

## Оглавление

<b>Оглавление</b>	<b>1</b>
<b>Общее описание протокола</b>	<b>3</b>
<b>Физический уровень</b>	<b>3</b>
Физическая среда RS-232	3
Физическая среда USB	3
<b>Транспортный уровень</b>	<b>3</b>
Кадр запроса	3
Кадр ответа	3
Кадры ответа ACK/NACK	4
Байтстаффинг	4
FCS	5
Многобайтовые переменные	5
О повторном запросе	5
<b>Общие рекомендации по протоколу</b>	<b>6</b>
Рекомендации по разбору кадра	6
Таймауты	6
Назначение идентификатора кадра	6
Смена скорости интерфейса	6
Пример исходника для приема данных	6
Примеры корректных кадров	7
Инструменты работы с протоколом	8
<b>Начало работы с картами Mifare</b>	<b>9</b>
<b>Описание команд</b>	<b>12</b>
Коды команд	12
Параметры считывателя	12
Время в системе	13
Авторизация и ограничение доступа	13
<b>Замечания по тексту</b>	<b>14</b>
Общие замечания по работе с картами	14
О времени исполнения команд	14
<b>Общие команды</b>	<b>15</b>
0x00 – "Заголовок устройства"	15
0x01 – "Запись параметров"	15
0x02 – "Чтение параметров"	16
0x03 – "Управление энергопотреблением"	16
<b>Команды управления</b>	<b>17</b>
0x21 – "Управление индикацией"	17
0x22 – "Сброс накачки"	17
0x23 – "Выключение накачки"	17
<b>Команды карт формата ISO-14443A</b>	<b>18</b>
0x40 - "Request"	18
0x43 - "Halt"	18
0x44 - "Anticollision+Select"	18
0x45 - "Request+Anticollision+Select"	19
0x46 - "Request + ReSelect"	19
0x4A - "Транзакция ISO14443A-4"	20
0x4F – управление скоростью RF-канала считывателя	20
<b>Команды карт формата MIFARE</b>	<b>22</b>
0x50 - "Авторизация"	22
0x51 - "Чтение блока 16 байт"	22
0x52 - "Запись блока 16 байт"	22
0x53 - "Запись блока 4 байт"	23
0x54 - "Increment"	23

0x55 - "Decrement" _____	23
0x56 - "Transfer" _____	23
0x57 - "Restore" _____	23
0x5A - "Return losted sector" _____	24
0x5B - "Быстрое чтение карты" _____	24
0x5F - "Запись ключа карты в защищенную память" _____	25
<b>Команды карт формата ISO-15693</b> _____	<b>26</b>
0x60 - "Команда ISO-15693" _____	26
<b><i>Примечания по алгоритмам работы</i></b> _____	<b>27</b>
<b>Индикация по умолчанию</b> _____	<b>27</b>
<b>Общие алгоритмы работы со считывателем</b> _____	<b>27</b>
<b>Рекомендуемые алгоритмы работы с картами Mifare Standard</b> _____	<b>27</b>
Структура карт Mifare Standard _____	27
Структура трейлера _____	28
Структура и назначение Value блока _____	29
<b><i>Замечания по реализации устройств</i></b> _____	<b>31</b>
Релиз 18.09.2003 _____	31
Релиз 14.11.2004 _____	31
Релиз 23.08.2005 _____	31
<b>Замеченные ошибки</b> _____	<b>31</b>
Ошибка записи во внутреннюю EEPROM _____	31
Ошибка авторизации из внутренней EEPROM _____	31
Проблема выхода из спящего режима _____	31
<b><i>История</i></b> _____	<b>32</b>
История документа _____	32
История реализаций устройств _____	32
<b><i>Контактная информация</i></b> _____	<b>33</b>
<b>Санкт-Петербург</b> _____	Ошибка! Закладка не определена.
<b>Техподдержка</b> _____	<b>33</b>

## Общее описание протокола

### Физический уровень

Имеются версии считывателей с физическими средами передачи RS-232, RS-422, RS-485, USB 1.1, Ethernet 10BaseT.

Во всех случаях (кроме Ethernet) работа ведется с виртуальными или физическими COM-портами. В случае Ethernet в протоколе изменений нет, однако вместо COM-порта поток данных направляется в порт UDP.

#### Физическая среда RS-232

Для связи между компьютером и считывателем используется последовательный асинхронный интерфейс со следующими настройками: 8 data bits, 1 stop bit, no parity, no flow control. Скорость при поставке 9600, может быть изменена в процессе работы.

Электрические параметры интерфейса: см. соответствующие стандарты. Используются линии RxD, TxD, GND.

Протокол дуплексный, наличие эха не допускается.

#### Физическая среда USB

Для связи между компьютером и считывателем используется интерфейс USB. Со стороны компьютера интерфейс представляется стандартным виртуальным COM-портом. Необходимый драйвер под ОС Windows идет в комплекте (SDK). Никаких изменений ПО компьютера не требуется.

О драйверах USB.

В первоначальных устройствах использовались драйвера VCP (виртуального COM-порта) компании FTDI (чип FT232BM, [www.ftdichip.com](http://www.ftdichip.com)). Обновления драйверов (в том числе для не-Windows ОС) можно свободно скачать с их сайта.

Позже основными при поставке стали CDC драйвера от Майкрософта. К сожалению, Майкрософт не уделяет должного внимания качеству этих драйверов и они не обновлялись уже много лет. Сам драйвер идет в комплекте считывателя или может быть скачан в виде обновления при подключении считывателя. Считыватели в которых основным является CDC драйвер сохраняют возможность использования драйверов FTDI или SiLabs (чип CP2101, [www.silabs.com](http://www.silabs.com)). За инструкциями по переключению драйверов обращайтесь в техподдержку.

### Транспортный уровень

Связь осуществляется кадрами в режиме «запрос-ответ». Инициатором обмена может быть только мастер системы (компьютер). Считыватель обязан подтвердить получение ответом.

#### Кадр запроса

Поле	Длина, байт	Данные
Стартовое условие	1	0xFD
Идентификатор кадра	1	
Код команды	1	См. Таблицу команд
Данные команды	XX	Зависит от команды
FCS кадра	2	
Стоповое условие	1	0xFE

После получения запроса считыватель проверяет правильность приема путем анализа FCS. Подтверждением правильного приема является передача запрашиваемых данных либо, для команд, не запрашивающих данные, кадр ответа ACK (acknowledged). При неправильном приеме для всех команд передается кадр NACK (not acknowledged), никакие другие действия не выполняются.

#### Кадр ответа

Поле	Длина, байт	Данные
Стартовое условие	1	0xFD
Идентификатор кадра	1	Повторение идентификатора из запроса
Код команды	1	Повторение кода команды из запроса
Данные команды	XX	Зависит от команды
FCS кадра	2	

Стоповое условие	1	0xFE
------------------	---	------

### Кадры ответа ACK/NACK

Поле	Длина, байт	Данные
Стартовое условие	1	0xFD
Идентификатор кадра	1	Повторение идентификатора из запроса
Признак кадра ACK/NACK	1	0x2A
Данные	1	0x55 – ACK 0x01 – NACK 1 (ошибка FCS) 0x02 – NACK 2 (неверная команда) 0x03 – NACK 3 (неверные данные) 0x04 – NACK 4 (ресурсы устройства исчерпаны) 0x05 – NACK 5 (аппаратная авария при исполнении) 0x06 – NACK 6 (валидная карточка не обнаружена) 0x07 – NACK 7 (карточка дала нераспознанный ответ) 0x08 – NACK 8 (ошибка авторизации карты) 0x09 – NACK 9 (карта отказала в исполнении команды)
FCS кадра	2	
Стоповое условие	1	0xFE

Для формирования ответа считывателем должны быть приняты как минимум стартовый и стоповый байты, идентификатор кадра, код команды и FCS. Также должны отсутствовать ошибки байтстаффинга в пределах кадра. Иначе никакие ответы не формируются и мастер должен повторить запрос по окончании таймаута, выставив соответствующий признак повторного запроса.

Считыватель вправе игнорировать новый запрос если он не успел полностью обработать предыдущий. Гарантируется что считыватель готов к приему новой команды к моменту окончания передачи первого кадра ответа на последний запрос.

Считыватель вправе игнорировать запрос если количество байт между стартовым и стоповым условиями больше чем оговорено самой большой командой.

Ответ NACK 1 посылается в случае ошибки FCS. Единственная цель такого ответа - ускорить повтор запроса до окончания таймаута.

Ответ NACK 2 посылается на все неверные коды команд (при условии совпадения FCS). Цель ответа - уведомить мастера об отсутствии поддержки команды в этой версии программы.

Ответ NACK 3 посылается на неприемлемый для данной команды набор данных (при условии совпадения FCS). Это может быть неверное количество байт данных, их неприемлемое значение или неразрешенные условия доступа к данным.

Ответ NACK 4 формируется для команд, работающих с лимитированными объемами памяти устройства (как на чтение, так и на запись) при исчерпании запрошенного ресурса. Например, это относится к памяти событий.

Ответ NACK 5 посылается некоторыми командами при выявлении сбоев устройства. Это может быть вызвано как неисправностью (случайным сбоем) считывателя, так и ошибками при программировании считывателя.

Ответ NACK 6 посылается командами работы с картой, если карта ответа не дала.

Ответ NACK 7 посылается командами работы с картой, если карта дала нераспознанный (несоответствующий запросу либо принятый с ошибками) ответ. Из-за шумов этот ответ может появляться и в случае отсутствия карты или ответа от нее (т.е. вместо NACK 6).

Ответ NACK 8 посылается командами работы с картой, когда карта ответила требованием авторизоваться либо произошла ошибка в процессе авторизации.

Ответ NACK 9 посылается командами работы с картой, когда карта ответила мотивированным отказом на команду. Мотивация указана в конкретных командах.

От компьютера к считывателю кадры ACK / NACK не посылаются: если компьютер не получил валидный ответ, запрос передается повторно согласно правилам повторного запроса.

### Байтстаффинг

Если между стартовым и стоповым условием встречаются специальные символы (0xFD, 0xFE, 0xFF), то они кодируются в соответствии с таблицей байтстаффинга:

Специальный символ	Кодирование
0xFD	0xFF 0x02
0xFE	0xFF 0x01
0xFF	0xFF 0x00

## FCS

FCS есть средство контроля целостности кадра. FCS считается над всеми полями кадра кроме стартового и стопового байт и поля самого FCS.

FCS в данном протоколе реализована согласно стандартам CCITT X.25 он же ISO/IEC 3309 он же RFC1331 (PPP). Контрольные последовательности, быстрые табличные реализации и другие фишки FCS могут быть найдены в соответствующих стандартах. Здесь приведен предельно упрощенный алгоритм реализации применительно к одному байту.

```
Unsigned int    FCS;
```

```
Void X_FCS (unsigned char _D)
```

```
{
unsigned char   _i;
unsigned int    _w;

    _w = (_D ^ FCS) & 0xFF;
    _i = 8;
    do {
        if (_w & 1) {
            _w >>= 1;
            _w ^= 0x8408;
        } else {
            _w >>= 1;
        }
    } while (--_i);
    FCS = _w ^ (FCS >> 8);
}
```

Для кадра имеем начальное значение FCS = 0xFFFF, байты считаются начиная с первого. По окончании расчета FCS инвертируется.

Вычисление FCS при передаче производится ДО проведения байтстаффинга, а при приеме сначала производится байтстаффинг, а потом производится проверка FCS.

## Многобайтовые переменные

Целые многобайтовые значения передаются младшим байтом вперед.

Числа с плавающей запятой имеют формат согласно IEEE754 и передаются экспонентой вперед.

Строки передаются первым символом вперед. Конец строки либо отмечается нулевым байтом, либо по достижении указанного размера поля. Если строка короче отведенного для нее поля, оставшиеся байты игнорируются (заполняются произвольным значением).

## О повторном запросе

Если мастер не смог получить от считывателя ответ на запрос, он повторяет запрос. Однако мастер не знает доподлинно, выполнил ли считыватель команду в первый раз или нет. Многие команды недопустимо или нежелательно выполнять повторно.

Чтобы избежать повторного выполнения команды, реализован простой механизм с использованием идентификатора кадра.

Мастер, не получив ответ, передает снова тот же запрос с тем же идентификатором кадра.

Считыватель, приняв кадр, анализирует идентификатор кадра и если последняя выполненная команда имела те же код команды и идентификатор кадра, игнорирует выполнение самой команды, а только повторяет передачу последнего ответа. При этом данные в новом кадре игнорируются. Если же код команды или идентификатор кадра не совпали с последними выполненными, команда исполняется полностью.

## Общие рекомендации по протоколу

### Рекомендации по разбору кадра

При разборе кадра следует придерживаться следующих правил.

- Стартовое условие обладает наивысшим приоритетом при разборе потока.
- Стоповое условие обладает вторым уровнем приоритета. Принятие стопового условия есть сигнал к началу разбора принятого кадра. Весь поток до следующего стартового условия отбрасывается.
- Байтстаффинг обладает третьим уровнем приоритета. Ошибка байтстаффинга в процессе разбора потока есть основание для отбрасывания всего потока до следующего стартового условия.
- Разбор кадра начинается с проверки минимально допустимой длины кадра (длина кадра без поля данных) и его FCS (ищется как смещение от стопового условия). Все остальные поля ищутся как смещения от стартового условия.

### Таймауты

Утверждается, что суммарное время от стоп-бита последнего байта кадра запроса до стоп-бита последнего байта кадра ответа не превышает суммы: задержка считывателя на формирование ответа плюс время на передачу ответного кадра плюс 10 мс. Задержка считывателя на формирование ответа указывается индивидуально для каждой команды.

При передаче данных считывателем максимальная задержка между байтами (окончанием одного и началом другого) не нормируется.

При передаче кадра компьютером не ставится условий по максимальной задержке между байтами одного кадра.

### Назначение идентификатора кадра

В медленных компьютерах с громоздкими ОС при плотном трафике наблюдается эффект рассинхронизации запросов и ответов. Идентификатор кадра позволяет решить эту проблему. Компьютер назначает каждому новому запросу новый идентификатор и при приеме точно знает, какому запросу соответствовал ответ. Рекомендуется инкрементировать идентификатор каждый раз перед посылкой нового запроса.

Кроме того, идентификатор кадра используется в алгоритме опознавания повторных запросов и множественных ответов на одну команду.

### Смена скорости интерфейса

Смена скорости производится немедленно после ответа АСК на команду смены скорости.

Потенциально возможна ситуация рассогласования рабочей скорости на компьютере и считывателе. Абстрагируясь от причин такой ситуации, предложим универсальное решение этой проблемы. В случае если считыватель не отвечает на запросы мастера, рекомендуется выполнить опрос считывателя какой-либо нейтральной командой на всех разрешенных скоростях и автоматически переустановить скорость на рабочую.

### Пример исходника для приема данных

```
Void RX_Int(char _UDR)
{
static char   _Q;
static u16_t  _CRC;
static char   _State = 0;

if (_UDR == 0xFD) {
    _State = 1;
    _Q = 0;
    _CRC = 0xFFFF;
} else {
    switch (_State) {
    default:
        _State = 0;
        break;
    case 1:
```

```

switch (_UDR) {
case 0xFF:
    _State = 2;
    break;
case 0xFE:
    if (_Q >= 4) {
        if (_CRC == 0xf0b8) {
            // Кадр с данными успешно принят...
        }
    }
    _State = 0;
    break;
default:
    goto _Rcv_GetChar;
}
break;
case 2:
    if (_UDR <= 0x02) {
        _UDR = 0xFF - _UDR;
        _State = 1;
    } else {
        _State = 0;
    }
}
_Rcv_GetChar:
if (_Q >= UART_RX_BufLength) {
    _State = 0;
} else {
    u16_t _w;

    // Сохранение принятого байта в буфере
    UART_RX_Buf[_Q++] = _UDR;

    // Расчет CRC принятого байта
    _w = (_UDR ^ _CRC) & 0xFF;
    _UDR = 8;
    do {
        if (_w & 1) {
            _w >>= 1;
            _w ^= 0x8408;
        } else {
            _w >>= 1;
        }
    } while (--_UDR);
    _CRC = _w ^ (_CRC >> 8);
}
break;
}
}
}

```

### Примеры корректных кадров

Все значения в примерах - шестнадцатичные. Точки показаны только для удобства чтения.

Запрос заголовка устройства:

FD.00.00.47.0F.FE

Ответ ACK:

FD.00.2A.55.A7.1D.FE

Ответ NACK 2:

FD.00.2A.02.9D.3B.FE

## Инструменты работы с протоколом

Имеется два основных инструмента для работы с данным протоколом. Оба ориентированы на демонстрацию и тестирование функциональности считывателя.

Первый инструмент – низкоуровневая терминальная программа W\_Term.exe, заточенная на работу с данным протоколом. Ее назначение – работа с на уровне транспортного протокола. Терминалка поддерживает работу с COM портами и с IP-стеком; упаковывает заданную пользователем команду и данные в кадр протокола; распознает и распаковывает кадр ответа; считает таймауты. Ее всегда можно скачать с нашего сайта или получить по адресам техподдержки.

Второй инструмент – высокоуровневая программа демонстрации функциональности собственно считывателей карт CashTest.exe. Позволяет эффективно работать с конкретными командами считывателей и с картами соответствующих стандартов. Этот инструмент регулярно обновляется, запрашивайте новые версии.

Оба инструмента основаны на ProX232.dll, реализующей низкий уровень данного протокола. Описание интерфейса dll доступно в отдельном документе.

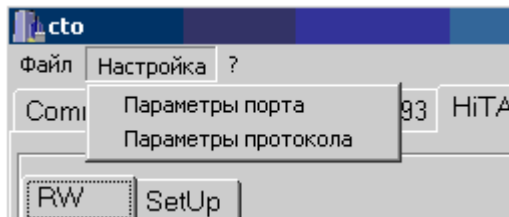
**ВНИМАНИЕ!** Имелось несколько версий ProX232.dll. Основные отличия:

- Старые версии библиотеки содержали ошибку из за которой не умели корректно открывать COM-порты выше 10-го.
- Также старые версии ProX232.dll при каждом обмене с устройством открывали и закрывали порт. Плюс подобного подхода – с устройством могут работать несколько независимых программ одновременно. Минус – процедура открытия-закрытия виртуальных COM-портов (USB, Bluetooth) занимает весьма значительное время и резко тормозит обмен.

Далее кратко изложен порядок работы с инструментами.

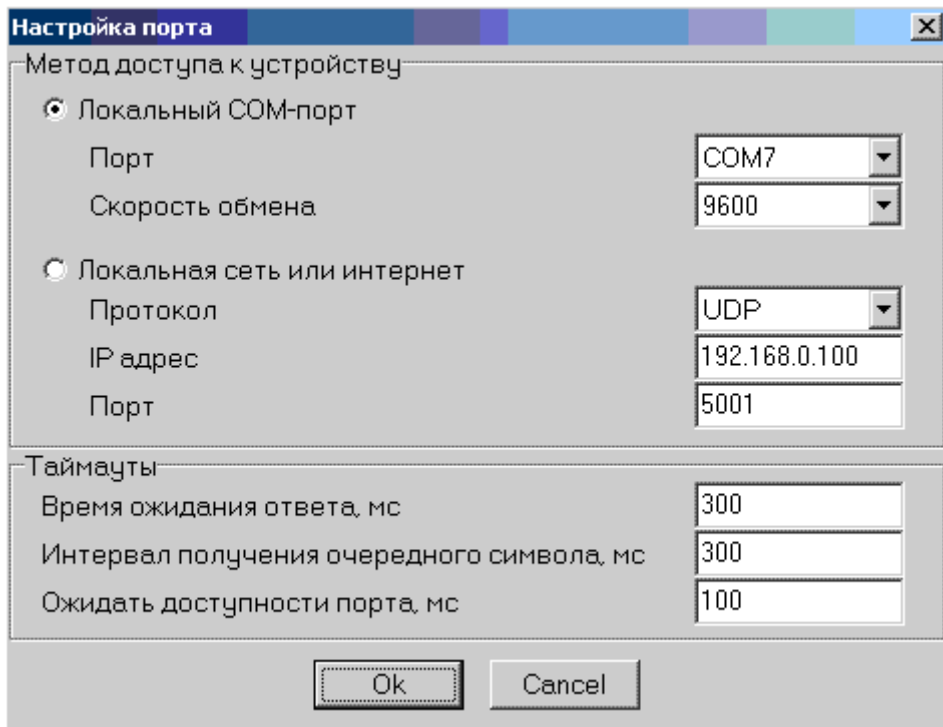
Оба инструмента не имеют инсталлятора и запускаются «как есть». В одной папке должны находиться для CashTest: CashTest.exe, ProX232.dll, CashTest.ini. Для терминалки: W\_Term.exe, ProX232.dll. При этом для CashTest желательна возможность перезаписи CashTest.ini, иначе невозможно будет сменить настройки программы. Терминалка хранит свои настройки в реестре (путь ).

При первом запуске необходимо установить параметры порта считывателя (в первую очередь номер COM порта или параметры IP соединения для Ethernet считывателей). Как это сделать – поясняют картинки ниже.

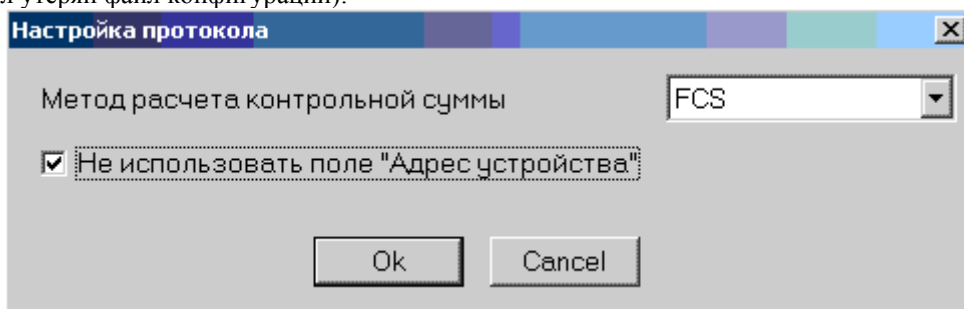


К настройке порта следует сказать, что USB считыватели видны в системе как виртуальный COM-порт (после установки драйверов, естественно). Номер COM порта можно узнать в свойствах системы (например, наберите compmgmt.msc и там выберите “Device manager”->”Ports (COM & LPT)”).





Параметры настройки протокола следует просто проверить (или выставить заново как на картинке если был утерян файл конфигурации):

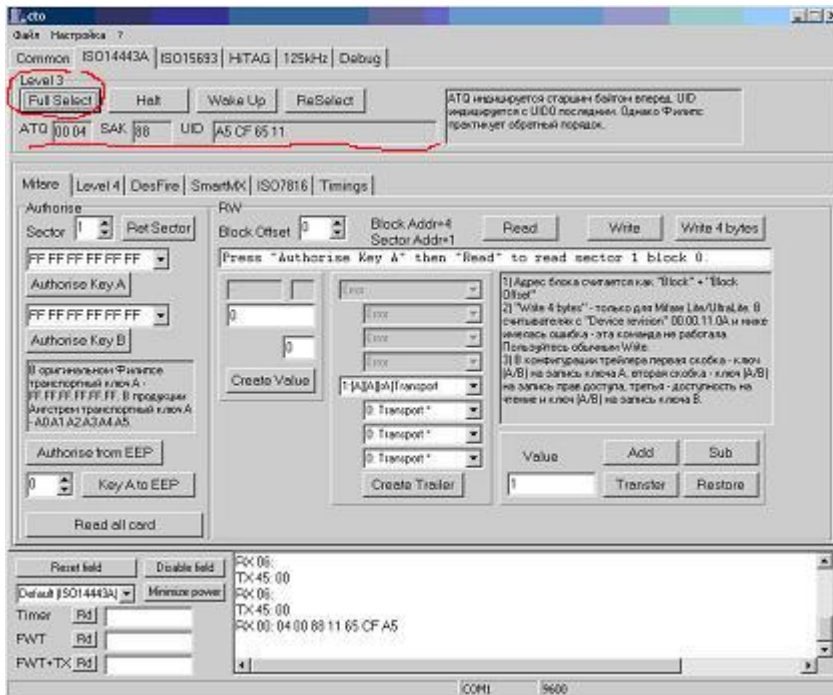


### Начало работы с картами Mifare

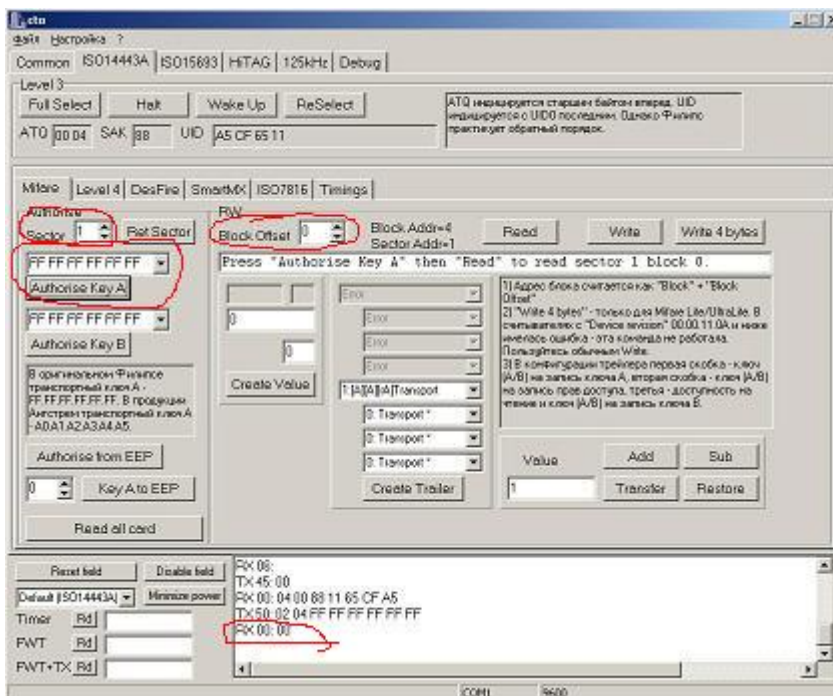
Положите карту на считку. Нажмите Full Select (см. рис. ниже). Если все хорошо – появится номер карты.

Если карта не ответила (хотя и лежит на считывателе), то, возможно, считыватель в режиме энергосбережения. Нажмите Reset Field чтобы отменить его. Снова попоробуйте Full Select.

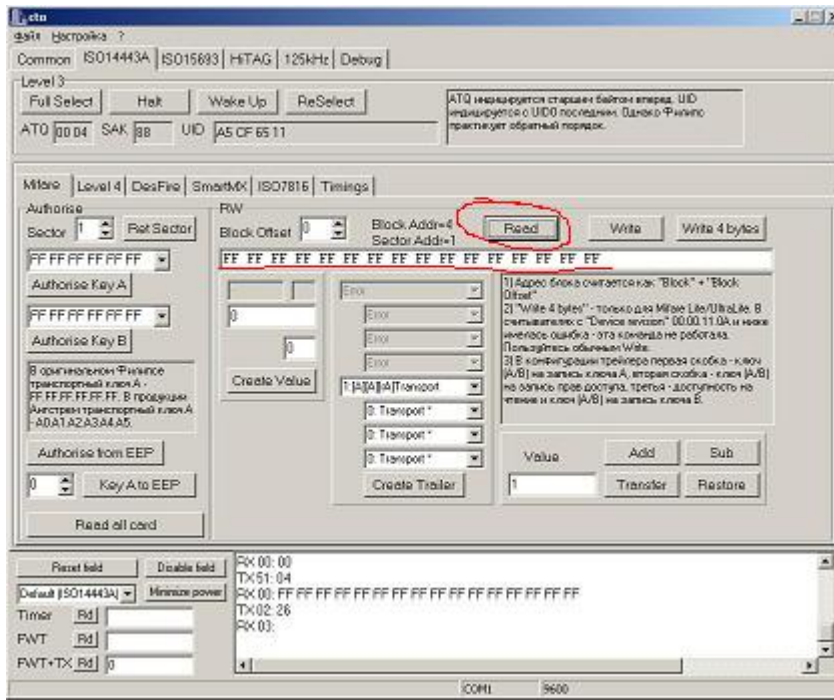
Примечание: если вы снова попытаетесь нажать на Full Select – карта будет отвечать через раз. Если хочется понять почему это и зачем – читайте ISO14443 и спецификацию Mifare.



Далее начинают играть роль КЛЮЧИ карты. При продаже карта имеет так называемые транспортные ключи (FF.FF.FF.FF.FF.FF). Для такой карты выберите нужные Вам Sector и Block Offset, ключ A и нажмите Authorise Key A. Если успешно – внизу придет ответ 0, если ключ не подошел – карта не ответит (ответ 06 или 07) и вам надо начать сначала и попробовать другой ключ.



Теперь вы успешно авторизовались в секторе и можно читать-писать. Нажмите Read и посмотрите результат. Block Offset позволяет выбрать блок в пределах сектора. Если меняете сектор – нужно заново авторизовываться в секторе.



## Описание команд

### Коды команд

Код команды имеет длину 1 байт. Коды команд считывателя сведены в таблицу:

Таблица команд считывателя

Команда	Код Команды	Возможные ответы
<b>Общие команды</b>		
Заголовок устройства	0x00	Данные / NACK
Запись параметров	0x01	ACK / NACK
Чтение параметров	0x02	Данные / NACK
Управление энергопотреблением	0x03	ACK / NACK
<b>Команды управления</b>		
Управление индикацией	0x21	ACK / NACK
Сброс накачки	0x22	ACK / NACK
Выключение накачки	0x23	ACK / NACK
<b>Команды карт формата ISO-14443A (базовые)</b>		
Request	0x40	Данные / NACK
Halt	0x43	ACK / NACK
Anticollision + Select	0x44	Данные / NACK
Request + Anticollision + Select	0x45	Данные / NACK
Request + ReSelect	0x46	ACK / NACK
Транзакция ISO14443A-4	0x4A	Данные / NACK
Управление скоростью приема-передачи RF-канала считывателя	0x4F	ACK / NACK
<b>Команды карт формата MIFARE (базовые)</b>		
Authentication	0x50	Данные / NACK
Read 16	0x51	Данные / NACK
Write 16	0x52	ACK / NACK
Write 4	0x53	ACK / NACK
Increment	0x54	ACK / NACK
Decrement	0x55	ACK / NACK
Transfer	0x56	ACK / NACK
Restore	0x57	ACK / NACK
Return losted sector	0x5A	ACK / NACK
Write Mifare Key to secure EEPROM	0x5F	ACK / NACK
<b>Команды карт формата ISO-15693</b>		
Команда ISO-15693	0x60	Данные / NACK

### Параметры считывателя

В считывателе есть ряд параметров. Каждый параметр снабжен кодом и доступен по чтению и/или по записи с помощью команд чтения и записи параметров. Код параметра имеет длину 1 байт.

При записи любого параметра реальное его значение в устройстве обновляется немедленно после ответа ACK.

Несуществующие коды параметров или ненормированный способ доступа к ним вызывает ответ NACK 3.

Таблица параметров считывателя

Код Параметра	Описание	Длина, байт	Возможные значения	Доступ
0x02	Скорость интерфейса	1	Перечислимый тип: 3 - скорость 9600 4 - скорость 19200	Rd/Wr

			5 - скорость 38400 6 - скорость 57600 7 - скорость 115200 8 - скорость 230400** 9 - скорость 460800** 10 - скорость 921600**	
0x03	Уникальный номер изделия	4		Rd/Wr*
0x06	Набор общих флагов	1	См. Таблицу общих флагов	Rd/Wr
0x20	Таймаут на запись ISO15693	1	Выбирается из ряда 2, 4, 6, 8, 12, 16, 20, 26 мс. (значения 0...7). По умолчанию 26 мс.	Rd/Wr
0x21	Глубина модуляции ISO15693	1	Абстрактная величина в диапазоне 0...0x3F.	Rd/Wr
0x80	Ячейка авторизации	8		Rd/Wr
0x81	Дата начала гарантии	6	См. Время в системе.	Rd/Wr*
0x82	Набор специальных флагов	1	См. Таблицу специальных флагов	Rd/Wr*
0x83	Счетчик ошибок интерфейса	1	Инкрементируется при каждой ошибке FCS	Rd
0x90	Конфигурация системы	1	Внутривыпускной параметр	Rd/Wr*

\* - некоторые изделия могут иметь жестко заданное при производстве значение параметра либо запись разрешается только после авторизации.

\*\* - скорости выше 115200 поддерживаются не всеми считывателями (см. историю реализаций).

Если эти скорости не поддерживаются, считыватель ответит NACK3 на попытку их установить. Установка скорости выше 115200 не сохраняется энергонезависимо и сбрасывается на 9600 при включении питания.

Параметр «Набор общих флагов» есть битовая структура. Состав структуры:

Бит	Назначение
0-6	Резерв
7	Разрешение скоростей выше базовой для ISO14443A (0 - только базовая скорость).

Параметр «Набор общих флагов» энергонезависимый.

Параметр «Набор специальных флагов» есть битовая структура. Состав структуры:

Бит	Назначение
0-6	Конфигурационные параметры
7	Включение в ответ отладочной информации (1 - есть отладочная информация).

Параметр «Набор специальных флагов» энергонезависимый, однако бит 7 при включении питания всегда выставляется в ноль.

### Время в системе

Время в системе представлено 6-байтовой структурой.

Первым идет год. Порядок следования полей указан в таблице. Каждое поле занимает 1 байт.

Тип содержимого	Допустимые значения	Примечание
Год	00...99	2000...2099
Месяц	1...12	
День месяца	1...31	Учет високосных годов
Часы	0...23	Нет перехода летнее-зимнее время
Минуты	0...59	
Секунды	0...59	

В изделиях, не имеющих энергонезависимых часов, при включении питания в счетчик записывается значение 1 января 2001 года.

### Авторизация и ограничение доступа

Процедура авторизации является закрытой для пользователя.

Попытка пользователя обратиться к параметрам, закрытым для прямого доступа, вызывает ответ NACK 3.

## Замечания по тексту

### Общие замечания по работе с картами

Если не указано иначе, никаких дополнительных проверок факта исполнения команды картой не производится. Проверок валидности команды применительно к данной карте также не производится. Отсутствие ответа карты транслируется считывателем в NACK6, нераспознанные ответы или ошибки в теле ответа карты транслируются считывателем в NACK7. В любом случае команда не повторяется если не оговорено иначе.

### О времени исполнения команд

Команды бывают однократные и продолжительные. Однократные команды вызывают один ответ на команду и переходят в режим ожидания. Продолжительные команды (например, команды поиска карты) продолжают выполняться после первого (обязательного) ответа на команду и могут формировать повторные ответы.

Везде под временем исполнения команды карте подразумевается время на первый (обязательный) ответ.

Время исполнения команды (а значит и время на ответ) зависит от нескольких условий:  
Собственно чистое время исполнения команды.

Время на инициализацию радиочастотной части в случае, если считыватель находился в режиме энергосбережения, была выключена накачка или произошла смена стандарта карт. Инициализация производится только если это требуется для выполнения команды.

Время на завершение предыдущей команды в случае, если предыдущая команда была продолжительной.

## Общие команды

### 0x00 – "Заголовок устройства"

Команда считывает из устройства его тип, номер версии, номер устройства.

Время исполнения команды: немедленно.

Код команды: 0x00.

Команда поля данных не имеет.

Варианты ответа: кадр ответа NACK или ответ со структурой данных, описывающей устройство:

Поля данных	Длина, Байт	Данные
Тип устройства	20	Текстовая строка с названием устройства.
Идентификатор устройства	4	unsigned long
Номер версии устройства	4	unsigned long
Номер версии протокола	4	unsigned long
Уникальный номер изделия	4	unsigned long
Дополнительные флаги	4	unsigned long

Назначение полей:

«Тип устройства» - для текстовой справки.

«Идентификатор устройства» - идентификатор типа считывателя. Значения указываются в разделе «История».

«Номер версии устройства» - для справки в разделе About. Значения указываются в разделе «История».

«Номер версии протокола» - для справки в разделе About. Нужно предусмотреть возможность сверки со списком поддерживаемых версий. Текущее значение указано в разделе «История».

«Уникальный номер изделия» - для справки в разделе About. Некоторые изделия не поддерживают энергонезависимое хранение этого параметра, при снятии питания сбрасывают его в 0. Некоторые изделия могут иметь жестко заданный при производстве номер. Значение то же, что и в параметре «Уникальный номер изделия».

«Дополнительные флаги» - инструмент индикации особенностей реализации. Битовый массив, назначение бит:

- Биты 0...7 – поддержка ISO14443A.
  - Бит 0 – имеется поддержка ISO14443A-3.
  - Бит 1 – имеется поддержка нескольких карт в поле (антиколлизия в ISO-14443A).
  - Бит 2 – имеется поддержка ISO14443A-4.
  - Бит 3 – имеется поддержка всех скоростей ISO14443A.
  - Бит 4 – имеется поддержка карт Mifare Lite, Mifare Standart (Philips).
  - Биты 5...7 – резерв.
- Биты 8...15 – поддержка ISO15693 и его клонов.
  - Бит 8 – имеется поддержка ISO15693.
  - Бит 9 – резерв.
  - Бит 10 – имеется поддержка I-CODE (Philips).
  - Бит 11 – имеется поддержка TagIt (Texas Inst.).
  - Биты 11...15 – резерв.
- Биты 16...23 – резерв (Future ISO14443B support).
- Биты 24...26 – резерв.
- Бит 27 – установка бита означает что нет поддержки продолжительных команд.
- Биты 28...31 – максимальный размер одной транзакции с картой (максимальное количество данных передаваемых в карту или принимаемых из карты без учета CRC). Эквивалент параметров FSD и FSC в ISO14443-4. Выбирается из ряда 16, 24, 32, 40, 48, 64, 96, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 байт (значения поля 0...0x0F). Этот параметр отражает ограничения самого считывателя и никак не отражает ограничения карты, управляющая программа должна учитывать их дополнительно. Это ограничение применимо к ISO14443A-4, ISO15693, ISO14443B-3, ISO7816.

### 0x01 – "Запись параметров"

Команда записывает в устройство параметры.

Время исполнения команды: не более 10 мс.

Код команды: 0x01.

Данные команды:

Тело команды	Длина, Байт	Данные
Код параметра	1	См. таблицу "Коды параметров".
Значение параметра	1...20	Длина зависит от кода параметра

Варианты ответа: Кадр ответа ACK / NACK.

### 0x02 – "Чтение параметров"

Команда читает из устройства значения параметров его работы. Команда дуальна по отношению к команде 0x01 – "Запись параметров".

Время исполнения команды: не более 5 мс.

Код команды: 0x02.

Данные команды:

Поле	Длина, байт	Данные
Код параметра	1	См. таблицу "Коды параметров".

Коды параметров совпадают по назначению и длине возвращаемых данных с кодами команды 0x01.

Варианты ответа: Кадр ответа NACK или данные:

Поле данных ответа	Длина, Байт	Данные
Значение параметра	1...20	Зависит от кода параметра

### 0x03 – "Управление энергопотреблением"

Команда снижает потребление устройства до минимума, обеспечивая лишь реакцию на команды интерфейса. Конкретные значения потребления в разных режимах определяются версией устройства.

Для выхода из режима подайте команду 0x22 – "Сброс накачки".

Код команды: 0x03.

Время исполнения команды: немедленно.

Команда данных не содержит.

Варианты ответа: Кадр ответа ACK / NACK.



## Команды управления

### 0x21 – “Управление индикацией”

Время исполнения команды не более 5 мс.

Код команды: 0x21.

Данные команды:

Тело команды	Длина, Байт	Данные
Параметр	1	Биты 0-1: 00 – выключение зеленого светодиода; 01 – включение зеленого светодиода; 10 – периодическое мигание зеленого светодиода; 11 – сопровождать пищалку; Биты 2-3: 00 – выключение красного светодиода; 01 – включение красного светодиода; 10 – периодическое мигание красного светодиода; 11 – сопровождать пищалку; Биты 4-5: 00 – пищалка выключена; 01 – одиночный писк; 10 – двойной писк; 11 – одиночный длинный писк; Биты 6-7: 00 – индикация как определено в битах 5...0; 10 – индикация по умолчанию.

Варианты ответа: Кадр ответа ACK / NACK.

### 0x22 – "Сброс накачки"

Команда кратковременно сбрасывает накачку с целью рестарта всех карт в поле. Если накачка была выключена, то ее включают. При сбросе накачки производится перезагрузка рабочего формата карт для считывателей, поддерживающих несколько форматов.

Код команды: 0x22.

Время исполнения команды: TBD мс.

Команда имеет два формата.

Первый формат не содержит данных в запросе, при этом при сбросе накачки тип карт ставится по умолчанию.

Второй формат содержит 1 байт данных:

Тело команды	Длина, Байт	Данные
Параметр	1	Тип карт: 0 – ISO14443A 1 – ISO15693 2 – ISO14443B 3 – I-CODE 4 – TagIt

Варианты ответа: Кадр ответа ACK / NACK.

### 0x23 – "Выключение накачки"

Команда выключает накачку. Для включения накачки подайте команду 0x22 – "Сброс накачки".

Код команды: 0x23.

Время исполнения команды: немедленно.

Команда данных не содержит.

Варианты ответа: Кадр ответа ACK / NACK.

## Команды карт формата ISO-14443A

### 0x40 - "Request"

Команда осуществляет полный цикл выбора карты.

Код команды: 0x40.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	См. ниже.

Параметры команды: байт с битовыми флагами:

- Биты 0...4: игнорируются.
- Бит 6: длительный поиск (1) или однократный (0). Некоторые считыватели не поддерживают длительный поиск, см. команду «заголовок устройства».
- Бит 7: Request IDLE (0) or ALL (1).

Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
ATQ карты	2	

Алгоритм работы команды:

Если задан однократный поиск, считыватель пытается обнаружить карту командой Request. Если карта найдена, дает соответствующий ответ. Если карта не обнаружена отвечает NACK6 (или 7).

Если задан длительный поиск, считыватель полностью выполняет действия однократного поиска (включая полноценный ответ). Если поиск неудался, то примерно через каждые 50 мс поиск повторяется. Как только карта обнаружена, считыватель отправляет кадр ответа с прочитанными данными и прекращает поиск карты.

Если задан длительный поиск, то считыватель каждый раз обращаясь к карте отправляет квиток. Квиток - это неполный кадр (только стартовое условие и идентификатор команды) он дает компьютеру знание что считыватель жив и выполняет последнюю команду. После обнаружения карты и отсылки полноценного ответа никакие действия с картой не ведутся и, соответственно, компьютеру ничего не шлетя.

Все кадры ответа на команду имеют один и тот же идентификатор.

Прекращение длительного поиска происходит по поступлению любой другой команды. Если в момент поступления новой команды началась процедура поиска карты, то она будет доведена до конца включая передачу кадра с данными и только после этого будет выполнена новая команда. Это может изменить время задержки реального ответа на новую команду.

Согласно стандарту ISO14443A ATQ карты есть логическое ИЛИ ATQ всех активных карт в поле.

### 0x43 - "Halt"

Код команды: 0x43.

Время исполнения команды: TBD мс.

Тело команды: команда данных не имеет.

Варианты ответа: Кадр ответа ACK / NACK.

NACK 9 означает отказ карты перейти в спящий режим.

### 0x44 - "Anticollision+Select"

Команда осуществляет цикл выбора карты.

Код команды: 0x44.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	См. ниже.

Параметры команды: байт с битовыми флагами:

- Биты 0...7: игнорируются.

Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
SAK карты	1	

UID карты	4, 7 или 10	Первыми идут старшие UID (UID0). Длина поля определяется длиной кадра.
-----------	-------------	--

Скорость работы с картой выбирается согласно ISO14443A по последнему валидному ATQ, полученному командой Request.

Кроме того, последний ATQ карты (полученный последней командой Request) проверяется на поддержку картой антиколлизии. Если поддержка не указана, ответ NACK5.

#### 0x45 - “Request+Anticollision+Select”

Команда осуществляет полный цикл выбора карты.

Код команды: 0x45.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	См. ниже.

Параметры команды: байт с битовыми флагами:

- Биты 0...5: игнорируются.
- Бит 6: длительный поиск (1) или однократный (0). Некоторые считыватели не поддерживают длительный поиск, см. команду «заголовок устройства».
- Бит 7: Request IDLE (0) or ALL (1).

Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
ATQ карты	2	
SAK карты	1	
UID карты	4, 7 или 10	Первыми идут старшие UID (UID0). Длина поля определяется длиной кадра.

Алгоритм работы команды:

Если задан однократный поиск, считыватель пытается обнаружить карту командой Request. Если карта найдена, дает соответствующий ответ. Если карта не обнаружена отвечает NACK6 (или 7).

Если задан длительный поиск, считыватель полностью выполняет действия однократного поиска (включая ответ). Если поиск неудался, то примерно через каждые 50 мс поиск повторяется. Как только карта обнаружена, считыватель отправляет кадр ответа с прочитанными данными и прекращает поиск карты.

Если задан длительный поиск, то считыватель каждый раз обращаясь к карте отправляет квиток. Квиток - это неполный кадр (только стартовое условие и идентификатор команды) он дает компьютеру знание что считыватель жив и выполняет последнюю команду. После обнаружения карты и отсылки полноценного ответа никакие действия с картой не ведутся и, соответственно, компьютеру ничего не шлется.

Все кадры ответа на команду имеют один и тот же идентификатор.

Прекращение длительного поиска происходит по поступлению любой другой команды. Если в момент поступления новой команды началась процедура поиска карты, то она будет доведена до конца включая передачу кадра с данными и только после этого будет выполнена новая команда. Это может изменить время задержки реального ответа на новую команду.

Если ATQ карты указывает на отсутствие поддержки антиколлизии, то дальнейшие процедуры не выполняются и в ответе поля SAK и UID отсутствуют (кадр имеет длину данных 2 байта).

Согласно стандарту ISO14443A ATQ карты есть логическое ИЛИ ATQ всех активных карт в поле.

#### 0x46 - “Request + ReSelect”

Команда предназначена для восстановления контроля над текущей картой в случае ошибки связи. Команда осуществляет прямой выбор последней активной карты по ее UID без выполнения процедуры антиколлизии.

Код команды: 0x46.

Время исполнения команды: TBD мс.

Тело команды: команда данных не имеет.

Варианты ответа: Кадр ответа ACK / NACK.

Выбирается всегда базовая скорость работы с картой.

Команда позволяет вернуть контроль над картой при ошибке связи. Считыватель не производит процедуру антиколлизии, а целенаправленно пытается выбрать карту с последним известным ему UID.

На случай если карта находилась в состоянии selected в момент подачи команды запросы карте будут продублированы при неответе в первый раз.

Внимание! Результат команды не детерминирован если последняя команда, содержащая антиколлизии, окончилась неудачей.

#### 0x4A - “Транзакция ISO14443A-4”

Команда передает данные карте согласно ISO14443A-4 и запускает ожидание ответа. В карту передается все содержимое поля данных. Таймаута на прием нет, команда прекращает ожидание ответа от карты только после подачи другой команды на работу со считывателем. Ответов на одну команду может быть неограниченно много, определяется количеством сообщений от карты.

Код команды: 0x4A.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	См. ниже.
Данные, передаваемые карте	XX	

Параметры команды: байт с битовыми флагами:

- Бит 0: отмена холостого ответа АСК на команду.
- Биты 1...7: резерв, должны быть 0.

Варианты ответа: кадр ответа АСК, NACK или данные ответа:

Поле данных ответа	Длина, Байт	Данные
Ответ карты	XX	

Все ответы на эту команду будут иметь один и тот же идентификатор кадра.

После получения команды считыватель:

- передает карте данные;
- отвечает компьютеру АСК (если бит 0 параметров команды выставлен в 1 ответ АСК не посылается);
- переходит в режим ожидания сообщений от карты или новых команд от компьютера.

Выполнение команды прекратится если будет послана любая другая команда работы с картой.

Допускаются команды работы с таймером, индикацией. Если новая команда пришла в момент когда карточка передает сообщение, то сначала будет закончен прием этого сообщения и оно будет передано компьютеру, и затем обработана новая команда.

Таймер времени исполнения команды сбрасывается после каждой транзакции. Соответственно отчет о времени исполнения команды – это время ожидания от окончания предпоследней транзакции (передачи или приема) до окончания последнего приема. Если приема еще не было, сохраняется старое значение.

При передаче считыватель автоматически добавляет к сообщению CRC согласно стандарту ISO14443A. При приеме считыватель анализирует правильность CRC и передает компьютеру только данные без CRC. Ошибка CRC, равно как и другие ошибки при приеме вызывают сообщение NACK7 и продолжение ожидания новых сообщений.

Вспомогательные сообщения (например, WTX) никак не интерпретируются считывателем.

Соответственно реакция на них – задача управляющей программы компьютера.

О минимальных межфреймовых интервалах. Компьютер может перегрузить межфреймовые защитные интервалы новыми значениями (см. команду 0x4E), соответственно считыватель их будет соблюдать. Однако следует помнить что команды работы с картой в которых так или иначе присутствует команда карте ISO14443A3 Request будут сбрасывать межфреймовые интервалы в значения по умолчанию.

Ограничения команды. Размер данных, передаваемых карте или принимаемых от карты, не может превышать максимальный размер одной транзакции с картой (см. команду «Заголовок устройства»).

#### 0x4F – управление скоростью RF-канала считывателя

Команда осуществляет смену скорости радиоинтерфейса считывателя согласно ISO14443A-2.

ВНИМАНИЕ! Команда поддерживается не всеми считывателями, см. заголовок устройства.

Код команды: 0x4F.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
DSI	1	0..3
DRI	1	0..3

Варианты ответа: Кадр ответа АСК / NACK.

Назначение DRI/DSI – см. ISO14443A.

Любая команда так или иначе выбирающая карту (Select) сбрасывает эти параметры к значению по умолчанию (DRI/DSI = 0/0).

## Команды карт формата MIFARE

### 0x50 - “Авторизация”

После этой команды считыватель авторизует себя в указанном секторе карты, выделенной последней командой «Select». Если успешно, отвечает ACK, иначе NACK6, 7.

Код команды: 0x50.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	См. ниже.
Адрес блока для авторизации	1	0...255. Это адрес блока в секторе, а не сектора!
Параметр ключа карты	6	

Параметры команды: байт с битовыми флагами:

- Бит 0: Ключ А (0) или Ключ В (1).
- Бит 1: Ключ берется из защищенной памяти ключей (0) или содержится непосредственно в команде (1).
- Биты 2...7: игнорируются.

Варианты ответа: Кадр ответа NACK или данные:

Поле данных ответа	Длина, байт	Данные
Номер ключа, которым авторизовались	1	Для ключа из памяти: 0...47, число соответствует биту в маске ключей, которым успешно авторизовались. Для ключа из команды: всегда 0.

Алгоритм работы команды:

Если ключ содержится в команде, то параметр ключа карты содержит собственно ключ в явном виде.

Если ключ берется из защищенной памяти ключей, то параметр ключа карты содержит битовую маску, каждый бит которой сопоставлен ключу, который должен быть опробован для авторизации. Соответственно возможен перебор многих ключей в одной команде. Процесс перебора начинается с младшего бита.

Биты 0...31: соответствуют адресам защищенной памяти ключей (младший бит - младшему адресу).

Биты 32...47: не используются.

Очевидно, что чем больше ключей указано, тем дольше выполняется команда.

Ответ NACK 5 генерится если в указанных ячейках памяти ключей не нашлось ни одного валидного ключа.

### 0x51 - “Чтение блока 16 байт”

После этой команды считыватель читает указанный блок (16 байт) в авторизованном секторе. Если успешно, отвечает данными, иначе NACK6 (или 7). Если сектор неавторизован NACK8.

Код команды: 0x51.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Адрес блока	1	

Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
Данные блока	16	Нулевой байт идет первым.

### 0x52 - “Запись блока 16 байт”

После этой команды считыватель пишет указанный блок (16 байт) в авторизованном секторе. Если успешно, отвечает ACK, иначе NACK6 (или 7). Если сектор неавторизован NACK8.

Код команды: 0x52.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина,	Данные

	<b>Байт</b>	
Адрес блока	1	
Данные блока	16	Нулевой байт идет первым.

Варианты ответа: Кадр ответа ACK / NACK.

#### 0x53 - “Запись блока 4 байт”

После этой команды считыватель пишет указанный блок (4 байт). Если успешно, отвечает ACK, иначе NACK6 (или 7). Если сектор неавторизован NACK8.

Код команды: 0x52.

Время исполнения команды: TBD мс.

Тело команды:

<b>Тело команды</b>	<b>Длина, Байт</b>	<b>Данные</b>
Адрес блока	1	
Данные блока	4	Нулевой байт идет первым.

Варианты ответа: Кадр ответа ACK / NACK.

#### 0x54 - “Increment”

После этой команды считыватель выполняет команду Increment для указанного value-блока в авторизованном секторе. Если успешно, отвечает ACK, иначе NACK6 (или 7). Если сектор неавторизован NACK8. Если ошибка value-значения, NACK9.

Код команды: 0x54.

Время исполнения команды: TBD мс.

Тело команды:

<b>Тело команды</b>	<b>Длина, Байт</b>	<b>Данные</b>
Адрес блока	1	
Данные команды	4	Величина инкремента.

Варианты ответа: Кадр ответа ACK / NACK.

#### 0x55 - “Decrement”

После этой команды считыватель выполняет команду Decrement для указанного value-блока в авторизованном секторе. Если успешно, отвечает ACK, иначе NACK6 (или 7). Если сектор неавторизован NACK8. Если ошибка value-значения, NACK9.

Код команды: 0x55.

Время исполнения команды: TBD мс.

Тело команды:

<b>Тело команды</b>	<b>Длина, Байт</b>	<b>Данные</b>
Адрес блока	1	
Данные команды	4	Величина декремента.

Варианты ответа: Кадр ответа ACK / NACK.

#### 0x56 - “Transfer”

После этой команды считыватель выполняет команду Transfer для указанного value-блока в авторизованном секторе. Если успешно, отвечает ACK, иначе NACK6 (или 7). Если сектор неавторизован NACK8. Если ошибка value-значения, NACK9.

Код команды: 0x56.

Время исполнения команды: TBD мс.

Тело команды:

<b>Тело команды</b>	<b>Длина, Байт</b>	<b>Данные</b>
Адрес блока	1	

Варианты ответа: Кадр ответа ACK / NACK.

#### 0x57 - “Restore”

После этой команды считыватель выполняет команду Restore для указанного value-блока в авторизованном секторе. Если успешно, отвечает ACK, иначе NACK6 (или 7). Если сектор неавторизован NACK8. Если ошибка value-значения, NACK9.

Код команды: 0x57.  
 Время исполнения команды: TBD мс.  
 Тело команды:

Тело команды	Длина, Байт	Данные
Адрес блока	1	

Варианты ответа: Кадр ответа ACK / NACK.

**0x5A - “Return losted sector”**

Команда предназначена для восстановления контроля над текущим сектором текущей карты в случае ошибки связи. Команда осуществляет прямой выбор последней активной карты по ее UID без выполнения процедуры антиколлизии и последующую авторизацию последнего авторизованного сектора с теми же ключами .

Код команды: 0x5A.  
 Время исполнения команды: TBD мс.  
 Тело команды: команда данных не имеет.  
 Варианты ответа: Кадр ответа ACK / NACK.  
 Выбирается всегда базовая скорость работы с картой.

Команда позволяет вернуть контроль над картой при ошибке связи. Считыватель не производит процедуру антиколлизии, а целенаправленно пытается выбрать карту с последним известным ему UID. Если успешно, производит авторизацию в том же секторе что и последняя команда авторизации с теми же параметрами.

Если карта не ответила, эта процедура повторяется автоматически один раз.

Внимание! Результат команды недетерминирован если последняя команда, содержащая антиколлизию, окончилась неудачей.

**0x5B - “Быстрое чтение карты”**

Команда предназначена для чтения множества секторов карты Mifare Standard 1K или 4K за одну команду. Считыватель читает все указанные сектора карты с одним ключом (заданным при последней авторизации). Все сектора передаются в порядке увеличения адреса. Чтение заканчивается или после прочтения всех заданных секторов или после первой же ошибки авторизации. Компьютер сам должен оценить корректность чтения по длине данных.

Обязательные условия работы команды: последняя команда с антиколлизией должна быть с этой картой, последняя авторизация должна содержать правильный ключ.

**ВНИМАНИЕ!** Команда присутствует не во всех моделях считывателей, см. «Заголовок устройства».

Код команды: 0x5B.  
 Время исполнения команды: TBD мс.  
 Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	
Флаги секторов	1...5	Если сектор нужно прочесть – отмечается единицей. Если старшие байты все нулевые – их можно не передавать.

Параметры команды: байт с битовыми флагами:

- Бит 0: Игнорировать трейлеры секторов.
- Бит 1: Игнорировать блок 0 сектора 0 (только если отмечен сектор 0).
- Биты 2...7: игнорируются.

Варианты ответа: данные длины кратной 16 байт (0...4096 байт).

Алгоритм работы с командой:

- Компьютер выбирает карту (Request + Anticollision + Select).
- Произойдет попытка авторизации с нужным ключом. Не обязательно успешная.
- Выполняется команда 0x5B. Проверяется длина полученных данных. Если она меньше требуемой, процедура повторяется сначала но уже без тех секторов что были успешно прочитаны. Считыватель не возвращает NACK при ошибках, просто прекращает чтение. Если карты в поле нет, возвращается кадр нулевой длины.



**0x5F - “Запись ключа карты в защищенную память”**

Эта команда записывает указанный ключ в указанную ячейку защищенной памяти считывателя.  
Теперь ключом можно только пользоваться при авторизации, узнать его значение невозможно.

Код команды: 0x5F.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Адрес ячейки	1	
Ключ	6	

Варианты ответа: Кадр ответа ACK / NACK.

## Команды карт формата ISO-15693

### 0x60 - “Команда ISO-15693”

Команда осуществляет передачу указанной последовательности карте и прием ответа.

Код команды: 0x60.

Время исполнения команды: TBD мс.

Тело команды:

Тело команды	Длина, Байт	Данные
Параметры команды	1	См. ниже.
Данные, передаваемые карте	XX	

Параметры команды: байт с битовыми флагами:

- Биты 0...3: время таймаута на исполнение команды. Выбирается из ряда 2, 4, 6, 8, 12, 16, 20, 26, 32, 42, 52, 64, 80, 100, 128, 160 мс (значения поля 0...0x0F).
- Бит 4: Глубина модуляции при передаче определяется параметром 0x21 (1) или 100% (0).
- Бит 5: передача EOF после таймаута на запись (величина таймаута - параметр 0x20). Прием ожидается в пределах этого таймаута на запись (если был принят валидный кадр то EOF не передается) и после EOF в течение таймаута указанного в команде. Согласно стандарта глубина модуляции EOF всегда 100%.
- Бит 6: Передача в формате 1 из 4 (1) или 1 из 256 (0).
- Бит 7: Передать только EOF. Данные на передачу карте игнорируются.

Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
Позиция бита в ответе карты, в котором произошла коллизия	1	Если коллизии не было, поле равно 0.
Ответ карты	XX	Если была коллизия, все биты после нее сбрасываются в ноль.

Ограничения команды.

Размер данных, передаваемых карте или принимаемых от карты, не может превышать максимальный размер одной транзакции с картой (см. команду «Заголовок устройства»).

Карте могут передаваться два вида пакетов:

- Стандартный пакет с данными. Всегда содержит первым байт флагов согласно ISO-15693 и вторым байт кода команды карты. Далее – данные по необходимости. Считыватель сам добавляет к ним CRC пакета.
- Разделитель (EOF). Передается в двух случаях (согласно ISO-15693):
  - При получении подтверждения факта записи в карту. В этом случае нужно выставить бит 5 в параметрах команды, соответственно вернется или немедленный отказ от выполнения команды либо подтверждение записи, полученное после EOF. Разумеется, EOF можно послать и вручную, отдельно.
  - При 16 slots Inventory. В этом случае после команды нужно послать еще до 15 кадров EOF.

Особенности работы отдельных команд ISO-15693.

Inventory. Поскольку это единственный законный способ поиска карты, то, соответственно только в этом режиме можно ожидать коллизий. В остальных командах факт коллизии нужно приравнять к ошибке.

## Примечания по алгоритмам работы

### Индикация по умолчанию

Зеленый светодиод вспыхивает при включении и при активности по сети.

Красный светодиод вспыхивает при включении.

Пищалка звучит при включении.

### Общие алгоритмы работы со считывателем

Поиск считывателя осуществляется командой 0x00 "Заголовок устройства".

- Считыватель считается найденным если в ответ пришли данные нужной длины и код ошибки 0.
- Если порт считывателя не открывается, продолжайте попытки открыть порт.
- Если порт открылся, но считыватель не дал распознанного ответа, рекомендуется провести поиск считывателя по всем возможным скоростям и при необходимости переустановить скорость на рабочую.

### Рекомендуемые алгоритмы работы с картами Mifare Standard

Общее замечание для всех команд: получение в ответ от считывателя сообщений NACK 6,7,8,9 – карта потеряна, начинаем работу с пункта 1.

Если код ошибки не подпадает под указанные в «Общие алгоритмы работы со считывателем» и в вариантах ответа на команду, то такой код ошибки считать случайным и повторить команду.

1. Поиск карты выполняется периодической подачей команды 0x45 "Request+Anticollision+Select" с параметром 0x00. Ответ с данными и кодом ошибки 0 означает успешное обнаружение карты. Можно проверить тип карты и, если это карты Mifare Standard, переходим к пункту 2.
2. Выполняем авторизацию в рабочем секторе. Для этого выполняем команду 0x50 "Авторизация" с соответствующими параметрами. Ответ с данными и кодом ошибки 0 означает успешную авторизацию. Переходим к пункту 3.
3. Теперь можно читать и писать в любые блоки в пределах сектора в котором авторизовались. Валидны команды 0x51 "Чтение блока 16 байт", 0x52 "Запись блока 16 байт", 0x54 "Increment", 0x55 "Decrement", 0x56 "Transfer" и 0x57 "Restore". Если необходимо сменить рабочий сектор, переходим к пункту 2.
4. После окончания работы с картой переводим ее в спящий режим командой 0x43 "HALT". Такая карта не будет реагировать на команду 0x45 "Request+Anticollision+Select" с параметром 0x00 пока ее не удалят из поля считывателя. Поэтому можно продолжить поиск новой карты с пункта 1.
5. Если вам все же захочется работать со «спящей» картой снова (не вынося ее из поля и не выключая накачки), подайте команду 0x45 "Request+Anticollision+Select" с параметром 0x80.

### Структура карт Mifare Standard

Вся информация в картах доступна порциями по 16 байт. Эта порция называется **блоком**. Блоки объединены в группы по несколько штук (по 4 блока или по 16, в зависимости от типа карты), эта группа блоков именуется **сектором**. Вся адресация данных в карте идет применительно к номерам блоков, адреса начинаются с 0. Внутри блока байты условно нумеруются с 0-го по 15-й.

Каждый сектор обладает уникальными условиями доступа к нему – набор паролей (ключей доступа) и параметров доступа. Эти ключи и параметры хранятся в блоке с самым старшим адресом внутри сектора (например, сектор с блоками 4..7 хранит свои параметры в блоке 7). Этот особый блок именуется **трейлером** сектора. Писать в трейлер нужно с четким пониманием что делаешь, в противном случае сектор полностью блокируется и восстановить доступ к нему невозможно.

Таким образом, вся карта оказывается поделена на множество маленьких независимых секторов, доступ к которым без знания ключей невозможен.

Блок с адресом 0 (Manufacturer block) имеет специальную функцию – в нем хранится производственная информация о карте. Переписать этот блок невозможно, он заблокирован на заводе.

Карта Mifare Standard 1K содержит 1кБайт данных, поделенных на 16 секторов по 4 блока в каждом. Карта Mifare Standard 4K содержит 4кБайт данных, поделенных на 40 секторов. Первые 32 сектора содержат по 4 блока в каждом, последние 8 – по 16 блоков. Различать карты Mifare Standard можно так:

```
If ((UID_Length == 4) && (SAK & (1<<3))) {
    // Это может быть Mifare Standard;
    If (SAK & (1<<4)) {
        // Это 4K
    } else {
        // Это 1K
    }
}
```

}  
}

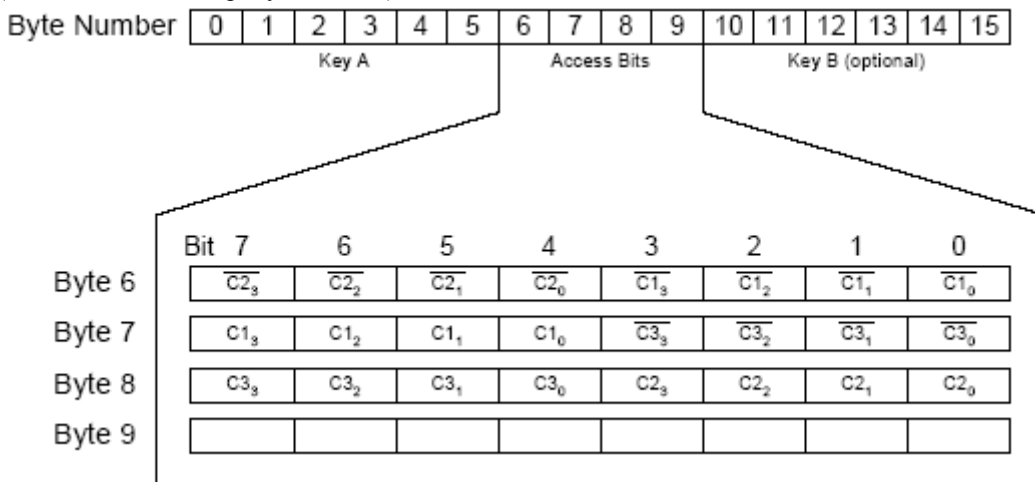
На рисунке ниже приведена структура карты Mifare Standard 1K. Следует учесть, что в протоколе все адреса – это номера блоков, получаются как (4\*Sector + Block). А чтобы обратиться к сектору (например, при авторизации) необходимо указать адрес любого блока в этом секторе.

Sector	Block	Byte Number within a Block														Description
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	
15	3	Key A				Access Bits				Key B						Sector Trailer 15
	2															Data
	1															Data
	0															Data
14	3	Key A				Access Bits				Key B						Sector Trailer 14
	2															Data
	1															Data
	0															Data
:	:															
:	:															
:	:															
1	3	Key A				Access Bits				Key B						Sector Trailer 1
	2															Data
	1															Data
	0															Data
0	3	Key A				Access Bits				Key B						Sector Trailer 0
	2															Data
	1															Data
	0															Manufacturer Block

**Структура трейлера**

Номер байта в трейлере	Назначение поля	Примечание
0...5	Ключ А	
6...9	Параметры доступа	
10...15	Ключ В	Необязателен

Параметры доступа – 4 битовых массива, определяющий индивидуально условия обращения с каждой из 4 областей данных (data area) сектора. Для секторов размером в 4 блока каждый битовый массив определяет доступ к каждому из блоков. Для секторов размером в 16 блоков первые три битовых массива определяют доступ к пяти блокам каждый (блоки 0...4, 5...9, 10...14), последний битовый массив определяет доступ к блоку трейлера. Каждый битовый массив состоит из 3 бит, хранится с проверочными кодами (см. пояснительные рисунки ниже).



Access Bits	Valid Commands			Description
C1 <sub>3</sub> C2 <sub>3</sub> C3 <sub>3</sub>	read, write	→	3	sector trailer
C1 <sub>2</sub> C2 <sub>2</sub> C3 <sub>2</sub>	read, write, increment, decrement, transfer, restore	→	2	data area <sup>1</sup>
C1 <sub>1</sub> C2 <sub>1</sub> C3 <sub>1</sub>	read, write, increment, decrement, transfer, restore	→	1	data area <sup>1</sup>
C1 <sub>0</sub> C2 <sub>0</sub> C3 <sub>0</sub>	read, write, increment, decrement, transfer, restore	→	0	data area <sup>1</sup>

На рисунке ниже указаны значения битов в массиве для data area.

Access bits			Access condition for				Application
C1	C2	C3	Read	write	increment	decrement, transfer, restore	
0	0	0	key A B <sup>1</sup>	Key A B <sup>1</sup>	key A B <sup>1</sup>	key A B <sup>1</sup>	transport configuration
0	1	0	key A B <sup>1</sup>	Never	never	never	read/write block
1	0	0	key A B <sup>1</sup>	Key B <sup>1</sup>	never	never	read/write block
1	1	0	key A B <sup>1</sup>	Key B <sup>1</sup>	key B <sup>1</sup>	key A B <sup>1</sup>	value block
0	0	1	key A B <sup>1</sup>	Never	never	key A B <sup>1</sup>	value block
0	1	1	key B <sup>1</sup>	Key B <sup>1</sup>	never	never	read/write block
1	0	1	key B <sup>1</sup>	Never	never	never	read/write block
1	1	1	never	Never	never	never	read/write block

На рисунке ниже указаны значения битов в массиве для trailer area.

Access bits			Access condition for						Remark
			KEYA		Access bits		KEYB		
C1	C2	C3	read	write	read	write	read	write	
0	0	0	never	key A	key A	never	key A	key A	Key B may be read
0	1	0	never	never	key A	never	key A	never	Key B may be read
1	0	0	never	key B	key A B	never	never	key B	
1	1	0	never	never	key A B	never	never	never	
0	0	1	never	key A	key A	key A	key A	key A	Key B may be read, transport configuration
0	1	1	never	key B	key A B	key B	never	key B	
1	0	1	never	never	key A B	key B	never	never	
1	1	1	never	never	key A B	never	never	never	

**Важно!** При чтении из трейлера те места где лежат ключи ВСЕГДА читаются как нули. Поэтому не стоит пытаться записать обратно то же что прочли.

Помни – если ошибся с правами доступа – сектор будет заблокирован и более недоступен!

### Структура и назначение Value блока

Value-блок – это любой блок кроме трейлеров, отформатированный специальным образом (см. рисунок ниже). Его назначение – хранить защищенную от посягательства «сумму на счету». Особенность этого блока в том, что эта «сумма» по одному ключу (ключу A) может быть только уменьшена, но не увеличена. Для увеличения суммы или перезаписи блока (командой Write) служит ключ B. Таким образом реализуются простейшие платежные схемы.

Для особо пытливых сообщу что значение Value – знаковое, и знак «минуса» накручиванием счета на уменьшение не удаляется.

Byte Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	Value				Value				Value				Adr	Adr	Adr	Adr

Здесь value – число, составляющее сумму счета (signed long). Adr – абстрактный идентификатор к этому счету, он не меняется при операциях с Value.

Для операций с Value есть специальные команды – Increment, Decrement, Transfer и Restore.

Increment, Decrement – увеличивают или уменьшают Value из соответствующего блока и помещают новое значение во внутренний вспомогательный регистр.

Transfer – Value из вспомогательного регистра записывают в указанный блок.

Restore – Value из соответствующего блока помещают во внутренний вспомогательный регистр. Эта команда позволяет копировать содержимое Value блока в пределах сектора.

Проверка «состояния счета» делается чтением блока.

Замечу что карта не гарантирует однозначности результата записи значения при каких-либо ошибках в обмене. Поэтому программист приложения обязан реализовывать простой механизм бэкапа с использованием двух блоков под одно Value значение.

## Замечания по реализации устройств

### Релиз 18.09.2003

Максимальный размер транзакции с картой для протокола ISO15693 64 байта. Изменен для протокола 0x00.0C.00.06 и последующих.

### Релиз 14.11.2004

В протокол 0x00.0C.00.06 добавлено понятие «максимальный размер одной транзакции с картой». Соответственно скорректирован максимальный размер транзакции с картой для протокола ISO15693 (старый размер транзакции 64 байта).

### Релиз 23.08.2005

В протокол 0x00.0C.00.07 добавлен признак «нет поддержки длительных команд». Сделано это вследствие почти полного отмирания интерфейса RS232 при работе под медленными громоздкими ОС.

Введено ручное переключение формата карт для многостандартных считывателей.

Добавлена новая команда – “Быстрое чтение карты”. В предыдущих моделях считывателей команда отсутствовала, используйте ее с осторожностью.

**ВНИМАНИЕ!** Был изменен алгоритм выхода из режима энергосбережения. Для выхода из режима энергосбережения теперь необходимо подать команду 0x22 – "Сброс накачки". Подобный алгоритм выхода из режима энергосбережения допустим и для старых версий считывателей поэтому рекомендуем использовать его всегда.

## Замеченные ошибки

### Ошибка записи во внутреннюю EEPROM

При записи во внутреннюю EEPROM (например, ключей по команде 0x5F) необходимо выдерживать паузу не менее 20 мс после подачи команды. Иначе запись не гарантируется. Изделия с DevVer до 0x00001110 включительно.

При записи ключей во внутреннюю EEPROM код ошибки 0x05 следует интерпретировать как успешную запись. Изделия с DevVer до 0x00001200 включительно.

### Ошибка авторизации из внутренней EEPROM

Авторизация по команде 0x50 из внутренней EEPROM для последней, 32-й ячейки, не работает. Пользуйтесь только ячейками с номерами 0-30. Изделия с DevVer до 0x00001110 включительно.

### Проблема выхода из спящего режима

По недобдуманному решению в процессе совершенствования изделий был изменен способ выхода из режима пониженного потребления. Если ранее считыватель автоматически выходил из спячки по любой команде, то теперь необходимо подать команду сброса накачки. Эта фишка вызвала большое количество нареканий со стороны пользователей и была нами отнесена к разряду ошибок. Посыпаем голову пеплом но таких считывателей выпущено очень много и необходимо с этим считаться.

## История

### История документа

Дата	Новый номер протокола	Примечание
18.09.2003	0x00.0C.00.04	Добавлены команды 0x46, 0x5A
20.01.2004	0x00.0C.00.05	Добавлены команды ISO15693
14.11.2004	0x00.0C.00.06	Добавлена команда 0x4A
23.08.2005	0x00.0C.00.07	Добавлен параметр в команду 0x22, бит 27 во флагах команды 0x00. Добавлены команды 0x5B, 0x4F.
01.12.2005	0x00.0C.00.08	Закрытые дополнения

В таблицу заносятся только изменения, приведшие к смене алгоритма протокола. Текстовые правки и дополнения изменяют только дату документа.

### История реализаций устройств

Device ID: 0x00.03.1C.02

Device revision history:

Дата	Версия устройства	Примечание
18.09.2003	0x00.00.11.08	Протокол 0x00.0C.00.04
20.01.2004	0x00.00.11.10	Протокол 0x00.0C.00.05
14.11.2004	0x00.00.11.11	Протокол 0x00.0C.00.06
23.08.2005	0x00.00.11.16	Протокол 0x00.0C.00.07
01.12.2005	0x00.00.12.00 0x00.00.12.01	Протокол 0x00.0C.00.08

В таблицу заносятся только версии, идущие в серию. Промежуточные реализации не учитываются. Устройство имело предшественников, не изготавливавшихся серийно и/или имевших другие протоколы.

Если в номерах версий устройства в таблице Device revision history имеются пропуски, то неуказанные версии имеют тот же протокол, что и самая младшая отмеченная версия.



## Контактная информация

### Техподдержка

Если у Вас возникли проблемы с документами, реализацией протокола, работой устройств – обращайтесь к нам.

ICQ: 463627544

Email: [undefined@mail.ru](mailto:undefined@mail.ru)

Телефон: +7 (812) 91-444-19

Официальные сайты: [www.prox.ru](http://www.prox.ru) [www.1st-mile.net](http://www.1st-mile.net)