



Версия драйвера:	1.7
Версия документации:	1.7
Дата последнего изменения:	14 мая 2009

«ШТРИХ-М: Драйвер считывателей Mifare»

Руководство программиста

«ШТРИХ-М»
2009

Содержание

Карты MIFARE®.....	5
Сокращения	5
Характеристики считывателей Mifare®	5
Диаграмма состояний считывателей Mifare	5
Работа с памятью карты	6
Блок-значение.....	6
Драйвер	8
Комплект поставки	8
Типы данных	8
Подключение.....	9
1С:Предприятие	9
Delphi.....	9
Методы.....	10
AboutBox ОПрограмме	10
AddField ДобавитьПоле.....	10
AuthStandard АвторизацияСтандартнымКлючом	10
ClearBlock ОчиститьБлок	10
ClearFieldValues ОчиститьЗначенияПолей	10
ClosePort ЗаккрытьПорт	10
Connect УстановитьСоединение	11
DecodeValueBlock Декодировать Блок	11
DeleteAllFields УдалитьВсеПоля	11
DeleteAppSectors УдалитьСектораПриложения	11
DeleteField УдалитьПоле	11
DeleteSector УдалитьСектор	11
Disconnect РазорватьСоединение	11
EncodeKey КодироватьКлюч	12
EncodeValueBlock КодироватьБлок	12
FindDevice ПоискУстройства.....	12
GetFieldParams ПолучитьПараметрыПоля	12
GetSectorParams ПолучитьПараметрыСектора	12
InterfaceSetTimeout УстановитьТаймаут	12
LoadFieldsFromFile ЗагрузитьПоляИзФайла	12
LoadParams ЗагрузитьПараметры.....	13
LoadValue ЗагрузитьЗначение	13
OpenPort ОткрытьПорт	13
PcdBeep Гудок	13
PcdConfig НастроитьСчитыватель	13
PcdGetFwVersion ПолучитьВерсиюПО	13
PcdGetRicVersion ПолучитьВерсиюRIC.....	13
PcdGetSerialNumber ПолучитьСерийныйНомер	14
PcdLoadKeyE2 ЗаписатьКлюч	14
PcdReadE2 СчитатьКлюч16	14
PcdReset СброситьСчитыватель	14
PcdRfReset ОтключитьПоле.....	14
PcdSetDefaultAttrib УстановитьНачальныеПараметры	14
PcdSetTmo УстановитьТаймаутКоманды	14
PcdWriteE2 ЗаписатьКлюч16	15
PiccActivateIdle ВыбратьIdleКарты	15
PiccActivateWakeur ВыбратьВсеКарты	15

PiccAnticoll Антиколлизия	15
PiccAuth Авторизоваться	15
PiccAuthE2 АвторизоватьсяПоНомеру	15
PiccAuthKey АвторизоватьсяПоКлючу	16
PiccCascAnticoll ПоследовательнаяАнтиколлизия	16
PiccCascSelect ПоследовательныйВыбор	16
PiccCommonRead ПрочитатьДанные	16
PiccCommonRequest ЗапроситьКарту	16
PiccCommonWrite ЗаписатьДанные	16
PiccHalt ОстановитьКарту	16
PiccRead ПрочитатьДанные16	17
PiccSelect ВыбратьКартуПоНомеру	17
PiccValue ИзменитьЗначение	17
PiccValueDebit ИзменитьЗначениеДебит	17
PiccWrite ЗаписатьДанные16	17
PortOpened ПортОткрыт	17
ReadDirectory ПрочестьКаталог	17
ReadFields ПрочитатьПоля	18
RequestAll ЗапроситьВсеКарты	18
RequestIdle ЗапроситьIdleКарты	18
ResetCard СброситьКарту	18
SaveFieldsToFile СохранитьПоляВФайле	18
SaveParams СохранитьПараметры	18
SendEvent ПослатьСобытие	18
SetDefaults ПараметрыПоУмолчанию	18
SetFieldParams УстановитьПараметрыПоля	19
SetSectorParams УстановитьПараметрыСектора	19
ShowDirectoryDlg ПоказатьКаталог	19
ShowFirmsDlg ДиалогФирм	19
ShowProperties ПоказатьСтраницуСвойств	19
ShowSearchDlg ДиалогПоиска	19
PollStart НачатьОпрос	19
StartTransTimer ЗапуститьТаймер	19
PollStop ПрерватьОпрос	20
StopTransTimer ОстановитьТаймер	20
TestBit ПроверитьБит	20
WriteDirectory ЗаписатьКаталог	20
WriteFields ЗаписатьПоля	20
Свойства	21
AppCode КодПриложения	21
ATQ ОтветКарты	21
BaudRate Скорость	21
BeepTone НомерЗвука	21
BitCount КоличествоБит	22
BlockAddr АдресБлока	22
BlockData БлокДанных	22
BlockDataHex БлокДанныхHex	22
BlockNumber НомерБлока	22
BlockValue ЗначениеБлока	23
CardDescription ОписаниеКарты	23
CardType ТипКарты	23
Command Команда	23

Connected СоединениеУстановлено	23
DataLength ДлинаДанных	24
DeltaValue Изменение.....	24
DirectoryStatus СостояниеКаталога	24
DirectoryStatusText ОписаниеСостоянияКаталога	24
ErrorText ТекстОшибки	24
ExecutionTime ВремяВыполнения.....	24
FieldCount КоличествоПолей.....	25
FieldIndex ИндексПоля	25
FieldSize РазмерПоля.....	25
FieldType ТипПоля.....	25
FieldValue ЗначениеПоля	25
FileName ИмяФайла.....	26
FirmCode КодФирмы	26
IsClient1С Клиент1С	26
IsShowProperties СтраницаСвойствОткрыта	26
KeyA КлючА.....	26
KeyB КлючВ.....	26
KeyEncoded КодированныйКлюч.....	27
KeyNumber НомерКлюча	27
KeyType ТипКлюча.....	27
KeyUncoded НекодированныйКлюч	27
LibInfoKey ТипИнформацииОБиблиотеке	27
LockDevices БлокироватьЛУ	27
ParentWnd ОкноПриложения.....	28
PasswordHeader ПарольЗаголовка	28
PcdFwVersion ВерсияПО.....	28
PcdRicVersion ВерсияRIC.....	28
PortNumber НомерПорта	28
ReqCode КодЗапроса.....	29
ResultCode Результат	29
ResultDescription ОписаниеРезультата	29
RfResetTime ВремяОтключенияПоля	29
RICValue ЗначениеRIC	29
SAK SAK.....	29
SectorCount КоличествоСекторов.....	30
SectorIndex ИндексСектора.....	30
SectorNumber НомерСектора	30
SelectCode КодВыбора.....	30
Timeout Таймаут.....	30
TransBlockNumber НомерБлокаХранения	31
TransTime ВремяОперации	31
UID UID	31
UIDHex UIDHex	31
UIDLen ДлинаUID	31
ValueOperation ОперацияНадЗначением	32
Version ВерсияФайла.....	32
PollStarted ОпросНачат	32
PollInterval ИнтервалОпроса.....	32
PollAutoDisable АвтоотключениеОпроса	32
UpdateTrailer ОбновитьТрейлер	33
DataAuthMode РежимАвторизации.....	33

DataFormat ФорматДанных.....	33
Работа с каталогом.....	33

Карты MIFARE®

Сокращения

ATQ	Answer to Request	ответ на запрос
ATQA	Answer to Request, type A	ответ на запрос, тип A
PCD	Proximity Coupling Device	считыватель карт
PICC	Proximity Intergrated Circuit Card	карта
REQA	Request command, Type A	команда запроса
SAK	Select Acknowledge, Type A	ответ на команду выбора
SELECT	Select command, Type A	команда выбора
UID	Unique identifier	уникальный идентификатор

Характеристики считывателей Mifare®

Бесконтактная передача данных и энергии для питания карты (не требуются батареи питания)

Дальность работы до 100 мм. (зависит от размеров антенны)

Рабочая частота: 13.56 МГц

Высокая скорость передачи данных – 106 Кбод

Высокая степень защиты данных: 16 битная контрольная сумма, четность, кодирование битов, подсчет битов

Эффективный механизм антиколлизии

Типичное время транзакции: < 100 мс.

EEPROM

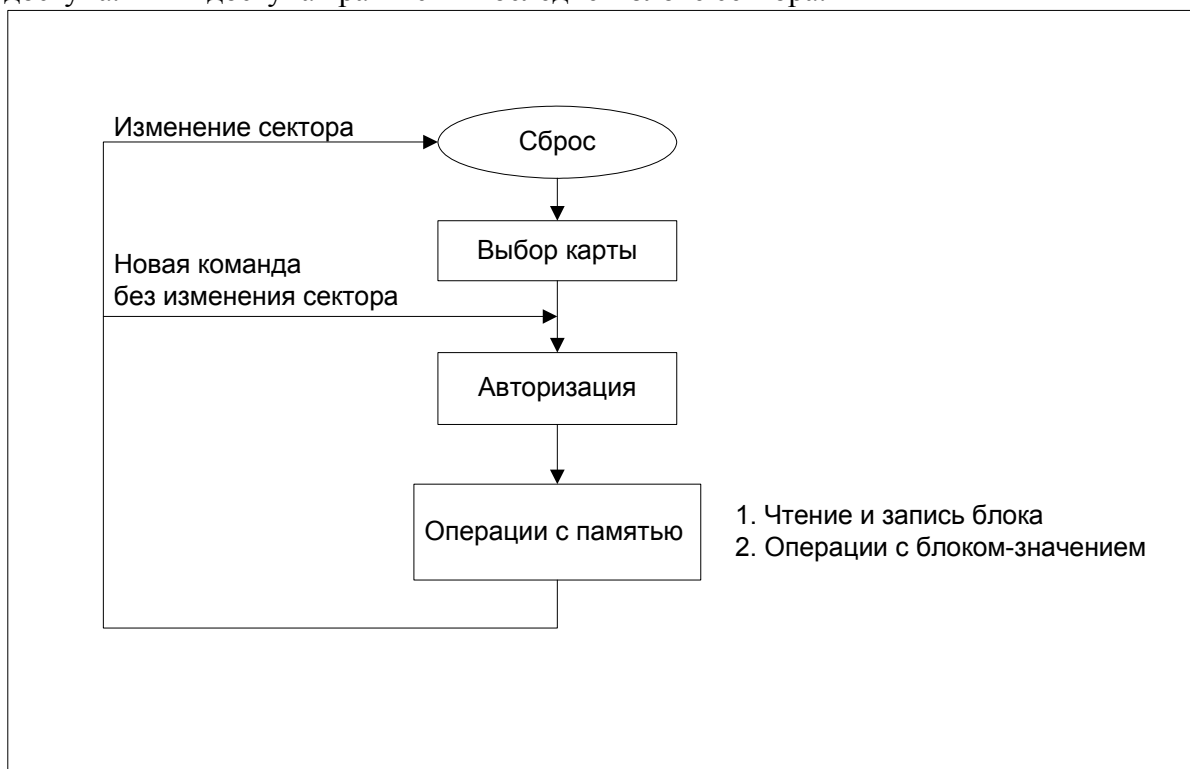
Характеристики карт семейства MIFARE®

	MIFARE® Ultralight	MIFARE® Standard 1K	MIFARE® Standard 4K
EEPROM	512 бит	1 КБ	4КБ
Шифрование	-	MIFARE® CRYPTO	MIFARE® CRYPTO

Диаграмма состояний считывателей Mifare

Работа с памятью карты

Перед работой с памятью нужно выбрать карту и авторизоваться к нужному блоку. Операции, которые можно производить с блоком определяются типом ключа доступа (А или В) и битами доступа. Биты доступа хранятся в последнем блоке сектора.



Операции с памятью

Операция	Описание	Тип блока
Чтение	Чтение блока	Все блоки
Запись	Запись блока	Все блоки
Инкремент	Увеличение значения блока	Блок-значение
Декремент	Уменьшение значения блока	Блок-значение
Загрузка	Чтение значения блока	Блок-значение
Сохранение	Запись значения блока	Блок-значение

Блок-значение

Блок-значение - специально сформированный блок карты, с которым можно производить операции увеличения, уменьшения, загрузки и восстановления. Блок-значение имеет фиксированный формат, позволяющий обнаруживать и исправлять ошибки, а также производить сохранение.

Номер байта	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Описание	Значение				Значение				Значение				Адр	$\overline{\text{Адр}}$	Адр	$\overline{\text{Адр}}$

Значение: Знаковое число длиной 4 байта. Наименее значащий байт числа хранится в байте с меньшим адресом. Отрицательные числа хранятся в стандартном дополнительном коде. Для обеспечения целостности данных, значение хранится трижды - два раза неинвертированное и один раз инвертированное.

Адр.: Адрес блока, используемого для сохранения. Адрес блока не изменяется при операциях увеличения, уменьшения, загрузки и восстановления. Значение адреса может быть изменено только командой записи.

Драйвер

Комплект поставки

MifareDrv.dll - драйвер для операционных систем Windows.

MifareTst.exe - тест драйвера.

Примеры использования для программ «1С:Предприятие» версий 7.5, 7.7 и 8.0.

Примеры использования драйвера для Borland Delphi 5.0.

Типы данных

В данном документе для описания данных используются типы:

<i>Тип</i>	<i>Русское</i>	<i>Длина</i>	<i>Диапазон значений</i>
Integer	Целое	4 байта со знаком	-2147483648 до 2147483647
Currency	Денежный	8 байт	-922337203685477,5808 до 922337203685477,5807
Double	Дробное	8 байт	5,0 x 10 ⁻³²⁴ до 1,7 x 10 ⁺³⁰⁸
WideString	Строка	неограниченная	
WordBool	Логическое	4 байта	0 ..1
DateTime	ДатаВремя	4 байта	В Windows тип DATETIME

Подключение

Имя объекта драйвера: **AddIn.MifareDrv**

1С:Предприятие

Глобальный модуль:

```
Перем Драйвер Экспорт;  
//Загрузка внешней компоненты  
Если ЗагрузитьВнешнююКомпоненту(«MifareDrv.dll») <> 0 Тогда  
    Драйвер = СоздатьОбъект(«AddIn.MifareDrv»);  
Иначе  
    Сообщить(«Не найдена внешняя компонента.»);  
КонецЕсли;  
// После этого возможен доступ к свойствам и методам драйвера  
// во всех модулях, например:  
Если Драйвер.Беep() < 0 Тогда  
    // Обращение к свойству  
    Сообщить(«Ошибка:» + Строка(Драйвер.ResultDescription));  
КонецЕсли;
```

Delphi

Через OleControl

```
uses MifareLib_TLB;  
var  
    Driver: TMifareDrv;  
begin  
    Driver := TMifareDrv.Create(nil);  
    Driver.Beep;  
end;
```

Через интерфейс IMifareDrv

```
uses MifareLib_TLB;  
var  
    Driver: IMifareDrv;  
begin  
    Driver := CreateOleObject('Addin.MifareDrv') as IMifareDrv;  
    Driver.Beep;  
end;
```

Через OleVariant

Используется позднее связывание. Если программист допускает ошибку в названии метода или свойства, то узнает он об этом только в момент выполнения.

```
var  
    Driver: OleVariant;  
begin  
    Driver := CreateOleObject('Addin.MifareDrv');  
    Driver.Beep;  
end;
```

Методы

Все методы драйвера являются функциями и возвращают код ошибки (кроме метода [AboutBox](#)). Этот же код ошибки записывается в свойство [ResultCode](#). Приложение может проверить результат вызова метода или свойство [ResultCode](#).

AboutBox

ОПрограмме

procedure AboutBox;

Выводит на экран окно «О программе». Стандартный метод для элементов управления ActiveX.

Используемые свойства: -

Изменяемые свойства: -

AddField

ДобавитьПоле

function AddField: Integer;

Добавляет поле с параметрами [FieldSize](#), [FieldType](#) и [FieldValue](#)

Используемые свойства: [FieldSize](#), [FieldValue](#), [FieldType](#)

Изменяемые свойства: -

AuthStandard

АвторизацияСтандартнымКлючом

function AuthStandard: Integer;

Авторизация стандартным ключом к блоку с номером [BlockNumber](#).

Используемые свойства: [BlockNumber](#)

Изменяемые свойства: [KeyEncoded](#), [KeyType](#), [KeyNumber](#)

ClearBlock

ОчиститьБлок

function ClearBlock: Integer;

Метод очищает блок данных с номером [BlockNumber](#). Перед вызовом метода необходимо авторизоваться к блоку с номером [BlockNumber](#).

Используемые свойства: [BlockNumber](#)

Изменяемые свойства: -

ClearFieldValues

ОчиститьЗначенияПолей

function ClearFieldValues: Integer;

Метод очищает все поля.

Используемые свойства: -

Изменяемые свойства: -

ClosePort

ЗакрытьПорт

function ClosePort: Integer;

Закрывает COM порт.

Используемые свойства: -

Изменяемые свойства: -

Connect

УстановитьСоединение

function Connect: Integer;

Метод открывает COM порт и выполняет инициализацию считывателя.

Используемые свойства: -

Изменяемые свойства: -

DecodeValueBlock

Декодировать Блок

function DecodeValueBlock: Integer;

По значению [BlockData](#) формирует [BlockValue](#), [BlockAddr](#).

Используемые свойства: [BlockData](#)

Изменяемые свойства: [BlockValue](#), [BlockAddr](#)

DeleteAllFields

УдалитьВсеПоля

function DeleteAllFields: Integer;

Метод удаляет все поля.

Используемые свойства: -

Изменяемые свойства: -

DeleteAppSectors

УдалитьСектораПриложения

function DeleteAppSectors: Integer;

Метод удаляет сектора приложения с кодом [AppCode](#) и кодом фирмы [FirmCode](#).

Используемые свойства: [AppCode](#), [FirmCode](#)

Изменяемые свойства: -

DeleteField

УдалитьПоле

function DeleteField: Integer;

Метод удаляет поле с индексом [FieldIndex](#)

Используемые свойства: [FieldIndex](#)

Изменяемые свойства: -

DeleteSector

УдалитьСектор

function DeleteSector: Integer;

Удаление сектора с номером [SectorNumber](#). Перед удалением сектора осуществляется авторизация.

Используемые свойства: [SectorNumber](#), [KeyB](#)

Изменяемые свойства: -

Disconnect

РазорватьСоединение

function Disconnect: Integer;

Метод закрывает COM порт.

Используемые свойства: -

Изменяемые свойства: -

EncodeKey

КодироватьКлюч

function EncodeKey: Integer;

Кодирует 6-байтовый ключ в 12-байтовый кодированный формат, более устойчивый к искажениям.

Используемые свойства: [KeyUncoded](#)

Изменяемые свойства: [KeyEncoded](#)

EncodeValueBlock

КодироватьБлок

function EncodeValueBlock: Integer;

Формирует блок [BlockData](#) из полученных данных [BlockValue](#), [BlockAddr](#).

Используемые свойства: [BlockValue](#), [BlockAddr](#)

Изменяемые свойства: [BlockData](#)

FindDevice

ПоискУстройства

function FindDevice: Integer;

Метод осуществляет поиск устройства по всем COM портам. Если устройство найдено, то в свойстве [PortNumber](#) будет номер порта, на котором оно найдено.

Используемые свойства: -

Изменяемые свойства: [PcdFWVersion](#), [PortNumber](#)

GetFieldParams

ПолучитьПараметрыПоля

function GetFieldParams: Integer;

Метод получает параметры поля с индексом [FieldIndex](#).

Используемые свойства: [FieldIndex](#)

Изменяемые свойства: [FieldSize](#), [FieldValue](#), [FieldType](#)

GetSectorParams

ПолучитьПараметрыСектора

function GetSectorParams: Integer;

Метод получает параметры сектора с индексом SectorIndex.

Используемые свойства: [SectorIndex](#)

Изменяемые свойства: [SectorNumber](#), [AppCode](#), [FirmCode](#)

InterfaceSetTimeout

УстановитьТаймаут

function InterfaceSetTimeout: Integer;

Устанавливает таймаут работы с COM портом. Значение таймаута хранится только в течение сеанса работы, т.е. если вы хотите использовать свое значение, данную функцию надо вызывать каждый раз перед началом работы с библиотекой.

Используемые свойства: [Timeout](#)

Изменяемые свойства: -

LoadFieldsFromFile

ЗагрузитьПоляИзФайла

function LoadFieldsFromFile: Integer;

Метод загружает поля из файла с именем [FileName](#).

Используемые свойства: [FileName](#)

Изменяемые свойства: -

LoadParams

ЗагрузитьПараметры

function LoadParams: Integer;

Метод выполняет чтение параметров драйвера из реестра. Параметры драйвера - номер COM порта. Чтение выполняется автоматически при загрузке драйвера.

Используемые свойства: -

Изменяемые свойства: -

LoadValue

ЗагрузитьЗначение

function LoadValue: Integer;

Метод выполняет чтение блока-значения.

Используемые свойства: [BlockNumber](#)

Изменяемые свойства: [BlockData](#), [BlockValue](#)

OpenPort

ОткрытьПорт

function OpenPort: Integer;

Открывает COM порт с номером [PortNumber](#).

Используемые свойства: [PortNumber](#)

Изменяемые свойства: -

PcdBeep

Гудок

function PcdBeep: Integer;

Издает звуковой сигнал с номером [BeepTone](#).

Используемые свойства: [BeepTone](#)

Изменяемые свойства: -

PcdConfig

НастроитьСчитыватель

function PcdConfig: Integer;

Загружает в считыватель базовые настройки. В общем случае в вызове этой функции нет необходимости. Настройки загружаются автоматически при включении питания считывателя.

Используемые свойства: -

Изменяемые свойства: -

PcdGetFwVersion

ПолучитьВерсиюПО

function PcdGetFwVersion: Integer;

Получает версию ПО считывателя.

Используемые свойства: -

Изменяемые свойства: [PcdFWVersion](#)

PcdGetRicVersion

ПолучитьВерсиюRIC

function PcdGetRicVersion: Integer;

Возвращает версию RIC считывателя.

Используемые свойства: -
Изменяемые свойства: [PcdRicVersion](#)

PcdGetSerialNumber **ПолучитьСерийныйНомер**

function PcdGetSerialNumber: Integer;
Возвращает серийный номер считывателя.

Используемые свойства: -
Изменяемые свойства: [UID](#)

PcdLoadKeyE2 **ЗаписатьКлюч**

function PcdLoadKeyE2: Integer;
Записывает ключ в EEPROM MfRC500

Используемые свойства: [KeyType](#), [KeyNumber](#), [KeyUncoded](#)
Изменяемые свойства: -

PcdReadE2 **СчитатьКлюч16**

function PcdReadE2: Integer;
Считывает ключ из энергонезависимой памяти считывателя из блока с номером [BlockNumber](#).

Используемые свойства: [DataLength](#), [BlockNumber](#)
Изменяемые свойства: [BlockData](#)

PcdReset **СброситьСчитыватель**

function PcdReset: Integer;
Выполняет перезапуск считывателя

Используемые свойства: -
Изменяемые свойства: -

PcdRfReset **ОтключитьПоле**

function PcdRfReset: Integer;
Отключает радиополе считывателя на время [RfResetTime](#). При отключении поля пропадает питание карточек.

Используемые свойства: [RfResetTime](#)
Изменяемые свойства: -

PcdSetDefaultAttrib **УстановитьНачальныеПараметры**

function PcdSetDefaultAttrib: Integer;
Устанавливает значение скорости считывателя в 106 КБод. Надобности в вызове этого метода нет, он вызывается автоматически при вызове других методов.

Используемые свойства: -
Изменяемые свойства: -

PcdSetTmo **УстановитьТаймаутКоманды**

function PcdSetTmo: Integer;
Устанавливает таймаут выполнения команды.

Используемые свойства: -
Изменяемые свойства: -

PcdWriteE2

ЗаписатьКлюч16

function PcdWriteE2: Integer;

Записывает ключ в энергонезависимую память считывателя в блок с номером [BlockNumber](#).

Используемые свойства: [BlockNumber](#)
Изменяемые свойства: -

PiccActivateldle

ВыбратьIdleКарты

function PiccActivateIdle: Integer;

Выполняет запрос, антиколлизию и выбор для карт в режиме ожидания. Циклический вызов этого метода позволяет получить номера всех карт в поле.

Используемые свойства: -
Изменяемые свойства: [UID](#), [SAK](#), [ATQ](#)

PiccActivateWakeup

ВыбратьВсеКарты

function PiccActivateWakeup: Integer;

Выполняет запрос, антиколлизию и выбор для всех карт. Циклический вызов этого метода позволяет получить номера всех карт в поле.

Используемые свойства: -
Изменяемые свойства: [UID](#), [SAK](#), [ATQ](#)

PiccAnticoll

Антиколлизия

function PiccAnticoll: Integer;

Команда антиколлизии. Если количество бит не равно 0, то карта ищется по фрагменту номера. При успешном выполнении возвращается номер карты полностью.

Используемые свойства: [BitCount](#), [UID](#)
Изменяемые свойства: [UID](#)

PiccAuth

Авторизоваться

function PiccAuth: Integer;

Авторизация к блоку с номером [BlockNumber](#) по ключу [KeyType](#), который записан по адресу [KeyNumber](#).

Используемые свойства: [KeyType](#), [KeyNumber](#), [BlockNumber](#)
Изменяемые свойства: -

PiccAuthE2

АвторизоватьсяПоНомеру

function PiccAuthE2: Integer;

Авторизация к блоку с номером [BlockNumber](#) по ключу [KeyType](#), который записан по адресу [KeyNumber](#).

Используемые свойства: [KeyType](#), [KeyNumber](#), [BlockNumber](#)
Изменяемые свойства: -

PiccAuthKey

Авторизоваться По Ключу

function PiccAuthKey: Integer;

Авторизация к блоку с номером [BlockNumber](#) по ключу [KeyType](#), который записан по адресу [KeyNumber](#).

Используемые свойства: [KeyType](#), [KeyNumber](#), [BlockNumber](#)

Изменяемые свойства: -

PiccCascAnticoll

Последовательная Анतिकоллизия

Метод последовательной антиколлизии существует для соответствия стандарту ISO/IEC 14443. При работе с Mifare Classic (1K и 4K) нет необходимости пользоваться этим методом – можно вызывать метод [PiccAnticoll](#).

Используемые свойства: [SelectCode](#), [BitCount](#)

Изменяемые свойства: -

PiccCascSelect

Последовательный Выбор

Метод последовательного выбора существует для соответствия стандарту ISO/IEC 14443. При работе с Mifare Classic (1K и 4K) нет необходимости пользоваться этим методом – можно вызывать метод [PiccSelect](#).

Используемые свойства: [SelectCode](#), [BitCount](#)

Изменяемые свойства: -

PiccCommonRead

Прочитать Данные

Метод читает блок произвольной длины. Для карточек MIFARE® Classic длина блока должна быть равна 16. При работе с карточками MIFARE® Classic лучше использовать метод [PiccRead](#).

Используемые свойства: [Command](#), [BlockNumber](#), [DataLength](#)

Изменяемые свойства: [BlockData](#)

PiccCommonRequest

Запросить Карту

Запрашивает карты в радиополе. Команда для соответствия ISO, аналогична команде [PiccRequest](#).

Используемые свойства: [ReqCode](#)

Изменяемые свойства: [ATQ](#), [CardType](#), [CardDescription](#)

PiccCommonWrite

Записать Данные

Команда записывает на карту блок произвольной длины. Для MIFARE® Classic размер блока должен быть равен 16 байт. Для работы с картами MIFARE® Classic лучше использовать метод [PiccWrite](#).

Используемые свойства: [Command](#), [BlockNumber](#), [DataLength](#), [BlockData](#)

Изменяемые свойства: -

PiccHalt

Остановить Карту

Команда останавливает текущую карту. Карта переходит в состояние Halt и перестает откликаться на запросы [Request Idle](#). Просыпается карта только по запросу [Request All](#). Диаграмма состояний для карты MIFARE® UltraLight приведена на рисунке 1.

Используемые свойства: [Command](#), [BlockNumber](#), [DataLength](#), [BlockData](#)

Изменяемые свойства: -

PiccRead

ПрочитатьДанные16

Команда читает 16-байтовый блок данных с номером [BlockNumber](#). Перед вызовом необходимо авторизоваться к этому блоку.

Используемые свойства: [BlockNumber](#)

Изменяемые свойства: [BlockData](#)

PiccSelect

ВыбратьКартуПоНомеру

Выбор карты с номером [UID](#). Возвращает в [SAK](#) тип карты.

Используемые свойства: [UID](#)

Изменяемые свойства: [SAK](#), [CardType](#), [CardDescription](#)

PiccValue

ИзменитьЗначение

Метод выполняет операцию над блоком-значением с номером [BlockNumber](#) и записывает результат операции в блок с номером [TransBlockNumber](#). Тип операции задается свойством [ValueOperation](#). Допустимые значения свойства [ValueOperation](#):

voIncrement, voDecrement, voRestore, voTransfer

Используемые свойства: [ValueOperation](#), [BlockNumber](#), [TransBlockNumber](#), [DeltaValue](#)

Изменяемые свойства: [BlockValue](#)

PiccValueDebit

ИзменитьЗначениеДебит

Аналогичен методу PiccValue , но для карт с автоматической записью результата операции.

Используемые свойства: [ValueOperation](#), [BlockNumber](#), [DeltaValue](#)

Изменяемые свойства: [BlockValue](#)

PiccWrite

ЗаписатьДанные16

Записывает на карту 16-байтовый блок данных с номером [BlockNumber](#). Перед вызовом метода необходимо авторизоваться к блоку с номером [BlockNumber](#).

Используемые свойства: [BlockNumber](#), [BlockData](#)

Изменяемые свойства: -

PortOpened

ПортОткрыт

Если порт открыт, то функция возвращает 0.

Используемые свойства: -

Изменяемые свойства: -

ReadDirectory

ПрочестьКаталог

Читает каталог карты.

Используемые свойства: -

Изменяемые свойства: -

ReadFields

ПрочитатьПоля

Читает поля для приложения с кодом [AppCode](#) и кодом фирмы [FirmCode](#).

Используемые свойства: [AppCode](#), [FirmCode](#)

Изменяемые свойства: -

RequestAll

ЗапроситьВсеКарты

Запрашивает все карты в радиополе. Карта возвращает в ответе [ATQ](#).

Используемые свойства: -

Изменяемые свойства: [ATQ](#), [CardType](#), [CardDescription](#)

RequestIdle

ЗапроситьIdleКарты

Запрашивает карты в радиополе, которые находятся в состоянии ожидания Idle. Карта возвращает в ответе [ATQ](#).

Используемые свойства: -

Изменяемые свойства: [ATQ](#), [CardType](#), [CardDescription](#)

ResetCard

СброситьКарту

Перезапуск карты.

Используемые свойства: -

Изменяемые свойства: [UID](#), [SAK](#), [ATQ](#)

SaveFieldsToFile

СохранитьПоляВФайле

Метод сохраняет поля в файле с именем [FileName](#).

Используемые свойства: [FileName](#)

Изменяемые свойства: -

SaveParams

СохранитьПараметры

Сохранение параметров драйвера в реестре.

Используемые свойства: [PortNumber](#)

Изменяемые свойства: -

SendEvent

ПослатьСобытие

Посылает событие CardFound.

Используемые свойства: -

Изменяемые свойства: -

SetDefaults

ПараметрыПоУмолчанию

Установка параметров драйвера в начальные значения. Метод присваивает [PortNumber](#) = 1.

Используемые свойства: -

Изменяемые свойства: [PortNumber](#)

SetFieldParams

УстановитьПараметрыПоля

Метод устанавливает параметры поля с индексом [FieldIndex](#).

Используемые свойства: [FieldIndex](#), [FieldSize](#), [FieldValue](#), [FieldType](#)

Изменяемые свойства: -

SetSectorParams

УстановитьПараметрыСектора

Метод устанавливает параметры сектора с индексом [SectorIndex](#).

Используемые свойства: [SectorIndex](#), [AppCode](#), [FirmCode](#)

Изменяемые свойства: [SectorNumber](#)

ShowDirectoryDlg

ПоказатьКаталог

Выводит на экран страницу каталога.

Используемые свойства: -

Изменяемые свойства: -

ShowFirmsDlg

ДиалогФирм

Выводит на экран диалог фирм.

Используемые свойства: -

Изменяемые свойства: -

ShowProperties

ПоказатьСтраницуСвойств

Выводит на экран страницу свойств драйвера.

Используемые свойства: -

Изменяемые свойства: -

ShowSearchDlg

ДиалогПоиска

Выводит на экран диалог поиска считывателя.

Используемые свойства: -

Изменяемые свойства: -

PollStart

НачатьОпрос

Начинает опрос с интервалом [PollInterval](#)

Используемые свойства: [PollInterval](#)

Изменяемые свойства: -

StartTransTimer

ЗапуститьТаймер

Метод запоминает текущее значение системного таймера в свойстве [TransTime](#) в миллисекундах.

Используемые свойства: -

Изменяемые свойства: [TransTime](#)

PollStop

ПрерватьОпрос

Останавливает опрос

Используемые свойства: -

Изменяемые свойства: -

StopTransTimer

ОстановитьТаймер

Метод вычисляет время в миллисекундах, прошедшее с момента последнего вызова метода StartTransTimer и запоминает его в свойстве [TransTime](#).

Используемые свойства: -

Изменяемые свойства: [TransTime](#)

TestBit

ПроверитьБит

Метод проверяет бит параметра AValue с номером ABitIndex на равенство 1. Если бит равен 1, то возвращается TRUE, иначе – FALSE.

Входные параметры:

AValue

Тип: Integer / Целое

Допустимые значения: 0..0xFFFFFFFF

ABitIndex

Тип: Integer / Целое

Допустимые значения: 0..31

Используемые свойства: -

Изменяемые свойства: TransTime

WriteDirectory

ЗаписатьКаталог

Записывает каталог карты.

Используемые свойства: -

Изменяемые свойства: -

WriteFields

ЗаписатьПоля

Записывает поля для приложения с кодом [AppCode](#) и кодом фирмы [FirmCode](#).

Используемые свойства: [AppCode](#), [FirmCode](#)

Изменяемые свойства: -

Свойства

AppCode

КодПриложения

Код приложения.

Тип:	Integer / Целое
Размер:	2 байта
Допустимые значения:	0..65535
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись
Используется методами:	DeleteAppSectors , ReadFields , SetSectorParams , WriteFields
Изменяется методами:	GetSectorParams

ATQ

ОтветКарты

Ответ карты на команду запроса. По ответу можно определить тип карты. Возможные ответы для считывателей Mifare описаны в документе "Type Identification Procedure" (1).

Тип:	Integer / Целое
Размер:	2 байта
Допустимые значения:	0..65535
Значение по умолчанию:	0
Доступ:	R, только чтение
Используется методами:	-
Изменяется методами:	RequestAll , RequestIdle , PiccCommonRequest , PiccActivateIdle , PiccActivateWakeup

BaudRate

Скорость

Скорость обмена данными с картой. Карты MIFARE поддерживают только одну скорость: 106 Кбит/с.

Тип:	Integer / Целое
Размер:	2 байта
Допустимые значения:	0 106 Кбит/с 1 212 Кбит/с 2 424 Кбит/с 3 848 Кбит/с
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись
Используется методами:	PiccActivateIdle , PiccActivateWakeup
Изменяется методами:	-

BeepTone

НомерЗвука

Номер звукового сигнала, издаваемого считывателем при вызове метода [PcdBeep](#).

Тип:	Integer / Целое
Допустимые значения:	0..4
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись
Используется методами:	PcdBeep
Изменяется методами:	-

BitCount

КоличествоБит

Количество известных битов серийного номера карты. Используется для команды антиколлизии.

Тип:	Integer / Целое
Допустимые значения:	0..0xFFFFFFFF
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись
Используется методами:	PiccAnticoll , PiccCascAnticoll
Изменяется методами:	-

BlockAddr

АдресБлока

Адрес блока. Используется для формирования блока-значения.

Тип:	Integer / Целое
Допустимые значения:	0..0xFFFFFFFF
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись
Используется методами:	EncodeValueBlock
Изменяется методами:	DecodeValueBlock

BlockData

БлокДанных

Данные блока. Двоичные данные. В строке может содержаться символ 0.

Тип:	WideString / Строка
Значение по умолчанию:	""
Доступ:	R/W, чтение и запись
Используется методами:	PiccCommonWrite , PiccWrite , PcdWriteE2 , DecodeValueBlock
Изменяется методами:	PiccRead , PiccCommonRead , PcdReadE2 , EncodeValueBlock

BlockDataHex

БлокДанныхHex

Данные блока в шестнадцатеричном формате. При записи свойства изменяется значение свойства [BlockData](#).

Тип:	WideString / Строка
Значение по умолчанию:	""
Доступ:	R/W, чтение и запись
Используется методами:	PiccCommonWrite , PiccWrite , PcdWriteE2 , DecodeValueBlock
Изменяется методами:	PiccRead , PiccCommonRead , PcdReadE2 , EncodeValueBlock

BlockNumber

НомерБлока

Номер блока. Нумерация блоков карты начинается с 0.

Тип:	Integer / Целое
Допустимые значения:	0..0xFFFFFFFF
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись

Используется методами: [PiccAuth](#), [PiccAuthE2](#), [PiccAuthKey](#), [PiccCommonWrite](#), [PiccWrite](#), [PiccRead](#), [PiccCommonRead](#), [PcdReadE2](#), [PcdWriteE2](#)

Изменяется методами: -

BlockValue

ЗначениеБлока

Значение блока-значения. Целое число со знаком размером 4 байта.

Тип: Integer / Целое
Значение по умолчанию: 0
Доступ: R/W, чтение и запись
Используется методами: [EncodeValueBlock](#),
Изменяется методами: [DecodeValueBlock](#), [PiccValue](#)

CardDescription

ОписаниеКарты

Описание карты. Формируется по [ATQ](#) или [SAK](#).

Тип: WideString / Строка
Доступ: R, только чтение
Используется методами: -
Изменяется методами: [PiccCommonRequest](#), [RequestAll](#), [RequestIdle](#), [PiccCaseSelect](#), [PiccSelect](#), [PiccActivateWakeup](#), [PiccActivateIdle](#)

CardType

ТипКарты

Тип карты. Формируется по [ATQ](#) или [SAK](#).

Тип: Integer / Целое
Доступ: R, только чтение
Возможные значения: ctUnknown, ctMifareUltraLight, ctMifare1K, ctMifare4K, ctMifareDESfire, ctMifareProximity, ctSmartMX
Используется методами: -
Изменяется методами: [PiccCommonRequest](#), [RequestAll](#), [RequestIdle](#), [PiccCaseSelect](#), [PiccSelect](#), [PiccActivateWakeup](#), [PiccActivateIdle](#)

Command

Команда

Тип: Integer/ Целое
Доступ: R, только чтение
Возможные значения: dcRead16 = 0x30h
dcWrite4 = 0xA2h
dcWrite16 = 0xA0h
Используется методами: -
Изменяется методами: [Connect](#), [Disconnect](#), [OpenPort](#), [ClosePort](#)

Connected

СоединениеУстановлено

Если значение свойства True - COM порт открыт, False - закрыт.

Тип: WordBool / Логическое
Доступ: R, только чтение
Используется методами: -
Изменяется методами: [Connect](#), [Disconnect](#), [OpenPort](#), [ClosePort](#)

DataLength

ДлинаДанных

Длина записываемых или читаемых данных в байтах.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Используется методами:	PiccCommonRead , PiccCommonWrite , PcdReadE2 , PcdWriteE2
Изменяется методами:	-

DeltaValue

Изменение

Величина изменения блока-значения.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Используется методами:	PiccValue
Изменяется методами:	-

DirectoryStatus

СостояниеКаталога

Состояние каталога.

Тип:	Integer / Целое
Доступ:	R, только чтение
Возможные значения:	dsOK = 0
	DsNotFound = 1
	DsCorrupted = 2
Используется методами:	-
Изменяется методами:	ReadDirectory

DirectoryStatusText

ОписаниеСостоянияКаталога

Описание состояния каталога.

Тип:	WideString / Строка
Доступ:	R, только чтение
Используется методами:	-
Изменяется методами:	ReadDirectory

ErrorText

ТекстОшибки

Текст сообщения об ошибке.

Тип:	WideString / Строка
Доступ:	R, только чтение
Используется методами:	-
Изменяется методами:	всеми

ExecutionTime

ВремяВыполнения

Время выполнения метода.

Тип:	Integer / Целое
Доступ:	R, только чтение
Используется методами:	-

Изменяется методами: всеми

FieldCount КоличествоПолей

Количество полей.

Тип: Integer / Целое
Доступ: R, только чтение
Используется методами: -
Изменяется методами: [AddField](#), [DeleteAllFields](#), [DeleteField](#), [LoadFieldsFromFile](#)

FieldIndex ИндексПоля

Индекс поля.

Тип: Integer / Целое
Доступ: R/W, чтение/запись
Используется методами: [DeleteField](#), [GetFieldParams](#), [SetFieldParams](#)
Изменяется методами: -

FieldSize РазмерПоля

Размер поля в байтах.

Тип: Integer / Целое
Доступ: R/W, чтение/запись
Используется методами: [AddField](#), [SetFieldParams](#)
Изменяется методами: [GetFieldParams](#)

FieldType ТипПоля

Тип поля.

Тип: Integer / Целое
Доступ: R/W, чтение/запись
Возможные значения:
ftByte = 0
ftSmallint = 1
ftBool = 2
ftInteger = 3
ftDouble = 4
ftString = 5
Используется методами: [AddField](#), [SetFieldParams](#)
Изменяется методами: [GetFieldParams](#)

FieldValue ЗначениеПоля

Значение поля.

Тип: WideString / Строка
Доступ: R/W, чтение/запись
Используется методами: [AddField](#), [SetFieldParams](#)
Изменяется методами: [GetFieldParams](#)

FileName ИмяФайла

Имя файла полей.

Тип:	WideString / Строка
Доступ:	R/W, чтение/запись
Используется методами:	LoadFieldsFromFile , SaveFieldsToFile
Изменяется методами:	-

FirmCode КодФирмы

Код фирмы.

Тип:	Integer / Целое
Размер:	2 байта
Допустимые значения:	0..65535
Значение по умолчанию:	0
Доступ:	R/W, чтение и запись
Используется методами:	DeleteAppSectors , ReadFields , SetSectorParams , WriteFields
Изменяется методами:	GetSectorParams

IsClient1C Клиент1С

Драйвер загружен в программу 1С. Служит для внутренних целей.

Тип:	WordBool / Логическое
Доступ:	R, только чтение
Используется методами:	-
Изменяется методами:	-

IsShowProperties СтраницаСвойствОткрыта

Вызван метод ShowProperties - открыта страница свойств драйвера. Служит для внутренних целей.

Тип:	WordBool / Логическое
Доступ:	R, только чтение
Используется методами:	-
Изменяется методами:	-

KeyA КлючА

Ключ для авторизации типа А

Тип:	WideString / Строка
Доступ:	R/W, чтение/запись
Используется методами:	ReadFields , WriteFields
Изменяется методами:	-

KeyB КлючВ

Ключ для авторизации типа В

Тип:	WideString / Строка
------	---------------------

Доступ:	R/W, чтение/запись
Используется методами:	ReadFields , WriteFields
Изменяется методами:	DeleteSector

KeyEncoded

КодированныйКлюч

12 байтовый кодированный ключ в формате Hex.

Тип:	WideString / Строка
Доступ:	R/W, чтение/запись
Используется методами:	PiccAuthKey
Изменяется методами:	EncodeKey

KeyNumber

НомерКлюча

Номер сектора в памяти (0..15), куда записывается ключ.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Допустимые значения:	0..15
Используется методами:	PiccAuth , PiccAuthE2 , PcdLoadKeyE2 ,
Изменяется методами:	-

KeyType

ТипКлюча

Тип ключа, А или В. True - ключ А, False - ключ В.

Тип:	WordBool / Логическое
Доступ:	R/W, чтение/запись
Используется методами:	PiccAuth , PiccAuthE2 , PcdLoadKeyE2 , PiccAuthKey
Изменяется методами:	-

KeyUncoded

НекодированныйКлюч

6 байтовый некодированный ключ в формате Hex.

Тип:	WideString / Строка
Доступ:	R/W, чтение/запись
Используется методами:	PcdLoadKeyE2 , EncodeKey
Изменяется методами:	-

LibInfoKey

ТипИнформацииОБиблиотеке

Параметр, передаваемый в метод [LibInfo](#). Не используется.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Используется методами:	LibInfo
Изменяется методами:	-

LockDevices

БлокироватьЛУ

Блокировка логических устройств. Предназначено для запрещения изменения логических устройств на странице свойств.

Тип:	WordBool / Логическое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	False
Используется методами:	ShowProperties
Изменяется методами:	-

ParentWnd ОкноПриложения

Дескриптор родительского окна.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Используется методами:	ShowSearchDlg
Изменяется методами:	-

PasswordHeader ПарольЗаголовка

Пароль заголовка..

Тип:	WideString / Строка
Доступ:	R, только чтение
Используется методами:	DeleteSector
Изменяется методами:	-

PcdFwVersion ВерсияПО

Версия ПО считывателя.

Тип:	WideString / Строка
Доступ:	R, только чтение
Значение по умолчанию:	-
Используется методами:	-
Изменяется методами:	PcdGetFwVersion

PcdRicVersion ВерсияRIC

Версия микросхемы RIC.

Тип:	WideString / Строка
Доступ:	R, только чтение
Значение по умолчанию:	-
Используется методами:	-
Изменяется методами:	PcdGetRicVersion

PortNumber НомерПорта

Номер COM порта.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	1
Допустимые значения:	0..0xFFFFFFFFh
Используется методами:	Connect , OpenPort
Изменяется методами:	SetDefaults

ReqCode

КодЗапроса

Код запроса метода [PiccCommonRequest](#).

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Используется методами:	PiccCommonRequest
Изменяется методами:	-

ResultCode

Результат

Результат выполнения последнего метода.

Тип:	Integer / Целое
Доступ:	R, только чтение
Значение по умолчанию:	0
Используется методами:	-
Изменяется методами:	всеми

ResultDescription

ОписаниеРезультата

Описание результат выполнения последнего метода.

Тип:	WideString / Строка
Доступ:	R, только чтение
Значение по умолчанию:	""
Используется методами:	-
Изменяется методами:	всеми

RfResetTime

ВремяОтключенияПоля

Время отключения RF поля в миллисекундах.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Используется методами:	PcdRfReset
Изменяется методами:	-

RICValue

ЗначениеRIC

Значение регистра микросхемы RIC.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Допустимые значения:	0..255
Используется методами:	
Изменяется методами:	

SAK

SAK

Значение ответа на команду выбора.

Тип:	Integer / Целое
Доступ:	R, только чтение
Значение по умолчанию:	0
Допустимые значения:	0..255
Используется методами:	-
Изменяется методами:	PiccCaseSelect , PiccSelect , PiccActivateIdle , PiccActivateWakeup

SectorCount

КоличествоСекторов

Значение ответа на команду выбора.

Тип:	Integer / Целое
Доступ:	R, только чтение
Используется методами:	GetSectorParams , SetSectorParams
Изменяется методами:	-

SectorIndex

ИндексСектора

Индекс сектора.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Используется методами:	GetSectorParams , SetSectorParams
Изменяется методами:	-

SectorNumber

НомерСектора

Номер сектора.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Используется методами:	GetSectorParams , SetSectorParams
Изменяется методами:	DeleteAppSectors

SelectCode

КодВыбора

Выбор уровня запроса.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Допустимые значения:	
PICC_ANTICOLL1	0x93, anticollision level 1 106 kBaud
PICC_ANTICOLL11	0x92, anticollision level 1 212 kBaud
PICC_ANTICOLL12	0x94, anticollision level 1 424 kBaud
PICC_ANTICOLL13	0x98, anticollision level 1 848 kBaud
PICC_ANTICOLL2	0x 95, anticollision level 2
PICC_ANTICOLL3	0x 97, anticollision level 3
Используется методами:	PiccCascAnticoll , PiccCascSelect
Изменяется методами:	-

Timeout

Таймаут

Таймаут чтения байта.

Тип:	Integer / Целое
------	-----------------

Доступ:	R/W, чтение/запись
Значение по умолчанию:	20
Допустимые значения:	0..0xFFFF, 0..65535
Используется методами:	InterfaceSetTimeout
Изменяется методами:	-

TransBlockNumber

НомерБлокаХранения

Номер блока для сохранения измененного значения.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Допустимые значения:	0..0xFF, 0..255
Используется методами:	PiccValue
Изменяется методами:	-

TransTime

ВремяОперации

Время операции.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Допустимые значения:	0..0xFFFFFFFF
Используется методами:	StopTransTimer
Изменяется методами:	StartTransTimer , StopTransTimer

UID

UID

Серийный номер карты в двоичном формате.

Тип:	WideString / Строка
Доступ:	R/W, чтение/запись
Значение по умолчанию:	""
Используется методами:	PiccAnticoll , PiccCascAnticoll , PiccCascSelect , PiccSelect , PiccActivateIdle , PiccActivateWakeup , PiccAuthE2 , PiccAuthKey
Изменяется методами:	PcdGetSerialNumber

UIDHex

UIDHex

Серийный номер карты в формате Hex.

Тип:	WideString / Строка
Доступ:	R/W, чтение/запись
Значение по умолчанию:	""
Используется методами:	PiccAnticoll , PiccCascAnticoll , PiccCascSelect , PiccSelect , PiccActivateIdle , PiccActivateWakeup , PiccAuthE2 , PiccAuthKey
Изменяется методами:	PcdGetSerialNumber

UIDLen

ДлинаUID

Длина серийного номера карты в байтах.

Тип:	Integer / Целое
Доступ:	R, только чтение
Значение по умолчанию:	0

Используется методами: -
Изменяется методами: [PiccActivateWakeup](#), [PiccActivateIdle](#)

ValueOperation

ОперацияНадЗначением

Тип операции над значением блока.

Тип: Integer / Целое
Доступ: R/W, чтение/запись
Значение по умолчанию: 0
Допустимые значения:
 voIncrement 0xC1, увеличение значения блока
 voDecrement 0xC0, уменьшение значения блока
 voRestore 0xC2, загрузка значения блока
 voTransfer 0xB0, сохранение значения блока
Используется методами: [PiccValue](#)
Изменяется методами: -

Version

ВерсияФайла

Версия файла драйвера.

Тип: WideString / Строка
Доступ: R, только чтение
Используется методами: -
Изменяется методами: -

PollStarted

ОпросНачат

Если свойство равно TRUE, то опрос начат. Иначе свойство равно FALSE.

Тип: WordBool / Логическое
Доступ: R, только чтение
Используется методами: -
Изменяется методами: -

PollInterval

ИнтервалОпроса

Интервал опроса в миллисекундах.

Тип: Integer / Целое
Доступ: R/W, чтение/запись
Значение по умолчанию: 500
Допустимые значения: 0..0xFFFFFFFF
Используется методами: PollStart
Изменяется методами: -

PollAutoDisable

АвтоотключениеОпроса

Автоматическое отключение опроса после обнаружения карты.

Тип: WordBool / Логическое
Доступ: R/W, чтение/запись
Значение по умолчанию: true
Используется методами: PollStart
Изменяется методами: -

UpdateTrailer ОбновитьТрейлер

Обновлять трейлер секторов при записи данных. Значение по умолчанию: False. Трейлер нужно обновлять при инициализации карты. При обновлении данных карта ключи скорее всего неизвестны и обновление трейлера не требуется.

Тип:	WordBool / Логическое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	False
Используется методами:	WriteFields, ReadFields
Изменяется методами:	-

DataAuthMode РежимАвторизации

Режим авторизации при чтении или записи полей. Возможные значения: 0..1. 0 – авторизация переданным ключом, 1 – авторизация по ключа считывателя.

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Используется методами:	WriteFields, ReadFields
Изменяется методами:	-

DataFormat ФорматДанных

Режим передачи данных. Свойство определяет формат данных для полей UncodedKey, EncodedKey, KeyA, KeyB, NewKeyA, NewKeyB. Допустимые значения: 0..1.

0 – dfHex – данные передаются в формате Hex

1 – dfBin – данные передаются в формате Bin

Тип:	Integer / Целое
Доступ:	R/W, чтение/запись
Значение по умолчанию:	0
Используется методами:	-
Изменяется методами:	-

Работа с каталогом

Каталог позволяет хранить на карте информацию нескольких приложений одновременно. Драйвер поддерживает работу с каталогом, разработанным компанией MickleSoft (Описание формата можно найти на <http://www.rfidchip.info>). В этом каталоге каждое приложение идентифицируется кодом приложения (AppCode) и кодом фирмы-разработчика этого приложения (FormCode). Для данных приложения может выделяться один или несколько секторов карты. Сектор 0 занят заголовком каталога, сектор 1 – данными каталога. Приложениям пользователя доступны сектора со 2 до конца карты (Для считывателей Mifare 1K это 2 –15 сектора).

На сегодня зарегистрированы следующие разработчики приложений:

0xA0 = MickleSoft - программы лояльности, контроль доступа и прочее не связанное с платежами

0xA1 = MickleSoft - решения, связанные с финансовыми транзакциями(операциями с цифровыми титульными знаками частных платёжных систем)

0x01 = ШТРИХ-М

0x02 = НТКФ "Гамбит"(г.Новосибирск)

0x03 = ["Центр готовых решений - Isoft"](#), г.Алмата, базовый системный интегратор Республики Казахстан.

Пример работы с каталогом в драйвере Mifare на Delphi:

```
Memo.Clear;
Update;
Memo.Lines.Add("");
Memo.Lines.Add('    Тест работы с каталогом');
Memo.Lines.Add(' -----');
// Запись каталога
Memo.Lines.Add(' Запись каталога:');
for i := 0 to Driver.SectorCount - 1 do
begin
    Driver.SectorIndex := i;
    Driver.FirmCode := 10+i;
    Driver.AppCode := 20+i;
    Check(Driver.SetSectorParams);
end;
Memo.Lines.Add(' -----');
Memo.Lines.Add(' Сектор  Фирма  Приложение');
for i := 0 to Driver.SectorCount - 1 do
begin
    Driver.SectorIndex := i;
    Check(Driver.GetSectorParams);
    Memo.Lines.Add(Format('%6d %6d %11d',[Driver.SectorNumber, Driver.FirmCode,
Driver.AppCode]));
end;
Memo.Lines.Add("");
Check(Driver.WriteDirectory);
// Очищаем каталог
Driver.AppCode := 0;
Driver.FirmCode := 0;
for i := 0 to Driver.SectorCount - 1 do
begin
    Driver.SectorIndex := i;
    Check(Driver.SetSectorParams);
end;
// Чтение каталога
Memo.Lines.Add(' Чтение каталога:');
Check(Driver.ReadDirectory);
Memo.Lines.Add(' -----');
Memo.Lines.Add(' Сектор  Фирма  Приложение');
for i := 0 to Driver.SectorCount - 1 do
begin
    Driver.SectorIndex := i;
    Check(Driver.GetSectorParams);
    Memo.Lines.Add(Format('%6d %6d %11d',[Driver.SectorNumber, Driver.FirmCode,
Driver.AppCode]));
end;
Memo.Lines.Add("");
Memo.Lines.Add(' Статус каталога: ' + Driver.DirectoryStatusText);
// Очищаем каталог на карте
Driver.AppCode := 0;
Driver.FirmCode := 0;
```

```
for i := 0 to Driver.SectorCount - 1 do
begin
  Driver.SectorIndex := i;
  Check(Driver.SetSectorParams);
end;
Driver.WriteDirectory;
```