

Динамическая библиотека PRG07.dll

Руководство пользователя

Оглавление

История документа.....	7
Введение.....	8
Назначение.....	8
Демонстрационный пример.....	8
Соглашения о вызовах.....	8
Совместимость.....	8
Возможные проблемы.....	8
Установка адресов ethernet.....	9
Утилита EGP.....	9
Интерфес RS-485.....	9
Выбор интерфейса ethernet на ПК.....	10
Протокол нижнего уровня.....	10
Поддержка.....	10
Внутренняя структура считывателя.....	11
Радиоканалы.....	11
Стеки тагов.....	11
Логика доступа.....	12
Блок реле.....	12
Режимы работы считывателя.....	12
Режим мониторинга.....	12
Режим регистрации.....	12
Режим однократного чтения.....	13
Режим без ворот.....	13
Расширенный режим.....	13
Коды ошибок.....	14
Функции библиотеки.....	15
Работа с интерфейсом.....	15
Установка IP – адреса сервера для DLL.....	15
Назначение.....	15
Формат вызова.....	15
Входные параметры.....	15
Возвращаемые значения.....	16
Установка IP – адреса считывателя для DLL.....	16
Назначение.....	16
Формат вызова.....	16
Входные параметры.....	16
Возвращаемые значения.....	16
Получение IP – адреса считывателя для DLL.....	16
Назначение.....	16
Формат вызова.....	16
Входные параметры.....	16

Возвращаемые значения.....	16
Старт работы с интерфейсом DllStart.....	16
Назначение.....	16
Формат вызова.....	16
Входные параметры.....	17
Возвращаемые значения.....	17
Заккрытие интерфейса DllStop.....	17
Назначение.....	17
Формат вызова.....	17
Входные параметры.....	17
Получение статуса интерфейса DllStarted.....	17
Назначение.....	17
Формат вызова.....	17
Входные параметры.....	17
Возвращаемые значения.....	17
Установка адреса RS-485 SetReaderAddr.....	17
Назначение.....	17
Формат вызова.....	17
Входные параметры.....	18
Получение адреса RS-485 GetReaderAddr.....	18
Назначение.....	18
Формат вызова.....	18
Входные параметры.....	18
Возвращаемые значения.....	18
Возвращаемые значения.....	18
Сброс пароля администратора WEB.....	18
Назначение.....	18
Формат вызова.....	18
Входные параметры.....	18
Возвращаемые значения.....	18
Установка пароля администратора WEB.....	19
Назначение.....	19
Формат вызова.....	19
Входные параметры.....	19
Возвращаемые значения.....	19
Получение сетевых настроек.....	19
Назначение.....	19
Формат вызова.....	19
Входные параметры.....	19
Возвращаемые значения.....	19
Установка сетевых настроек.....	20
Назначение.....	20
Формат вызова.....	20

Входные параметры.....	20
Возвращаемые значения.....	20
Получение версии ПО считывателя.....	20
Получение информации о считывателе GetVersion.....	20
Назначение.....	20
Формат вызова.....	20
Входные параметры.....	20
Возвращаемые значения.....	21
Получение версии в виде строки GetVersionStr.....	21
Назначение.....	21
Формат вызова.....	21
Входные параметры.....	21
Возвращаемые значения.....	21
Работа с датой и временем.....	21
Установка даты и времени SetDateTime.....	22
Назначение.....	22
Формат вызова.....	22
Входные параметры.....	22
Получение даты и времени.....	22
Назначение.....	22
Формат вызова.....	22
Входные параметры.....	22
Возвращаемые значения.....	22
Работа со стеком тегов.....	23
Функция GetNumTags.....	23
Назначение.....	23
Формат вызова.....	23
Входные параметры.....	23
Возвращаемые значения.....	23
Функция GetFirstTag.....	23
Назначение.....	23
Формат вызова.....	23
Входные параметры.....	23
Возвращаемые значения.....	23
Функция GetNextTag.....	24
Назначение.....	24
Формат вызова.....	24
Входные параметры.....	24
Возвращаемые значения.....	24
Настройка параметров радиоканалов.....	24
Получить затухание в канале GetDumping.....	24
Назначение.....	24
Формат вызова.....	24

Входные параметры.....	24
Возвращаемые значения.....	24
Установить затухание в канале SetDumping.....	25
Назначение.....	25
Формат вызова.....	25
Входные параметры.....	25
Возвращаемые значения.....	25
Получить время памяти тага.....	25
Назначение.....	25
Формат вызова.....	25
Входные параметры.....	25
Возвращаемые значения.....	25
Установить время памяти тага.....	25
Назначение.....	25
Формат вызова.....	25
Входные параметры.....	26
Возвращаемые значения.....	26
Установка активности радиоканала SetChannelActivity.....	26
Назначение.....	26
Формат вызова.....	26
Входные параметры.....	26
Возвращаемые значения.....	26
Блокировка стека радиоканала SetStackLocked.....	26
Назначение.....	26
Формат вызова.....	26
Входные параметры.....	27
Возвращаемые значения.....	27
Получение статуса радиоканалов GetRfChannelsStat.....	27
Назначение.....	27
Формат вызова.....	27
Входные параметры.....	27
Возвращаемые значения.....	27
Настройка параметров радиоканалов.....	27
Функции подсистемы доступа.....	28
Установить режим работы SetDeviceMode.....	28
Назначение.....	28
Формат вызова.....	28
Входные параметры.....	28
Возвращаемые значения.....	28
Получить текущий режим GetDeviceMode.....	28
Назначение.....	28
Формат вызова.....	28
Входные параметры.....	28

Возвращаемые значения.....	29
Установить конфигурацию доступа SetAccessConfig.....	29
Назначение.....	29
Формат вызова.....	29
Входные параметры.....	29
Возвращаемые значения.....	29
Получить конфигурацию доступа GetAccessConfig.....	30
Назначение.....	30
Формат вызова.....	30
Входные параметры.....	30
Возвращаемые значения.....	30
Интервал вывода тегов SetTagOutInterval.....	30
Назначение.....	30
Формат вызова.....	30
Входные параметры.....	30
Возвращаемые значения.....	31
Установка активного уровня датчиков SetInputLevel.....	31
Назначение.....	31
Формат вызова.....	31
Входные параметры.....	31
Возвращаемые значения.....	31
Получение статуса датчиков GetInputsState.....	31
Назначение.....	31
Формат вызова.....	31
Входные параметры.....	31
Возвращаемые значения.....	32
Работа с транзакциями.....	32
Очистка буфера транзакций ClearTransactions.....	32
Назначение.....	32
Формат вызова.....	32
Входные параметры.....	32
Возвращаемые значения.....	32
Получение очередной транзакции AskTransaction.....	32
Назначение.....	32
Формат вызова.....	32
Входные параметры.....	33
Возвращаемые значения.....	33
Получение числа транзакций GetTranCount.....	33
Назначение.....	33
Формат вызова.....	33
Входные параметры.....	33
Возвращаемые значения.....	34
Статистический радио тест считывателя.....	34

Запуск радиотеста StartRadioTest.....	34
Назначение.....	34
Формат вызова.....	34
Входные параметры.....	34
Возвращаемые значения.....	34
Остановка радиотеста StopRadioTest.....	34
Назначение.....	34
Формат вызова.....	35
Входные параметры.....	35
Возвращаемые значения.....	35
Функции работы с реле считывателя.....	35
Установка активности реле SetRelayActivity.....	35
Назначение.....	35
Формат вызова.....	35
Входные параметры.....	35
Возвращаемые значения.....	35
Установка времени работы реле SetRelayParams.....	36
Назначение.....	36
Формат вызова.....	36
Входные параметры.....	36
Возвращаемые значения.....	36
Получение параметров реле GetRelayParams.....	36
Назначение.....	36
Формат вызова.....	36
Входные параметры.....	36
Возвращаемые значения.....	36
Сброс на заводские установки.....	37
Сброс настроек считывателя ResetReaderParams.....	37
Назначение.....	37
Формат вызова.....	37
Входные параметры.....	37
Возвращаемые значения.....	37
Приложение 1.....	38
Таблица кодов транзакций.....	38
Таблица имен компонентов считывателя.....	39
Для заметок.....	40

История документа

Версия	Дата	Изменения
1.0	15.10.14	Первая редакция документа. Версия библиотеки 1.1.4.x

Введение

Назначение

Данный документ описывает функционал динамической библиотеки PRG07.dll, предназначенной для работы со считывателями дальней идентификации PR-G07.N. Библиотека имеет две реализации: для 32 — битной и 64 — битной версий Windows и поддерживает работу как по интерфейсу RS-485, так и по интерфейсу Ethernet.

Следует иметь в виду, что библиотека не совместима со старой редакцией считывателей PR-G07.

Демонстрационный пример

Для упрощения освоения функционала библиотеки в комплекте поставляется демонстрационный пример, иллюстрирующий использование большинства функций библиотеки.

Пример написан с использованием среды программирования Delphi (Embarcadero RAD studio XE2), однако легко может быть переведен на другие языки программирования (например, C или C++).

Интерфейс демонстрационного примера и комментарии в исходном тексте сделаны на английском языке (возможно, не вполне идеальном), но для программиста это не должно составить проблемы.

Соглашения о вызовах

Библиотека скомпилирована с использованием соглашения о вызовах типа **stdcall**, что обеспечивает совместимость с большинством сред разработки программного обеспечения.

В качестве параметров функций используются типы short, word (unsigned short), uint32 и указатели на массивы (*char или PByteArray являются синонимами).

Структуры, используемые совместно библиотекой и приложением, должны быть упакованы.

Совместимость

Библиотека протестирована на совместимость с 32-х 64-х битными версиями Windows XP, Windows 7. Тестирование под Windows 8 не производилось.

Для работы с интерфейсом RS-485 требуется NIP-A01, другие типы конверторов библиотека не поддерживает.

Возможные проблемы

При начале работе со считывателем с использованием SDK могут возникнуть разного рода проблемы, связанные с некорректными установками, незнанием значений параметров и так далее. Ниже перечислены основные пути разрешения проблемных ситуаций.

Установка адресов ethernet

Если при подключении по интерфейсу ethernet не получается добиться обмена со считывателем, то, скорее всего, проблема в адресации.

Считыватель имеет свой IP адрес, а также IP — адрес сервера. Если адрес сервера (вашего приложения) установлен некорректно, то считыватель на запросы отвечать не будет. Установить требуемые значения сетевых настроек можно как с помощью интерфейса RS-485, так и с помощью встроенного WEB — сервера считывателя или утилиты EGPx.exe (используется для программирования сетевых параметров контроллеров Parsec). Для работы с последним необходимо только знать IP – адрес считывателя.

Если вы не знаете и последнего, то можно аппаратно сбросить все сетевые настройки считывателя на поумолчательные (заводские). Их значения следующие:

- | | |
|------------------------|------------------|
| • IP адрес считывателя | 192, 168, 0, 198 |
| • IP адрес сервера | 192, 168, 0, 141 |
| • IP адрес шлюза | 192, 168, 0, 2 |
| • Маска подсети | 255, 255, 255, 0 |

Теперь вы сможете «достучаться» до считывателя по WEB — интерфейсу и поменять настройки на требуемые. Новые настройки вступят в силу только после перезагрузки считывателя, для чего можно снять и снова подать питание на считыватель, либо нажать кнопку RESET (если корпус считывателя открыт).

Утилита EGP

Данная утилита (EGP3.exe или EGP4.exe) предназначена для конфигурирования сетевых параметров различных устройств торговой марки Parsec, имеющих интерфейс ethernet.

Для получения информации о работе с утилитой обратитесь к ее документации. Скачать утилиту можно с официального сайта www.paresc.ru.

Интерфес RS-485

Динамическая библиотека работает со считывателем по RS-485 только через интерфейс NIP - A01, причем используется первый найденный интерфейс. Если к ПК подключено более одного NIP, программа использует первый найденный, и это может оказаться не тем интерфейсом, к которому подключен считыватель.

Для гарантированного соединения со считывателем проверьте, что к ПК подключен только один NIP.

По интерфейсу RS-485, как и по WEB — интерфейсу, вы можете поменять все адресные параметры считывателя на требуемые. Проще всего для этого использовать штатную утилиту, поставляемую со считывателем.

Необходимо иметь в виду следующие особенности:

- Считыватель обрабатывает запросы и формирует ответы при обращении по его установленному адресу (в диапазоне от 1 до 254).

- Считыватель всегда реагирует на команды по широковещательному адресу (broadcast address), равному 0xFF (255), но никогда не отвечает на принятые по этому адресу пакеты.
- Считыватель обрабатывает запросы и формирует ответы при обращении к нему по нулевому адресу. Используя данный адрес, всегда можно поменять адресные параметры считывателя на требуемые.
При этом обращение по адресу «0» допустимо только в случае, если на линии RS-485 подключен только **один** считыватель.

Выбор интерфейса ethernet на ПК

На современном ПК чаще всего имеется несколько сетевых интерфейсов — это может быть вторая ethernet карта, адаптер WiFi и так далее.

Библиотека WinSoc Windows по умолчанию открывает первый попавшийся интерфейс, и он может оказаться не тем, который нужен для работы со считывателем.

Для исключения недоразумений перед инициализацией библиотеки для работы с интерфейсом ethernet (DllStart()) установить правильный IP адрес сетевого интерфейса с помощью функции SetServerIp().

Протокол нижнего уровня

Если предполагается работать со считывателем с ПК, работающего под отличной от Windows операционной системой, либо для работы со считывателем используется микропроцессорный контроллер (возможно, даже без операционной системы, то полезным окажется описание низкоуровневого протокола обмена.

Работать с низкоуровневым протоколом сложнее, однако в описанных ситуациях это является единственным выходом.

Низкоуровневый протокол описан в отдельном документе.

Поддержка

По всем проблемам, связанным с использованием данной библиотеки, следует обращаться в службу технической поддержки по адресу:

support@parsec.ru

Ваши замечания по документу также можно присылать на указанный адрес.

Внутренняя структура считывателя

Для понимания логики работы считывателя необходимо представлять его внутреннюю структуру и принципы взаимодействия программных компонентов. Упрощенно структуру считывателя можно проиллюстрировать следующим рисунком.

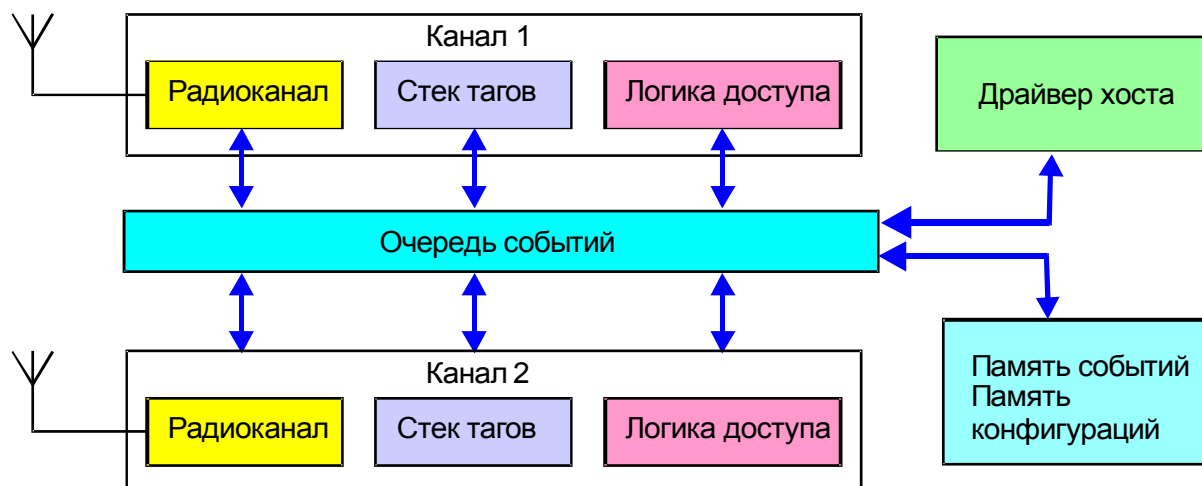


Рисунок 1. Структура считывателя.

Радиоканалы

На входы радиоканалов с находящихся в зоне чтения тагов поступают их коды, состоящие в общем случае из двух компонентов: собственно кода тага, занимающего три младших байта, и байт статуса (старший байт кода). В байте статуса находятся биты кнопок (для кнопочного тага) и бит состояния батарейки.

Каждый радиоканал может быть программно включен или выключен в произвольный момент времени.

Стеки тагов

С выхода радиоканалов, производящих предварительную обработку принятых из эфира пакетов, коды тагов поступают через общую очередь событий в стеки радиоканалов, в которых производится их обработка во времени. При каждом приходе пакета от тага для него взводится счетчик времени памяти тага, который перевзводится при каждом прочтении тага. При первом появлении тага в зоне чтения формируется транзакция появления нового тага в зоне чтения.

Этот счетчик с интервалом в 100 миллисекунд декрементируется, и если таг не выходит в эфир, то по достижении счетчиком нулевого значения формируется транзакция пропадания тага, и он стирается из стека.

Все таги, находящиеся в конкретный момент времени в стеке, могут быть программно прочитаны из стека специальными функциями, то есть внешнее приложение всегда может получить информацию о всех тагах, находящихся в зоне чтения считывателя.

Размер стека ограничен 64 — мя записями, то есть если в поле чтения считывателя находится более 64 — х тагов, то информация о части из них будет потеряна.

Логика доступа

Факты появления и пропадания тагов из зоны чтения поступают через очередь событий на обработку в логику доступа (FSM), где в зависимости от режима работы считывателя производятся те или другие действия. Если в соответствии с выбранным режимом необходимо отправить на внешний контроллер доступа код конкретного тага, то от выводятся через специальную очередь вывода, которая предназначена для исключения перегрузки контроллера доступа при одновременном прочтении сразу нескольких тагов.

Дополнительно в определенных режимах логика доступа обрабатывает факты срабатывания датчиков автоматики ворот.

Блок реле

Считыватель имеет 4 встроенных реле (по 2 на канал, на рисунке не показаны), которые могут быть настроены для работы с внешними устройствами. В каждом канале одно реле может быть запрограммировано на работу по наличию тагов в стеке радиоканала, а второе реле включаться на заданное время при каждой отправке кода тага во внешний контроллер системы доступа.

Режимы работы считывателя

Считыватель имеет несколько режимов работы, устанавливаемых программно, и позволяющих адаптировать считыватель к различным условиям применения.

Режим мониторинга

В общем случае, режим предназначен для слежения за объектами (оборудованными тагами) на подконтрольной территории. Каналы работают всегда отдельно, каждый канал программно может быть включен или выключен, для каждого канала можно установить время памяти тага и затухание в радиоканале, меняющее чувствительность и, как следствие — дальность чтения тагов.

В данном режиме производится фиксация всех тагов, находящихся в зоне чтения считывателя. Дополнительно при появлении каждого тага в зоне чтения на контроллер доступа выдается код тага по входному каналу, а при пропадании тага из зоны чтения — по выходному каналу. Тем самым контроллер доступа при появлении тага фиксирует факт входа, а при пропадании тага — факт выхода владельца тага из контролируемой считывателем области.

Факт выдачи кода тага на контроллер доступа может дублироваться срабатыванием соответствующего реле.

Режим регистрации

Предназначен для регистрации перемещающихся объектов, например, автомобилей вдоль контролируемой зоны. Оба канала всегда работают совместно, перемещение в том или ином направлении определяется порядком пересечения зон чтения радиоканалов. В зависимости от направления перемещения (только после пересечения зон чтения обоих радиоканалов) код тага выдается на один или другой канал внешнего контроллера доступа.

Факт выдачи кода тага на контроллер доступа может дублироваться срабатыванием соответствующего реле.

При установке данного режима оба канала считывателя активируются автоматически.

Режим однократного чтения

Режим может использоваться, например, на автомобильных проходных, оборудованных датчиками присутствия автомобиля в зоне ворот или шлагбаума. До появления автомобиля в зоне датчика присутствия радиоканал заблокирован. При появлении автомобиля в зоне датчика канал активируется на заданное время, и в случае появления в зоне чтения радиоканала в течении этого времени тага его код передается на контроллер доступа, после чего радиоканал снова блокируется до следующего появления автомобиля на датчике присутствия.

В этом режиме каналы могут работать как отдельно, так и в связке. В последнем случае, если автомобиль (его таг) уже был зафиксирован в одном радиоканале (например, на въезде), появление автомобиля в зоне выезда будет проигнорировано, что исключит ложную выдачу кода тага на внешний контроллер доступа.

В данном режиме также можно включить использование признака кнопки — в этом случае при работе тага в кнопочном (не непрерывном) режиме дополнительно осуществляется селекция по нажатой кнопке. При нажатии одной кнопки реакция последует только в первом канале, при нажатии другой кнопки реакция последует только во втором канале считывателя.

При включении режима реакция на датчик присутствия включается автоматически, дополнительно можно задать только активный уровень при срабатывании датчика.

Режим без ворот

Режим позволяет организовать простой механизм доступа через точку проезда без участия датчиков автоматики. Код прочитанного тага сразу выдается на контроллер (если включена опция проверки соседнего канала, то блокируются ложные выдачи кода тага на контроллер при пересечении зоны чтения противоположного радиоканала).

В данном режиме также можно включить опцию использования признака кнопки, когда нажатие одной или другой кнопки приводит к срабатыванию только соответствующего канала считывателя.

Расширенный режим

Наиболее сложный режим работы считывателя, при котором в процессе предоставления доступа обрабатываются датчики присутствия автомобиля и датчики состояния ворот, по которым определяется факт предоставления доступа тагу, код которого был передан на контроллер.

В данном режиме также можно использовать признак кнопки, либо связанную работу каналов.

Для более полного понимания работы в данном режиме следует обратиться к пользовательской документации на считыватель.

Коды ошибок

Все функции библиотеки возвращают результат операции типа word. Нулевой результат соответствует успешному выполнению функции, результат, отличный от нуля характеризует ошибку, возникшую во время выполнения функции.

Коды ошибок библиотеки приведены в таблице ниже.

Код	Мнемоника	Описание
0	RES_NO_ERROR	Функция завершена успешно, без ошибок
1	RES_INVALID_PARAM	Недопустимые значения параметров функции
2	RES_INVALID_COMMAND	Неизвестная команда считывателю
3	RES_BAD_CRC	Ошибка контрольной суммы
4	RES_TIMEOUT	Ответ от считывателя не получен за заданное время
5	RES_INTERFACE_ERR	Ошибка интерфейса (например, при открытии)
6	RES_NO_DATA	Нет данных для возврата вызывающей программе

Если функция завершена с ошибкой (код результата отличен от нуля), то возвращаемые функцией значения не содержат корректной информации.

Функции библиотеки

Работа с интерфейсом

В данной группе собраны функции, так или иначе связанные с работой интерфейсов считывателя. Считыватель имеет три интерфейса для связи с хостом и программирования параметров:

- Интерфейс RS-485. Позволяет подключить несколько считывателей на одну витую пару длиной до 1000 метров. В этом случае каждый считыватель должен иметь на шине уникальный адрес в диапазоне от 1 до 254. Интерфейс поддерживается штатной утилитой считывателя.
- Ethernet. Позволяет работать со считывателем с ПК по стандартной сети ethernet, используется протокол UDP. Интерфейс поддерживается штатной утилитой считывателя.
- WEB — интерфейс. Позволяет программировать считыватель с использованием любого браузера. Требуется подключения к ethernet. Работает независимо от выбора основного интерфейса связи с хостом (RS-485, Ethernet). Для открытия и работы с интерфейсом связи со считывателем имеются три функции, описанные ниже.

Важно, что при старте DLL на ПК должны быть установлены драйвера Parsec для работы с конвертером USB<>RS-485 типа NI-A01. Поскольку при старте библиотеки не известно, через какой интерфейс будет осуществляться обмен со считывателем, библиотека сразу при старте пытается найти драйвера работы с NIP-A01.

При программировании и получении адресов соответствующего интерфейса необходимо знание текущих настроек, в противном случае связь со считывателем будет невозможна. В общем случае можно использовать следующие методы:

- Использовать для настройки всех адресных параметров встроенный WEB — сервер. Для его работы требуется только знание IP – адреса считывателя.
- Если известен адрес по RS-485, то по этому интерфейсу можно установить параметры для интерфейса Ethernet.
- Если известны и правильно установлены IP – параметры, то по интерфейсу Ethernet можно установить адреса для интерфейса RS-485

Установка IP – адреса сервера для DLL

Назначение

Для корректной работы DLL в режиме интерфейса Ethernet перед запуском DLL (до вызова функции DllStart()) необходимо установить ей IP – адрес считывателя, что обеспечивается вызовом данной функции.

Формат вызова

```
function SetServerIp(ipadr: uint32): word;
```

Входные параметры

ipaddr

адрес считывателя на в сети Ethernet.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Установка IP – адреса считывателя для DLL

Назначение

Если в ПК имеется более одного сетевого интерфейса (например, кроме сетевого адаптера Ethernet имеется модуль WiFi), то библиотека WinSoc при открытии сокета использует первый попавшийся адрес, что приведет к невозможности работать со считывателем. В этом случае требуемый IP — адрес следует задать с помощью данной функции до запуска DLL (до вызова функции DllStart()).

Формат вызова

```
function SetReaderIp(ipadr: uint32): word;
```

Входные параметры

ipadr	адрес сетевой платы компьютера, на котором запускается DLL.
-------	---

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получение IP – адреса считывателя для DLL

Назначение

Функция позволяет

Формат вызова

```
function GetReaderIp(var ipadr: uint32): word; stdcall;
```

Входные параметры

нет

Возвращаемые значения

ipadr	возвращаемое значение текущего IP — адреса считывателя.
-------	---

Старт работы с интерфейсом DllStart

Назначение

Функция открывает требуемый интерфейс, открывая канал работы со считывателем.

Формат вызова

```
function DllStart(useethernet: word): word;
```

Входные параметры

В качестве параметра передается тип запрашиваемого интерфейса: при нулевом значении открывается интерфейс RS-485, при положительном ненулевом открывается интерфейс Ethernet.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Закрытие интерфейса DllStop**Назначение**

Функция закрывает открытый ранее интерфейс обмена со считывателем и освобождает занятые при работе ресурсы.

Формат вызова

```
function DllStop(): word;
```

Входные параметры

Функция параметров не имеет.

Получение статуса интерфейса DllStarted**Назначение**

Функция позволяет получить текущее состояние библиотеки: открыт интерфейс работы со считывателем, либо нет.

Формат вызова

```
function DllStarted(): word;
```

Входные параметры

Функция параметров не имеет.

Возвращаемые значения

Функция возвращаемых нулевое значение, если DLL не запущена, и отличное от нуля значение, если DLL стартовала.

Установка адреса RS-485 SetReaderAddr**Назначение**

Функция позволяет назначить адрес считывателя при работе по интерфейсу RS-485. Назначенный адрес сохраняется в энергонезависимой памяти устройства.

Формат вызова

```
function SetReaderAddr(addr: word; newadr: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
newadr	новый адрес считывателя на линии RS-485 в диапазоне от 1 до 254.

Получение адреса RS-485 GetReaderAddr**Назначение**

Функция позволяет получить текущий установленный адрес для интерфейса RS-485. Очевидно, что имеет смысл вызывать данную функцию при работе по интерфейсу Ethernet, если адрес RS-485 неизвестен.

Формат вызова

```
function GetReaderAddr(var adr: word): word;
```

Входные параметры

нет

Возвращаемые значения

adr	адрес считывателя на интерфейсе RS-485.
-----	---

Возвращаемые значения

Функция возвращает нулевое значение, если интерфейс не открыт, и отличное от нуля значение, если интерфейс для работы со считывателем открыт.

Сброс пароля администратора WEB**Назначение**

Функция позволяет обнулить (сбросить) пароль администратора для WEB — интерфейса (например, при утере последнего). По умолчанию пароль не установлен (пустой) и должен назначаться пользователем.

Формат вызова

```
function ClearWebPassword(addr: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Установка пароля администратора WEB

Назначение

Функция позволяет установить новый пароль администратора для WEB — интерфейса считывателя. Пароль можно также установить или поменять в самом WEB — интерфейсе. Пароль сохраняется в энергонезависимой памяти считывателя.

Формат вызова

```
function SetWebPassword(addr: word; buf: pbytearray; len: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

buf	указатель на буфер с паролем в виде нуль — terminated строки символов в кодировке ANSI.
len	длина пароля в символах.

Получение сетевых настроек

Назначение

Функция позволяет получить сетевые настройки считывателя. Имеет смысл при вызове по интерфейсу RS-485, когда сетевые настройки не известны.

Формат вызова

```
function GetNetworkConfig(addr: word; data: pbytearray): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

data	указатель на структуру, куда будут помещены текущие сетевые настройки.
------	--

Структура сетевых настроек имеет следующий вид:

```
#define      MAX_BOARD_NAME_LEN      16

#pragma pack(1)
typedef struct _tNetworkConfig {
    BYTE      IPAddr[IP_ADRLLEN];      // IP адрес считывателя
    BYTE      GateAddr[IP_ADRLLEN];    // IP адрес сетевого шлюза
    BYTE      NetMask[IP_ADRLLEN];     // Маска подсети
    BYTE      ServerAddr[IP_ADRLLEN];  // IP адрес сервера, с кем работать
    BYTE      FirstDnsAddr[IP_ADRLLEN]; // IP адрес первого DNS сервера
```

```
BYTE      SecondDnsAddr[IP_ADRLLEN]; // IP адрес второго DNS сервера
BYTE      MACAddr[ETH_ADRLLEN];      // MAC - адрес считывателя
BYTE      BoardName[MAX_BOARD_NAME_LEN]; // Сетевое имя считывателя
WORD      NetSignature;               // Сигнатура корректности данных
} tNetworkConfig;
```

Если в сети установлен DNS сервер, то к WEB — серверу считывателя можно обращаться как по IP — адресу, так и по имени считывателя. Естественно, что при наличии в сети нескольких считывателей их имена должны отличаться.

Установка сетевых настроек

Назначение

Функция позволяет установить новые сетевые настройки считывателя. Установленные параметры сохраняются в энергонезависимой памяти считывателя и вступают в силу после его перезагрузки.

Формат вызова

```
function SetNetworkConfig(addr: word; data: pbytearray; len: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
data	указатель на структуру, из которой будут скопированы новые сетевые настройки.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получение версии ПО считывателя

Для получения информации о версии программного обеспечения считывателя имеется две функции, описанные ниже.

Получение информации о считывателе GetVersion

Назначение

Функция предназначена для получения полной информации об устройстве — его ID, тип, версия на момент производства и версия на данный момент (например, после обновления прошивки считывателя).

Формат вызова

```
function GetVersion(addr: word; ver: PVersion): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

ver	указатель на структура для возвращаемых параметров.
-----	---

Возвращаемые значения

ver структурв с информацией об устройстве.

Информация об устройстве возвращается в виде следующей структуры:

type

```

TVersion = packed record
    dev_id: word;           // ID в производстве
    devtype: byte;          // Тип устройства
    svsoft: byte;           // Исходная программная версия
    svhard: byte;           // Исходная аппаратная версия
    verhi: byte;            // Старшая текущая версия
    verlo: byte;            // Младшая текущая версия
end;
PVersion = ^TVersion;

```

Получение версии в виде строки GetVersionStr

Назначение

Функция позволяет получить информацию об устройстве в виде текстовой строки.

Формат вызова

```
function GetVersionStr(addr: word; ver: pbytearray): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не имеет.
ver	указатель на буфер для возвращаемой строки. Размер буфера должен быть не менее 32 байтов.

Возвращаемые значения

ver	нуль — терминированная строка с информацией об устройстве.
-----	--

Пример возвращаемого функцией значения:

Reader PR-G07-N Ver. 2.1

Работа с датой и временем

В группу водят две функции для установки и для получения даты и времени из внутренних часов считывателя. Установка даты и времени требуется для корректной работы кольцевого буфера транзакций.

Установка даты и времени SetDateTime

Назначение

Функция предназначена для установки даты и времени во встроенных часах считывателя. Корректное значение даты и времени требуется для правильной интерпретации транзакций считывателя.

Формат вызова

```
function SetDateTime(addr: word; time: uint32; date: uint32): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
time	текущее время в формате: старший байт — день недели следующий байт — часы следующий байт — минуты младший байт — секунды
date	два старших байта — год следующий байт — месяц младший байт — число месяца

Получение даты и времени

Назначение

Функция позволяет получить текущие дату и время из часов считывателя (например, для проверки корректности хода часов).

Формат вызова

```
function GetDateTime(addr: word; var time: uint32; var date: uint32): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

time	текущее время в формате: старший байт — день недели следующий байт — часы следующий байт — минуты младший байт — секунды
date	два старших байта — год следующий байт — месяц младший байт — число месяца

Работа со стеком тагов

Данная группа функций позволяет в реальном времени отслеживать таги, находящиеся в стеках каждого из радиоканалов.

Функция GetNumTags

Назначение

Функция позволяет получить количество тагов в стеке заданного радиоканала на момент вызова.

Формат вызова

```
function GetNumTags(addr: word; chan: word; var num: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)

Возвращаемые значения

num	число тагов в стеке канала
-----	----------------------------

Функция GetFirstTag

Назначение

Функция позволяет получить код первого тага в стеке заданного радиоканала и одновременно спозиционировать внутренний указатель на начало стека. Обязательно вызывается один раз перед первым вызовом функции GetNextTag.

Формат вызова

```
function GetFirstTag(addr: word; chan: word; var tag: uint32): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)

Возвращаемые значения

tag	4 – х байтовый код тага
-----	-------------------------

Функция GetNextTag

Назначение

Функция предназначена для получения кода очередного тага из стека радиоканала. Перед первым вызовом данной функции обязательно должна быть вызвана функция GetFirstTag.

Признаком того, что все таги из стека выбраны, является нулевой код тага.

Формат вызова

```
function GetNextTag(addr: word; chan: word; var tag: uint32): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)

Возвращаемые значения

tag	4 – х байтовый код очередного тага
-----	------------------------------------

Настройка параметров радиоканалов

Группа функций предназначена для работы с параметрами радиоканалов — затуханием, временем памяти тагов и настройками радиоканала.

Получить затухание в канале GetDumping

Назначение

Функция предназначена для получения величины затухания заданного радиоканала считывателя.

Формат вызова

```
function GetDumping(addr: word; chan: word; var dump: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)

Возвращаемые значения

dump	величина затухания в децибеллах от 0 до 31.
------	---

Установить затухание в канале SetDumping

Назначение

Функция предназначена для установки величины затухания заданного радиоканала считывателя.

Формат вызова

```
function SetDumping(addr: word; chan: word; dump: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)
dump	величина затухания в децибеллах от 0 до 31.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получить время памяти тага

Назначение

Функция предназначена для получения времени памяти тага в заданном радиоканале считывателя.

Формат вызова

```
function GetTagMemTime(addr: word; chan: word; var time: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)

Возвращаемые значения

time	время памяти тага в радиоканале в квантах по 100 мсек.
------	--

Установить время памяти тага

Назначение

Функция предназначена для установки времени памяти тага в заданном радиоканале считывателя.

Формат вызова

```
function SetTagMemTime(addr: word; chan: word; time: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала (1 или 2)
time	время памяти тага в радиоканале в квантах по 100 мсек.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Установка активности радиоканала SetChannelActivity**Назначение**

Функция предназначена для отдельного включения или выключения радиоканалов считывателя. Выключение радиоканала может потребоваться, например, при использовании считывателя в одноканальном режиме при мониторинге объектов в поле зрения устройства.

Формат вызова

```
function SetChannelActivity(addr: word; chan: word; state: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер радиоканала (1 или 2)
state	устанавливаемое состояние: «0» - канал не активен (отключен), «1» - канал включен (активен).

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Блокировка стека радиоканала SetStackLocked**Назначение**

Функция предназначена для блокировки обработки тагов в стеке радиоканала считывателя. В заблокированном состоянии не производится обработка уже имеющихся в стеке тагов и не заносятся новые таги, то есть стек «замораживается» в текущем состоянии.

Формат вызова

```
function SetStackLocked(addr: word; chan: word; state: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер радиоканала (1 или 2)
state	устанавливаемое состояние: «1» - стек заблокирован, «0» - стек активен.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получение статуса радиоканалов GetRfChannelsStat**Назначение**

Функция предназначена для получения статуса радиоканалов считывателя.

Формат вызова

```
function GetRfChannelsStat(addr: word; var actch1: word; var actch2: word;  
var locked1: word; var locked2: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

actch1, actch2	активность соответственно первого и второго радиоканалов (устанавливается функцией SetChannelActivity()).
locked1, locked 2	состояние блокировки стеков соответственно первого и второго радиоканалов, устанавливается функцией SetStackLocked().

Настройка параметров радиоканалов

Для изменения настроек радиоканалов (частотного канала, адреса, формата проверочной контрольной суммы и ряда других) и получения значений этих настроек имеются соответствующие две функции:

```
function GetRadioConfig(addr: word; chan: word; data: pbytearray): word;
```

и

```
function SetRadioConfig(addr: word; chan: word; data: pbytearray): word;
```

Функции не предназначены для использования в пользовательском режиме, так как изменение любого из параметров радиоканалов данными функциями приведет к несовместимости со стандартными тагами.

В случае, если по каким-то причинам настройки радиоканалов сбились, их можно восстановить с помощью функции `ResetReaderParams()`, которая восстанавливает все по умолчанию (заводские) настройки считывателя.

Описание функции приведено далее в данном документе.

Функции подсистемы доступа

К данной группе отнесены функции, определяющие работу считывателя в составе системы доступа. К этой же группе отнесены и функции изменения режима работы считывателя.

Установить режим работы `SetDeviceMode`

Назначение

Функция предназначена для установки режима работы считывателя, который, в свою очередь, определяет алгоритм обработки входных воздействий (появление или исчезновение тага, срабатывание датчиков автоматики ворот).

Формат вызова

```
function SetDeviceMode(addr: word; mode: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
mode	устанавливаемый режим работы: 0 — режим мониторинга 1 — режим однократного чтения 2 — режим регистрации 3 — режим без ворот 4 — расширенный режим

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получить текущий режим `GetDeviceMode`

Назначение

Функция предназначена для получения текущего режима работы считывателя.

Формат вызова

```
function GetDeviceMode(addr: word; var mode: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

mode	текущий режим работы считывателя:
	0 — режим мониторинга
	1 — режим однократного чтения
	2 — режим регистрации
	3 — режим без ворот
	4 — расширенный режим

Установить конфигурацию доступа SetAccessConfig**Назначение**

Функция предназначена для установки основных параметров подсистемы доступа независимо от режима работы считывателя. В различных режимах используются различные наборы параметров из данной структуры.

Конфигурации каналов могут различаться в зависимости от требований конкретной инсталляции.

Формат вызова

```
function SetAccessConfig(addr: word; chan: word; cfg: PAccessConfig): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала, для которого устанавливается конфигурация (1 или 2).
cfg	указатель на структуру, содержащую устанавливаемую конфигурацию подсистемы доступа.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Структура конфигурации подсистемы доступа выглядит следующим образом:

```
type
```

```

TAccessConfig = packed record
    // Битовые параметры (значения 0 или 1)
    CheckOpposit      : byte; // Проверять таг в противоположном канале
    UseCarSensor       : byte; // Использовать датчик автомобиля
    CarSensInversion   : byte; // Инверсия датчика автомобиля
    UseGateSensor      : byte; // Использовать датчик ворот
    GateSensInversion  : byte; // Инверсия датчика ворот
    ResetTimeByDc      : byte; // Сброс таймера по датсикку ворот
    RemovePassedTag    : byte; // Удалять таг после проезда
    NoReadWhileAccess  : byte; // Не читать таги в цикле доступа
    UseButtonDir       : byte; // Использовать кнопку для селекции канала
    // Числовые параметры
    TagWaitTime        : word; // Ожидание тага при однократном
    CarWaitTime        : word; // Ожидание автомобиля на датчике
    AccWaitTime        : word; // Ожидание предоставления доступа

```

```
GateOpenTime      : word;  // Время открытых ворот
AccDefaultTime    : word;  // Время доступа по умолчанию (без датчиков)
Reserv1           : word;
Reserv2           : word;
end;
// Тип указателя на структуру
PAccessConfig = ^TAccessConfig;
```

Назначения параметров понятны из комментариев к каждому полю структуры.

Получить конфигурацию доступа GetAccessConfig

Назначение

Функция предназначена для получения параметров подсистемы для заданного канала считывателя.

Формат вызова

```
function GetAccessConfig(addr: word; chan: word; cfg: PAccessConfig): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер канала, для которого запрашивается конфигурация (1 или 2).

Возвращаемые значения

cfg	указатель на структуру, содержащую установленную конфигурацию подсистемы доступа заданного канала.
-----	--

Интервал вывода тагов SetTagOutInterval

Назначение

Функция предназначена для установки интервала вывода тагов на контроллер доступа. Поскольку в некоторых ситуациях за короткое время может быть прочитано более одного тага, для исключения потерь информации о них таги помещаются в специальную очередь, из которой выводятся на контроллер доступа с заданными интервалами.

Формат вызова

```
function SetTagOutInterval(addr: word; time: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
time	интервал вывода тагов в квантах по 100 мсек.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Установка активного уровня датчиков SetInputLevel

Назначение

Функция предназначена для установки активного уровня датчиков автоматики ворот, то есть уровня (низкий или высокий), который появляется на входе в момент срабатывания конкретного датчика.

Заметьте, что значение активного уровня для подсистемы доступа может быть задано в конфигурации доступа (функция SetAccessConfig()) параметрами инверсии соответственно датчиков присутствия автомобиля и датчиков ворот.

Формат вызова

```
function SetInputLevel(addr: word; name: word; number: word; data: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
name	имя датчика автоматики, может принимать значения: NAME_SENS (0x15) - датчик автомобиля NAME_DC (0x11) - датчик ворот
number	номер датчика (канала считывателя) может принимать значения 1 или 2.
data	значение активного уровня, может принимать значения «0» (активный низкий уровень) или «1» (активный высокий уровень).

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получение статуса датчиков GetInputsState

Назначение

Функция позволяет получить текущее значение входов датчиков автоматики ворот считывателя: датчиков автомобиля и датчиков ворот.

Формат вызова

```
function GetInputsState(addr: word; var stat: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

stat	состояние входов датчиков автоматики в вибе битового поля, соответствие битов следующее: бит 0 — датчик ворот канала 1 бит 1 — датчик автомобиля канала 1 бит 2 — датчик ворот канала 2 бит 3 — датчик автомобиля канала 2
------	--

Работа с транзакциями

Считыватель имеет кольцевой энергонезависимый буфер на 1000 транзакций, общий для всех компонентов считывателя. События всегда сначала сохраняются в указанном буфере, а по запросу хоста могут ему передаваться. Отправленная хосту транзакция из буфера удаляется.

При переполнении кольцевого буфера ранее записанные транзакции затираются, начиная с самых старых.

Транзакции содержат поля даты и времени, заполняемые из встроенных энергонезависимых часов считывателя, в связи с чем для поддержания актуальности меток даты и времени необходимо периодически синхронизировать часы (см. выше раздел «Работа с датой и временем»).

Для работы с транзакциями доступны следующие функции:

Очистка буфера транзакций ClearTransactions

Назначение

Функция предназначена для очистки встроенного энергонезависимого буфера транзакций считывателя.

Формат вызова

```
function ClearTransactions(addr: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получение очередной транзакции AskTransaction

Назначение

Функция предназначена для получения очередной транзакции из энергонезависимого буфера транзакций считывателя.

Формат вызова

```
function AskTransaction(addr: word; var tran: TTransaction): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
tran	указатель на буфер, куда помещается очередная транзакция.

Возвращаемые значения

tran	очередная транзакция, если буфер не пуст. При пустом буфере все поля структуры возвращаются заполненными нулями.
------	--

Транзакция представляет собой структуру следующего вида:

```
type
  TTransaction = packed record
    addr: byte;           // Адрес считывателя на линии RS-485
    srcname: byte;        // Имя компонента, породившего транзакцию
    srcnumber: byte;      // Номер компонента, породившего транзакцию
    entname: byte;        // Имя детали, породившей транзакцию
    entnumber: byte;      // Номер детали, породившей транзакцию
    code: WORD;           // Код транзакции
    year: word;           // Год
    month: byte;          // Месяц
    day: byte;            // День
    hour: byte;           // Часы
    minute: byte;         // Минуты
    sec: byte;            // Секунды
    card: uint32;         // Номер тага
    extdata: uint32;      // Дополнительные данные
  end;
```

В Приложении 1 приведены таблицы с кодами транзакций, а также именами компонентов, которые могут порождать транзакции.

Получение числа транзакций GetTranCount**Назначение**

Функция предназначена для получения количества транзакций, находящихся на момент вызова функции в энергонезависимом буфере транзакций считывателя.

Формат вызова

```
function GetTranCount(addr: word; var cnt: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

cnt	количество транзакций в буфере.
-----	---------------------------------

Статистический радио тест считывателя

Статистический радиотест предназначен в большей степени для контроля считывателей в процессе производства, однако может оказаться полезным и при установке системы (например, для оценки зон чтения радиоканалов).

В процессе теста подсчитывается количество прочтений кодов эталонного и тестового тагов (названия тагов условные), и по завершению теста эти количества передаются вызывающей программе.

Тест одновременно может быть запущен только для одного из радиоканалов.

При радиотесте используется текущее установленное затухание радиоканалов считывателя.

Находящиеся в поле чтения таги с отличающимися от заданных для теста номерами (кодами) на работу теста не влияют при условии, что немного (в пределах 10 — 15 штук).

Запуск радиотеста StartRadioTest

Назначение

Функция запускает статистический радиотест для заданного радиоканала считывателя.

Формат вызова

```
function StartRadioTest(addr: word; chan: word; etalontag: uint32; testtag: uint32): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер радиоканала для теста (1 или 2).
etalontag	код эталонного тага (с нулевым старшим байтом).
testtag	код тестового тага (с нулевым старшим байтом).

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Остановка радиотеста StopRadioTest

Назначение

Функция останавливает статистический радиотест для заданного радиоканала считывателя и возвращает количество прочтений тагов за время теста.

Формат вызова

```
function StopRadioTest(addr: word; var chan: word; var etaloncount: word;  
var testcount: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
chan	номер радиоканала для теста (1 или 2).

Возвращаемые значения

etaloncount	количество прочтений эталонного тага.
testcount	количество прочтений тестового тага.

Функции работы с реле считывателя

Считыватель имеет четыре реле, по два на каждый канал. Функции реле фиксированы: одно из реле может находиться во включенном состоянии при наличии хотя бы одного тага в стеке радиоканала, второе реле срабатывает на заданное время в момент отправки кода тага в контроллер доступа соответствующего канала.

По умолчанию реле не работают, если необходимо их использование, то следует активировать реле функцией SetRelayActivity().

Установка активности реле SetRelayActivity**Назначение**

Функция позволяет включить в работу реле считывателя: реле наличия тагов в стеке и реле отправки тагов в контроллер доступа. Активность соответствующего реле включается сразу для обоих каналов.

Формат вызова

```
function SetRelayActivity(addr: word; lock: word; aux: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
lock	устанавливает активность реле отправки тага в контроллер доступа: «0» - отключить реле, «1» - включить реле.
aux	устанавливает активность реле наличия тагов в стеке радиоканалов: «0» - отключить реле, «1» - включить реле.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Установка времени работы реле SetRelayParams

Назначение

Функция позволяет назначить время работы реле (время, в течение которого реле остается во включенном состоянии после срабатывания).

Формат вызова

```
function SetRelayParams(addr: word; num: word; tim: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
num	номер реле отправки тага в контроллер доступа, 1 или 2.
tim	время срабатывания реле в квантах по 100 мсек, допустимые значения от 100 мсек до 100 секунд.

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Получение параметров реле GetRelayParams

Назначение

Функция позволяет получить текущие настройки реле считывателя: их активность и время срабатывания.

Формат вызова

```
function GetRelayParams(addr: word; var aux: word; var lock: word; var t1: word; var t2: word): word;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

aux	активность обоих реле наличия тагов в стеке: «0» - не активны, «1» - активны.
lock	активность обоих реле отправки тага в контроллер: «0» - не активны, «1» - активны.
t1	время срабатывания реле первого канала
t2	время срабатывания реле второго канала

Сброс на заводские установки

Приведенная ниже функция позволяет восстановить все основные заводские установки считывателя, а именно:

- Очищается буфер событий (транзакций)
- Все конфигурации сбрасываются на поумолчательные значения
- Устанавливается поумолчательная конфигурация радиоканалов
- Устанавливается поумолчательная конфигурация подсистемы доступа
- Устанавливается первый адрес по интерфейсу RS-485

После выполнения указанных операций считыватель автоматически перезагружается для актуализации сделанных изменений.

Значения поумолчательных параметров считывателя можно найти в Руководстве по эксплуатации.

Сброс настроек считывателя ResetReaderParams

Назначение

Функция сбрасывает все настройки считывателя (кроме сетевых настроек — они сбрасываются аппаратно) на поумолчательные значения.

Формат вызова

```
function ResetReaderParams(addr: word): word; stdcall;
```

Входные параметры

addr	адрес считывателя на линии RS-485. При работе по Ethernet значение параметра не принципиально.
------	--

Возвращаемые значения

Функция возвращаемых параметров не имеет.

Приложение 1

Таблица кодов транзакций

При анализе смыслового значения транзакций следует, кроме кода, анализировать имя источника и его номер.

Мнемоника	Код	Описание
TRN_NEW_TAG_FOUND	0x3001	В стеке появился новый таг
TRN_TAG_LOSS	0x3002	Таг удален из стека (по времени)
TRN_DC_IS_ON	0x3003	Датчик ворот включился
TRN_DC_IS_OFF	0x3004	Датчик ворот выключился
TRN_SENS_IS_ON	0x3005	Датчик автомобиля включился
TRN_SENS_IS_OFF	0x3006	Датчик автомобиля выключился
TRN_SEND_TAG_1	0x3007	Таг отправлен на контроллер по входному каналу
TRN_SEND_TAG_2	0x3008	Таг отправлен на контроллер по выходному каналу
TRN_LOCKED	0x3009	Заблокирован (радиоканал или стек)
TRN_UNLOCKED	0x300A	Разблокирован (радиоканал или стек)
TRN_SET_ON	0x300B	Включено (см. имя источника)
TRN_SET_OFF	0x300C	Выключено (см. имя источника)
TRN_TAG_IN_OPPOSIT	0x300D	Таг в противоположном канале
TRN_STACK_CLEARED	0x300E	Стек очищен
TRN_TAG_REMOVED	0x300F	Таг удален из стека
TRN_WAIT_CAR	0x3010	Ожидание автомобиля (датчик SENS)
TRN_WAIT_ACCESS	0x3011	Ожидание доступа (датчик DC)
TRN_WAIT_DEF_GATE	0x3012	Ожидание поумолчательного времени ворот
TRN_WAIT_GATE_CLOSE	0x3013	Ожидание закрывания ворот
TRN_ACCESS_FINISH	0x3014	Цикл доступа завершен
TRN_CAR_NOT_FOUND	0x3015	Автомобиль не появился в зоне датчика
TRN_GATE_NOT_OPEN	0x3016	Ворота не открылись (по DC)
TRN_GATE_NOT_CLOSE	0x3017	Ворота не закрылись (по DC)
TRN_WAIT_DEF_ACCESS	0x3018	Ожидание поумолчатльного времени доступа
TRN_STACK_OVERFLOW	0x3019	Переполнение стека тагов в канале

Таблица имен компонентов считывателя

Мнемоника	Описание	Код
NAME_LOG_INPUTS	Логический вход	0x10
NAME_DC	Датчик ворот (шлагбаума)	0x11
NAME_SENS	Датчик присутствия автомобиля	0x15
NAME_RFCHANNEL	Радиоканал	0x19
NAME_TAG_STACK	Стек тагов	0x1A
NAME_TAG_QUEUE	Очередь тагов для вывода на контроллер доступа	0x1B
NAME_MAINS	Сетевое питание	0x23
NAME_CONTROLLER	Контроллер (считыватель)	0xC0
NAME_HOST	Драйвер хоста	0xC2
NAME_MEMORY	Память контроллера (считывателя)	0xC3

Для заметок

[illegible]