ОКБ ОСОБОЕ КОНСТРУКТОРСКОЕ БЮРО
СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УТВЕРЖДЕН

37222406.26.20.40.140.083:08 90–ЛУ

Специальное программное обеспечение «Резидентный компонент безопасности»

РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ

37222406.26.20.40.140.083:08 90

Подп. и дата | Инв. № дубл. | Взаим. инв. № | Подп. и дата | Инв. № подл.

Настоящий документ предназначен для специалистов, осуществляющих эксплуатацию специального программного обеспечения «Резидентный компонент безопасности» 37222406.26.20.40.140.083:08 (далее – СПО «РКБ», программное изделие) и содержит основные сведения о СПО «РКБ» и правила его эксплуатации.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

ICSP	–	In Circuit Serial Programming
USB	–	Universal Serial Bus
ИМС	–	Интегральная микросхема
ПО	–	Программное обеспечение
РКБ	–	Резидентный компонент безопасности
СВТ	–	Средство вычислительной техники
СДЗ	–	Средство доверенной загрузки
СПО	–	Специальное программное обеспечение

1. ОПИСАНИЕ И РАБОТА

1.1 Назначение

СПО «РКБ» предназначено для выполнения отчужденных от центрального процессора функций безопасности.

СПО «РКБ» имеет следующие функциональные возможности:

- генерацию, хранение и использование ключевой информации;
- хранение базы данных СПО СДЗ «Аккорд-МКТ»;
- передача случайных чисел от датчика случайных чисел;
- поддержка интерфейса USB.

1.2 Технические характеристики

Программное изделие скомпилировано в виде бинарного файла (файла firmware) и устанавливается на конфигурируемую ИМС по технологии ICSP.

СПО «РКБ» хранится и исполняется на ИМС как в составе самостоятельных изделий, так и как встроенной составной части СВТ.

2. ИСПОЛЬЗОВАНИЕ ПО НАЗНАЧЕНИЮ

2.1 Установка СПО «РКБ»

2.1.1 Установка СПО «РКБ» на аппаратную платформу в составе СВТ

Установка СПО «РКБ» на аппаратную платформу в составе микрокомпьютера m-TrusT производится с использованием утилиты AndroidTool.exe.

Для установки СПО «РКБ» необходимо выполнить следующие действия:

1) На ПЭВМ с Windows запустить AndroidTool.exe.

2) Подать питание на изделие.

3) Подключить USB-порт изделия к USB-порту ПЭВМ, зажав кнопку ubt (рядом с портом mini HDMI).

4) После появления надписи «Found One LOADER Device» кнопку ubt можно отпустить. В утилите нажать кнопку «Run» (Рис. 2.1).

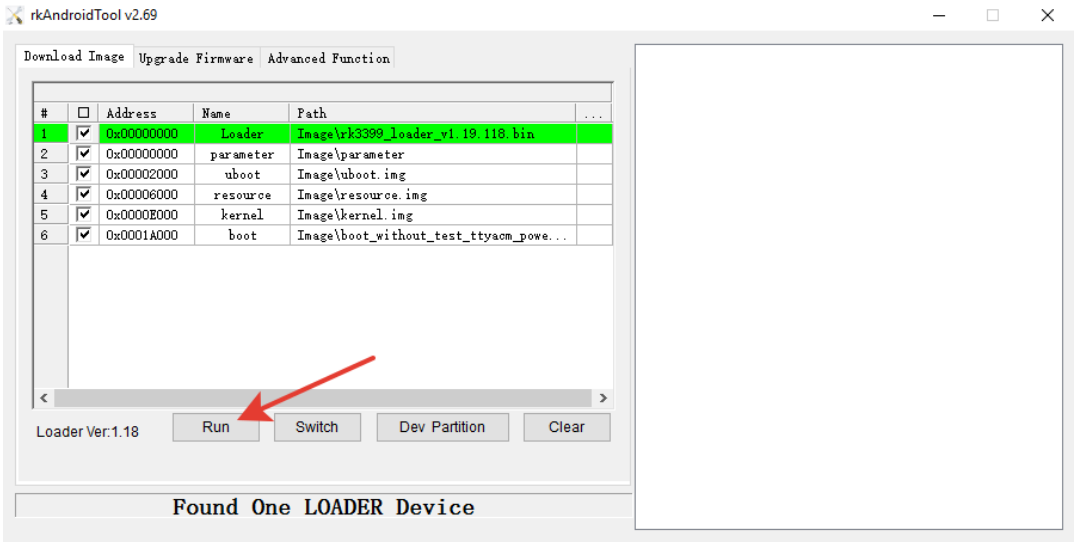

Рис. 2.1

Внимание: При выполнении первой прошивки процесс может прерваться ошибкой. В этом случае необходимо повторить действия 3), 4).

5) После окончания прошивки изделие автоматически перезапустится и начнет процедуры инициализации РКБ.

Внимание: Не отключайте питание изделия во процедуры прошивки.

6) Наблюдать за зеленым светодиодом (WK). Светодиод сначала будет гореть непрерывно, после чего начнет мигать. После окончания процедуры он погаснет.

7) После того, как зеленый светодиод погас и более не мигает, на ПЭВМ, запустить утилиту rkbgetmd5C и получить контрольную сумму СПО «РКБ», записанного в изделие. Сравнить значение контрольной суммы СПО «РКБ»,

установленного в изделии со значением контрольной суммы файла СПО «РКБ» - значения контрольных сумм должны совпасть, что свидетельствует о правильной установке СПО «РКБ».

2.1.2 Установка в самостоятельное изделие

Установка СПО «РКБ» в самостоятельное изделие производится с ПЭВМ (с операционной системой Windows x32), для подготовки которой к установке необходимо выполнить следующие действия:

1) При использовании ОС Windows XP SP3 необходимо установить обновление KB967048-v2 (http://support.microsoft.com/kb/967048).

2) Установить утилиту SAM-BA (sam-ba_2.12.exe) и обновление 2a к ней (sam-ba_2.12_patch2a.exe).

3) Заменить файл c:\Program Files\Atmel\sam-ba_2.12\tcl_lib\at91sam3u4-ek\at91sam3u4-ek.tcl файлом at91sam3u4-ek.tcl.

4) Добавить в каталог c:\Program Files\Atmel\sam-ba_2.12\tcl_lib\at91sam3u4-ek\ файл isp-flash-at91sam3u4.bin.

5) Запустить службу смарт-карты.

6) Установить ПО «АРМ Инициализации» (SetupSecretInitialization_v*.exe).

7) Заменить файл c:\Program Files\OKB SAPR JC\Secret\Initialization\InitializationConsole.exe файлом InitializationConsole.exe.

8) Запустить ПО «АРМ Инициализации». Убедиться, что список конфигураций пуст.

9) Закрыть ПО «АРМ Инициализации», нажав кнопку «Отмена».

10) Установить файл СПО «РКБ».

11) Запустить ПО «АРМ Инициализации». Убедиться, что в описаниях прошивок есть файл СПО «РКБ».

12) Выбрать конфигурацию «РКБ», нажать кнопку «ОК».

13) Закрыть ПО «АРМ Инициализации», нажав кнопки «Выход» и «Да».

Для установки СПО «РКБ» необходимо выполнить следующие действия:

1) Подключить самостоятельное изделие к USB-порту ПЭВМ. Убедиться, что драйвер AT91 USB to Serial Converter установлен.

2) В ПО «АРМ Инициализации» установить СПО «РКБ» стартовым загрузчиком, нажав кнопку «Запуск!». Убедиться, что инициализация завершилась успешно (в логах нет сообщений об ошибках, логи завершаются сообщением: «Количество не прошитых устройств: 0»). Нажать кнопку «Закрыть».

3) Для контроля правильности установки СПО «РКБ» запустить утилиту rkbgetmd5C и получить контрольную сумму СПО «РКБ». Сравнить значение контрольной суммы СПО «РКБ», установленного в СВТ со значением контрольной суммы файла СПО «РКБ» - значения контрольных сумм должны совпасть, что свидетельствует о правильной установке СПО «РКБ».

4) Отключить СВТ от USB-порта ПЭВМ, закрыть ПО «АРМ Инициализации», нажав кнопки «Выход» – «Да».

2.2 Работа с программой

Работа с СПО «РКБ», установленным на микроконтроллер РКБ, производится из прикладного ПО через прикладной интерфейс, реализуемый библиотекой librkb.

Набор функций, предоставляемых библиотекой:

Генерация неизвлекаемого ключа подписи

Команда подается прикладным ПО. По этой команде на основе случайной последовательности байт, полученных с ФДСЧ, firmware РКБ создается ключ подписи и записывается с во внутреннюю память РКБ.

Название команды: rkb_cmd_gen_sign_key

Входные параметры: отсутствуют

Выходные данные: код ошибки с возможными значениями: SIGN_KEY_GEN_ERR, SUCCESS (см. Таблицу 2.1)

Получение ключа проверки подписи

Команда подается прикладным ПО. По этой команде вырабатывается ключ проверки подписи, соответствующий неизвлекаемому ключу подписи, и передается в процессор.

Название команды: rkb_cmd_gen_verify_key

Входные параметры: отсутствуют

Выходные данные: код ошибки с возможными значениями: VERIFY_KEY_BUFFER_TOO_SMALL, VERIFY_KEY_GEN_ERR, SUCCESS (см. Таблицу 2.1), длина массива ключа проверки подписи (verify_key_len), массив ключа проверки подписи (verify_key)

Расшифрование сессионного ключа

Команда подается прикладным ПО. По этой команде полученный зашифрованный сессионный ключ расшифровывается на ключе защиты ключей,

который в свою очередь вырабатывается на основе неизвлекаемого ключа подписи и переданного ключа проверки подписи.

Название команды: rkb_cmd_extract_session_key

Входные параметры: длина массива ключа проверки подписи (verify_key_len), массив ключа проверки подписи (verify_key), длина массива зашифрованного сессионного ключа (crypted_session_key_len), массив зашифрованного сессионного ключа (crypted_session_key)

Выходные данные: код ошибки с возможными значениями: VERIFY_KEY_BUFFER_TOO_SMALL, CRYPTED_SESSION_KEY_BUFFER_TOO_SMALL, EXTRACT_SESSION_KEY_ERR, SUCCESS (см. Таблицу 2.1), длина массива сессионного ключа (session_key_len), сессионный ключ (session_key)

Получение последовательности случайных чисел

Команда подается прикладным ПО. По этой команде происходит получение случайных чисел с шумовых диодов, их обработка, контроль качества и запись в массив.

Название команды: rkb_cmd_get_rnd

Входные параметры: длина массива случайных чисел (len)

Выходные данные: код ошибки с возможными значениями: RND_GEN_ERR, SUCCESS (см. Таблицу 2.1), массив случайных чисел (buf)

Установка ключа подписи

Команда подается прикладным ПО. По этой команде происходит запись во внутреннюю память РКБ ключа подписи.

Название команды: rkb_cmd_set_sign_key

Входные параметры: длина ключа подписи (sign_key_len), массив ключа подписи (sign_key)

Выходные данные: код ошибки с возможными значениями: SIGN_KEY_SET_ERROR, SUCCESS (см. Таблицу 2.1)

Выработка подписи

Команда подается прикладным ПО. По этой команде происходит выработка подписи на основе ключа подписи и массива со значением хэша

Название команды: rkb_cmd_sign

Входные параметры: длина массива с хэшем (hash_len), массив с хэшем (hash)

Выходные данные: код ошибки с возможными значениями: SIGN_ERR, SUCCESS (см. Таблицу 2.1), длина массива подписи (sign_len), массив подписи (sign)

Сброс ключа подписи

Команда подается прикладным ПО. По этой команде происходит удаление ключа подписи из внутренней памяти РКБ

Название команды: rkb_cmd_reset

Входные параметры: отсутствуют

Выходные данные: код ошибки с возможными значениями: RESET_ERR, SUCCESS (см. Таблицу 2.1)

Проверка подписи

Команда подается прикладным ПО. По этой команде происходит проверка подписи на базе ключа подписи, массива со значением хэша и массива со значением подписи

Название команды: rkb_cmd_verify

Входные параметры: длина массива со значением хэша (hash_len), массив со значением хэша (hash), длина массива со значением подписи (sign_len), массив со значением подписи (sign)

Выходные данные: код ошибки с возможными значениями: HASH_BUFFER_TOO_SMALL, SIGN_BUFFER_TOO_SMALL, VERIFY_ERR, SUCCESS (см. Таблицу 2.1)

Установка сертификата

Команда подается прикладным ПО. По этой команде происходит запись во внутреннюю память РКБ сертификата.

Название команды: rkb_cmd_set_cert (uint32_t, uint8_t *)

Входные параметры: длина массива с сертификатом (cert_len), массив с сертификатом (cert)

Выходные данные: код ошибки с возможными значениями: TOO_MUCH_DATA, SET_CERT_ERR, SUCCESS (см. Таблицу 2.1)

Получение сертификата

Команда подается прикладным ПО. По этой команде происходит чтение сертификата из внутренней памяти РКБ.

Название команды: rkb_cmd_get_cert (uint32_t *cert_len, uint8_t *cert)

Входные параметры: отсутствуют

Выходные данные: код ошибки с возможными значениями: GET_CERT_ERR, SET_CERT_ERR, SUCCESS (см. Таблицу 2.1), длина массива с сертификатом (cert_len), массив с сертификатом (cert)

2.3 Сообщения системному программисту

Перечень сообщений, получаемых системным программистом при работе с СПО «РКБ», отражен в Таблице 2.1.

Таблица 2.1

Сообщение	Код
#define SUCCESS	0x0
#define VERIFY_KEY_BUFFER_TOO_SMALL	0x201
#define SESSION_KEY_BUFFER_TOO_SMALL	0x202
#define CRYPTED_SESSION_KEY_BUFFER_TOO_SMALL	0x203
#define SIGN_KEY_GEN_ERR	0x204
#define VERIFY_KEY_GEN_ERR	0x205
#define SESSION_KEY_GEN_ERR	0x206
#define EXTRACT_SESSION_KEY_ERR	0x207
#define RND_GEN_ERR	0x208
#define SIGN_KEY_BUFFER_TOO_SMALL	0x209
#define SIGN_KEY_SET_ERROR	0x20A
#define HASH_BUFFER_TOO_SMALL	0x20B
#define SIGN_ERR	0x20C
#define RESET_ERR	0x20D
#define SIGN_BUFFER_TOO_SMALL	0x20E
#define VERIFY_ERR	0x20F

#define TOO_MUCH_DATA	0x210
#define SET_CERT_ERR	0x211
#define GET_CERT_ERR	0x212