

Динамическая библиотека R2x245.dll

Версия 1.2

октябрь 2010 г.

Оглавление

Оглавление.....	1
История документа.....	4
Введение.....	5
Что нового в этом документе.....	5
Назначение.....	5
Демонстрационный пример.....	5
Соглашения о вызовах.....	5
Совместимость.....	5
Коды ошибок.....	6
Функции библиотеки.....	7
Работа с интерфейсом.....	7
Создание списка интерфейсов InterfaceEnumerate.....	7
Описание.....	7
Возвращаемые результаты.....	7
Получение описателя первого интерфейса.....	8
Описание.....	8
Возвращаемые результаты.....	8
Получение описателя следующего интерфейса.....	8
Описание.....	8
Возвращаемые результаты.....	8
Открытие интерфейса InterfaceOpen.....	8
Описание.....	8
Возвращаемые результаты.....	8
Закрытие интерфейса InterfaceClose.....	9
Описание.....	9
Возвращаемые результаты.....	9
Функции – утилиты.....	10
Получение информации о версии считывателя GetReaderVersion.....	10
Описание.....	10
Возвращаемые результаты.....	10
Получение даты GetReaderDate.....	10
Описание.....	10
Возвращаемые результаты.....	10
Получение времени GetReaderTime.....	11
Описание.....	11
Возвращаемые результаты.....	11
Установка даты SetReaderDate.....	11
Описание.....	11
Возвращаемые результаты.....	11
Установка времени SetReaderTime.....	11

Описание.....	11
Возвращаемые результаты.....	11
Смена адреса считывателя SetNewReaderAddr.....	12
Описание.....	12
Возвращаемые результаты.....	12
Программирование параметров каналов.....	13
Установка времени памяти тага в радиоканале SetTagMemTime_1.....	13
Описание.....	13
Возвращаемые результаты.....	13
Установка времени памяти тага в радиоканале SetTagMemTime_2.....	13
Описание.....	13
Возвращаемые результаты.....	13
Получить время памяти тага GetTagMemTime_1.....	13
Описание.....	13
Возвращаемые результаты.....	13
Получить время памяти тага GetTagMemTime_2.....	14
Описание.....	14
Возвращаемые результаты.....	14
Установка затухания SetDumping_1.....	14
Описание.....	14
Возвращаемые результаты.....	14
Установка затухания SetDumping_2.....	14
Описание.....	14
Возвращаемые результаты.....	14
Получить затухание в первом канале GetDumping_1.....	15
Описание.....	15
Возвращаемые результаты.....	15
Получить затухание во втором канале GetDumping_2.....	15
Описание.....	15
Возвращаемые результаты.....	15
Установка активности канала SetChannelActivity_1.....	15
Описание.....	15
Возвращаемые результаты.....	15
Установка активности канала SetChannelActivity_2.....	16
Описание.....	16
Возвращаемые результаты.....	16
Функции для работы со стеком тагов.....	17
Получить число тагов GetNumTags_1.....	17
Описание.....	17
Возвращаемые результаты.....	17
Получить первый tag GetFirstTag_1.....	17
Описание.....	17
Возвращаемые результаты.....	17

Получить следующий таг GetNextTag_1.....	17
Описание.....	17
Возвращаемые результаты.....	17
Получить число тагов GetNumTags_2.....	18
Описание.....	18
Возвращаемые результаты.....	18
Получить первый таг GetFirstTag_2.....	18
Описание.....	18
Возвращаемые результаты.....	18
Получить следующий таг GetNextTag_2.....	18
Описание.....	18
Возвращаемые результаты.....	18
Функции для работы с буфером транзакций.....	19
Очистка буфера транзакций ClearTransactions.....	19
Описание.....	19
Возвращаемые результаты.....	19
Получение числа транзакций в буфере GetNumTransactions.....	19
Описание.....	19
Возвращаемые результаты.....	19
Получение следующей транзакции GetNextTransaction.....	19
Описание.....	19
Возвращаемые результаты.....	20
Включение и выключение аудита AuditOnOff.....	20
Описание.....	20
Возвращаемые результаты.....	20
Приложение 1.....	21
Коды транзакций считывателя.....	21
Для заметок.....	22

История документа

Версия	Дата	Изменения
1.0	25.10.2007	Первая редакция документа
1.1	21.11.2007	Версия 1.1.2.x DLL, добавлены функции SetNewReaderAddr, ClearTransactions, GetNumTransactions, GetNextTransaction
1.2	05.09.2010	Добавлено описание включения и исключения аудита, коды транзакций.

Введение

Данный документ и описанная в нем динамическая библиотека являются рабочей редакцией. В дальнейшем функционал библиотеки может расширяться в соответствии с потребностями пользователей и изменением функционала считывателей, а документация дополнена. При этом зафиксированный в данном документе функционал останется без изменений.

Что нового в этом документе

Данный документ описывает функционал динамической библиотеки R2x245.dll. В отличие от библиотеки R245.dll, текущая версия DLL не требует поддержки TLP.dll. При этом работа со считывателями поддерживается как через COM – порты (реальные или виртуальные), так и через NIP-01 (USB).

Работа с несколькими считывателями на одной линии RS-485 библиотекой также поддерживается.

Версия 1.1 документа дополнена описанием функций, введенных в библиотеку версии 1.1.2.x: SetNewReaderAddr, ClearTransactions, GetNumTransactions, GetNextTransaction.

Назначение

Динамическая библиотека R245.dll предназначена для поддержки считывателей дальней идентификации RP-G07-M и инкапсулирует протокол обмена со считывателями нижнего уровня, предоставляя программисту более простой прикладной интерфейс (API).

Библиотека предназначена для создания приложений, использующих считыватели PR-G07 в режиме мониторинга.

Демонстрационный пример

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

Пример написан с использованием среды программирования Delphi 5 (Object Pascal), однако легко может быть переведен на другие языки программирования (например, C или C++).

Для версии библиотеки 1.1.2.x демонстрационный пример доработан.

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), указатели на массивы (* char, PByteArray).

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

Совместимость

Данный документ полностью соответствует функционалу библиотеки R2x245.dll с номером версии 1.1.2.x.

Коды ошибок

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

Коды ошибок библиотеки приведены в таблице 1 ниже. Вместе с демонстрационным примером поставляется файл ErrCodes.pas, содержащий декларации констант кодов ошибок библиотеки.

Таблица 1

Код	Мнемоника	Описание
0	R245_RES_OK	Функция завершена успешно
1	R245_ERR_WRONG_COMPORT	Недопустимый номер COM – порта, либо COM – порт уже открыт
2	R245_ERR_WRONG_BAUD	Неправильное значение скорости для COM – порта
3	R245_ERR_COM_CLOSED	Выбранный COM – порт закрыт
4	R245_ERR_CLOSE_PORT	Ошибка закрытия COM – порта
5	R245_ERR_COM_WRITEERR	Ошибка записи в COM – порт
6	R245_ERR_COM_TIMEOUT	Таймаут при приеме ответа считывателя
7	R245_ERR_OPEN_PORT	Ошибка открытия COM – порта
8	R245_ERR_OVERFLOW	Переполнение буфера обмена
9	R245_ERR_INVALID_PACK_NUM	Неправильный номер пакета при обмене со считывателем
20	R245_ERR_WRONG_PARAM	Переданный функции параметр недопустим (выходит за диапазон)
21	R245_ERR_INVALID_DATA	Получены некорректные данные
22	R245_ERR_CONNECT	Ошибка соединения (не используется)
80	R245_ERR_NO_TAG	Нет тага в поле считывателя (при попытке получения номера тага)
200	R245_ERR_UNKNOWN	Неизвестная ошибка
201	R245_ERR_NO_IMLEMENT	Функция не реализована в считывателе
202	R245_ERR_BYSY_RADIO	Считыватель занят тестом радиоканала

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

Функции библиотеки

Работа с интерфейсом

Прежде, чем связываться со считывателем, необходимо открыть интерфейс, в качестве которого может выступать либо COM – порт, либо USB интерфейс типа NIP-01.

Общая последовательность доступа к интерфейсу следующая:

- Необходимо определить все доступные интерфейсы (функция `InterfaceEnumerate()`)
- Получить описания (номер и текстовое описание) всех доступных интерфейсов – функции `InterfaceGetFirst()` и `InterfaceGetNext()`
- Открыть выбранный интерфейс по его номеру функцией `InterfaceOpen()`

По окончании работы программы перед ее завершением следует закрыть ранее открытый интерфейс с помощью функции `InterfaceClose()`.

Описатели интерфейсов возвращаются в упакованной структуре следующего вида:

```
type
  TInterfDescriptor = packed record
    Number: short;
    Desc:   packed array[0..DESC_STR_LEN - 1] of char;
  end;
```

где `DESC_STR_LEN = 64`.

Создание списка интерфейсов `InterfaceEnumerate`

Функция проверяет наличие COM – портов и подключенных к ПК конверторов типа NIP-01 и создает их внутренний список для дальнейшего использования.

Описание

```
function InterfaceEnumerate(): short
```

Возвращаемые результаты

В качестве результата функция возвращает число обнаруженных (доступных) интерфейсов. Устройства типа NIP-01 всегда находятся в начале данного внутреннего списка.

Примечание: это единственная функция библиотеки, которая возвращает не результат выполнения, а число интерфейсов.

Получение описателя первого интерфейса

Функция позиционируется на начало внутреннего списка интерфейсов и возвращает первый из описателей (его индекс равен нулю).

Описание

```
function InterfaceGetFirst(var desc: TInterfDescriptor): short
```

Возвращаемые результаты

При успешном получении первого интерфейса в переменной desc возвращается первый интерфейс с номером Number = 0 и описателем Desc (например, COM1). Строка описателя Desc всегда заканчивается нулевым байтом.

Если доступен хотя бы один интерфейс, функция возвращает результат R245_RES_OK, если не обнаружено ни одного интерфейса, возвращается результат R245_ERR_WRONG_COMPORT.

Получение описателя следующего интерфейса

Последовательные вызовы данной функции обеспечивают получение списка всех остальных доступных интерфейсов. Функцию следует вызвать до тех пор, пока код возврата не станет неравным R245_RES_OK – это признак того, что список исчерпан.

Описание

```
function InterfaceGetNext(var desc: TInterfDescriptor): short
```

Возвращаемые результаты

Функция возвращает R245_RES_OK до тех пор, пока не будет исчерпан внутренний список интерфейсов, полученный при вызове функции InterfaceEnumerate.

Формат возвращаемого описателя аналогичен функции InterfaceGetFirst.

Открытие интерфейса InterfaceOpen

Функция открывает для последующей работы интерфейс, заданный его номером от 0 до N.

Описание

```
function InterfaceOpen(num: short): short
```

Возвращаемые результаты

При успешном открытии интерфейса возвращается код R245_RES_OK. Если интерфейс занят другим приложением, либо возникли другие проблемы при его открытии, возвращается результат R245_ERR_OPEN_PORT или R245_ERR_WRONG_COMPORT.

Закрытие интерфейса InterfaceClose

Функция закрывает ранее открытый интерфейс.

Описание

```
function InterfaceClose(): short
```

Возвращаемые результаты

Функция всегда возвращает значение R245_RES_OK.

Функции – утилиты

Все функции, описываемые в данном и последующих разделах, в качестве первого параметра имеют адрес считывателя на линии RS-485.

При описании кодов возвращаемых значений не перечисляются коды ошибок интерфейса

R245_ERR_COM_CLOSED,
R245_ERR_COM_WRITEERR,
R245_ERR_COM_TIMEOUT,
R245_ERR_OVERFLOW,
R245_ERR_INVALID_PACK_NUM,

которые могут возникать при любом обмене со считывателем – указываются только коды, характерные для данной функции.

Получение информации о версии считывателя `GetReaderVersion`

Функция возвращает строковое описание текущей версии ПО считывателя. При вызове функции передается указатель на буфер `ver` и его длина `maxlen`.

Описание

```
function GetReaderVersion(adr: short; ver: PByteArray; maxlen: short):  
    short
```

Возвращаемые результаты

Функция возвращает строку с версий считывателя, заканчивающуюся нулевым байтом. Если выделенный вызывающей программой буфер короче строки версии, то она урезается до значения `maxlen - 1` (один символ резервируется для нулевого байта). Рекомендуемая длина буфера не менее 50 символов.

Получение даты `GetReaderDate`

Функция получает дату из часов реального времени считывателя. При вызове функции передаются указатели на три возвращаемых параметра, назначение которых понятно из названия.

Описание

```
function GetReaderDate(adr: short; var Year, Month, Day: word): short
```

Возвращаемые результаты

Функция возвращает текущие год, месяц и день из встроенных часов реального времени.

Получение времени GetReaderTime

Функция получает время и день недели из часов реального времени считывателя. При вызове функции передаются указатели на четыре возвращаемых параметра, назначение которых понятно из названия.

Описание

```
function GetReaderTime(adr: short; var Hour, Min, Sec, DOW: word): short
```

Возвращаемые результаты

Функция возвращает часы, минуты, секунды и день недели из встроенных часов реального времени.

Установка даты SetReaderDate

Функция предназначена для установки даты в часах реального времени считывателя в соответствии со значениями передаваемых параметров.

Описание

```
function SetReaderDate(adr: short; Year, Month, Day: word): short
```

Возвращаемые результаты

При неправильных значениях даты возможно возвращение значения R245_ERR_WRONG_PARAM.

Установка времени SetReaderTime

Функция предназначена для установки времени и дня недели в часах реального времени считывателя в соответствии со значениями передаваемых параметров.

Описание

```
function SetReaderTime(adr: short; Hour, Min, Sec, DOW: word): short
```

Возвращаемые результаты

При неправильных значениях даты возможно возвращение значения R245_ERR_WRONG_PARAM.

Смена адреса считывателя SetNewReaderAddr

Функция позволяет программно изменить адрес считывателя. Новое значение адреса сохраняется в энергонезависимой памяти считывателя. По умолчанию (на производстве) считывателю присваивается адрес, равный единице.

Если адрес считывателя неизвестен, то команда смены адреса может быть отправлена по broadcast адресу, равному 0xFF.

При программировании адреса с использованием в качестве адресата broadcast адреса на линии должен быть включенным только один считыватель.

Функция доступна в DLL версии не ниже 1.1.2.x.

Описание

```
function SetNewReaderAddr(adr, newaddr: short): short
```

Функция помимо текущего адреса считывателя (adr) получает новое значение адреса (newaddr), которое необходимо присвоить данному считывателю.

Возвращаемые результаты

При неправильных значениях назначаемого адреса (0x00 или 0xFF) возвращается значение R245_ERR_WRONG_PARAM.

Программирование параметров каналов

Данная группа функций предназначена для программирования и получения параметров радиоканалов считывателя – времени памяти тагов в стеках радиоканалов, затухания (чувствительности) радиоканалов, а также для программного включения или выключения каждого из радиоканалов считывателя.

Установка времени памяти тага в радиоканале SetTagMemTime_1

Функция позволяет установить время памяти тагов в стеке первого канала считывателя.

Описание

```
function SetTagMemTime_1(adr: short; num100ms: short): short
```

В качестве параметра функция получает время памяти тага в секундах (aTime) в диапазоне от 10 до 2000 единиц по 100 мсек (то есть от 1 до 200 секунд).

Возвращаемые результаты

R245_RES_OK, R245_ERR_WRONG_PARAM

Установка времени памяти тага в радиоканале SetTagMemTime_2

Функция позволяет установить время памяти тагов в стеке второго канала считывателя.

Описание

```
function SetTagMemTime_2(adr: short; num100ms: short): short
```

В качестве параметра функция получает время памяти тага в секундах (aTime) в диапазоне от 10 до 2000 единиц по 100 мсек (то есть от 1 до 200 секунд).

Возвращаемые результаты

R245_RES_OK, R245_ERR_WRONG_PARAM

Получить время памяти тага GetTagMemTime_1

Функция позволяет получить установленное время памяти тага в первом радиоканале считывателя.

Описание

```
function GetTagMemTime_1(adr: short; var num100ms: short): short
```

Значение установленного времени памяти возвращается в десятых долях секунды (единица соответствует 100 мсек).

Возвращаемые результаты

R245_RES_OK

Получить время памяти тага GetTagMemTime_2

Функция позволяет получить установленное время памяти тага во втором радиоканале считывателя.

Описание

```
function GetTagMemTime_2(adr: short; var num100ms: short): short
```

Значение установленного времени памяти возвращается в десятых долях секунды (единица соответствует 100 мсек).

Возвращаемые результаты

R245_RES_OK

Установка затухания SetDumping_1

Функция предназначена для установки затухания встроенного аттенюатора первого канала считывателя.

Примечание: используемый тип модуляции сигнала тага в значительной мере ослабляет работу аттенюатора, поэтому его влияние на дальность считывания несколько меньше ожидаемого.

Описание

```
function SetDumping_1(adr: short; dump: short): short
```

В качестве параметра функции передается величина затухания от 0 до 31.

Возвращаемые результаты

R245_RES_OK, R245_ERR_WRONG_PARAM

Установка затухания SetDumping_2

Функция предназначена для установки затухания встроенного аттенюатора второго канала считывателя.

Описание

```
function SetDumping_2(adr: short; dump: short): short
```

В качестве параметра функции передается величина затухания от 1 до 31.

Возвращаемые результаты

R245_RES_OK, R245_ERR_WRONG_PARAM

Получить затухание в первом канале GetDumping_1

Функция позволяет получить установленное затухание аттенюатора первого радиоканала считывателя.

Описание

```
function GetDumping_1(adr: short; var dump: short): short
```

Возвращаемые результаты

При успешном завершении функция возвращает установленное затухание аттенюатора первого радиоканала считывателя. Значение может быть от 0 до 31.

Получить затухание во втором канале GetDumping_2

Функция позволяет получить установленное затухание аттенюатора второго радиоканала считывателя.

Описание

```
function GetDumping_2(adr: short; var dump: short): short
```

Возвращаемые результаты

При успешном завершении функция возвращает установленное затухание аттенюатора второго радиоканала считывателя. Значение может быть от 0 до 31.

Установка активности канала SetChannelActivity_1

Функция позволяет устанавливать активность (включать или выключать) для первого радиоканала считывателя.

Описание

```
function SetChannelActivity_1(adr: short; act: short): short
```

Передаваемый параметр (act) может принимать значение «0» - выключено, или «1» - включено.

Возвращаемые результаты

R245_RES_OK, R245_ERR_WRONG_PARAM

Установка активности канала SetChannelActivity_2

Функция позволяет устанавливать активность (включать или выключать) для второго радиоканала считывателя.

Описание

```
function SetChannelActivity_2(adr: short; act: short): short
```

Передаваемый параметр (act) может принимать значение «0» - выключено, или «1» - включено.

Возвращаемые результаты

R245_RES_OK, R245_ERR_WRONG_PARAM

Функции для работы со стеком тагов

Это основная группа функций, позволяющая контролировать таги, находящиеся в зоне чтения считывателя раздельно по двум каналам.

Замечание: количество тагов, возвращаемое функциями *GetNumTags_x*, может не соответствовать реальному к моменту получения последнего номера тага с использованием функции *GetNextTag_x*, поэтому «вычерпывание» тагов следует делать до возвращения функцией *GetNextTag_x* результата *R245_ERR_NO_TAG*.

Получить число тагов *GetNumTags_1*

Функция позволяет получить число тагов, которые на момент запроса находятся в стеке первого радиоканала.

Описание

```
function GetNumTags_1(adr: short; var num: short): short
```

Возвращаемые результаты

При результате *R245_RES_OK* функция возвращает число тагов в первом радиоканале считывателя. Возможные значения – от 0 до 63.

Получить первый таг *GetFirstTag_1*

Функция позиционирует внутренний указатель на первый таг во внутреннем стеке и возвращает его четырехбайтовый серийный номер.

Описание

```
function GetFirstTag_1(adr: short; var serial: DWORD): short
```

Возвращаемые результаты

При результате *R245_RES_OK* функция возвращает серийный номер первого тага из стека. Если тагов в стеке нет, то возвращается код ошибки *R245_ERR_NO_TAG*.

Получить следующий таг *GetNextTag_1*

Функция возвращает серийный номер очередного тага из стека первого радиоканала считывателя.

Описание

```
function GetNextTag_1(adr: short; var serial: DWORD): short
```

Возвращаемые результаты

При результате *R245_RES_OK* функция возвращает серийный номер очередного тага из стека первого радиоканала. Если больше тагов в стеке нет, то возвращается код ошибки *R245_ERR_NO_TAG*.

Получить число тагов GetNumTags_2

Функция позволяет получить число тагов, которые на момент запроса находятся в стеке второго радиоканала.

Описание

```
function GetNumTags_2(adr: short; var num: short): short
```

Возвращаемые результаты

При результате R245_RES_OK функция возвращает число тагов во втором радиоканале считывателя. Возможные значения – от 0 до 63.

Получить первый таг GetFirstTag_2

Функция позиционирует внутренний указатель на первый таг во внутреннем стеке и возвращает его четырехбайтовый серийный номер.

Описание

```
function GetFirstTag_2(adr: short; var serial: DWORD): short
```

Возвращаемые результаты

При результате R245_RES_OK функция возвращает серийный номер первого тага из стека. Если тагов в стеке нет, то возвращается код ошибки R245_ERR_NO_TAG.

Получить следующий таг GetNextTag_2

Функция возвращает серийный номер очередного тага из стека второго радиоканала считывателя.

Описание

```
function GetNextTag_2(adr: short; var serial: DWORD): short
```

Возвращаемые результаты

При результате R245_RES_OK функция возвращает серийный номер очередного тага из стека второго радиоканала. Если больше тагов в стеке нет, то возвращается код ошибки R245_ERR_NO_TAG.

Функции для работы с буфером транзакций

Данная группа функций предназначена для работы с кольцевым буфером транзакций, имеющимся в считывателе.

Буфер транзакций всегда работает в считывателях с режимом мониторинга. В считывателях для систем доступа буфер транзакций активируется только при включении режима аудита.

Буфер транзакций считывателя имеет кольцевую структуру и является энергонезависимым. Емкость буфера равна 1000 транзакций.

Очистка буфера транзакций ClearTransactions

Функция предназначена для очистки кольцевого буфера транзакций считывателя и установки внутренних указателей в исходное положение.

Должна вызываться однократно после первого включения нового считывателя, а также может использоваться для устранения сбоев указателей буфера (например, если питание выключалось в момент записи транзакции в память).

Описание

```
function ClearTransactions(adr: short): short
```

Возвращаемые результаты

При отсутствии ошибок связи функция всегда возвращает результат R245_RES_OK.

Получение числа транзакций в буфере GetNumTransactions

Функция позволяет получить количество транзакций, которые находятся в кольцевом буфере на момент запроса.

Описание

```
function GetNumTransactions(adr: short; var num: short): short
```

Возвращаемые результаты

При отсутствии ошибок связи функция всегда возвращает результат R245_RES_OK.

Получение следующей транзакции GetNextTransaction

Функция используется для получения очередной транзакции из кольцевого буфера. Всегда получается самая старая транзакция. После передачи транзакции считыватель стирает ее из кольцевого буфера и перемещает внутренние указатели на следующую транзакцию.

Описание

```
function GetNextTransaction(adr: short; var code, ch: short; var tag:
    DWORD; var Year, Month, Day, Hour, Min, Sec, DOW: word): short
```

Возвращаемые результаты

Функция при успешном завершении (результат функции равен R245_RES_OK) возвращает код транзакции, номер канала (1 или два для транзакций с тагом, 0 для транзакции, относящиеся к считывателю в целом. Примером транзакции с нулевым значением канала может служить транзакция включения питания).

Далее возвращаются код тага, год, месяц, число, часы, минуты, секунды и день недели транзакции.

Если транзакций в буфере нет, то возвращается код R245_ERR_WRONG_PARAM.

Коды транзакций приведены в Приложении 1 к данному документу.

Включение и выключение аудита AuditOnOff

С версии 1.2.2.x библиотеки добавлена функция AuditOnOff. Функция используется для включения и выключения записи транзакций в буфер транзакция считывателя. По умолчанию в стандартном режиме (не в режиме мониторинга) аудит выключен, и транзакции по событиям считывателя в буфер не заносятся. При включении аудита каждое событие считывателя (обнаружение тага, потеря тага, передача его на ПК через интерфейс wiegand и так далее) заносятся в кольцевой энергонезависимый буфер, из которого могут быть считаны функцией GetNextTransaction.

Описание

```
function AuditOnOff(adr: short; on: short): short
```

Возвращаемые результаты

Функция при успешном обмене со считывателем всегда возвращает результат R245_RES_OK).

Приложение 1.

Коды транзакций считывателя

Ниже в таблице приведены все возможные коды транзакций, возвращаемых считывателем.

Код	Название	Примечание
0x001	Считыватель включен	
0x002	Считыватель выключен	
0x010	Новый таг	
0x011	Таг потерян	
0x020	Сработал вибродатчик	Транзакции порождаются только специальными тагами со встроенными датчиками
0x021	Вибродатчик восстановлен	
0x022	Сработал тампер тага	
0x023	Тампер восстановлен	
0x024	Разряжена батарейка тага	
0x025	Батарейка восстановлена	
0x101	Датчик автомобиля включился	Транзакции генерируются только при наличии датчиков автоматике ворот
0x102	Датчик автомобиля выключился	
0x103	Датчик ворот включился	
0x104	Датчик ворот выключился	
0x121	Таг есть в опозитонном канале	
0x122	Канал заблокирован	
0x123	Канал разблокирован	
0x124	Таг отправлен в контроллер	
0x125	Переход в ожидание автомобиля	
0x126	Автомобиль обнаружен датчиком	
0x127	Автомобиль не появился в течение времени ожидания	
0x128	Ворота не открылись за время ожидания	
0x129	Ожидаем закрывания ворот	
0x12A	Алгоритм проезда завершен	
0x12B	Ожидаем допуска автомобиля	
0x12C	Отработка времени ворот	
0x12D	Датчик ворот закрылся	
0x12E	Время истекло, ворота не закрылись	

Для заметок
