

**Программное обеспечение устройства работы с
бесконтактными картами по спецификации EMV,
Управляющая программа считывателя бесконтактных
банковских карт ВСАМ-01**

Руководство по эксплуатации

Листов 57

Москва, 2023

Оглавление

Список сокращений.....	4
Аннотация.....	5
1. Общие сведения.....	6
1.1. Общие требования к аппаратной платформе.....	7
1.2. Требования к программным и техническим средствам	7
1.3. Начало работы.....	7
2. Эксплуатация функционала УПО	8
2.1. Инициализация узлов и интерфейсов устройства	9
2.2. Связь с терминальным ПО, поддержка протокола PB3P	9
2.2.1. Запрос версии прошивки (0x02).....	11
2.2.2. Запрос версии прошивки с расширенным перечнем параметров (0xA1) .	12
2.2.3. Команда запроса расширенного серийного номера считывателя (0x10) .	13
2.2.4. Команда установки скорости порта на устройстве (0x0F)	14
2.2.5. Команда занесения в ПЗУ устройства списка поддерживаемых AID (0x92)	15
2.2.6. Команда чтения из ПЗУ устройства списка AID (0x9B)	17
2.2.7. Команда занесения в ПЗУ устройства настроек для поддерживаемых платёжных систем (0x91)	19
2.2.8. Команда чтения из ПЗУ устройства списка настроек для поддерживаемых платёжных систем (0x9A)	21
2.2.9. Команда занесения в ПЗУ устройства открытых ключей платёжных систем (0x93).....	23
2.2.10. Команда чтения из ПЗУ устройства открытых ключей платёжных систем (0x9C).....	26
2.2.11. Команда занесения в ПЗУ устройства списка отозванных сертификатов платёжных систем (0xA3)	29
2.2.12. Команда чтения из ПЗУ устройства списка отозванных сертификатов платёжных систем (0xA4)	31
2.2.13. Команда передачи устройству системных настроек (0x98).....	33
2.2.14. Команда чтения с устройства текущих настроек (0x99).....	35

2.2.15. Команда получения с устройства информации о платёжных ядрах (0x9D)	37
2.2.16. Команда начала транзакции (0x94)	39
2.2.17. Команда отмены транзакции (0x95)	43
2.2.18. Команда получения хостом системных сообщений Outcome (0xA0)	45
2.2.19. Команда сообщения MSG UI (0xA2)	48
2.2.20. Команды обмена данными по протоколу Mastercard DEK/DET	49
2.2.21. Команда протокола ПС «МИР» Data Exchange (0xA5)	50
2.2.22. Команды логов обмена с картой (0x9E, 0x9F)	52
2.2.23. Команда отправки устройству ответа эмитента (0xB3)	53
2.2.24. Команда включения поля антенны (0x70)	54
2.2.25. Команда отключения поля антенны (0x71)	56
2.3. Передача команд управления во встроенные программные модули	57

Список сокращений

Сокращение	Расшифровка
AID	Application IDentifier - номер приложения
APDU	Application Protocol Data Unit – команда протокола обмена между устройством и картой
DEK	Data Exchange Kernel – данные ядра в рамках протокола Mastercard DEK/DET
DET	Data Exchange Terminal – данные терминала в рамках протокола Mastercard DEK/DET
EMV	Europay + MasterCard + VISA – международный стандарт для операций по банковским картам с чипами
NFC	Near Field Communications - технология ближней бесконтактной связи
PB3P	Название протокола обмена данными
RTC	Real Time Clock - часы реального времени
SPI	Serial Peripheral Interface - последовательный периферийный интерфейс
UART	Universal Asynchronous Receiver-Transmitter - универсальный приёмопередатчик
ПЗУ	Постоянное запоминающее устройство
ПО	Программное обеспечение
УПО	Управляющее программное обеспечение
ЭВМ	Электронно-вычислительная машина

Аннотация

Данный документ содержит описание функциональных характеристик программного обеспечения устройства работы с бесконтактными картами по спецификации EMV.

Документ предназначен для ознакомления с функциональными возможностями программного обеспечения.

1. Общие сведения

Управляющая программа устройства для работы с бесконтактными картами в стандарте EMV (далее УПО - управляющее программное обеспечение) предназначено для:

- инициализации и функционирования аппаратных узлов, внутренних и внешних интерфейсов устройства ВСАМ-01 (так называемый аппаратный абстрактный уровень, при переносе УПО на другую аппаратную платформу, этот уровень должен быть легко сменяем, неизменным должна оставаться функция: инициализация узлов и интерфейсов устройства);
- организации связи с терминальным программным обеспечением на основе разработанного протокола обмена данными РВЗР;
- передачи команд управления и данных от терминального ПО встроенным программным модулям таким, как: модуль Level 1, модуль ядра Mastercard, модуль ядра МИР и модуль ядра Visa.

Для работы с УПО необходимо предварительно скомпилировать, собрать его под определённую аппаратную платформу и установить. Процесс сборки и установки УПО лежит за рамками этого документа. Подразумевается, что аппаратная платформа, на которой производится установка, обладает необходимым функционалом реализации физического уровня протокола ISO/IEC 14443(A/B), другими словами, должна иметь в своём составе микросхему NFC. Для упрощения здесь и далее предлагается называть такую аппаратную платформу ридером или устройством.

Для управления ридером и обмена данными с ридером должен быть разработан протокол, реализующий набор необходимых команд для эксплуатации функций программных модулей, установленных на ридер, в частности одним из таких модулей должно быть УПО.

Для управления ридером по разработанному протоколу необходима реализация так называемого терминального ПО, которое может быть установлено на другое устройство и осуществлять взаимодействие с ридером в автоматическом режиме, либо такое ПО может быть выполнено в виде программы как с графическим интерфейсом, так и без него, установленной на ПЭВМ и работающей по командам или манипуляциям человека-оператора.

В данном документе предлагается описание эксплуатации УПО по второму варианту, где в качестве терминального ПО выступает приложение с графическим интерфейсом, которое устанавливается на ПЭВМ. Приложение было разработано для сертификации в одной из лабораторий компании EMVCo, по требованиям к этому ПО интерфейс приложения - английский. В качестве примера аппаратной платформы в данном документе будет рассмотрен ридер "ВСАМ-01",

разработанный российской компанией ООО "Социальные системы". Протокол обмена и управления "BCAM-01" - РВЗР, разработанный также ООО "Социальные системы".

1.1. Общие требования к аппаратной платформе

УПО разработано на алгоритмическом языке "C" в стандарте POSIX, поэтому переносимо между системами и платформами, вписывающимися в этот стандарт.

Для поддержки функционала УПО аппаратная платформа должна обладать:

- микропроцессором с тактовой частотой не ниже 120 МГц;
- модулем NFC и интерфейсом обмена с ним со скоростью не ниже 600 кбод;
- постоянным запоминающим устройством, имеющим возможность хранить не менее 64 кб данных, а также обладающим функционалом чтения и записи данных с/на него;
- оперативной памятью не менее 256 кб;
- интерфейсом связи с терминальным ПО (например, UART или USB) со скоростью не ниже 115 кбод.

1.2. Требования к программным и техническим средствам

Для работы с УПО, где терминальной программой является приложение, установленное на ПЭВМ, и с выбранной аппаратной платформой "BCAM-01" требуется:

- инструментальная ПЭВМ под управлением ОС Windows, версии не ниже 8, с установленной на ПЭВМ программой "L2 Test Environment" (далее ТЕИ), версии не ниже 1.2.0;
- устройство "BCAM-01" с загруженной на него рабочей программой, версии не ниже 2.5.1, с модулем УПО ("xch_st_machine" (Exchange State Machine) - автомат состояний обмена данными - рабочее название, которое возвращается при опросе версий ПО на устройстве), версии не ниже 1.00; устройство должно быть подключено к ПЭВМ по USB-кабелю;
- банковская карта с бесконтактным интерфейсом.

1.3. Начало работы

Для начала работы требуется запустить приложение ТЕИ, открыть вкладку 'Settings', выбрать порт, который назначен для ридера, нажать на форме кнопку 'Open'. Приложение ТЕИ откроет порт, автоматически опросит версии ПО ридера и выведет их в поле 'Versions' (рисунок 1).

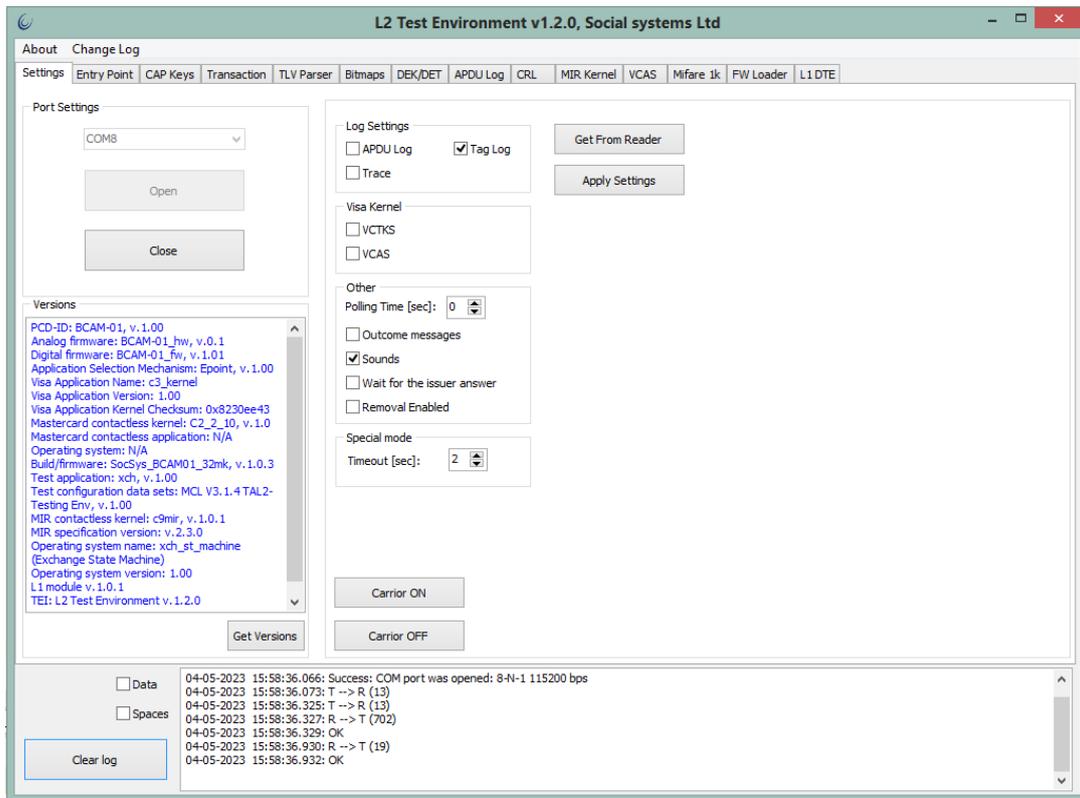


Рисунок 1 - Начало работы, открытие порта.

На рисунке 1 в нижней части формы располагается консоль, куда выводится лог обмена устройства с терминальным ПО. Для более подробной информации следует отметить поле "Data".

2. Эксплуатация функционала УПО

УПО разработано для:

- а) инициализации и функционирования аппаратных узлов, внутренних и внешних интерфейсов устройства BCAM-01. В перечень узлов и интерфейсов входят:
- а.1) узел микропроцессора устройства;
 - а.2) узел модуля NFC;
 - а.3) узел постоянного запоминающего устройства - ПЗУ;
 - а.4) внешние и внутренние интерфейсы UART;
 - а.5) интерфейсы SPI;
 - а.6) узел RTC (Real Time Clock) - для организации временных интервалов и поддержки системного времени на устройстве;
 - а.7) узел световой индикации;
 - а.8) узел звуковой индикации;

б) организации связи с терминальным программным обеспечением через внешний интерфейс UART на основе разработанного протокола обмена данными РВЗР;

в) передачи команд управления и данных от терминального ПО встроенным программным модулям таким, как: модуль Level 1, модуль ядра Mastercard, модуль ядра МИР и модуль ядра Visa.

2.1. Инициализация узлов и интерфейсов устройства

При включении устройства в функционале УПО предусмотрено:

- инициализация программируемых пины микропроцессора в соответствии с принципиальной электрической схемой устройства, назначая им функции, предусмотренные архитектурой микропроцессора;

- инициализация задействованных в логической схеме устройства интерфейсов: для SPI устанавливается подходящая частота тактирования в соответствии с возможностями подключаемых узлов ПЗУ и NFC, для UART устанавливаются выбранные скорости передачи данных, настройки чётности, количество стоповых битов, каждому UART назначается буфер приёма и прерывания по событиям приёма данных и обработки ошибок приёма/передачи;

- установка необходимых таймеров, назначение и разрешение их прерываний.

Проверка корректности инициализации узлов и интерфейсов проверяется на заводе-изготовителе с помощью ТПО (тестового программного обеспечения).

2.2. Связь с терминальным ПО, поддержка протокола РВЗР

Обмен данными с устройством осуществляется по протоколу РВЗР. Формат посылки следующий: {<блок заголовка РВЗР>, <блок данных>}. Данный формат должен соблюдаться в обоих направлениях: от ведущего устройства (далее хост) к ведомому (далее ридер или устройство) и обратно.

При обмене между хостом и устройством соблюдается следующий порядок: инициатором обмена является хост (кроме исключений, описанных в данном документе), после каждой посылки хост должен дожидаться ответной посылки от устройства или события timeout при отсутствии данных в ресивере СОМ-порта. Время ожидания ответной посылки – 3 [сек].

Заголовок протокола представляет собой последовательность байт. Формат заголовка приведен в Таблице 1.

Таблица 1. Формат заголовка протокола RV3P

Смещение	Наименование поля	Тип	Размерность, [байт]	Комментарий
0x00	ver	Беззнаковое целое	1	Версия протокола (текущая версия - 2)
0x01	datablock_len	Беззнаковое целое	2	Размерность блока данных, идущего за заголовком. Размерность задаётся в байтах. В текущей версии устройства размер блока данных ограничен размером принимаемого буфера: 0x4000 (16384) байт.
0x03	datablock_crc	Беззнаковое целое	2	Контрольная сумма блока данных. Алгоритм: CRC-16-CCITT (образующий полином 0x1021, начальное значение 0xFFFF). См. п.1.1.
0x05	cmd	Беззнаковое целое	1	Команда устройству. В данном документе приведён перечень доступных команд
0x06	direction	Беззнаковое целое	1	Направление передачи данных. В данной версии устройства доступны следующие значения: 0 – направление передачи данных от устройства к хосту, 3 – направление передачи данных от хоста к устройству.
0x07	timeout	Беззнаковое целое	2	В данной версии протокола не используется, должно инициализироваться 0 (0x0000).
0x09	err	Беззнаковое целое	2	В данном поле содержится значение ошибки протокола. Актуально для ответной посылки от устройства к хосту. При передаче пакета в направлении от хоста к устройству значение поля должно быть 0 (0x00)
0x0b	sync	Беззнаковое целое	1	Байт синхронизации пакетов. Использование опционально: если значение 0x00, то не

				учитывается в процессе работы, если значение от 0x01 до 0xFF, то ридер должен ответить на команду с таким же значением в заголовке.
0x0c	header_crc	Беззнаковое целое	1	Контрольная сумма заголовка РВЗР. Расчёт КС осуществляется по полям [ver, ..., RFU] (по первым 12 байтам заголовка). Алгоритм: CRC-8-DALLAS, полином 0x8C, начальное значение 0x00. См. п.1.2.
			Σ 13	

Значения полей передаются в кортеже байт в формате little-endian.

2.2.1. Запрос версии прошивки (0x02)

По данной команде устройство возвращает упрощённую версию текущей прошивки.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x02
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0003
datablock_crc	0xXXXX
cmd	0x02
direction	0x00

timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки имеет следующий формат:

{<MAJOR_VERSION (1)>, <MINOR_VERSION (1)>, <BETA_VERSION (1)>}

Пример:

```
(команда, hex)    >>    02 00 00 00 00 02 03 00 00 00 00 00 F3
(ответ, hex)     <<    02 03 00 09 5D 02 00 00 00 00 00 00 54 02 05 00
```

```
MAJOR_VERSION = 2 (0x02);
MINOR_VERSION = 05 (0x05);
BETA_VERSION = 00 (0x00).
```

2.2.2. Запрос версии прошивки с расширенным перечнем параметров (0xA1)

По данной команде устройство возвращает расширенную версию текущей прошивки.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0XXXXX
cmd	0xA1
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0XXXXX
datablock_crc	0XXXXX
cmd	0xA1

direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки имеет следующий формат:

{<MAJOR_VERSION (1)>, <MINOR_VERSION (1)>, <BETA_VERSION (1)>, <DATE (N)>}

Пример:

```
(команда, hex)    >>    02 00 00 00 00 00 A1 03 00 00 00 00 00 00 CC
(ответ, hex)     <<    02 0C 00 5B 19 A1 00 00 00 00 00 00 00 E1 00 19 0E 31 31
                  2E 31 32 2E 31 38 00
```

```
MAJOR_VERSION = 0 (0x00);
MINOR_VERSION = 25 (0x19);
BETA_VERSION = 14 (0x0E);
DATE = "11.12.18" - ASCII-последовательность, длина N байт с учётом нулевого символа.
```

2.2.3. Команда запроса расширенного серийного номера считывателя (0x10)

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x10
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x000D
datablock_crc	0xXXXX
cmd	0x10

direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки имеет следующий формат:

{<ERROR (2)>, <SERIAL (11)>}

Пример:

```
(команда, hex)  >>  02 00 00 00 00 10 03 00 00 00 00 00 72
(ответ, hex)   <<  02 0D 00 AF 91 10 00 00 00 00 00 00 30 00 00 53 36 36
                30 35 34 10 0E 21 17 17
```

```
ERROR = 0 (0x0000);
SERIAL = 0x533636303534100E211717.
```

2.2.4. Команда установки скорости порта на устройстве (0x0F)

Данная команда предназначена для изменения скорости передачи данных на устройстве. После отправки команды с хоста, устройство присылает подтверждение получения команды, используя старую скорость передачи. После установки нового значения скорости необходимо установить такую же скорость на хосте, к которому подключён ридер.

Блок заголовка команды:

ver	0x02
datablock_len	0x0004
datablock_crc	0xXXXX
cmd	0x0F
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды:

{<BAUDRATE (4)>}, где

BAUDRATE - значение желаемой скорости, скорость может быть как из стандартного ряда скоростей COM-порта, так и любая.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0002
datablock_crc	0XXXXX
cmd	0x1C
direction	0x00
timeout	0x0000
err	0XX
sync	0x00
header_crc	0XX

Блок данных ответной посылки имеет следующий формат:

{<ERROR (2)>}, где

ERROR - результат выполнения команды на устройстве

Пример:

```
>> 02 04 00 B5 A0 0F 03 00 00 00 00 DE BA 20 A1 07 00
<< 02 02 00 0F 1D 0F 00 00 00 00 00 DE 4B 00 00
```

```
BAUDRATE = 0x0007A120 = 500000
ERROR = 0 (0x0000).
```

2.2.5. Команда занесения в ПЗУ устройства списка поддерживаемых AID (0x92)

В рамках работы с EMV-картами устройству необходим перечень поддерживаемых AID (Application IDentifier). Данная команда предназначена для передачи с хоста на устройство этого списка.

Максимальный размер списка с учётом его формата - 3068 байт. Ограничение связано с размером постоянной памяти устройства.

Блок заголовка команды:

ver	0x02
datablock_len	0XXXXX

datablock_crc	0xXXXX
cmd	0x92
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды:

{<AID_LIST (0xXXXX)>}, где

AID_LIST - список AID, поддерживаемых ридером.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0002
datablock_crc	0xXXXX
cmd	0x92
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки имеет следующий формат:

{<ERROR (2)>}, где

ERROR - результат выполнения команды

Пример:

```
>> 02 3F 00 FC 18 92 03 30 E0 E7 7A 72 0A AA A7 B0 C0 45 58 97 DF 3F DA C9 A8 EB
96 8D BC 19 DF 45 FB F9 FF 33 0B 4F 05 A0 00 00 00 04 DF 7E 01 01 FF 33 0B 4F 05 A0
00 00 06 58 DF 7E 01 01 FF 33 0B 4F 05 A0 00 00 00 03 DF 7E 01 01
<< 02 02 00 0F 1D 92 00 00 00 00 00 72 13 00 00
```

AID01 = A0 00 00 00 04 - все приложения ПС Mastercard;
AID02 = A0 00 00 06 58 - все приложения ПС МИР;

AID03 = A0 00 00 00 03 - все приложения IC Visa;

ERROR = 0x0000.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Send', в секции 'Processing Configuration', на вкладке 'Entry Point'.

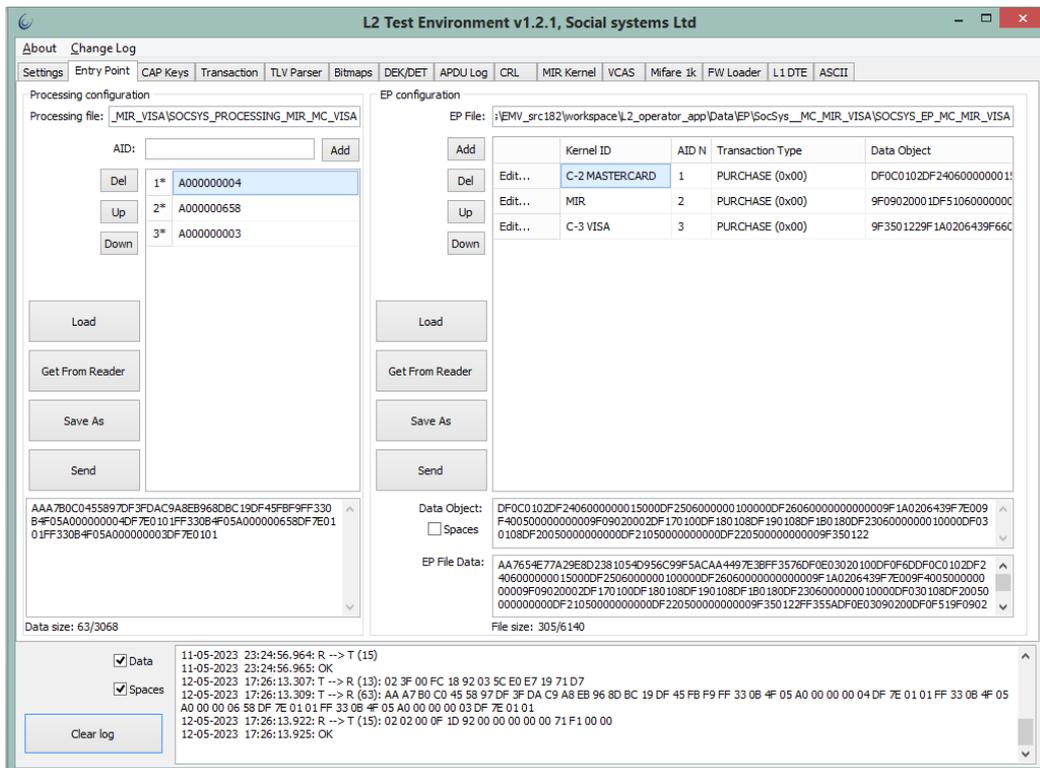


Рисунок 2 - Результат выполнения команды передачи устройству списка AID

2.2.6. Команда чтения из ПЗУ устройства списка AID (0x9B)

Команда используется для проверки или контроля данных списка AID. При выполнении команды на хост передаётся содержимое списка AID, записанного в ПЗУ ридера.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x9B
direction	0x03
timeout	0x0000
err	0x00

sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0XXXXX
datablock_crc	0XXXXX
cmd	0x9B
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>, <AID_LIST (0XXXXX)>}, где

ERROR - результат выполнения команды на устройстве,
AID_LIST - список AID, поддерживаемых ридером.

Пример:

```
>> 02 00 00 00 00 00 9B 03 00 00 00 00 00 53
<< 02 41 00 22 FA 9B 00 00 00 00 00 00 00 D7 00 00 AA A7 B0 C0 45 58 97 DF 3F DA C9
A8 EB 96 8D BC 19 DF 45 FB F9 FF 33 0B 4F 05 A0 00 00 00 04 DF 7E 01 01 FF 33 0B 4F
05 A0 00 00 06 58 DF 7E 01 01 FF 33 0B 4F 05 A0 00 00 00 03 DF 7E 01 01
```

```
ERROR = 0x0000,
AID01 = A0 00 00 00 04 - все приложения IC Mastercard,
AID02 = A0 00 00 06 58 - все приложения IC МИР,
AID03 = A0 00 00 00 03 - все приложения IC Visa.
```

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Get From Reader', в секции 'Processing Configuration', на вкладке 'Entry Point'.

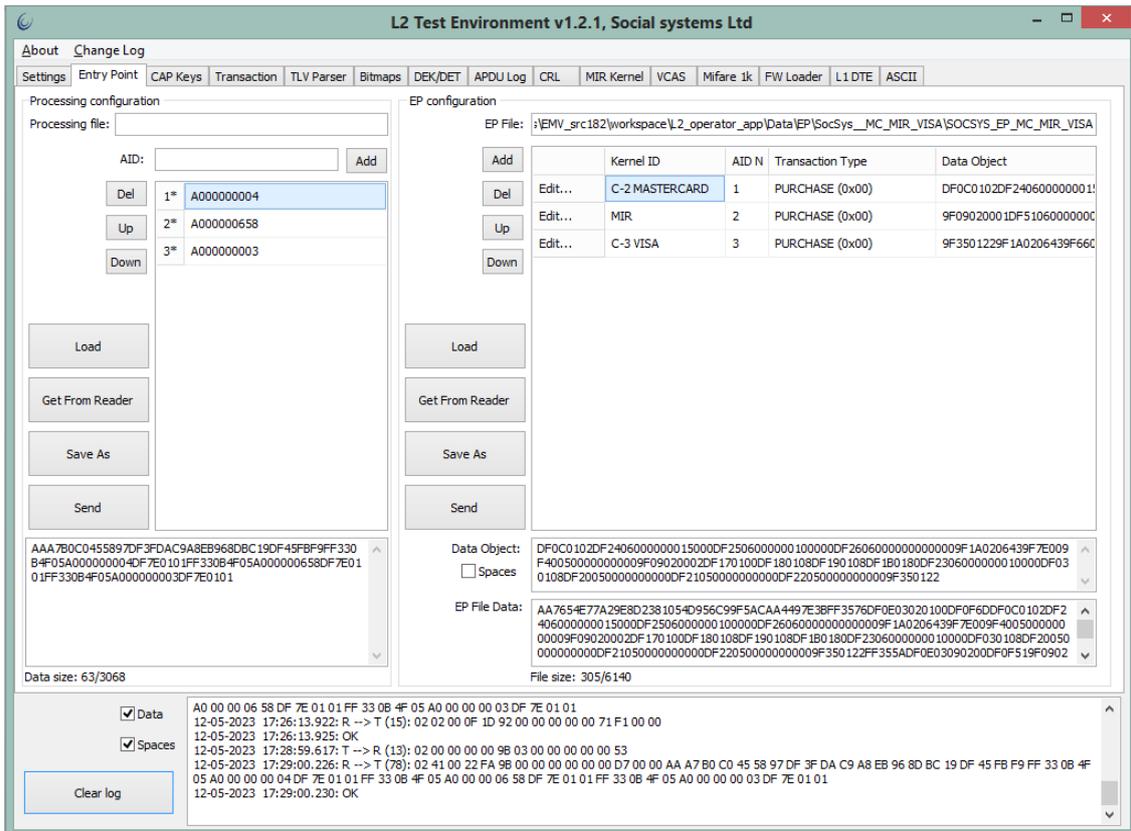


Рисунок 3 - Результат выполнения команды чтения списка AID из ПЗУ устройства

2.2.7. Команда занесения в ПЗУ устройства настроек для поддерживаемых платёжных систем (0x91)

В данной реализации программного обеспечения ридера для каждого элемента списка AID на устройстве содержатся настройки, соответствующие этому элементу списка AID.

Максимальный размер списка настроек с учётом их формата - 6140 байт.

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x91
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды:

{<EP_LIST (0xXXXX)>}, где

EP_LIST- лист настроек для элементов AID_LIST.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0002
datablock_crc	0xXXXX
cmd	0x91
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>}, где

ERROR - результат выполнения команды на устройстве.

Пример:

```
>> 02 31 01 6C 79 91 03 30 E0 E7 7A 72 04 AA F2 35 F4 88 5F 9E A1 F1 E3 1E 96 53
CE 3D 3C 85 E4 B5 DE 70 FF 35 76 DF 0E 03 02 01 00 DF 0F 6D DF 0C 01 02 DF 24 06 00
00 00 01 50 00 DF 25 06 00 00 00 10 00 00 DF 26 06 00 00 00 00 00 00 9F 1A 02 06 43
9F 7E 00 9F 40 05 00 00 00 00 9F 09 02 00 02 DF 17 01 00 DF 18 01 08 DF 19 01 08
DF 1B 01 80 DF 23 06 00 00 00 01 00 00 DF 03 01 08 DF 20 05 00 00 00 00 DF 21 05
00 00 00 00 DF 22 05 00 00 00 00 00 9F 35 01 22 FF 35 5A DF 0E 03 09 02 00 DF 0F
51 DF 55 02 20 00 9F 35 01 22 DF 20 05 00 00 00 00 DF 21 05 00 00 00 00 DF 22
05 00 00 00 DF 56 01 00 9F 09 02 00 01 DF 51 06 00 00 00 01 50 00 DF 52 06 00
00 00 01 50 00 DF 53 06 00 00 00 01 50 00 DF 54 06 00 00 00 01 50 00 DF 23 00 FF 35
43 DF 0E 03 03 03 00 DF 0F 3A 9F 35 01 22 9F 1A 02 06 43 9F 66 04 20 00 40 00 DF 30
01 F8 DF 32 01 02 9F 1B 04 00 00 00 DF 23 06 00 00 00 03 00 00 DF 24 06 00 00 00
03 00 00 DF 26 06 00 00 00 03 00 00
<<02 02 00 0F 1D 91 00 00 00 00 00 72 54 00 00
```

ERROR = 0x0000.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Send', в секции 'EP Configuration', на вкладке 'Entry Point'.

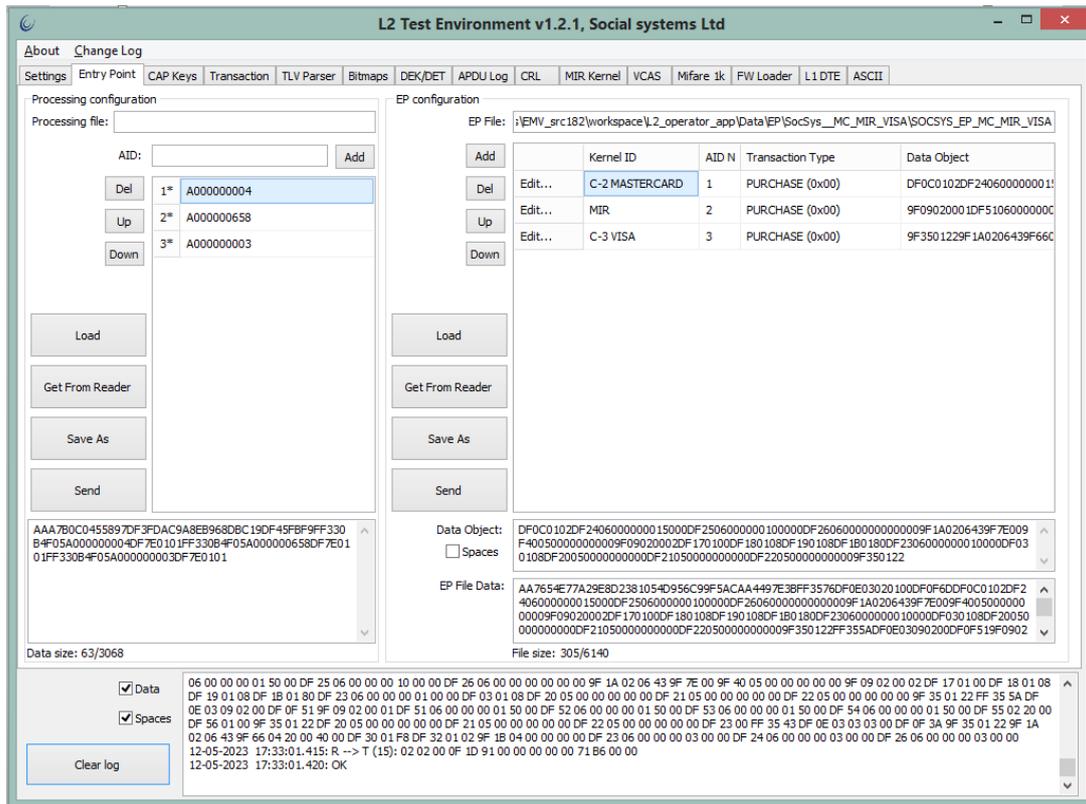


Рисунок 4 - Результат выполнения команды отправки настроек ПС устройству

2.2.8. Команда чтения из ПЗУ устройства списка настроек для поддерживаемых платёжных систем (0x9A)

Команда используется для проверки или контроля данных списка настроек платёжных систем. При выполнении команды на хост с устройства передаётся список настроек, хранимый в ПЗУ ридера.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x9A
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x9A
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>, <EP_LIST (0xXXXX)>}, где

ERROR - результат выполнения команды на устройстве,
EP_LIST - список настроек.

Пример:

```
>> 02 00 00 00 00 00 9A 03 00 00 00 00 00 00 6E
<< 02 33 01 81 51 9A 00 00 00 00 00 00 00 FE 00 00 AA F2 35 F4 88 5F 9E A1 F1 E3 1E
96 53 CE 3D 3C 85 E4 B5 DE 70 FF 35 76 DF 0E 03 02 01 00 DF 0F 6D DF 0C 01 02 DF 24
06 00 00 00 01 50 00 DF 25 06 00 00 00 10 00 00 DF 26 06 00 00 00 00 00 00 9F 1A 02
06 43 9F 7E 00 9F 40 05 00 00 00 00 00 9F 09 02 00 02 DF 17 01 00 DF 18 01 08 DF 19
01 08 DF 1B 01 80 DF 23 06 00 00 00 01 00 00 DF 03 01 08 DF 20 05 00 00 00 00 00 DF
21 05 00 00 00 00 00 DF 22 05 00 00 00 00 00 9F 35 01 22 FF 35 5A DF 0E 03 09 02 00
DF 0F 51 DF 55 02 20 00 9F 35 01 22 DF 20 05 00 00 00 00 00 DF 21 05 00 00 00 00 00
DF 22 05 00 00 00 00 00 DF 56 01 00 9F 09 02 00 01 DF 51 06 00 00 00 01 50 00 DF 52
06 00 00 00 01 50 00 DF 53 06 00 00 00 01 50 00 DF 54 06 00 00 00 01 50 00 DF 23 00
FF 35 43 DF 0E 03 03 03 00 DF 0F 3A 9F 35 01 22 9F 1A 02 06 43 9F 66 04 20 00 40 00
DF 30 01 F8 DF 32 01 02 9F 1B 04 00 00 00 00 DF 23 06 00 00 00 03 00 00 DF 24 06 00
00 00 03 00 00 DF 26 06 00 00 00 03 00 00
```

ERROR = 0x0000.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Get From Reader', в секции 'EP Configuration', на вкладке 'Entry Point'.

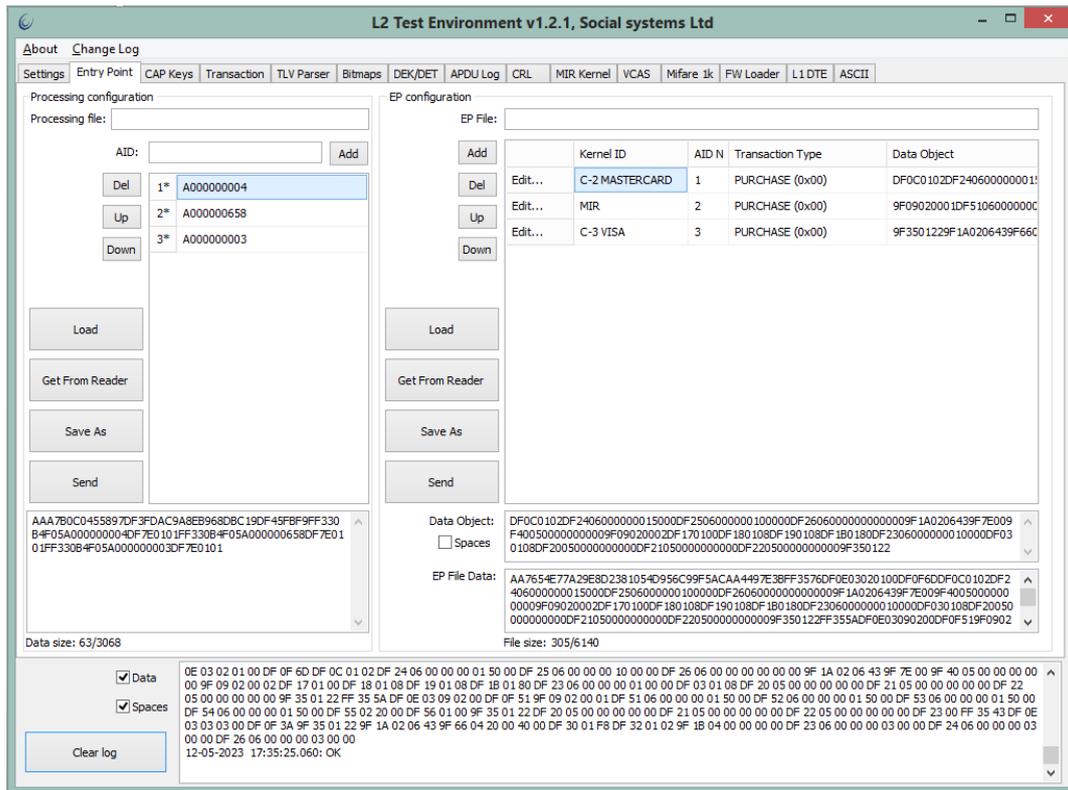


Рисунок 5 - Результат выполнения команды чтения настроек ПС из ПЗУ устройства.

2.2.9. Команда занесения в ПЗУ устройства открытых ключей платёжных систем (0x93)

Команда используется для передачи на устройство списка открытых ключей платёжных систем для CDA (Combined Data Authentication).

Максимальный размер списка открытых ключей с учётом его формата - 4092 байта.

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x93
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды:

{<CAPK_LIST (0xXXXX)>}, где

CAPK_LIST - список открытых ключей платёжных систем, поддерживаемых на устройстве.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0002
datablock_crc	0xXXXX
cmd	0x93
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>}, где

ERROR - результат выполнения команды на устройстве.

Пример:

```
>> 02 F1 05 7D 48 93 03 00 0D 73 A1 30 73 A0 00 00 00 04 05 01 B0 03 B8 04 8A BC
30 C9 0D 97 63 36 54 3E 3F D7 09 1C 8F E4 80 0D F8 20 ED 55 E7 E9 48 13 ED 00 55 5B
57 3F EC A3 D8 4A F6 13 1A 65 1D 66 CF F4 28 4F B1 3B 63 5E DD 0E E4 01 76 D8 BF 04
B7 FD 1C 7B AC F9 AC 73 27 DF AA 8A A7 2D 10 DB 3B 8E 70 B2 DD D8 11 CB 41 96 52 5E
A3 86 AC C3 3C 0D 9D 45 75 91 64 69 C4 E4 F5 3E 8E 1C 91 2C C6 18 CB 22 DD E7 C3 56
8E 90 02 2E 6B BA 77 02 02 E4 52 2A 2D D6 23 D1 80 E2 15 BD 1D 15 07 FE 3D C9 0C A3
10 D2 7B 3E FC CD 8F 83 DE 30 52 CA D1 E4 89 38 C6 8D 09 5A AC 91 B5 F3 7E 28 BB 49
EC 7E D5 97 A0 00 00 00 04 06 01 F8 03 CB 26 FC 83 0B 43 78 5B 2B CE 37 C8 1E D3 34
62 2F 96 22 F4 C8 9A AE 64 10 46 B2 35 34 33 88 3F 30 7F B7 C9 74 16 2D A7 2F 7A 4E
C7 5D 9D 65 73 36 86 5B 8D 30 23 D3 D6 45 66 76 25 C9 A0 7A 6B 7A 13 7C F0 C6 41 98
AE 38 FC 23 80 06 FB 26 03 F4 1F 4F 3B B9 DA 13 47 27 0F 2F 5D 8C 60 6E 42 09 58 C5
F7 D5 0A 71 DE 30 14 2F 70 DE 46 88 89 B5 E3 A0 86 95 B9 38 A5 0F C9 80 39 3A 9C BC
E4 4A D2 D6 4F 63 0B B3 3A D3 F5 F5 FD 49 5D 31 F3 78 18 C1 D9 40 71 34 2E 07 F1 BE
C2 19 4F 60 35 BA 5D ED 39 36 50 0E B8 2D FD A6 E8 AF B6 55 B1 EF 3D 0D 7E BF 86 B6
6D D9 F2 9F 6B 1D 32 4F E8 B2 6C E3 8A B2 01 3D D1 3F 61 1E 7A 59 4D 67 5C 44 32 35
0E A2 44 CC 34 F3 87 3C BA 06 59 29 87 A1 D7 E8 52 AD C2 2E F5 A2 EE 28 13 20 31 E4
8F 74 03 7E 3B 34 AB 74 7F A0 00 00 06 58 31 03 F0 01 00 01 DB 2E 92 58 EA 94 6B E4
5E B4 DD 8C C5 EB E9 C0 95 84 C1 85 62 52 22 37 2D E4 05 0A B0 44 C6 EE 75 4D 37 4E
75 15 6B 4F B2 63 25 7B 7D 4D B9 DB 28 B6 94 38 09 80 A7 5C C7 27 45 3B 80 16 E0 6A
1D B1 CA 5E 1C 3E BB 33 14 A9 00 25 D3 7D FC 9C 88 CA 2F A0 43 4F 60 71 3A D3 6F 9D
49 3E B4 79 4E 3F 96 2C 9E C1 4B 6A CF DF C3 C0 94 EB 1A E8 04 22 29 A1 38 7C F8 64
```

```

0F 19 89 13 03 BE 95 23 80 DB F8 20 ED BD 9C 3F 08 98 E1 46 73 3B 87 EE 9F 48 76 82
11 D8 C2 8A 60 42 E6 54 90 54 CA 60 90 E3 1F 6B 00 F5 9A 19 14 16 9E 8D 55 C2 C2 44
FF E8 17 FD 9A 33 09 11 2D D0 8E 56 4F 7C 44 EE 0C 3E 99 67 62 0D 94 2C 95 6B 0F 3B
93 42 27 58 76 84 80 2B 75 1B 07 07 4F 9F B0 63 F2 31 DC EE B8 99 70 18 58 46 D9 4F
B1 47 77 AE 9D E4 34 4D A0 00 00 06 58 32 03 C8 01 00 01 D4 D0 A1 A2 62 21 E0 3E 4D
B8 14 61 F4 7D B1 93 76 94 83 3F 97 3A 38 91 CA 63 27 01 42 F5 EC 67 31 8E 2B A7 B2
83 39 C7 8F 17 0A 21 B9 93 67 76 69 AA 47 19 4A AB 89 45 10 A9 37 8E F1 77 F9 E2 59
08 72 28 C7 38 30 FF 59 3E E5 E4 DD 4B D4 BB E9 50 FA 44 87 61 BA DE 6F 50 93 C5 62
B5 82 B8 E0 D0 7E 03 07 A6 30 34 A4 CF 69 B4 69 2B 10 39 05 21 97 DB F8 8F CD 1C 77
CF E8 65 41 05 93 CE 8D 6D EB 41 FE F7 A1 02 37 4C C0 6D 28 D8 BB AA B5 8B 60 A5 D3
D9 D2 38 CE 3F 2D 9F 3A 2C A6 3E 15 AD D8 39 63 85 56 F4 CA 8D 59 B3 9D AB B1 14 D3
55 80 83 D8 EB 9C 7F 97 DD F0 2B 7D 94 7F 3C B1 67 CB 86 9C C4 C5 47 A0 00 00 06 58
33 03 90 01 00 01 A8 30 6E 9E 57 9A A0 71 AE 06 DC 87 B8 DC 5E 1F 61 8B 6F 37 DE 4A
FB BA B9 A3 7E 21 73 8C 5E 19 BC 4A 13 06 89 A1 1E D1 0D D6 BE DD FD A6 31 8E 44 AE
2A EB 5E 5C 62 AF CF BF 54 71 71 35 EE 0D C0 71 55 61 9D 20 74 D6 6D BA A0 1A DF DA
76 38 A6 61 C8 EA 31 BE 6D 05 30 BE BF BA 9E 4C 74 88 73 C6 57 58 87 76 1F A4 E0 E2
C5 30 F1 D8 1E A3 E2 F0 40 8B 13 FD CA 97 59 5D 69 5E 92 B4 52 6A 86 F0 33 AE 95 C7
98 72 AF F7 16 5C CF 7A 08 FD A0 00 00 00 03 08 01 B0 03 D9 FD 6E D7 5D 51 D0 E3 06
64 BD 15 70 23 EA A1 FF A8 71 E4 DA 65 67 2B 86 3D 25 5E 81 E1 37 A5 1D E4 F7 2B CC
9E 44 AC E1 21 27 F8 7E 26 3D 3A F9 DD 9C F3 5C A4 A7 B0 1E 90 70 00 BA 85 D2 49 54
C2 FC A3 07 48 25 DD D4 C0 C8 F1 86 CB 02 0F 68 3E 02 F2 DE AD 39 69 13 3F 06 F7 84
51 66 AC EB 57 CA 0F C2 60 34 45 46 98 11 D2 93 BF EF BA FA B5 76 31 B3 DD 91 E7 96
BF 85 0A 25 01 2F 1A E3 8F 05 AA 5C 4D 6D 03 B1 DC 2E 56 86 12 78 59 38 BB C9 B3 CD
3A 91 0C 1D A5 5A 5A 92 18 AC E0 F7 A2 12 87 75 26 82 F1 58 32 A6 78 D6 E1 ED 0B A0
00 00 00 03 09 01 F8 03 9D 91 22 48 DE 0A 4E 39 C1 A7 DD E3 F6 D2 58 89 92 C1 A4 09
5A FB D1 82 4D 1B A7 48 47 F2 BC 49 26 D2 EF D9 04 B4 B5 49 54 CD 18 9A 54 C5 D1 17
96 54 F8 F9 B0 D2 AB 5F 03 57 EB 64 2F ED A9 5D 39 12 C6 57 69 45 FA B8 97 E7 06 2C
AA 44 A4 AA 06 B8 FE 6E 3D BA 18 AF 6A E3 73 8E 30 42 9E E9 BE 03 42 7C 9D 64 F6 95
FA 8C AB 4B FE 37 68 53 EA 34 AD 1D 76 BF CA D1 59 08 C0 77 FF E6 DC 55 21 EC EF 5D
27 8A 96 E2 6F 57 35 9F FA ED A1 94 34 B9 37 F1 AD 99 9D C5 C4 1E B1 19 35 B4 4C 18
10 0E 85 7F 43 1A 4A 5A 6B B6 51 14 F1 74 C2 D7 B5 9F DF 23 7D 6B B1 DD 09 16 E6 44
D7 09 DE D5 64 81 47 7C 75 D9 5C DD 68 25 46 15 F7 74 0E C0 7F 33 0A C5 D6 7B CD 75
BF 23 D2 8A 14 08 26 C0 26 DB DE 97 1A 37 CD 3E F9 B8 DF 64 4A C3 85 01 05 01 EF C6
50 9D 7A 41 DA 65 38 56 DF 69 2F 97 61 7E 76 12 1C 98 71 10 5B CE E5 FC
<< 02 02 00 0F 1D 93 00 00 00 00 00 30 D4 00 00

```

ERROR = 0x0000.

В примере приведён действующий на июнь 2022 года список открытых ключей платёжных систем: Mastercard, МИР, Visa.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Send', на вкладке 'CAP Keys'.

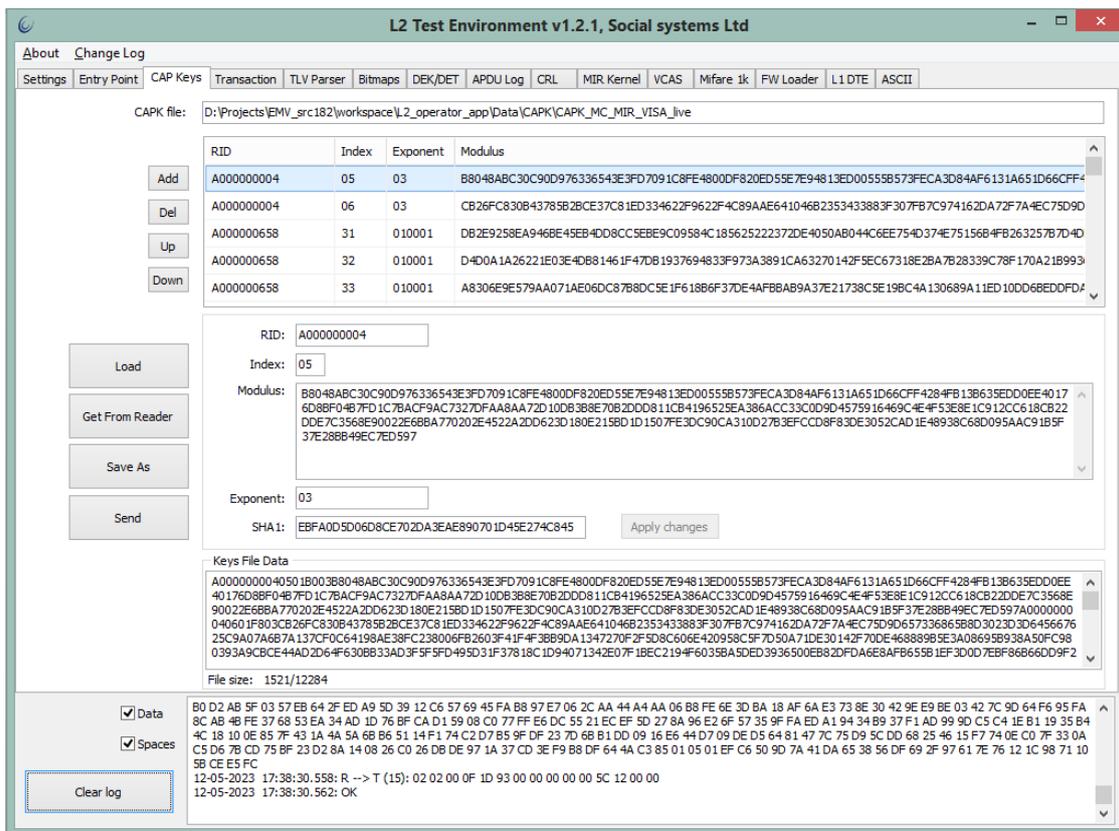


Рисунок 6 - Результат выполнения команды отправки открытых ключей устройству.

2.2.10. Команда чтения из ПЗУ устройства открытых ключей платёжных систем (0x9C)

Команда используется для проверки или контроля списка открытых ключей платёжных систем, хранимого в ПЗУ устройства. При выполнении данной команды на хост передаётся список открытых ключей платёжных систем, хранимый на устройстве.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x9C
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX


```

76 82 11 D8 C2 8A 60 42 E6 54 90 54 CA 60 90 E3 1F 6B 00 F5 9A 19 14 16 9E 8D 55 C2
C2 44 FF E8 17 FD 9A 33 09 11 2D D0 8E 56 4F 7C 44 EE 0C 3E 99 67 62 0D 94 2C 95 6B
0F 3B 93 42 27 58 76 84 80 2B 75 1B 07 07 4F 9F B0 63 F2 31 DC EE B8 99 70 18 58 46
D9 4F B1 47 77 AE 9D E4 34 4D A0 00 00 06 58 32 03 C8 01 00 01 D4 D0 A1 A2 62 21 E0
3E 4D B8 14 61 F4 7D B1 93 76 94 83 3F 97 3A 38 91 CA 63 27 01 42 F5 EC 67 31 8E 2B
A7 B2 83 39 C7 8F 17 0A 21 B9 93 67 76 69 AA 47 19 4A AB 89 45 10 A9 37 8E F1 77 F9
E2 59 08 72 28 C7 38 30 FF 59 3E E5 E4 DD 4B D4 BB E9 50 FA 44 87 61 BA DE 6F 50 93
C5 62 B5 82 B8 E0 D0 7E 03 07 A6 30 34 A4 CF 69 B4 69 2B 10 39 05 21 97 DB F8 8F CD
1C 77 CF E8 65 41 05 93 CE 8D 6D EB 41 FE F7 A1 02 37 4C C0 6D 28 D8 BB AA B5 8B 60
A5 D3 D9 D2 38 CE 3F 2D 9F 3A 2C A6 3E 15 AD D8 39 63 85 56 F4 CA 8D 59 B3 9D AB B1
14 D3 55 80 83 D8 EB 9C 7F 97 DD F0 2B 7D 94 7F 3C B1 67 CB 86 9C C4 C5 47 A0 00 00
06 58 33 03 90 01 00 01 A8 30 6E 9E 57 9A A0 71 AE 06 DC 87 B8 DC 5E 1F 61 8B 6F 37
DE 4A FB BA B9 A3 7E 21 73 8C 5E 19 BC 4A 13 06 89 A1 1E D1 0D D6 BE DD FD A6 31 8E
44 AE 2A EB 5E 5C 62 AF CF BF 54 71 71 35 EE 0D C0 71 55 61 9D 20 74 D6 6D BA A0 1A
DF DA 76 38 A6 61 C8 EA 31 BE 6D 05 30 BE BF BA 9E 4C 74 88 73 C6 57 58 87 76 1F A4
E0 E2 C5 30 F1 D8 1E A3 E2 F0 40 8B 13 FD CA 97 59 5D 69 5E 92 B4 52 6A 86 F0 33 AE
95 C7 98 72 AF F7 16 5C CF 7A 08 FD A0 00 00 00 03 08 01 B0 03 D9 FD 6E D7 5D 51 D0
E3 06 64 BD 15 70 23 EA A1 FF A8 71 E4 DA 65 67 2B 86 3D 25 5E 81 E1 37 A5 1D E4 F7
2B CC 9E 44 AC E1 21 27 F8 7E 26 3D 3A F9 DD 9C F3 5C A4 A7 B0 1E 90 70 00 BA 85 D2
49 54 C2 FC A3 07 48 25 DD D4 C0 C8 F1 86 CB 02 0F 68 3E 02 F2 DE AD 39 69 13 3F 06
F7 84 51 66 AC EB 57 CA 0F C2 60 34 45 46 98 11 D2 93 BF EF BA FA B5 76 31 B3 DD 91
E7 96 BF 85 0A 25 01 2F 1A E3 8F 05 AA 5C 4D 6D 03 B1 DC 2E 56 86 12 78 59 38 BB C9
B3 CD 3A 91 0C 1D A5 5A 5A 92 18 AC E0 F7 A2 12 87 75 26 82 F1 58 32 A6 78 D6 E1 ED
0B A0 00 00 00 03 09 01 F8 03 9D 91 22 48 DE 0A 4E 39 C1 A7 DD E3 F6 D2 58 89 92 C1
A4 09 5A FB D1 82 4D 1B A7 48 47 F2 BC 49 26 D2 EF D9 04 B4 B5 49 54 CD 18 9A 54 C5
D1 17 96 54 F8 F9 B0 D2 AB 5F 03 57 EB 64 2F ED A9 5D 39 12 C6 57 69 45 FA B8 97 E7
06 2C AA 44 A4 AA 06 B8 FE 6E 3D BA 18 AF 6A E3 73 8E 30 42 9E E9 BE 03 42 7C 9D 64
F6 95 FA 8C AB 4B FE 37 68 53 EA 34 AD 1D 76 BF CA D1 59 08 C0 77 FF E6 DC 55 21 EC
EF 5D 27 8A 96 E2 6F 57 35 9F FA ED A1 94 34 B9 37 F1 AD 99 9D C5 C4 1E B1 19 35 B4
4C 18 10 0E 85 7F 43 1A 4A 5A 6B B6 51 14 F1 74 C2 D7 B5 9F DF 23 7D 6B B1 DD 09 16
E6 44 D7 09 DE D5 64 81 47 7C 75 D9 5C DD 68 25 46 15 F7 74 0E C0 7F 33 0A C5 D6 7B
CD 75 BF 23 D2 8A 14 08 26 C0 26 DB DE 97 1A 37 CD 3E F9 B8 DF 64 4A C3 85 01 05 01
EF C6 50 9D 7A 41 DA 65 38 56 DF 69 2F 97 61 7E 76 12 1C 98 71 10 5B CE E5 FC

```

ERROR = 0x0000.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Get From Reader', на вкладке 'CAP Keys'.

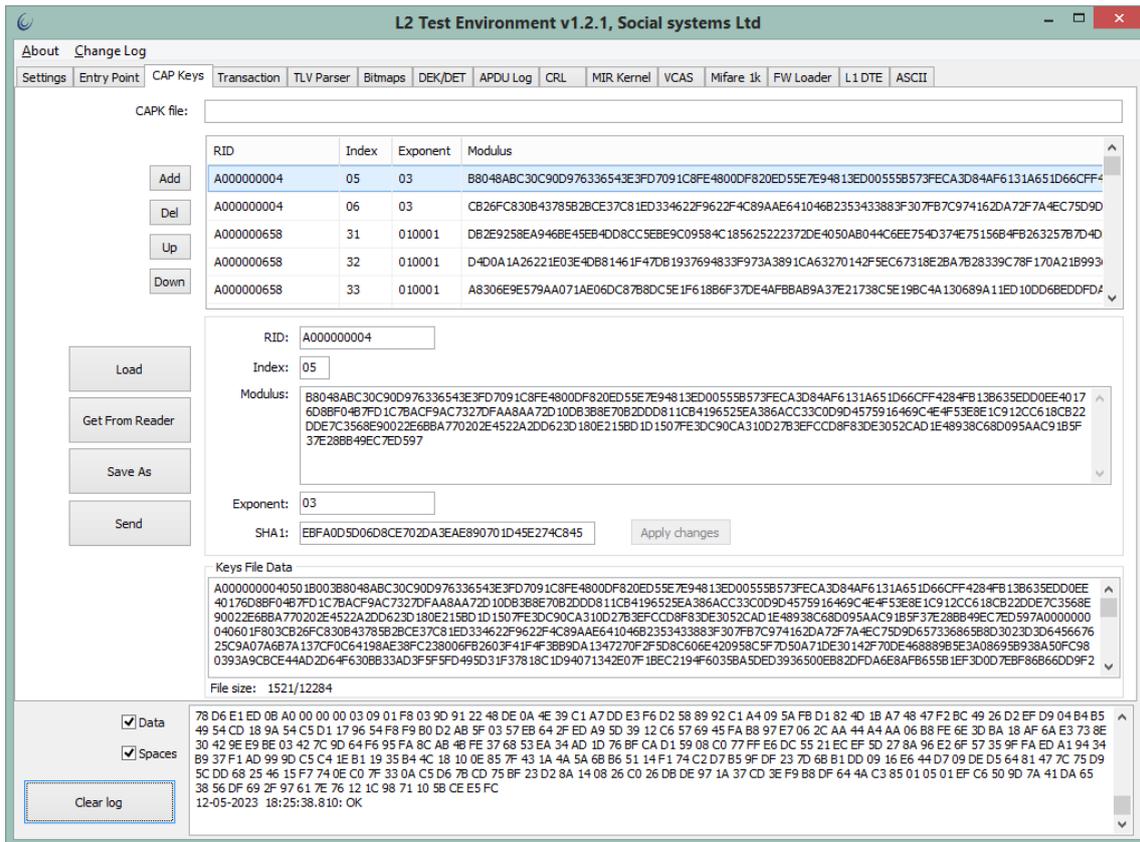


Рисунок 7 - Результат выполнения команды чтения открытых ключей из ПЗУ устройства.

2.2.11. Команда занесения в ПЗУ устройства списка отозванных сертификатов платёжных систем (0xA3)

Устройство хранит в своей энергонезависимой памяти список отозванных сертификатов платёжных систем. Данная команда служит для занесения списка отозванных сертификатов или CRL (Certification Revocation List) в ПЗУ ридера.

Максимальный размер списка с учётом его формата - 5372 байта.

Блок заголовка команды:

ver	0x02
datablock_len	0xFFFF
datablock_crc	0xFFFF
cmd	0xA3
direction	0x03
timeout	0x0000
err	0x00
sync	0x00

header_crc	0xXX
------------	------

Блок данных команды:

{<CRL (0xXXXX)>}, где

CRL - список отозванных сертификатов платёжных систем.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0002
datablock_crc	0xXXXX
cmd	0xA3
direction	0x00
timeout	0x0000
err	0xXX
RFU	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>}, где

ERROR - результат выполнения команды на устройстве.

Пример:

```
>> 02 1D 00 40 8C A3 03 00 0D 73 A1 30 72 A0 00 00 00 03 51 00 00 01 3E 8D D1 18
6F BA 84 A5 8B 7E 16 F5 D0 6D 43 FD 4F 19 77 89
<< 02 02 00 0F 1D A3 00 00 00 00 00 30 C0 00 00
```

ERROR = 0x0000.

В примере приведён тестовый набор отозванных сертификатов для ПС Visa, состоящий из одного элемента с индексом 51.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Send', на вкладке 'CRL'.

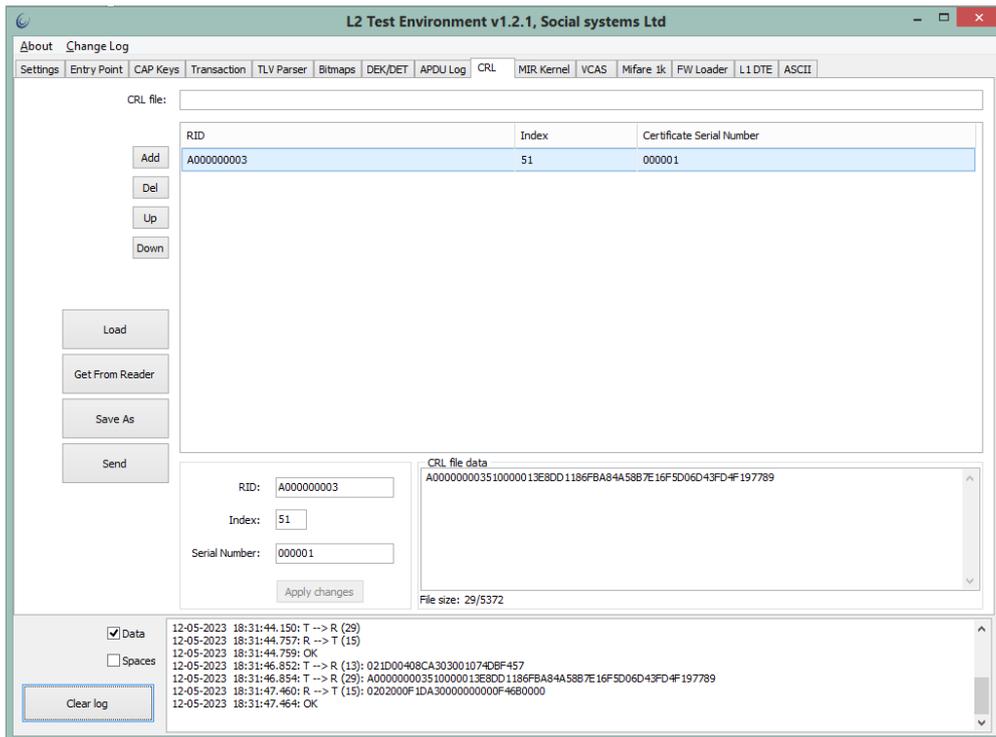


Рисунок 8 - Результат выполнения команды отправки списка отозванных сертификатов устройству.

2.2.12. Команда чтения из ПЗУ устройства списка отозванных сертификатов платёжных систем (0xA4)

Команда используется для проверки или контроля списка отозванных сертификатов платёжных систем. При выполнении данной команды на хост передаётся список отозванных сертификатов платёжных систем.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0xA4
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0xA4
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>, <CRL (0xXXXX)>}, где

ERROR - результат выполнения команды на устройстве,
CRL - список отозванных сертификатов платёжных систем.

Пример:

```
>> 02 00 00 00 00 00 A4 03 00 00 00 00 00 05
<< 02 1F 00 0E A7 A4 00 00 00 00 00 00 1A 00 00 A0 00 00 00 03 51 00 00 01 3E 8D
D1 18 6F BA 84 A5 8B 7E 16 F5 D0 6D 43 FD 4F 19 77 89
```

ERROR = 0x0000.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Get From Reader', на вкладке 'CRL'.

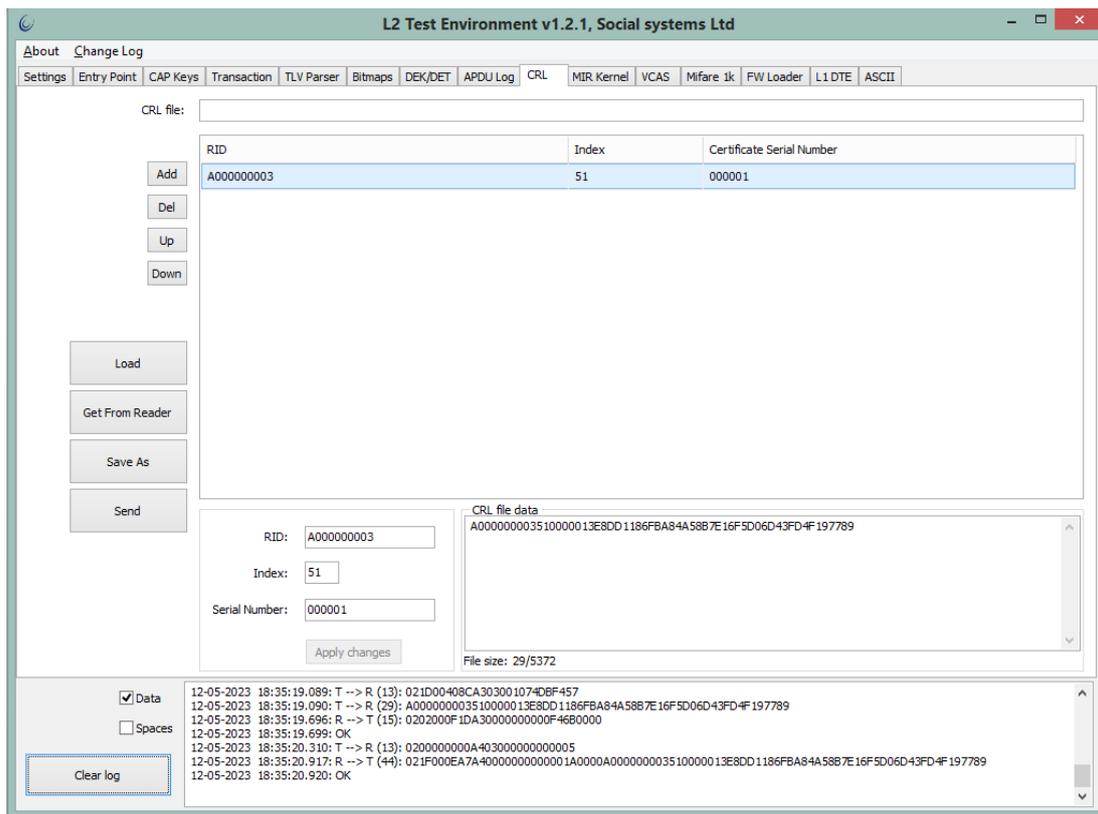


Рисунок 9 - Результат выполнения команды чтения списка отозванных сертификатов из ПЗУ устройства.

2.2.13. Команда передачи устройству системных настроек (0x98)

При выполнении данной команды устройству передаётся набор системных настроек таких, как: включение/отключение звукового сопровождения работы ридера, время ожидания карты в поле антенны и так далее.

Настройки не сохраняются в ПЗУ устройства, при каждом отключении или сбросе устройства настройки сбрасываются к своим значениям по умолчанию. Команда используется для инициализации ридера.

Блок заголовка команды:

ver	0x02
datablock_len	0x0004
datablock_crc	0xXXXX
cmd	0x98
direction	0x03
timeout	0x0000
err	0x00
sync	0x00

header_crc	0xXX
------------	------

Блок данных команды:

{<OPTS (4)>}, где
OPTS – 4 байта настроек.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0002
datablock_crc	0xXXXX
cmd	0x98
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>}, где

ERROR - результат выполнения команды на устройстве.

Пример:

```
>> 02 04 00 DC B5 98 03 00 00 00 00 00 61 06 00 02 07
<< 02 02 00 0F 1D 98 00 00 00 00 00 00 DC 00 00
```

ERROR = 0x0000.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Apply Settings', на вкладке 'Settings'.

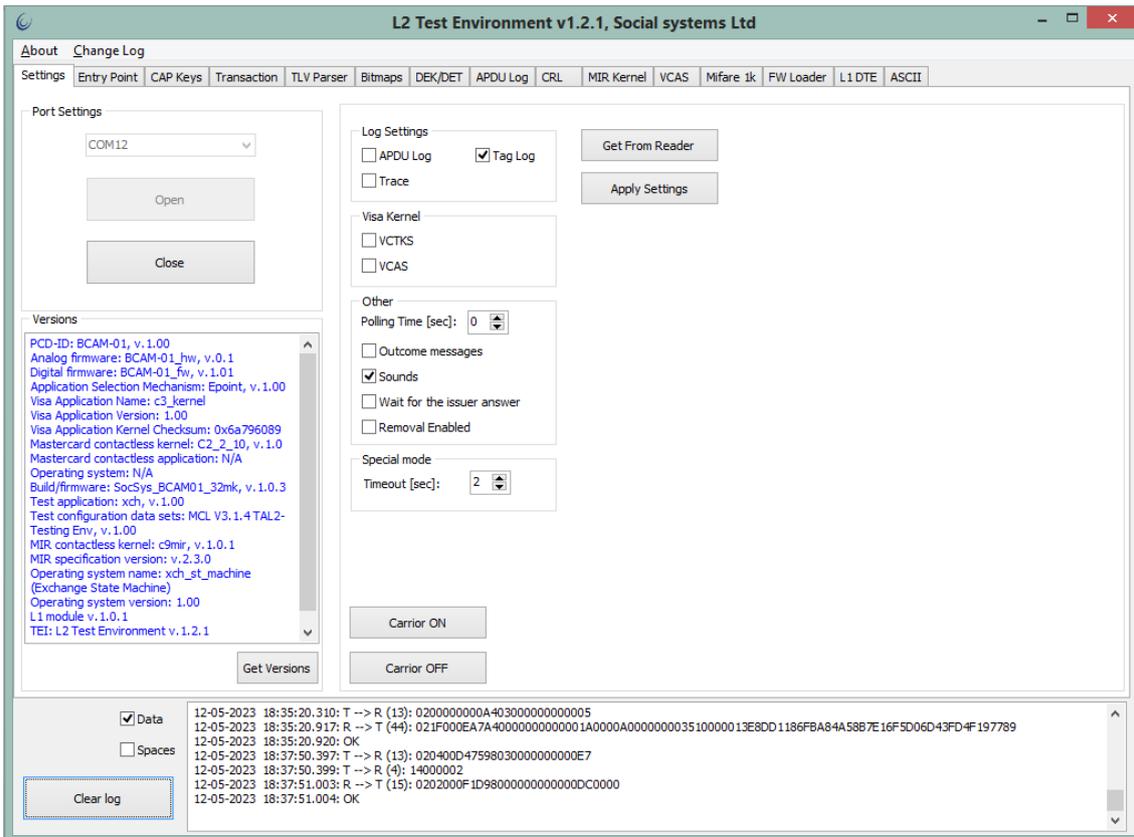


Рисунок 10 - Результат выполнения команды отправки устройству системных настроек.

2.2.14. Команда чтения с устройства текущих настроек (0x99)

При выполнении данной команды с устройства передаётся набор текущих системных настроек.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x99
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0006
datablock_crc	0xXXXX
cmd	0x99
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<OPTS (4)>}, где

OPTS – 4 байта настроек.

Пример:

```
>> 02 00 00 00 00 00 99 03 00 00 00 00 00 29
<< 02 06 00 82 F4 99 00 00 00 00 00 00 72 00 00 04 00 00 03
```

04 00 00 03 - системные настройки.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Get From Reader', на вкладке 'Settings'.

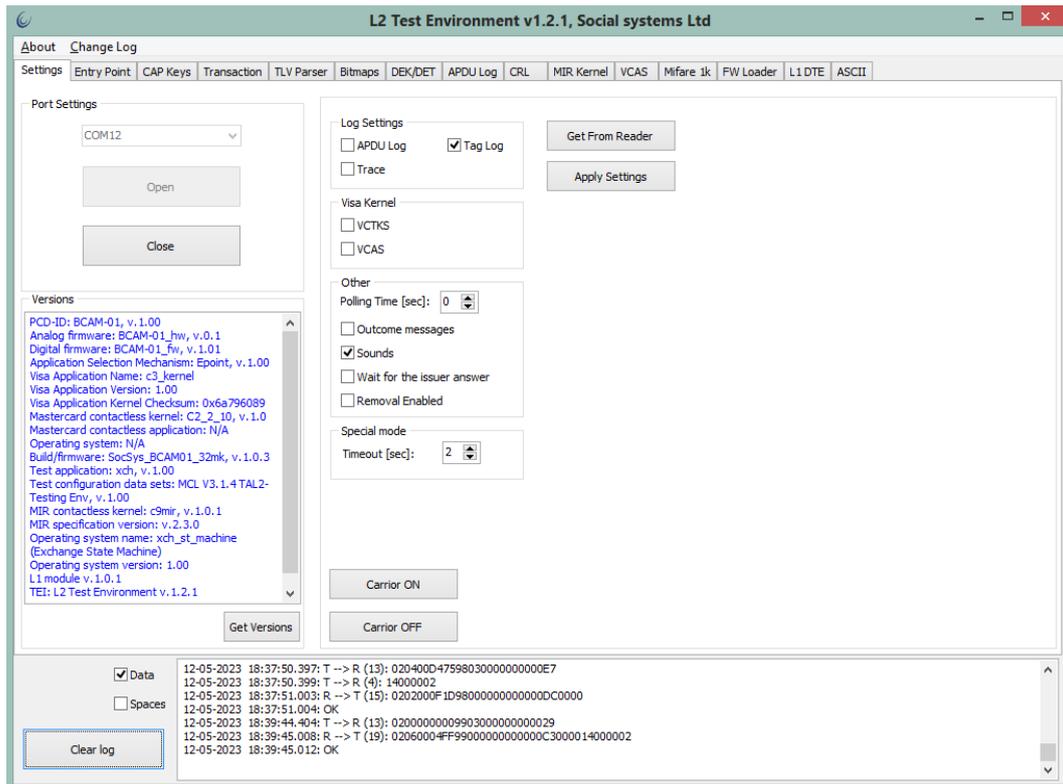


Рисунок 11 - Результат выполнения команды чтения с устройства текущих системных настроек.

2.2.15. Команда получения с устройства информации о платёжных ядрах (0x9D)

При выполнении этой команды устройство присылает информацию о поддерживаемых бесконтактных платёжных ядрах. Информация присылается в виде ASCII-данных.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0XXXXX
cmd	0x9D
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0XX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x9D
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)> <EMV_VER_STR (0xXXXX)>}, где

EMV_VER_STR – строка с версиями поддерживаемых платёжных ядер.

Пример:

```
>> 02 00 00 00 00 00 9D 03 00 00 00 00 00 00 DD
<< 02 B1 02 9C 84 9D 00 00 00 00 00 00 00 EA 00 00 50 43 44 2D 49 44 3A 20 42 43 41
4D 2D 30 31 2C 20 76 2E 31 2E 30 30 0D 0A 41 6E 61 6C 6F 67 20 66 69 72 6D 77 61 72
65 3A 20 42 43 41 4D 2D 30 31 5F 68 77 2C 20 76 2E 30 2E 31 0D 0A 44 69 67 69 74 61
6C 20 66 69 72 6D 77 61 72 65 3A 20 42 43 41 4D 2D 30 31 5F 66 77 2C 20 76 2E 31 2E
30 31 0D 0A 41 70 70 6C 69 63 61 74 69 6F 6E 20 53 65 6C 65 63 74 69 6F 6E 20 4D 65
63 68 61 6E 69 73 6D 3A 20 45 70 6F 69 6E 74 2C 20 76 2E 31 2E 30 30 0D 0A 56 69 73
61 20 41 70 70 6C 69 63 61 74 69 6F 6E 20 4E 61 6D 65 3A 20 63 33 5F 6B 65 72 6E 65
6C 0D 0A 56 69 73 61 20 41 70 70 6C 69 63 61 74 69 6F 6E 20 56 65 72 73 69 6F 6E 3A
20 31 2E 30 30 0D 0A 56 69 73 61 20 41 70 70 6C 69 63 61 74 69 6F 6E 20 4B 65 72 6E
65 6C 20 43 68 65 63 6B 73 75 6D 3A 20 30 78 62 34 64 30 65 61 63 64 0D 0A 4D 61 73
74 65 72 63 61 72 64 20 63 6F 6E 74 61 63 74 6C 65 73 73 20 6B 65 72 6E 65 6C 3A 20
43 32 5F 32 5F 31 30 2C 20 76 2E 31 2E 30 0D 0A 4D 61 73 74 65 72 63 61 72 64 20 63
6F 6E 74 61 63 74 6C 65 73 73 20 61 70 70 6C 69 63 61 74 69 6F 6E 3A 20 4E 2F 41 0D
0A 4F 70 65 72 61 74 69 6E 67 20 73 79 73 74 65 6D 3A 20 4E 2F 41 0D 0A 42 75 69 6C
64 2F 66 69 72 6D 77 61 72 65 3A 20 53 6F 63 53 79 73 5F 42 43 41 4D 30 31 5F 33 32
6D 6B 2C 20 76 2E 31 2E 30 2E 33 0D 0A 54 65 73 74 20 61 70 70 6C 69 63 61 74 69 6F
6E 3A 20 78 63 68 2C 20 76 2E 31 2E 30 30 0D 0A 54 65 73 74 20 63 6F 6E 66 69 67 75
72 61 74 69 6F 6E 20 64 61 74 61 20 73 65 74 73 3A 20 4D 43 4C 20 56 33 2E 31 2E 34
20 54 41 4C 32 2D 54 65 73 74 69 6E 67 20 45 6E 76 2C 20 76 2E 31 2E 30 30 0D 0A 4D
49 52 20 63 6F 6E 74 61 63 74 6C 65 73 73 20 6B 65 72 6E 65 6C 3A 20 63 39 6D 69 72
2C 20 76 2E 31 2E 30 2E 30 0D 0A 4D 49 52 20 73 70 65 63 69 66 69 63 61 74 69 6F 6E
20 76 65 72 73 69 6F 6E 3A 20 76 2E 32 2E 31 2E 30 0D 0A 4F 70 65 72 61 74 69 6E 67
20 73 79 73 74 65 6D 20 6E 61 6D 65 3A 20 78 63 68 5F 73 74 5F 6D 61 63 68 69 6E 65
20 28 45 78 63 68 61 6E 67 65 20 53 74 61 74 65 20 4D 61 63 68 69 6E 65 29 20 0D 0A
4F 70 65 72 61 74 69 6E 67 20 73 79 73 74 65 6D 20 76 65 72 73 69 6F 6E 3A 20 31 2E
30 30 0D 0A
```

Результат:

```
ERROR = 0,  
Версии: "PCD-ID: BCAM-01, v.1.00  
Analog firmware: BCAM-01_hw, v.0.1  
Digital firmware: BCAM-01_fw, v.1.01  
Application Selection Mechanism: Epoint, v.1.00  
Visa Application Name: c3_kernel  
Visa Application Version: 1.00  
Visa Application Kernel Checksum: 0xb4d0eacd  
Mastercard contactless kernel: C2_2_10, v.1.0  
Mastercard contactless application: N/A  
Operating system: N/A  
Build/firmware: SocSys_BCAM01_32mk, v.1.0.3  
Test application: xch, v.1.00  
Test configuration data sets: MCL V3.1.4 TAL2-Testing Env, v.1.00  
MIR contactless kernel: c9mir, v.1.0.0  
MIR specification version: v.2.1.0  
Operating system name: xch_st_machine (Exchange State Machine)  
Operating system version: 1.00"
```

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Get Versions', на вкладке 'Settings'.

2.2.16. Команда начала транзакции (0x94)

При выполнении данной команды ридер ожидает внесение карты или мобильного устройства с платёжным приложением в поле действия антенны. Время ожидания задаётся хостом в системных настройках. При внесении карты в поле действия антенны, устройство осуществляет работу с картой по установленному EMV-спецификациями протоколу: L1 (EMV Book D), Entry Point (EMV Book B) и далее L2 по спецификации бесконтактного ядра одной из поддерживаемых платёжных систем.

Особенность этой команды заключается в том, что между отправкой хостом этой команды и получением ответной посылки на команду по протоколу PB3P могут происходить следующие события:

- отправка с хоста команды отмены транзакции, в этом случае хост получит ответ на команду начала транзакции с соответствующим кодом ошибки;
- получение хостом системных сообщений Outcome и сигнала MSG UI, данные сообщения включаются и отключаются в системных настройках;
- получение хостом от устройства команд прямого обмена данными с ядром платёжной системы и последующим ответом на эти команды (для Mastercard - это так называемый DEK/DET протокол, для ПС МИР - это Data Exchange протокол, описание этих протоколов можно найти в спецификациях соответствующих платёжных систем, для Visa такой протокол не предусмотрен);

- получение хостом от устройства команд с логом обмена с картой, данные команды не предусматривают ответ, получение логов опционально, то есть задаётся системными настройками;

- отправка хостом команды ответа эмитента в случае онлайн-транзакции, данная команда тоже опциональна, то есть может не отправляться в ответ на получение Outcome с Online Request, в этом случае в системных настройках задаётся настройка не ждать ответа от эмитента.

В блоке данных команды начала транзакции передаются обязательные EMV-теги.

Таблица 2. Теги для передачи с командой начала транзакции.

Тег	EMV. Имя тега	Комментарий
9C	Transaction Type	Тип операции. Тег необязательный, если тип операции - Purchase. Если тег отсутствует, то устройство автоматически выбирает тип операции Purchase (0x00).
9F02	Amount, Authorised	Сумма транзакции. Обязательный тег.
9F03	Amount, Other	Дополнительная сумма транзакции, например, Cashback. Обязательный тег.
5F2A	Transaction Currency Code	Код валюты операции. Для России - это 0643. Обязательный тег.
9A	Transaction Date	Дата транзакции. Обязательный тег.
9F21	Transaction Time	Время транзакции. Обязательный тег.

С командой могут также передаваться и другие EMV-теги.

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x94
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды:

{<tag 0x9C> <tag 0x9F02> <tag 0x9F03> <tag 0x5F2A> <tag 0x9A> <tag 0x9F21>}

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x94
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)> [TAGLOG(0xXXXX)]}, где

ERROR – 2 байта результата, которые должны интерпретироваться как знаковое, 16-битовое целое, при этом значения, меньшие 0 - это возможные ошибки, 0 - неопределённый результат, должен интерпретироваться как ошибка, 1 - транзакция была отклонена, 2 - транзакция одобрена, 3 - Online Request (то есть на устройстве транзакция считается успешной, а одобрение или отклонение транзакции - это уже дело эмитента карты и логики работы хоста);

TAGLOG - теги транзакции в формате BER-TLV.

Пример:

```
>> 02 25 00 26 9A 94 03 00 00 00 00 00 9C 9C 01 00 9F 02 06 00 00 00 00 11 00 9F
03 06 00 00 00 00 00 00 5F 2A 02 06 43 9A 03 22 06 29 9F 21 03 19 35 00
<< 02 B8 04 F1 31 94 00 00 00 00 00 00 4F 8D FF 9F 1A 02 06 43 9F 35 01 22 9F 40
05 00 00 00 00 00 9F 7E 01 01 DF 81 1B 01 80 9F 09 02 00 02 DF 81 03 01 08 DF 81 17
01 00 DF 81 18 01 08 DF 81 19 01 08 DF 81 20 05 00 00 00 00 00 DF 81 21 05 00 00 00
00 00 DF 81 22 05 00 00 00 00 00 DF 81 23 06 00 00 00 01 00 00 DF 81 24 06 00 00 00
01 50 00 DF 81 25 06 00 00 00 10 00 00 DF 81 26 06 00 00 00 00 00 DF 81 1A 03 9F
6A 04 DF 81 30 01 0D DF 81 0C 01 02 9F 6D 02 00 01 DF 81 1E 01 F0 DF 81 2C 01 F0 DF
81 1C 02 01 2C DF 81 1D 01 00 DF 81 2D 03 00 00 13 DF 81 33 02 00 32 DF 81 32 02 00
14 DF 81 36 02 01 2C DF 81 37 01 32 DF 81 34 02 00 12 DF 81 35 02 00 18 DF 81 27 02
13 88 84 07 A0 00 00 00 04 10 10 50 0A 4D 61 73 74 65 72 43 61 72 64 87 01 01 9F 11
01 01 9F 12 0A 4D 61 73 74 65 72 43 61 72 64 5F 2D 08 72 75 65 6E 66 72 64 65 9F 4D
02 0B 0A 9F 6E 07 06 43 00 00 30 30 00 9F 37 04 8B EA 60 E1 9C 01 00 9F 02 06 00 00
00 00 11 00 9F 03 06 00 00 00 00 00 00 5F 2A 02 06 43 9A 03 22 06 29 9F 21 03 19 35
00 DF 81 0E 01 00 DF 81 0F 01 00 82 02 19 80 94 10 08 01 01 00 10 01 01 01 18 01 02
00 20 01 02 00 9F 62 06 00 00 00 38 00 00 9F 63 06 00 00 00 00 E0 E0 56 3E 42 35 35
33 36 39 31 33 37 35 30 38 35 38 39 31 38 5E 20 2F 5E 32 32 30 35 32 30 31 30 30 30
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
```

```

30 30 30 9F 64 01 03 9F 65 02 00 0E 9F 66 02 07 70 9F 6B 13 55 36 91 00 00 00 00 0 D2
20 52 01 00 00 00 00 00 00 0F 9F 67 01 03 57 13 55 36 91 00 00 00 00 00 D2 20 52 01
12 36 55 79 00 00 0F 5A 08 55 36 91 00 00 00 00 00 5F 24 03 22 05 31 5F 25 03 11 09
01 5F 28 02 06 43 5F 34 01 01 8C 27 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03
9C 01 9F 37 04 9F 35 01 9F 45 02 9F 4C 08 9F 34 03 9F 21 03 9F 7C 14 9F 07 02 FF 00
9F 08 02 00 02 9F 0D 05 B4 50 84 00 00 9F 0E 05 00 00 00 00 00 9F 0F 05 B4 70 84 80
00 9F 42 02 06 43 9F 4A 01 82 8F 01 05 9F 32 01 03 92 24 0C D4 A8 20 3D 78 3A ED 7A
46 7B 94 D7 EB 5D D4 11 D3 E7 D8 5A E0 1C 17 EF 72 B4 61 F6 9A DC 96 94 45 03 73 90
81 B0 A5 64 50 43 9B 72 5D 38 C3 43 6F E2 7B 6B 48 8E 88 34 F4 E7 D9 DE 6F EA 64 28
66 E2 C1 84 42 F0 08 5D 1C C1 5B A2 A3 E5 B7 05 C5 66 FA 09 33 02 82 07 CB 7E 42 EB
0B 4D 1A 05 75 27 C3 A3 A6 E5 D8 33 8E FE 75 B3 6F 1D CE B7 4C 97 92 31 10 E5 6A B2
A3 61 A8 37 86 9D 6F BF 5F 71 10 98 96 00 F4 2E 64 06 9C 9B C9 67 5B EA 0F 6A 89 59
B3 06 6D A8 EB 7E F0 36 98 D5 15 4E FC 6E C3 06 0E 88 3F C5 70 82 87 F8 40 6E 2A 86
E6 78 CF E0 B8 A9 4C 79 E9 24 1A DB AB 26 BC 2B 39 89 94 DC 04 EB 33 34 1B 17 F8 F6
D1 F6 13 5C AE 3E 0B F0 EF F9 9F 47 01 03 9F 46 81 B0 07 F8 04 8D 4D CA EF E0 C1 68
DB F6 21 C1 E1 05 96 28 EE 27 00 89 FF 4E B5 BC 02 F5 32 03 16 41 3E AA 09 7E 6F F7
89 3F 55 9F A8 C4 FA 3F 3F 8E 78 EA 73 B1 43 9E C5 AA 88 19 FF B5 82 48 E0 83 DC 69
F2 90 41 3D CF 02 7D 41 C7 F6 AC BF DA 33 99 A1 E2 AA AB D0 66 37 84 F9 55 9A 28 C0
69 52 1E 3E BE 5C DB 0E 78 BC 8E 83 8A 6E 26 7F D2 C7 12 27 31 FD 33 3C F5 B3 09 FD
37 A8 FD 18 FB 39 20 A1 C2 A3 C8 9F 62 F6 8C 6E EB 29 8E 85 75 2B B1 67 38 C0 E6 E2
CE CC 63 8D 0F DC D2 05 FD 99 EE 9D 6E EA 95 EB 5F 14 8F 70 87 13 AD 09 7C ED 8E 0E
00 00 00 00 00 00 00 00 42 03 1E 03 1F 03 9F 33 03 00 08 08 9F 34 03 1F 03 02 95 05
00 40 00 00 01 9F 27 01 80 9F 36 02 05 D4 9F 4B 78 7B 97 D8 E9 3D 2C 02 86 80 11 19
84 39 8D 39 D8 82 BB 22 DE 6E C9 E1 AD 67 1D AA 47 52 6A 52 07 A5 FA DF ED C2 BD A7
B4 4D A2 AE DC C7 3E 04 75 D4 D4 57 10 29 A1 F8 BC FA 37 A8 37 EB CD F9 A6 52 CB B1
A8 53 6D 25 29 09 0C 9A 37 DE 8F A3 DB 8C C9 55 FF 0D 7B 6A C1 F0 D6 D5 A0 D0 FA E6
87 31 24 04 D2 79 5E 07 9C 6A 92 85 DC 5B A5 2B 89 48 22 7D 4B C5 DF F5 18 9F 10 12
01 10 A0 40 03 22 00 00 00 00 00 00 00 90 00 01 FF

```

ERROR = 0xFF8D = -115 - по таблице 3.1 "Obtaining key and cert for CDA", "Нет подходящего ключа для CDA или ошибка в формате сертификата (например, просрочен)";

TAGLOG - в тегах транзакции PAN заменён на '55 36 91 00 00 00 00 00' (для примера использовалась реальная карта).

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Start Transaction', на вкладке 'Transaction'.

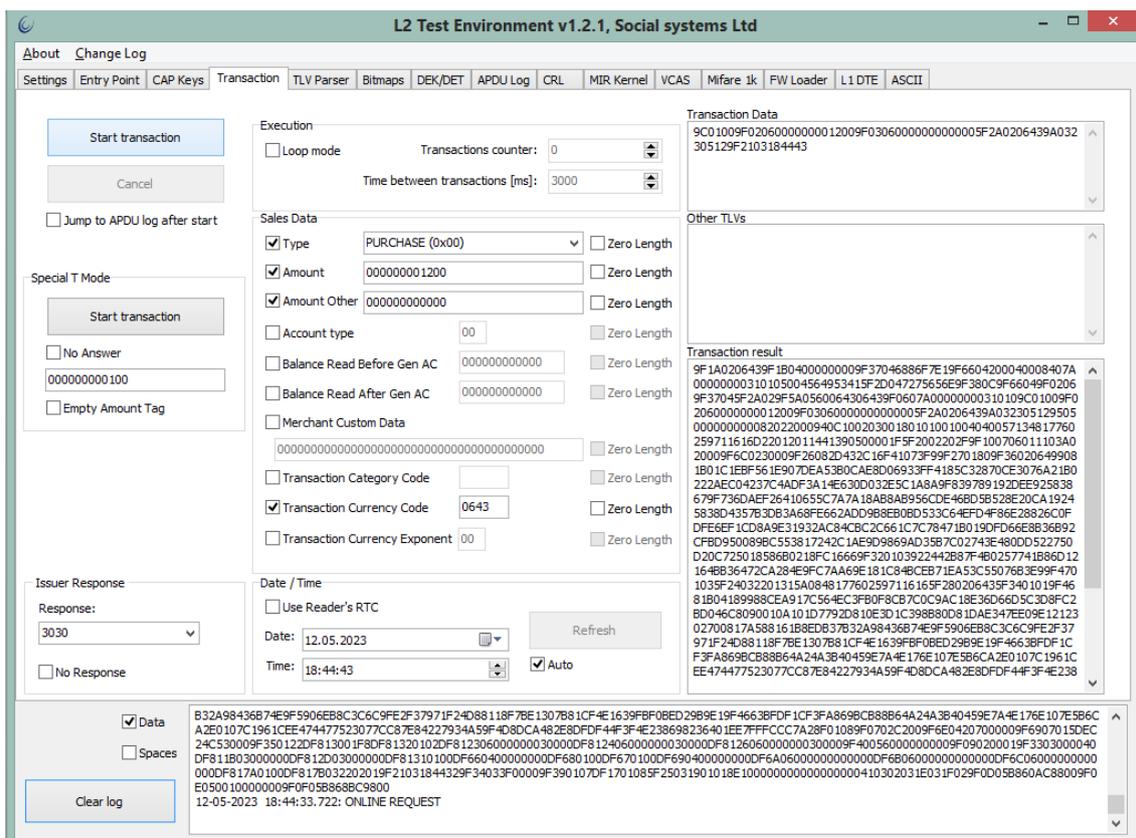


Рисунок 12 - Результат выполнения команды начала транзакции.

2.2.17. Команда отмены транзакции (0x95)

По требованиям платёжных систем оператор должен иметь возможность вмешаться в транзакцию и отменить её. Для этого существует данная команда. Транзакция может быть отменена вплоть до процедуры чтения данных с карты.

Команда имеет смысл только после предварительной команды начала транзакции, во всех остальных случаях команда будет считаться неизвестной, и ответ на неё будет ошибка протокола 0x07 - 'Неизвестная команда'. В случае отправки команды после начала транзакции ответ на команду не предполагается.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x95
direction	0x03
timeout	0x0000
err	0x00
sync	0x00

header_crc	0xXX
------------	------

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXXX
cmd	0x95
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Ответ на команду отмены транзакции может быть только в случае ошибки протокола PB3P.

Блок данных ответной посылки: отсутствует.

Пример:

В примере приведена команда начала транзакции и следом команда отмены транзакции.

```
>> 02 25 00 84 59 94 03 00 00 00 00 00 13 9C 01 00 9F 02 06 00 00 00 00 11 00 9F 03
06 00 00 00 00 00 00 5F 2A 02 06 43 9A 03 22 06 30 9F 21 03 12 36 52
```

```
>> 02 00 00 00 00 95 03 00 00 00 00 00 2C
```

```
<< 02 02 00 A9 28 94 00 00 00 00 00 00 40 7E FF
```

Как видно из дампа, после команды завершения транзакции, устройство присылает результат транзакции ERROR = 0xFF7E = -130 - 'Транзакция отменена пользователем'.

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Cancel', на вкладке 'Transaction'. При этом транзакция должна уже быть запущена.

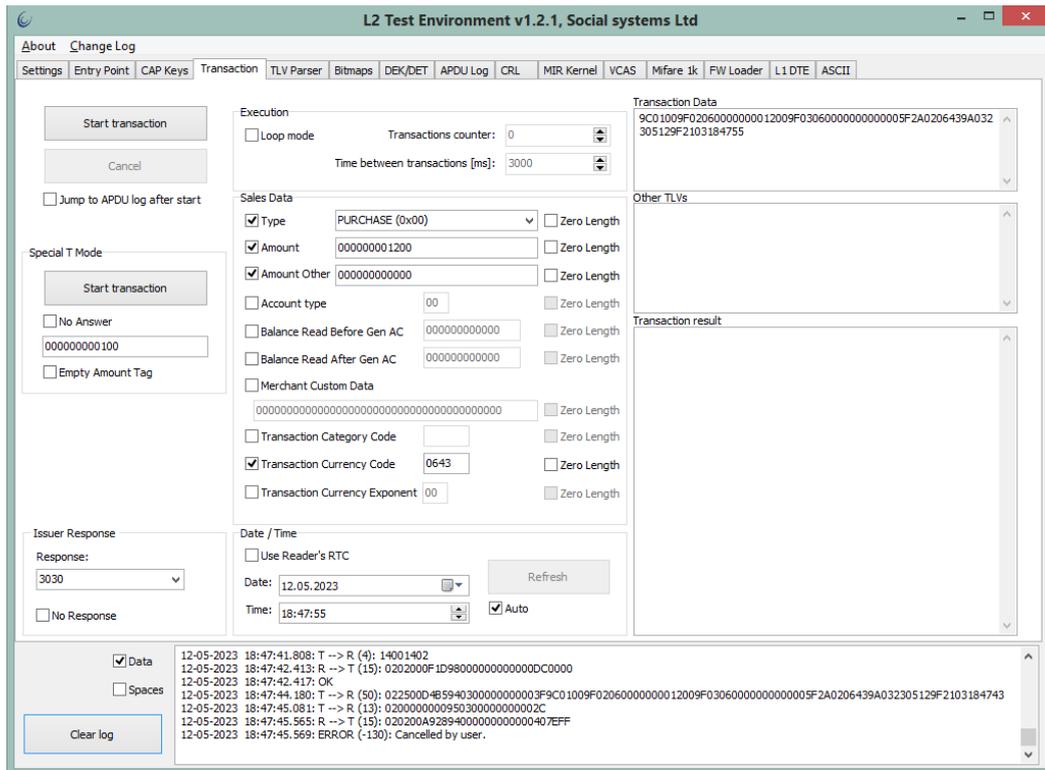


Рисунок 13 - Результат выполнения команды отмены транзакции.

2.2.18. Команда получения хостом системных сообщений Outcome (0xA0)

По требованию EMVCo платёжные ядра во время транзакции должны генерировать формализованные сообщения – Outcome.

Команда может быть прислана устройством во время проведения транзакции, то есть после команды начала транзакции и до ответа на команду начала транзакции.

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0xA0
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных команды:

{<ERROR (2)> <tag DF8129> <tag DF8116> [tag FF8111] [tag FF8110 {<tag DF8115>}]},

где:

ERROR - оставлен для совместимости, всегда 0x0000;

tag DF8129 - тег, содержащий структуру сообщения OUT;

tag DF8116 - тег, содержащий структуру сообщения UI outcome;

tag FF8111 - темплейт Data Record (DR), содержащий обязательные теги транзакции;

tag FF8110 - темплейт Discretionary Data (DD), содержащий единственный тег со структурой DD;

tag DF8115 - тег, содержащий структуру Error Indication (EI).

Пример:

PAN в примере заменен.

```
<< 02 F8 00 E9 77 A0 00 00 00 00 00 00 3E 00 00 DF 81 29 08 30 F0 F8 00 B0 F0 FF
00 DF 81 16 16 1B 00 00 00 00 72 75 65 6E 00 00 00 00 00 00 00 00 00 00 00 00 FF
81 11 81 B4 9F 39 01 07 DF 17 01 08 57 13 48 17 76 00 00 00 00 00 00 D2 20 12 01 14 41
39 05 00 00 1F 5A 08 48 17 76 00 00 00 00 00 00 5F 20 02 20 2F 5F 24 03 22 01 31 5F 2A
02 06 43 5F 34 01 01 82 02 20 00 95 05 00 00 00 00 00 9A 03 22 06 30 9C 01 00 9F 02
06 00 00 00 00 11 00 9F 03 06 00 00 00 00 00 00 9F 10 07 06 01 11 03 A0 20 00 9F 1A
02 06 43 9F 26 08 FA 95 51 6F EB 8F C0 FB 9F 27 01 80 9F 34 03 3F 00 00 9F 36 02 05
71 9F 37 04 4C BB 13 80 9F 6E 04 20 70 00 00 9F 33 03 00 00 40 9F 21 03 12 36 52 9F
06 07 A0 00 00 00 03 10 10 9F 66 04 20 00 40 00 00 00 00 00 00 00 FF 81 10 0A DF 81 15
06 00 00 00 00 00 FF 00 00 00 00, где:
```

ERROR = 00 00;

OUT = DF 81 29 08 30 F0 F8 00 B0 F0 FF 00;

UI outcome = DF 81 16 16 1B 00 00 00 00 72 75 65 6E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00;

DR = FF 81 11 81 B4 9F 39 01 07 DF 17 01 08 57 13 48 17 76 00 00 00 00 00 00 D2 20 12 01 14 41 39 05 00 00 1F 5A 08 48 17 76 00 00 00 00 00 00 5F 20 02 20 2F 5F 24 03 22 01 31 5F 2A 02 06 43 5F 34 01 01 82 02 20 00 95 05 00 00 00 00 00 9A 03 22 06 30 9C 01 00 9F 02 06 00 00 00 00 11 00 9F 03 06 00 00 00 00 00 9F 10 07 06 01 11 03 A0 20 00 9F 1A 02 06 43 9F 26 08 FA 95 51 6F EB 8F C0 FB 9F 27 01 80 9F 34 03 3F 00 00 9F 36 02 05 71 9F 37 04 4C BB 13 80 9F 6E 04 20 70 00 00 9F 33 03 00 00 40 9F 21 03 12 36 52 9F 06 07 A0 00 00 00 03 10 10 9F 66 04 20 00 40 00 00 00 00 00 00 00 00 00;

DD = FF 81 10 0A DF 81 15 06 00 00 00 00 00 FF 00 00 00 00, где

EI = DF 81 15 06 00 00 00 00 00 FF 00 00 00 00.

Пример расшифровки структуры:

```
OUT - OUTCOME PARAMETR SET
Data: 30F0F800B0F0FF00
Status: ONLINE REQUEST
Start: N/A
Online Resp Data: N/A
CVM: NO CVM
UI Req. On Outcome: 1
UI Req. On Restart: 0
Receipt: N/A
```

DR: 1
DD: 1
Alternate Interface: N/A
Field Off: N/A
Removal Timeout: 0

OUT - DISCRETIONARY DATA

Data: DF8115060000000000FF
Error Indication 0000000000FF
L1 error: OK
L2 error: OK
L3 error: OK
SW1SW2: 0000
Message on error: N/A

OUT - DATA RECORD

9F39 07
DF17 08
57 4817760000000000D22012011441390500001F
5A 4817760000000000
5F20 202F
5F24 220131
5F2A 0643
5F34 01
82 2000
95 0000000000
9A 220630
9C 00
9F02 000000001100
9F03 000000000000
9F10 06011103A02000
9F1A 0643
9F26 FA95516FEB8FC0FB
9F27 80
9F34 3F0000
9F36 0571
9F37 4CBB1380
9F6E 20700000
9F33 000040
9F21 123652
9F06 A0000000031010
9F66 20004000

OUT - UI REQUEST DATA

Data: 1B000000007275656E000000000000000000000000
Message: AUTHORISING, PLEASE WAIT
Status: NOT READY
Hold Time: 000000
Language Preference: 7275656E00000000
Value Qualifier: NONE
Value: 000000000000
Currency Code: 0000


```
Status:                READY TO READ
Hold Time:             000000
Language Preference:   000000000000000000
Value Qualifier:       NONE
Value:                 00000000000000
Currency Code:         0000
```

2.2.20. Команды обмена данными по протоколу Mastercard DEK/DET

Платёжное ядро Mastercard поддерживает протокол DEK/DET, при котором ядро обменивается напрямую с хостом по заранее составленным сценариям. Инициатором обмена в этом случае является ядро, которое по признакам в элементе из списка настроек поддерживаемых платёжных систем, относящемуся к платёжному приложению Mastercard, первым присылает команду DEK (Data Exchange Kernel), в ответ ожидает одну или несколько команд DET (Data Exchange Terminal). Подробнее о протоколе можно узнать из спецификации бесконтактного платёжного ядра Mastercard.

Обмен по протоколу может состояться только после команды начала транзакции и до её завершения.

2.2.20.1. Команда DEK (0x97)

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x97
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных команды:

{<ERROR (2)> <DEK>}, где

ERROR - оставлен для совместимости, всегда 0x0000;

DEK - данные от ядра.

2.2.20.2. Команда DET (0x96)

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x96
direction	0x03
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных команды:

{<ERROR (2)> <DET>}, где

ERROR - оставлен для совместимости, всегда 0x0000;

DET - данные от терминала.

Пример:

```
<< 02 0D 00 B8 D9 97 00 00 00 00 00 00 9A 00 00 FF 81 04 00 DF 81 06 03 DF 81 12
>> 02 05 00 63 47 96 03 00 00 00 00 00 1A DF 81 12 01 82
<< 02 0E 00 52 09 97 00 00 00 00 00 00 65 00 00 FF 81 04 04 82 02 18 80 DF 81 06
00
>> 02 05 00 E8 88 96 03 00 00 00 00 00 3E DF 81 10 01 01
```

2.2.21. Команда протокола ПС «МИР» Data Exchange (0xA5)

По аналогии с Mastercard платёжное ядро МИР поддерживает свой протокол обмена данными между ядром и хостом. Протокол носит название Data Exchange. Обмен инициализирует ядро при наличии в настройках ядра признака необходимости такого обмена (тер Data Exchange Tag List). В ответ ожидает ту же команду с обновлённым списком данных: Terminal TPM Capabilities и Amount. Подробнее о протоколе в спецификации бесконтактного платёжного ядра МИР.

Обмен по протоколу может состояться только после команды начала транзакции и до её завершения.

Блок заголовка команды:

ver	0x02
-----	------

datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0xA5
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных команды:

{<Kernel Data>}, где

Kernel Data - данные ядра в рамках обмена по протоколу MIP Data Exchange.

Блок заголовка ответной команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0xA5
direction	0x03
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных команды:

{<Terminal Data>}, где

Terminal Data - данные терминала в рамках обмена по протоколу MIP Data Exchange.

Пример:

```
<< 02 0A 00 35 4D A5 00 00 00 00 00 00 C8 5A 08 22 01 38 94 00 00 00 30
>> 02 0E 00 61 17 A5 03 00 00 00 00 00 9D 9F 02 06 00 00 00 09 00 00 DF 55 02 E8
00
```

2.2.22. Команды логов обмена с картой (0x9E, 0x9F)

В программном обеспечении предусмотрен режим, в котором устройство логирует обмен данными с картой, ретранслируя на хост APDU-команды обмена. Данный режим подходит для тестирования, разбора ситуации, но не подходит для штатной работы устройства, так как на пересылку данных тратится дополнительное системное время, которое по итогу выходит за пределы временного интервала, отпущенного на транзакцию платёжными системами. Включение и выключение данного режима предусмотрено через системные настройки.

Команды присылаются устройством на хост только после отправки устройству команды начала транзакции.

Для удобства выделены две команды логирования: одна - для команд устройства - C-APDU (Command APDU), другая - для команд карты - R-APDU (Respond APDU). Устройство не ожидает ответа от на команды логирования.

2.2.22.1. Команда получения лога C-APDU (0x9E)

Блок заголовка команды:

ver	0x02
datablock_len	0xFFFF
datablock_crc	0xFFFF
cmd	0x9E
direction	0x00
timeout	0x0000
err	0xFF
sync	0x00
header_crc	0xFF

Блок данных команды:

{<ERROR (2)> <PCB (1)> <C-APDU (n)>}, где

ERROR - оставлен для совместимости, всегда 0x0000;

PCB - Protocol Control Byte - служебный байт протокола ISO14443, при сохранении данных команды или при выводе на экран можно игнорировать;

C-APDU - команда карте от устройства.

Ответная посылка на команду не предусмотрена.

2.2.22.2. Команда получения лога R-APDU (0x9F)

Блок заголовка команды:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x9F
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных команды:

{<ERROR (2)> <R-APDU (n)>}, где

ERROR - оставлен для совместимости, всегда 0x0000;

R-APDU - ответ карты на команду от устройства.

Ответная посылка на команду не предусмотрена.

Пример:

В примере приведён фрагмент лога обмена с картой: команда SELECT и ответ на неё.

```
<< 02 10 00 B5 93 9E 00 00 00 00 00 00 59 00 00 03 00 A4 04 00 07 A0 00 00 00 03
10 10 00
<< 02 38 00 D8 71 9F 00 00 00 00 00 00 50 00 00 6F 32 84 07 A0 00 00 00 03 10 10
A5 27 50 04 56 49 53 41 5F 2D 04 72 75 65 6E 9F 38 0C 9F 66 04 9F 02 06 9F 37 04 5F
2A 02 BF 0C 08 9F 5A 05 60 06 43 06 43 90 00
```

C-APDU: 00 A4 04 00 07 A0 00 00 00 03 10 10 00

R-APDU: 6F 32 84 07 A0 00 00 00 03 10 10 A5 27 50 04 56 49 53 41 5F 2D 04 72 75 65 6E 9F 38 0C 9F 66 04 9F 02 06 9F 37 04 5F 2A 02 BF 0C 08 9F 5A 05 60 06 43 06 43 90 00

2.2.23. Команда отправки устройству ответа эмитента (0xB3)

Команда пересылает устройству ответ эмитента. Команда работает только в случае PC Visa и присылается хостом на устройство до получения ответа на команду начала транзакции.

Блок заголовка команды:

ver	0x02
datablock_len	0x0002
datablock_crc	0xXXXX
cmd	0xB3
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды:

{<ISS_ANSW (2)>}, где

ISS_ANSW - ответ эмитента, коды ответа можно узнать в стандарте [ISO 8583]:1987, коды 0x3030, 0x3130 и 0x3131 - одобрение транзакции, все остальные - отклонение.

Блок заголовка ответной посылки: отсутствует.

Блок данных ответной посылки: отсутствует.

Пример:

```
>> 02 02 00 C9 2E B3 03 00 00 00 00 00 61 30 30
```

ISS_ANSW = 0x3030 = "00" - транзакция одобрена.

2.2.24. Команда включения поля антенны (0x70)

Поле антенны ридера включается при инициализации устройства, после транзакции остаётся включённым. Возникает потребность в ряде случаев иметь возможность отключить поле антенны и, соответственно, включить при необходимости. Кроме того, поле включается каждый раз при команде начала транзакции (команда 0x94) при условии, что результаты предпроцессинга позволяют продолжить транзакцию.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000

datablock_crc	0xXXXX
cmd	0x70
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x70
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>}

Пример:

```
>> 02 00 00 00 00 00 70 03 00 00 00 00 00 5A
<< 02 02 00 0F 1D 70 00 00 00 00 00 00 92 00 00
```

ERROR = 0

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Carrier ON', на вкладке 'Transaction'.

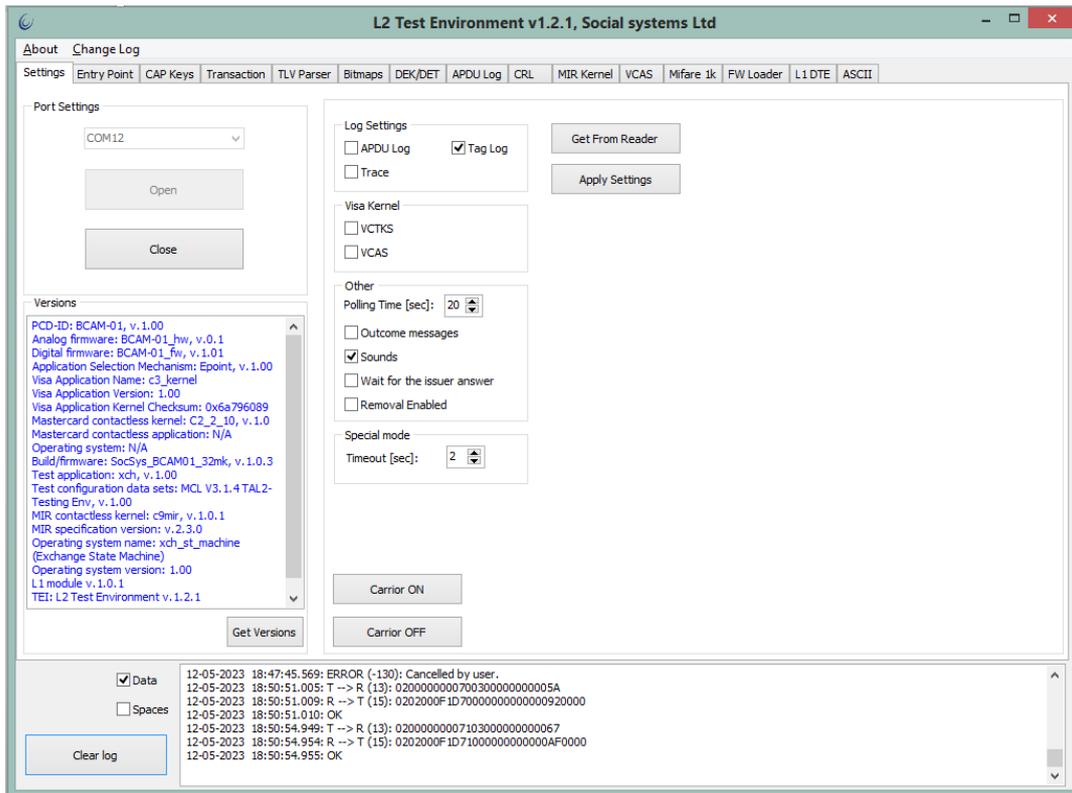


Рисунок 14 - Результат команд включения и отключения поля.

2.2.25. Команда отключения поля антенны (0x71)

Команда отключает поле антенны.

Блок заголовка команды:

ver	0x02
datablock_len	0x0000
datablock_crc	0xXXXX
cmd	0x71
direction	0x03
timeout	0x0000
err	0x00
sync	0x00
header_crc	0xXX

Блок данных команды: отсутствует.

Блок заголовка ответной посылки:

ver	0x02
-----	------

datablock_len	0xXXXX
datablock_crc	0xXXXX
cmd	0x71
direction	0x00
timeout	0x0000
err	0xXX
sync	0x00
header_crc	0xXX

Блок данных ответной посылки:

{<ERROR (2)>}

Пример:

```
>> 02 00 00 00 00 00 71 03 00 00 00 00 00 67
<< 02 02 00 0F 1D 71 00 00 00 00 00 00 AF 00 00

ERROR = 0
```

Для выполнения данной команды в приложении TEI следует нажать кнопку 'Carrier OFF', на вкладке 'Transaction' (Рисунок 14).

2.3. Передача команд управления во встроенные программные модули

В процессе обмена устройства с терминальной программой УПО получает команды от ведущего устройства и ретранслирует их в соответствующие программные модули, вызывая соответствующие подпрограммы. Возвращает результат работы подпрограммы терминальному ПО.

Могут вызываться подпрограммы модулей: Level 1 (например включение/выключение поля), ядер платёжных систем, если в них предусмотрена процедура прямого общения с терминальным ПО, например "DEK/DET"-обмен.