Протокол обмена считывателя Mifare Plus

Параметры соединения

Считыватель Mifare Plus имеет следующие параметры соединения: скорость обмена по умолчанию 57600, 1 стоп бит, проверка на чётность. Скорость обмена может быть изменена, см. ниже.

Формат сообщений

Формат сообщений между ведущим устройством и считывателем:

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Стартовый байт, код 02h | 1 |
| 2 | Длина данных N. Учитывается только поле 3 | 1 |
| 3 | Данные | N |
| 4 | Контрольная сумма сообщения, от поля 2 до 3 включительно.  Вычисляется путём сложения байт по XOR. | 1 |

Минимальная длина сообщения 1 байт.

В сообщения от ведущего устройства первый байт сообщения содержит код команды.

По умолчанию в ответах считывателя первый байт содержит код ошибки. При включении режима ответа с кодом команды (см. команду F4h), первый байт код команды и второй байт код ошибки. Т.е. в ответ добавляется код команды, длина сообщения при этом увеличивается на 1 байт.

В полях сообщений содержащих значения разрядностью больше 1байта, первым всегда идёт младший байт.

Если код ошибки не равен MI\_OK (00h), то другие данные содержащиеся в ответе не валидны.

Терминология

PICC - карта Mifare Classic, Mifare Ultralight, Mifare Ultralight C, Mifare Plus или My-D Move.

PCD – считыватель.

PICC UID – уникальный номер карты. Возможны UID длиной в 4, 7, 10 байт.

AtqA - Answer to Request Type A. Ответ карты на запрос. Описание AtqA см. в ISO/IEC 14443-3 или AN10833 Mifare Type Identification Procedure.

SAK - Select Acknowledge. Ответ карты на выбор. Описание SAK см. в ISO/IEC 14443-3 или AN10833 Mifare Type Identification Procedure.

SL0 – режим работы чистой Mifare Plus карты (болванки). Нужен только для эмиссии карт.

SL1 – режим совместимости Mifare Plus карты с картами Mifare Classic.

SL2 – SL1 + авторизация по AES к карте. Не используется в новых картах Mifare Plus EV1.

SL3 – режим с авторизацией к карте по AES ключу, авторизацией по AES к блокам, подпись передаваемых данных, шифрование передаваемых данных. Возможно шифрование обмена с картой, цифровая подпись сообщений.

RATS (Request for Answer to Select) – команда запроса ATS у PICC и переключающая PICC на протокол ISO14443-4.

PPS (Protocol and Parameter Selection) – команда установления параметров обмена между PICC и PCD. Выполняется после RATS.

SAM AV2 модуль – модуль хранящий ключи для авторизации к PICC и для оффлайн криптографии.

SRI512 – интегральная микросхема бесконтактных карт от компании STMicroelectronics. Объём памяти 512 бит. Не совместима с картами Mifare по протоколу обмена. Используется протокол ISO/IEC 14443-3B и своя система команд.

SRI2K – интегральная микросхема бесконтактных карт от компании STMicroelectronics. Объём памяти 2048 бит. Не совместима с картами Mifare по протоколу обмена. Используется протокол ISO/IEC 14443-3B и своя система команд.

SRI4K – интегральная микросхема бесконтактных карт от компании STMicroelectronics. Объём памяти 4096 бит. Не совместима с картами Mifare по протоколу обмена. Используется протокол ISO/IEC 14443-3B и своя система команд.

SRT512 – интегральная микросхема бесконтактных карт от компании STMicroelectronics. Объём памяти 512 бит. Не совместима с картами Mifare по протоколу обмена. Используется протокол ISO/IEC 14443-3B и своя система команд.

SRT2K – интегральная микросхема бесконтактных карт от компании STMicroelectronics. Объём памяти 2048 бит. Не совместима с картами Mifare по протоколу обмена. Используется протокол ISO/IEC 14443-3B и своя система команд.

SRT4K – интегральная микросхема бесконтактных карт от компании STMicroelectronics. Объём памяти 4096 бит. Не совместима с картами Mifare по протоколу обмена. Используется протокол ISO/IEC 14443-3B и своя система команд.

ESMART – SAM модуль компании ISBC.

Команды протокола

В данном разделе перечислены основные команды работы со считывателем.

Список команд

[Команда 02h “Проверка связи” 8](#_Toc153486051)

[Команда 04h “Запрос карт” 9](#_Toc153486052)

[Команда 05h “Антиколлизия” 9](#_Toc153486053)

[Команда 06h “Выбор карты” 10](#_Toc153486054)

[Команда 07h “Активация карты в режиме ожидания” 11](#_Toc153486055)

[Команда 09h “Активация карты в режиме остановки” 12](#_Toc153486056)

[Команда 0Ah, 12h “Авторизация к блоку карты” 13](#_Toc153486057)

[Команда 0Bh “Запись ключа авторизации Crypto-1” 13](#_Toc153486058)

[Команда 0Ch “Авторизация к блоку карты по переданному ключу” 14](#_Toc153486059)

[Команда 0Dh “Чтение блока карты” 16](#_Toc153486060)

[Команда 0Eh “Запись блока карты” 16](#_Toc153486061)

[Команда 0Fh “Изменение блок значения с сохранением” 17](#_Toc153486062)

[Команда 10h “Изменение блок значения” 18](#_Toc153486063)

[Команда 11h “Чтение блок значения” 19](#_Toc153486064)

[Команда 13h “Останов выбранной карты” 19](#_Toc153486065)

[Команда 14h “Запись блок значение” 20](#_Toc153486066)

[Команда 15h “Запрос серийного номера считывателя” 20](#_Toc153486067)

[Команда 16h “Запрос описания считывателя” 21](#_Toc153486068)

[Команда 17h “Запрос версии RIC” 21](#_Toc153486069)

[Команда 19h “Команда DESELECT ISO14443-4” 22](#_Toc153486070)

[Команда 1Ah “Сброс радиополя” 22](#_Toc153486071)

[Команда 1Bh “Отправка APDU напрямую карте” 23](#_Toc153486072)

[Команда 1Сh “Переключение карты на протокол ISO14443-4” 24](#_Toc153486073)

[Команда 1Dh “Подтверждение персонализации” 25](#_Toc153486074)

[Команда 1Eh “Запись данных при персонализации” 25](#_Toc153486075)

[Команда 1Fh “Сброс считывателя” 26](#_Toc153486076)

[Команда 20h “Звуковой сигнал” 26](#_Toc153486077)

[Команда 21h “Чтение страницы с карты Ultraligth” 27](#_Toc153486078)

[Команда 22h “Запись страницы на карту Ultraligth” 27](#_Toc153486079)

[Команда 23h “Запись страницы на карту Ultraligth в режиме совместимости” 28](#_Toc153486080)

[Команда 24h “Запись 2K3DES ключа для карт Ultralight C в SAM AV2 модуль” 29](#_Toc153486081)

[Команда 25h “Авторизация к карте Ultralight C 2K3DES ключом” 29](#_Toc153486082)

[Команда 27h “Переключение слота SAM AV2 модуля” 30](#_Toc153486083)

[Команда 28h “Работа с картой на ИС SRI512” 31](#_Toc153486084)

[Команда 29h “Привязка записываемого ключа к ключу хоста” 33](#_Toc153486085)

[Команда 2Ah “Авторизация к SAM AV2 модулю по переданному ключу” 33](#_Toc153486086)

[Команда 2Bh “Запись AES ключа авторизации к SAM AV2 модулю” 34](#_Toc153486087)

[Команда 2Ch “Получение параметров счётчика использования” 35](#_Toc153486088)

[Команда 2Dh “Задание параметров счётчика использования” 36](#_Toc153486089)

[Команда 2Eh “Задание счётчика использования для записанного ключа” 37](#_Toc153486090)

[Команда 2Fh “Задание счётчика использования для ключа перед его записью” 38](#_Toc153486091)

[Команда 30h “Запрос информации производителя SAM AV2 модуля” 38](#_Toc153486092)

[Команда 31h “Запись ключей авторизации Crypto-1, A и B” 39](#_Toc153486093)

[Команда 32h “Авторизация к блоку карты” 40](#_Toc153486094)

[Команда 33h “Чтение или запись сектора с авторизацией” 41](#_Toc153486095)

[Команда 34h “Запись AES ключа авторизации к SAM AV2 модулю” 43](#_Toc153486096)

[Команда 35h “Чтение параметров ключей” 44](#_Toc153486097)

[Команда 36h “Получить полный серийный номер считывателя” 45](#_Toc153486098)

[Команда 37h “Включить привязку SAM AV2 модуля к считывателю” 46](#_Toc153486099)

[Команда 38h “Включить привязку SAM AV2 модуля к считывателю с заданным серийным номером” 47](#_Toc153486100)

[Команда 39h “Сброс авторизации к карте Mifare Plus” 47](#_Toc153486101)

[Команда 3Ah “Запись блока данных на карту Mifare Plus” 48](#_Toc153486102)

[Команда 3Bh “Чтение блока данных с карты Mifare Plus” 49](#_Toc153486103)

[Команда 3Ch “Запись блока значения на карту Mifare Plus” 50](#_Toc153486104)

[Команда 3Dh “Чтение блока значения с карты Mifare Plus” 51](#_Toc153486105)

[Команда 3Eh “Инкремент блока значения на карте Mifare Plus” 52](#_Toc153486106)

[Команда 3Fh “Декремент блока значения на карте Mifare Plus” 52](#_Toc153486107)

[Команда 40h “Сохранение блок значения на карте Mifare Plus” 53](#_Toc153486108)

[Команда 41h “Восстановление блок значения с карты Mifare Plus” 54](#_Toc153486109)

[Команда 42h “Инкремент блока значения на карте Mifare Plus с сохранением” 54](#_Toc153486110)

[Команда 43h “Декремент блока значения на карте Mifare Plus с сохранением” 55](#_Toc153486111)

[Команда 44h “Авторизация к карте Mifare Plus в режиме SL3” 56](#_Toc153486112)

[Команда 45h “Запись нескольких блоков данных на карту Mifare Plus” 57](#_Toc153486113)

[Команда 46h “Чтение нескольких блоков данных с карту Mifare Plus” 58](#_Toc153486114)

[Команда 47h “Авторизация к карте Mifare Plus в режиме SL1” 59](#_Toc153486115)

[Команда 48h “Запись AES128 ключа для карт Mifare Plus в SAM AV2 модуль” 60](#_Toc153486116)

[Команда 49h “Авторизация к сектору карты Mifare Plus в режиме SL2, ключом AES” 60](#_Toc153486117)

[Команда 4Ah “Авторизация к сектору карты Mifare Plus в режиме SL2, ключом Crypto-1” 61](#_Toc153486118)

[Команда 4Bh “Запись нескольких блоков данных на карту Mifare Plus” 62](#_Toc153486119)

[Команда 4Ch “Чтение нескольких блоков данных с карты Mifare Plus” 63](#_Toc153486120)

[Команда 50h “Чтение страницы с карты Ultraligth EV1” 64](#_Toc153486121)

[Команда 51h “Команда GetVersion карты Ultraligth EV1” 64](#_Toc153486122)

[Команда 52h “Быстрое чтение страниц с карты Ultraligth EV1” 65](#_Toc153486123)

[Команда 53h “Запись страницы на карту Ultraligth EV1” 66](#_Toc153486124)

[Команда 5Ch “Запись страницы на карту Ultraligth EV1 в режиме совместимости” 66](#_Toc153486125)

[Команда 5Fh “Чтение счётчика карты Ultraligth EV1” 67](#_Toc153486126)

[Команда 60h “Инкремент счётчика карты Ultraligth EV1” 67](#_Toc153486127)

[Команда 61h “Авторизация по паролю к карте Ultraligth EV1” 68](#_Toc153486128)

[Команда 62h “Чтение сигнатуры карты Ultralight EV1” 69](#_Toc153486129)

[Команда 63h “Проверка события Tearing карты Ultralight EV1” 69](#_Toc153486130)

[Команда 64h “Выполнение команды VCSL карты Ultralight EV1” 70](#_Toc153486131)

[Команда 65h “Отправка APDU напрямую в SAM модуль” 70](#_Toc153486132)

[Команда 66h “Команда работы с Master SAM” 71](#_Toc153486133)

[Команда 67h “Настройка параметров асинхронного интерфейса ведомого устройства” 73](#_Toc153486134)

[Команда 68h “Отправка данных ведомому устройству” 74](#_Toc153486135)

[Команда 69h “Приём данных от ведомого устройства” 75](#_Toc153486136)

[Команда 6Ah “Задать состояние линии DTR” 76](#_Toc153486137)

[Команда 6Bh “Запрос состояния линии DSR” 76](#_Toc153486138)

[Команда 6Ch “Запрос значения счётчика APDU” 77](#_Toc153486139)

[Команда 70h “Запись трейлера на карту из SAM AV2 модуля на уровне SL0” 77](#_Toc153486140)

[Команда 71h “Запись ключей AES на карту из SAM AV2 модуля” 79](#_Toc153486141)

[Команда 72h “Запись трейлера на карту из SAM AV2 модуля на уровне SL1” 79](#_Toc153486142)

[Команда 73h “Формирование криптограммы и запись на карту в режиме SL1” 81](#_Toc153486143)

[Команда 74h “Формирование криптограммы и запись на карту в режиме SL3” 82](#_Toc153486144)

[Команда 75h “Запись AES ключа Offline обновления ключей” 83](#_Toc153486145)

[Команда 76h “Низкоуровневый обмен с картой” 84](#_Toc153486146)

[Команда C0h “Генерация пары RSA ключей” 85](#_Toc153486147)

[Команда C1h “Экспорт приватного RSA ключа” 86](#_Toc153486148)

[Команда C2h “Экспорт публичного RSA ключа” 87](#_Toc153486149)

[Команда C3h “Чтение буфера считывателя” 88](#_Toc153486150)

[Команда C4h “Запись буфера считывателя” 89](#_Toc153486151)

[Команда C5h “Вычислить цифровую подпись” 89](#_Toc153486152)

[Команда C6h “Получить вычисленную цифровую подпись” 90](#_Toc153486153)

[Команда C7h “Проверка цифровой подписи” 91](#_Toc153486154)

[Команда C8h “Импорт RSA ключа” 92](#_Toc153486155)

[Команда С9h “Шифрование данных, используя Offline Crypto” 92](#_Toc153486156)

[Команда СAh “Дешифрование данных, используя Offline Crypto” 93](#_Toc153486157)

[Команда CBh “Вычисление хэш для данных в буфере” 94](#_Toc153486158)

[Команда CCh “Активация Offline Crypto ключа” 95](#_Toc153486159)

[Команда СDh “Вычисление MAC, используя Offline Crypto” 96](#_Toc153486160)

[Команда СEh “Верификация MAC, используя Offline Crypto” 96](#_Toc153486161)

[Команда СFh “Генерация случайной последовательности” 97](#_Toc153486162)

[Команда D0h “Чтение страниц с карты My-D Move” 98](#_Toc153486163)

[Команда D1h “Чтение двух страниц с карты My-D Move” 98](#_Toc153486164)

[Команда D2h “Запись страницы на карту My-D Move” 99](#_Toc153486165)

[Команда D3h “Запись двух страниц на карту My-D Move” 100](#_Toc153486166)

[Команда D4h “Запись страницы на карту My-D Move в режиме совместимости” 100](#_Toc153486167)

[Команда D5h “Запись пароля на карту My-D Move” 101](#_Toc153486168)

[Команда D6h “Авторизация по паролю к карте My-D Move” 101](#_Toc153486169)

[Команда D7h “Декремент счётчика карты My-D Move” 102](#_Toc153486170)

[Команда D8h “Вычисление хэш функции SHA256” 103](#_Toc153486171)

[Команда D9h “Вычисление хэш функции SHA224” 103](#_Toc153486172)

[Команда DAh “Вычисление хэш функции SHA-1” 104](#_Toc153486173)

[Команда DBh “Запись ключа для Offline Crypto в SAM AV2 модуль” 104](#_Toc153486174)

[Команда DCh “Активация Offline Crypto ключа” 106](#_Toc153486175)

[Команда DDh “Шифрование данных, используя Offline Crypto” 106](#_Toc153486176)

[Команда DEh “Дешифрование данных, используя Offline Crypto” 107](#_Toc153486177)

[Команда DFh “Деактивация авторизации к SAM AV2 модулю” 107](#_Toc153486178)

[Команда EFh “Управление хранением ключей считывателя” 108](#_Toc153486179)

[Команда F0h “Управление светодиодами и чтение состояния кнопки” 109](#_Toc153486180)

[Команда F1h “Переход в режим программирования” 111](#_Toc153486181)

[Команда F2h “Управление светодиодами” 111](#_Toc153486182)

[Команда F3h “Чтение состояния кнопки” 112](#_Toc153486183)

[Команда F4h “Включение режима ответа с кодом команды” 113](#_Toc153486184)

[Команда F5h “Установить скорость обмена” 114](#_Toc153486185)

[Команда F6h “Чтение настроек считывателя” 115](#_Toc153486186)

[Команда F7h “Запись настроек считывателя” 116](#_Toc153486187)

[Команда F8h “Запрос расширенного кода ошибки” 117](#_Toc153486188)

[Команда F9h “Запись ключа авторизации к SAM AV2 модулю в память считывателя” 118](#_Toc153486189)

[Команда FAh “Запись UID считывателя в память” 119](#_Toc153486190)

[Команда FBh “Чтение UID считывателя из памяти” 119](#_Toc153486191)

[Команда FCh “Ротация светодиодов” 120](#_Toc153486192)

[Команда FEh “Переключение антенного входа” 121](#_Toc153486193)

[Команда FFh “Снижение энергопотребления ” 121](#_Toc153486194)

[Команды протокола для работы с 3K5 122](#_Toc153486195)

[Команда E0h “Разрешение/запрет приёма карт” 122](#_Toc153486196)

[Команда E1h “Запросить статус считывателя” 123](#_Toc153486197)

[Команда E2h “Выдать карту” 123](#_Toc153486198)

[Команда E3h “Захватить карту” 124](#_Toc153486199)

[Команда E4h “Получить ответ на последнюю поданную команду” 124](#_Toc153486200)

[Команда E5h “Управление светодиодом” 125](#_Toc153486201)

[Команды протокола для работы с SCD2500 126](#_Toc153486202)

[Команда E0h “Разрешение/запрет приёма карт” 126](#_Toc153486203)

[Команда E1h “Запросить статус считывателя” 127](#_Toc153486204)

[Команда E2h “Выдать карту” 127](#_Toc153486205)

[Команда E3h “Захватить карту” 128](#_Toc153486206)

[Команда E4h “Получить ответ на последнюю поданную команду” 128](#_Toc153486207)

[Команда E5h “Запрос конфигурации” 129](#_Toc153486208)

[Команда E6h “Выдача карты из кассеты” 130](#_Toc153486209)

[Команда E7h “Включение или выключение цикла чистящей карты” 130](#_Toc153486210)

[Команда E8h “Управление блокировочным штифтом” 131](#_Toc153486211)

[Команда E9h “Сброс состояния вмешательства” 132](#_Toc153486212)

[Команда EAh “Сброс состояния залипания” 132](#_Toc153486213)

[Команда EBh “Прочитать информацию о функционировании” 133](#_Toc153486214)

[Команда ECh “Запрос состояния оптических датчиков” 134](#_Toc153486215)

[Команда EDh “Режим отчистки” 134](#_Toc153486216)

[Команды протокола для эмиссии карт 135](#_Toc153486217)

[Команда 26h “Запись зашифрованных AES128 ключом данных на карту” 135](#_Toc153486218)

[Команды эмуляции карт NFC Forum Type 2 Tag 136](#_Toc153486219)

[Команда FD00h “Включить режим эмуляции карты T2T” 136](#_Toc153486220)

[Команда FD01h “Выключить режим эмуляции карты T2T” 137](#_Toc153486221)

[Команда FD02h “Записать данные в память карты” 137](#_Toc153486222)

[Команда FD03h “Считать данные из памяти карты” 138](#_Toc153486223)

[Команды протокола версии считывателя для турникета 139](#_Toc153486224)

[Команда E0h “Разрешение/запрет приёма карт” 139](#_Toc153486225)

# Команда 02h “Проверка связи”

Проверка связи со считывателем.

Включает поле считывателя, если оно было выключено.

Сообщение к считывателю

Длина сообщения: 1/2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 02h | 1 |
| 1 | Опциональный параметр включения шифрования обмена.  Может не передаваться.  Значения: 0 – выключить шифрование, 1 – включить шифрование, остальные зарезервированы.  Если не передается, то шифрование будет выключено.  Алгоритм шифрования см. в Приложение 14. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 04h “Запрос карт”

Запрос карт находящихся в поле считывателя.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 04h | 1 |
| 2 | Параметр запроса.  PICC\_REQALL(52h) – запрос всех карт.  PICC\_REQIDL (26h) – запрос карт находящихся в режиме ожидания. | 1 |

Ответ от считывателя

Длина сообщения: 3 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Ответ карты Answer to Request Type A (AtqA).  Описание AtqA см. в ISO/IEC 14443-3. | 2 |

# Команда 05h “Антиколлизия”

Запрос номера PICC UID карты находящейся в поле считывателя.

Считыватель получает номер одной из приложенных карт и возвращает его в ответе.

Сообщение к считывателю

Длина сообщения: 7 - 15 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 05h | 1 |
| 2 | Параметр запроса. Код антиколлизии.  SELECT\_CASCADE\_LEVEL\_1(93h) – первый уровень.  SELECT\_CASCADE\_LEVEL\_2 (95h) – второй уровень.  SELECT\_CASCADE\_LEVEL\_3 (97h) – третий уровень. | 1 |
| 3 | Число известных бит UID | 1 |
| 4 | Массив, содержащий известные биты номера карты | 4 - 12 |

Ответ от считывателя

Длина сообщения: 5, 8 или 11 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | PICC UID – уникальный номер карты.  Содержит 4 байта UID для “Single Size UID PICC”, 7 байт UID для “Double Size UID PICC”, 10 байта UID для “Triple Size UID PICC” | 4, 7, 10 |

# Команда 06h “Выбор карты”

Выбор карты PICC с заданным номером.

Считыватель выбирает карту номер которой задан.

Сообщение к считывателю

Длина сообщения: 6 или 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 06h | 1 |
| 2 | Параметр запроса. Код антиколлизии.  SELECT\_CASCADE\_LEVEL\_1(93h) – первый уровень.  SELECT\_CASCADE\_LEVEL\_2 (95h) – второй уровень.  SELECT\_CASCADE\_LEVEL\_3 (97h) – третий уровень. | 1 |
| 3 | PICC UID – номер карты. | 4, 7 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | PICC UID – уникальный номер карты.  Содержит 4 байта UID для “Single Size UID PICC”, 7 байт UID для “Double Size UID PICC”, 10 байта UID для “Triple Size UID PICC” | 4, 7, 10 |
| 3 | Select Acknowledge (SAK). Ответ карты на выбор.  Описание SAK см. в ISO/IEC 14443-3. | 1 |

# Команда 07h “Активация карты в режиме ожидания”

Активация карты PICC находящейся в режиме ожидания.

Происходит запрос, получение (антиколлизия) номера и выбор карты находящейся в режиме ожидания.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 07h | 1 |
| 2 | Зарезервировано. Всегда 0. | 1 |

Ответ от считывателя

Длина сообщения: 9, 12, 14 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Ответ карты Answer to Request Type A (AtqA).  Описание AtqA см. в ISO/IEC 14443-3. | 2 |
| 3 | Select Acknowledge (SAK). Ответ карты на выбор.  Описание SAK см. в ISO/IEC 14443-3. | 1 |
| 4 | Длина номера карты. 4 байта UID для “Single Size UID PICC”, 7 байт UID для “Double Size UID PICC”, 10 байта UID для “Triple Size UID PICC” | 1 |
| 5 | PICC UID – уникальный номер карты. | 4, 7, 10 |

# Команда 09h “Активация карты в режиме остановки”

Активация карты PICC находящейся в режиме остановки.

Происходит запрос, получение (антиколлизия) номера и выбор карты находящейся в режиме ожидания.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 09h | 1 |
| 2 | Зарезервировано. Всегда 0. | 1 |

Ответ от считывателя

Длина сообщения: 9, 12, 14 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Ответ карты Answer to Request Type A (AtqA).  Описание AtqA см. в ISO/IEC 14443-3. | 2 |
| 3 | Select Acknowledge (SAK). Ответ карты на выбор.  Описание SAK см. в ISO/IEC 14443-3. | 1 |
| 4 | Длина номера карты. 4 байта UID для “Single Size UID PICC”, 7 байт UID для “Double Size UID PICC”, 10 байта UID для “Triple Size UID PICC” | 1 |
| 5 | PICC UID – уникальный номер карты. | 4, 7, 10 |

# Команда 0Ah, 12h “Авторизация к блоку карты”

Авторизация с блоку карты PICC по заданному ключу.

Происходит авторизация к карте по ключу Crypto-1. Используется для работы с картами Mifare Classic и Mifare Plus в режиме SL1.

Рекомендуется использовать команду 32h.

Сообщение к считывателю

Длина сообщения: 4 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 0Ah или 12h | 1 |
| 2 | Тип ключа.  Для ключа A значение 60h.  Для ключа B значение 61h.  Для ключа A, с диверсификацией в SAM AV2, значение E0h.  Для ключа B, с диверсификацией в SAM AV2, значение E1h. | 1 |
| 3 | Номер записи (Key Entry) с ключом, используемым для авторизации и хранящегося на SAM AV2 модуле.  Значение от 0 до 126.  Значение инкрементируется считывателем на 1. | 1 |
| 4 | Номер блока, к которому производится авторизация.  Этот же номер используется при диверсификации. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 0Bh “Запись ключа авторизации Crypto-1”

Реализована для обеспечения совместимости с предыдущими считывателями Mifare.

Записывает один ключ в SAM AV2 модуль, только A или только B.

Рекомендуется использовать команду 31h.

См. команду 66h, если нужно записать ключ с возможностью чтения.

Опционально можно задать параметры диверсификации. Ключ для диверсификации можно записать командой DBh с типом ключа 2 – 2K3DES.

Сообщение к считывателю

Длина сообщения: 9 байт или 11 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 0Bh | 1 |
| 2 | Тип ключа.  Для ключа A значение 60h.  Для ключа B значение 61h. | 1 |
| 3 | Номер записи (Key Entry) в которую будет записан ключ.  Значение от 0 до 126.  Значение инкрементируется считывателем на 1, что бы нельзя было переписать Master Key в ячейке 0. | 1 |
| 4 | Ключ | 6 |
| 5 | Опциональный параметр. Номер ячейки с ключом 3DES используемым для диверсификации. Номер инкрементирован на 1. | 1 |
| 6 | Опциональный параметр. Версия ключа 3DES используемого для диверсификации. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 0Ch “Авторизация к блоку карты по переданному ключу”

Авторизация с блоку карты PICC с использованием переданного от хоста ключа.

Происходит авторизация к карте по ключу Crypto-1. Используется для работы с картами Mifare Classic и Mifare Plus в режиме SL1.

**Не рекомендуемая к использованию команда, т.к. ключ передаётся в открытом виде.**

**Ключ в открытом виде передаётся контроллеру считывателя и от него к приёмопередатчику.**

После успешного выполнения этой команды включается шифрование обмена, поэтому команда авторизации 0Ah не будет работать. Т.к. после этой команды шифрование производит приёмопередатчик, а в команде 0Ah SAM AV2 модуль. Сбросить шифрование можно командой 07h или 09h.

Аналогично обратное. После 0Ah не работает 0Ch.

Сообщение к считывателю

Длина сообщения: 19, 22 или 25 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 0Ch | 1 |
| 2 | Тип ключа.  Для ключа A значение 60h.  Для ключа B значение 61h. | 1 |
| 3 | Номер блока, к которому производится авторизация. | 1 |
| 4 | Номер (UID) карты длиной 4, 7 или 10 байт. | 4, 7, 10 |
| 5 | Кодированный ключ.  Кодирование ключа:   1. Каждый байт ключа разбивается на 4 бита. 2. Эти 4 бита инвертируются и сдвигаются на 4 бита влево. 3. Результат складывается с исходными 4 битами. 4. Получается один байт кодированного ключа.   Пример: Исходный ключ 904D54F144D0. Кодированный ключ 69F0B42DA5B40FE1B4B42DF0. | 12 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 0Dh “Чтение блока карты”

Считывает 16 байт блока данных с карты.

Сообщение к считывателю

Длина сообщения: 4 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 0Dh | 1 |
| 2 | Не используется. Оставлено для совместимости. Всегда 0. | 1 |
| 3 | Номер читаемого блока. | 1 |
| 4 | Не используется. Оставлено для совместимости. Всегда 0. | 1 |

Ответ от считывателя

Длина сообщения: 17 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Данные считанные из заданного блока | 16 |

# Команда 0Eh “Запись блока карты”

Записывает 16 байт данных в заданный блок с карты.

Сообщение к считывателю

Длина сообщения: 20 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 0Eh | 1 |
| 2 | Код операции записи. Всегда A0h | 1 |
| 3 | Номер записываемого блока. | 1 |
| 4 | Число записываемых байт данных. Всегда 16. | 1 |
| 5 | Записываемые данные. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 0Fh “Изменение блок значения с сохранением”

Изменение блок значения (Value блока) с выполнением команды Transfer. Описание Value блока см. в Приложении 7.

Сообщение к считывателю

Длина сообщения: 8 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 0Fh | 1 |
| 2 | Код операции.  Декремент PICC\_DECREMENT (C0h). Декрементирует значение данных из блока источника и сохраняет его во временном регистре. Затем данные сохраняются в блоке получателе.  Инкремент PICC\_INCREMENT (0xC1) . Инкрементирует значение данных из блока источника и сохраняет его во временном регистре. Затем данные сохраняются в блоке получателе.  Восстановление значения во временном регистре PICC\_RESTORE (C2h). Происходит запись значения в регистр из блока источника. Затем данные сохраняются в блоке получателе.  Сохранение значения в Value блоке PICC\_TRANSFER (0xB0). Записывает значение из временного регистра в Value блоке получателя. | 1 |
| 3 | Номер блока источника. Из него считываются данные перед операцией.  Игнорируется для PICC\_TRANSFER. | 1 |
| 4 | Номер блока получателя. В него сохраняются данные. | 1 |
| 5 | Значение. | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 10h “Изменение блок значения”

Изменение блок значения (Value блока). Описание Value блока см. в Приложении 7.

Сообщение к считывателю

Длина сообщения: 8 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 10h | 1 |
| 2 | Код операции.  Декремент PICC\_DECREMENT (C0h). Декрементирует значение данных из блока источника и сохраняет его во временном регистре.  Инкремент PICC\_INCREMENT (C1) . Инкрементирует значение данных из блока источника и сохраняет его во временном регистре.  Восстановление значения во временном регистре PICC\_RESTORE (C2h). Происходит запись значения в регистр из блока источника.  Сохранение значения в Value блоке PICC\_TRANSFER (0xB0). Записывает значение из временного регистра в Value блок. | 1 |
| 3 | Для Increment, Decrement: Номер блока источника. Из него считываются данные перед операцией.  Для Restore: Номер блока источника. Из него считываются данные во временный регистр.  Для Transfer: Номер блока получателя. В него сохраняются данные. | 1 |
| 4 | Зарезервирован. Значение игнорируется. | 1 |
| 5 | Значение. | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 11h “Чтение блок значения”

Считывает 4 байта блок значения с карты и его адрес.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 11h | 1 |
| 2 | Номер читаемого блока. | 1 |

Ответ от считывателя

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Данные считанные из заданного блока | 4 |
| 3 | Адрес блока. См. описание Value Block у Mifare | 1 |

# Команда 13h “Останов выбранной карты”

Переводит выбранную карту в Halt состояние.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 13h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 14h “Запись блок значение”

Записывает 4 байта блок значения с карты и его адрес.

Сообщение к считывателю

Длина сообщения: 7 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 14h | 1 |
| 2 | Номер записываемого блока. | 1 |
| 3 | Записываемые данные. | 4 |
| 4 | Адрес блока | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 15h “Запрос серийного номера считывателя”

Возвращает 4 байта серийного номера считывателя.

Команда оставлена для совместимости с предыдущими считывателями Mifare.

См. новую команду 36h, которая возвращает 16 байт.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 15h | 1 |

Ответ от считывателя

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Серийный номер. | 4 |

# Команда 16h “Запрос описания считывателя”

Возвращает ASCII строку с названием считывателя и версий прошивки.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 16h | 1 |

Ответ от считывателя

Длина сообщения: различна для разных версий.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Описание |  |

# Команда 17h “Запрос версии RIC”

Оставлено для совместимости.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 17h | 1 |

Ответ от считывателя

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | RIC, всегда 11h, 02h, A7h, 10h, ABh | 5 |

# Команда 19h “Команда DESELECT ISO14443-4”

Переводит выбранную карту на уровне SL3 в Halt состояние.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 19h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 1Ah “Сброс радиополя”

Сброс радиополя на заданное время.

**Для безопасности рекомендуется всегда сбрасывать поле после работы с картой. Для сброса авторизации.**

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 1Ah | 1 |
| 2 | Время сброса в мс. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 1Bh “Отправка APDU напрямую карте”

Отправляет переданный APDU карте и отправляет обратно ответ от карты.

Сообщение к считывателю

Длина сообщения: 1 + N.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 1Bh | 1 |
| 2 | Запрос С-APDU | N |

Ответ от считывателя

Длина сообщения: 1 + N.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Ответ R-APDU | N |

# Команда 1Сh “Переключение карты на протокол ISO14443-4”

Выполняет команды RATS (Request for Answer to Select) и команду PPS (Protocol and Parameter Selection).

**Необходима для работы с картами в режиме SL2 и SL3, для смены режима, для выполнения команд персонализации.**

Карта перед выполнением команды должна быть активирована (выбрана).

Выполняется однократно после активации карты.

Сообщение к считывателю

Длина сообщения: 3 байт или 4 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 1Ch | 1 |
| 2 | Скорость передачи данных от PCD к PICC.  Значения:  PHPAL\_I14443P4A\_DATARATE\_106 (00h) – 106 кбит/сек.  PHPAL\_I14443P4A\_DATARATE\_212 (01h) – 212 кбит/сек.  PHPAL\_I14443P4A\_DATARATE\_424 (02h) – 424 кбит/сек.  PHPAL\_I14443P4A\_DATARATE\_848 (03h) – 848 кбит/сек. | 1 |
| 3 | Скорость передачи данных от PICC к PCD.  Значения:  PHPAL\_I14443P4A\_DATARATE\_106 (00h) – 106 кбит/сек.  PHPAL\_I14443P4A\_DATARATE\_212 (01h) – 212 кбит/сек.  PHPAL\_I14443P4A\_DATARATE\_424 (02h) – 424 кбит/сек.  PHPAL\_I14443P4A\_DATARATE\_848 (03h) – 848 кбит/сек. | 1 |
| 4 | Опциональный параметр. Флаг запроса ATS возвращённого картой.  0 - не возвращать, > 0 – вернуть ATS. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 1Dh “Подтверждение персонализации”

Переводит карту из режима SL0 в режим SL1.

**Перед выполнением команды необходимо записать на карту все ключи, биты доступа, настройки карты, начальные значения данных.**

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 1Dh | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 1Eh “Запись данных при персонализации”

Запись блоков данных при персонализации карты.

Блоки данных по 16 байт содержат ключи, настройки, начальные значения.

**При персонализации карты категорически нельзя оставлять ключи со значениями по умолчанию. Особенно это касается AES ключей**

Сообщение к считывателю

Длина сообщения: 19 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 1Eh | 1 |
| 2 | Адрес для записи блока данных. См. описание карты. | 2 |
| 3 | Данные. См. описание карты. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 1Fh “Сброс считывателя”

Сброс считывателя.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 1Fh | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 20h “Звуковой сигнал”

Включение звукового сигнала.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 20h | 1 |
| 2 | Номер проигрываемого звукового сигнала | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 21h “Чтение страницы с карты Ultraligth”

Считывает данные с карты Ultralight или Ultralight C.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 21h | 1 |
| 2 | Номер читаемой страницы | 1 |

Ответ от считывателя

Длина сообщения: 17 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | 16 байт считанных данных. 4 первых байта для заданной страницы.  12 байт со следующих 3-х страниц. | 16 |

# Команда 22h “Запись страницы на карту Ultraligth”

Записывает данные на карту Ultralight или Ultralight C.

Сообщение к считывателю

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 22h | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 23h “Запись страницы на карту Ultraligth в режиме совместимости”

Записывает данные на карту Ultralight или Ultralight C в режиме совместимости с Mifare Classic.

Сообщение к считывателю

Длина сообщения: 18 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 23h | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные. 4 первых байта должны содержать записываемые данные. 12 следующих байт игнорируются, в них рекомендуется записывать 0. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 24h “Запись 2K3DES ключа для карт Ultralight C в SAM AV2 модуль”

Записывает ключ авторизации к карте Ultralight в SAM AV2 модуль.

**В одной записи можно хранить 3 ключа с разными версиями, в разных позициях.**

См. команду 66h, если нужно записать ключ с возможностью чтения.

Сообщение к считывателю

Длина сообщения: 20 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 24h | 1 |
| 2 | Номер записи (Key Entry) в которую будет записан ключ.  Значение от 0 до 126.  Значение инкрементируется считывателем на 1, что бы нельзя было переписать Master Key в ячейке 0. | 1 |
| 3 | Позиция ключа. Значение 0 – 2. | 1 |
| 4 | Версия ключа. Значение 0 - 255. | 1 |
| 5 | Ключ 2K3DES. Два ключа по 8 байт. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 25h “Авторизация к карте Ultralight C 2K3DES ключом”

Производит авторизацию к карте Ultralight C 2K3DES ключом.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 25h | 1 |
| 2 | Номер записи (Key Entry) в которую записан ключ.  Значение от 0 до 126.  Значение инкрементируется считывателем на 1. | 1 |
| 3 | Версия используемого ключа. Значение 0 - 255. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 27h “Переключение слота SAM AV2 модуля”

Производит переключение слота SAM модуля. Если SAM AV2 нет в слоте или нет такого слота (номер больше 4), то переключение не произойдёт. Есть поддержка ESMART.

Сообщение к считывателю

Длина сообщения: 2 байта или 3 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 27h | 1 |
| 2 | Номер слота. Значение от 0 до 4.  Можно использовать значение выше 4, тогда переключение не произойдёт. Но будет возвращён текущий слот.  При отправке значения слота 27h не будет возвращена ошибка и будет возвращён текущий слот. | 1 |
| 3 | Опциональный байт.  При значении равном 0x01 происходит переключение на компоненты работы без SAM модуля. | 1 |

Ответ от считывателя

Длина сообщения: 7 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Текущий слот. Значение от 0 до 4. Либо 0xFF если работа без SAM модуля. См. опциональный параметр команды.  Если нет ошибки, то вернёт номер слота, который был задан в команде.  Если в слоте нет SAM AV2 модуля или нет такого слота, то вернёт текущий слот. | 1 |
| 3 | Статус слота 0. Значение 0 – нет SAM AV2 модуля в слоте или не удалось авторизоваться, 255 – есть SAM AV2 в слоте, 1 – не известный SAM модуль в слоте, 2 - ESMART. | 1 |
| 4 | Статус слота 1. Значение 0 – нет SAM AV2 модуля в слоте или не удалось авторизоваться, 255 – есть SAM AV2 в слоте, 1 – не известный SAM модуль в слоте, 2 - ESMART. | 1 |
| 5 | Статус слота 2. Значение 0 – нет SAM AV2 модуля в слоте или не удалось авторизоваться, 255 – есть SAM AV2 в слоте, 1 – не известный SAM модуль в слоте, 2 - ESMART. | 1 |
| 6 | Статус слота 3. Значение 0 – нет SAM AV2 модуля в слоте или не удалось авторизоваться, 255 – есть SAM AV2 в слоте, 1 – не известный SAM модуль в слоте, 2 - ESMART. | 1 |
| 7 | Статус слота 4. Значение 0 – нет SAM AV2 модуля в слоте или не удалось авторизоваться, 255 – есть SAM AV2 в слоте, 1 – не известный SAM модуль в слоте, 2 - ESMART. Этот слот обычно отсутствует. | 1 |

# Команда 28h “Работа с картой на ИС SRI512”

Реализует функционал работы с картами на основе ИС SRI512, SRI2K, SRI4K, SRT512, SRT2K, SRT4K.

**Запись на карту с этими ИС необходимо проверять командой чтения. Карта на команду записи не возвращает ответ. Код ошибки будет возвращён в случае, если карты нет или происходит обращение к несуществующему блоку карты.**

Сообщение к считывателю

Длина сообщения: 2 байт для активации или остановки, и 4 байта для чтения, и 4 + 4\*N для записи.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 28h | 1 |
|  | Код операции.  0x01 активация карты. Антиколлизия до первой карты.  0x02 для чтения заданного (N) числа блоков.  0x03 для записи заданного (N) числа блоков.  0x04 остановка карты. Команда Completion из протокола карты. | 1 |
| 3 | Адрес первого читаемого или записываемого блока.  Присутствует только для операций чтения или записи. | 1 |
| 4 | Число читаемых или записываемых блоков N.  Минимальное значение равно 1, максимальное равно 60.  Присутствует только для операций чтения или записи. | 1 |
| 5 | Записываемые в карту данные.  Размерность 4\*N.  Один блок карты имеет размер 4 байта.  Присутствует только для операции записи. | 1 |

Ответ от считывателя

Длина сообщения: 9 байт для активации, 1 байт для остановки или записи, 1 + N\*4 байт для чтения.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | UID карты для операции активации. Присутствует только для операции активации. Длина UID равна 8 байт.  Данные с карты. Присутствует только для операций чтения.  Один блок карты имеет размер 4 байта. | 8/N\*4 |

# Команда 29h “Привязка записываемого ключа к ключу хоста”

Позволяет задать ключу, перед его записью в SAM AV2 модуль, параметр его привязки к другому ключу. После записи ключа эта настройка сбрасывается в 0.

Длина сообщения: 4 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 29h | 1 |
| 2 | Тип ключа: 0 –привязка к ключу по умолчанию (мастер ключу) 0, > 0 – привязка к заданному ключу. | 1 |
| 3 | Номер ключа, значение от 0 до 127. Здесь нет смещения на +1. | 1 |
| 4 | Версия ключа, значение от 0 до 255. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 2Ah “Авторизация к SAM AV2 модулю по переданному ключу”

Производит авторизацию к SAM AV2 модулю по переданному ключу.

**В данной команде нет инкремента номера ключа в считывателе. Переданный номер реально соответствует номеру ячейки в SAM AV2 модуле. За счёт этого доступна ячейка 0 содержащая Master Key.**

Есть поддержка ESMART. Считыватель проверяет тип SAM модуля в текущем слоте.

Сообщение к считывателю

Длина сообщения: 20 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 2Ah | 1 |
| 2 | Тип ключа:  0 – ключ из памяти считывателя (нули, записанный в EEPROM, или генерируемый по серийным номерам считывателя и SAM модуля). Остальные параметры команды игнорируются, номер ячейки и версия будут взяты 0. Для SAM AV2.  1 – ключ, переданный в команде, номер ячейки и версия из параметров команды. Для SAM AV2.  2 – ключ из памяти считывателя, номер ячейки и версия из параметров команды. Для SAM AV2.  3 – повторяет параметр 0, но добавлено переключение указателей на работу SAM. Для SAM AV2.  4 – повторяет параметр 1, но добавлено переключение указателей на работу SAM. Для SAM AV2.  5 – повторяет параметр 2, но добавлено переключение указателей на работу SAM. Для SAM AV2.  6 – авторизация к модулю ESMART. Ключ, адрес и версия ключа передаются в параметрах команды. | 1 |
| 3 | Для SAM AV2 номер ячейки SAM модуля. От 0 до 127. Здесь не смещения номера на +1.  Для ESMART адрес ключа. | 1 |
| 4 | Версия ключа в позиции. | 1 |
| 5 | Ключ авторизации.  Для SAM AV2 размер ключа 16 байт.  Для ESMART размер ключа 32 байта. | 16/32 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 2Bh “Запись AES ключа авторизации к SAM AV2 модулю”

Записывает в SAM AV2 модуль ключ для авторизации к нему.

После записи ключа, требуется авторизация к нему. См. команду 2Ah или можно использовать сброс считывателя 1Fh.

Используется для персонализации SAM AV2 модуля. Считыватель перед началом работы должен авторизоваться к SAM AV2 модулю этим ключом.

**В данной команде нет инкремента номера ключа в считывателе. Переданный номер реально соответствует номеру ячейки в SAM AV2 модуле. За счёт этого доступна ячейка 0 содержащая Master Key.**

**Необходимо внимательно отнесись к записи этого ключа. Ошибка приведёт к невозможности использовать модуль.**

См. команду 66h, если нужно записать ключ с возможностью чтения.

Сообщение к считывателю

Длина сообщения: 22 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 2Bh | 1 |
| 2 | Номер ячейки SAM модуля. От 0 до 127. Здесь нет смещения номера на +1. | 1 |
| 3 | Номер позиции ключа в ячейке. От 0 до 2. | 1 |
| 4 | Версия ключа в позиции. | 1 |
| 5 | Ключ авторизации. | 16 |
| 6 | CRC ключа авторизации. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 2Ch “Получение параметров счётчика использования”

Команда позволяет получить параметры счётчика использования.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 2Ch | 1 |
| 2 | Номер счётчика. Значение от 0 до 15.  Младший байт первый. | 2 |

Ответ от считывателя

Длина сообщения: 13 байт или 1 байт при ошибке.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Предельное (порог) значение счётчика.  Младший байт первый. | 4 |
| 3 | Текущее значение счётчика.  Младший байт первый. | 4 |
| 4 | Номер ключа, к которому нужна авторизация для изменения этого счётчика.  Младший байт первый. | 2 |
| 5 | Версия ключа, к которому нужна авторизация для изменения этого счётчика.  Младший байт первый. | 2 |

# Команда 2Dh “Задание параметров счётчика использования”

Команда задаёт параметры самого счётчика использования.

Сообщение к считывателю

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 2Dh | 1 |
| 2 | Номер счётчика. Значение от 0 до 15.  Младший байт первый. | 2 |
| 3 | Предельное (порог) значение счётчика.  Младший байт первый. | 4 |
| 4 | Номер ключа изменения счётчика, значение от 0 до 127. Здесь нет смещения +1. К этому ключу нужно будет авторизоваться для изменения значения счётчика.  Значение от 0x00 до 0xFF.  Значение 0xFF – блокирует любое дальнейшее изменение счётчика. | 1 |
| 5 | Версия ключа изменения счётчика, значение от 0 до 255. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 2Eh “Задание счётчика использования для записанного ключа”

Задаёт параметр счётчика использования для ключа, который уже был записан. После записи ключа эта настройка сбрасывается в 0xFF.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 2Eh | 1 |
| 1 | Номер ключа без смещения в +1. | 1 |
| 2 | Номер счётчика. Значение от 0 до 15 или 0xFF.  При значении 0xFF счётчик не используется ключом. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 2Fh “Задание счётчика использования для ключа перед его записью”

Задаёт параметр счётчика использования для ключа перед его записью. После записи ключа эта настройка сбрасывается в 0xFF.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 2Fh | 1 |
| 2 | Номер счётчика. Значение от 0 до 15 или 0xFF.  При значении 0xFF счётчик не используется ключом. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 30h “Запрос информации производителя SAM AV2 модуля”

Запрашивает у SAM AV2 модуля информацию производителя и возвращает ведущему устройству. Информация производителя содержит: ID производителя (04h для NXP), тип, подтип, версию, размер памяти, тип протокола (01h для T1), уникальный серийный номер, номер продукта, дата производства. Более подробно см. в описании SAM AV2 модуля.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 30h | 1 |

Ответ от считывателя

Длина сообщения: зависит от SAM AV2 модуля, но на момент описания протокола 32 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Информация производителя | X, 31 |

# Команда 31h “Запись ключей авторизации Crypto-1, A и B”

Одновременно записывает пару ключей A и B в SAM AV2 модуль.

**В одной записи можно хранить 3 пары ключей с разными версиями, в разных позициях.**

См. команду 66h, если нужно записать ключ с возможностью чтения.

Опционально можно задать параметры диверсификации. Ключ для диверсификации можно записать командой DBh с типом ключа 2 – 2K3DES.

Сообщение к считывателю

Длина сообщения: 16 байт или 20 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 31h | 1 |
| 2 | Номер записи (Key Entry) в которую будет записана пара ключей.  Значение от 0 до 126.  Значение инкрементируется считывателем на 1, что бы нельзя было переписать Master Key в ячейке 0. | 1 |
| 3 | Позиция ключей для записи | 1 |
| 4 | Версия ключей | 1 |
| 5 | Ключ A | 6 |
| 6 | Ключ B | 6 |
| 7 | Опциональный параметр. Номер ячейки с ключом 3DES используемым для диверсификации ключа A. Номер инкрементирован на 1. | 1 |
| 8 | Опциональный параметр. Версия ключа 3DES используемого для диверсификации ключа A. | 1 |
| 9 | Опциональный параметр. Номер ячейки с ключом 3DES используемым для диверсификации ключа B. Номер инкрементирован на 1. | 1 |
| 10 | Опциональный параметр. Версия ключа 3DES используемого для диверсификации ключа B. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 32h “Авторизация к блоку карты”

Авторизация к блоку карты PICC по заданному ключу.

Происходит авторизация к карте по ключу Crypto-1. Используется для работы с картами Mifare Classic и Mifare Plus в режиме SL1.

Сообщение к считывателю

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 32h | 1 |
| 2 | Тип ключа.  Для ключа A значение 60h.  Для ключа B значение 61h.  Для ключа A, с диверсификацией в SAM AV2, значение E0h.  Для ключа B, с диверсификацией в SAM AV2, значение E1h. | 1 |
| 3 | Номер записи (Key Entry) с ключом, используемым для авторизации и хранящегося на SAM AV2 модуле.  Значение от 0 до 126. Номер инкрементируется считывателем на 1. | 1 |
| 4 | Версия используемого ключа. | 1 |
| 5 | Номер блока, к которому производится авторизация.  Этот же номер используется при диверсификации. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 33h “Чтение или запись сектора с авторизацией”

Команда чтения или записи 3-х блоков сектора с предварительной авторизацией к сектору.

**При выполнении операции записи, адрес блока должен выбираться так, что бы не перезаписывать трейлер карты.**

Сообщение к считывателю

Длина сообщения: 7 байт для чтения и 55/71/22/11/7+N\*16 байта для записи.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 33h | 1 |
| 2 | Код операции.  0x0D для чтения 3-х блоков.  0x1D для чтения 4-х блоков.  0x2D для чтения 1-го блока.  0x0E для записи 3-х блоков.  0x1E для записи 4-х блоков.  0x2E для записи 1 блока.  0x3E для записи 1 страницы Ultralight.  0xAN для чтения N + 1 блоков. Младшие 4 бита задают число блоков N.  0xBN для записи N + 1 блоков. Младшие 4 бита задают число блоков N.  В версии прошивки для считывателя турникета только младшие 7 бит содержат код операции.  Старший бит 7-й содержит флаг необходимости блокировки второго УКПД. Если этот бит выставлен в 1, то блокировка включена.  Выключение блокировки происходит по таймеру через 300 мс или по команде активации от первого УКПД.  При включенной блокировке второй УКПД на все запросы получит ответ “Нет карты”.  При включенной блокировке коды команд будут соответственно: 0x8D, 0x9D, 0x8E, 0x9E, 0xAE, 0xBE. | 1 |
| 3 | Необходимость авторизации.  0 – не производить авторизацию перед чтением или запись  >0 – производить авторизацию перед чтением или записью | 1 |
| 4 | Тип ключа.  Для ключа A значение 60h.  Для ключа B значение 61h.  Для ключа A, с диверсификацией в SAM AV2, значение E0h.  Для ключа B, с диверсификацией в SAM AV2, значение E1h. | 1 |
| 5 | Номер записи (Key Entry) с ключом, используемым для авторизации и хранящегося на SAM AV2 модуле.  Значение от 0 до 126. Номер инкрементируется считывателем на 1. | 1 |
| 6 | Версия используемого ключа. | 1 |
| 7 | Номер блока, к которому производится авторизация.  Начиная с этого блока, будет происходить чтение или запись данных.  Этот же номер используется при диверсификации. | 1 |
| 8 | Данные для записи в блоки сектора, или 4 блока, или один блок, или одна страница, или N блоков. Присутствует только для операции записи. | 48/64/  16/4/N\*16 |

Ответ от считывателя

Длина сообщения: 1 байт для ошибок, 48/64/N\*16 байт для чтения и 1 байт для записи.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Считанные данные 3-х, 4-х блоков или N блоков. Присутствует только для операции чтения при отсутствии ошибок. | 48/64/N\*16 |

# Команда 34h “Запись AES ключа авторизации к SAM AV2 модулю”

Записывает в SAM AV2 модуль ключ для авторизации к нему.

Ключ записывается в ячейку 0 SAM модуля, позицию 0, с версией 0. Если требуется иное, то см. команду 2Bh.

После записи ключа, требуется авторизация к нему. См. команду 2Ah или можно использовать сброс считывателя 1Fh.

Используется для персонализации SAM AV2 модуля. Считыватель перед началом работы должен авторизоваться к SAM AV2 модулю этим ключом.

**Необходимо внимательно отнесись к записи этого ключа. Ошибка приведёт к невозможности использовать модуль.**

Сообщение к считывателю

Длина сообщения: 19 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 34h | 1 |
| 2 | Ключ авторизации. | 16 |
| 3 | CRC ключа авторизации. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 35h “Чтение параметров ключей”

Читает параметры ключей (тип и версию) в ключевой записи.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 35h | 1 |
| 2 | Номер записи (Key Entry) с ключами.  Значение от 0 до 127.  Здесь нет инкремента ключа на +1. | 1 |

Ответ от считывателя

Длина сообщения: 20 байт или 1 байт при ошибке.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Тип ключей.  00h – AES128.  01h – AES192.  02h – AES256.  03h – DES.  04h – 2K3DES.  05h – 3K3DES.  06h – MIFARE (Crypto-1). | 2 |
| 3 | Длина списка версий ключей. Содержит число ключей в записи и сообщает сколько полей версий ключей заполнено.  Значение от 0 до 3.  3 – для Crypto-1, AES128, 2K3DES. | 2 |
| 4 | Версия ключа в позиции 0. | 2 |
| 5 | Версия ключа в позиции 1. | 2 |
| 6 | Версия ключа в позиции 2. | 2 |
| 7 | Номер ключа, к которому нужна авторизация для изменения этого ключа.  Младший байт первый. | 2 |
| 8 | Версия ключа, к которому нужна авторизация для изменения этого ключа.  Младший байт первый. | 2 |
| 9 | Номер счётчика использования связанного с этим ключом.  Значение FFh означает, что никакой счётчик использования не привязан.  Младший байт первый. | 2 |
| 10 | Биты SET ключа. Младший байт первый. См. документацию на SAM AV2. Биты задающие тип ключа не возвращаются. | 2 |
| 11 | Биты ExtSET ключа. См. документацию на SAM AV2. | 1 |

# Команда 36h “Получить полный серийный номер считывателя”

Возвращает полный серийный номер считывателя. Может использоваться для привязки SAM AV2 модуля к считывателю.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 36h | 1 |

Ответ от считывателя

Длина сообщения: 17 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Полный серийный номер. | 16 |

# Команда 37h “Включить привязку SAM AV2 модуля к считывателю”

Привязывает SAM AV2 модуль установленный в считыватель к этому считывателю.

При этом в SAM AV2 модуль записывается ключ авторизации полученный на основе серийного номера модуля и серийного номера считывателя.

В дальнейшем этот SAM AV2 модуль можно будет использовать только в этом считывателе. Либо нужно будет перезаписать этот ключ.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 37h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 38h “Включить привязку SAM AV2 модуля к считывателю с заданным серийным номером”

Привязывает SAM AV2 модуль, установленный в считыватель к считывателю, серийный номер которого задается в команде.

При этом в SAM AV2 модуль записывается ключ авторизации полученный на основе серийного номера модуля и серийного номера считывателя.

Используется для выпуска SAM AV2 модулей.

Сообщение к считывателю

Длина сообщения: 19 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 38h | 1 |
| 2 | Серийный номер считывателя | 16 |
| 3 | CRC серийного номера. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 39h “Сброс авторизации к карте Mifare Plus”

Сбрасывает авторизацию к карте. Используется при необходимости авторизации другим ключом.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 39h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 3Ah “Запись блока данных на карту Mifare Plus”

Записывает блок данных на карту Mifare Plus, находящуюся в режиме SL2 или SL3.

Есть поддержка ESMART. Считыватель проверяет тип SAM модуля в текущем слоте. На данный момент ESMART не поддерживает подпись ответа и шифрование. Эти параметры должны быть равны 0.

Сообщение к считывателю

Длина сообщения: 21 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 3Ah | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена с картой, шифрование  1 – включено.  0 – выключено. | 1 |
| 4 | Настройка обмена с картой, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 5 | Записываемые данные. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 3Bh “Чтение блока данных с карты Mifare Plus”

Читает блок данных с карты Mifare Plus находящейся в режиме SL2 или SL3.

Есть поддержка ESMART. Считыватель проверяет тип SAM модуля в текущем слоте. На данный момент ESMART не поддерживает шифрование и подпись ответа. Эти параметры должны быть равны 0.

Сообщение к считывателю

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 3Bh | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена с картой, шифрование  1 – включено.  0 – выключено. | 1 |
| 4 | Настройка обмена с картой, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 5 | Настройка обмена с картой, подпись команды  1 – включено.  0 – выключено. | 1 |

Ответ от считывателя

Длина сообщения: 17 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Прочитанные данные. | 16 |

# Команда 3Ch “Запись блока значения на карту Mifare Plus”

Записывает блок значение (value блок) на карту Mifare Plus, находящуюся в режиме SL2 или SL3.

Сообщение к считывателю

Длина сообщения: 10 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 3Ch | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена, шифрование  1 – включено.  0 – выключено. | 1 |
| 4 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 5 | Записываемые данные. | 4 |
| 6 | Адрес Value блока. См. описание Value блока. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 3Dh “Чтение блока значения с карты Mifare Plus”

Читает блок значение (value блок) с карты Mifare Plus, находящейся в режиме SL2 или SL3.

Сообщение к считывателю

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 3Dh | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена, шифрование  1 – включено.  0 – выключено. | 1 |
| 4 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 5 | Настройка обмена, подпись команды  1 – включено.  0 – выключено. | 1 |

Ответ от считывателя

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Прочитанные данные. Значение Value блока | 4 |
| 3 | Адрес Value блока. См. описание Value блока. | 1 |

# Команда 3Eh “Инкремент блока значения на карте Mifare Plus”

Инкрементирует блок значение (value блок) на карте Mifare Plus, находящейся в режиме SL2 или SL3. Результат сохраняется во временном регистре.

Сообщение к считывателю

Длина сообщения: 8 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 3Eh | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 4 | Величина инкремента. | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 3Fh “Декремент блока значения на карте Mifare Plus”

Декрементирует блок значение (value блок) на карте Mifare Plus, находящейся в режиме SL2 или SL3. Результат сохраняется во временном регистре.

Сообщение к считывателю

Длина сообщения: 8 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 3Fh | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 4 | Величина декремента. | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 40h “Сохранение блок значения на карте Mifare Plus”

Пересылает данные из временного регистра в блок значение.

Сообщение к считывателю

Длина сообщения: 4 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 40h | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 41h “Восстановление блок значения с карты Mifare Plus”

Пересылает данные из блок значения во временный регистр.

Сообщение к считывателю

Длина сообщения: 4 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 41h | 1 |
| 2 | Номер блока. | 2 |
| 3 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 42h “Инкремент блока значения на карте Mifare Plus с сохранением”

Инкрементирует блок значение (value блок) на карте Mifare Plus, находящейся в режиме SL2 или SL3. Результат сохраняется в том же блоке.

Сообщение к считывателю

Длина сообщения: 10 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 42h | 1 |
| 2 | Номер блока источника. Из него берётся значения для инкремента. | 2 |
| 3 | Номер блока назначения. В него сохраняется значение после инкремента. | 2 |
| 4 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 5 | Величина инкремента. | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 43h “Декремент блока значения на карте Mifare Plus с сохранением”

Декрементирует блок значение (value блок) на карте Mifare Plus, находящейся в режиме SL2 или SL3. Результат сохраняется в том же блоке.

Сообщение к считывателю

Длина сообщения: 10 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 43h | 1 |
| 2 | Номер блока источника. Из него берётся значения для инкремента. | 2 |
| 3 | Номер блока назначения. В него сохраняется значение после инкремента. | 2 |
| 4 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 5 | Величина декремента. | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 44h “Авторизация к карте Mifare Plus в режиме SL3”

Производит авторизация к карте Mifare Plus, находящуюся в режиме SL3, по ключу AES128, находящемуся в SAM AV2 модуле.

Есть поддержка ESMART. Считыватель проверяет тип SAM модуля в текущем слоте. Поддерживается только FOLLOWING\_AUTH, данные для диверсификации не передаются, номер ключа не инкрементируется.

Сообщение к считывателю

Длина сообщения: 6 байт или от 7 до 37. Если не используется диверсификация ключа, то длина сообщения 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 44h | 1 |
| 2 | Тип авторизации.  Для SAM AV2: FOLLOWING\_AUTH (00h) – повторная авторизация, FIRST\_AUTH (01h) – первичная авторизация.  Для ESMART: 00h – повторная авторизация без диверсификации ключа, 01h – первичная без диверсификации, 02h – повторная с диверсификацией, 03h – первичная с диверсификацией. | 1 |
| 3 | Номер блока. | 2 |
| 4 | Номер ключа в SAM AV2 модуле. Номер инкрементируется считывателем на 1.  Для ESMART номер ключа не инкрементируется. | 1 |
| 5 | Версия ключа в SAM AV2 модуле. | 1 |
| 6 | Данные для диверсификации ключа. Если диверсификация не нужна, то длина данного поля равна 0. Т.е. оно отсутствует.  Для ESMART диверсификация задаётся в типе авторизации. Это поле игнорируется. | 0 - 31 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 45h “Запись нескольких блоков данных на карту Mifare Plus”

Записывает несколько блоков данных на карту Mifare Plus, находящуюся в режиме SL2 или SL3. Ключ авторизации для записываемых блоков должен быть или один или ключи должны совпадать.

Сообщение к считывателю

Длина сообщения: зависит от числа записываемых блоков.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 45h | 1 |
| 2 | Номер блока, начиная с которого производится запись. | 2 |
| 3 | Число записываемых блоков N. Значение должно быть не более 15. Ограничение протокола | 1 |
| 4 | Настройка обмена, шифрование  1 – включено.  0 – выключено. | 1 |
| 5 | Настройка обмена, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 6 | Записываемые данные. Один блок данных содержит 16 байт. | 16\*N |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 46h “Чтение нескольких блоков данных с карту Mifare Plus”

Читает несколько блоков данных с карты Mifare Plus, находящейся в режиме SL2 или SL3. Ключ авторизации для читаемых блоков должен быть или один или ключи должны совпадать.

Есть поддержка ESMART. Считыватель проверяет тип SAM модуля в текущем слоте. На данный момент ESMART не поддерживает шифрование и подпись ответа. Эти параметры должны быть равны 0.

Сообщение к считывателю

Длина сообщения: 7.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 46h | 1 |
| 2 | Номер блока, начиная с которого производится чтение. | 2 |
| 3 | Число записываемых блоков N. Значение должно быть не более 15. Здесь ограничение протокола. | 1 |
| 4 | Настройка обмена с картой, шифрование  1 – включено.  0 – выключено. | 1 |
| 5 | Настройка обмена с картой, подпись ответа  1 – включено.  0 – выключено. | 1 |
| 6 | Настройка обмена с картой, подпись команды  1 – включено.  0 – выключено. | 1 |

Ответ от считывателя

Длина сообщения: зависит от числа читаемых блоков.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Считанные данные. Один блок данных содержит 16 байт. | 16\*N |

# Команда 47h “Авторизация к карте Mifare Plus в режиме SL1”

Производит авторизация к карте Mifare Plus находящуюся в режиме SL1 по ключу AES128, находящемуся в SAM AV2 модуле.

Команда нужна для перевода карты из режима SL1 в режим SL2 или SL3. Требует предварительного перевода карты в режим ISO14443-4 (см. команду 1Ch).

Так же используется для проверки карты на уровне SL1 (протокол ISO14443\_3), используя AES128 ключ (см. описание карты, ключ 0x9004). Позволяет отличить карту Mifare Plus работающую на уровне SL1, от карт Mifare Classic.

Сообщение к считывателю

Длина сообщения: 7 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 47h | 1 |
| 2 | Тип авторизации.  FOLLOWING\_AUTH (00h) – повторная авторизация.  FIRST\_AUTH (01h) – первичная авторизация. | 1 |
| 3 | Протокол  ISO14443\_3 (00h) – для авторизации к карте (не сектору) на уровне SL1.  ISO14443\_4 (01h) – для переключения на следующий уровень. | 1 |
| 4 | Номер блока. | 2 |
| 5 | Номер ключа в SAM AV2 модуле. Номер инкрементируется считывателем на 1. | 1 |
| 6 | Версия ключа в SAM AV2 модуле. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 48h “Запись AES128 ключа для карт Mifare Plus в SAM AV2 модуль”

Записывает ключ авторизации к карте Mifare Plus в SAM AV2 модуль.

**В одной записи можно хранить 3 ключа с разными версиями, в разных позициях.**

**Версии записываемых в одну запись ключей не должны совпадать.**

См. команду 66h, если нужно записать ключ с возможностью чтения.

Сообщение к считывателю

Длина сообщения: 20 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 48h | 1 |
| 2 | Номер записи (Key Entry) в которую будет записан ключ.  Значение от 0 до 126.  Значение инкрементируется считывателем на 1, что бы нельзя было переписать Master Key в ячейке 0. | 1 |
| 3 | Позиция ключа. Значение 0 – 2. | 1 |
| 4 | Версия ключа. Значение 0 - 255. | 1 |
| 5 | Ключ AES128. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 49h “Авторизация к сектору карты Mifare Plus в режиме SL2, ключом AES”

Производит авторизация к карте Mifare Plus находящуюся в режиме SL2 по ключу AES128, находящемуся в SAM AV2 модуле.

Команда нужна для перевода карты из режима SL2 в режим SL3. Требует предварительного перевода карты в режим ISO14443-4 (см. команду 1Ch).

Так же используется для авторизации к секторам карты находящейся на уровне SL2, ключом AES128. Затем требуется авторизация к сектору ключом Crypto-1, командой 0x4A. При этом ключи AES128 для секторов начинаются по адресам с 0x4000. Cм. приложение 2.

Сообщение к считывателю

Длина сообщения: 7 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 49h | 1 |
| 2 | Тип авторизации.  FOLLOWING\_AUTH (00h) – повторная авторизация.  FIRST\_AUTH (01h) – первичная авторизация. | 1 |
| 3 | Протокол  ISO14443\_3 (00h) – для авторизации к карте (к сектору) на уровне SL2.  ISO14443\_4 (01h) – для переключения на следующий уровень. | 1 |
| 4 | Номер блока. | 2 |
| 5 | Номер ключа в SAM AV2 модуле. Номер инкрементируется считывателем на 1. | 1 |
| 6 | Версия ключа в SAM AV2 модуле. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 4Ah “Авторизация к сектору карты Mifare Plus в режиме SL2, ключом Crypto-1”

Производит авторизация к сектору карты Mifare Plus, находящуюся в режиме SL2 по ключу Crypto-1. Предварительно требует авторизацию ключом AES128 по команде 0x49.

Сообщение к считывателю

Длина сообщения: 10, 13, 16 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 4Ah | 1 |
| 2 | Номер блока.  Этот же номер используется при диверсификации. | 1 |
| 3 | Тип ключа.  Для ключа A значение 60h.  Для ключа B значение 61h.  Для ключа A, с диверсификацией в SAM AV2, значение E0h.  Для ключа B, с диверсификацией в SAM AV2, значение E1h. | 1 |
| 4 | Номер ключа в SAM AV2 модуле. Номер инкрементируется считывателем на 1. | 1 |
| 5 | Версия ключа в SAM AV2 модуле. | 1 |
| 6 | Длина UID карты (4, 7 или 10). | 1 |
| 7 | UID карты | 4, 7, 10 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 4Bh “Запись нескольких блоков данных на карту Mifare Plus”

Записывает несколько блоков данных на карту Mifare Plus, находящуюся в режиме SL2. Ключ авторизации для записываемых блоков должен быть или один или ключи должны совпадать.

Сообщение к считывателю

Длина сообщения: зависит от числа записываемых блоков.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 4Bh | 1 |
| 2 | Номер блока, начиная с которого производится запись. | 1 |
| 3 | Число записываемых блоков N. Значение должно быть не более 3. Ограничение карты. | 1 |
| 4 | Записываемые данные. Один блок данных содержит 16 байт. | 16\*N |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 4Ch “Чтение нескольких блоков данных с карты Mifare Plus”

Читает несколько блоков данных с карты Mifare Plus, находящейся в режиме SL2. Ключ авторизации для читаемых блоков должен быть или один или ключи должны совпадать.

Сообщение к считывателю

Длина сообщения: 3.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 4Ch | 1 |
| 2 | Номер блока, начиная с которого производится чтение. | 1 |
| 3 | Число читаемых блоков N. Значение должно быть не более 3. Ограничение карты. | 1 |

Ответ от считывателя

Длина сообщения: зависит от числа читаемых блоков.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Считанные данные. Один блок данных содержит 16 байт. | 16\*N |

# Команда 50h “Чтение страницы с карты Ultraligth EV1”

Считывает данные с карты Ultralight EV1.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 50h | 1 |
| 2 | Номер читаемой страницы | 1 |

Ответ от считывателя

Длина сообщения: 17 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | 16 байт считанных данных. 4 первых байта для заданной страницы.  12 байт со следующих 3-х страниц. | 16 |

# Команда 51h “Команда GetVersion карты Ultraligth EV1”

Запрос идентификационной информации от карты.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 51h | 1 |

Ответ от считывателя

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | 8 байт информации о продукте (карте).  Байт 0 – фиксированное значение 00h  Байт 1 – Vendor ID, 04h – NXP.  Байт 2 – Product type, 03h – Ultralight.  Байт 3 – Product subtype, 01h – 17 pF.  Байт 4 – Major product version , 01h – EV1.  Байт 5 – Minor product version , 00h – V0.  Байт 6 – Storage size, 0Bh – MFUL11, 0Eh – MFUL21.  Байт 7 – Protocol type, 03h – ISO/IEC14443-3 compliant. | 8 |

# Команда 52h “Быстрое чтение страниц с карты Ultraligth EV1”

Считывает данные с карты Ultralight EV1 в заданном диапазоне номеров страниц.

Возвращает N страниц памяти.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 52h | 1 |
| 2 | Номер первой читаемой страницы | 1 |
| 3 | Номер конечной читаемой страницы | 1 |

Ответ от считывателя

Длина сообщения: 1 + N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | N запрошенных страниц по 4 байта | N\*4 |

# Команда 53h “Запись страницы на карту Ultraligth EV1”

Записывает данные на карту Ultralight EV1.

Сообщение к считывателю

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 53h | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 5Ch “Запись страницы на карту Ultraligth EV1 в режиме совместимости”

Записывает данные на карту Ultralight EV1 в режиме совместимости с Mifare Classic.

Сообщение к считывателю

Длина сообщения: 18 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 5Ch | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные. 4 первых байта должны содержать записываемые данные. 12 следующих байт игнорируются, в них рекомендуется записывать 0. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 5Fh “Чтение счётчика карты Ultraligth EV1”

Считывает значение заданного счётчика карты Ultralight EV1.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 5Fh | 1 |
| 2 | Адрес счётчика от 0 до 2 | 1 |

Ответ от считывателя

Длина сообщения: 4 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Значение счётчика. Младший байт первый. | 3 |

# Команда 60h “Инкремент счётчика карты Ultraligth EV1”

Инкрементирует значение заданного счётчика карты Ultralight EV1.

Сообщение к считывателю

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 60h | 1 |
| 2 | Адрес счётчика от 0 до 2 | 1 |
| 3 | Значение инкремента счётчика. Младший байт первый. | 3 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 61h “Авторизация по паролю к карте Ultraligth EV1”

Передача пароля карте для его проверки.

Сообщение к считывателю

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 61h | 1 |
| 2 | Значение пароля, 4 байта | 4 |

Ответ от считывателя

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Подтверждение авторизации PACK | 2 |

# Команда 62h “Чтение сигнатуры карты Ultralight EV1”

Возвращает 32 байта ECC сигнатуры (цифровой подписи) карты для проверки производителя.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 62h | 1 |

Ответ от считывателя

Длина сообщения: 33 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Сигнатура карты | 32 |

# Команда 63h “Проверка события Tearing карты Ultralight EV1”

Возвращает флаг события Tearing для заданного счётчика. Это событие пропадания поля в момент записи нового значения счётчика, т.е. проверка валидности значения счётчика.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 63h | 1 |
| 2 | Адрес счётчика от 0 до 2 | 1 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Флаг валидности.  Значение флага BDh – значение счётчика валидно, любое иное не валидно | 1 |

# Команда 64h “Выполнение команды VCSL карты Ultralight EV1”

Сообщение к считывателю

Длина сообщения: 21 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 64h | 1 |
| 2 | Installation identifier IID | 16 |
| 3 | PCD capabilities | 4 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Virtual card identifier | 1 |

# Команда 65h “Отправка APDU напрямую в SAM модуль”

Отправляет переданный APDU в SAM AV2 модуль и отправляет обратно ответ SAM AV2 модуля.

Сообщение к считывателю

Длина сообщения: 2 + N.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 65h | 1 |
| 2 | Зарезервировано.  Значение 00h | 1 |
| 3 | Запрос С-APDU | N |

Ответ от считывателя

Длина сообщения: 1 + N.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Ответ R-APDU | N |

# Команда 66h “Команда работы с Master SAM”

Обеспечивает функционал работы с Master SAM AV2 модулем.

**Необходимо помнить, что включение бита Allow Dump Secret Key даёт возможность считать ключ (с этим битом) из SAM AV2 модуля при наличии ключа авторизации к этому SAM модулю. Этот функционал можно использовать при создании Master SAM модуля.**

См. Приложение 11, где описан процесс создания и использования Master SAM.

После деактивации или активации SAM в слоте нужно всегда вызвать команду 27h для переключения слота на новый активированный или старый активный.

Параметр, который отсутствует для операции, при выполнении этой операции игнорируется и должен быть равен 0.

Сообщение к считывателю

Длина сообщения: 3/5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 66h | 1 |
| 2 | Код операции.  00h – включить/выключить бит Allow Dump Secret Key перед операцией записи ключа.  01h – чтение ключа Crypto1 (с установленным битом Allow Dump Secret Key) в память контроллера считывателя.  02h – чтение ключа PICC DES или PICC AES128 (с установленным битом Allow Dump Secret Key) в память контроллера считывателя.  03h – генерация ключа AES128/2K3DES на основе CMAC, с заданным усечением. Ключ не передаётся считывателю.  04h – сохранение сгенерированного в операции 03h ключа в память авторизации к SAM модулю.  05h – деактивация SAM модуля в заданном слоте.  06h – активация SAM модуля в заданном слоте.  07h – деактивация SAM модуля в заданном слоте и переключение указателей на работу без SAM.  08h – активация SAM модуля в заданном слоте и переключение на работу с SAM.  09h – чтение ключа PICC DES или PICC AES128 (с установленным битом Allow Dump Secret Key). Т.е. выдача прочитанного ключа из считывателя.  0Bh – получить ATR принятый от SAM модуля при его активации. | 1 |
| 3 | Состояние бита Allow Dump Secret Key. Присутствует для операции 00h. Значение 00h – выключить бит, больше ноля – включить бит.  Номер ячейки ключа для чтения. Присутствует для операций 01h, 02h, 09h, 0Ah. Значение от 0 до 126. Смещение на +1.  Размер усечения CMAC, для операции 03h.  Номер слота для операция 05h, 06h, 07h, 08h, 0Ah. | 1 |
| 4 | Версия ключа для чтения. Присутствует для операции 01h, 02h, 09h. Значение от 0 до 255.  Позиция ключа для операции 0Ah. Значение от 0 до 2 включительно. | 1 |
| 5 | Тип читаемого ключа. Для операции 01h.  0Ah – ключ A, 0Bh – ключ B, 0Ch – ключ A и ключ B.  Версия ключа для записи в операции 0Ah. Значение от 0 до 255. | 1 |

Ответ от считывателя

Длина сообщения: 1, 2, 17, N + 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Код ошибки активации SAM модуля в слоте, для операций 06h, 08h.  0 – успех активации.  1 – ошибочные параметры (не существующий слот).  2 – нет SAM модуля в слоте.  3 – провал по питанию считывателя.  4 – короткое замыкание по питанию SAM модуля.  5 – либо нет ответа ATR, либо ATR получен слишком рано.  6 – SAM модуль не активен.  7 – ошибка ATR.  8 – ошибка задания параметров обмена с SAM модулем.  Для команды 09h - 16 байт ключа.  Для команды 0Ah – N байт ATR. | 1/16/N |

# Команда 67h “Настройка параметров асинхронного интерфейса ведомого устройства”

Команда используется для задания параметров интерфейса подключаемого к ведомому устройству. Подача данной команды также сбрасывает буфер на передачу и приём данных. Записанные параметры сохраняются в энергонезависимой памяти и загружаются при следующем запуске считывателя.

Сообщение к считывателю

Длина сообщения: 7 или 8.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 67h | 1 |
| 2 | Скорость обмена от 1200 до 230400 бит в секунду.  Младший байт первый. | 4 |
| 3 | Число стоповых бит: значение 5 – 0.5 стоповых бит, значение 10 – 1 стоповых бит, значение 15 – 1.5 стоповых бит, значение 20 – 2 стоповых бит. Остальные значения дадут 1 стоповый бит. | 1 |
| 4 | Бит чётности: 0 или значения больше 2 – нет проверки на чётность, 1 – even, 2 – odd. | 1 |
| 5 | Опциональный параметр. Режим обмена с ведомым устройством  0 – ведомым устройством является моторизованный считыватель 3K5 или 3K7.  1 – ведомым устройством является диспенсер SCD-2500.  2 – проброс данных ведомому устройству через считыватель. См. команду 68h. | 1 |

Ответ от считывателя

Длина сообщения: 8.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Скорость обмена.  Младший байт первый. | 4 |
| 3 | Число стоповых бит | 1 |
| 4 | Бит чётности. | 1 |
| 1 | Режим обмена с ведомым устройством. | 1 |

# Команда 68h “Отправка данных ведомому устройству”

Команда используется для передачи (проброса) данных ведомому устройству через считыватель. Переданные командой данные накапливаются в буфере считывателя и затем передаются при выставленном флаге передачи данных. За одну команду можно передать до 253 байт. При необходимости передать больше, их можно накопить в буфере считывателя и затем передать ведомому устройству. Размер буфера 4096 байт. Данные записываются со сдвигом на размер уже записанных данных.

Сообщение к считывателю

Длина сообщения: от 1 до 255.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 68h | 1 |
| 2 | Флаг необходимости передачи данных из буфера. При значении равном 0 данные записываются в буфер | 1 |
| 3 | Данные для записи в буфер и их последующей выдачи. | 0..253 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки. Может быть только две ошибки: переполнение буфера, не верные параметры (если передан только код команды). | 1 |

# Команда 69h “Приём данных от ведомого устройства”

Команда используется для приёма данных от ведомого устройства через считыватель. Считыватель непрерывно получает байты от ведомого устройства и сохраняет их в буфер. Затем эти данные можно вычитать. Вычитывать пока размер принятых данных не станет равным нулю.

Сообщение к считывателю

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 69h | 1 |

Ответ от считывателя

Длина сообщения: от 1 до 251.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки. Есть один код ошибки: переполнение приёмного буфера. | 1 |
| 2 | Принятые данные. Размер принятых данных равен размеру ответа минус один байт кода ошибки. | 0..250 |

# Команда 6Ah “Задать состояние линии DTR”

Команда используется для задания состояния сигнала DTR к ведомому устройству и задания таймаута на сброс состояния. Можно задать текущее состояние сигнала и состояние после истечения таймаута.

Сообщение к считывателю

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 6Ah | 1 |
| 2 | Состояние сигнала DTR. | 1 |
| 3 | Состояние сигнала DTR после срабатывания таймаута. | 1 |
| 4 | Значение таймаута от 0 до 65535. Младший байт первый.  Значение 0 означает, что таймаут отключён. | 2 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 6Bh “Запрос состояния линии DSR”

Команда используется для запроса состояния сигнала DSR от ведомого устройства.

Сообщение к считывателю

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 6Bh | 1 |

Ответ от считывателя

Длина сообщения: 2.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Состояние сигнала DSR. | 1 |

# Команда 6Ch “Запрос значения счётчика APDU”

Команда используется для получения значения счётчика APDU команд отправленных SAM модулю в заданном слоте. При активации SAM модуля значение счётчика сбрасывается.

Сообщение к считывателю

Длина сообщения: 2 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 6Ch | 1 |
| 2 | Номер слота. Значение от 0 до 4. | 1 |

Ответ от считывателя

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Значение счётчика, младший байт первый | 4 |

# Команда 70h “Запись трейлера на карту из SAM AV2 модуля на уровне SL0”

Команда производит чтение ключей Crypto-1 (A и B) из SAM AV2 модуля и их запись в карту на уровне SL0. Считывается два ключа A и B из разных ячеек, и к ним добавляется переданные биты доступа карты. Длина ключей по 6 байт и 4 байт бит доступа. Т.е. всего 16 байт. Эти 16 байт пишутся в заданный блок на карте.

Трейлер, который пишется на карту, содержит 16 байт: 6 байт ключа A, 4 байта битов доступа, 6 байт ключа B.

Сообщение к считывателю

Длина сообщения: 20.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 70h | 1 |
| 2 | Номер ячейки ключа A в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 3 | Версия ключа A в заданной ячейке. Значение от 0 до 255. | 1 |
| 4 | Тип ключа для чтения из ячейки SAM AV2. Значение или 0x0A или 0x0B. Т.е. можно выбрать какой ключ из SAM AV2 будет считан и записан как ключ A на карту. Т.е. как ключ A можно записать ключ A или ключ B из SAM AV2. | 1 |
| 5 | Опция чтения ключа A. Значение 0x01 – чтение ключа A. Значение 0x02 – диверсификация ключа. | 1 |
| 6 | Номер ячейки ключа B в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 7 | Версия ключа B в заданной ячейке. Значение от 0 до 255. | 1 |
| 8 | Тип ключа для чтения из ячейки SAM AV2. Значение или 0x0A или 0x0B. Т.е. можно выбрать какой ключ из SAM AV2 будет считан и записан как ключ B на карту. Т.е. как ключ B можно записать ключ A или ключ B из SAM AV2. | 1 |
| 9 | Опция чтения ключа B. Значение 0x01 – чтение ключа B. Значение 0x02 – диверсификация ключа. | 1 |
| 10 | Номер блока карты для диверсификации. | 1 |
| 11 | Номер блока карты для записи. Младший байт первый. Значение от 0 до 255. | 2 |
| 12 | Байты битов доступа | 4 |
| 13 | Номер (UID) карты для диверсификации. | 4 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 71h “Запись ключей AES на карту из SAM AV2 модуля”

Команда производит чтение ключей AES из SAM AV2 модуля и их запись в карту на уровне SL0. Возможна диверсификация ключа, но для этого нужно передать данные для диверсификации.

Сообщение к считывателю

Длина сообщения: не менее 7.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 71h | 1 |
| 2 | Номер ячейки ключа A в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 3 | Версия ключа в заданной ячейке. | 1 |
| 4 | Опция чтения ключа. Действительны два младших бита: бит 0 – шифрование (0 –не шифрованный ключ, 1 – шифрованный в режиме обмена MAC), бит 1 – диверсификация (0 – вернуть ключ, 1 – вернуть диверсифицированный ключ).  Бит 1 выставляется при наличии данных для диверсификации. | 1 |
| 5 | Номер блока карты для записи и для диверсификации. Младший байт первый. Значение от 0 до 255. | 2 |
| 6 | Длина N данных для диверсификации. Значение от 0 до 31. | 1 |
| 7 | Данные для диверсификации. | N |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 72h “Запись трейлера на карту из SAM AV2 модуля на уровне SL1”

Команда производит чтение ключей Crypto-1 (A и B) из SAM AV2 модуля и их запись в карту на уровне SL1. Считывается два ключа A и B из разных ячеек, и к ним добавляется переданные биты доступа карты. Длина ключей по 6 байт и 4 байт бит доступа. Т.е. всего 16 байт. Эти 16 байт пишутся в заданный блок на карте.

Трейлер, который пишется на карту, содержит 16 байт: 6 байт ключа A, 4 байта битов доступа, 6 байт ключа B.

Перед записью трейлера нужна предварительная авторизация к сектору (блоку сектора).

Сообщение к считывателю

Длина сообщения: 20.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 72h | 1 |
| 2 | Номер ячейки ключа A в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 3 | Версия ключа A в заданной ячейке. Значение от 0 до 255. | 1 |
| 4 | Тип ключа для чтения из ячейки SAM AV2. Значение или 0x0A или 0x0B. Т.е. можно выбрать какой ключ из SAM AV2 будет считан и записан как ключ A на карту. Т.е. как ключ A можно записать ключ A или ключ B из SAM AV2. | 1 |
| 5 | Опция чтения ключа A. Значение 0x01 – чтение ключа A. Значение 0x02 – диверсификация ключа. | 1 |
| 6 | Номер ячейки ключа B в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 7 | Версия ключа B в заданной ячейке. Значение от 0 до 255. | 1 |
| 8 | Тип ключа для чтения из ячейки SAM AV2. Значение или 0x0A или 0x0B. Т.е. можно выбрать какой ключ из SAM AV2 будет считан и записан как ключ B на карту. Т.е. как ключ B можно записать ключ A или ключ B из SAM AV2. | 1 |
| 9 | Опция чтения ключа B. Значение 0x01 – чтение ключа B. Значение 0x02 – диверсификация ключа. | 1 |
| 10 | Номер блока карты для диверсификации. | 1 |
| 11 | Номер блока карты для записи и для диверсификации. Младший байт первый. Значение от 0 до 255. | 2 |
| 12 | Байты битов доступа | 4 |
| 13 | Номер (UID) карты для диверсификации. | 4 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 73h “Формирование криптограммы и запись на карту в режиме SL1”

Команда производит формирование криптограммы на основе записи с ключами Crypto-1 и её запись на карту. Команда позволяет производить смену ключей на карте в режиме SL1. Криптограмму формирует SAM AV2 модуль на основе заданной ячейки с ключём.

Перед выполнением команды необходимо произвести авторизацию к сектору карты, в который происходит запись. Необходимо т.к. SAM AV2 модуль шифрует данные на отправку по алгоритму Crypto-1. Авторизоваться необходимо ключём позволяющим выполнить перезапись ключей. Обычно это ключ B.

Сообщение к считывателю

Длина сообщения: 16.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 73h | 1 |
| 2 | Номер ячейки ключей A и B в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 3 | Версия ключа A в заданной ячейке. Значение от 0 до 255. | 1 |
| 4 | Версия ключа B в заданной ячейке. Значение от 0 до 255. | 1 |
| 5 | Опции диверсификации. Задаются как логическое ИЛИ. Значение 0x02 – диверсифицировать ключ A. Значение 0x04 – диверсифицировать ключ B. | 1 |
| 6 | Номер блока карты для диверсификации. Его надо будет использовать для авторизации к сектору. | 1 |
| 7 | Номер блока карты для записи и для диверсификации. Младший байт первый. Значение от 0 до 255. | 2 |
| 8 | Байты битов доступа | 4 |
| 9 | Номер (UID) карты для диверсификации. | 4 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 74h “Формирование криптограммы и запись на карту в режиме SL3”

Команда производит формирование криптограммы на основе записи с ключами AES-128 и её запись на карту. Команда позволяет производить смену ключей на карте в режиме SL3. Криптограмму формирует SAM AV2 модуль на основе заданной ячейки с ключём.

Перед выполнением команды необходимо произвести авторизацию к сектору карты, в который происходит запись. Необходимо т.к. SAM AV2 модуль шифрует данные на отправку по алгоритму AES-128. Авторизацию нужно производить ключём позволяющим выполнить запись ключей. Это ключ B.

Для выполнения данной команды карта должна быть переведена режим SL3 или должен быть переведён в режим SL3 отдельный сектор карты (Mifare Plus EV1) , или отдельный сектор карты должен быть переведён в режим MixMode (SL1 + SL3),

Сообщение к считывателю

Длина сообщения: не менее 7.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 74h | 1 |
| 2 | Код команды записи на карту Mifare Plus в режиме SL3.  A0 – запись на карту с ответом от карты без подписи MAC.  A1 – запись на карту с ответом от карты с подписью MAC. | 1 |
| 3 | Номер ячейки ключей A и B в SAM AV2 без смещения в 1. Значение от 1 до 127. | 1 |
| 4 | Версия ключа в заданной ячейке. Значение от 0 до 255. | 1 |
| 5 | Номер блока карты для записи и для диверсификации. Младший байт первый. Номера для SL3 начинаются со значения 4000h. | 2 |
| 6 | Длина N данных для диверсификации. Значение от 0 до 31. | 1 |
| 7 | Данные для диверсификации. | N |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 75h “Запись AES ключа Offline обновления ключей”

Записывает в SAM AV2 модуль ключ, который можно использовать для Offline обновления ключей. При этом ключи с таким обновление должны ссылаться на этот OfflieChangeKey. См. команду 29h, она позволяет задать такую ссылку перед записью ключа.

Такой OfflieChangeKey ключ не позволяет производить авторизацию или операции шифрования-дешифрования.

Перед обновлением ключей через OfflieChangeKey ключ необходимо его активировать. См. команду DCh.

**В данной команде нет инкремента номера ключа в считывателе. Переданный номер реально соответствует номеру ячейки в SAM AV2 модуле. Но запись в ячейку 0 запрещена в считывателе, что бы невозможно было испортить SAM модуль.**

**Необходимо внимательно отнесись к записи этого ключа. Ошибка приведёт к невозможности использовать модуль или часть его ячеек.**

Сообщение к считывателю

Длина сообщения: 22 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 75h | 1 |
| 2 | Номер ячейки SAM модуля. От 1 до 127. Здесь нет смещения номера на +1. | 1 |
| 3 | Номер позиции ключа в ячейке. От 0 до 2. | 1 |
| 4 | Версия ключа в позиции. | 1 |
| 5 | Ключ авторизации. | 16 |
| 6 | CRC ключа авторизации. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда 76h “Низкоуровневый обмен с картой”

Выполняет обмен с бесконтактно картой на низком уровне (ISO14443-3), с заданием параметров обмена.

Сообщение к считывателю

Длина сообщения: 8 + N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 76h | 1 |
| 2 | Параметры обмена: бит 0 – чётность включить/выключить, биты 1 - 3 – число действительных/передаваемых в последнем байте (0 – передаётся весь байт), бит 4 – CRC на передачу включить/выключить, бит 5 – CRC на приём включить/выключить. | 1 |
| 3 | Параметр TX\_WAIT – пауза на передачу после приёма. Значение по умолчанию 87. Младший байт первый. | 2 |
| 4 | Параметр RX\_DEAF – пауза в течении которого игнорируются принимаемые данные. Значение по умолчанию 8. Младший байт первый. | 2 |
| 5 | Таймаут на приём. Значение по умолчанию 150. Младший байт первый. | 2 |
| 6 | Данные на передачу. | N |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Расширенный статус. Младший байт первый. | 2 |
| 3 | Число действительных бит в последнем принятом байте. 0 – означает, что действителен весь байт. | 1 |
| 4 | Принятые данные. | N |

# Команда C0h “Генерация пары RSA ключей”

Выполняет генерацию RSA ключа (пары) с его сохранением в SAM AV2 модуле.

Всего в SAM AV2 модуле есть 3 ключевых записи для RSA ключей. Эти ключи хранятся отдельно от симметричных ключей.

Каждый ключ может иметь модуль от 256 бит (32 байта) до 2048 бит (256 байт) и с экспонентой от 32 бит (4 байта) до 2048 бит (256 байт). Для модуля размер кратен 64 битам (8 байт), для экспоненты 32 битам (4 байта). Модуль должен быть больше или равен экспоненте.

Ключевая запись с номером 2 может хранить только открытый ключ и не может использоваться при генерации пары ключей.

Ключи после генерации могут быть получены командами экспорта.

Формат хранения ключей в SAM AV2 модуле приведён в Приложении 5.

Сообщение к считывателю

Длина сообщения: 11.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C0h | 1 |
| 2 | Номер ключевой записи: [0:1] | 1 |
| 3 | Конфигурационные биты:  0 – наличие приватного ключа: 0 нет приватного ключа (не будет сгенерирован), но будет публичный; 1 есть.  1 – разрешение экспорта приватного ключа: 0 запрещён, 1 разрешён.  2 – запретить ключ: 0 запись разрешена, 1 запись запрещена.  3 – запрет использования ключа для шифрования и дешифрования: 0 разрешено, 1 –запрещено.  4 – запрет на использование ключа при работе с ЭЦП: 0 разрешено, 1 запрещено.  5 – разрешение использовать ключ для обновления ключей работы с картами. 0 запрещено, 1 разрешено.  6 – использовать CRT представление при хранении ключа: всегда 1.  7 – 15 – неиспользуемые биты, всегда 0. | 2 |
| 4 | Номер ключа (симметричного) авторизация по которому требуется для изменения этого ключа. Значение по умолчанию 0. | 1 |
| 5 | Версия ключа (симметричного) авторизация по которому требуется для изменения этого ключа. Значение по умолчанию 0. | 1 |
| 6 | Счётчик использования ключа. По умолчании 0xFF – без ограничений. | 1 |
| 7 | Модуль | 2 |
| 8 | Экспонента | 2 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда C1h “Экспорт приватного RSA ключа”

Выполняет экспорт приватной части RSA ключа (пары).

Приватная часть сохраняется в памяти считывателя (буфер) и затем может быть оттуда считана.

Реализация вызвана ограничением протокола считывателя: максимальная длина сообщения 255 байт.

Формат экспортируемого ключа приведён в Приложении 6.

Сообщение к считывателю

Длина сообщения: 2.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C1h | 1 |
| 2 | Номер ключевой записи: [0:1] | 1 |

Ответ от считывателя

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Объём данных полученных от SAM AV2 модуля и сохранённых в памяти считывателя. | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |

# Команда C2h “Экспорт публичного RSA ключа”

Выполняет экспорт публичной части RSA ключа (пары).

Приватная часть сохраняется в памяти считывателя (буфер) и затем может быть оттуда считана.

Реализация вызвана ограничением протокола считывателя: максимальная длина сообщения 255 байт.

Формат экспортируемого ключа приведён в Приложении 7.

Сообщение к считывателю

Длина сообщения: 2.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C2h | 1 |
| 2 | Номер ключевой записи: [0:2] | 1 |

Ответ от считывателя

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Объём данных полученных от SAM AV2 модуля и сохранённых в памяти считывателя. | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |

# Команда C3h “Чтение буфера считывателя”

Выполняет чтение буфера считывателя.

Сообщение к считывателю

Длина сообщения: 3.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C3h | 1 |
| 2 | Адрес читаемого блока | 1 |
| 3 | Число N читаемых из блока байт. При значении 0, может быть использована для получения параметров буфера | 1 |

Ответ от считывателя

Длина сообщения: зависит от параметров.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер буфера | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |
| 5 | Считанные данные | N |

# Команда C4h “Запись буфера считывателя”

Выполняет запись буфера считывателя.

Сообщение к считывателю

Длина сообщения: зависит от параметров.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C4h | 1 |
| 2 | Адрес записываемого блока | 1 |
| 3 | Число N записываемых байт. При значении 0, может быть использована для получения параметров буфера. | 1 |
| 4 | Записываемые данные | N |

Ответ от считывателя

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер буфера | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |

# Команда C5h “Вычислить цифровую подпись”

Выполняет вычисление цифровой подписи с использованием приватного RSA ключа.

Сообщение к считывателю

Длина сообщения: 23, 31, 35.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C5h | 1 |
| 2 | Используемый алгоритм вычисления хеш.  0 – SHA1  1 – SHA2-224  3 – SHA2-256 | 1 |
| 3 | Номер используемого ключа: [0, 1] | 1 |
| 4 | Значение хеш.  Длина 20 байт для SHA1, 28 для SHA2-224, 32 для SHA2-256 | 20, 28, 32 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда C6h “Получить вычисленную цифровую подпись”

Считывает вычисленную (0xC5) цифровую подпись из SAM AV2 модуля в память считывателя.

Сообщение к считывателю

Длина сообщения: 1

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C6h | 1 |

Ответ от считывателя

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Объём данных полученных от SAM AV2 модуля и сохранённых в памяти считывателя. Т.е. размер полученной подписи. Равен длине ключа | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |

# Команда C7h “Проверка цифровой подписи”

Проверяет цифровую подпись.

Проверяемая подпись должна быть записана в буфер считывателя.

Формат импортируемого ключа приведён в Приложении 8.

Сообщение к считывателю

Длина сообщения: зависит от параметров команды

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C7h | 1 |
| 2 | Номер используемого ключа: [0, 2] | 1 |
| 3 | Используемый алгоритм вычисления хеш.  0 – SHA1  1 – SHA2-224  3 – SHA2-256 | 1 |
| 4 | Число N байт передаваемого хеша: 20, 28, 32 | 1 |
| 5 | Число байт передаваемой на проверку подписи | 2 |
| 6 | Передаваемый хеш | N |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда C8h “Импорт RSA ключа”

Производит запись RSA ключа в SAM AV2 модуль или обновление его параметров.

Сообщение к считывателю

Длина сообщения: 3

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C8h | 1 |
| 2 | Номер используемого ключа: [0, 2] | 1 |
| 3 | Опции импорта.  При значении 0 происходим импорт ключа, при значении 1 происходит обновление настроек ключа | 1 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда С9h “Шифрование данных, используя Offline Crypto”

Производит шифрование данных активированным ключом (команда 0xDC). Объём данных ограничен размером буфера считывателя.

Данные должны быть предварительно записаны в буфер командой C4h и затем считаны командой C3h.

Сообщение к считывателю

Длина сообщения: 3/4

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: C9h | 1 |
| 2 | Размер шифруемых данных | 2 |
| 3 | Опциональный байт. Если отсутствует, то его значение принимается равным 0.  Значение 0 – шифруется один блок данных или последний блок данных в цепочке.  Значение 1 – шифруется цепочка данных и после переданного блока должен быть ещё блок.  Для шифрования цепочки данных нужно записывать каждый блок командой C4h, шифровать командой C9h и вычитывать результат командой C3h. Для всех блоков, кроме последнего, нужно указывать данный параметр равным 1. | 1 |

Ответ от считывателя

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер зашифрованных данных | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |

# Команда СAh “Дешифрование данных, используя Offline Crypto”

Производит дешифрование данных активированным ключом (команда 0xDC). Объём данных ограничен размером буфера считывателя.

Данные должны быть предварительно записаны в буфер командой C4h и затем считаны командой C3h.

Сообщение к считывателю

Длина сообщения: 3/4

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: CAh | 1 |
| 2 | Размер шифруемых данных | 2 |
| 3 | Опциональный байт. Если отсутствует, то его значение принимается равным 0.  Значение 0 – дешифруется один блок данных или последний блок данных в цепочке.  Значение 1 – дешифруется цепочка данных и после переданного блока должен быть ещё блок.  Для шифрования цепочки данных нужно записывать каждый блок командой C4h, шифровать командой C9h и вычитывать результат командой C3h. Для всех блоков, кроме последнего, нужно указывать данный параметр равным 1. | 1 |

Ответ от считывателя

Длина сообщения: 5.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер дешифрованных данных | 2 |
| 3 | Размер блоков буфера. Т.е. максимальный размер блока, который может быть считан за один раз. | 1 |
| 4 | Число блоков буфера. | 1 |

# Команда CBh “Вычисление хэш для данных в буфере”

Вычисляет хеш для данных находящихся в буфере считывателя.

Сообщение к считывателю

Длина сообщения: 4

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: CBh | 1 |
| 2 | Алгоритм вычисления хеш:  0 – SHA1  1 – SHA2-224  3 – SHA2-256 | 1 |
| 3 | Размер данных | 2 |

Ответ от считывателя

Длина сообщения: 21, 29, 33.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Хеш. 20 байт для SHA1, 28 байт для SHA2-224, 32 байт для SHA2-256 | 20, 28, 32 |

# Команда CCh “Активация Offline Crypto ключа”

Производит активацию ключа, который будет использован для Offline Crypto. Ключ должен быть предварительно записан командой DBh.

Команда аналог команды DCh, но добавлен параметр диверсификации.

См. команду 66h, если нужно записать ключ с возможностью чтения.

Сообщение к считывателю

Длина сообщения: 4 + N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: CCh | 1 |
| 2 | Номер записи (Key Entry) в которую записан ключ.  Значение от 0 до 126. | 1 |
| 3 | Версия ключа. Значение 0 - 255. | 1 |
| 4 | Длина данных для диверсификации ключа N | 1 |
| 5 | Данные для диверсификации ключа | N |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда СDh “Вычисление MAC, используя Offline Crypto”

Производит вычисление подписи MAC активированным ключом (команда 0xDC). Данные должны быть предварительно записаны в буфер командой C4h.

Сообщение к считывателю

Длина сообщения: 3

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: CDh | 1 |
| 2 | Длина (усечение) вычисленной подписи MAC.  Если задано значение 0, то данные передаются в SAM модуль и вычисляется промежуточное значение MAC, которое не выдаётся. Окончательно MAC будет вычислен и выдан при подаче длины усечения не равной 0. Это позволяет вычислять подпись для больших объёмов данных. Для этого нужно записывать блок данных командой C4h, затем вызывать CDh с усечением в 0. И для последнего блока вызвать CDh с нужным усечением. | 1 |
| 3 | Размер данных в буфере | 1 |

Ответ от считывателя

Длина сообщения: 2 + N.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер вычисленного MAC N.  Равен 0 при передаче усечения равного 0. | 1 |
| 3 | Подпись MAC | N |

# Команда СEh “Верификация MAC, используя Offline Crypto”

Производит проверку подписи MAC активированным ключом (команда 0xDC).

Данные должны быть предварительно записаны в буфер командой C4h.

Команде надо передать данные, для которых вычислена подпись MAC и саму подпись MAC. В итоге поле 3 должно содержать N байт данных и M байт подписи.

Сообщение к считывателю

Длина сообщения: 3

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: CEh | 1 |
| 2 | Длина подписи MAC M.  Если задано значение 0, то данные передаются в SAM модуль и вычисляется промежуточное значение MAC, которое не проверяется. Окончательно MAC будет вычислен и проверен при подаче длины усечения не равной 0. При этом в конце последнего блока данных должны быть байты подписи. Это позволяет проверять подпись для больших объёмов данных.  Для этого нужно записывать блок данных командой C4h, затем вызывать CEh с усечением в 0. И для последнего блока (с подписью) вызвать CEh с нужным усечением. Последний блок всегда должен содержать данные и MAC. Хотя бы один байт данных. | 1 |
| 3 | Размер данных в буфере N + M.  Сначала идут байты данных, затем байты подписи. | 1 |

Ответ от считывателя

Длина сообщения: 1.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда СFh “Генерация случайной последовательности”

Производит проверку генерацию случайной последовательности на основе датчика температуры считывателя.

Сообщение к считывателю

Длина сообщения: 2

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: CEh | 1 |
| 2 | Размер последовательности N.  Значение от 0 до 250. | 1 |

Ответ от считывателя

Длина сообщения: 1 + N.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Случайная последовательность | N |

# Команда D0h “Чтение страниц с карты My-D Move”

Считывает данные с карты My-D Move компании Infineon.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D0h | 1 |
| 2 | Номер читаемой страницы | 1 |

Ответ от считывателя

Длина сообщения: 17 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | 16 байт считанных данных. 4 первых байта для заданной страницы.  12 байт со следующих 3-х страниц. | 16 |

# Команда D1h “Чтение двух страниц с карты My-D Move”

Считывает данные с карты My-D Move компании Infineon.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D1h | 1 |
| 2 | Номер читаемой страницы | 1 |

Ответ от считывателя

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | 8 байт считанных данных. 4 первых байта для заданной страницы.  4 байта со следующей страницы. | 8 |

# Команда D2h “Запись страницы на карту My-D Move”

Записывает данные на карту My-D Move.

Сообщение к считывателю

Длина сообщения: 6 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D2h | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда D3h “Запись двух страниц на карту My-D Move”

Записывает данные на карту My-D Move.

Сообщение к считывателю

Длина сообщения: 10 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D3h | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные | 8 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда D4h “Запись страницы на карту My-D Move в режиме совместимости”

Записывает данные на карту My-D Move в режиме совместимости с Mifare Classic.

Сообщение к считывателю

Длина сообщения: 18 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D4h | 1 |
| 2 | Номер записываемой страницы | 1 |
| 3 | Записываемые данные. 4 первых байта должны содержать записываемые данные. 12 следующих байт игнорируются, в них рекомендуется записывать 0. | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда D5h “Запись пароля на карту My-D Move”

Записывает 4-х байтный пароль на карту My-D Move.

Сообщение к считывателю

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D5h | 1 |
| 2 | Записываемый пароль | 4 |

Ответ от считывателя

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Записанный пароль | 4 |

# Команда D6h “Авторизация по паролю к карте My-D Move”

Производит авторизацию к карте My-D Move.

Сообщение к считывателю

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D6h | 1 |
| 2 | Пароль | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда D7h “Декремент счётчика карты My-D Move”

Производит декремент счётчика на карте My-D Move.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D7h | 1 |
| 2 | Значение декремента | 2 |

Ответ от считывателя

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Значение счётчика после выполнения декремента | 2 |

# Команда D8h “Вычисление хэш функции SHA256”

Производит вычисление hash значения для переданных данных.

Сообщение к считывателю

Длина сообщения: 3 - 255 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D8h | 1 |
| 2 | Данные | 0 - 252 |
| 3 | CRC16 | 2 |

Ответ от считывателя

Длина сообщения: 33 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Вычисленное значение hash | 32 |

# Команда D9h “Вычисление хэш функции SHA224”

Производит вычисление hash значения для переданных данных.

Сообщение к считывателю

Длина сообщения: 3 - 255 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: D9h | 1 |
| 2 | Данные | 0 - 252 |
| 3 | CRC16 | 2 |

Ответ от считывателя

Длина сообщения: 29 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Вычисленное значение hash | 28 |

# Команда DAh “Вычисление хэш функции SHA-1”

Производит вычисление hash значения для переданных данных.

Сообщение к считывателю

Длина сообщения: 3 - 255 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: DAh | 1 |
| 2 | Данные | 0 - 252 |
| 3 | CRC16 | 2 |

Ответ от считывателя

Длина сообщения: 21 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Вычисленное значение hash | 20 |

# Команда DBh “Запись ключа для Offline Crypto в SAM AV2 модуль”

Записывает ключ Offline Crypto в SAM AV2 модуль.

**В одной записи можно хранить 2/3 ключа с разными версиями, в разных позициях.**

Сообщение к считывателю

Длина сообщения: 21/29 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: DBh | 1 |
| 2 | Тип ключа:  0 – AES128,  1 – AES192,  2 – 2K3DES,  3 – 3K3DES | 1 |
| 3 | Номер записи (Key Entry) в которую будет записан ключ.  Значение от 0 до 126. Номер смещается на +1. | 1 |
| 4 | Позиция ключа.  Значение 0 – 2 для AES128 и 2K3DES.  Значение 0 – 1 для AES192 и 3K3DES. | 1 |
| 5 | Версия ключа. Значение 0 - 255. | 1 |
| 6 | Ключ:  AES128 – 16 байт,  AES192 – 24 байт,  2K3DES – 16 байт,  3K3DES – 24 байт | 16/24 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда DCh “Активация Offline Crypto ключа”

Производит активацию ключа, который будет использован для Offline Crypto. Ключ должен быть предварительно записан командой DBh.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: DCh | 1 |
| 2 | Номер записи (Key Entry) в которую записан ключ.  Значение от 0 до 126. | 1 |
| 3 | Версия ключа. Значение 0 - 255. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда DDh “Шифрование данных, используя Offline Crypto”

Производит шифрование данных активированным ключом.

Сообщение к считывателю

Длина сообщения: зависит от данных.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: DDh | 1 |
| 2 | Данные (размер должен быть кратен 16 байтам для AES128 и 8 байтам для DES) | до 240 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер данных | 1 |
| 3 | Зашифрованные данные | до 240 |

# Команда DEh “Дешифрование данных, используя Offline Crypto”

Производит дешифрование данных активированным ключом.

Сообщение к считывателю

Длина сообщения: зависит от данных.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: DEh | 1 |
| 2 | Данные (размер должен быть кратен 16 байтам для AES128 и 8 байтам для DES) | до 240 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Размер данных | 1 |
| 3 | Дешифрованные данные | до 240 |

# Команда DFh “Деактивация авторизации к SAM AV2 модулю”

Деактивирует авторизацию считывателя к SAM AV2 модулю.

После этой команды считыватель не может работать с SAM AV2 модулем. Требуется сброс для повторной авторизации. Команда может быть использована в целях безопасности, при вскрытии изделия.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: DFh | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда EFh “Управление хранением ключей считывателя”

Команда служит для управления тем, откуда считыватель берёт ключи для авторизации к картам.

Кроме SAM AV2 модуля ключи могут храниться в SRAM (оперативная память) микроконтроллера считывателя. И для версий считывателя микросхему SPI Flash памяти есть возможность сохранять туда ключи и считывать их.

Сохранение ключей в SPI Flash памяти и чтение их оттуда доступно только для исполнений считывателя с этой памятью.

**Не рекомендуется к использованию т.к. ключ передаётся считывателю в открытом виде. Кроме того при каждой авторизации контроллер считывателя передаёт ключ приёмопередатчику в открытом виде.**

Сообщение к считывателю

Длина сообщения: 2 или 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: EFh | 1 |
| 2 | Код операции:  00h – режим работы. Для этой операции передаётся 3-й байт сообщения. Необходим сброс считывателя для переключения режима после его задания.  01h – сохранить ключи из SRAM в SPI Flash. Время выполнения записи 6 секунд.  02h – считать ключи из SPI Flash в SRAM. Время выполнения чтения 1 секунда.  03h – запрос режима работы. | 1 |
| 3 | Режим работы.  00h – использовать SAM AV2 модуль для хранения ключей.  01h – использовать SRAM для хранения ключей. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт или 2 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Режим работы, для операции 03h.  Значение 0 – c SAM AV2 модулем.  Значение >0 – без SAM AV2 модуля. | 1 |

# Команда F0h “Управление светодиодами и чтение состояния кнопки”

Управление светодиодами, установленными в считывателе и чтение состояния кнопки.

См. конкретную реализацию аппаратной части считывателя. От неё зависит число светодиодов и наличие кнопки.

В модификации агентского считывателя вместо состояния кнопки, возвращается состояние антенны.

Сообщение к считывателю

Длина сообщения: 2 байта или 4 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F0h | 1 |
| 2 | Биты управления светодиодами.  Значения битов: 1 – зажечь, 0 – погасить.  Бит 0 – синий, бит 1 – красный, бит 2 – зелёный, бит 3 – жёлтый. | 1 |
| 3 | Опциональный байт, присутствует для считывателя SM16089.  Биты управления трехцветным светодиодом 0, подключённым через IO экспандер.  Значения битов: 1 – зажечь, 0 – погасить.  Бит 0 – синий, бит 1 – красный, бит 2 – зелёный. | 1 |
| 4 | Опциональный байт, присутствует для считывателя SM16089.  Биты управления трехцветным светодиодом 1, подключённым через IO экспандер.  Значения битов: 1 – зажечь, 0 – погасить.  Бит 0 – синий, бит 1 – красный, бит 2 – зелёный. | 1 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 1 | Состояние кнопки.  0 –не была нажата.  > 0 – была нажата.  Состояние сбрасывается после выполнения команды.  Всегда 0, если кнопка не установлена в считывателе.  Состояние антенны.  0 – антенна не подключена.  1 –антенна подключена.  2 – короткое замыкание в цепи антенны.  >=3 – неисправность цепей диагностики антенны. | 1 |

# Команда F1h “Переход в режим программирования”

Передача управления Bootloader-у.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F1h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда F2h “Управление светодиодами”

Управление светодиодами, установленными в считывателе.

См. конкретную реализацию аппаратной части считывателя. От неё зависит число светодиодов.

Сообщение к считывателю

Длина сообщения: 2 байта или 4 байта.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F2h | 1 |
| 2 | Биты управления светодиодами.  Значения битов: 1 – зажечь, 0 – погасить.  Бит 0 – синий, бит 1 – красный, бит 2 – зелёный, бит 3 – жёлтый. | 1 |
| 3 | Опциональный байт, присутствует для считывателя SM16089.  Биты управления трехцветным светодиодом 0, подключённым через IO экспандер.  Значения битов: 1 – зажечь, 0 – погасить.  Бит 0 – синий, бит 1 – красный, бит 2 – зелёный. | 1 |
| 4 | Опциональный байт, присутствует для считывателя SM16089.  Биты управления трехцветным светодиодом 1, подключённым через IO экспандер.  Значения битов: 1 – зажечь, 0 – погасить.  Бит 0 – синий, бит 1 – красный, бит 2 – зелёный. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда F3h “Чтение состояния кнопки”

Чтение состояния кнопки.

См. конкретную реализацию аппаратной части считывателя. От неё зависит наличие кнопки.

В модификации агентского считывателя вместо состояния кнопки, возвращается состояние антенны.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F3h | 1 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Состояние кнопки.  0 –не была нажата.  > 0 – была нажата.  Состояние сбрасывается после выполнения команды.  Всегда 0, если кнопка не установлена в считывателе.  Состояние антенны.  0 – антенна не подключена.  1 –антенна подключена.  2 – короткое замыкание в цепи антенны.  >=3 – неисправность цепей диагностики антенны. | 1 |

# Команда F4h “Включение режима ответа с кодом команды”

Включает режим работы при котором в ответ от считывателя добавляется код команды.

Код команды добавляется после кода ошибки. Длина ответа при этом увеличивается на 1.

Режим сбрасывается после сброса считывателя.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F4h | 1 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Код команды F4h. | 1 |

# Команда F5h “Установить скорость обмена”

Устанавливает скорость обмена с ведущим устройством, используемая считывателем.

Установленное значение сохраняется в энергонезависимой памяти.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F5h | 1 |
| 2 | Скорость обмена.  0 – 2400 бод.  1 – 4800 бод.  2 – 9600 бод.  3 – 14400 бод.  4 – 19200 бод.  5 – 28800 бод.  6 – 38400 бод.  7 – 57600 бод. **Значение по умолчанию.**  8 – 115200 бод. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда F6h “Чтение настроек считывателя”

Чтение настроек считывателя из энергонезависимой памяти.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F6h | 1 |

Ответ от считывателя

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Скорость обмена, см. команду 0xF5 | 1 |
| 3 | Параметр REG\_RFCFG | 1 |
| 4 | Параметр REG\_RXSEL | 1 |
| 5 | Параметр REG\_MODE | 1 |
| 6 | Параметр REG\_GSN | 1 |
| 7 | Параметр REG\_CWGSP | 1 |
| 8 | Параметр REG\_MODGSP | 1 |
| 9 | Зарезервировано | 1 |

# Команда F7h “Запись настроек считывателя”

Запись настроек считывателя в энергонезависимую память.

Команда позволяет записать скорость обмена и параметры приёмо-передатчика считывателя (глубину модуляции, амплитуду сигнала).

Сообщение к считывателю

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F7h | 1 |
| 2 | Скорость обмена, см. команду 0xF5 | 1 |
| 3 | Параметр REG\_RFCFG | 1 |
| 4 | Параметр REG\_RXSEL | 1 |
| 5 | Параметр REG\_MODE | 1 |
| 6 | Параметр REG\_GSN | 1 |
| 7 | Параметр REG\_CWGSP | 1 |
| 8 | Параметр REG\_MODGSP | 1 |
| 9 | Зарезервировано | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда F8h “Запрос расширенного кода ошибки”

Команда служит для получения расширенного кода ошибки последней ошибочно выполненной команды.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F8h | 1 |

Ответ от считывателя

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Старший байт кода ошибки | 1 |
| 3 | Младший байт кода ошибки | 1 |
| 4 | Код ошибки возникшей при старте ридера  0x00 Нет ошибки  0x01 Не отвечает TDA8026  0x02 Не SAM модулей  0x03Не отвечает приёмопередатчик PN512  0x04 Ошибка авторизации к SAM модулю  0x05 Не отвечает SAM модуль  0x06 Не отвечает моторизованный считыватель  0x09 Перегрузка по питания SAM AV2 модуля (модулей). КЗ.  0x0A Слишком низкое питание SAM AV2 модуля. |  |

# Команда F9h “Запись ключа авторизации к SAM AV2 модулю в память считывателя”

Команда служит для записи ключа AES128 в память считывателя. После перезагрузки считывателя он будет авторизоваться к SAM AV2 модулю этим ключём.

Для записи ключа в SAM AV2 модуль служит команда 0x34h.

Сообщение к считывателю

Длина сообщения: 20 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: F9h | 1 |
| 2 | Параметр, указывающий на использование ключа.  0 – ключ не используется для авторизации при включении считывателя  Больше 0 - ключ используется для авторизации при включении считывателя | 1 |
| 3 | Ключ авторизации AES128 | 16 |
| 4 | CRC полей 2 и 3. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда FAh “Запись UID считывателя в память”

Запись 16-ти байтного UID в память считывателя. Не путать с серийным номером считывателя.

Сообщение к считывателю

Длина сообщения: 19 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FAh | 1 |
| 2 | UID считывателя | 16 |
| 3 | CRC поля 2. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда FBh “Чтение UID считывателя из памяти”

Чтение 16-ти байтного UID из памяти считывателя. Не путать с серийным номером считывателя.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FBh | 1 |

Ответ от считывателя

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | UID | 16 |

# Команда FCh “Ротация светодиодов”

Управление режимом работы светодиодов считывателя.

Включает или выключает “ротацию” светодиодов: красный-зелёный, синий-жёлтый.

Нужно для плат SME10158\_387\_01 и SME10158\_384\_01.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FCh | 1 |
| 2 | Режим:  0 – выключено  1 - включено | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда FEh “Переключение антенного входа”

Переключение антенного входа приёмопередатчика. Реализовано для платы считывателя SME10145.622.001\_2.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FEh | 1 |
| 2 | Номер антенны:  0 –первая антенна.  1 – вторая антенна. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команда FFh “Снижение энергопотребления ”

Выключает поле считывателя и светодиоды.

Для включения поля необходимо подать команду 02h.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FFh | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команды протокола для работы с 3K5

В данном разделе перечислены команды работы считывателя с моторизованным считывателем 3K5.

## Команда E0h “Разрешение/запрет приёма карт”

Команда разрешения или запрета приёма карт.

После разрешения приёма карт, моторизованный считыватель переходит в режим, при котором самостоятельно принимает карты.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E0h | 1 |
| 002 | Параметр команды.  0x30 – разрешить приём карт.  0x31 – запретить приём карт. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E1h “Запросить статус считывателя”

Команда запроса статус считывателя. Используется для определения наличия карты в считывателе.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E1h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E2h “Выдать карту”

Команда выдачи карты клиенту.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E2h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E3h “Захватить карту”

Команда захвата карты.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E3h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E4h “Получить ответ на последнюю поданную команду”

Команда возвращает ответ от считывателя на последнюю поданную команду.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E4h | 1 |

Ответ от считывателя

Длина сообщения: 2 + N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  Всегда 0x00 | 1 |
| 2 | Код ответа считывателя   1. Ответ ещё не получен 2. ACK 3. NAK 4. STX - данные 5. STX – данные, но ошибка CRCC 6. DLE 7. EOT 8. таймаут закончился, но считыватель не ответил 9. Нет сигнала линии DSR, отключен кабель к 3K5/3K7 | 1 |
| 3 | Данные от считывателя. Актуально при коде ответа равном 3. | N |

## Команда E5h “Управление светодиодом”

Команда управления светодиодом.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E0h | 1 |
| 2 | Параметр команды.  0x30 – погасить.  0x31 – зажечь зелёный.  0x32 – зажечь красный.  0x33 – зажечь желтый. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

# Команды протокола для работы с SCD2500

В данном разделе перечислены команды работы считывателя с диспенсером SCD2500.

## Команда E0h “Разрешение/запрет приёма карт”

Команда разрешения или запрета приёма карт.

После разрешения приёма карт, моторизованный считыватель переходит в режим, при котором самостоятельно принимает карты.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E0h | 1 |
| 2 | Параметр команды.  0x30 – разрешить приём карт.  0x31 – запретить приём карт. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E1h “Запросить статус считывателя”

Команда запроса статус считывателя. Используется для определения наличия карты в считывателе.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E1h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E2h “Выдать карту”

Команда выдачи карты клиенту.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E2h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E3h “Захватить карту”

Команда захвата карты.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E3h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E4h “Получить ответ на последнюю поданную команду”

Команда возвращает ответ от считывателя на последнюю поданную команду.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E4h | 1 |

Ответ от считывателя

Длина сообщения: 2 + N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  Всегда 0x00 | 1 |
| 2 | Код ответа считывателя   1. Ответ ещё не получен 2. R<ACK> 3. B<ACK> 4. ACK 5. NAK 6. STX - данные 7. STX – данные, но ошибка CRCC 8. таймаут закончился, но считыватель не ответил | 1 |
| 3 | Данные от считывателя. Актуально при коде ответа равном 5. | N |

## Команда E5h “Запрос конфигурации”

Команда запроса конфигурации считывателя (описание номера версии программного обеспечения устройства).

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E5h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E6h “Выдача карты из кассеты”

Команда выдачи карты из кассеты в контактный разъём.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E6h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E7h “Включение или выключение цикла чистящей карты”

Команда включения или выключения циклического режима. При включении циклического режима карта перемещается в положение подключения и отключения относительно контактного разъема смарт-карт. Режим используется для удаления мелких частиц с контактного разъема смарт-карт.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E7h | 1 |
| 2 | Параметр команды.  0x30 – выключить циклический режим.  0x31 – включить циклический режим. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E8h “Управление блокировочным штифтом”

Команда управления блокировочным штифтом. Используется для открывания или закрывания блокировочного штифта.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E8h | 1 |
| 2 | Параметр команды.  0x30 – открыть блокировочный штифт.  0x31 – закрыть блокировочный штифт. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда E9h “Сброс состояния вмешательства”

Команда сброса состояния вмешательства.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E9h | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда EAh “Сброс состояния залипания”

Команда сброса состояния залипания и, либо задержать, выбросить или принять карту, находящуюся в транспортном канале.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: EAh | 1 |
| 2 | Параметр команды.  0x30 – сбросить и задержать карту.  0x31 – сбросить и выбросить карту.  0x32 – сбросить и принять карту. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда EBh “Прочитать информацию о функционировании”

Команда запроса информации о функционировании устройства.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: EBh | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда ECh “Запрос состояния оптических датчиков”

Команда запроса состояния оптических датчиков устройства.

Сообщение к считывателю

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: ECh | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

## Команда EDh “Режим отчистки”

Команда включения или выключения режима отчистки (ручной).

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: EDh | 1 |
| 2 | Параметр команды.  0x30 – выключить режим отчистки.  0x31 – включить режим отчистки. | 1 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – нет ошибок.  MI\_WRONG\_PARAMETER\_VALUE (-60) – ошибка в парметрах команды  0xFF – моторизованный считыватель занят предыдущей командой | 1 |

# Команды протокола для эмиссии карт

## Команда 26h “Запись зашифрованных AES128 ключом данных на карту”

Записывает на карту блок данных, предварительно расшифровав его указанным AES ключом.

Сообщение к считывателю

Длина сообщения: 22 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: 26h | 1 |
| 2 | Протокол, используемый для записи:  0x00 – ISO14443-3 (для карт Mifare Classic)  0x01 – ISO14443-4 (для карт Mifare Plus) | 1 |
| 3 | Номер блока для записи | 2 |
| 4 | Номер записи (Key Entry) с ключом AES для дешифрования блока | 1 |
| 5 | Версия ключа | 1 |
| 6 | Данные | 16 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

# Команды эмуляции карт NFC Forum Type 2 Tag

В данном разделе перечислены команды работы считывателя в режиме эмуляции карт NFC Forum Type 2 Tag. Есть ограничение: перед каждой операцией чтения или записи, необходима активация карты.

Объём памяти эмулируемой карты 2К.

Код команды содержит два байта. Первым передаётся старший байт.

Первый байт UID карты всегда равен 0x08, ограничение приёмопередатчка считывателя.

Рекомендуется сначала записать данные в память и затем включать режим эмуляции. С чтением наоборот. Сначала выключить режим эмуляции и затем считать полученные данные. Связанно это с невозможностью одновременной обработки команд от ведущего устройства и эмуляцией карты.

## Команда FD00h “Включить режим эмуляции карты T2T”

Включает режим эмуляции карты.

Сообщение к считывателю

Длина сообщения: 9 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FD00h | 2 |
| 2 | ATQ (Answer To Request) карты. Ответ карты на запрос согласно ISO/IEC14443-3. Для Type 2 Tag это значение равно 0044h. | 2 |
| 3 | SAK карты. Ответ карты на команду выбора. Для Type 2 Tag это значение равно 00h. | 1 |
| 4 | UID карты. Первый байт всегда подменяется на 08h | 4 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

## Команда FD01h “Выключить режим эмуляции карты T2T”

Выключает режим эмуляции карты.

Сообщение к считывателю

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FD01h | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

## Команда FD02h “Записать данные в память карты”

Производит запись данных в память эмулируемой карты.

Сообщение к считывателю

Длина сообщения: 7+N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FD02h | 2 |
| 2 | Адрес, начиная с которого будет идти запись | 2 |
| 3 | Размер данных для записи N | 1 |
| 4 | Данные | N |
| 5 | Контрольная сумма поля 4. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

Ответ от считывателя

Длина сообщения: 1 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |

## Команда FD03h “Считать данные из памяти карты”

Производит чтение данных из памяти эмулируемой карты.

Сообщение к считывателю

Длина сообщения: 5 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: FD03h | 2 |
| 2 | Адрес, начиная с которого будет идти чтение | 2 |
| 3 | Размер данных для чтения N | 1 |

Ответ от считывателя

Длина сообщения: 3 + N байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки | 1 |
| 2 | Данные | N |
| 3 | Контрольная сумма поля 2. Используется CCITT-CRC16.  Полином X^16 + X^12 + X^5 + 1. Начальное значение 0000h.  См. Приложение 1. | 2 |

# Команды протокола версии считывателя для турникета

## Команда E0h “Разрешение/запрет приёма карт”

Команда блокировки второго УКПД с заданием времени блокировки. Заблокированный УКПД на запросы будет получать ответ “Нет карты”.

Сообщение к считывателю

Длина сообщения: 3 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код команды: E0h | 1 |
| 2 | Время блокировки от 0 до 65535.  При передаче значения 0, включённая блокировка будет выключена.  Младший байт передаётся первым. | 2 |

Ответ от считывателя

Длина сообщения: 2 байт.

| **№** | **Описание** | **Размер, байт** |
| --- | --- | --- |
| 1 | Код ошибки  0x00 – всегда нет ошибок. | 1 |
| 1 | Состояние карты.  0 – нет карты.  1 – карта в состоянии Halt.  2 – карта есть и успешно активирована.  3 – карта есть и успешно выполнена авторизация. | 1 |

Приложение 1

Реализация алгоритма вычисление CCITT-CRC16.

uint16\_t crc\_ccitt\_update (uint16\_t crc, uint8\_t data)

{

data ^= (uint8\_t) (crc & 0xff);

data ^= (uint8\_t) (data << 4);

return ((((uint16\_t)data << 8) | ((crc & 0xff00) >> 8))

^ (uint8\_t)(data >> 4)

^ ((uint16\_t)data << 3));

}

uint16\_t CRC16(uint8\_t \*data, uint16\_t sizedata)//Подсчёт CRC

{

uint16\_t i;

uint16\_t crc = 0x0000;//CRC

for(i = 0; i < sizedata; i++)

{

crc = crc\_ccitt\_update(crc, \*(data + i));

}

return crc;

}

Приложение 2

Вычисление адресов AES128 ключей для режимов SL2 и SL3.

/\*\*

\* calculate the coresponding AES key Address

\* Key A even and Key B odd at the Physical AES start address 0x4000U

\*/

#define AES\_SECTOR\_KEYS\_A(sectorNr) (((sectorNr)\*2) + PHYSICAL\_AES\_START\_ADDRESS)

#define AES\_SECTOR\_KEYS\_B(sectorNr) ((((sectorNr)\*2) + 1) + PHYSICAL\_AES\_START\_ADDRESS)

Приложение 3

Ключи AES128 карт Mifare Plus.

/\*\*

\* Block Addresses

\*/

#define CARD\_MASTER\_KEY 0x9000U /\*\*< Card Master Key Address \*/

#define CARD\_CONFIGURATIOM\_KEY 0x9001U /\*\*< Card Configuration Key Address \*/

#define LEVEL\_2\_SWITCH\_KEY 0x9002U /\*\*< Level 2 Switch Key Address \*/

#define LEVEL\_3\_SWITCH\_KEY 0x9003U /\*\*< Level 3 Switch Key Address \*/

#define SL\_1\_CARD\_AUTHENTICATION\_KEY 0x9004U /\*\*< SL1 Card Authentication Key Address \*/

#define SELECT\_VC\_KEY 0xA000U /\*\*< Select VC Key Address \*/

#define PROXIMITY\_CHECK\_KEY 0xA001U /\*\*< Proximity Check Key Address \*/

#define VC\_POLLING\_ENC\_KEY 0xA080U /\*\*< VC Polling ENC Key Address \*/

#define VC\_POLLING\_MAC\_KEY 0xA081U /\*\*< VC Polling MAC Key Address \*/

#define MFP\_CONFIGURATION\_BLOCK 0xB000U /\*\*< MIFARE Plus Configuration block Address \*/

#define INSTALLATION\_IDENTIFIER 0xB001U /\*\*< Installation Identifier Address \*/

#define FIELD\_CONFIGURATION\_BLOCK 0xB003U /\*\*< Field Configuration block Address \*/

#define PHYSICAL\_AES\_START\_ADDRESS 0x4000U /\*\*< physical start address of AES key location Address \*/

Приложение 4

Описание поведения индикации считывателя при включении:

//Всегда: зажглись и погасли 4 светодиода. Если не так то или нет питания, или неисправен/непропаян стабилизатор питания, или неисправен/непоропаян микроконтроллер SMT32F103RCT6

//Всё в порядке: ридер проиграл мелодию (мелодия только когда нет ошибок), не горит светодиодов

//Не отвечает TDA8026 (не пропай на маленькой плате SM11006.07.0x), нет мелодии, загорается синий светодиод

//Нет SAM модулей (не пропай на маленькой плате SM11006.07.0x, нет модулей реально, не исправный модуль), нет мелодии, загорается красный светодиод

//Не отвечает приёмопередатчик PN512 (не пропай), нет мелодии, загораются синий и красный светодиоды

//Ошибка авторизации к SAM модулю (там или записан какой то ключ авторизации, которого нет у ридера или модуль не исправен), нет мелодии, загорается зелёный светодиод

//Не отвечает SAM модуль (в слот установлен не SAM AV2 модуль), нет мелодии, загораются синий и зелёный светодиоды

//Не исправен кварцевый резонатор на 8 МГц (или не пропай), нет мелодии, загорается жёлтый светодиод

Приложение 5

Формат хранения ключей RSA в SAM AV2 модуле.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Длина, байт | Формат | Параметр | Описание |
|  | 1 | unsigned byte | PKI\_KeyNo | Номер записи с ключом |
|  | 2 | bit mask | PKI\_Set | Конфигурационные биты |
| Публичный  ключ | 2 | unsigned byte | PKI\_NLen | Длина ключа в байтах. От 32 до 256 байт, т.е. от 256 до 2048 бит.  Это длина PKI\_N |
| 2 | unsigned byte | PKI\_eLen | Длина публичной экспоненты в байтах. От 4 до 256 байт. Это длина PKI\_e.  Не может быть больше PKI\_N |
| 256 | string | PKI\_N | Модуль RSA (N = p\*q) |
| 256 | string | PKI\_e | Публичная экспонента |
| Приватный  ключ | 248 | string | PKI\_p | Простое число. Длиной PKI\_NLen/2 |
| 248 | string | PKI\_q | Простое число. Длиной PKI\_NLen/2 |
| 248 | string | PKI\_dP | Приватная экспонента dp. Длиной PKI\_NLen/2. CRT представление приватного ключа |
| 248 | string | PKI\_dQ | Приватная экспонента dq. Длиной PKI\_NLen/2. CRT представление приватного ключа |
| 248 | string | PKI\_ipq | Инверсия p-1 mod q. CRT представление приватного ключа |
|  | 1 | unsigned byte | PKI\_KeyNoCEK | Ключ, к которому необходимо авторизоваться перед работой с этой записью.  Значение 0xFE выключает необходимость авторизации |
|  | 1 | unsigned byte | PKI\_KeyVCEK | Версия ключа |
|  | 1 | unsigned byte | PKI\_RefNoCUK | Номер счётчика использования ключа.  Если счётчик не используется пишется значение 0xFF |

Конфигурационные биты:

0 – наличие приватного ключа: 0 нет приватного ключа (не будет сгенерирован), но будет публичный; 1 есть.

1 – разрешение экспорта приватного ключа: 0 запрещён, 1 разрешён.

2 – запретить ключ: 0 запись разрешена, 1 запись запрещена.

3 – запрет использования ключа для шифрования и дешифрования: 0 разрешено, 1 –запрещено.

4 – запрет на использование ключа при работе с ЭЦП: 0 разрешено, 1 запрещено.

5 – разрешение использовать ключ для обновления ключей работы с картами. 0 запрещено, 1 разрешено.

6 – использовать CRT представление при хранении ключа: всегда 1.

7 – 15 – неиспользуемые биты, всегда 0.

Приложение 6

Формат экспортируемого приватного ключа RSA.

|  |  |  |
| --- | --- | --- |
| Параметр | Описание | Размер |
| PKI\_Set | Конфигурационные биты | 2 |
| PKI\_KeyNoCEK | Ключ, к которому необходимо авторизоваться перед работой с этой записью | 1 |
| PKI\_KeyVCEK | Версия ключа | 1 |
| PKI\_RefNoCUK | Номер счётчика использования ключа | 1 |
| PKI\_NLen | Длина ключа в байтах | 2 |
| PKI\_eLen | Длина публичной экспоненты в байтах | 2 |
| PKI\_pLen | Длина простого числа P | 2 |
| PKI\_qLen | Длина простого числа Q | 2 |
| PKI\_N | Модуль RSA (N = p\*q) | PKI\_NLen |
| PKI\_e | Публичная экспонента | PKI\_eLen |
| PKI\_p | Простое число | PKI\_pLen |
| PKI\_q | Простое число | PKI\_qLen |
| PKI\_dP | Приватная экспонента dp | PKI\_pLen |
| PKI\_dQ | Приватная экспонента dq | PKI\_qLen |
| PKI\_ipq | Инверсия p-1 mod q | PKI\_qLen |

Приложение 7

Формат экспортируемого публичного ключа RSA.

|  |  |  |
| --- | --- | --- |
| Параметр | Описание | Размер |
| PKI\_Set | Конфигурационные биты | 2 |
| PKI\_KeyNoCEK | Ключ, к которому необходимо авторизоваться перед работой с этой записью | 1 |
| PKI\_KeyVCEK | Версия ключа | 1 |
| PKI\_RefNoCUK | Номер счётчика использования ключа | 1 |
| PKI\_NLen | Длина ключа в байтах | 2 |
| PKI\_eLen | Длина публичной экспоненты в байтах | 2 |
| PKI\_N | Модуль RSA (N = p\*q) | PKI\_NLen |
| PKI\_e | Публичная экспонента | PKI\_eLen |

Приложение 8

Формат импортируемого RSA ключа.

|  |  |  |
| --- | --- | --- |
| Параметр | Описание | Размер |
| PKI\_Set | Конфигурационные биты | 2 |
| PKI\_KeyNoCEK | Ключ, к которому необходимо авторизоваться перед работой с этой записью | 1 |
| PKI\_KeyVCEK | Версия ключа | 1 |
| PKI\_RefNoCUK | Номер счётчика использования ключа | 1 |
| PKI\_NLen | Длина ключа в байтах. Значение от 32 до 256 байт.  Имеет значение 0 при обновлении настроек ключа. | 2 |
| PKI\_eLen | Длина публичной экспоненты в байтах. Значение от 4 до 256 байт. Не должно быть больше PKI\_NLen.  Имеет значение 0 при обновлении настроек ключа. | 2 |
| PKI\_pLen | Длина простого числа P. Значение от 6 до 252 байт.  Поле отсутствует при обновлении настроек ключа или при импорте публичного ключа. Присутствует только при импорте приватного ключа | 2 |
| PKI\_qLen | Длина простого числа Q. Значение от 6 до 252 байт.  Поле отсутствует при обновлении настроек ключа или при импорте публичного ключа. Присутствует только при импорте приватного ключа | 2 |
| PKI\_N | Модуль RSA (N = p\*q).  Поле отсутствует при обновлении настроек ключа | PKI\_NLen |
| PKI\_e | Публичная экспонента.  Поле отсутствует при обновлении настроек ключа | PKI\_eLen |
| PKI\_p | Простое число.  Поле отсутствует при обновлении настроек ключа или при импорте публичного ключа. Присутствует только при импорте приватного ключа | PKI\_pLen |
| PKI\_q | Простое число.  Поле отсутствует при обновлении настроек ключа или при импорте публичного ключа. Присутствует только при импорте приватного ключа | PKI\_qLen |
| PKI\_dP | Приватная экспонента dp.  Присутствует только при импорте приватного ключа | PKI\_pLen |
| PKI\_dQ | Приватная экспонента dq.  Присутствует только при импорте приватного ключа | PKI\_qLen |
| PKI\_ipq | Инверсия p-1 mod q.  Присутствует только при импорте приватного ключа | PKI\_qLen |

Приложение 9

Блок-значение (Value блоки) это формат записи данных на карту, допускающий операции инкремента, декремента, чтения, записи, сохранения и восстановления данных с контролем их целостности. При этом для операций с данными определяются права доступа по ключам (A и B).

Блок-значение может быть создан только при операциях записи (write) и сохранения из временного регистра (transfer). При этом в трейлере сектора этот блок должен быть указан как блок-значение.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер байта | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Описание | Value | | | | Value | | | | Value | | | | Adr | Adr | Adr | Adr |

Блок-значение хранит:

1. Значение Value. 32-х битное значение со знаком в дополнительном коде. Для обеспечения контроля целостности хранится в трёх копиях: две не инвертированных и одну инвертированную. Именно это значение участвует в операциях инкремента и декремента. Младший значащий байт значения хранится в младшем по адресу байте.
2. Адрес Adr. Внутренне (картой) никак не интерпретируется. Используется для хранения адреса блока. Например, адреса другого блока, в котором лежит копия этого. Не изменяется при выполнении команд инкремента и декремента. Может быть изменён командами записи (write) или сохранения (transfer).

Операции с блоком-значением:

1. Инкремент (increment). Данные в блоке проверяются (картой) и считываются во временный регистр (карты). Затем данные в регистре увеличиваются на заданное число. Результат хранится во временном регистре. Если данные повреждены, то возникает ошибка инкремента.
2. Декремент (decrement). Данные в блоке проверяются (картой) и считываются во временный регистр (карты). Затем данные в регистре уменьшаются на заданное число. Результат хранится во временном регистре. Если данные повреждены, то возникает ошибка декремента.
3. Восстановление (restore). Данные проверяются и считываются во временный регистр. Если данные повреждены, то возникает ошибка восстановления.
4. Сохранение (transfer). Данные из временного регистра сохраняются в блок-значение. Если данные в регистре не валидны (не были туда записаны), то происходит ошибка сохранения. По сути это команда сохранения результата других команд.
5. Чтение (read). Команда чтения всех 16-ти байт блока. Проверок целостности не выполняется, это задача ПО обрабатывающего данные карты.
6. Запись (write). Команда записи 16 байт данных в блок. Проверок целостности не выполняется, это задача ПО обрабатывающего данные карты.

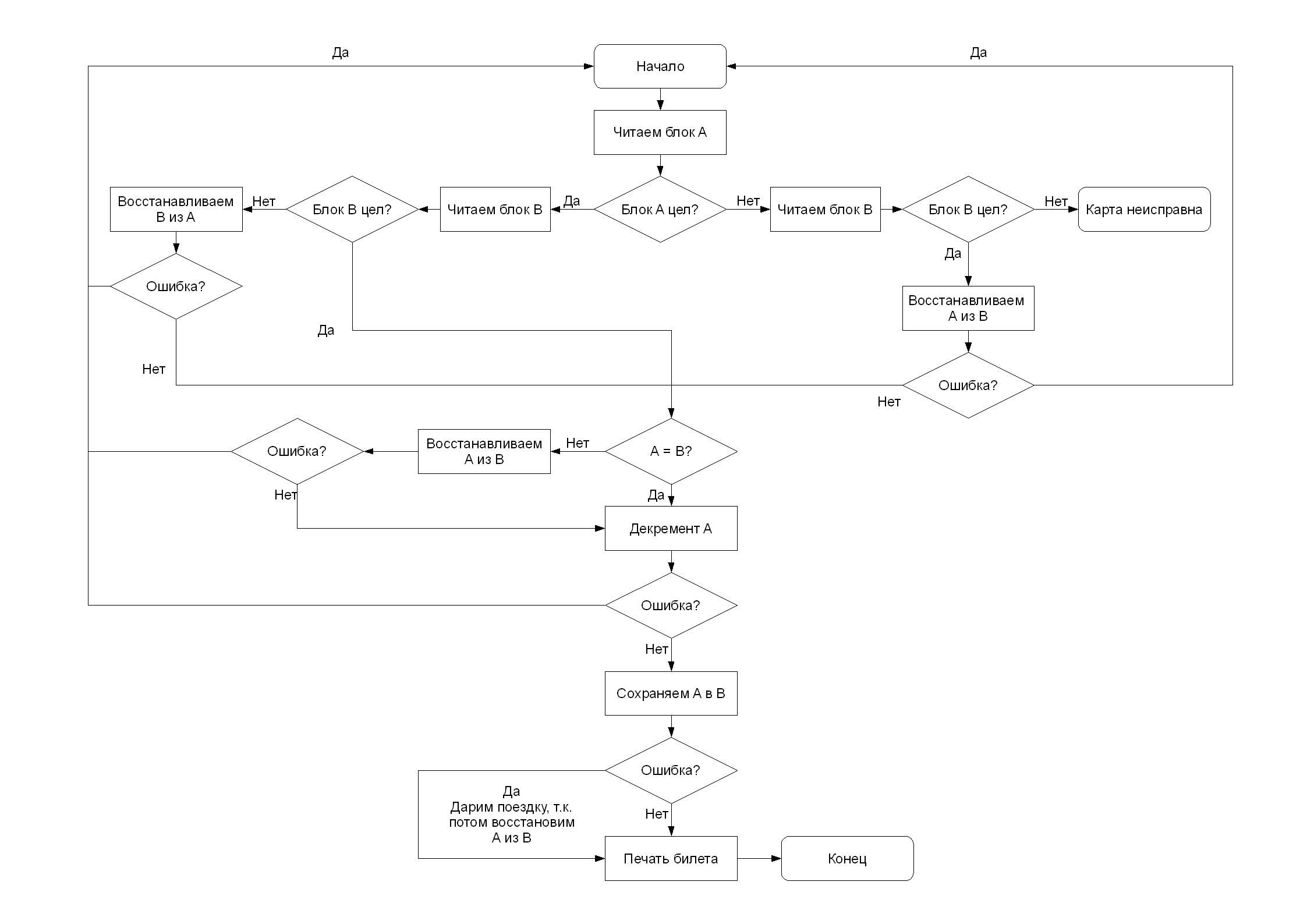
Совмещение команд Восстановления и Сохранения, позволяет переписать данные из одного блока (A) в другой (B). При этом для операции Восстановления должен быть указан блок A, а при операции Сохранения блок B.

Данные в блоках карты (любых блоков) могут быть повреждены при выполнении операции записи, если они не будут завершены. Если карта будет убрана из поля считывателя при выполнении операции.

Описание типовой работы с блоками-значениями, которое позволяет обеспечить целостность данных:

1. Данные хранятся в двух блоках одного сектора. Условно блок A и блок B.
2. Изначально эти два блока записываются при эмиссии карт или при добавлении услуги, командой записи (rite). Для этого нужно использовать ключ авторизации, который имеет права на запись.
3. Перед выполнением операций инкремента или декремента данные блоков должны быть считаны (read). Если данные в одном блоке повреждены, то он должен быть восстановлен из другого. Если повреждён блок A, то командой restore данные из B читаются в регистр и затем записываются в A командой transfer. И наоборот, если повреждён блок B. Если повреждены оба блока, то восстановление не возможно.
4. Если оба блока валидны, но значения в них отличаются, то значение из одного них (большего или меньшего) восстанавливается таким же образом. Какое значение при этом считать верным зависит от разработчиков системы.
5. Если значения блоков верны, то выполняем операцию инкремента или декремента над одним из блоков (например A). При этом во временном регистре будет результат.
6. Сохраняем (transfer) значение в другой блок (B). В этом блоке будет новое значение.
7. Копируем данные из второго блока (B) в первый (A) используя операции restore и transfer.

Такая работа с блоками-значениями позволяет всегда иметь одну целую копию данных.



Приложение 10

Адреса ключей в считывателе для внешнего УКПД и старой версии (до 2.0.57a) прошивки считывателя вычисляются как:

**Key Addr =100 + Card Sector + (Key Type \* 2)**

В таблице описан пример хранения ключей для внешнего УКПД.

|  |  |  |
| --- | --- | --- |
| Сектор карты | Тип ключа | Ячейка считывателя |
| 2 (блоки 8 - 11) | 0 | 102 |
| 3 (блоки 12 - 15) | 0 | 103 |
| 2 (блоки 8 - 11) | 1 | 104 |
| 3 (блоки 12 - 15) | 1 | 105 |
| 2 (блоки 8 - 11) | 2 | 106 |
| 3 (блоки 12 - 15) | 2 | 107 |
| 2 (блоки 8 - 11) | 3 | 108 |
| 3 (блоки 12 - 15) | 3 | 109 |

Нужно учитывать, что реально номера ключей сдвинуты на 1. Ключ 0 при записи через считыватель это на деле ключ 1 в SAM AV2 модуле. Так сбыло сделано для совместимости со старыми считывателями. В этом считывателе ключ SAM AV2 с номером 0, используется для авторизации к SAM AV2 модулю.

Обычный проездной – параметр 01.

ТТ – 03.

Для всех новых версий прошивки производиться циклический перебор ключа от 100-й ячейки до 111-й. Перебор происходит, если текущим ключом не удалось авторизоваться к сектору карты. От текущего номера до первого подошедшего, который станет текущим.

Приложение 11

**Для всех описанных ниже действий нужно обеспечить секретность Master Key для Master SAM.**

Описание последовательности создания Master SAM:

1. Чистый SAM AV2 модуль установить в считыватель, включить его и дождаться его перезагрузки (моргания светодиодами). При этом считыватель сам переведёт SAM модуль в режим AV2 и при этом пропишет на место Master Key нули во все три позиции, с версиями 0,1,2. Master Key находиться в 0-й ячейка SAM AV2 модуля.
2. Заменить во всех трёх позициях значение Master Key (команда 2Bh). Версии при этом должны отличатся, но в нулевой позиции должна быть версия 0. После каждой замены ключа нужно авторизоваться Master Key с версией 0. **Если не заменить ключ хоть в одной позиции, то это будет уязвимость. У злоумышленника получившего Master SAM будет возможность считать ключи с выставленным битом Allow Dump Secret Key, используя ключ по умолчанию.**
3. Авторизоваться к Master ключу (команда 2Ah). Проще к версии 0, т.к. эта версия задана по умолчанию на изменение других ключей.
4. Записать ключи в SAM AV2 модуль, предварительно включая бит Allow Dump Secret Key. Для этого есть команда 66h с кодом операции 00h. Бит необходимо выставлять перед каждой записью ключа, т.к. он автоматически сбрасывается.

Описание последовательности записи агентского SAM модуля с использованием Master SAM:

1. Чистый SAM AV2 модуль установить в считыватель, включить его и дождаться его перезагрузки (моргания светодиодами). При этом считыватель сам переведёт SAM модуль в режим AV2 и при этом пропишет на место Master Key нули во все три позиции, с версиями 0,1,2. Master Key находиться в 0-й ячейка SAM AV2 модуля. Можно установить чистые SAM AV2 модули во все слоты считывателя (2 или 4 в зависимости от модели) и дождаться последовательных перезагрузок считывателя (2 или 4).
2. Записать для агентского SAM модуля все Master Key (команда 2Bh). В нулевой позиции должна быть версия 0.
3. Установить в другой слот Master SAM. Если при этом нужно было извлечь другой SAM, то нужно сбросить считыватель (команда F1h).
4. Переключить слот на тот, в котором установлен Master SAM (команда 27h).
5. Авторизоваться к Master SAM используя Master Key (команда 2Ah).
6. Считать копируемый ключ в память контроллера считывателя (команда 66h, операция 01h для Crypto1 или операция 02h для PICC AES/DES).
7. Переключить слот на слот с агентским считывателем (команда 27h).
8. Авторизоваться к Master SAM используя Master Key (команда 2Ah) с версией 0.
9. Подать команду записи ключа в агентский считыватель. Для Mifare Crypto1 это команда 31h. Для PICC DES это 24h. И для PICC AES - 48h.
10. Повторить пункты от 4 до 9 для всех копируемых ключей.

Приложение 12

Реализация функции кодирования ключа для команды 0Ch.

uint8\_t i, j, k;

j = 0;//Индекс в выходном кодированном ключе

for(i = 0; i < 6; i++)//Цикл кодирования ключа

{

k = ~key[i];//Инвертируем байт ключа

//Суммируем части кодированного ключа

coded\_key[j] = (key[i] & 0xF0) + (key[i] >> 4);

coded\_key[j + 1] = ((key[i] & 0x0F) << 4) + (key[i] & 0x0F);

j += 2;//Инкремент индекса

}

Приложение 13

Коды ошибок выводимые на светодиод статуса или на один из светодиодов считывателя. В случае возникновения ошибки после зажигания светодиода на 350 мс, он зажигается на 3 секунды и затем после паузы в 1 секунду выдает вспышками код ошибки.

В таблице указаны коды ошибок данного считывателя и наиболее вероятные причины ошибок.

|  |  |
| --- | --- |
| Код ошибки | Описание ошибки |
| 0 | Нет ошибки. Светодиод зажигается один раз на 350 мс. |
| 1 | Не отвечает мультиплексор для SAM AV2 модулей TDA8026. Возможные причины: не пропаяна плата считывателя. |
| 2 | Нет SAM AV2 модулей. Возможные причины: SAM модуль не установлен, не пропаян слот, не пропаяна плата считывателя. |
| 3 | Не отвечает приёмопередатчик PN512. Возможная причина: дефект пайки. |
| 4 | Ошибка авторизации к SAM модулю. Возможная причина: установлен SAM модуль с записанным ключом авторизации. |
| 5 | Нет SAM модулей. Возможные причины: SAM модуль не установлен, не пропаян слот, не пропаяна плата считывателя. |
| 7 | Ошибка кварцевого резонатора микроконтроллера. Возможные причины: не исправен кварцевый резонатор или не пропаяна плата считывателя. |
| 9 | Перегрузка по питания SAM модуля. Возможные причины: не исправный SAM модуль, КЗ на плате считывателя. |
| 10 | Ошибка супервизора SAM модуля (провал по питанию). Возможные причины: не пропаяна плата считывателя, не исправный стабилизатор на 3.3 В. |
| 11 | Ошибка микросхемы Flash памяти. Возможные причины: не исправна или не пропаяна микросхема памяти. |
| 12 | Ошибочные записи в микросхеме Flash памяти. Нужно выключить питание и включить повторно. Если ошибка сохраняется, то память не исправна. |

Приложение 14

Шифрование обмена считывателя. Для шифрования обмена используется алгоритм по ГОСТ 28147-89 (МАГМА) в режиме гаммирования.

Реализация ниже. Использована функция EncryptDecryptBlockGamma.

//Константы режима гаммирования

const *uint32\_t* C1 = 0x1010104;

const *uint32\_t* C2 = 0x1010101;

//Таблица замен

//Узел замены, определенный Техническим комитетом по стандартизации "Криптографическая защита информации" (сокращенно - ТК 26) Росстандарта

*uint8\_t* SBlocks[8][16] =

{0x0C, 0x04, 0x06, 0x02, 0x0A, 0x05, 0x0B, 0x09, 0x0E, 0x08, 0x0D, 0x07, 0x00, 0x03, 0x0F, 0x01,

0x06, 0x08, 0x02, 0x03, 0x09, 0x0A, 0x05, 0x0C, 0x01, 0x0E, 0x04, 0x07, 0x0B, 0x0D, 0x00, 0x0F,

0x0B, 0x03, 0x05, 0x08, 0x02, 0x0F, 0x0A, 0x0D, 0x0E, 0x01, 0x07, 0x04, 0x0C, 0x09, 0x06, 0x00,

0x0C, 0x08, 0x02, 0x01, 0x0D, 0x04, 0x0F, 0x06, 0x07, 0x00, 0x0A, 0x05, 0x03, 0x0E, 0x09, 0x0B,

0x07, 0x0F, 0x05, 0x0A, 0x08, 0x01, 0x06, 0x0D, 0x00, 0x09, 0x03, 0x0E, 0x0B, 0x04, 0x02, 0x0C,

0x05, 0x0D, 0x0F, 0x06, 0x09, 0x02, 0x0C, 0x0A, 0x0B, 0x07, 0x08, 0x01, 0x04, 0x03, 0x0E, 0x00,

0x08, 0x0E, 0x02, 0x05, 0x06, 0x09, 0x01, 0x0C, 0x0F, 0x04, 0x0B, 0x00, 0x0D, 0x0A, 0x03, 0x07,

0x01, 0x07, 0x0E, 0x0D, 0x00, 0x05, 0x08, 0x03, 0x04, 0x0F, 0x0A, 0x06, 0x09, 0x0C, 0x0B, 0x02};

//Для вычисления Гаммы

*uint32\_t* N1 = 0, N2 = 0, N3 = 0, N4 = 0, N5 = 0;

*uint32\_t* Substitute(*uint32\_t* value)//Функция подстановки по S блокам

{

*uint8\_t* i, index, sblock;

*uint32\_t* result = 0;

*uint32\_t* tmp;

for(i = 0; i < 8; i++)

{

index = (value >> (4 \* i)) & 0x0F;

sblock = SBlocks[i][index];

tmp = sblock;

tmp = tmp << (4 \* i);

result = result | tmp;

}

return result;

}

*uint32\_t* EcryptFuction(*uint32\_t* data, *uint32\_t* subkey)//Функция шифрования

{

data = (data + subkey) & 0xFFFFFFFF;//Сумма

data = Substitute(data);//Перестановки

data = (data << 11) | (data >> 21);//Циклический сдвиг на 11 бит

return data;

}

*uint8\_t* GetKeyIndex(*uint8\_t* round, *uint8\_t* mode)//Вычисление индекса ключа

{

//В зависимости от режима mode

//True - шифрование

//X0,X1,X2,X3,X4,X5,X6,X7,X0,X1,X2,X3,X4,X5,X6,X7,X0,X1,X2,X3,X4,X5,X6,X7,X7,X6,X5,X4,X3,X2,X1,X0

//False - расшифрование

//X0,X1,X2,X3,X4,X5,X6,X7,X7,X6,X5,X4,X3,X2,X1,X0,X7,X6,X5,X4,X3,X2,X1,X0,X7,X6,X5,X4,X3,X2,X1,X0

//round - раунд от 0 до 31

*uint8\_t* result = 0;

if(mode)

{

if(round < 24)

{

result = round % 8;

}

else

{

result = 7 - (round % 8);

}

}

else

{

if(round < 8)

{

result = round % 8;

}

else

{

result = 7 - (round % 8);

}

}

return result;

}

*uint8\_t* EncryptDecryptGOST(*uint32\_t* \*A, *uint32\_t* \*B, const *uint32\_t* \*key, *uint8\_t* mode)//Шифрование по ГОСТ

{

*uint8\_t* i;

*uint8\_t* key\_index;

*uint32\_t* sub\_key;

*uint32\_t* func\_value;

*uint32\_t* round;

*uint32\_t* Aint, Bint;

*uint8\_t* result = 0;

if((A != *NULL*) && (B != *NULL*) && (key != *NULL*))

{

Aint = \*A;

Bint = \*B;

for(i = 0; i < 32; i++)

{

key\_index = GetKeyIndex(i, mode);

sub\_key = key[key\_index];

func\_value = EcryptFuction(Aint, sub\_key);

round = func\_value ^ Bint;

if(i < 31)

{

Bint = Aint;

Aint = round;

}

else

{

Bint = round;

}

}

\*A = Aint;

\*B = Bint;

}

else

{

result = 255;

}

return result;

}

//Вычисление начальной Гаммы

*uint8\_t* CalcInitGamma(*uint32\_t* InitN1, *uint32\_t* InitN2, const *uint32\_t* \*key)

{

*uint8\_t* result;

N1 = InitN1;

N2 = InitN2;

result = EncryptDecryptGOST(&N1, &N2, key, 0xFF);//Шифруем синхропосылку

N3 = N1;

N4 = N2;

return result;

}

//Вычисление Гаммы

*uint8\_t* CalcGamma(const *uint32\_t* \*key)

{

*uint8\_t* result;

//Вычисление гаммы

N3 = (N3 + C2) & 0xFFFFFFFF;//Суммируем N3 с C2 по модулю 2^32

N5 = (N4 + C1) & 0xFFFFFFFF;//Суммируем N4 с C1 по модулю 2^32 - 1, как в ГОСТ, а не как в математике

if((N5 < N4) | (N5 < C1))//Проверка переполнения

{

N5 = N5 + 1;

}

N4 = N5;

N1 = N3;

N2 = N4;

result = EncryptDecryptGOST(&N1, &N2, key, 0xFF);

return result;

}

*uint8\_t* EncryptDecryptGOSTGamma(*uint32\_t* \*A, *uint32\_t* \*B)//Шифрование/дешифрование Гаммой

{

*uint16\_t* result = 0;

if((A != *NULL*) && (B != *NULL*))

{

\*A = (\*A) ^ N1;

\*B = (\*B) ^ N2;

}

else

{

result = 255;

}

return result;

}

*uint8\_t* EncryptDecryptBlockGamma(*uint8\_t* \*data, *uint32\_t* data\_size, *uint32\_t* initGamma1 ,*uint32\_t* initGamma2, const *uint32\_t* \*key)//Шифрование массива гаммой

{

*uint8\_t* result = 0;

*uint32\_t* A = 0, B = 0;//Для шифрования и дешифрования

*uint8\_t* i = 0, j = 0;//Счётчики байт

*uint32\_t* itr = 0, itw = 0;//Счётчик для чтения данных и счётчик записи данных

//Проверка аргументов

if((data != *NULL*) && (data\_size != 0) && (key != *NULL*))

{

//Инициализация Гаммы

CalcInitGamma(initGamma1, initGamma2, key);

//Цикл шифрования/дешифрования

while(itr < data\_size)

{

//Выделим 4 байта в A

if(i <= 3)

{

A = A << 8;

A = A | data[itr];

}

//Выделим 4 байта в B

if((i >= 4) & (i <= 7))

{

B = B << 8;

B = B | data[itr];

}

//Если выделили 8 байт или данные закончились

if((i == 7) || ((itr + 1) >= data\_size))

{

//Вычисление Гаммы

CalcGamma(key);

//Досдвиг данных, что бы не потерять

//Т.к. ниже выделяю со старшего байта сдвингая на 24 бита

if(i < 3)

{

A = A << ((3 - i) \* 8);

}

if((i > 3) && (i < 7))

{

B = B << ((7 - i) \* 8);

}

//Шифруем

EncryptDecryptGOSTGamma(&B, &A);

//Пишем данные назад в массив с учётом его длины

while(itw <= itr)

{

//Пишем 4 байта из A

if(j <= 3)

{

data[itw] = (A >> 24) & 0xFF;

A = A << 8;

}

//Пишем 4 байта из B

if((j >= 4) & (j <= 7))

{

data[itw] = (B >> 24) & 0xFF;

B = B << 8;

}

//Циклический подсчёт выделеных байт

j++;

if(j > 7)

{

j = 0;

}

//Считаем записанные байты

itw++;

}

//Сброс A и B

A = 0;

B = 0;

}

//Циклический подсчёт выделеных байт

i++;

if(i > 7)

{

i = 0;

}

//Считаем вычитанные байты

itr++;

}

}

else

{

result = 255;

}

return result;

}

*uint8\_t* EncryptDecryptBlock(*uint8\_t* \*data, *uint32\_t* data\_size, const *uint32\_t* \*key, *uint8\_t* mode)//Шифрование массива простой заменой

{

*uint16\_t* result = 0;

*uint32\_t* A = 0, B = 0;//Для шифрования и дешифрования

*uint8\_t* i = 0, j = 0;//Счётчики байт

*uint32\_t* itr = 0, itw = 0;//Счётчик для чтения данных и счётчик записи данных

//Проверка аргументов

if((data != *NULL*) && (data\_size != 0) && (key != *NULL*) && ((data\_size % 8) == 0))

{

//Цикл шифрования/дешифрования

while(itr < data\_size)

{

//Выделим 4 байта в A

if(i <= 3)

{

A = A << 8;

A = A | data[itr];

}

//Выделим 4 байта в B

if((i >= 4) & (i <= 7))

{

B = B << 8;

B = B | data[itr];

}

//Если выделили 8 байт или данные закончились

if((i == 7) || ((itr + 1) >= data\_size))

{

//Шифруем

EncryptDecryptGOST(&B, &A, key, mode);

//Пишем данные назад в массив с учётом его длины

while(itw <= itr)

{

//Пишем 4 байта из A

if(j <= 3)

{

data[itw] = (A >> 24) & 0xFF;

A = A << 8;

}

//Пишем 4 байта из B

if((j >= 4) & (j <= 7))

{

data[itw] = (B >> 24) & 0xFF;

B = B << 8;

}

//Циклический подсчёт выделеных байт

j++;

if(j > 7)

{

j = 0;

}

//Считаем записанные байты

itw++;

}

//Сброс A и B

A = 0;

B = 0;

}

//Циклический подсчёт выделеных байт

i++;

if(i > 7)

{

i = 0;

}

//Считаем вычитанные байты

itr++;

}

}

else

{

result = 255;

}

return result;

}

История версий прошивки и протокола

1.56 – добавлена команда 0xF8.

1.58 – добавлен байт кода ошибки (ошибки старта) в команде 0xF8.

1.58 for 3K5 – добавлены команды работы с моторизованным считывателем Sankyo 3K5.

1.59 – добавлены команды работы с My-D Move.

1.60 – изменены команды записи, чтения, инкремента, декремента, пересылки для Mifare Plus. Добавлены параметры шифрования данных и подписи ответа.

1.61 – 1.65 – латание багов.

1.66 – добавлена команда 0x48 – запись ключа AES128, изменена команда 0x47. Добавлены команды 0x49, 0x4A, 0x4B, 0x4C. Изменение команд 0x3A – 0x43, добавлены параметры шифрования и подписи.

1.67 – изменение в командах 0x3A, 0x3C – добавлен контроль длины сообщения. Исправил описание команд 0x45, 0x46. Исправил команды 0x45, 0x46.

1.69 – добавлены команды для вычисления хэш и для оффлайн криптографии.

1.70 – 1.75 – добавлены команды для генерации, экспорта и импорта RSA ключей.

1.77 – добавлено вычисление хэш для блоков до 2048 байт.

1.78 – исправлена работа с картами с 7-ми байтными UID.

1.80 – изменение адреса старта, для работы с BootLoader.

1.81 – добавлена команда записи UID считывателя.

1.83 – изменения в хранении настроек. Учтено выравнивание структур памяти.

1.84 – 1.86 – добавление команд MIFARE Ultralight EV1.

1.87 – добавление настройки ротации светодиодов, для исправления бага платы антенны.

1.88 – 1.90 – добавление команд работы с KUC и CEK.

1.91 – исправлена ошибка авторизации к картам с 7-и байтным UID и исправлена запись ключей.

1.92 – добавлены команды вычисления MAC.

1.93 – исправлена работа с защитой (авторизацией) SAM AV2 модуля. Теперь выставляются биты запрещающие не только записывать ключи без авторизации, но и использовать их.

1.95 – добавлена диагностика КЗ и Supervisor по питанию SAM AV2 модулей через TDA8026.

1.96 – добавлены команды турникета. Увеличен порог напряжения супервизора микроконтроллера с 2.8 до 2.9 В.

1.97 – реализация списания поездок в турникете с карт Ultralight C.

1.98 – изменение порядка инициализации считывателя.

1.981 – добавление повторных попыток инициализации SAM AV2 модуля. 10 попыток с паузой 200 мс.

1.982 – добавление возможности энергосбережения в прошивку. Выключение поля когда нет карты. Включается директивой условной компиляции.

1.983 – добавлены команды для турникета.

1.984 – добавлена команда для эмиссии 26h. Запись данных с дешифрованием.

1.985 – доработка команды эмиссии.

1.986 – доработка команд для турникета.

1.987 – выкинут протокол для отдела Систем Самообслуживания. Добавлена блокировка доступа двух устройств при параллельном доступе к считывателю. Первое успешно авторизовавшееся получит доступ, второе будет получать фиктивный ответ ‘Нет карты’.

1.988 – исправлена ошибка протокола для турникета. Прописаны правильные состояния карты в условии авторизации.

1.989 – исправление команд работы с блок значениями Mifare Plus.

1.990 – реализована работа с микросхемой Flash памяти.

2.0.0 – смена библиотеки работы с картами на более новую. Изменение нумерации версии прошивки. Нумерация теперь ведётся по схеме A.B.C.D.[r].

Формат номера версии A.B.C[r], где:

• A – главный номер версии (major version number).

• B – вспомогательный номер версии (minor version number).

• C – номер сборки, номер логической итерации по работе над функционалом версии A.B (build number).

• [r] – условное обозначение релиза. Присутствует не всегда.

Совокупность главного и вспомогательного номеров версии (A.B) дают информацию о функционале приложения. Главный номер версии увеличивается только при очень серьёзном изменении функционала.

Номер билда (С) должен увеличиваться руководителем проекта по разработке всякий раз, когда продукт передаётся на тестирование.

Обозначение релиза соответствует этапу работы над проектом в рамках жизненного разработки. Выделяют следующие релизы:  
• **Pre-alpha** (pa) – соответствует этапу начала работ над версией. Характеризуется большими изменениями в функционале и большим количеством ошибок. Pre-alpha релизы не покидают отдела разработки ПО.  
• **Alpha**(a) – соответствует этапу завершения разработки нового функционала. Начиная с alpha версии новый функционал не разрабатывается, а все заявки на новый функционал уходят в план работ по следующей версии. Этап характеризуется высокой активностью по тестированию внутри подразделения разработки ПО и устранению ошибок.  
• **Beta** (b) – соответствует этапу публичного тестирования. Это первый релиз, который выходит за пределы отдела разработки ПО. На этом этапе принимаются замечания от пользователей по интерфейсу продукта и прочим найденным пользователями ошибкам и неточностям.  
• **Release Candidate** (rc) – весь функционал реализован и полностью оттестирован, все найденные на предыдущих этапах ошибки исправлены. На этом этапе могут вноситься изменения в документацию и конфигурации продукта.  
• **Release to manufacturing** или **Release to marketing** (rtm) – служит для индикации того, что ПО соответствует всем требованиям качества, и готово для массового распространения. RTM не определяет способа доставки релиза (сеть или носитель) и служит лишь для индикации того, что качество достаточно для массового распространения.  
• **General availability** (ga) – финальный релиз, соответствующий завершению всех работ по коммерциализации продукта, продукт полностью готов к продажам через веб или на физических носителях.  
• **End of life** (eol) – работы по развитию и поддержке продукта завершены.

2.0.1pa – унификация индикации ошибок. Индикация производится одним светодиодом.

В случае возникновения ошибки после зажигания светодиодов (или светодиода, если он один) на 350 мс, они зажигается на 3 секунды и затем после паузы в 1 секунду выдает вспышками код ошибки. Затем идёт пауза в 3 секунды и индикация повторяется с зажигания на 3 секунды.

2.0.2pa – изменение команды сброса для версии турникета.

2.0.3pa – добавление команд NFC, ISO18092.

2.0.4pa – исправление проблемы интерфейса связи с хостом.

Была ошибка пропадания связи при попытке связи без чётности. После этого связь нарушалась (при чётности). Причина – неправильный приём длины пакета. Введён контроль чётности.

2.0.5pa – исправление ошибки появившейся в 2.0.4pa. Ошибка ложного срабатывания контроля чётности. Приводило к эпизодическим пропаданиям связи.

07.04.2014 16:58 – добавление описания Value блоков.

2.0.6pa – обновление библиотеки для реализации эмуляции карт, для реализации передачи и приёма данных по ISO14443a напрямую между считывателями.

2.0.7pa – увеличение скорости обмена контроллера и приёмопередатчика с 460800 бит/сек до 921600 бит/сек.

Добавление команд эмуляции карт: включить эмуляцию 0xFD 0x00, выключить эмуляцию 0xFD 0x01.

25.04.2014 18:58 – добавление описания команд эмуляции карт.

2.0.8pa – изменение способа выдачи данных хосту. Данные теперь передаются не в цикле, а по прерыванию TXE. Введено для освобождения контроллера для эмуляции карт.

2.0.9pa – исправлена ошибка с ответом при смене скорости обмена командой 0xF5h. Из-за передачи данных по прерыванию, скорость менялась до передачи всего ответа. Введён контроль окончания передачи. Скорость теперь меняется после отправки ответа.

2.0.10pa –увеличена скорость обмена контроллера и SAM AV2 Модулей до 250000 бит/сек. Для этого было изменено значение Di (ISO/IEC 7816).

Добавлена команда 0x33 – чтение или запись сектора с предварительной авторизацией к нему.

2.0.11pa –увеличена частота тактирования SAM AV2 модулей до 4 МГц. Скорость обмена та же (250000 бит/сек).

Отключен лог для работы с картой. Это привело к уменьшению прошивки на 20 кбайт.

2.0.12pa –оптимизация функции чтения данных от приёмопередатчика PN512. Результат – уменьшение задержек при работе с картой.

2.0.15pa –оптимизация функции чтения данных от приёмопередатчика PN512. Результат – уменьшение задержек при работе с картой.

2.0.16pa –оптимизация функции чтения данных от SAM AV2 модуля. Результат – уменьшение задержек при работе с картой и SAM AV2 модулем.

2.0.17pa –добавлена авторизация к SAM AV2 модулю вычисляемым ключом при не удаче авторизации записанным.

2.0.18pa – оптимизация функции чтения данных от SAM AV2 модуля и оптимизация функции чтения данных от приёмопередатчика PN512.

В исходные коды добавлена возможность установить частоту SAM AV2 модуля в 8 МГц. Проверена возможность работы с SAM AV2 модулем при скорости обмена 1 Мбит/сек.

Добавлены и проверены скорости обмена с PN512 968.571 кбит/сек и 1130.000 кбит/сек.

При частоте SAM AV2 модуля 8 МГц, скорости обмена 1 Мбит/сек и скорости обмена с PN512 1130.000 кбит/сек, скорости обмена с ПК 115200 бит/сек, достигнуты следующие временна работы с картой:

* Активация карты с 4 байтным номером 7.6 мс.
* Чтение сектора с авторизацией (команда 33h) 25.6 мс.
* Запись сектора с авторизацией (команда 33h) 48.8 мс.

В версии 2.0.12pa, при скорости обмена с ПК 57600 бит/сек, частоте SAM AV2 модуля 4 МГц, скорости обмена 0.5 Мбит/сек и скорости обмена с PN512 921.6 кбит/сек было:

* Чтение сектора с авторизацией (команда 33h) 90 мс.
* Запись сектора с авторизацией (команда 33h) 130 мс.

В версии 2.0.11pa, при скорости обмена с ПК 57600 бит/сек, частоте SAM AV2 модуля 4 МГц, скорости обмена 0.5 Мбит/сек и скорости обмена с PN512 921.6 кбит/сек было:

* Чтение сектора с авторизацией (команда 33h) 109 мс.
* Запись сектора с авторизацией (команда 33h) 157 мс.

В модификации прошивки 2.0.18pa для диспенсера отключено энергосбережение, т.к. поле не успевает включаться при активации карт. В результате карты Mifare Plus и УЭК были не видны.

2.0.19pa –исправление ошибки обмена с SAM AV2 модулем. Ошибка появилась в 2.0.18pa. Для увеличения быстродействия было закомментирован сброс буфера обмена после передачи данных.

2.0.20pa – скорость обмена с SAM AV2 модулем установлена в 500 кбит/сек. Это обусловлено нестабильностью обмена с SAM AV2 на скорости 1 мбит/cек.

Частота тактирования SAM AV2 для терминала оплаты проезда и для RS232/USB считывателей 4 МГц, для диспенсера и турникета 8 МГц.

2.0.21pa – выяснена и исправлена причина ошибок на скорости 1 мбит/сек. Проблема была в коде ожидания последнего переданного байта принятого приёмной частью USART.

Вместо этого ожидания прерыванию на приём передаётся число передаваемых байт. Прерывание игнорирует эти байты.

Кроме того оптимизирован код.

При частоте SAM AV2 модуля 8 МГц, скорости обмена 1 Мбит/сек и скорости обмена с PN512 1130.000 кбит/сек, скорости обмена с ПК 115200 бит/сек, достигнуты следующие временна работы с картой:

* Активация карты с 4 байтным номером 7.2 мс.
* Чтение сектора с авторизацией (команда 33h) 26.2 мс.
* Запись сектора с авторизацией (команда 33h) 45.6 мс.

На 35 мс ускорен старт считывателя.

Добавлена поддержка скорости обмена с ПК 256000 бит/сек.

Добавлена возможность компиляции под Актуализатор.

2.0.22pa – добавлена команда авторизации по ключу переданному от хоста 0Ch.

Увеличено до 29 число проигрываемых мелодий по команде 0x20h.

Ключи в памяти расположены так, что 0-й занят под авторизацию к SAM AV2 модулю. В памяти 128 ячеек ключей. 0-й - для SAM AV2, 127-й - для команды 0Ch. Пользователю доступны от 1-го до 126-го. Со сдвигом на 1. Т.е. для него от 0 до 125

Ключи могут быть как Crypto1 так и AES, DES. Но Crypto1 могут храниться парами A и B.

Всего три позиции (версии) ключа в ячейке.

Добавлена возможность работы без SAM AV2 модуля с хранением ключей в SRAM контроллера. При частоте SAM AV2 модуля 8 МГц, скорости обмена 1 Мбит/сек и скорости обмена с PN512 1130.000 кбит/сек, скорости обмена с ПК 115200 бит/сек, достигнуты следующие временна работы с картой:

* Активация карты с 4 байтным номером 6.9 мс.
* Чтение сектора с авторизацией (команда 33h) 16.6 мс.
* Запись сектора с авторизацией (команда 33h) 31.4 мс.

2.0.23pa – добавлена команда EFh. Реализовано переключение режима работы считывателя с SAM AV2 Модулем или без. Значение по умолчанию с SAM AV2 модулем.

2.0.24pa – в команду EFh добавлена возможность сохранения ключей из памяти SRAM во внешнюю SPI Flash память и чтение ключей в SRAM из SPI Flash.

Данные хранящие ключи в SPI Flash шифруются по алгоритмы AES128.

Одна запись с ключами содержит три версии ключа как в SAM AV2 модуле.

Одна такая запись имеет длину 128 байт, в SPI Flash выделено 16384 байта (4 сектора памяти) для хранения записей. Т.е. всего 128 записей. Так же выделено 16384 байта для копии первой записи. Т.е. выделено 32768 байт для хранения записей с ключами.

Прошивка при включении проверяет 127-ю (считать от 0) запись в каждой копии на целостность, используя CRC. Если обе записи повреждены, то обе копии стираются.

Если одна запись цела, то она используется для восстановления второй.

2.0.25pa – исправлена команда переключения слотов SAM AV2 модуля 27h.

2.0.26pa – доработана команда 27h, добавлен статус слотов.

2.0.27pa – доработка команд работы с блок значениями в версии прошивки для турникета: 0Fh, 10h, 11h, 14h.

Ключи для УКПД хранятся в памяти, а не в SAM AV2 модуле.

2.0.28pa – доработка команды 33h. Добавлена возможность чтения 4-х боков карты.

Работа двух УКПД: работа с картой по запросу, тот УКПД который успешно авторизовался, получает доступ к карте. Второй УКПД блокируется и получает на команды ответ: нет карты. Блокировка снимается по таймауту 500 мс или по запросу активации от УКПД, который захватил карту.

2.0.29pa – версия для турникета. Активацию карты выполняет сам считыватель.

Блокировки убраны. Введена буферизация данных для второго УКПД. При первом запросе читаются три блока сектора. При последующих запросах они будут возвращены из буфера.

2.0.30pa – исправление в реализации автономной активации. Сброс флагов наличия карты не только по таймеру или по ошибке, но и по командам активации.

2.0.31pa – отказ от автономной активации карт. Активация только по команде.

Были случаи возврата номера карты, когда карты уже нет. Сделана активация перед командами УКПД: 63h, 6Ch, 33h. Эта активация выполняется, если нет карты или она в состоянии Halt.

2.0.32pa – в команду 33h добавлены операции 1Eh, 2Eh, 3Eh.

2.0.33pa – в команде 33h удалёна защита от записи трейлера. Была рассчитана на запись 3-х блоков, но добавились другие операции записи.

2.0.34pa – тестовая версия. Введена блокировка второго УКПД в случае, если первый смог авторизоваться к блоку больше 16.

2.0.35pa – обнаружена ошибка настройки приёмопередатчика. После смены библиотеки настройки реально не записывались в регистры. Устройства для БПА и турникета были с настройками по умолчанию.

В этой версии настройки для старых устройств записываются. Для БПА и турникета нет.

В версии прошивки для турникета закомментированы не используемые команды. Для увеличения быстродействия.

2.0.36pa – в версии для турникета ответ для второго УКПД перенесён в конец исходного кода прошивки. Ответ отправляется после обработки и отправки ответа первому УКПД. Если от первого УКПД не было запроса, то ответ отправляется сразу.

2.0.37a – в версии для турникета изменена команда 33h. Сделана блокировка второго УКПД по запросу первого.

Настройки приёмопередатчика теперь нормально записываются в его регистры.

2.0.38a – тестовая версия. Вернул ответ второму УКПД сразу после обработки команды от него.

2.0.39a – тестовая версия. Добавлена команда E0h в версии для турникета. Блокировка второго УКПД.

2.0.40a и 2.0.41a – исправление ошибки чтения карт Ultralight вторым УКПД.

Так же изменена буферизация данных карт для второго УКПД. Для Mifare Classic буферизируется теперь 3 блока за один запрос, а не 4. Трейлер не буферизируется.

Для карт Ultralight буферизируются 8 страниц памяти за один запрос.

2.0.42a – блокировка второго УКПД не выключается по активации карт первым. Таймаут отключения блокировки увеличен с 300 мс до 5 секунд.

2.0.43a – перенос настроек прошивок в отдельный файл. Реализация обмена с приёмопередатчиком PN512 по SPI (SPI1 в микроконтроллере) интерфейсу на скорости 8 Мбит/сек.

Перенос, для версий считывателя с PN512 на SPI, интерфейса SAM AV2 модулей на USART2.

Для версии агентского считывателя (SM13048.61.00) продублирован светодиод диагностики на обратную сторону платы.

Добавилась настройка версий внешних считывателей SME14078 и SME14079.

При частоте SAM AV2 модуля 8 МГц, скорости обмена 1 Мбит/сек и скорости обмена с PN512 8 Мбит/сек, скорости обмена с ПК 115200 бит/сек, достигнуты следующие временна работы с картой:

* Чтение сектора с авторизацией (команда 33h) 23.4 мс.

Изменение интерфейсов подключения актуально для модификаций считывателя, появившихся с 2015 года. Это Агентский считыватель (SME13048.61.01) и Настольный считыватель (SME10478, SME14079).

2.0.45a – программно исправлено изменение подключение синего светодиода в версии настольного считывателя. Синий и жёлтый светодиоды переставлены.

Сделана версия прошивки настольного считывателя с остановкой карты перед активацией. Как в версии для турникета.

2.0.46a – для версии Агентского считывателя добавлена диагностика подключения антенны. См. команды F0h и F3h. Вывод состояния антенны сделан вместо состояния кнопки, которой в этом считывателе нет.

2.0.47a - отладочная версия для добавления функционала работы с картами содержащими ИС SRI512, SRI2K, SRI4K.

2.0.48a – добавление команды работы с картами SRI512, SRI2K, SRI4K, код команды 28h.

Увеличение частоты работы микроконтроллера с 64 МГц до 72 МГц, для плат настольных считывателей. При этом частота обмена с PN512 равна 9 Мбит/сек.

2.0.49a – оптимизация кода обработки команд второго УКПД. Перенос кода переключения протокола ISO14443B в ISO1443A и перенос кода выключения шифрования в приёмопередатчике.

Вместо флагов карт, для обработки команд второго УКПД, введён тип карт.

Исправлен ответ второму УКПД. Теперь обнуляются не 10 байт UID, а все 12.

Код каскадирования добавляется для всех карт с UID длиной в 7 байт, а не только для Ultralight (Ultralight C).

Уменьшение паузы между записью страниц Ultralight, командой второго УКПД.

Упрощение буферизации данных карты для второго УКПД. Выделена память объёмом 4К вместо 1К. Благодаря этому каждый блок лежит по жёстко заданному адресу.

Не нужен поиск блоков при чтении данных, удалении данных, добавлении данных.

2.0.50a – 2.0.51a – оптимизация кода обработки команд второго УКПД. Перенос кода активации карты и повторной активации перед чтением или записью.

Реализовано определение типа карт по ATQ и SAK. Реализовано определение номера ключа авторизации по типу карты. Есть определение совпадения ATQ и SAK для разных карт (с разными ключами авторизации). Сделана повторная активация и авторизация для таких карт, при ошибке. При повторной авторизации используется второй ключ.

2.0.52a – добавлена пауза 4 мс перед активацией карт для второго УКПД. Нужно для карт типа Стрелка. Без этой паузы такие карты не видны.

2.0.54a – добавление задержки в 2 мс перед записью карт Ultralight вторым УКПД.

Упрощение буферизации данных карт Ultralight для второго УКПД. Теперь читаю только 4 страницы одной командой и только их буферизирую.

2.0.55a – укорочен буфер для второго УКПД до 1К.

2.0.56a – введён сброс принятого пакета от второго УКПД, если ещё не завершена отправка ответа на предыдущий запрос. Это защита от зависания обмена из-за нарушения порядка запросов и ответов.

2.0.57a – изменён метод получения ключа для авторизации к карте для второго УКПД. Реализован поиск подходящего ключа.

2.0.58a – поиск ключа зациклен. 10 попыток от текущего значения в диапазоне от 0 до 10.

2.0.59a – 2.0.60a – версии прошивки для турникета с управлением полем в команде активации карты. Поле выключено, при поступлении команды активации оно включается.

Затем если нет карты, то выключается, если есть включается.

2.0.61a – увеличены таймауты на запись данных в Flash память и на стирание страниц памяти. Сделано это из-за замены памяти SST25VF016 на MX25L1606E.

2.0.62a – добавлено выключение напряжения питания со слотов, в которых нет SAM AV2 модулей. Добавлена возможность работать с SAM AV2 модулями в режиме Full protect communication. При этом происходит шифрование и подпись обмена считывателя и SAM AV2 модуля. Нужно для реализации перезаписи ключей из мастер-SAM модуля в SAM модуль.

В неописанной к данной версии команде 0x2B убран инкремент ключа. Это позволяет записывать ключи Host, начиная с 0-го. Т.е, позволяет перезаписывать этой командой Master Key.

В неописанной к данной версии команде 0x2A убран инкремент ключа. Это позволяет авторизоваться к Master Key, после его перезаписи.

Добавлена возможность включения бита Allow Dump Secret Key перед записью ключа в память SAM AV2 модуля. После записи или ошибки записи, бит сразу сбрасывается. Команда 66h, операция 00h.

2.0.63a – в команду 66h добавлена операция 01h. Операция копирования ключа Mifare из SAM в память контроллера считывателя. Для этого у ключа должен быть выставлен бит Allow Dump Secret Key. При успехе чтения ключа выставляется флаг наличия ключа памяти. При выполнении операции предыдущий ключ стирается. Записать ключ из памяти можно командой 31h.

В команде 31h добавлен функционал записи ключа из памяти контроллера при выставленном флаге наличия ключа в памяти. Значения ключей передаваемых от хоста при этом игнорируются.

После выполнения записи ключа (не зависимо от результата) флаг и ключ стираются.

2.0.64a – изменения в командах 2Ah, 37h, 38h, F9h.

В 2Ah – сделана авторизация по ключу из памяти для слота. Или по ключу, которым удалось авторизоваться для этого слота при включении или тот, что был записан в командах 37h, 38h, F9h. Если этот ключ получен при включении, то это или нули, или ключ из эмуляции EEPROM, или ключ, генерируемый при включении.

Изменение в командах 2Ah и F9h позволяют хранить ключ для Master SAM в памяти считывателя и не передавать его при эмиссии агентских SAM модулей.

В версии для турникета уменьшена задержка между командами записи данных на карту Ultralight с 8 до 4 мс.

2.0.65a – 2.0.67a – изменение в команде E0h, исправление ошибки перевода SAM модуля в режим AV2 (надо было проводить авторизацию в PLAIN mode), исправлен поиск установленных SAM модулей (выключаю IO линии всех слотов перед поиском).

Для турникетной версии в команде E0h добавлен байт кода статуса карты.

2.0.68a – 2.0.70a – изменения в версии для турникета. Поле выключается не сразу, если при активации нет карты, а только после 10 неуспешных активаций. После переключения на второй УКПД первая активация не производится.

2.0.73a – в команду 66h добавлена операция 02h. Операция копирования ключа PICC AES128 и PICC DES из SAM в память контроллера считывателя. Для этого у ключа должен быть выставлен бит Allow Dump Secret Key. При успехе чтения ключа выставляется флаг наличия ключа памяти. При выполнении операции предыдущий ключ стирается. Записать ключ из памяти можно командами 24h и 48h.

В командах 24h и 48h добавлен функционал записи ключа из памяти контроллера при выставленном флаге наличия ключа в памяти. Значения ключей передаваемых от хоста при этом игнорируются.

После выполнения записи ключа (не зависимо от результата) флаг и ключ стираются.

Исправлена ошибка версии для турникета. В командах были 09h и 33h разные переменные с номером карты. Это было введено в версиях 2.0.68a – 2.0.70a.

2.0.74a – для версии прошивки для турникета возвращена команда 0Dh.

В этой же версии данные в буфере второго УКПД обновляются в команде 33h.

Авторизация к секторам при чтении для второго УКПД производится один раз, а не при каждом обращении.

2.0.75a – в версии прошивки для турникета изменена команда 32h, исправлено различие переменных 09h и 32h.

2.0.76a – в версии прошивки для турникета отключена буферизация в команде 33h.

Отключена смена протокола на ISO14443A, т.к. нет необходимости. Отключено выключение Crypto1 шифрования в PN512. Выключение поля производится после 5 неудачных активаций.

23.07.2015 – в описании протокола обмена введено оглавление для списка команд.

2.0.77a – убрана проверка авторизации к сектору для второго УКПД. Теперь снова нужна авторизация перед каждым чтением блока.

2.0.78a – исправлена ошибка, связанная с буферизацией данных для второго УКПД.

Изначально буфер сбрасывался при активации карты вторым УКПД. Затем был сделан пропуск активации вторым УКПД при его разблокировке, т.к. карта была уже активирована первым УКПД. Но данные в буфере не сбрасывались.

В этой версии буфер сбрасывается при активации карты первым УКПД. В этой же версии данные для второго УКПД буферизируются в команде 33h.

2.0.79a – в команде EFh добавлена операция 03h. Операция запроса режима работы считывателя: с SAM AV2 модулем или без.

2.0.80a – добавлено отключение циклического вывода кода ошибки при обращении к командам F0h и F2h.

2.0.81a – исправлена ошибка в команде 33h.

2.0.82a – в версии прошивки для турникета разрешены команды в диапазоне C0h – CEh.

2.0.84a – в версии для диспенсера отключён запрет прерывания на приём данных от хоста на время передачи ответа.

2.0.85a – реализована аппаратная генерация случайных чисел.

2.0.88a – выключение поля сразу после сброса PN512. Изменёно обнаружение и переключение в T1 для SAM AV2.

2.0.89a - убрал инициализацию модуля USART, который не использован в ридере с PN512 на интерфейсе SPI.

2.0.90a – в версии для турникета, в команды второго УКПД, добавлены команды чтения счётчика и списание счётчика. Для карт Ultralight EV1.

2.0.91a - 2.0.94a – реализация для работы с приёмопередатчиком CLRC663.

2.0.94a - изменения в 33h команде. Сделана передача в ActivateCard UID карты от предыдущей активации.

2.0.95a - увеличен таймаут в Rc663\_Wait. Иначе не работает с Ultralight EV1, таймаут истекает на чтении нескольких страниц. Добавление операций 0x03, 0x04 в команде 0x66. Удаление инкремента номера ключа в некоторых командах, которых пока нет в описании протокола. Это, к примеру, запись KUC - счётчики.

2.0.96a - доработки под метро и добавлен код для считывателя под 3K5/3K7, SME10145.622.001.

2.0.97a – убран Warm reset при активации SAM в слотах.

2.0.98a – исправление возврата расширенного кода ошибки для команд 0x07, 0x09.

2.0.99a – вынос кода работы с I2C в HAL. Изменения в коде активации SAM в слотах.

2.1.00a и 2.1.01a – изменения кода для версий с CLRC663 в файлах HAL и Rc663\_Wait. И изменение кода в активации карт для версий CLRC663. Сделано для исправления ошибки многостраничного чтения карт Ultralight EV1.

2.1.02a – уменьшил паузу в Rc663\_Wait.

2.1.03a – добавил операции в команду 0x66: деактивация SAM в слоте и активация SAM в слоте. Изменения в команде 0x2A.

2.1.04a – добавлена поддержка считывателя SM16089.

2.1.05a – изменение команд работы с 3K5/3K7.

2.1.06a – изменения для поддержки считывателя SM16089.

2.1.07a – изменения команд 0xCD и 0xCE. Добавлена возможность подписи и проверки подписи для данных размером больше 240 байт.

2.1.08a - изменения команд 0xC9 и 0xCA. Добавлена возможность шифрования и дешифрования для данных размером больше 240 байт.

2.1.08a – для считывателя SM16089 добавлено управление светодиодами на плате антенны, подключенными через I2C IO экспандер. И для этого же считывателя изменены команды 0xF0 и 0xF2.

2.1.09a – доработка кода управления светодиодами в считывателе SM16089. Управление светодиодами осуществлено изменением конфигурации выводов IO экспандера. До этой версии было изменение состояния выходов IO экспандера (0 или 3.3 В) и было тусклое красное свечение светодиодов. Вызвано было разностью напряжений питания светодиодов (5В) и IO экспандера (3.3 В). Теперь для выключения светодиода вывод IO экспандера переводиться в режим входа.

2.1.10a – добавление команды 0xCF, генерация случайной последовательности чисел.

2.1.11a – доработка для подключения любых SAM модулей. Изменения в HAL и изменение в приёме ATR.

Уменьшение выходного напряжения CLRC663 для считывателя турникета.

2.1.12a - 2.1.14a – доработка для подключения любых SAM модулей. Переход на библиотеку ISO7816.

2.1.15a - 2.1.16a – доработка для подключения любых SAM модулей. Производиться определение типа SAM модуля в слоте (SAM AV2 или нет). Производиться вычисление скорости обмена с SAM в зависимости от его типа и выдаваемого в ATR параметра обмена TA1. Теперь каждый SAM слот имеет свою скорость обмена, которая переключается при переключении слота.

Доработана библиотека ISO7816.

Изменение в команде 0x27. Добавлен код ответа 0x01 – для SAM модулей не SAM AV2.

2.1.17a – доработка команды 0x65. Решена проблема перезагрузки считывателя при подаче команды 0x65 при отсутствии SAM модулей.

2.118a - убрал лишний запрос параметров в RATS команде.

2.119a - доработка для добавления версии ручного считывателя.

2.120a - изменения в 3K5. Исправил ошибку реализации обмена APDU в команде 65h.

2.121a – доработка для реализации работы по двум интерфейсам. Добавлена возможность работы с RS485, с переключением приём-передача. Добавлена обработка кнопок. Это только для внешнего считывателя с интерфейсами RS232, RS485 и с двумя входами для кнопок. Для этой версии изменены выходные параметры команд F0h, F3h. Теперь выдаётся два байта состояния кнопок, вместо одного.

2.122a – для считывателя турникета разрешена команда 0Eh.

2.123a - 2.127a – доработка для работы с разными SAM модулями.

2.128a – добавление возможности шифрования обмена по ГОСТ28147-89 Магма.

Изменение в команде проверки связи 02h, добавлен параметр включения шифрования.

2.129a – в команду 33h добавлен параметр чтения одного блока.

2.130a – исправление ошибки задания настроек по умолчанию, при первом запуске устройства.

2.131a - увеличен таймаут для обмена с SAM модулями. Таймаут до этого был уменьшен при переходе на библиотеку ISO7816.

2.131a – 2.135a – реализация команд 67h – 69h, для работы с диспенсером карт CRT571. Доработки под версию платы считывателя под CRT571.

2.136a – 2.139a – исправление ошибки работы с несколькими SAM модулями.

При выборе первого активного SAM не было отключение линий IO других модулей.

Доработки для считывателя 3K5/3K7 в версии с двумя антеннами. Добавлена команда FEh – переключение антенны.

2.140a – 2.141a – доработки под диспенсер SCD-2500.

2.142a – 2.144a – реализация универсальной прошивки считывателя для работы с диспенсером SCD-2500, моторизованными считывателями 3K5 и 3K7, пробросом данных в интерфейс ведомого устройства и обратно.

Изменения в команде 67h: добавлен опциональный параметр режима обмена с ведомым устройством, добавлен возврат параметров обмена.

2.145a – добавление команды 6Ah для управления линией DTR и команда 6Bh для запроса состояния линии DSR.

2.147a – добавление параметра в команду 27h. Добавлена возможность переключения на работу без SAM модуля.

2.148a – добавление параметров команд 2Ah и 66h. Появилась возможность авторизации к SAM с переключение указателей на работу с ним.

2.149a – реализация возможности работы без SAM модуля. Хранения ключей в Flash памяти.

2.151a – уменьшение выходной мощности приёмопередатчика для версии считывателя турникета.

2.152a – 2.162a – замена библиотеки ISO7816.

2.163a – 2.164a – добавление команд 70h, 71h, 72h. Изменение команды 35h, добавлены выводимые параметры ключей.

2.165a – 2.171a – доработка команд 70h, 71h, 72h. Добавление команды 73h.

Добавление возможности диверсификации ключей при авторизации командами 0Ah, 12h, 32h, 33h.

2.172a - 2.175a – доработки по просьбе ЦППК, ПП, ISBC.

2.176a – доработка команд 71h, 72h, 73h. Добавление команды 74h.

2.177a – добавлена команда 75h.

2.178a – добавлена операция 09h в команде 66h. Чтение ключа AES128/2K3DES карты из SAM AV2 модуля. Ключ должен быть записан с выставленным битом Allow Dumping Secret Key.

2.180a – исправлена ошибка. Не было обработчика прерывания для считывателя SME10145.626.001 в ветке ПП.

2.183a – в команду 44h добавлена возможность диверсификации ключа.

2.184a – доработка для чтения блоков карты Тройка с микросхемой Mifare Plus EV1.

2.185a – ввёл версию для SME10145.626 для БАМ и антенны SME10145.631.001.

2.186a – добавил паузу в опрос NFC чипа и увеличил паузу WAITTX, из-за новых карт Тройка.

2.187a – 2.188a – сменил библиотеку ISO7816, начал делать поддержку ESMART.

2.189а – добавил определение ESMART при активации в слоте. Добавил код 0x02 для ESMART в команде 27h.

2.190a – 2.192a – сделал запрос версии и UID модуля ESMART при запуске считывателя, сделал авторизацию к ESMART тестовым ключём.

2.193a – 2.194a – в команды 27h, 2Ah, 3Ah, 3Bh, 44h, 46h добавлена поддержка ESMART. Для них используется ESMART, если он установлен в текущем слоте.

2.195a – версия для проверки ESMART. Тактирование подаётся равным 8 МГЦ вместо 4 МГц. Не используется. Возврат на 4 МГц.

2.196a – добавлена команда 19h DESELECT для карт Mifare Plus на 3-м уровне.

2.198a – добавлена команда 76h, для обмена с картой на низком уровне ISO14443-3.

2.200a – добавлено определение SAM AV3 по ATR и по команде запроса версии.

2.201a – в команде 33h добавлены операции ANh и BNh.

2.202a – 2.203a – изменение активации карты. Убрал запросы к SAM модулю.

2.204a – сброс шифрования обмена, после авторизации к карте через SAM, не зависимо от успеха авторизации. Сделано для SAM AV3. SAM AV3 отличается в этом от SAM AV2.

2.205a – изменения в команде 33h для операций ANh и BNh. Теперь читается и пишется N + 1 блоков.

2.206a – изменение в команду 27h. При задании номера слота равного 27h не будет возвращена ошибка ошибочного параметра команды.

2.207a - исправил команду 33h в плане операция 0xAN, 0xBN.

2.208a – 2.210a — доработки для SCD2500.

2.112a - изменения под ППР для Тройки в которой была ошибка чтения 32 блока.

2.118a - добавил паузу в команды 0x0D, 0x0E. Настроил регистры OverShootT1 и OverShootT2 для считывателя SME10145.626.

2.220a – 2.221a – добавил поддержку микроконтроллера GigaDevice.

2.222a – 2.223a - добавил поддержку микроконтроллеров Artery и Geehy. Требуется доработка схемы.

2.224a – добавил поддержку контроллера SGS8103ARPT6. Не совпадает полностью ни с STM32, ни с AT32.

2.225a – проверка NFC на основе JSM663.

2.226a – 2.229a – смена среды разработки, смена компилятора, переход с SVN на Git.

2.230a – 2240a – смена библиотеки для поддержки SAM AV3.

2.241a – ввёл счётчик APDU команд для SAM и ввёл команду 0x6C для возврата значения счётчика.

2.242a – 2.244a – автоматический перезапуск SAM AV3 без авторизации. Добавлена команда переключения сектора карты MFP EV1 в MixMode SL1/SL3.

2.245a – 2.246a – исправления для реализации роботы с картой MFP в режиме SL3 командами APDU.

2.247a – исправил ошибку запроса информации производителя SAM AV2/AV3, команда 0x30. Если был установлен не SAM AV2 или SAM AV3, то возвращался мусор или значение от предыдущей команды. Проявлялась ошибка если в одном слоте SAM AV2/AV3, в другом ESMART. Если первый запрос к AV2/AV3, а второй к ESMART, то для ESMART возвращалось значение AV2/AV3. В этой версии для SAM не AV2/AV3 будет ошибка.