда горит зеленым цветом и данная функция не влияет на его состояние.
Параметры: **hnd** — Описатель модуля.
Возвращаемое значение: Код ошибки.