МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт Факультет приборостроения, информационных технологий и электроники	Кафедра «Информационная безопасность систем и технологий»
Специальность 10.05.02 –	Информационная безопасность телекоммуникационных систем
Специализация –	Разработка защищенных телекоммуникационных систем

ДИПЛОМНЫЙ ПРОЕКТ

на тему

УСТРОЙСТВО ПЕРЕДАЧИ ЗАШИФРОВАННОЙ ИНФОРМАЦИИ СО ВСТРОЕННЫМ АППАРАТНО-ПРОГРАММНЫМ МЕХАНИЗМОМ ЗАЩИТЫ ПАМЯТИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К КЛЮЧЕВОЙ ИНФОРМАЦИИ

Студент _____ Хабибуллин Фатих Вялиевич

Руководитель ведущий инженер

АО «ПНИЭИ» _____ Дуплянский А.Л.

Нормоконтролёр _____ Иванов А.П.

Рецензент начальник НИС-103

АО «ПНИЭИ» _____ Акмаев А.Ж.

Работа допущена к защите (протокол заседания кафедры от _____ № ___)

Заведующий кафедрой _____ С.Л. Зефиров

Работа защищена с отметкой _____(протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК _____ А.Г. Фатеев

Пенза 2018

Оглавление

1 Анализ состояния рассматриваемой технической проблемы 7
1.1 Актуальность разработки устройства передачи зашифрованной информации 7
1.2 Требования, предъявляемые к разрабатываемым средствам криптографической защиты информации..... 8
1.3 Анализ угроз информационной безопасности 11
2 Обзор существующих аппаратно-программных механизмов защиты памяти от несанкционированного доступа 13
3 Аппаратно-программный механизм защиты памяти от несанкционированного доступа 16
3.1 Выбор и обоснование аппаратно-программного механизма защиты памяти от несанкционированного доступа 16
3.2 Описание встроенного механизма защиты памяти в МК STM32F4xx 17
4 Разработка схемы электрической структурной устройства передачи зашифрованной информации 26
5 Выбор и обоснование элементной базы для построения устройства передачи зашифрованной информации 28
6 Разработка схемы электрической принципиальной устройства передачи зашифрованной информации 31
7 Реализация и проверка работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа 35
7.1 Алгоритм изменения и применения уровней защиты от чтения 35
7.2 Способы применения механизма защиты памяти 36
7.3 Разработка схемы алгоритма аппаратно-программного механизма защиты памяти от несанкционированного доступа 38
7.4 Программное обеспечение, используемое для разработки программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа 39

7.5 Разработка программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа 42
8 Проверка работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа 45
8.1 Способы проверки работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа 45
8.2 Получение содержимого FLASH-памяти МК по отладочному интерфейсу SWD 46
8.3 Получение содержимого FLASH-памяти МК по отладочному интерфейсу JTAG 49
8.4 Получение содержимого FLASH-памяти МК через встроенный аппаратный загрузчик 51
8.5 Тестирование программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа 54
8.6 Результаты проверки работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа 58
9 Вредные психофизиологические факторы, влияющие на опорно-двигательный аппарат специалиста по защите информации. Методы и средства физической культуры, снижающие их воздействие 61
Заключение 64
Список используемых источников 65
Приложение А. Схема электрическая принципиальная аппаратного механизма защиты от НСД изделия-450 (устройство-прототип) 66
Приложение Б. Схема электрическая структурная устройства передачи зашифрованной информации 68
Приложение В. Схема электрическая принципиальная устройства передачи зашифрованной информации 70
Приложение Г. Перечень элементов 75
Приложение Д. Исходный код программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа ... 81

Приложение Е. Протокол проверки ВКР в системе «Антиплагиат» 88

Введение

Устройство передачи зашифрованной информации применяется в портативной аппаратуре связи [1] низового звена управления. В этих звеньях передается информация, не содержащая сведений, составляющих государственную тайну, тем не менее на основании передаваемой информации принимаются решения по управлению войсками и оружием. Поэтому возникает необходимость обеспечения её конфиденциальности, целостности (достоверности) и доступности. Основной функцией разрабатываемого устройства является обеспечение криптографической защиты информации передаваемой в канал связи.

Необходимость разработки новых устройств передачи зашифрованной информации, применяемых в портативной аппаратуре связи, обусловлена современными научно-техническими достижениями в области технологии производства элементной базы с точки зрения её миниатюризации, энергоэффективности, производительности и функциональности, а также с быстрым устареванием и прекращением поддержки производителем вычислительных устройств, которые всего несколько лет назад считались лучшими по быстродействию.

Современные устройства передачи зашифрованной информации должны иметь малое энергопотребление, быть компактными, обеспечивать высокую скорость шифрования, и быть защищенным от несанкционированного доступа к ключевой информации.

Целью выпускной квалификационной работы является разработка схем электрической структурной и электрической принципиальной устройства передачи зашифрованной информации со встроенным аппаратно-программным механизмом защиты памяти от несанкционированного доступа к ключевой информации.

Аппаратно-программный механизм защиты памяти должен обеспечивать защиту от получения доступа к ключевой информации в памяти устройства передачи зашифрованной информации.

Задачами выпускной квалификационной работы являются:

- анализ состояния рассматриваемой технической проблемы;

- обзор существующих аппаратно-программных механизмов защиты памяти от несанкционированного доступа;

- выбор и обоснование аппаратно-программного механизма защиты памяти от несанкционированного доступа;

- разработка схемы электрической структурной устройства передачи зашифрованной информации;

- выбор и обоснование элементной базы для построения устройства передачи зашифрованной информации;

- разработка схемы электрической принципиальной устройства передачи зашифрованной информации;

- реализация и проверка работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа;

- исследование вредных психофизиологических факторов, влияющих на опорно-двигательный аппарат специалиста по защите информации, а также методов и средств физической культуры, снижающих их воздействие.

1 Анализ состояния рассматриваемой технической проблемы

1.1 Актуальность разработки устройства передачи зашифрованной информации

Целью данной работы является разработка электрических схем (структурной и принципиальной) устройства передачи зашифрованной информации (далее - УПЗИ). Предполагается, что разрабатываемое УПЗИ будет применяться в портативной аппаратуре связи [1] в звеньях управления от роты до отдельного военнослужащего. В этих звеньях передается информация, не содержащая сведений, составляющих государственную тайну. Актуальность этой информации не является долговременной и составляет от 12 часов до 1-2 суток. Тем не менее, на основании передаваемой информации принимаются решения по управлению войсками и оружием. Поэтому возникает необходимость обеспечения её конфиденциальности, целостности (достоверности) и доступности.

Портативной аппаратурой связи согласно [1] именуется техническое средство радиосвязи и/или вспомогательное оборудование, носимые пользователем (располагаемые на теле пользователя) при их применении, получающие электропитание от собственной встроенной батареи.

Основная функция разрабатываемого УПЗИ - обеспечение криптографической защиты информации передаваемой в канал связи.

Разработка новых радиоэлектронных изделий ведется постоянно, ввиду растущих возможностей современной элементной базы, требований заказчика и конъюнктуры рынка.

К средствам криптографической защиты информации, применяемым в портативной аппаратуре связи, предъявляются определенные требования. Такие средства должны иметь малое энергопотребление, быть компактными, обеспечивать достаточно высокую скорость шифрования, быть защищенным от получения доступа к ключевой информации.

Под средствами криптографической защиты информации (далее – СКЗИ) понимают шифровальное (криптографическое) средство, предназначенное для защиты информации, не содержащих сведений, составляющих государственную тайну [2].

Согласно требованиям, предъявляемым к разрабатываемым средствам криптографической защиты информации [2], в качестве составной части СКЗИ должна быть реализована система защиты от несанкционированного доступа (НСД) к используемой СКЗИ ключевой и криптографически опасной информации. Для выполнения данного требования в разрабатываемом УПЗИ применяется встроенный аппаратно-программный механизм защиты памяти от НСД к ключевой информации (КИ). Кроме того, применение встроенного аппаратно-программного механизма защиты памяти от НСД к КИ преследует цель минимизации стоимости, габаритов, энергопотребления и повышения надежности УПЗИ путем использования встроенных функций современных вычислительных устройств.

1.2 Требования, предъявляемые к разрабатываемым средствам криптографической защиты информации

Согласно Положению ПКЗ-2005 [2] к средствам криптографической защиты информации относятся:

а) средства шифрования - аппаратные, программные и аппаратно-программные средства, системы и комплексы, реализующие алгоритмы криптографического преобразования информации и предназначенные для защиты информации при передаче по каналам связи и (или) для защиты информации от несанкционированного доступа при ее обработке и хранении;

б) средства имитозащиты - аппаратные, программные и аппаратно-программные средства, системы и комплексы, реализующие алгоритмы криптографического преобразования информации и предназначенные для защиты от навязывания ложной информации;

в) средства электронной цифровой подписи;

г) средства кодирования - средства, реализующие алгоритмы криптографического преобразования информации с выполнением части преобразования путем ручных операций или с использованием автоматизированных средств на основе таких операций;

д) средства изготовления ключевых документов (независимо от вида носителя ключевой информации);

е) ключевые документы (независимо от вида носителя ключевой информации).

СКЗИ могут быть реализованы программным, программно-аппаратным или аппаратным способом. У каждой реализации свои особенности, достоинства и недостатки. Так программная и программно-аппаратная реализация СКЗИ существенно проигрывают по скорости криптографического преобразования и энергопотреблению аппаратной реализации. В то же время аппаратные средства менее гибкие, в том смысле, что программные и программно-аппаратные СКЗИ позволяют менять алгоритмы криптографического преобразования.

Программно-аппаратные и аппаратные СКЗИ производятся в различных исполнениях для использования в различной аппаратуре связи. Для портативной аппаратуры связи изготавливаются малогабаритные встраиваемые модули в виде плат с разъёмом для подключения в каналообразующую аппаратуру, а также в виде отдельных блоков с внешними интерфейсами для подключения к радиостанциям. Для установки в подвижные и стационарные объекты изготавливаются блоки с соединителями для подключения к оконечному оборудованию данных, каналообразующей аппаратуре и аппаратуре, предназначенной для ввода ключевой информации.

В зависимости от класса СКЗИ и состава криптографических функций, которые должны быть реализованы в СКЗИ, определяются требования предъявляемые к СКЗИ. Для выполнения требований, предъявляемых к СКЗИ, применяются инженерно-криптографические механизмы.

Инженерно-криптографический механизм – алгоритмическая или техническая мера, реализуемая в СКЗИ для защиты информации от атак, возникающих вследствие неисправностей или сбоев аппаратных средств СКЗИ и аппаратных средств среды функционирования.

Использование инженерно-криптографических механизмов в СКЗИ основывается на следующих принципах:

- защита от непреднамеренных действий пользователя, не предусмотренных правилами пользования СКЗИ и приводящие к возможности осуществления успешных атак на СКЗИ;

- система защиты от несанкционированного доступа к используемой СКЗИ ключевой и криптографически опасной информации и защищаемой СКЗИ информации;

- механизм блокировки работы СКЗИ. Блокировка работы СКЗИ должна выполняться в случаях:

  - диагностики неисправности СКЗИ;

  - достижение предельных значений технических характеристик СКЗИ;

  - истечение срока действия ключей;

  - нарушение целостности программного обеспечения СКЗИ;

- механизм контроля работоспособности мер криптографической и инженерно-криптографической защиты;

- механизм контроля целостности СКЗИ должен обеспечивать:

  - контроль целостности с использованием криптографических механизмов;

  - контроль целостности должен включать средства контроля собственной корректной работы;

  - контроль целостности должен проводиться до начала обработки информации, безопасность которой должна обеспечивать СКЗИ;

  - реализация периодического контроля целостности;

- в случае обнаружения нарушения целостности должна осуществляться блокировка СКЗИ;

- механизм, обеспечивающий контроль целостности ключевой информации;

- компоненты, обеспечивающие очистку областей памяти, используемых СКЗИ для хранения защищаемой, ключевой и криптографически опасной информации, при освобождении и/или перераспределении областей памяти, путем записи в области памяти случайной информации, вырабатываемой датчиком случайных чисел;

- механизм аварийного (экстренного) уничтожения ключевой и криптографически опасной информации;

- механизм регистрации событий;

- механизм контроля целостности журналов регистрации событий.

Предполагается, что в разрабатываемом УПЗИ будут реализованы следующие криптографические функции:

- функция выработки псевдослучайной последовательности;

- функции зашифрования/расшифрования;

- функция имитозащиты (функция контроля целостности данных);

- функция аутентификации субъектов доступа.

1.3 Анализ угроз информационной безопасности

Подходя к решению вопросов обеспечения информационной безопасности, в первую очередь, необходимо рассмотреть аспекты, касающиеся видов предполагаемых угроз и ценности передаваемой информации. От этих параметров зависит требования к УПЗИ, а также стоимость разрабатываемого устройства.

Так как разрабатываемое УПЗИ предполагается для использования военнослужащими в нижнем тактическом звене управления, то в данном случае заданы следующие параметры:

- информация не составляет государственную тайну, но представляет большую ценность и вызывает интерес потенциального нарушителя (противника) с точки зрения ее чтения и искажения;

- виды информации, подлежащей защите: открытая информация (речевая информация или потоковые данные), ключевая информация, используемая для шифрования;

- алгоритм криптографического преобразования, применяемый в УПЗИ известен противнику;

- режим работы – зашифрование, расшифрование, имитозащита;

- актуальность информации не является долговременной и составляет от 12 часов до 1-2 суток;

- вероятность использования методов съема и анализа ПЭМИ для перехвата информации очень мала;

- вероятность дешифрования информации противником и попыток обнаружить побочные наводки открытой и ключевой информации в канале связи достаточно велика;

- вероятность захвата аппаратуры и компрометации ключей высока.

Исходя и сказанного выше, а также исходя из требований ТЗ на ВКР, при создании УПЗИ особое внимание уделено системе защиты от НСД.

2 Обзор существующих аппаратно-программных механизмов защиты памяти от несанкционированного доступа

Механизмы защиты информации от несанкционированного доступа предназначены для того, чтобы защищаемая информация не попала в руки противника или субъект, имеющий разрешение на доступ к части защищаемой информации, мог получать доступ только к той информации, которой он имеет разрешение на доступ.

Механизмы защиты информации (ЗИ) от несанкционированного доступа (НСД) могут быть разделены по способу реализации:

- программные;
- аппаратно-программные (программно-аппаратные);
- аппаратные.

Основным достоинством программных механизмы ЗИ от НСД является их дешевизна и гибкость. Наряду с этими существенными достоинствами у программных механизмов ЗИ от НСД имеются и существенные недостатки, заключенные, собственно, в их наибольшем достоинстве – возможности легкой модификации. Программа, реализующая механизм защиты информации от НСД, может быть модифицирована противником. Для устранения угрозы модификации следует осуществить контроль целостности этой программы, однако это возможно только с помощью другой программы. Проверка целостности одних программ при помощи других не является надежной. Необходимо четко представлять, каким образом обеспечивается целостность собственно программы проверки целостности. Если обе программы находятся на одних и тех же носителях, доверять результатам такой проверки нельзя. Исходя из выше перечисленного, к программным механизмам ЗИ от НСД следует относиться с особой осторожностью [3].

С учетом того, что программные механизмы ЗИ от НСД весьма часто реализуются не специалистами, а программистами, не имеющими специальной

подготовки, можно отметить относительно невысокую надежность этих реализаций.

Аппаратные механизмы ЗИ от НСД отличаются высокой надежностью и отказоустойчивостью по сравнению с программными механизмами. Однако аппаратным механизмам ЗИ от НСД также свойственна сложность разработки и реализации, а также высокая конечная стоимость такой реализации.

Аппаратно-программные механизмы ЗИ от НСД сочетают в себе положительные качества как программных – простота реализации, так и аппаратных механизмов – высокая надежность. Но нельзя также забывать о недостатках, которые также как и преимущества, присущи программной и аппаратной реализации.

Приведенная в приложении А электрическая принципиальная схема является одним из примеров аппаратной реализации механизма ЗИ от НСД. Данный аппаратный механизм ЗИ от НСД реализован в устройстве-прототипе, изделии 450.

Аппаратный механизм сброса ключевой информации питается от встроенной в аппаратуру связи батареи, что обеспечивает независимость от входного питающего напряжения.

Линия NSD_IN подключается к кнопке на корпусе аппаратуры связи для обнаружения вскрытия корпуса или для запуска процедуры стирания ключей из памяти в случае захвата аппаратуры противником.

При замыкании линии NSD_IN на 0 В происходит падение напряжения на резисторе R31, что открывает P-канальный полевой транзистор V2. Напряжение от батареи VBAT_IN поступает на вход ON силового ключа D10 и запускает его. Напряжение от батареи через микросхему D10 поступает на DC-DC понижающий импульсный преобразователь D14, и пониженное до уровня плюс 1,85 В напряжение поступает на микроконтроллер и он запускает начальную загрузку. Далее на затвор транзистора V4 поступает напряжение от батареи, тем самым открывая его. Через открытый транзистор V4 разряжается на землю конденсатор C62, при этом формируется сигнал низкого уровня, при

обнаружении на линии INT_KEY данного сигнала в микроконтроллере запускается процедура стирания ключей из памяти.

Формирование сигнала сброса на линии RESET, при замыкании линии NSD_IN на землю, также происходит через цепь V2-V3. Сигнал высокого уровня от батареи через резистор R31 подается на вход MR микросхемы D9. При этом низкий уровень сигнала на выходе RESET микросхемы D9 открывает транзистор V3, а затем открывается и транзистор V4, что также обеспечивает формирование сигнала сброса.

Контроль за уровнем заряда батареи выполняет микросхема D9. В случае падения напряжения на батарее ниже установленного уровня, на выходе RESET микросхемы D9 формируется сигнал низкого уровня, который открывает транзистор V3. Через открытый транзистор V3 напряжение от батареи поступает на D10 и V4, что также запускает процедуру стирания ключей.

Также в схеме реализована процедура контроля работоспособности механизма экстренного стирания ключей. При инициализации данной процедуры, микроконтроллер подает на линию NSD_CTRL сигнал высокого уровня, который открывает транзистор V1 и запускаются описанные выше процессы. В случае исправности цепи микроконтроллер принимает сигнал сброса, но ключи не стираются.

3 Аппаратно-программный механизм защиты памяти от несанкционированного доступа

3.1 Выбор и обоснование аппаратно-программного механизма защиты памяти от несанкционированного доступа

При выборе аппаратно-программного механизма защиты памяти от несанкционированного доступа в первую очередь необходимо руководствоваться требованиями, приведенными в техническом задании на выпускную квалификационную работу, а именно:

- аппаратно-программный механизм защиты памяти от несанкционированного доступа должен быть реализован с использованием встроенных средств микроконтроллера;
- аппаратная платформа для разработки УПЗИ – микроконтроллер семейства STM32F4.

Кроме того следует учитывать, что разрабатываемое устройство передачи зашифрованной информации предполагается использовать в носимой аппаратуре связи. Это означает, что основными требованиями к УПЗИ являются минимизация габаритов и энергопотребления разрабатываемого УПЗИ и, что также важно, стоимости изготовления.

Из приведенных требований следует, что аппаратно-программный механизм защиты памяти от несанкционированного доступа и УПЗИ в целом должен состоять из минимального количества электронных компонентов, а большую часть функций УПЗИ должен выполнять микроконтроллер.

Следуя предъявленным требованиям, при изучении документации к микроконтроллерам семейства STM32F4 было выяснено, что микроконтроллеры данного семейства имеют встроенный механизм защиты памяти от чтения, который реализован в так называемом интерфейсе FLASH-памяти.

Выбор в качестве аппаратно-программного механизма защиты памяти от несанкционированного доступа механизма защиты от чтения, встроенного в

микроконтроллеры семейства STM32F4, основан на том, что данный механизм удовлетворяет всем требованиям, предъявленным в техническом задании на ВКР. Также реализация данного механизма не требует применения дополнительных аппаратных компонентов. В сочетании с относительной, по сравнению с аппаратной, простотой реализации данный механизм является оптимальным вариантом.

3.2 Описание встроенного механизма защиты памяти в МК STM32F4xx

В микроконтроллерах серии STM32F4, как говорилось в предыдущем параграфе, реализована защита встроенной FLASH-памяти от чтения и записи. Функции защиты памяти от чтения и записи реализованы в интерфейсе FLASH-памяти. На рисунке 1 показаны связи интерфейса FLASH-памяти внутри архитектуры системы микроконтроллера серии STM32F4. Кроме механизма защиты памяти от чтения интерфейс FLASH-памяти также выполняет функции чтения, программирования и стирания FLASH-памяти.

Интерфейс FLASH-памяти управляет доступом ЦПУ по высокоскоростным шинам команд (I-Code) и данных (D-Code) к FLASH-памяти. Он также реализует операции записи и стирания FLASH-памяти и механизмы защиты от чтения и записи.

Встроенная FLASH -память микроконтроллеров семейства STM32F40x и STM32F41x организована следующим образом:

- основной блок памяти, разделенный на 4 сектора по 16 Кбайт, 1 сектор на 64 Кбайт и 7 секторов по 128 Кбайт;

- системная память, из которой микроконтроллер загружается в режиме загрузки из системной памяти;

- 512 однократно программируемых байтов (OTP – One Time Programmable) для постоянного хранения конфигурационных данных МК;

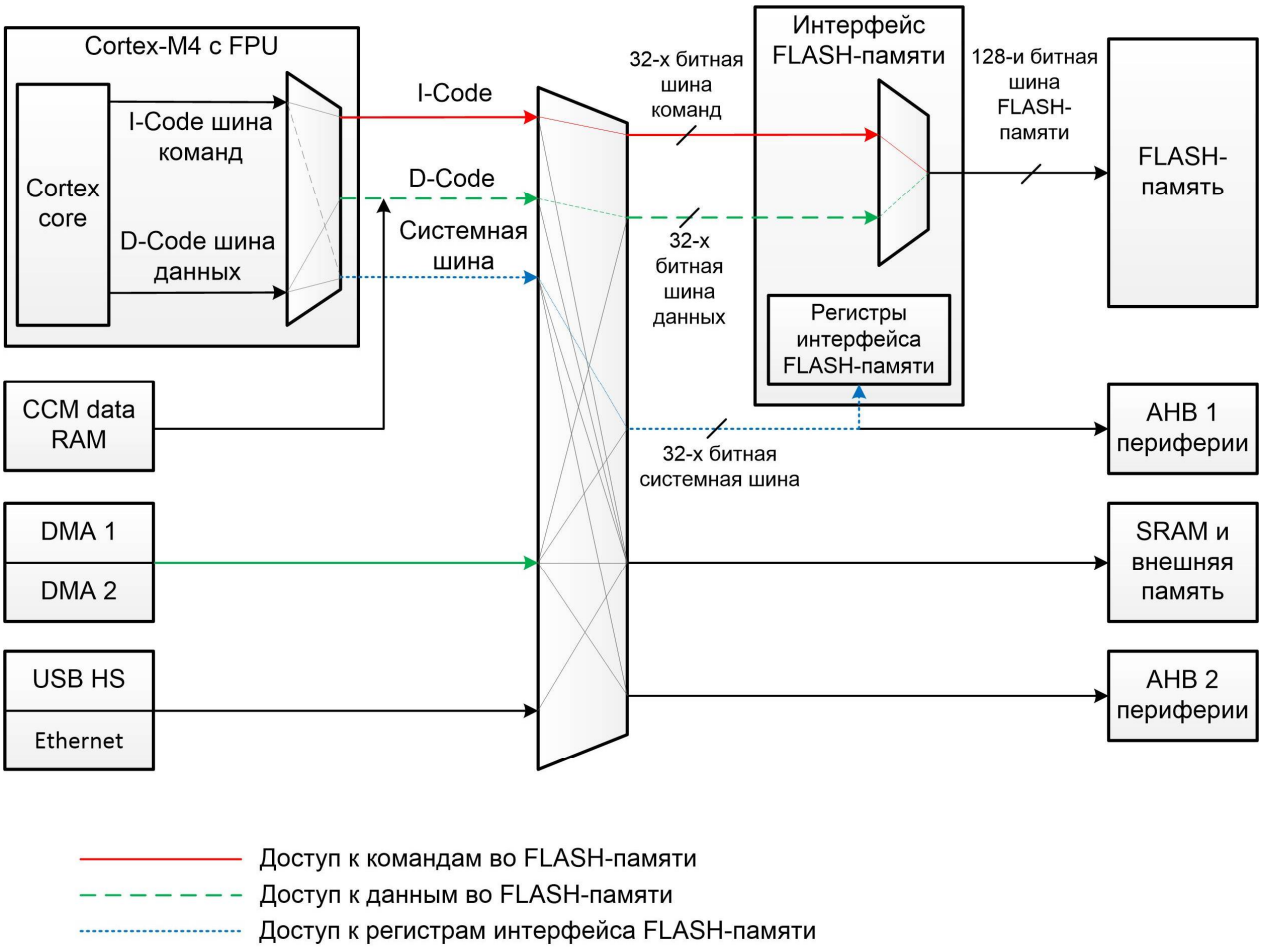


Рисунок 1 – Связи интерфейса FLASH-памяти внутри архитектуры системы микроконтроллера

- байты опций для настройки:

- защиты от чтения и записи;

- уровня BOR (Brown Out Reset) - перезагрузка микроконтроллера, если питающее напряжение ниже или выше установленного;

- запуска сторожевого таймера - программно или аппаратно, то есть автоматически при подаче питания на МК;

- перезагрузки микроконтроллера при входе в режим остановки (Stop mode);

- перезагрузки микроконтроллера при входе в режим ожидания (Standby mode).

На рисунке 2 приведена карта памяти МК серии STM32F4.

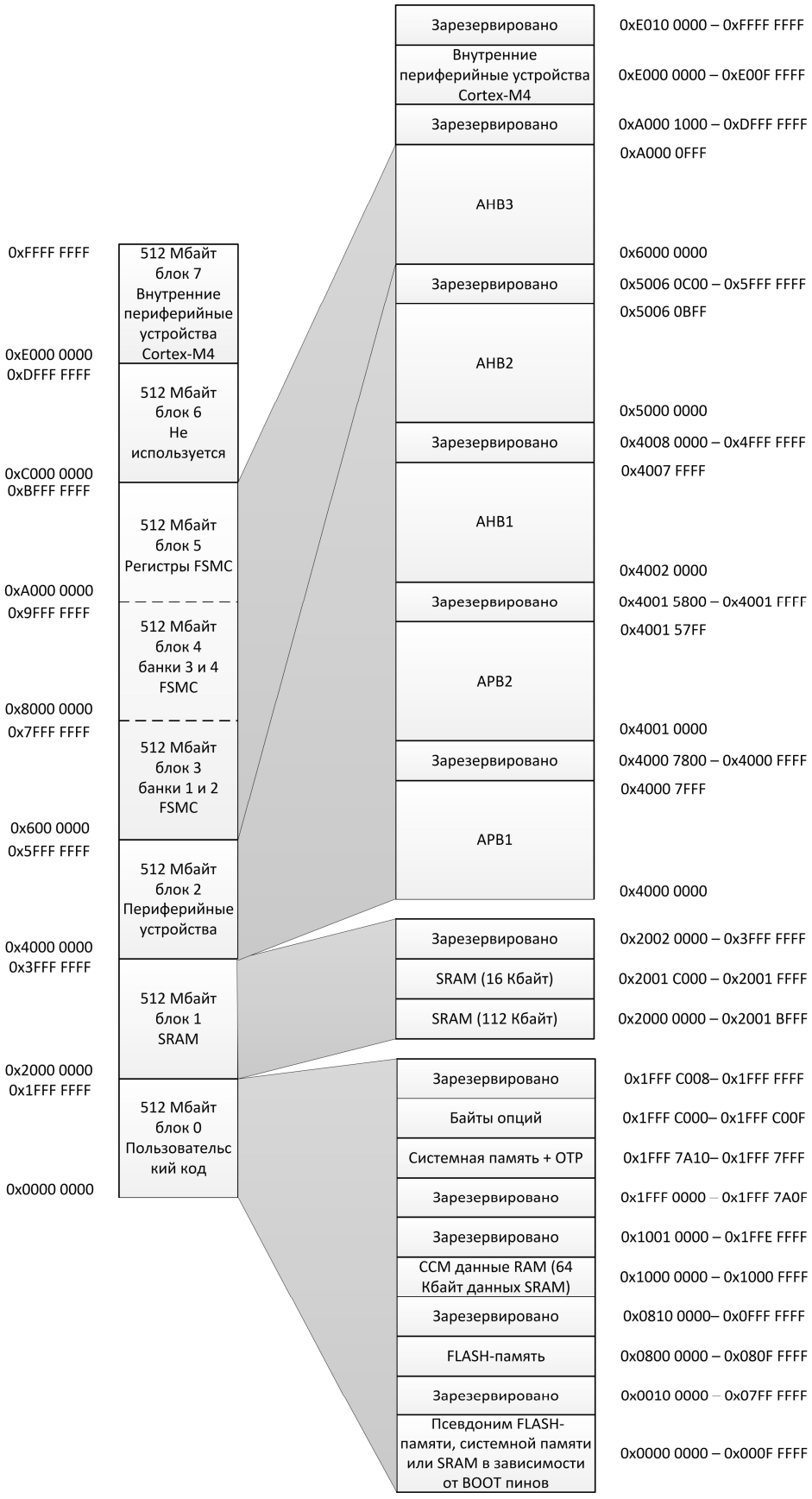


Рисунок 2 – Карта памяти МК серии STM32F4

В таблице 1 представлены регистры интерфейса FLASH-памяти и их значения по умолчанию.

Таблица 1 – Регистры FLASH-интерфейса и значения по умолчанию

Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_ACR	Зарезервировано																			DCRST	ICRST	DCEN	ICEN	PRFTEN	Зарезервировано					LATENCY [2:0]		
Знач. по умол.																				0	0	0	0	0						0	0	0
FLASH_KEYR	KEY[31:16]																KEY[15:0]															
Знач. по умол.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FLASH_OPTKEYR	OPTKEYR[31:16]																OPTKEYR[15:0]															
Знач. по умол.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FLASH_SR	Зарезервировано															BSY	Зарезервировано								PGSERR	PGPERR	PGAERR	WRPERR	Зарезервировано		OPERR	EOP
Знач. по умол.																0									0	0	0	0			0	0
FLASH_CR	LOCK	Зарезервировано							EOPIE	Зарезервировано						STRT	Зарезервировано						PSIZE [1:0]		Зарезервировано	SNB [3:0]				MER	SER	PG	
Знач. по умол.	1								0							0							0	0		0	0	0	0		0	0
FLASH_OPTCR	Зарезервировано			nWRP[11:0]													RDP[7:0]								nRST_STDBY	nRST_STOP	WDG_SW	Зарезервировано		BOR_LEV[1:0]		OPTSTRT	OPTLOCK
Знач. по умол.				1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1			1	1	0	1

Механизм защиты памяти от чтения и записи в МК серии STM32F4 настраивается путем изменения байтов опций, а для управления байтами опций используется регистр FLASH_OPTCR управления опциями FLASH-памяти. Регистр управления опциями FLASH-памяти (FLASH_OPTCR) приведен в таблице 2. В таблице указаны атрибуты доступа к битам регистра. Значение rw – чтение и запись бита, rs – чтение и установка бита; установка в «0» не влияет на значение бита.

Таблица 2 – Регистр управления опциями FLASH-памяти FLASH_OPTCR

Биты	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Поле	Зарезервировано				nWRP [11:0]											
Атрибуты					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Биты	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Поле	RDP [7:0]								nRST_STDBY	nRST_STOP	WDG_SW	Зарезервировано	BOR_LEV		OPTSTRT	OPTLOCK
Атрибуты	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rs	rs

Биты с 27 по 16 – nWRP [11:0]. Эти биты содержат значение байтов опций защиты от записи. При установке «1» защита от записи активна в выбранном секторе.

Биты от 15 до 8 – RDP [7:0]. Защита от чтения. Эти биты содержат значение байтов опций защиты от чтения. Значения:

- 0xAA – уровень 0, защита памяти от чтения не активна;

- 0xCC – уровень 2, защита памяти от чтения МК активна;

- другие значения – уровень 1, защита памяти от чтения памяти активна.

Биты от 7 до 5 – USER [2:0]. Пользовательские байты опций. Два бита nRST_STDBY и nRST_STOP управляют процессом входа МК в режимы STANDBY и STOP. Можно задать, чтобы при входе в тот и другой режим генерировался сброс. Третий бит конфигурирует процесс активации независимого сторожевого таймера. Сторожевой таймер имеет аппаратный и

программный режимы запуска. В аппаратном режиме сторожевой таймер включается сразу после сброса микроконтроллера. В программном режиме таймер запускается под управлением программного обеспечения МК.

Биты 3 и 2 служат для выбора уровня BOR.

Бит 1 – OPTSTRT. Этот бит запускает операцию программирования байтов опций при установке в «1». Он устанавливается только программно и очищается, когда бит BSY регистра FLASH_SR статуса FLASH-памяти очищается.

Бит 0 – OPTLOCK. Когда бит установлен в «1», он указывает, что регистр FLASH_OPTCR заблокирован. Этот бит очищается аппаратно после обнаружения последовательности разблокировки.

Механизм защиты памяти МК от чтения, называемый RDP (Read protection), предотвращает доступ к FLASH-памяти через отладочные интерфейсы (JTAG и SWD) и системный загрузчик и обеспечивает защиту памяти от чтения пользовательской области во FLASH-памяти.

Механизм защиты памяти от чтения является глобальной защитой от чтения FLASH-памяти, позволяющей защитить код прошивки от копирования, обратного проектирования (reverse engineering), сброса с помощью инструментов отладки или других методов вмешательства. Механизм защиты должен быть установлен пользователем после того, как прошивка будет загружена во встроенную FLASH-память.

Определены три уровня механизма защиты от чтения:

- уровень 0;
- уровень 1;
- уровень 2.

Уровень 0 - защита памяти от чтения не активна. Когда выбран уровень 0 все операции чтения/записи (если не установлена защита от записи) из/во FLASH-память или резервный SRAM доступны во всех конфигурациях загрузки (загрузка из FLASH-памяти, режим отладки или загрузка из ОЗУ).

Уровень 0 устанавливается путем записи 0xAA в байт опций защиты памяти от чтения (RDP).

Уровень 1 - защита памяти от чтения включена. Уровень 1 защиты памяти от чтения активируется записью любого значения (кроме 0xAA и 0xCC, они используются для установки уровня 0 и уровня 2, соответственно) в байт опций RDP. Когда установлен уровень 1 защиты от чтения:

- нет доступа (чтение, очистка, запись) во FLASH-память или резервный SRAM пока выполняется отладка или выполнена загрузка из RAM или из системной памяти (встроенный загрузчик). Любой доступ к FLASH-памяти и резервной SRAM вызывает системную ошибку, которая блокирует выполнение всего кода до следующего сброса при включении МК;

- при загрузке из FLASH-памяти разрешаются обращения (чтение, очистка, запись) во FLASH-память и резервный SRAM из программы пользователя.

Отключение механизма защиты, то есть переход к уровню 0 защиты от чтения, приводит к полному стиранию FLASH-памяти и резервного SRAM, кроме области OTP и настроек байт опций. Сброс настроек механизма защиты происходит только после окончания стирания FLASH-памяти. Стирание очищает только область памяти пользовательского кода.

Стирание выполняется только в случае перехода от уровня 1 на уровень 0. Когда уровень защиты повышается, стирание не выполняется.

Уровень 2 - отладка и защита памяти микроконтроллера от чтения отключена. Уровень 2 защиты от чтения активируется записью 0xCC в байт опций RDP. Когда установлен уровень 2 защиты от чтения:

- все защитные механизмы уровня 1 включены;

- загрузка из RAM или из системной памяти (встроенный загрузчик) не доступна;

- JTAG, SWV (single-wire viewer), ETM, и периферийное сканирование выключены;

- пользовательские байты опций больше нельзя изменить;

- в случае загрузки из FLASH-памяти, доступ (чтение, очистка и запись) во FLASH-память и резервный SRAM из программы пользователя разрешен.

Применение уровень 2 защиты памяти от чтения операция необратимая. Если установлен уровень 2, то уровень защиты не может быть понижен до уровня 0 или уровня 1.

При активном уровне 2 порт JTAG отключается перманентно (действует как плавкий предохранитель).

На рисунке 3 показан переход от одного уровня защиты памяти от чтения к другому.

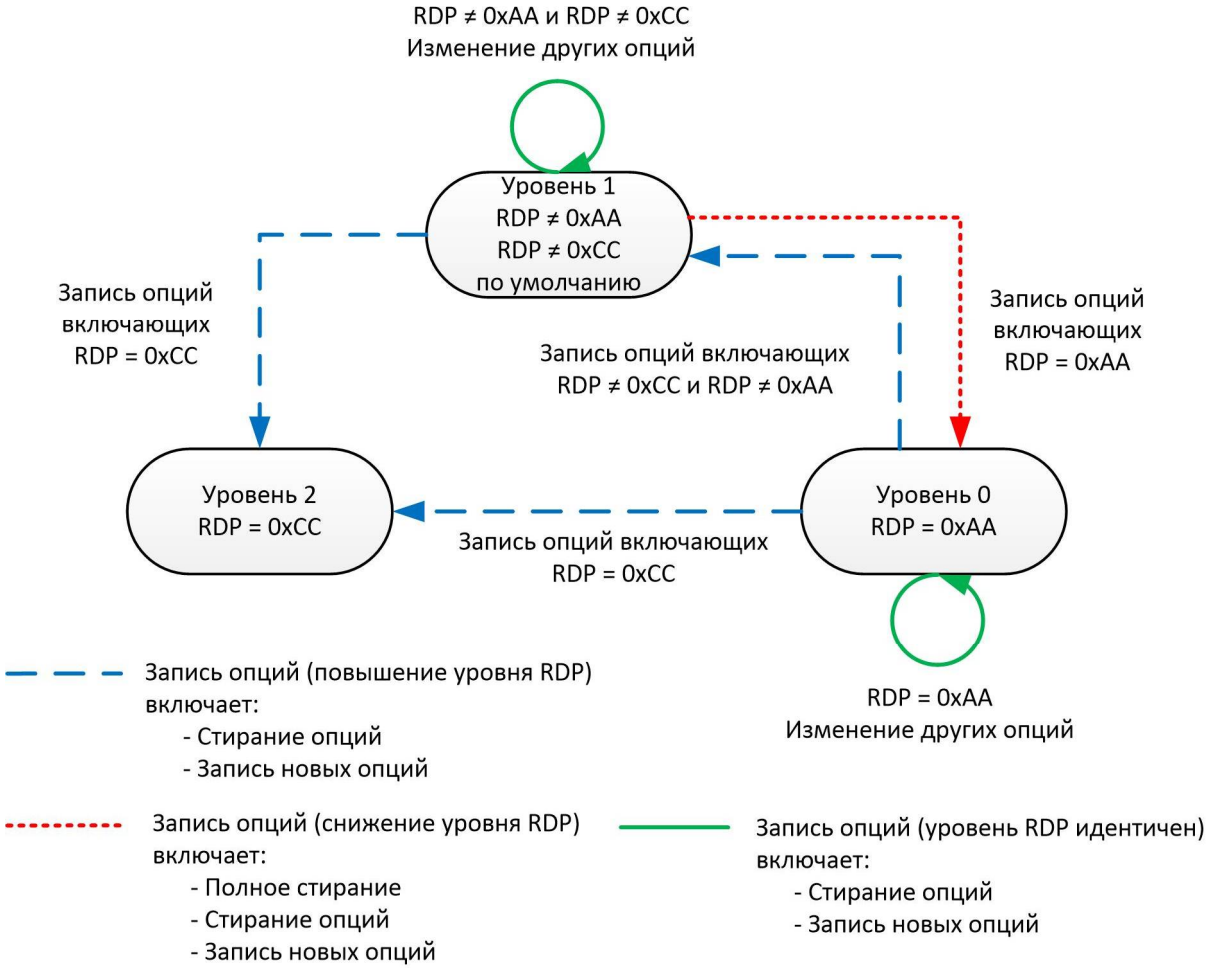


Рисунок 3– Изменение уровня защиты памяти от чтения

Байты опции вступают в силу немедленно, когда они запрограммированы. Это означает, что при включении защиты от считывания МК через байты опций программатор немедленно потеряет доступ к FLASH-

памяти без какой-либо возможности проверить окончание загрузки прошивки в МК.

Механизм защиты FLASH-памяти от чтения, настраивается в байтах опций FLASH-памяти, и настройки механизма считываются до старта встроенного загрузчика. При попытке чтения данных из FLASH-памяти с включённой защитой генерируется ошибка шины. Отключение механизма защиты приводит к полному стиранию FLASH-памяти, кроме области OTP и настроек байт опций. Сброс настроек механизма защиты происходит только после завершения цикла стирания FLASH-памяти.

В таблице 3 представлена возможность доступа к различным областям памяти МК при различных уровнях защиты памяти от чтения и вариантов загрузки.

Таблица 3 – Доступ к памяти при различных уровнях защиты от чтения

<table>
<tr><th rowspan="3">Область памяти</th><th rowspan="3">Уровень защиты</th><th colspan="3">Отладка, загрузка из ОЗУ или из системной памяти (встроенный загрузчик)</th><th colspan="3">Загрузка из FLASH-памяти</th></tr>
<tr><th>Чтение</th><th>Запись</th><th>Стирание</th><th>Чтение</th><th>Запись</th><th>Стирание</th></tr>
<tr><td rowspan="2">Основная FLASH-память и резервный SRAM</td><td>Уровень 1</td><td colspan="2">НЕТ</td><td>НЕТ*</td><td colspan="3">ДА</td></tr>
<tr><td>Уровень 2</td><td colspan="3">НЕТ</td><td colspan="3">ДА</td></tr>
<tr><td rowspan="2">Байты опций</td><td>Уровень 1</td><td colspan="3">ДА</td><td colspan="3">ДА</td></tr>
<tr><td>Уровень 2</td><td colspan="3">НЕТ</td><td colspan="3">НЕТ</td></tr>
<tr><td rowspan="2">OTP</td><td>Уровень 1</td><td colspan="2">НЕТ</td><td>Н/П</td><td colspan="2">ДА</td><td>Н/П</td></tr>
<tr><td>Уровень 2</td><td colspan="2">НЕТ</td><td>Н/П</td><td colspan="2">ДА</td><td>Н/П</td></tr>
</table>

* - FLASH-память и резервный SRAM удаляются только при изменении уровня защиты памяти от чтения с уровня 1 на 0. Область OTP остается неизменной.

4 Разработка схемы электрической структурной устройства передачи зашифрованной информации

Разработанная электрическая структурная схема приведена в приложении Б.

Согласно техническому заданию на ВКР аппаратной платформой для разработки УПЗИ является микроконтроллер семейства STM32F4. На данный микроконтроллер, D1 на структурной схеме, возложены основные функции УПЗИ, а именно:

- зашифрование и расшифрование передаваемых и принимаемых данных;
- управление ключевой информацией, её хранение во встроенной FLASH-памяти и уничтожение;
- прием и передача открытых данных на оконечное оборудование данных;
- защита открытой и ключевой информации от несанкционированного доступа;
- контроль над правильным функционированием всех элементов УПЗИ.

Открытая информация поступает с ООД через UART в микроконтроллер, где осуществляться зашифрование информации с последующей передачей её в ПЛИС, а затем в канал. Зашифрованная канальная информация поступает через ПЛИС в микроконтроллер для расшифрования, после чего передаётся через UART в ООД.

Программируемая логическая интегральная схема, элемент D2 на структурной схеме, предназначена для защиты от активации закладок в интегральных микросхемах иностранного производства. Защита заключается в формировании микроконтроллером ПСП, передаваемой в ПЛИС по интерфейсу SPI, и «подмешиваемой» к принимаемому канальному сигналу с последующим её «снятием» (в микроконтроллере).

К микроконтроллеру подключена микросхема D3, реализующая функции интерфейса RS232. Данный интерфейс служит для ввода ключевой информации

в устройство передачи зашифрованной информации, а также для целей диагностики и обновления ПО с помощью персонального компьютера.

Генератор G1 предназначен для формирования сигнала опорной тактовой частоты, обеспечивающего синхронную работу всех схем устройства.

Источник вторичного электропитания (ИВЭП), реализованный на микросхемах D4-D6, предназначен для формирования напряжений питания всех элементов устройства. В состав ИВЭП входит супервизор D6, предназначенный для обеспечения контроля над уровнем напряжения питания устройства и формирования сигнала «СБРОС» при низких напряжениях вторичных источников питания. ИВЭП в части питания микроконтроллера и ПЛИС реализован на микросхеме D5.

Все цепи интерфейсов сопряжения, питания и управления выведены на соединительный разъём X1.

5 Выбор и обоснование элементной базы для построения устройства передачи зашифрованной информации

Исходя из требований технического задания на ВКР, выбор микроконтроллера (далее – МК) ограничивается семейством STM32F4 фирмы STMicroelectronics. В данной серии существует множество микроконтроллеров с различными возможностями и отличиями в периферии. Выбор был сделан в пользу микроконтроллера STM32F415OGY6. Выбор основан на наличии необходимых интерфейсов, высокой производительности, наличии встроенных механизмов защиты памяти от чтения и записи в совокупности с низкой ценой.

Микроконтроллер, обозначен как элемент D1 на структурной схеме в приложении Б, STM32F415OGY6 фирмы STMicroelectronics, ядро ARM Cortex-M4 разрядностью 32 бита работающий на частоте до 168 МГц. Данный МК позиционируется как контроллер с RISC архитектурой, включающий блок вычислений с плавающей точкой и поддержкой DSP-команд цифровой обработки данных.

Ядро ARM Cortex-M4 позволяет производить вычисления с плавающей точкой. Данный микроконтроллер относится к семейству процессоров с RISC архитектурой с низким энергопотреблением. Оптимальное соотношение низкого энергопотребления (до 0,150 мВт), высокой производительности (до 210 MIPS) и низкой цены (от 7$ до 17$) делает процессоры данного семейства идеальной платформой для ресурсоемких приложений обработки голоса и данных в портативной и мобильной аппаратуре (сотовые телефоны, УКВ-радиостанции и т.п.).

МК STM32F415OGY6 имеет 192 Кбайт встроенного ОЗУ, 1 Мбайт памяти программ, а также богатый набор внутренней периферии:

- два интерфейса USB OTG (один из них поддерживает USB HS);
- 15 интерфейсов обмена данными:
  - 4 USART и 2 UART интерфейса, два из них со скоростью до 10,5 Мбит/с, а остальные со скоростью до 5,25 Мбит/с;

- три SPI интерфейса со скоростью до 42 Мбит/с;

- три I²C;

- два CAN и SDIO;

- два 12-битных ЦАП и три 12-битных АЦП;

- семнадцать 16- и 32-битных таймеров работающих на частоте до 168 МГц, два сторожевых таймера;

- аналоговый аппаратный генератор случайной последовательности (RNG);

- интегрированный криптографический процессор, обеспечивающий аппаратное шифрование по алгоритмам AES 128, 192, 256, Triple DES, и вычисления хеш сумм MD5 и SHA-1;

- 16-канальный контроллер DMA;

- 140 выводов общего назначения (GPIO);

- часы реального времени (RTC).

При выборе ПЛИС основным ограничением ставилось применение CPLD ПЛИС. Выбор CPLD ПЛИС основан на наличии в них встроенной энергонезависимой памяти, которая хранит прошивку при отключении питания, поэтому нет необходимости предусматривать загрузчик и внешнюю энергонезависимую память для хранения прошивки как в случае с FPGA.

Благодаря этому упрощается схема устройства и не требуются дополнительные компоненты, применяемые для начальной конфигурации ПЛИС. Это ограничение введено для сокращения элементов в УПЗИ.

Описанному выше ограничению удовлетворяет CPLD семейства CoolRunner-II фирмы Xilinx — это семейство программируемых вентильных матриц, предназначенное для применения в системах с напряжением питания 1,8 Вольт, с низким потреблением электроэнергии. Выбрана ПЛИС семейства CoolRunner-II модели XC2C256-7CPG132I ввиду сочетания высокой производительности и низкому потреблению энергии.

Технические характеристики ПЛИС XC2C256:

- электрически стираемое перепрограммируемое ПЗУ КМОП технологии;

- 256 макроячеек;

- 106 входов-выходов;

- задержка 6,7 нс;

- напряжение питания 1,8 Вольт;

- поддержка стандартов ввода/вывода 1.5, 1.8, 2.5 и 3.3 Вольт;

- 180 нм технология;

- 179-323 MHz (меньше ячеек — больше частота).

Для преобразования уровней КМОП микроконтроллера и интерфейса RS232, элемент D3 на структурной схеме, применена микросхема MAX3227 фирмы Maxim Integrated, реализующая функции интерфейса RS232.

Технические характеристики микросхемы MAX3227:

- один передатчик и один приемник;

- максимальная скорость передачи данных – 1 Мбит/с;

- напряжение питания от 3 до 5,5 В;

- номинальный ток потребления – 1 мкА;

- значения напряжений на приемнике:

- вход приемника – от минус 25 до плюс 25 В;

- выход приемника, при напряжении питания 3,3 В – от минус 0,3 до 3,6 В;

- значения напряжений на передатчике:

- вход передатчика – от минус 0,3 до плюс 3,6 В;

- выход передатчика – от минус 13,2 до плюс 13,2 В.

В микросхеме MAX3227 реализована функция автоматического отключения питания при отключенном кабеле RS232 или если на входе передатчика отсутствует сигнал более 30 секунд. Включение питания происходит автоматически при появлении сигнала на любом входе передатчика или приемника.

6 Разработка схемы электрической принципиальной устройства передачи зашифрованной информации

В соответствии с разработанной электрической структурной схемой была разработана схема электрическая принципиальная устройства передачи зашифрованной информации. Схема электрическая принципиальная и перечень элементов устройства передачи зашифрованной информации приведены в приложении В и Г соответственно.

Ниже приведены основные элементы схемы электрической принципиальной:

- микроконтроллер – элемент D1 STM32F415OGY6 фирмы STMicroelectronics;

- ПЛИС - элемент D2 XC2C256-7CPG132I семейства CoolRunner-II фирмы Xilinx;

- приемопередатчик RTX - элемент D5 MAX3227EETE+ фирмы Maxim;

- кварцевый генератор – элемент G1 ASD3-12.000MHz фирмы Abracon Corporation;

- источник вторичного электропитания – элементы D7 TPS62236DRY и D9 LP8758A1E0YFF фирмы Texas Instruments;

- супервизор из состава ИВЭП - элементы D8 MAX16029TG+ фирмы Maxim.

Программируемая логическая интегральная схема, элемент D2, как описывалось в предыдущем разделе, предназначена для защиты от активации закладок в импортных интегральных микросхемах. Предполагается, что защита будет выполнена следующим образом. В микроконтроллере вырабатывается псевдослучайная последовательность (ПСП), которая передается по интерфейсу SPI в ПЛИС. В ПЛИС ПСП накладывается на принимаемую из канала информацию, затем полученные данные передаются по UART в микроконтроллер, где ПСП «снимается». Защита применяется для исключения в принимаемом сигнале навязанной помехи.

Реализация механизма защиты от активации закладок в интегральных микросхемах иностранного производства не рассматривается в рамках данной работы.

Кварцевый генератор G1 предназначен для формирования сигнала опорной тактовой частоты 12 МГц, обеспечивающего синхронную работу всех элементов устройства. Частота работы процессора выбирается в зависимости от требуемого быстродействия и напрямую связана с его энергопотреблением.

Применяемый микроконтроллер оперирует уровнями сигналов 1,8 В, а приемопередатчик D5 уровнями 3,3 В. Поэтому для согласования их логических уровней, а также для защиты выходов микроконтроллера необходимо использовать преобразователей уровней напряжения. Для этого применены микросхемы преобразователей уровней SN74AVC4T234, элементы D3 и D4. На контакты питания данных микросхем подаются напряжения плюс 1,8 В и плюс 3,3 В.

ИВЭП реализован на микросхемах D7 и D9 и предназначен для формирования напряжений питания элементов устройства. Микросхема D7 TPS62236DRY применена для питания микроконтроллера и формирует на выходе напряжение плюс 1,85 В.

Для питания ПЛИС и остальных элементов устройства применена микросхема D9 DC/DC-преобразователь LP8758A1E0YFF. Данный преобразователь работает на частоте 3 МГц и имеет в рабочем режиме КПД порядка 98%.

Узел ИВЭП построен таким образом, чтобы питание элементов, которые обрабатывают открытую и ключевую информацию, было раздельным от питания элементов, которые обрабатывают канальную информацию. Микросхема D9 построена таким образом, что все четыре выходных каскада являются независимыми друг от друга, а также благодаря применяемым LC-фильтрам по питанию, можно исключить как взаимные помехи на плате, так и наводки от опасных сигналов, несущих открытую и ключевую информацию в

цепи первичного электропитания и канальные интерфейсы. Структура микросхемы LP8758A1E0YFF приведена на рисунке 4.

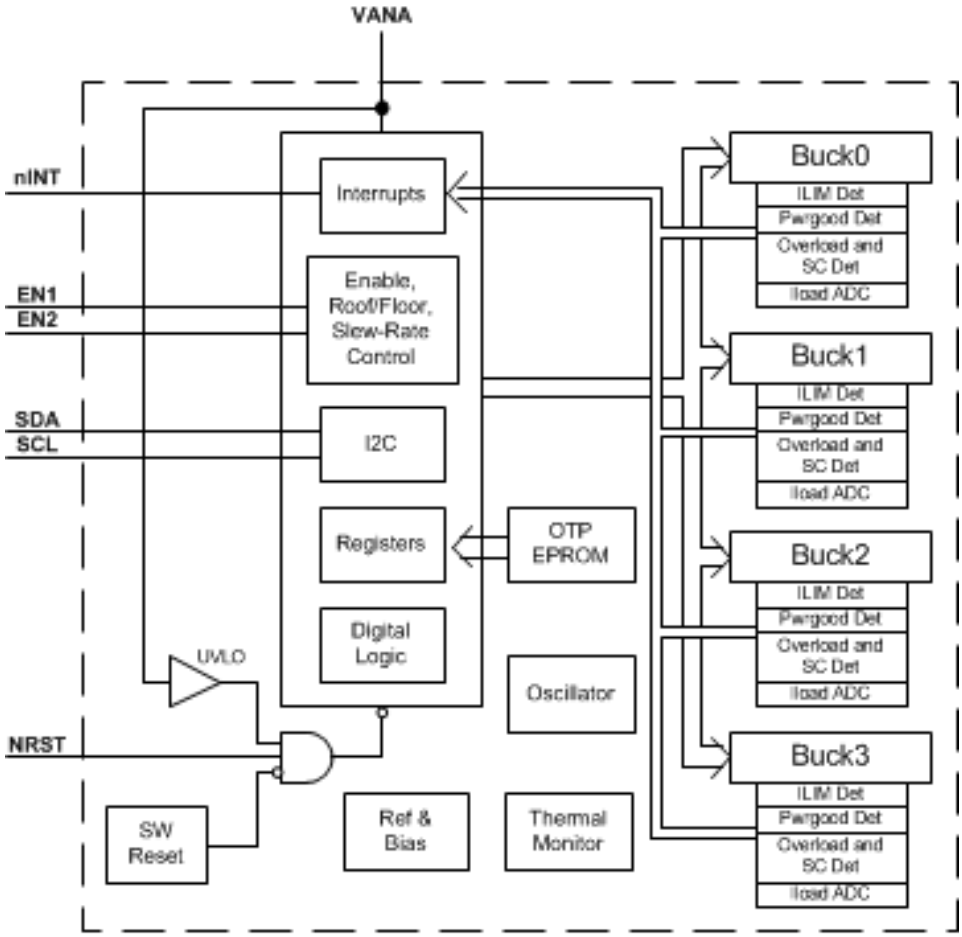


Рисунок 4 – Структура микросхемы LP8758A1E0YFF

Узел ИВЭП имеет функцию включения/отключения со стороны устройства, в которое будет встраиваться устройство передачи зашифрованной информации, в целях экономии ресурса аккумуляторной батареи в режиме экономичного дежурного приёма. Шина I2C между микроконтроллером и DC/DC-преобразователем D9 используется для управления питанием.

В состав ИВЭП входит супервизор D8 MAX16059ATT26+, предназначенный для обеспечения контроля над уровнем напряжения питания устройства и формирования сигнала «СБРОС» при низких напряжениях вторичных источников питания. Это обеспечивает блокировку работы УПЗИ в случае выхода питающих напряжений за допустимые границы. Блокировка работы УПЗИ необходима для предотвращения утечки открытой или криптографически опасной информации в канал связи.

УПЗИ питается от источника электропитания напряжением от 3,2 до 5,5 В. Максимальное расчетное потребление при питании от напряжения 5В должно составлять не более 200 мВт.

7 Реализация и проверка работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа

7.1 Алгоритм изменения и применения уровней защиты от чтения

Для изменения уровня защиты памяти от чтения МК STM32F4 через байты опций FLASH-памяти необходимо сначала разблокировать регистры FLASH-памяти, а затем разблокировать регистр управления опциями FLASH-памяти FLASH_OPTCR.

После сброса МК изменение регистра управления FLASH-памятью FLASH_CR не допускается. Это сделано для защиты FLASH-памяти от возможных нежелательных операций, вызванных, например, электрическими помехами. Для разблокировки регистра управления FLASH-памятью используется следующая последовательность:

а) записать KEY1 = 0x45670123 в регистр ключей FLASH-памяти (FLASH_KEYR);

б) записать KEY2 = 0xCDEF89AB в регистр ключей FLASH-памяти (FLASH_KEYR).

Запись неправильной последовательности вернет ошибку шины и заблокирует регистр управления FLASH-памятью FLASH_CR до следующего сброса МК.

Регистр управления FLASH-памятью FLASH_CR может быть снова заблокирован программно, установив бит LOCK в регистре управления FLASH-памятью FLASH_CR.

Регистр управления FLASH-памятью FLASH_CR недоступен для записи, когда бит BSY в регистре статуса FLASH-памяти FLASH_SR установлен. Любая попытка записи в регистр с установленным битом BSY приведет к остановке системной шины до тех пор, пока бит BSY не будет очищен.

Для выполнения любых операций с байтами опций FLASH-памяти, бит блокировки опций OPTLOCK в регистре контроля опций FLASH-памяти

FLASH_OPTCR должен быть очищен. Для того чтобы очистить этот бит, нужно выполнить следующие операции:

а) записать OPTKEY1 = 0x0819 2A3B в регистре ключей опций FLASH-памяти FLASH_OPTKEYR;

б) записать OPTKEY2 = 0x4C5D 6E7F в регистре ключей опций FLASH-памяти FLASH_OPTKEYR.

Байты опций могут быть защищены от нежелательного стирания или записи, путем программной установки бита OPTLOCK в регистре контроля опций FLASH-памяти FLASH_OPTCR.

Для изменения байт опций на МК серии STM32F4, нужно выполнить следующие операции:

а) убедиться, что не выполняется операций с FLASH-памятью, путем проверки бита BSY в регистре FLASH_SR;

б) введите нужное значение параметра в регистр FLASH_OPTCR;

в) установить бит запуска опций OPTSTRT в регистре FLASH_OPTCR;

г) подождать, пока бит BSY в регистре FLASH_SR будет очищен.

Для предотвращения доступа к байтам опций после окончания настройки опций FLASH-памяти необходимо заблокировать регистр контроля опций FLASH-памяти. Для этого следует установить бит OPTLOCK в регистре контроля опций FLASH-памяти FLASH_OPTCR.

Затем необходимо заблокировать доступ к регистрам FLASH-памяти, установив бит LOCK в регистре управления FLASH-памятью FLASH_CR.

7.2 Способы применения механизма защиты памяти

Аппаратно-программный механизм защиты памяти, встроенный в МК STM32F4xx, может быть применен различными способами. Изменение уровня защиты памяти от чтения может быть выполнено:

- с помощью программы STM32 ST-LINK Utility;
- с помощью программы STM Flash Loader Demo;

- из прошивки, записанной в МК, или из загрузчика при первом включении МК после программирования.

STM32 ST-LINK Utility это программный интерфейс для программирования микроконтроллеров STM32. Он обеспечивает удобную и эффективную среду для считывания, записи и проверки памяти МК.

Данное программное обеспечение предлагает широкий спектр функций для программирования внутренней памяти МК STM32 (FLASH-память, RAM, OTP и др.), внешней памяти, для проверки программ (контрольная сумма, проверка во время и после программирования МК, сравнение с файлом прошивки) и автоматизация программирования STM32.

STM32 ST-LINK Utility поставляется в виде графического интерфейса пользователя с интерфейсом командной строки.

Возможности STM32 ST-LINK Utility:

- поддержка Motorola S19, Intel HEX и двоичные форматы прошивок;

- загрузка, редактирование и сохранение исполняемых файлов и файлов данных, созданных компиляторами Assembler/Linker или C;

- стирание, программирование, просмотр и проверка содержимого FLASH-памяти МК;

- автоматизация программирования STM32 (стирание, проверка, программирование, настройка байт опций, вычисление контрольной суммы);

- программирование OTP памяти;

- поддержка изменения и программирования байт опций;

- сравнение файлов прошивок с прошивкой, записанной в память МК.

С помощью программы STM32 ST-LINK Utility настраиваются байты опций путем выбора нужного уровня защиты памяти от чтения и загрузка конфигурации.

Настройка байт опций при помощи программы STM32 ST-LINK Utility выполняется при подключении МК через программатор к ПК. Подключение МК к программатору выполняется по интерфейсу для отладки и прошивки Serial Wire Debug (SWD) или JTAG.

STM32 Flash loader demonstrator (FLASHER-STM32) представляет собой бесплатную программу для ПК от STMicroelectronics, которая работает на операционной системе Microsoft Windows и соединяется по интерфейсу RS232 с загрузчиком в системной памяти МК STM32.

Возможности STM32 Flash loader demonstrator:

- получение доступа к загрузчику в системной памяти МК по UART;
- утилита программирования FLASH-памяти по интерфейсу RS232;
- поддержка изменения и программирования байт опций.

Для настройки байт опций при помощи программы STM32 Flash loader demonstrator необходимо подключить МК к ПК по интерфейсу RS232. Для этого может использоваться преобразователь USB – UART или другие варианты приемопередатчиков, реализующих интерфейс RS232. Кроме того, нужно перевести МК в режим загрузки из системной памяти. Далее запустить программу STM32 Flash loader demonstrator, перезагрузить МК и соединиться с загрузчиком в системной памяти МК.

Настройка механизма защиты памяти от чтения из прошивки, записанной в МК, при первом включении устройства при загрузке из FLASH-памяти МК производится согласно алгоритму применения и изменения уровня защиты от чтения, приведенному в предыдущем параграфе.

7.3 Разработка схемы алгоритма аппаратно-программного механизма защиты памяти от несанкционированного доступа

Одним из способов применения защиты памяти от чтения, как говорилось в предыдущем параграфе, является изменение уровня защиты памяти от чтения из прошивки при первом включении после программирования МК. Для реализации данного способа был разработана схема программной реализации алгоритма аппаратно-программного механизма защиты памяти от несанкционированного доступа. Разработанная схема алгоритма приведена на рисунке 5.

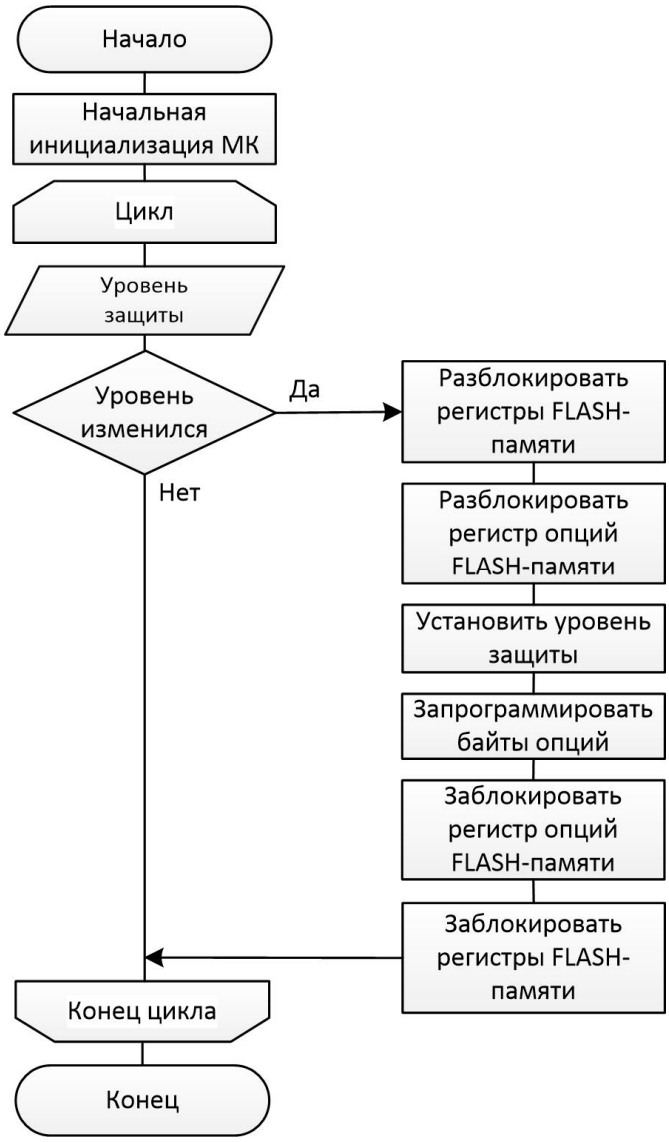


Рисунок 5 – Схема алгоритма аппаратно-программного механизма защиты памяти от несанкционированного доступа

7.4 Программное обеспечение, используемое для разработки программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа

Для разработки программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа для МК STM32, было использовано следующее программное обеспечение:

- STM32CubeMX;
- AC6 System Workbench (SW4STM32).

STM32CubeMX – это графический редактор для конфигурирования микроконтроллеров STM32, позволяющий автоматически генерировать конфигурационный С-код и «скелет» прикладной программы с помощью набора визуальных утилит. Благодаря STM32CubeMX этап создания функций инициализации происходит очень быстро.

Графические генераторы кода позволяют уйти от рутинного написания программ. Вместо самостоятельного создания текстовых файлов разработчик использует графические утилиты, которые автоматически формируют программный код. Кросплатформенный графический редактор STM32CubeMX, работает на всех популярных ОС: Windows, Linux и OS X.

STM32CubeMX позволяет [4]:

- создавать и редактировать проекты для микроконтроллеров STM32 с последующей генерацией С-кода для конкретных интегрированных сред: IAR EWARM, Keil MDK-ARM, Atollic TrueSTUDIO и AC6 System Workbench (SW4STM32);

- производить визуализированную настройку выводов МК с помощью утилиты «Pin Wizard» с возможностью формирования табличного файла для трассировщиков печатных плат;

- производить визуализированную настройку дерева тактирования с помощью утилиты «Clock Wizard»;

- производить визуализированную настройку периферийных блоков и ПО промежуточного уровня, файловой системы, стеков протоколов, ОС с помощью утилиты «Perepherial and middleware Wizard»;

- производить оценку уровня потребления и срока службы аккумулятора при заданных настройках МК с помощью утилиты «Power consumption Wizard»;

- автоматизировать процесс миграции проектов при переходе от одной модели микроконтроллера к другой.

Программа STM32CubeMX значительно упрощает работу программистов, однако у нее также есть ограничения [4]:

- генерируемый С-код охватывает только настройку блоков и периферии микроконтроллера. Это значит, что алгоритмическую часть программы автоматически сгенерировать нельзя, ее нужно будет дописать вручную;

- STM32CubeMX поможет создать только стартовую конфигурацию. Иногда в процессе работы пользователю требуется изменить частоту работы периферийного блока или изменить конфигурацию вывода. Все это придется прописывать самостоятельно;

- для генерации кода используются стандартные, разработанные ST, библиотеки нижнего уровня (HAL и LL);

- в процессе генерации С-файл выстраивается таким образом, что для пользователя выделяются специальные секции, в которые он может помещать свой код. Если пользовательский код окажется вне этих рамок – он будет затерт при следующих генерациях.

AC6 System Workbench (SW4STM32) – это интегрированная среда разработки встраиваемого ПО компании AC6 для микроконтроллеров STM32, позволяющая писать программы, загружать и отлаживать их.

Для редактирования, компиляции и отладки программ предназначены специальные интегрированные среды разработки. Большая часть из них является коммерческими продуктами (такие как IAR EWARM, Keil MDK-ARM, Atollic TrueSTUDIO и другие), но есть и бесплатные инструменты, например, System Workbench производства компании AC6. В рамках системы названий компании STMicroelectronics данная интегрированная среда разработки носит название SW4STM32.

Интегрированная среда SW4STM32 предназначена для работы с микроконтроллерами STM32. Она основана на базе платформы Eclipse и является бесплатной и кроссплатформенной. Основными ее достоинствами являются [5]:

- поддержка работы с микроконтроллерами STM32, аппаратными отладочными наборами, с программными библиотеками (Standard Peripheral library и STM32Cube HAL);

- отсутствие ограничений на объем программного кода;

- бесплатный компилятор GCC C/C++;

- свободный отладчик GDB (проект GNU);

- открытая платформа Eclipse IDE с поддержкой групповой разработки встраиваемого ПО с системой контроля версий SVN/GIT;

- поддержка ST-LINK;

- кроссплатформенность и совместимость с ОС Windows, Linux и OS X.

С помощью SW4STM32 можно редактировать, компилировать и отлаживать программы для STM32. Для этого удобно использовать скелеты программ, создаваемые в STM32CubeMX.

SW4STM32 имеет много возможностей, но у данной среды разработки есть и недостатки, достаточно характерные для бесплатных сред:

- отсутствие встроенного симулятора;

- компилятор GCC проигрывает своим коммерческим собратьям по объему кода и быстродействию;

- поддержка SW4STM32 со стороны разработчиков не будет столь оперативной, как в случае с платными средами.

7.5 Разработка программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа

Исходя из того, что проверка работоспособности и тестирование встроенного механизм защиты памяти от чтения в МК STM32, будет использоваться отладочная плата STM32F407G-DISC1, то программа также разрабатывалась под МК STM32F407VGTx.

STM32F407G-DISC1 – это отладочная плата, которая позволяет пользователю быстро разрабатывать приложения для 32-битного ARM Cortex-M4 микроконтроллера STM32F407. Отладочная плата основана на МК STM32F407VGT6. Данная отладочная плата выбрана из-за того, что используемый на ней МК из того же семейства STM32F4, что и МК, применяемый в устройстве передачи зашифрованной информации. Это

означает, что их функционал практически идентичен. Кроме того, на плате имеется встроенный программатор/отладчик ST-Link/V2-1, благодаря которому к МК можно подключаться по интерфейсу USB, то есть не требуются внешние программаторы.

По разработанной схеме алгоритма написана программа, реализующая аппаратно-программный механизм защиты памяти от несанкционированного доступа. Программа приведена в приложении Д.

Для разработки требуемой программы в конфигураторе для МК STMicroelectronics STM32CubeMX был создан проект для платформы STM32F407VGTx STM32F4-DISCOVERY. Далее в созданном проекте были определены назначение выводов МК, настроена система тактирования, выбрана целевая интегрированная среда разработки – AC6 System Workbench и сгенерирован С-код с использованием HAL-драйверов. В результате этого в указанной директории проекта был сгенерирован проект для среды разработки SW4STM32.

В среде разработки была написана программа по схеме алгоритма аппаратно-программного механизма защиты памяти от несанкционированного доступа. При написании программы были использованы функции библиотеки HAL.

Основная функция, реализующая алгоритм изменения и применения уровней защиты от чтения, SetRDPLevel() вызывается при получении команды по интерфейсу UART. Если получена команда на изменение уровня защиты от чтения, последовательно вызываются необходимые функции.

В начале, вызывается функция HAL_FLASH_Unlock(), которая служит для разблокировки регистра управления FLASH-памятью.

Затем вызывается функция HAL_FLASH_OB_Unlock(), которая разблокирует опции FLASH-памяти для получения доступа к байтам опций. Если удалось разблокировать опции FLASH-памяти, с помощью функции HAL_FLASHEx_OBProgram() в байты опций записывается желаемая конфигурация.

Далее с помощью функции HAL_FLASH_OB_Launch() происходит применение изменений в байтах опций.

В случае если был задан уровень 1 защиты от памяти, с этого момента больше нельзя получить доступ к FLASH-памяти и резервной SRAM МК через отладочные интерфейсы.

Для получения сведений о состоянии МК, а именно об установленном уровне защиты памяти от чтения, МК по интерфейсу UART отправляет информацию о том, какой уровень защиты сейчас установлен.

Для предотвращения доступа к байтам опций после окончания настройки опций FLASH-памяти с помощью функции HAL_FLASH_OB_Lock() происходит блокировка регистра контроля опций FLASH-памяти.

Затем с помощью функции HAL_FLASH_Lock() блокируется доступ к регистрам FLASH-памяти.

8 Проверка работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа

8.1 Способы проверки работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа

Проведение проверок встроенного механизма защиты памяти МК, применяемого в качестве аппаратно-программного механизма защиты памяти от НСД, необходимо для обоснования работоспособности и правильности функционирования. Необходимость проведения проверок вызвана тем, что используемый в УПЗИ МК является продуктом иностранного производства. Кроме того, применение рассматриваемого механизма в качестве аппаратно-программного механизма защиты памяти от НСД в УПЗИ вызывает необходимость проверки функционирования механизма при различных ситуациях, имитирующих несанкционированный доступ со стороны противника.

Проверки механизма защиты памяти от НСД были проведены на отладочной плате STM32F407G-DISC1 и на ячейке СД798 с контроллером STM32F415OGY6.

Для проверки возможности получения доступа к FLASH-памяти МК при включенном механизме защиты памяти от чтения могут быть применены следующие способы:

- получение содержимого FLASH-памяти МК по отладочному интерфейсу SWD или JTAG;
- получение содержимого FLASH-памяти МК через встроенный аппаратный загрузчик по интерфейсу UART.

Для подключения к МК по интерфейсу SWD используется встроенный в отладочную плату STM32F407G-DISC1 отладчик ST-Link/V2-1.

Получение сведений о состоянии МК, а именно об установленном уровне защиты памяти от чтения, МК был подключен по интерфейсу USB – UART к компьютеру. Для подключения по интерфейсу USB – UART используется

модуль преобразователя USB – UART, построенный на микросхеме FT232RL. По интерфейсу USB – UART через терминальную программу осуществляется получение сведений о состоянии МК и управление уровнями защиты памяти от чтения.

8.2 Получение содержимого FLASH-памяти МК по отладочному интерфейсу SWD

Подключение отладочной платы к компьютеру в конфигурации отладчик ST-Link/V2-1 и USB – UART преобразователь приведено на рисунке 6.

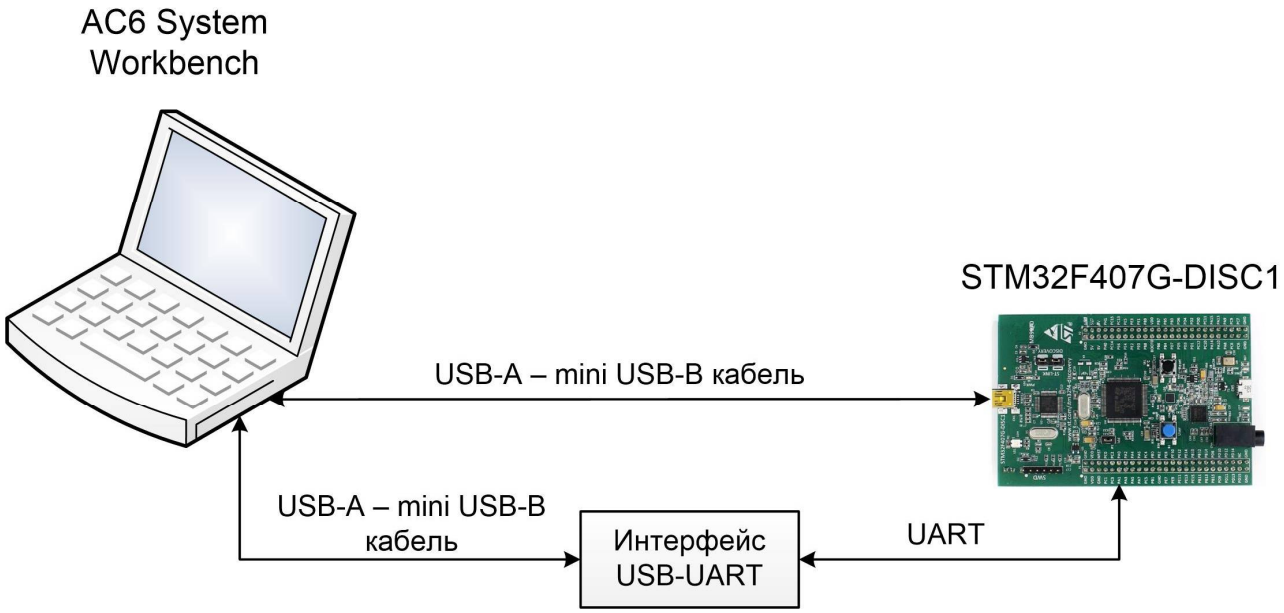


Рисунок 6 – Подключение отладочной платы к компьютеру в конфигурации отладчик ST-Link/V2-1 и USB – UART преобразователь

Получение содержимого FLASH-памяти МК было выполнено программой STM32 ST-LINK Utility. Для проверки работы испытательного стенда в FLASH-память МК отладочной платы была загружена тестовая прошивка без применения механизма защиты памяти от чтения. Далее через программу STM32 ST-LINK Utility была считана загруженная во FLASH-память МК прошивка. Результат считывания прошивки приведен на рисунке 7.

Сравнение считанной из FLASH-памяти МК прошивки с бинарным файлом тестового проекта средствами программы STM32 ST-LINK Utility показало их полную идентичность. Результат приведен на рисунке 8.

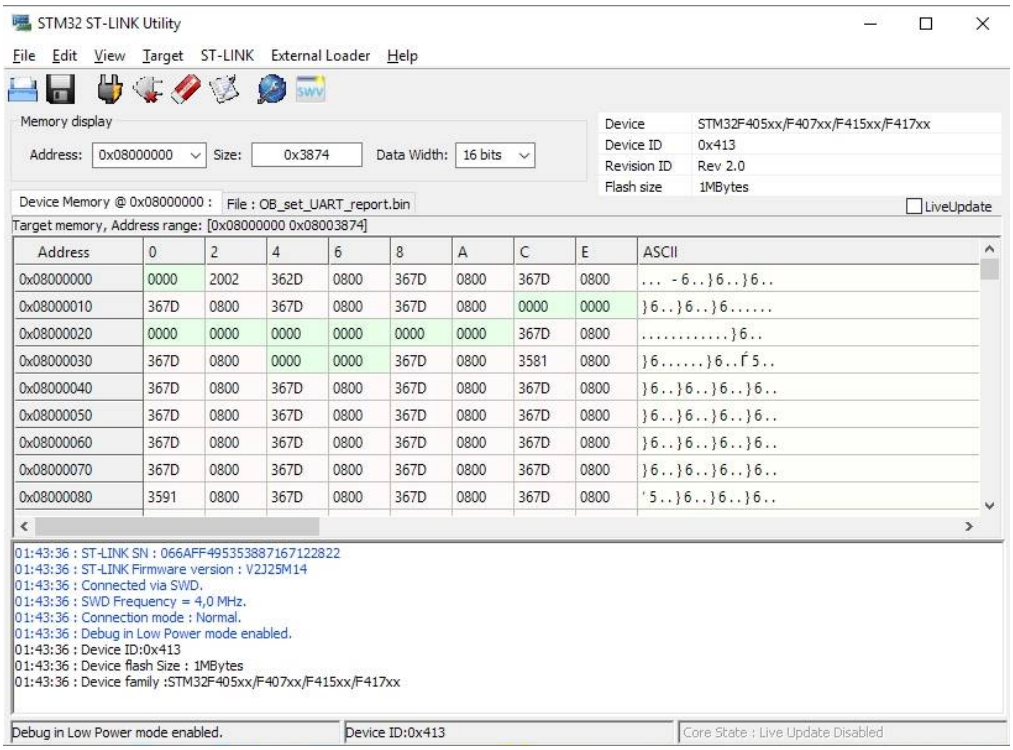


Рисунок 7 – Результат считывания прошивки

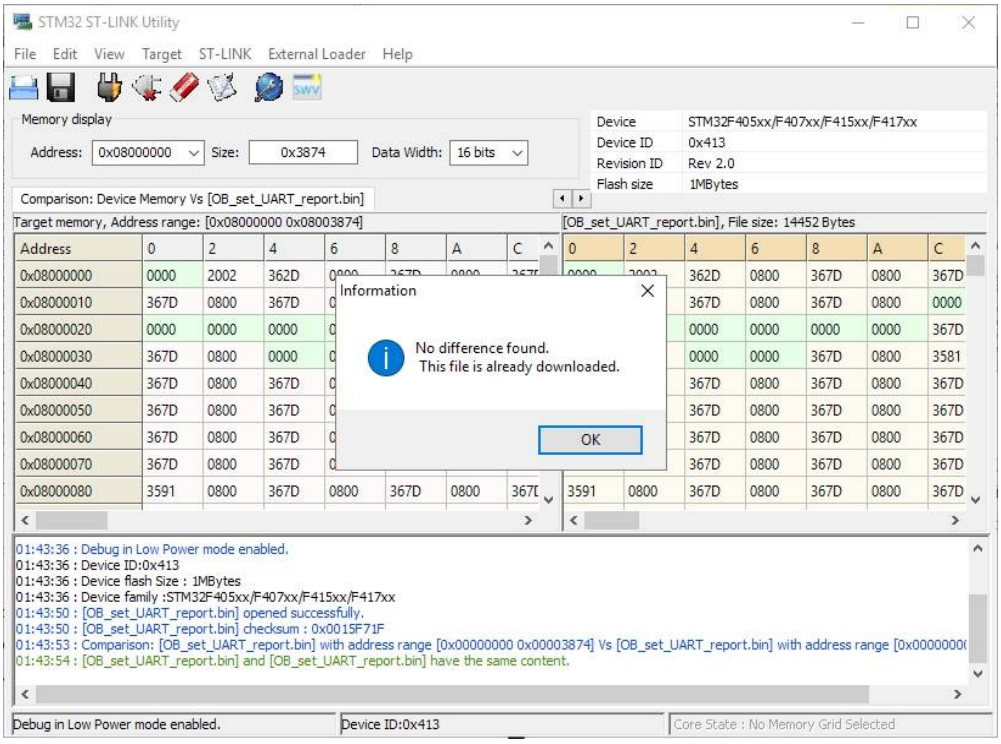


Рисунок 8 – Результат сравнения считанной прошивки с бинарным файлом тестового проекта

Применение механизма защиты памяти от НСД, встроенного в МК STM32F4xx, с помощью программы STM32 ST-LINK Utility было выполнено следующим образом. Во FLASH-память МК отладочной платы была загружена

тестовая прошивка. Далее через программу STM32 ST-LINK Utility был установлен и применен уровень 1 защиты памяти от чтения. Результат приведен на рисунках 9 и 10.

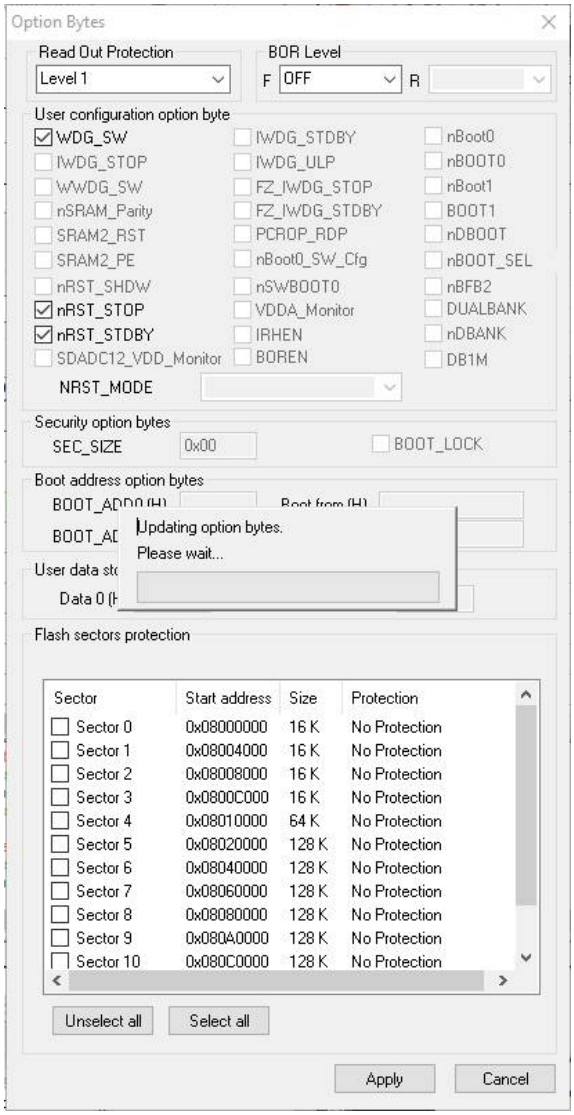


Рисунок 9 – Установка и применение уровня 1 защиты памяти от чтения

На рисунке 10 приведена попытка считывания FLASH-памяти МК после применения уровня 1 защиты памяти от чтения. Попытка завершилась ошибкой о невозможности чтения FLASH-памяти МК из-за включенного механизма защиты памяти от НСД.

Далее уровень 1 защиты памяти от чтения был понижен до уровня 0 с помощью той же программы STM32 ST-LINK Utility. В результате завершения операции FLASH-память МК была полностью затерта, путем записи во все

ячейки FLASH-памяти единиц. Результат снятия защиты памяти от чтения приведен на рисунке 11.

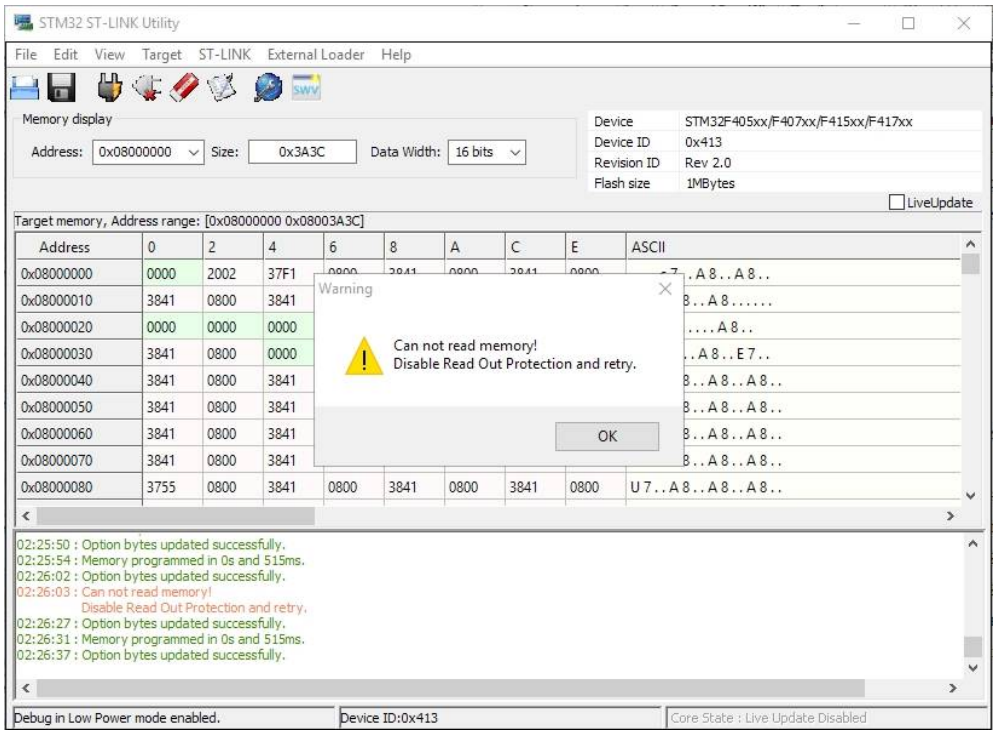


Рисунок 10 – Результат применения уровня 1 защиты от чтения

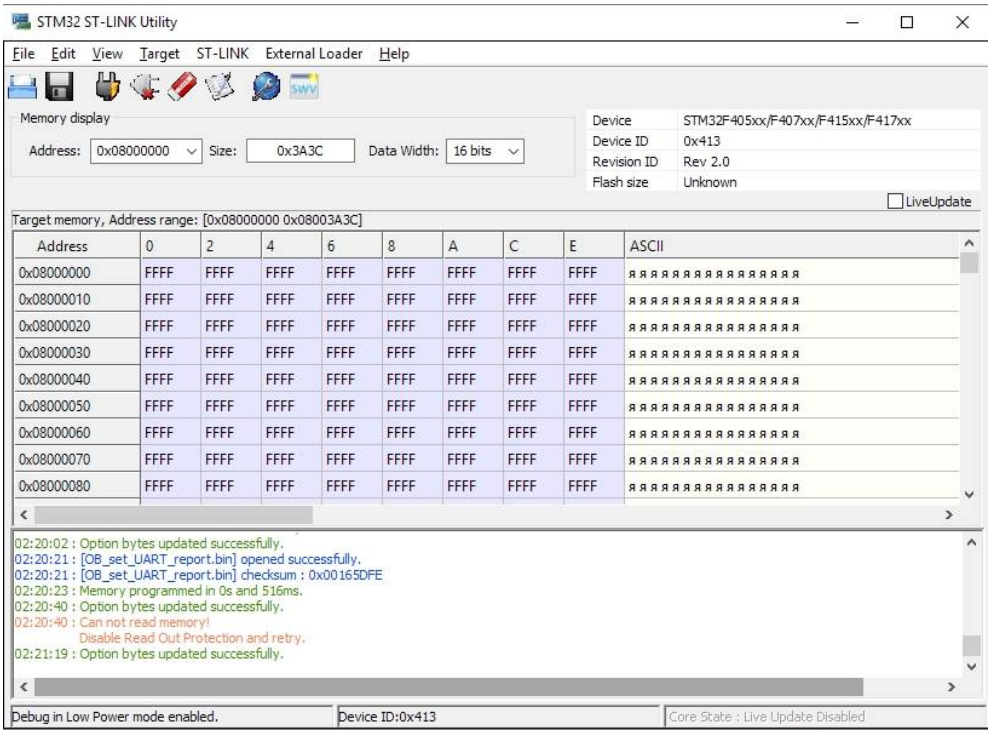


Рисунок 11 – Результат снятия защиты от чтения

8.3 Получение содержимого FLASH-памяти МК по отладочному интерфейсу JTAG

Для подключения к МК по отладочному интерфейсу JTAG был использован внутрисхемный эмулятор J-Link for ARM.

Подключение отладочной платы к компьютеру в конфигурации внутрисхемный эмулятор J-Link for ARM и USB – UART преобразователь приведено на рисунке 12.

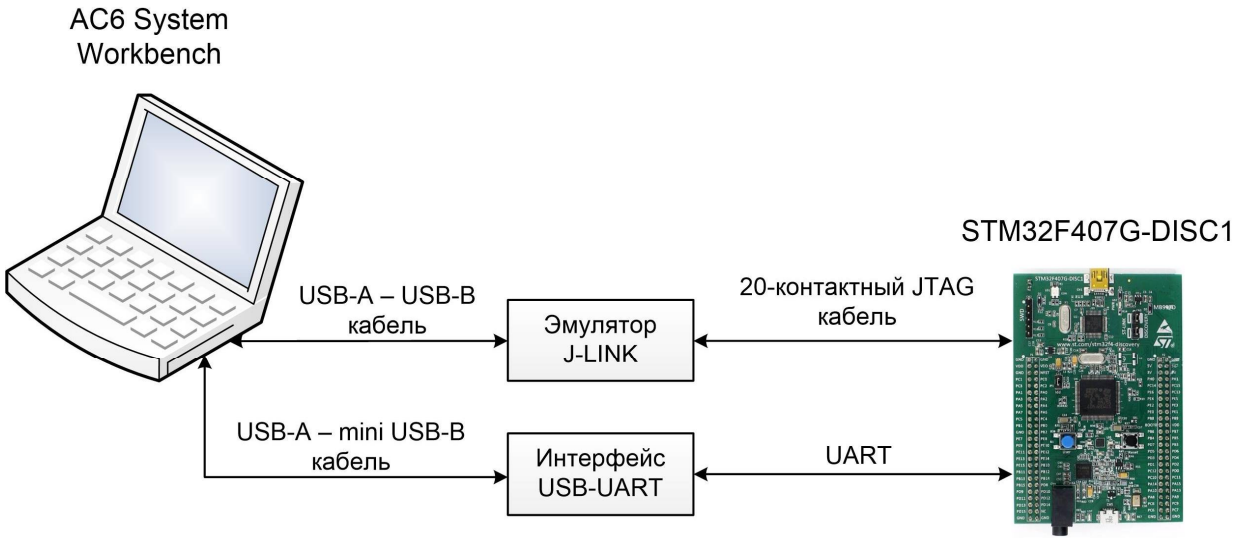


Рисунок 12 – Подключение отладочной платы к компьютеру в конфигурации внутрисхемный эмулятор J-Link for ARM и USB – UART преобразователь

Получение содержимого FLASH-памяти МК по отладочному интерфейсу JTAG было выполнено программой SEGGER J-Flash. Для проверки работы испытательного стенда, приведенного на рисунке 12, в FLASH-память МК отладочной платы была загружена тестовая прошивка без применения механизма защиты памяти от чтения. С помощью программы SEGGER J-Flash была считана записанная во FLASH-память МК прошивка. В результате считывания прошивки был получен бинарный файл. Сравнение полученного бинарного файла и бинарного файла тестового проекта средствами программы STM32 ST-LINK Utility показало их полную идентичность.

Для проверки возможности получение содержимого FLASH-памяти МК по отладочному интерфейсу JTAG, при включенном механизме защиты памяти от чтения, с помощью программы STM32 ST-LINK Utility был установлен уровень 1 защиты памяти от чтения. После подключения по отладочному интерфейсу JTAG SEGGER J-Flash было получено предупреждение о том, что

включен механизм защиты памяти от чтения. Результат приведен на рисунке 13.

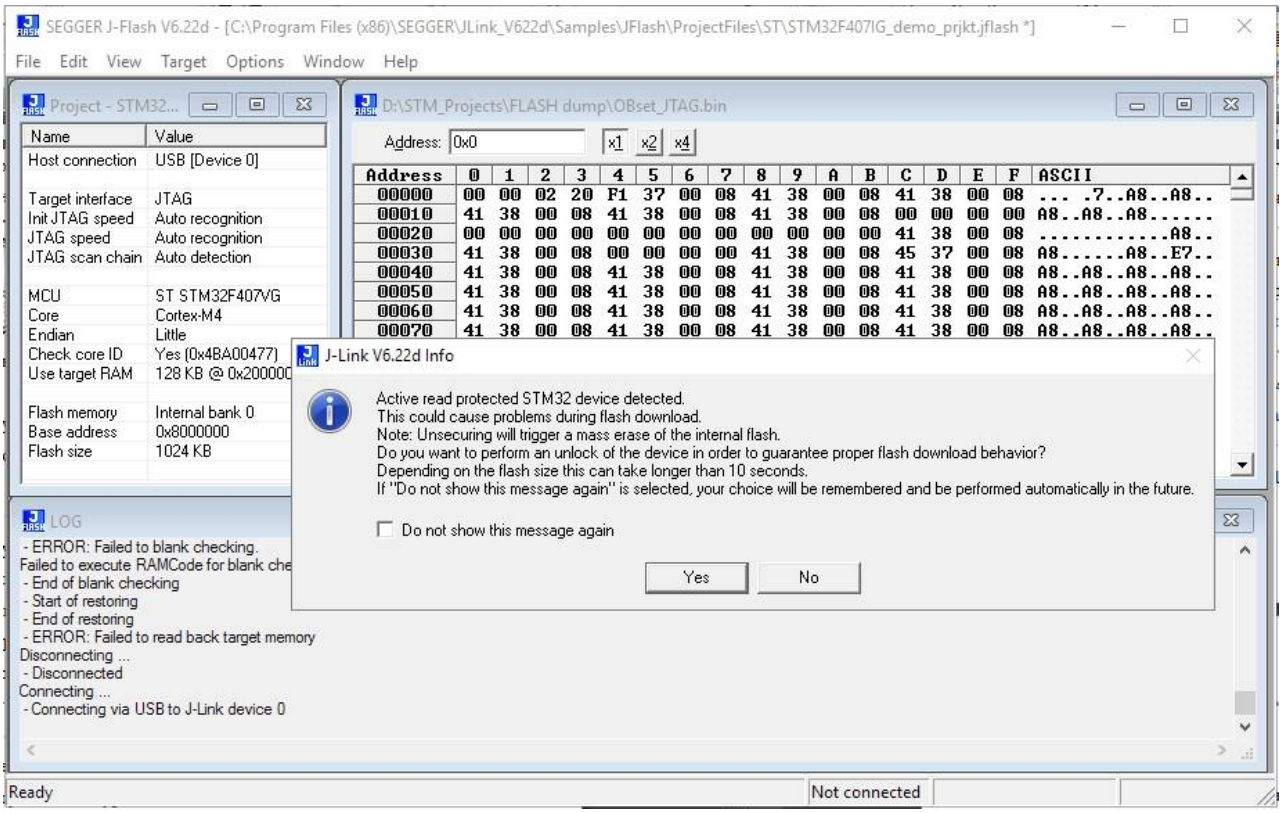


Рисунок 13 – Попытка доступа к FLASH-памяти МК с помощью программы JTAG SEGGER J-Flash

Подключение к МК по отладочному интерфейсу JTAG было установлено, но считать содержимое FLASH-память МК считать не удалось.

Отключение защиты памяти от чтения с помощью утилиты SEGGER J-Link Unlock tool for STM32 devices привело к стиранию FLASH-память МК.

8.4 Получение содержимого FLASH-памяти МК через встроенный аппаратный загрузчик

Получение содержимого FLASH-памяти МК через встроенный аппаратный загрузчик по интерфейсу UART было выполнено с помощью программы STM Flash Loader Demo.

Встроенный загрузчик расположен во встроенной загрузочной ROM памяти (системной памяти) микроконтроллеров STM32, которая записывается при производстве МК. Встроенный загрузчик используется для записи

прошивки во встроенную FLASH-памяти МК через один из доступных интерфейсов:

- USART;

- CAN;

-DFU (USB FS);

- I^2C;

- SPI и т.д.

Следует упомянуть, что при установленном уровне 2 защиты памяти от чтения, невозможно загрузиться из системной памяти МК и встроенный загрузчик не может быть запущен.

Для подключения к встроенному аппаратному загрузчику МК по интерфейсу UART необходимо изменить загрузочную конфигурацию МК STM32F4xx. По умолчанию, загрузка МК STM32F4xx возможна только из FLASH-памяти МК. Однако в МК STM32F4xx имеется специальный механизм, позволяющий загрузиться из других областей памяти (например, из встроенной SRAM).

Для МК STM32F4xx могут быть выбраны три разных режима загрузки, в зависимости от состояния пинов BOOT, как показано в таблице 4.

Таблица 4 – Режимы загрузки

Пины выбора режима загрузки		Режим загрузки	Описание
BOOT1	BOOT0		
x	0	Основная FLASH-память	Основная FLASH-память выбрана в качестве загрузочного пространства
0	1	Системная память	Системная память выбрана в качестве загрузочного пространства
1	1	Встроенная SRAM	Встроенная SRAM выбрана в качестве загрузочного пространства

Значения на пинах BOOT считываются сразу после перезагрузки МК. Это позволяет установить пины BOOT1 и BOOT0 после перезагрузки МК для выбора режима загрузки. Как видно из таблицы 5 для выбора в качестве

загрузочного пространства системной памяти нужно обеспечить на пине BOOT1 низкий, а на пине BOOT0 высокий логический уровень.

После конфигурации загрузки МК из системной памяти было выполнено подключение МК на отладочной плате STM32F407G-DISC1 через USB – UART преобразователь. На рисунке 14 показано подключение МК в режиме загрузки из системной памяти к компьютеру через USB – UART преобразователь.

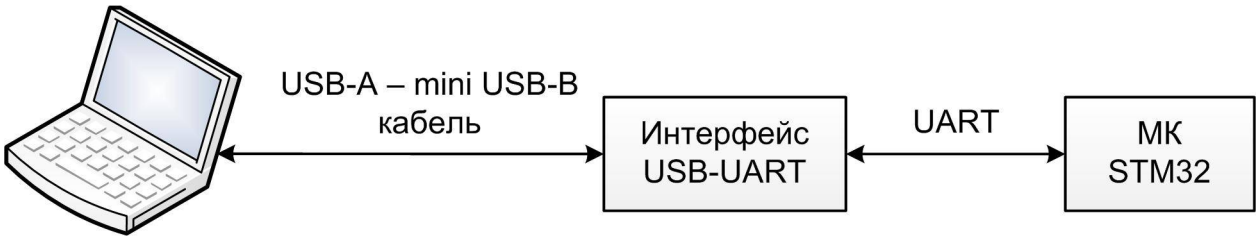


Рисунок 14 – Подключение МК в режиме загрузки из системной памяти к компьютеру через USB – UART преобразователь

Получение содержимого FLASH-памяти МК было выполнено программой STM Flash Loader Demo. Для проверки работы испытательного стенда в FLASH-память МК отладочной платы была загружена тестовая прошивка без применения механизма защиты памяти от чтения. Далее через программу STM Flash Loader Demo была считана записанная во FLASH-память МК прошивка. В результате считывания прошивки был получен бинарный файл. Сравнение полученного бинарного файла и бинарного файла тестового проекта средствами программы STM32 ST-LINK Utility показало их полную идентичность.

Для проверки возможности получение содержимого FLASH-памяти МК через встроенный аппаратный загрузчик по интерфейсу UART при включенном механизме защиты памяти от чтения, с помощью программы STM32 ST-LINK Utility был установлен уровень 1 защиты памяти от чтения. После подключения к встроенному загрузчику с помощью программы STM Flash Loader Demo было получено предупреждение о том, что включен механизм защиты памяти от чтения. Результат приведен на рисунке 15.

Отключение защиты памяти от чтения с помощью программы STM Flash Loader Demo привело к стиранию FLASH-память МК.

Рисунок 15 – Попытка доступа к FLASH-памяти МК с помощью программы STM Flash Loader Demo

8.5 Тестирование программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа

Проведено тестирование разработанной программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа. Для этого в FLASH-память МК отладочной платы была загружена разработанная программа.

По интерфейсу USB – UART через терминальную программу осуществляется получение сведений о состоянии МК и управление уровнями защиты памяти от чтения.

Подключение отладочной платы к компьютеру приведено на рисунке 6.

Управление уровнями защиты памяти от чтения происходит командами, передаваемыми через терминальную программу. Предусмотрены следующие команды:

- команда получения сведений об установленном уровне защиты памяти от чтения;

- команда установки и применения уровня 1 защиты памяти от чтения;

- команда установки и применения уровня 0 защиты памяти от чтения.

На рисунке 16 приведено окно терминальной программы, после отправки последовательности команд:

а) получения сведений об установленном уровне защиты памяти от чтения;

б) команда установки и применения уровня 1 защиты памяти от чтения;

в) получения сведений об установленном уровне защиты памяти от чтения;

г) команда установки и применения уровня 0 защиты памяти от чтения.

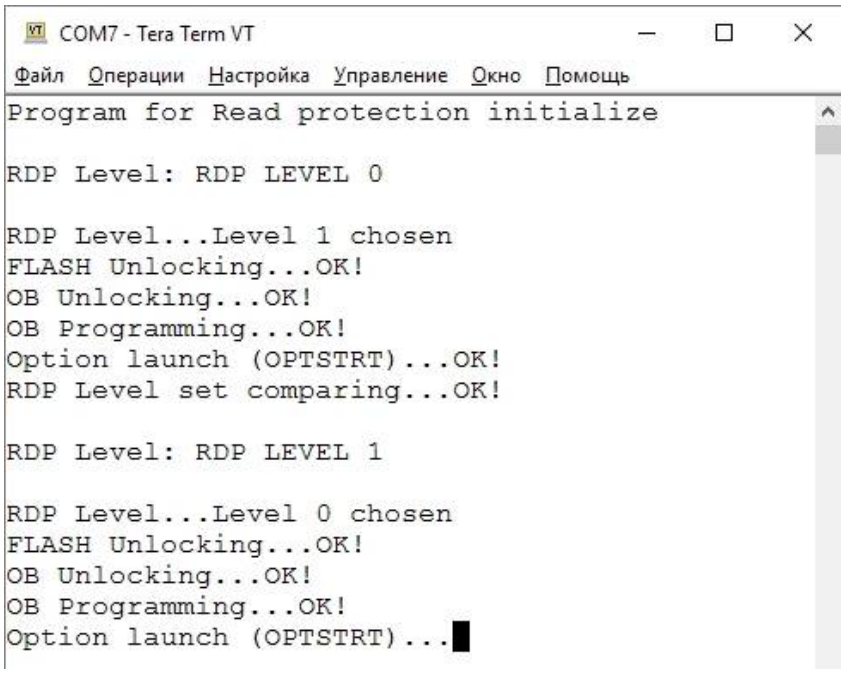


Рисунок 16 – Окно терминальной программы, после отправки последовательности команд

После отправки команды из пункта г дальнейшее взаимодействие с МК невозможно, так как при переходе снижении защиты памяти от чтения с уровня

1 до уровня 0 запускается полное стирание FLASH-памяти и резервного SRAM МК. Стирание FLASH-памяти занимает 10-15 секунд, затем механизм защиты памяти от чтения сбрасывается на уровень 0.

Проверка возможности чтения FLASH-памяти МК, при включенном механизме защиты памяти от чтения, с помощью программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа, была осуществлена способами, описанными ранее. Перечень способов проверок:

- получение содержимого FLASH-памяти МК по отладочному интерфейсу SWD с помощью программы STM32 ST-LINK Utility через отладчик ST-Link/V2-1;

- получение содержимого FLASH-памяти МК по отладочному интерфейсу JTAG с помощью программы SEGGER J-Flash через внутрисхемный эмулятор J-Link for ARM;

- получение содержимого FLASH-памяти МК через встроенный аппаратный загрузчик по интерфейсу UART с помощью программы STM Flash Loader Demo.

Все проведенные проверки завершились одинаково – при включенном механизме защиты памяти от НСД содержимое FLASH-памяти МК прочитать не удалось. А выключение механизма защиты приводило к полному стиранию FLASH-памяти МК.

Функция механизма защиты памяти от НСД, которая заключается в блокировании работы МК в случае обнаружения подключенного отладчика, при включенном механизме защиты, была проверена следующим образом.

Программа, реализующая аппаратно-программный механизм защиты памяти от НСД, была дополнена функцией отправки значения времени работы МК после последней перезагрузки МК. Отсчет начинается с нуля и каждую секунду по интерфейсу UART отправляется текущее значение времени работы МК. Если очередной отсчет не пришел, то можно сделать вывод об остановке работы МК.

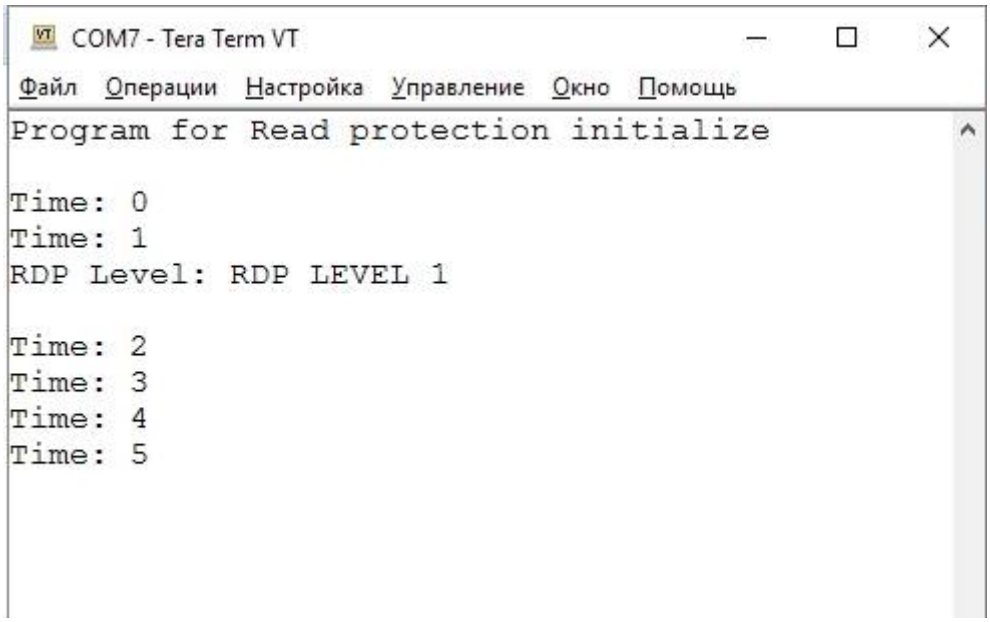


Рисунок 17 – Окно терминальной программы, при блокировке работы МК

Тестирование также было проведено на ячейке СД798 с контроллером STM32F415OGY6. Подключение ячейки СД798 к компьютеру через внутрисхемный эмулятор J-Link for ARM приведено на рисунке 18

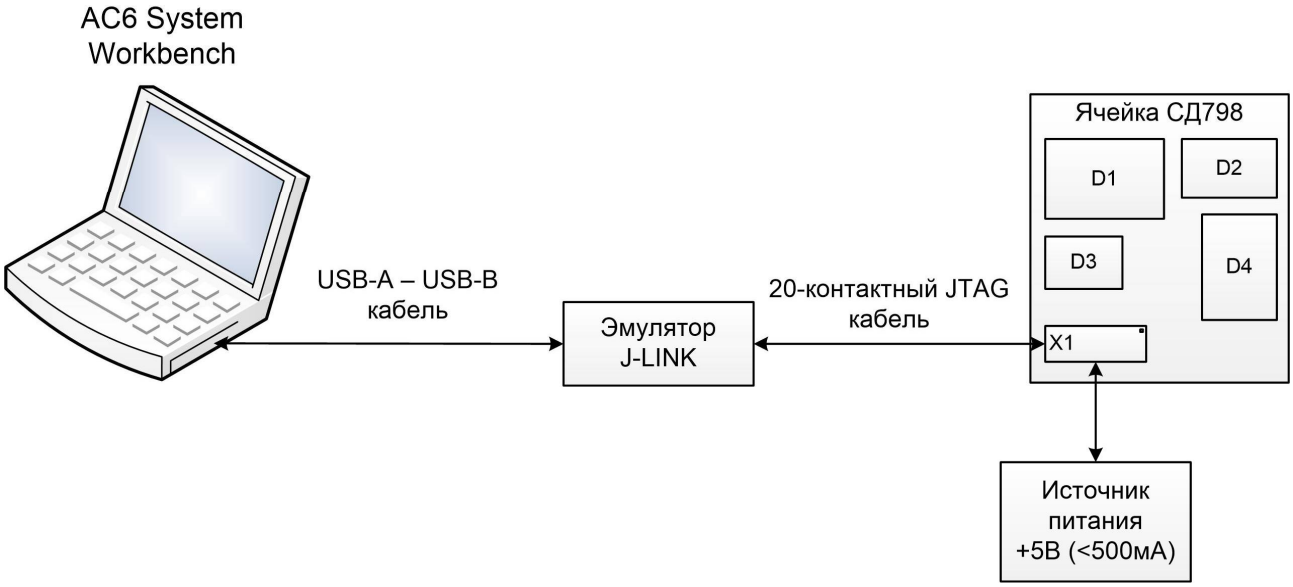


Рисунок 18 – Подключение ячейки СД798 к компьютеру через внутрисхемный эмулятор J-Link for ARM

Для тестирования разработанной программы на ячейке СД798 были использованы те же способы, что и для отладочной платы STM32F407G-DISC1.

Все проведенные проверки завершились одинаково – при включенном механизме защиты памяти от НСД содержимое FLASH-памяти МК прочитать

не удалось. А выключение механизма защиты приводило к полному стиранию FLASH-памяти МК.

8.6 Результаты проверки работоспособности аппаратно-программного механизма защиты памяти от несанкционированного доступа

При проведении проверок механизма защиты были использованы различные способы применения механизма защиты памяти от чтения, а также различные способы получения доступа к FLASH-памяти МК при включенном механизме защиты памяти от чтения.

Все приведенные способы проверок завершились одинаково – при включенном механизме защиты памяти от НСД содержимое FLASH-памяти МК прочитать не удалось. А выключение механизма защиты приводило к полному стиранию FLASH-памяти МК.

Механизм защиты памяти от чтения начинает действовать сразу же после применения опций FLASH-памяти (установка бита OPTSTRT в регистре FLASH_OPTCR). Из-за этого сразу после применения механизма защиты памяти от чтения дальнейшая отладка и доступ к FLASH-памяти МК невозможны.

При обнаружении подключенного отладчика по интерфейсу SWD или JTAG происходит блокировка работы МК до следующего сброса при включении МК. Выполнение программы блокируется до следующей перезагрузки МК – необходимо отключить и снова подать питание на МК.

Применение уровня 2 защиты памяти от чтения не рассматривается в данной работе, потому что уровень 2 является необратимым – после применения уровня 2 нельзя вернуться на уровень 1 или уровень 0. Применение уровня 2 защиты памяти от чтения приводит к отключению отладочных интерфейсов и возможности загрузки из системной и оперативной памяти. Из-за этого невозможно изменить загруженное в МК программное обеспечение, если не был предусмотрен пользовательский механизм обновления прошивки из FLASH-памяти, так называемый пользовательский загрузчик.

Применение уровня 2 защиты памяти от чтения возможно после завершения настройки и эксплуатации опытного образца для последующего серийного (массового) выпуска устройства передачи зашифрованной информации. Уровень 2 позволяет защититься от загрузки из оперативной памяти МК, в которой может находиться информация, подлежащая защите.

Из проведенных проверок можно сделать вывод, что рассматриваемый аппаратно-программный механизм защиты памяти от несанкционированного доступа функционирует на должном уровне.

Однако, не смотря на невозможность получения доступа к FLASH-памяти МК, при включенном механизме защиты памяти, описанными способами нельзя утверждать, что данный механизм стойкий.

Во-первых, это следует из того, что используемый в УПЗИ МК является продуктом иностранного производства. Не исключается наличие обходных путей получения доступа к FLASH-памяти МК предусмотренные производителем.

Во-вторых, рассматриваемый механизм не может быть применен в УПЗИ как единственный механизм защиты памяти от НСД без дополнительных программных механизмов защиты от НСД, контроля целостности СКЗИ, экстренного уничтожения ключевой и криптографически опасной информации и аутентификации субъектов доступа.

Исходя из того, что устройством передачи зашифрованной информации предполагается обработка информации, не содержащей сведений, составляющих государственную тайну, а также, что актуальность этой информации не является долговременной, основным объектом защиты в УПЗИ является ключевая информация.

Ключевую информацию в памяти УПЗИ предполагается хранить не в открытом, а в маскированном виде. Маска с ключевой информации снимается только перед непосредственным использованием.

Согласно требованиям [6] криптографические ключи, в отношении которых возникло подозрение в компрометации, а также действующие

совместно с ними другие криптографические ключи необходимо немедленно вывести из действия. Относительно разрабатываемого УПЗИ, наиболее вероятным является компрометация криптографических ключей при захвате противником аппаратуры связи, в которое встроено УПЗИ.

Учитывая, что ключевую информацию в памяти УПЗИ предполагается хранить в маскированном виде, обеспечивается вывод криптографических ключей из действия в случае их компрометации и кратковременной актуальности обрабатываемой информации, рассматриваемый аппаратно-программный механизм защиты памяти от НСД будет обеспечивать достаточную безопасность ключевой информации.

Применение встроенных в МК средств защиты памяти позволило исключить часть схемы, которая использовалась в ранее разработанных устройствах для реализации функций защиты от НСД. Сокращение количества элементов схемы позволит снизить стоимость будущего устройства, уменьшить его массогабаритные показатели, сократить время настройки при производстве, увеличить общую надёжность устройства.

Благодаря применению встроенного в МК STM32F4xx механизма защиты памяти от чтения удалось отказаться от схемы защиты от НСД, приведенного в приложении А. Из этого следует, что количество элементов на схеме сокращается, вслед за этим уменьшаются стоимость и габариты платы будущего устройства. В то же время повышается надежность устройства из-за уменьшения количества элементов, которые могут отказать.

По примерным оценкам габариты платы уменьшатся на 7%. Учитывая, что габариты изделия-450 составляют 30 мм на 44 мм, то габариты платы будущего устройства должны составлять примерно 32 мм на 44 мм.

9 Вредные психофизиологические факторы, влияющие на опорно-двигательный аппарат специалиста по защите информации. Методы и средства физической культуры, снижающие их воздействие

Согласно ГОСТ 12.0.003–2015 [7] опасные и вредные производственные факторы подразделяются на четыре группы:

- физические опасные и вредные производственные факторы;
- химические опасные и вредные производственные факторы;
- биологические опасные и вредные производственные факторы;
- психофизиологические факторы - это физические (статические и динамические) перегрузки, нервно-психические перегрузки [7].

Физические перегрузки подразделяются на динамические и статические.

Динамическая перегрузка связана с перемещением материалов, полуфабрикатов, готовых изделий и т.п. на необходимые расстояния.

Статическая нагрузка вызвана необходимостью специалиста по защите информации прилагать усилия без перемещения всего тела или отдельных частей тела.

Вредные психофизиологические факторы, влияющие на опорно-двигательный аппарат специалиста по защите информации, согласно этой классификации, возникают из-за физических статических перегрузок.

Специалист по защите информации подвергается длительным статическим нагрузкам в течение всего рабочего дня. Из-за длительных статических нагрузок затормаживаются обменные процессы в организме. Все это вызвано рабочей позой специалиста по защите информации, в которой он находится большую часть рабочего времени. Долгое нахождение в положении сидя негативно влияет на систему кровообращения организма, возникает большая нагрузка на шейный и поясничный отделы позвоночника. Это связано с длительным нахождением в положении сидя, особенно с наклоном головы и туловища вперед. Для снижения утомления мышц шеи и спины, то есть для

рациональной рабочей позы необходимо уменьшить величину наклона головы и корпуса [8].

Для предупреждения развития болезней опорно-двигательного аппарата при организации рабочего места специалиста по защите информации важно соблюсти требования к конструкции мебели (столы и стулья, кресла), а именно к основным анатомическим, физиологическим и эргономическим требованиям. Конструкция кресла должна быть разработана таким образом, чтобы как можно равномернее распределить давление тела на площадь опоры. Это возможно в том случае, когда кресло в наибольшей степени соответствует анатомическому строению тела работника.

Основным средством физической культуры служащим для борьбы с вредными психофизиологическими факторами, влияющими на опорно-двигательный аппарат специалиста по защите информации, являются так называемые «малые формы физической культуры».

К «малым формам физической культуры» относятся утренняя гимнастика, физкультурная пауза, небольшие паузы в труде специалиста по защите информации с использованием физических упражнений [9].

Наименее сложной является утренняя гимнастика. Однако, это достаточно эффективная форма для быстрого включения специалиста по защите информации в трудовой день. Утренняя гимнастика уменьшает время приведения организма в работоспособное состояние, усиливает ток крови во всех частях тела и учащает дыхание, благодаря чему активизируются обменные процессы организма. Ежедневное выполнение зарядки улучшает кровоток, укрепляет сердечно-сосудистую систему, нервную и дыхательную системы. Утренняя зарядка благотворно влияет на мышечный тонус и состояние суставов и позвоночника.

Эффективной и простой формой физической культуры является физкультурная пауза. Физкультурная пауза призвана решать задачу обеспечения активного отдыха специалиста по защите информации и повышения его работоспособности в перерывах от работы.

Физкультурные паузы важно проводить во время полуденных перерывов для улучшения кровообращения, как в нижних конечностях и органах тазовой области, так и во всем организме, и поддержания тонуса скелетной мускулатуры. Рекомендуется проводить физкультурные паузы продолжительностью 10 минут каждые 4 часа работы и продолжительностью 5 минут - после каждых 2 часов работы.

Небольшие паузы следует делать в течение рабочего дня специалиста по защите информации, во время которых выполняются физические упражнения. Благодаря этим паузам в работе можно снизить усталость, вызванную низкой двигательной активностью скелетной мускулатуры.

Использование физической культуры в труде специалиста по защите информации играет существенную роль в оздоровлении условий его работы и повышении работоспособности.

Основным методом снижения воздействия вредных психофизиологических факторов, влияющих на опорно-двигательный аппарат специалиста по защите информации, является повышение двигательной активности. Двигательная активность необходима человеку для правильного функционирования организма в целом и органов и тканей в частности. Снижение двигательной активности приводит к дегенеративным процессам в организме человека. Наиболее доступным средством увеличения двигательной активности являются занятия физической культурой, спортом. Интенсивность, форма и объем физических упражнений должны подбираться исходя из индивидуальных особенностей работников, таких как физическое развитие, состояние здоровья, подготовленность, психофизического состояния. Целесообразно подобранные формы и средства двигательной активности могут обеспечить снижение воздействия вредных психофизиологических факторов, влияющих на опорно-двигательный аппарат специалиста по защите информации.

Заключение

В ходе выполнения выпускной квалификационной работы были разработаны структурная и принципиальная электрические схемы устройства передачи зашифрованной информации со встроенным аппаратно-программным механизмом защиты памяти от несанкционированного доступа к ключевой информации.

В процессе работы был проведен анализ требований, предъявляемых к разрабатываемым средствам криптографической защиты информации, выполнен обзор существующих аппаратно-программных механизмов защиты памяти от несанкционированного доступа, проведён анализ угроз информационной безопасности применительно к разрабатываемому типу устройств.

В заключении было разработано и протестировано программное обеспечение, реализующее аппаратно-программный механизм защиты памяти от несанкционированного доступа.

Результатом выпускной квалификационной работы являются структурная и принципиальная электрические схемы устройства передачи зашифрованной информации со встроенным аппаратно-программным механизмом защиты памяти от несанкционированного доступа к ключевой информации и программное обеспечение, реализующее аппаратно-программный механизм защиты памяти от несанкционированного доступа.

Таким образом, задание на выпускную квалификационную работу выполнено в полном объеме.

Список используемых источников

1 ГОСТ Р 52459.1-2009. Совместимость технических средств электромагнитная. Технические средства радиосвязи. Часть 1. Общие технические требования и методы испытаний. – Введ. 01-01-2010. – М.: Стандартинформ, 2010. – 46 с.

2 Приказ ФСБ РФ "Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)" от 09.02.2005 № 66 // ФСБ РФ. с изм. и допол. в ред. от 12.04.2010.

3 Аппаратно-программные средства и методы защиты информации: Учебное пособие / Варлатая С.К., Шаханова М.В., Владивосток: ДВГТУ, 2007. – 318 с.

4 User manual. STM32CubeMX for STM32 configuration and initialization C code generation (UM1718). ST Microelectronics, 2017.

5 User manual. Description of STM32L4 HAL and Low-layer drivers (UM1884). ST Microelectronics, 2016.

6 Приказ ФАПСИ "Об утверждении Инструкции об организации и обеспечении безопасности хранения, обработки и передачи по каналам связи с использованием средств криптографической защиты информации с ограниченным доступом, не содержащей сведений, составляющих государственную тайну" от 13.06.2001 № 152 // ФАПСИ РФ.

7 ГОСТ 12.0.003–2015. Система стандартов по безопасности труда. Опасные и вредные производственные факторы. Классификация. – Введ. 01-03-2017. – М.: Стандартинформ, 2016. – 10 с.

8 Социальная гигиена и организация здравоохранения / Кучеренко В.З., Агарков Н.М., Яковлев А.П., Васильев С.Л., Моск. мед. акад. им. И.М. Сеченова, Мед. ин-т Орлов. гос. ун-та. - М.: [б. и.], 2000. - 432 с.

9 Физическая культура работников умственного труда / Виленский М.Я., Ильинич В.И., М.: Знание, 1987 г. – 96 с.

Приложение А

(обязательное)

Схема электрическая принципиальная аппаратного механизма защиты от НСД изделия-450 (устройство-прототип)

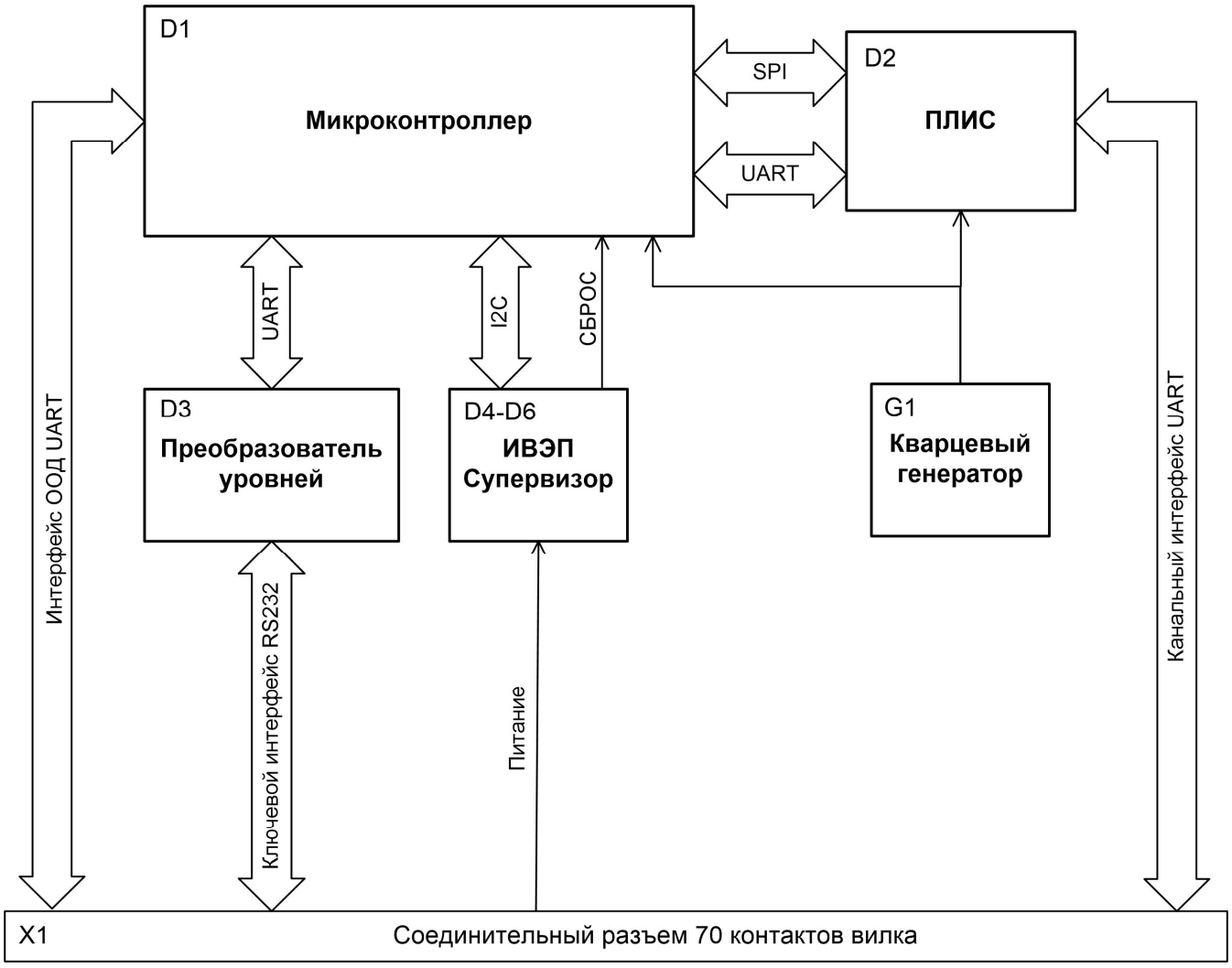
ПГУ.100502.С32.ВКР.12ПТ126.01 Э3

Приложение Б

(обязательное)

Схема электрическая структурная устройства передачи зашифрованной информации

ПГУ.100502.С32.ВКР.12ПТ126.02 Э1



					ПГУ.100502.С32.ВКР.12ПТ126.02 Э1	
Изм	Лист	№ докум.	Подпись	Дата	УПЗИ Схема электрическая структурная	Лит. / Масса / Масштаб
Разраб.		Хабибуллин				
Проверил		Дуплянский				Лист / Листов 1
Т. Контр.						
Н. Контр.		Иванов				
Утвердил						

Перв. примен. | Справочный № | Подп. и дата | Инв. № дубл. | Взам. инв. № | Подп. и дата | Инв. № подл.

Копировал

Формат А4

Приложение В

(обязательное)

Схема электрическая принципиальная устройства передачи зашифрованной информации

ПГУ.100502.С32.ВКР.12ПТ126.03 Э3

Приложение Г

(обязательное)

Перечень элементов

ПГУ.100502.С32.ВКР.12ПТ126.03 ПЭ3

Приложение Д

(обязательное)

Исходный код программы, реализующей аппаратно-программный механизм защиты памяти от несанкционированного доступа

Исходный код файла main.c

```
#include "main.h"
#include "stm32f4xx_hal.h"

UART_HandleTypeDef huart2;
DMA_HandleTypeDef hdma_usart2_rx;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_USART2_UART_Init(void);

void SetRDPLevel(void);
void PrintStat(void);

FLASH_OBProgramInitTypeDef OBInit;
HAL_StatusTypeDef status;
uint8_t str_rx;
uint8_t lvl_1_fl = 0;
uint8_t UART_RxCplt_int_fl = 0;

int main(void)
{

  HAL_Init();

  SystemClock_Config();

  MX_GPIO_Init();
  MX_DMA_Init();
  MX_USART2_UART_Init();


  unsigned long last_Tick = HAL_GetTick();
  unsigned int seconds = 47;

  uint8_t str_hello[] = "Program for Read protection initialize\n\n";
  HAL_UART_Transmit(&huart2, str_hello, sizeof(str_hello), 100);

  // прием команды
  // после получения команды вызывается прерывание
  HAL_UART_Receive_DMA(&huart2, &str_rx, 1);

  while (1)
  {
    // запрос конфигурации байта опций
    HAL_FLASHEx_OBGetConfig(&OBInit);

    if (OBInit.RDPLevel == OB_RDP_LEVEL_0)
    {
      HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_SET); // green led
    }
```

```
    else if (OBInit.RDPLevel == OB_RDP_LEVEL_1)
    {
      HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET); // red led
    }
    else
    {
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14|GPIO_PIN_12, GPIO_PIN_SET);
    }

    if (UART_RxCplt_int_fl)
    {
      UART_RxCplt_int_fl = 0;

      if ((str_rx == '5') || (str_rx == '0'))
      {
        SetRDPLevel();
      }
      else
      {
        PrintStat();
      }
    }

    unsigned long cur_Tick = HAL_GetTick();

    // если прошла секунда
    if (cur_Tick - last_Tick > 1000)
    {
      last_Tick = cur_Tick;
      // Do something
      uint8_t str_wrktm[] = "Time: ";
      HAL_UART_Transmit(&huart2, str_wrktm, sizeof(str_wrktm), 100);
      seconds += 1;
      HAL_UART_Transmit(&huart2, &seconds, 10, 100);
      uint8_t str_nl[] = "\n";
      HAL_UART_Transmit(&huart2, str_nl, sizeof(str_nl), 100);
    }
  }
}

void SetRDPLevel(void)
{
      uint8_t RDP_status;
      uint8_t str_RDP_level[] = "RDP Level...";
      uint8_t str_OB_unlock[] = "OB Unlocking...";
      uint8_t str_FLASH_unlock[] = "FLASH Unlocking...";
      uint8_t str_OB_programming[] = "OB Programming...";
      uint8_t str_OPTSTRT[] = "Option launch (OPTSTRT)...";
      uint8_t str_RDP_level_cmp[] = "RDP Level set comparing...";
      uint8_t str_RDP_level_0[] = "RDP Level 0 already set\n\n";
      uint8_t str_RDP_level_1[] = "RDP Level 1 already set\n\n";

      uint8_t str_status_OK[] = "OK!\n";
      uint8_t str_status_Error[] = "Error!\n";
```

```
      uint8_t str_status_Timeout[] = "Timeout!\n";
      uint8_t str_status_Try[] = "Try...\n";
      uint8_t str_nl[] = "\n";

      // запрос конфигурации байта опций
      HAL_FLASHEx_OBGetConfig(&OBInit);

      // if str_rx == '5' choose level 1 RDP
        if (str_rx == '5')
      {
          if (OBInit.RDPLevel != OB_RDP_LEVEL_1)
          {
            // choose level 1 RDP
            OBInit.RDPLevel = OB_RDP_LEVEL_1;
            RDP_status = OB_RDP_LEVEL_1;
            // set flag level 1
            lvl_1_fl = 1;
          }
          else
          {
            /* Print chosen RDP level */
            HAL_UART_Transmit(&huart2, str_RDP_level_1, sizeof(str_RDP_level_1),
100);
            return;
          }
      }
      // str_rx != '5' choose level 0 RDP
      else
      {
          if (OBInit.RDPLevel != OB_RDP_LEVEL_0)
          {
            // choose level 0 RDP
            OBInit.RDPLevel = OB_RDP_LEVEL_0;
            RDP_status = OB_RDP_LEVEL_0;
          }
          else
          {
            /* Print chosen RDP level */
            HAL_UART_Transmit(&huart2, str_RDP_level_0, sizeof(str_RDP_level_0),
100);
            return;
          }
      }

        /* Print chosen RDP level */
        HAL_UART_Transmit(&huart2, str_RDP_level, sizeof(str_RDP_level), 100);

        if (OBInit.RDPLevel == OB_RDP_LEVEL_0)
        {
          uint8_t chosen_level[] = "Level 0 chosen\n";
          HAL_UART_Transmit(&huart2, chosen_level, sizeof(chosen_level), 100);
        }
        else if (OBInit.RDPLevel == OB_RDP_LEVEL_1)
        {
          uint8_t chosen_level[] = "Level 1 chosen\n";
```

```
          HAL_UART_Transmit(&huart2, chosen_level, sizeof(chosen_level), 100);
        }
        else
        {
          HAL_UART_Transmit(&huart2, str_status_Error, sizeof(str_status_Error),
100);
        }

        /* Allow Access to Flash control registers and user Flash */
        HAL_UART_Transmit(&huart2, str_FLASH_unlock, sizeof(str_FLASH_unlock),
100);

        status = HAL_FLASH_Unlock();

        if (status == HAL_OK)
        {
          HAL_UART_Transmit(&huart2, str_status_OK, sizeof(str_status_OK), 100);
        }
        // HAL_ERROR - FLASH unlock error
        else
        {
          /* Print status */
          HAL_UART_Transmit(&huart2, str_status_Error, sizeof(str_status_Error),
100);
        }

        /* Allow Access to option bytes sector */
        HAL_UART_Transmit(&huart2, str_OB_unlock, sizeof(str_OB_unlock), 100);

        status = HAL_FLASH_OB_Unlock();

        if (status == HAL_OK)
        {
          /* Print status */
          HAL_UART_Transmit(&huart2, str_status_OK, sizeof(str_status_OK), 100);

          /* Set RDP config */
          OBInit.OptionType = OPTIONBYTE_RDP;

          /* Programm OB */
          HAL_UART_Transmit(&huart2, str_OB_programming,
sizeof(str_OB_programming), 100);

          status = HAL_FLASHEx_OBProgram(&OBInit);

          if (status == HAL_OK)
          {
            /* Print status */
            HAL_UART_Transmit(&huart2, str_status_OK, sizeof(str_status_OK),
100);

            /* Start the Option Bytes programming process */
            HAL_UART_Transmit(&huart2, str_OPTSTRT, sizeof(str_OPTSTRT), 100);

            status = HAL_FLASH_OB_Launch();
```

```
            while (status != HAL_OK)
            {
              /* Print status */
              HAL_UART_Transmit(&huart2, str_status_Try, sizeof(str_status_Try),
100);
            }

            if (status == HAL_OK)
            {
              /* Print status */
              HAL_UART_Transmit(&huart2, str_status_OK, sizeof(str_status_OK),
100);
            }

            // HAL_TIMEOUT - flash is busy
            else if (status == HAL_TIMEOUT)
            {
              /* Print status */
              HAL_UART_Transmit(&huart2, str_status_Timeout,
sizeof(str_status_Timeout), 100);
            }
            // HAL_ERROR - OB unlock error
            else
            {
              /* Print status */
              HAL_UART_Transmit(&huart2, str_status_Error,
sizeof(str_status_Error), 100);
            }
          }
          // HAL_TIMEOUT - flash is busy
          else
          {
            /* Print status */
            HAL_UART_Transmit(&huart2, str_status_Timeout,
sizeof(str_status_Timeout), 100);
          }
        }

        // HAL_ERROR - FLASH unlock error
        else
        {
          /* Print status */
            HAL_UART_Transmit(&huart2, str_status_Error,
sizeof(str_status_Error), 100);
        }

      /* Disable the Flash option control register access (recommended to
protect
      the option Bytes against possible unwanted operations) */
    HAL_FLASH_OB_Lock();

      /* Prevent Access to FLASH registers*/
      HAL_FLASH_Lock();
```

```
      /* Get Option Bytes configuration */
      HAL_FLASHEx_OBGetConfig(&OBInit);

      /* Check if RDP level is programmed */
      HAL_UART_Transmit(&huart2, str_RDP_level_cmp, sizeof(str_RDP_level_cmp),
100);

    if (OBInit.RDPLevel == RDP_status)
      {
          /* Print status */
          HAL_UART_Transmit(&huart2, str_status_OK, sizeof(str_status_OK), 100);
          HAL_UART_Transmit(&huart2, str_nl, sizeof(str_nl), 100);
      }
      else
      {
          /* Print status */
          HAL_UART_Transmit(&huart2, str_status_Error, sizeof(str_status_Error),
100);
      }
}

void PrintStat(void)
{
    // уровень 0
    if (OBInit.RDPLevel == OB_RDP_LEVEL_0)
    {
      uint8_t str_report[] = "RDP Level: RDP LEVEL 0\n\n";
      HAL_UART_Transmit(&huart2, str_report, sizeof(str_report), 100);
    }
    // уровень 1
    else if (OBInit.RDPLevel == OB_RDP_LEVEL_1)
    {
      uint8_t str_report[] = "RDP Level: RDP LEVEL 1\n\n";
      HAL_UART_Transmit(&huart2, str_report, sizeof(str_report), 100);
    }
    // иначе светится зеленый и красный
    else
    {
      uint8_t str_report[] = "RDP Level: UNKNOWN LEVEL\n\n";
      HAL_UART_Transmit(&huart2, str_report, sizeof(str_report), 100);
    }
}
```

Приложение Е

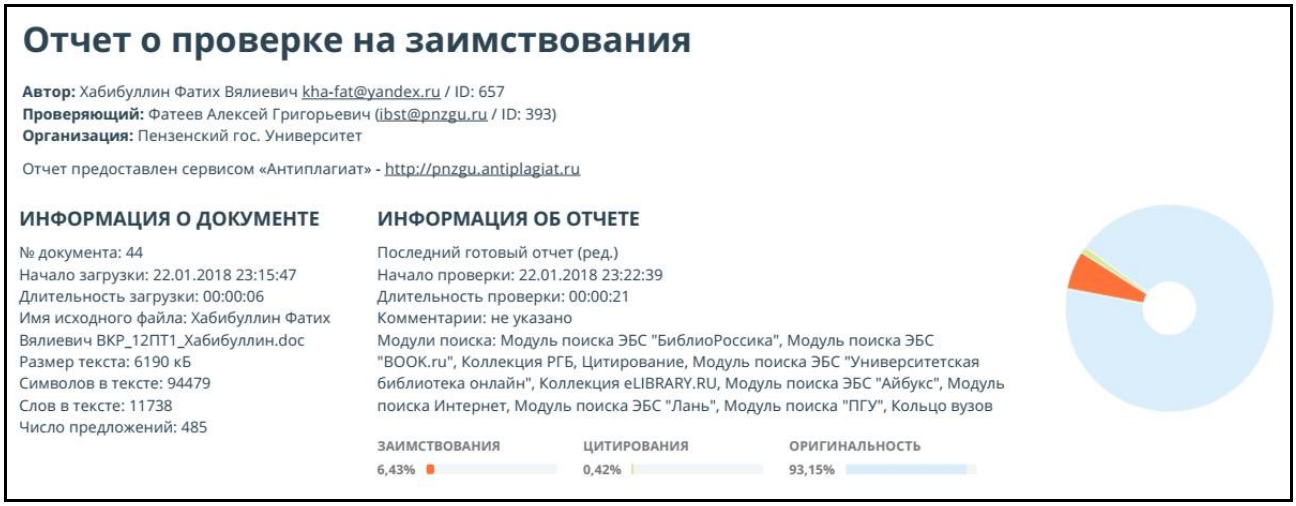
(обязательное)

Протокол проверки ВКР в системе «Антиплагиат»

Протокол

проверки на оригинальность в системе «Антиплагиат.ВУЗ» выпускной квалификационной работы на тему «Устройство передачи зашифрованной информации со встроенным аппаратно-программным механизмом защиты памяти от несанкционированного доступа» студента 6 курса Ф.В. Хабибуллина специальности 10.05.02 – «Информационная безопасность телекоммуникационных систем».

Руководитель ВКР ведущий инженер НИО-05 АО «ПНИЭИ» А.Л. Дуплянский.



Отчет о проверке на заимствования

Автор: Хабибуллин Фатих Вялиевич kha-fat@yandex.ru / ID: 657
Проверяющий: Фатеев Алексей Григорьевич (ibst@pnzgu.ru / ID: 393)
Организация: Пензенский гос. Университет

Отчет предоставлен сервисом «Антиплагиат» - http://pnzgu.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 44
Начало загрузки: 22.01.2018 23:15:47
Длительность загрузки: 00:00:06
Имя исходного файла: Хабибуллин Фатих Вялиевич ВКР_12ПТ1_Хабибуллин.doc
Размер текста: 6190 кБ
Символов в тексте: 94479
Слов в тексте: 11738
Число предложений: 485

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 22.01.2018 23:22:39
Длительность проверки: 00:00:21
Комментарии: не указано
Модули поиска: Модуль поиска ЭБС "БиблиоРоссика", Модуль поиска ЭБС "BOOK.ru", Коллекция РГБ, Цитирование, Модуль поиска ЭБС "Университетская библиотека онлайн", Коллекция eLIBRARY.RU, Модуль поиска ЭБС "Айбукс", Модуль поиска Интернет, Модуль поиска ЭБС "Лань", Модуль поиска "ПГУ", Кольцо вузов

ЗАИМСТВОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
6,43%	0,42%	93,15%

23.01.2018 _____ А.Л. Дуплянский