

**Министерство науки и высшего образования Республики Казахстан**

**Евразийский технологический университет**

Молганов А.А.

**ДИПЛОМНЫЙ ПРОЕКТ**

**на тему: «Разработка аппаратно-программного комплекса для безопасной  
передачи сообщения по оптическому каналу связи»**

Образовательная программа: «6В06106 – Информатика»

**Алматы, 2023 г.**

**Министерство науки и высшего образования Республики Казахстан**

**Евразийский технологический университет**

Молганов А.А.

**ДИПЛОМНЫЙ ПРОЕКТ**

**на тему: «Разработка аппаратно-программного комплекса для безопасной  
передачи сообщения по оптическому каналу связи»**

Образовательная программа: «6B06106 – Информатика»

**Алматы, 2023 г.**

**Министерство науки и высшего образования Республики Казахстан**

**Евразийский технологический университет**

«Допущен(а) к защите»

И.о декана факультета «ИТ»

Полегенько И.Г.

« \_\_\_\_ » \_\_\_\_\_ 2023 г.

## **ДИПЛОМНЫЙ ПРОЕКТ**

На тему: «Разработка аппаратно-программного комплекса для  
безопасной передачи сообщения по оптическому каналу связи»  
по образовательной программе «6В06106 – Информатика»

Выполнил: \_\_\_\_\_ Молганов А.А.  
(подпись)

Научный руководитель: \_\_\_\_\_ Савельева В.В.  
(подпись)

**Алматы, 2023 г.**

**Евразийский технологический университет**  
**Факультет «Информационные технологии»**

**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Студенту Молганову Андрею Александровичу

Обучающимся по ОП 6В06106 – Информатика

Тема дипломного проекта «Разработка аппаратно-программного комплекса для безопасной передачи сообщения по оптическому каналу связи»

Научный руководитель: Доктор PhD, ассоц.профессор Савельева В.В.

Утверждена приказом по университету № 16–19/884-1 от «31» октября 2022г.

Срок сдачи законченного дипломного проекта на кафедре: 09.06.2023 г.

Содержание и объем работы (пояснительной, расчетной и экспериментальной частей, т.е. перечень подлежащих разработке вопросов)

- 1) Анализ предметной области.
- 2) Проектирование аппаратно-программного комплекса.
- 3) Оптимизация криптографического алгоритма.

Рекомендуемая основная литература

1 Mitzner K., «Complete PCB Design Using OrCAD Capture and PCB Editor» – Second Edition. – Elseiver Academic Press, 2019. – 600 p.

2 Митцнер К., Доу Б., Акулин А., Супонин А., Мюллер Д., «Проектирование печатных плат в OrCAD Capture и OrCAD PCB Editor», Второе издание. – Москва: Техносфера, 2022. – 592 с.

3 Труднов А.В., «Высокоскоростные печатные платы. Практические рекомендации», – Москва: Ridero, 2019. – 152 с.

4 Коберниченко В.Г., «Основы цифровой обработки сигналов», – Изд-во Урал. Ун-та, 2018. – 150 с.

5 Пош М., «Программирование встроенных систем на C++ 17» / пер. с англ. А.В. Снастина. – М.: ДМК Пресс, 2020. – 394 с.

6 Беляков С.Л., Боженюк А.В., Петряева М.В., «Основы разработки программы на языке C++ для систем информационной безопасности: учебное пособие». – М.: Издательство Южного Федерального Университета, Ростов-на-Дону, 2020. – 152 с.

Перечень графического материала (с точным указанием обязательных таблиц, чертежей или графиков и т.п) Таблиц – 4, Рисунков – 24.

Дата выдачи задания 16.12.2022

И.о. декан факультета

\_\_\_\_\_ /  
(подпись)

Полегенько И.Г. /

(Ф.И.О.)

Руководитель дипломной работы

\_\_\_\_\_ /  
(подпись)

Савельева В.В. /

(Ф.И.О.)

Задание принял к исполнению студент

\_\_\_\_\_ /  
(подпись)

Молганов А.А. /

(Ф.И.О.)

Дата 14.12.2022

## СОДЕРЖАНИЕ

	ВВЕДЕНИЕ.....	5
1	АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	
1.1	Передача информации в открытом пространстве .....	8
1.2	Описание целей разработки .....	13
1.3	Анализ существующих технологий передачи информации в открытом пространстве .....	16
1.4	Обоснование проектных решений.....	19
2	ПРОЕКТИРОВАНИЕ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА.....	
2.1	Проектирование аппаратного обеспечения комплекса .....	24
2.2	Проектирование программного обеспечения комплекса .....	29
3	ОПТИМИЗАЦИЯ КРИПТОГРАФИЧЕСКОГО АЛГОРИТМА.....	
3.1	Криптографический алгоритм AES .....	31
3.2	Аппаратные ускорители шифрования и дешифрования информации	33
3.3	Оптимизация криптографического алгоритма AES .....	36
	ЗАКЛЮЧЕНИЕ.....	39
	СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	41

## ВВЕДЕНИЕ

Развитие высокотехнологичных отраслей экономики любого государства невозможно представить без комплексного развития системных компетенций, которые связывают между собой целые отрасли и разделы науки, образования, экономики и производства.

Информационно-коммуникационные технологии являются одним из основных факторов системного развития науки, образования и экономики любого государства. Развитие информационно-коммуникационных технологий с появлением телекоммуникационных комплексов создания принципиально новых способов передачи информации невзирая на количество передаваемых данных и условия окружающей среды.

Информационные системы, являясь вычислительным ядром информационно-коммуникационных технологий, отвечают за обработку и хранение различных данных. Особое значение информационные системы приобрели в государственных секторах экономики – структурах и органах безопасности, топливно-энергетических комплексах, финансовых организациях и других объектов, хранящих конфиденциальную информацию. Информационные системы, обрабатывающие конфиденциальную информацию, входят в состав критической информационной инфраструктуры.

Особенность проектирования и обслуживания критической информационной инфраструктуры заключается в соблюдении принципа абсолютной информационной безопасности. Принцип абсолютной информационной безопасности предполагает, что на информационная система имеет максимальный уровень защиты от физического и информационного вектора атаки на объект критической информационной инфраструктуры, поскольку такие объекты хранят информацию и данные, утечка которых за периметр критической информационной инфраструктуры может привести к созданию новых векторов атаки на государственные органы и структуры.

При этом, в вопросе проектирования критической информационной инфраструктуры, особое внимание уделяется аппаратно-программным телекоммуникационным комплексам, имеющим интерфейсы для приема и передачи информации с внешним информационным полем. Современное развитие критической информационной инфраструктуры сочетает в себе использование различных типов телекоммуникационного оборудования – коммутационные и распределительные системы, спутниковые, оптические, волоконно-оптические, беспроводные и высокочастотные комплексы передачи и приема данных.

Особое место в телекоммуникационном оборудовании для критической информационной инфраструктуры занимают аппаратно-программные комплексы для передачи и приема информации по атмосферной оптической линии связи. Атмосферные оптические линии связи, а также комплексы, оборудования и системы в их составе, передают и принимают информацию при помощи электромагнитных волн оптического диапазона, распространяемых

через атмосферу. Атмосферные оптические линии связи, а также комплексы, оборудования и системы в их составе, имеют целый ряд преимуществ:

- высокая скорость передачи, обусловленная отсутствием сигнальной задержки между принятием и отправкой кадров, пакетов и сообщений данных;
- низкая задержка в обработке кадров, пакетов и сообщений данных, обусловленная использованием в конструкции такого оборудования высокоскоростных печатных плат и высокочастотных электронных компонентов;
- безопасность оптического канала, ввиду невозможности перехвата и расшифровки данных отправляемых и передаваемых с помощью инфракрасного оптического излучения;
- свободное лицензирование по сравнению с радиочастотными и СВЧ-системами, для приобретения, установки и проведения пуско-наладочных работ которых требуется специальное разрешение государственного органа имеющего функции лицензирования в области специальных телекоммуникационных комплексов.

Одновременно с этим, атмосферные оптические линии связи, а также комплексы, оборудования и системы в их составе, имеют целый ряд недостатков, критичных для некоторых условий работы в составе критической информационной инфраструктуры:

- высокая чувствительность к плохим погодным условиям ввиду того, что инфракрасный оптический сигнал не имеет эффекта усиления и разреженная атмосфера является дополнительным слоем сигнальных помех при передаче и отправке информации по атмосферным оптическим линиям связи;
- ограниченная дальность атмосферных оптических линий связи, обусловленная наличием оптических линз в передающем и принимающей оптическом терминале;
- влияние окружающей местности на характеристики сигнала, передаваемого по атмосферным оптическим линиям связи ввиду отражения сигнала от различных плоскостей.

Целью данной работы является проектирование, изготовление и отладка атмосферной оптической линии связи с использованием оптимизированных для микроконтроллерной архитектуры алгоритмов легковесной криптографии. Для достижения данной цели необходимо решить следующие задачи:

- исследовать принципы передачи информации с использованием атмосферных оптических линий связи;
- исследовать принципы проектирования высокоскоростных печатных плат и высокочастотных элементов с использованием средств автоматизированного проектирования;
- исследовать принципы проектирования программного кода для микроконтроллерных платформ с поддержкой аппаратных криптографических алгоритмов;

- оптимизировать криптографический алгоритм для микроконтроллерной платформы, поддерживающей аппаратное ускорения криптографических алгоритмов;

- создать стенд для реализации принципа передачи информации с помощью атмосферной оптической линии связи с поддержкой ускорения криптографических операций для шифрования информации передаваемой по инфракрасному оптическому каналу связи.



# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Передача информации в открытом пространстве

Передача информации в телекоммуникационных системах зависит от типа используемой системы, топологии вычислительной сети, зоны действия и количества пользователей подключенных к вычислительной сети с помощью телекоммуникационных систем и комплексов. Телекоммуникационные системы по типу передачи сигналов делятся на несколько видов, и включают в себя следующие виды сигналов [15]:

- аналоговый сигнал – телекоммуникационные системы, передающие информацию с помощью аналогового сигнала, имеют фундаментальные недостатки в виде невозможности шифрования передаваемой информации в потоковой форме, при этом такие системы имеют простую конструкцию;

- цифровой сигнал – телекоммуникационные системы, передающие информацию с помощью цифрового сигнала, имеют большую пропускную способность за счет использования бинарной логики при отправке и приеме сигнала, при этом в конструкции таких систем предполагается использование различного питания постоянного тока ввиду необходимости дифференциации различных сигналов;

- оптический сигнал – телекоммуникационные системы, передающие информацию с помощью оптического сигнала, имеют большую пропускную способность и сочетают в себе все преимущества аналоговой и цифровой передачи сигналов на дальнее расстояние, при этом такие системы в своей конструкции используют дорогостоящие оптические системы и линзы для корректировки и автоматической наводки оптического сигнала на устройство-приемник;

- беспроводной сигнал – телекоммуникационные системы, передающие информацию с помощью беспроводного сигнала, имеют в своем составе радиочастотный трансивер которые позволяет передавать сигнал на большее расстояние, и в зависимости от ширины канала передачи данных, радиочастотный трансивер имеет большую частоту и амплитуду сигнала, что влечет за собой дополнительное тепловыделение и генерацию постороннего шума.

Атмосферные оптические линии связи (на англ. – Free Space Optics) в своей архитектуре передачи и приема информации используют оптический инфракрасный сигнал в качестве канала связи с использованием оптических линз и систем для корректировки и автоматической наводки конечной либо начальной точки передач и приема информации. Аппаратно-программные комплексы, использующиеся для передачи и приема информации по атмосферным оптическим линиям связи имеют иное строение ввиду необходимости принимать, генерировать и корректировать инфракрасный оптический сигнал с использованием полупроводниковых лазерных диодов (рисунок 1).



Рисунок 1 – Оптический инфракрасный трансивер, используемый для передачи и приема информации по атмосферной оптической линии связи

Аппаратно-программные комплексы для передачи информации по атмосферным оптическим линиям связи имеют специальную конструкцию составных элементов для корректной передачи и отправки сигнала по инфракрасному оптическому сигналу. Структурная схема передачи информации по атмосферным оптическим линиям связи между аппаратно-программными комплексами имеет следующий вид (рисунок 2):

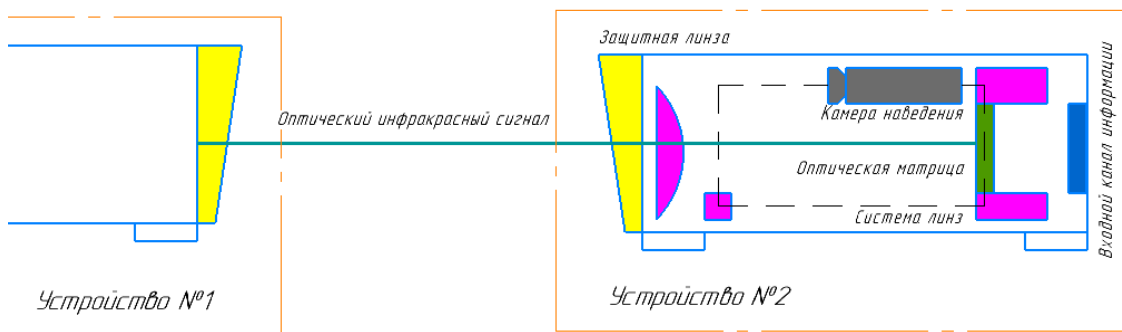


Рисунок 2 – Структурная схема устройства №1 (передатчик) и устройства №2 (приемник) для передачи и приема сообщения по атмосферным оптическим линиям связи

Аппаратно-программные комплексы, использующиеся для передачи и приема информации по атмосферным оптическим линиям связи в своей конструкции, имеют специализированные элементы, которые прямо влияют на конечные характеристики передачи и приема информации:

- защитная линза – набор диоптрических линз образующих единую оптическую систему для уменьшения искажения передаваемого и принимаемого

сигнала в автоматическом режиме без использования аппаратного или программного обеспечения комплекса;

- камера наведения – самостоятельное устройство использующее в своем составе цифровую камеру и датчик расстояния необходимый для автоматического наведения передающего устройства в область видимости принимающего устройства для создания канала передачи данных большой разрядности;

- система линз – набор прозрачных и угловых линз образующих единую оптическую систему для компактного преобразования инфракрасного оптического сигнала внутри комплекса и передачи принятого сигнала с датчиков на оптическую матрицу;

- оптическая матрица – интегральная микросхема с оптическим выходом, на входе принимающая инфракрасный оптический сигнал собранный с помощью системы линз;

- входной канал информации – блок печатных плат и электронных дискретно-аналоговых компонентов преобразующих инфракрасный оптический сигнал собранный с помощью оптической матрицы в последовательный цифровой сигнал.

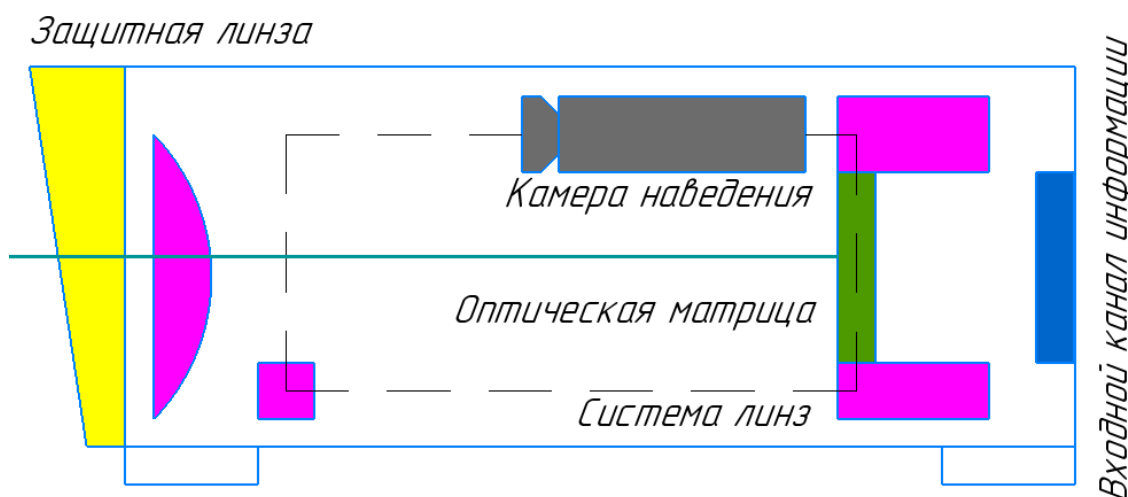


Рисунок 3 – Конструкция устройства для передачи и приема сообщения по атмосферным оптическим линиям связи

Инфракрасные лучи используемые в источниках генерации инфракрасного оптического сигнала имеют длину волны от 700 нм до 1 мм. При этом, в конструкции таких устройств обязательно используется система линз для корректировки, фокусировки и устранения эффектов оптической дифракции в процессе принятия и отправки информации. Инфракрасные лучи передают информацию в зоне прямой видимости приемника и передатчика без огибания препятствий в виде зданий, лесных массивов и других объектов которые могут служить в качестве препятствия для распространения инфракрасного оптического сигнала.

Во время передачи сигнала с помощью инфракрасного луча, используется сигнальная модуляция генерируемого сигнала, при этом модуляция изменяет сигнал, увеличивая или уменьшая его амплитуду и фазовую частоту в зависимости от массива данных поступающих на входной канал информации. Для приема сигнала используется специальное устройство – фотодиод, суть которого заключается в преобразовании оптического сигнала в последовательность бинарного цифрового детерминированного сигнала на основе полупроводникового эффекта, при котором между частотой сигнала и напряжением на выходе формируется прямая пропорциональность для формирования логического сигнала в битовой последовательности [15].

Модуляция сигнала передаваемого с помощью инфракрасных лучей применяется для уплотнения сигнала и соответственно увеличения количества передаваемой информации по инфракрасному оптическому каналу связи. В зависимости от типа сигнала, применяются следующие типы модуляции (рисунок 4).

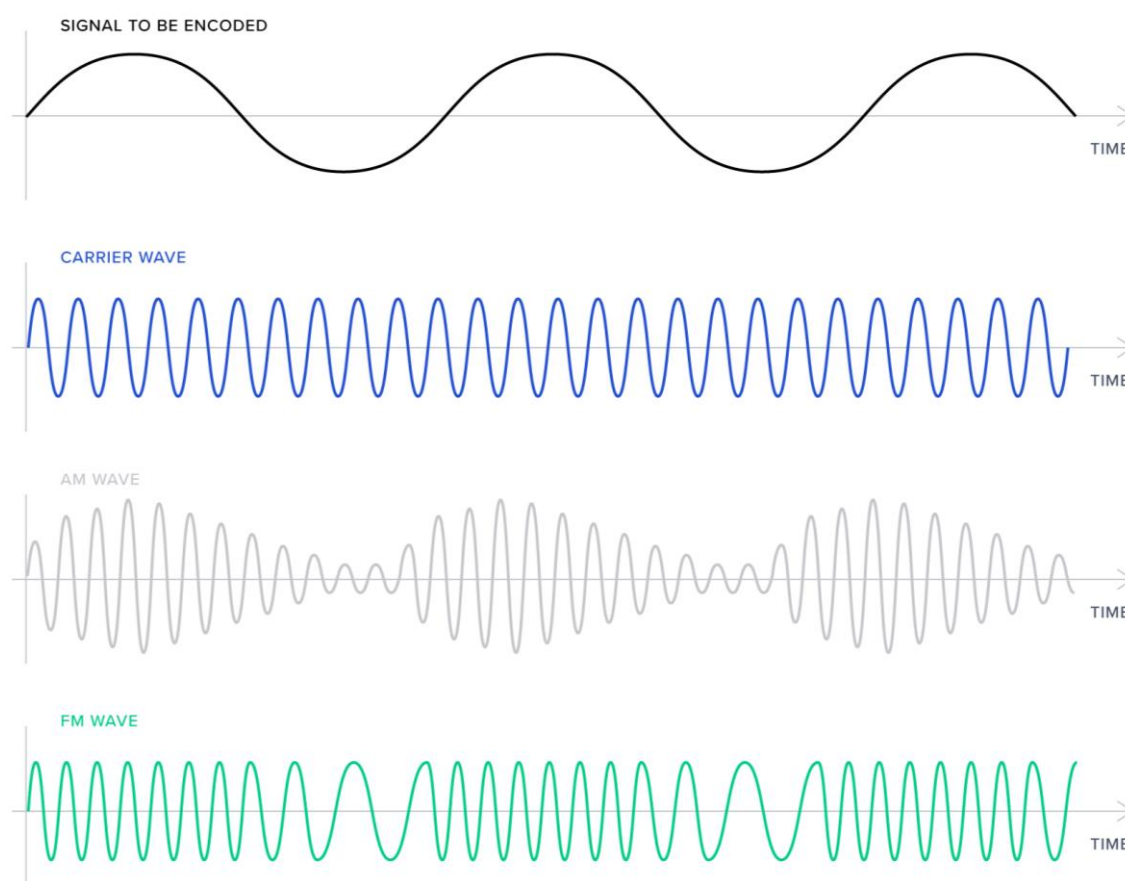


Рисунок 4 – Изменение сигнала с помощью различных типов модуляции

– амплитудная модуляция является самым простым типом модуляцией сигнала с точки зрения аппаратной и программной реализации. Метод амплитудной модуляции заключается в изменении амплитуды сигнала для передачи информации и уплотнения информации передаваемой по инфракрасному оптическому каналу связи. В качестве достоинств можно

отметить простоту реализации алгоритма амплитудной модуляции сигнала, относительную дешевизну оборудования и хорошее качество модуляции низкочастотных сигналов. При этом, амплитудная модуляция не лишена недостатков, а именно – модуляция сигнала по амплитуде имеет низкую помехоустойчивость и эффективность использования канала данных, при этом уплотнение информации передаваемой по каналу данных, имеет прямую пропорциональность с дальностью распространения этого сигнала;

- частотная модуляция изменяет сигнала с помощью изменения частоты и настройки тактирования передаваемого сигнала с помощью цифровых устройств и высокоскоростных протоколов передачи информации. Достоинства частотной модуляции заключаются в хорошем качестве передаваемого сигнала при высокой помехоустойчивости канала данных вкупе с эффективным использованием канала передачи данных. При этом, частотная модуляция не всегда достигается эффективным способом из-за необходимости интегрирования сигнала по частоте. Кроме этого, оборудование использующее частотную модуляцию имеет большую сложность в разработке, проектировании и обслуживании;

- импульсно-кодовая модуляция изменяет сигнал с помощью создания последовательности импульсов для передачи информации и внедрения кодовой информации в передаваемый сигнал. Импульсно-кодовая модуляция применяется в устройствах, где имеется ограничение по ширине канала данных, ввиду того что импульсно-кодовая модуляция имеет высокую скорость передачи данных и помехоустойчивость, при этом сложность реализации такого принципа модуляция ложится на инженеров аппаратного обеспечения ввиду необходимости использования высокочастотных элементов.

## 1.2 Описание целей разработки

Цель разработки комплекса для безопасной передачи сообщения по инфракрасному оптическому каналу связи заключается в проектировании аппаратно-программного комплекса, главной задачей которого является передача и прием зашифрованной информации передаваемой по инфракрасному каналу связи.

Для обеспечения надежной передачи информации по инфракрасному оптическому каналу связи необходимо обеспечить нужный уровень защиты передаваемой информации с помощью криптографических алгоритмов шифрования информации, оптимизированных для использования с микроконтроллерной платформой обладающей низкой производительностью по сравнению с персональными и одноплатными компьютерами общего назначения (рисунок 5).



Рисунок 5 – Цели разработки комплекса

Для передачи и приема инфракрасного оптического сигнала используются инфракрасный диод, генерирующий волну с определенной частотой, амплитудой и импульсом и передающий зашифрованную информацию в виде инфракрасного сигнала в сторону устройства-приемника, где расположен фотодиод в качестве принимающего оптического сенсора.

Одной из главных целей, поставленных в проекте, является обеспечение передачи зашифрованной информации благодаря оптимизации криптографических алгоритмов и использованию аппаратных и аппаратно-программных механизмов защиты информации. Набор возможностей аппаратных и аппаратно-программных механизмов защиты информации базируется на использовании микроконтроллера общего назначения с использованием, встроенного в микроконтроллер криптографического сопроцессора ускоряющего процесс шифрования и расшифровки информации, передаваемой по инфракрасному оптическому каналу [16].

Оптимизация криптографических алгоритмов осуществляется исходя из возможностей и особенностей программирования микроконтроллеров. Архитектура микроконтроллера предполагает разделение памяти на два типа:

- оперативное запоминающее устройство (сокр. ОЗУ) – память такого типа сохраняет в себя промежуточные значения и после совершения операции

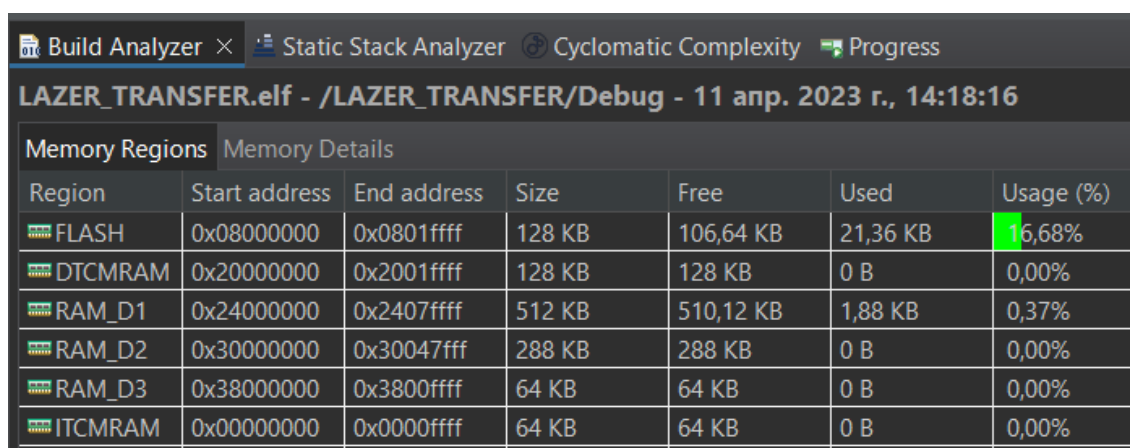


сохранения и выгрузки информации в арифметико-логическое устройство, стирает все данные, которые были сохранены в оперативное запоминающее устройство;

- постоянное запоминающее устройство (сокр. ПЗУ) – память такого типа сохраняет в себя только значения, заранее определенные в программном коде в виде программных констант.

Использование программных констант вкупе с разделением постоянного и оперативного запоминающего устройства позволяет реализовать принцип разбиения шифротекста на предварительно сконфигурированные блоки данных, упорядоченные по количеству битов данных исходя из типа данных, передаваемых по инфракрасному оптическому каналу связи [24].

Блоки в оперативном и постоянном запоминающем устройстве разделены на четное количество ячеек соединенных между собой меж-интегральными соединениями внутри оперативного и постоянного запоминающего устройства, при этом выполняется принцип модульности и независимости изменяемых и постоянных данных в памяти микроконтроллера (рисунок 6).



LAZER_TRANSFER.elf - /LAZER_TRANSFER/Debug - 11 апр. 2023 г., 14:18:16						
Memory Regions		Memory Details				
Region	Start address	End address	Size	Free	Used	Usage (%)
FLASH	0x08000000	0x0801ffff	128 KB	106,64 KB	21,36 KB	16,68%
DTCMRAM	0x20000000	0x2001ffff	128 KB	128 KB	0 B	0,00%
RAM_D1	0x24000000	0x2407ffff	512 KB	510,12 KB	1,88 KB	0,37%
RAM_D2	0x30000000	0x30047fff	288 KB	288 KB	0 B	0,00%
RAM_D3	0x38000000	0x3800ffff	64 KB	64 KB	0 B	0,00%
ITCMRAM	0x00000000	0x0000ffff	64 KB	64 KB	0 B	0,00%

Рисунок 6 – Карта памяти микроконтроллера

Программный код, осуществляющий управление микроконтроллером, операциями шифрования, расшифровки, а также передачи и приема информации по инфракрасному оптическому каналу связи, расположен в оперативном запоминающем устройстве микроконтроллера [25].

Данные шифротекста, переменные и постоянные значения которые изменяются в процессе шифрования и расшифровки информации хранятся внутри блоков постоянного запоминающего устройства для большей безопасности, ввиду того что ядра микроконтроллера при реализации аппаратных атак не имеют прямой доступ к блокам постоянного запоминающего устройства, и напротив данные после каждой передачи и приема сообщения сохраняющиеся в память оперативного запоминающего устройства автоматически обнуляются для экономии места и реализации метода абсолютной безопасности.

### 1.3 Анализ существующих технологий передачи информации в открытом пространстве

Комплексное и системное проектирование аппаратно-программного комплекса для безопасной передачи информации по инфракрасному оптическому каналу связи невозможно провести без анализа существующих технологий в области передачи информации с помощью атмосферных оптических линий связи (рисунок 7).

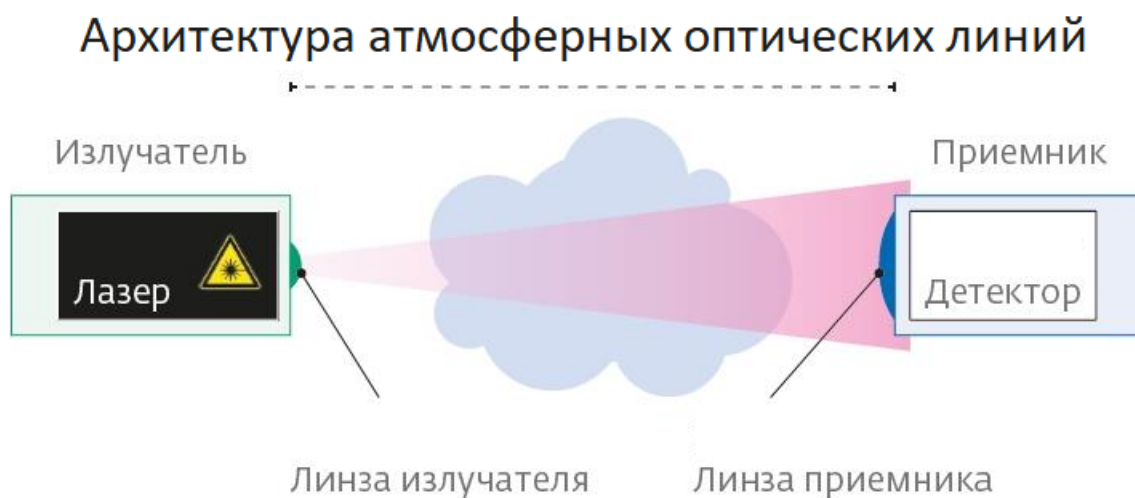


Рисунок 7 – Архитектура передачи данных по атмосферным оптическим линиям связи

Телекоммуникационное оборудование для передачи информации по атмосферным оптическим линиям связи классифицируется по типу сигнала, передаваемого с помощью трансиверов и ресиверов расположенных непосредственно на участках передачи и приема информации в открытом пространстве (рисунок 8).

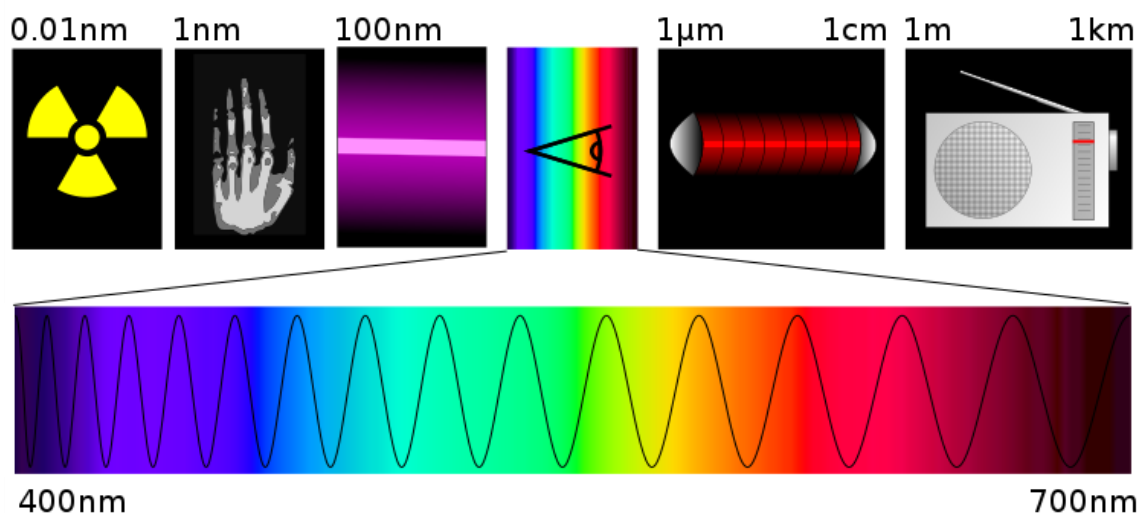


Рисунок 8 – Природа сигнала в зависимости от длины волны



Исходя из этих условий, можно классифицировать следующие технологии по типу сигнала, применяемого для передачи информации в открытом пространстве (таблица 1).

Таблица 1 – Сравнение типов передачи информации по типу сигнала

№	Тип	Описание	Достоинства	Недостатки	Пример
1	Радиоволны	Самый распространенный способ передачи информации в открытом пространстве, использующий радиоволны и частотно-амплитудную модуляцию, генерируемую с помощью радиочастотных модулей связи	Преимущества, данного метода передачи сигнала заключается в относительной дешевизне и простоте проектирования устройств, содержащих радиочастотные модули низкой частоты	В качестве недостатков выделяют сложность и дороговизну точек распространения и приема сигналов, а также базовых станций необходимых для создания единой телекоммуникационной сети	Wi-Fi, Bluetooth
2	Инфракрасное излучение	Самый распространенный способ передачи информации для малопроизводительных устройств и систем. В качестве опорного сигнала, передающего информацию, используется инфракрасное излучение, невидимое для человеческого глаза с длиной волны не более 450 нанометров	Использование простых технологий производства и относительно малой ширины канала позволят использовать инфракрасное излучение в качестве телекоммуникационной системы в компактных устройствах для сетей Peer-to-Peer	Природа сигнала, передаваемого по инфракрасному каналу связи не позволяет каналу связи проходить или отражаться от физических объектов и требует наличия прямой видимости между передающим и принимающим устройствами	IrDA, Giga-IR

3	Ультразвук	самый сложный способ передачи информации. В конструкции таких телекоммуникационных систем применяется модуляция сигнала по типу его природы – аналогово-цифровой преобразователь преобразует разницу между децибелами и напряжением в последовательность битов данных, соединенных между собой кодами проверки для лучшего качества передаваемого сигнала	Ультразвуковое излучение является эффективным и надежным средством передачи информации, благодаря высокой точности и надежности передачи, отсутствию электромагнитных помех	В качестве недостатков можно отметить дорогостоящие составные компоненты и сложность обслуживания таких телекоммуникационных систем	Data-over-Sound (DoS)
---	------------	---	---	---	-----------------------

Использование инфракрасного оптического излучения в качестве опорной точки для передачи сигнала между принимающим и передающим устройством является единственно верным способом гарантированно и без потерь передать предварительно зашифрованную информацию с помощью оптимизированного криптографического алгоритма для использования в микроконтроллерах общего назначения.

Кроме этого, всегда следует помнить, что методы и средства осуществления угроз безопасности информации постоянно совершенствуются. С развитием систем обработки и передачи данных появляются новые виды угроз. Поэтому постоянное развитие претерпевают и аппаратные системы защиты информации. Наряду с этим источники угроз ведут систематический поиск новых уязвимостей в существующих пользовательских приложениях, информационных службах и системах защиты, поэтому использование инфракрасного оптического излучения является единственно верным конструктивным решением для передачи зашифрованной информации.

## 1.4 Обоснование проектных решений

Для того, чтобы разрабатывать аппаратно-программный комплекс, использующийся для передачи и приема информации по атмосферным оптическим линиям связи с поддержкой шифрования информации, необходимо сначала определиться с классификацией встраиваемых систем на основе используемой микропроцессорной или микроконтроллерной платформы (таблица 2)

Таблица 2 – Классификация встраиваемых систем

№	Класс	Применение	Изображение
1	Встраиваемые системы на основе FPGA (Программируемые Логические Интегральные Схемы)	Системы аппаратного искусственного интеллекта Системы анализа телекоммуникационного трафика Комплексы аппаратной защиты информации Системы сбора и анализа данных	
2	Встраиваемые системы на основе микропроцессоров (ARMv9, RISC-V)	Системы управления мехатронными и робототехническими системами с искусственным интеллектом Системы управления энергетическими установками малого и среднего класса Системы сбора данных с помощью датчиков	
3	Встраиваемые системы на основе микроконтроллеров (ARM, RISC-V)	Обучающие системы и комплексы Системы сбора данных Системы управления электродвигателями малого класса	

Разработка встраиваемых приложений для «MPS-2.1» осуществляется с помощью интегрированной среды разработки STM32CubeIDE (рисунок 9) компании STMicroelectronics с бесплатной лицензией [24].



Рисунок 9 – Основные этапы разработки приложения для встраиваемых систем

В составе STM32CubeIDE есть также специальное программное обеспечение STM32CubeMX, служащее для упрощения программирования и проведения первоначальной настройки микроконтроллера. Основным преимуществом данной среды разработки является то, что благодаря специальному хранилищу, при подключенном интернете, пользователь имеет доступ ко множеству библиотек и примеров. Основное пространство окна занимает текстовый редактор кода, в нем пользователь набирает и редактирует свой программный код. В текстовом редакторе присутствует указание на ошибки, если код набран неверно, появляется надпись об этом. Также имеется очень удобная система авто дополнения, которая, при написании некоторой последовательности символов, предлагает разработчику дополнить текст чтобы получить необходимые функции в программировании [18].

STM32CubeIDE сочетает в себе несколько утилит, необходимых для правильной первоначальной настройки и программирования микроконтроллеров STM32, а именно:

- STM32CubeMX – проводит первоначальную настройку микроконтроллера и дает возможность использовать внешние и внутренние интерфейсы микроконтроллера.
- STM32CubeProgrammer – проводит глубокую настройку микроконтроллера, а также позволяет в реальном времени следить за основными характеристиками микроконтроллера.

– STM32CubeMonitor – проводит анализ всей встраиваемой системы с помощью внешних и внутренних датчиков.

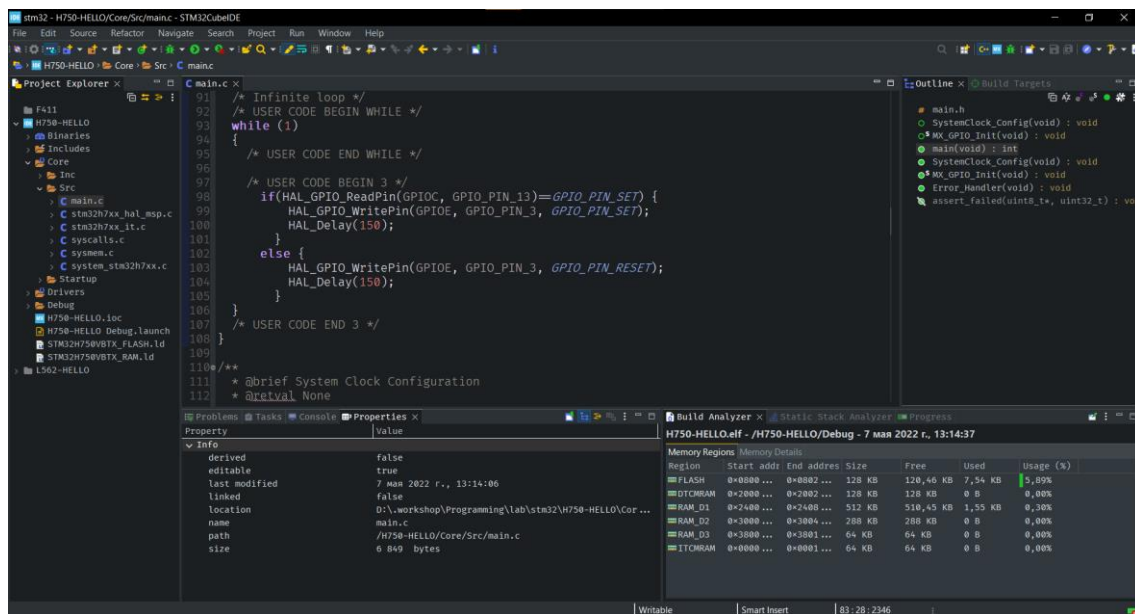


Рисунок 10 – Разработка и отладка программного кода встраиваемых систем на основе микроконтроллера ARM промышленного стандарта с использованием IDE STM32CubeIDE

На сегодняшний день, любое современное электротехническое устройство или изделие содержит в себе многослойную печатную плату. В зависимости от роли устройства или изделия, компоненты на печатной плате могут различаться по типу, пакету, форм-фактору, расположению и монтажу. Системы автоматизированного проектирования для проектирования электронных устройств, печатных плат и микросхем имеют свои различия и сходства. Главная задача систем автоматизированного проектирования заключается в изначально правильном проектировании электронных устройств и изделий для последующей передачи конструкторско-чертежной документации на производство и организации процесса производства [10].

Проектирование, разработка и создание устройств производится в Cadence Allegro 2022. Данная САПР обладает принципом модульности и взаимозаменяемости, поэтому системное проектирование устройства производится в следующих программах входящих в САПР Cadence Allegro:

– Allegro Design CIS – программа для проектирования электрических принципиальных схем и создания электрических цепей с использованием компонентов доступных в интегрированных библиотеках.

– Allegro PCB Designer – программа для проектирования печатной платы на основе уже созданной виртуальной модели устройства созданной на основе электрических цепей с использованием компонентов доступных в интегрированных библиотеках (рисунок 11).

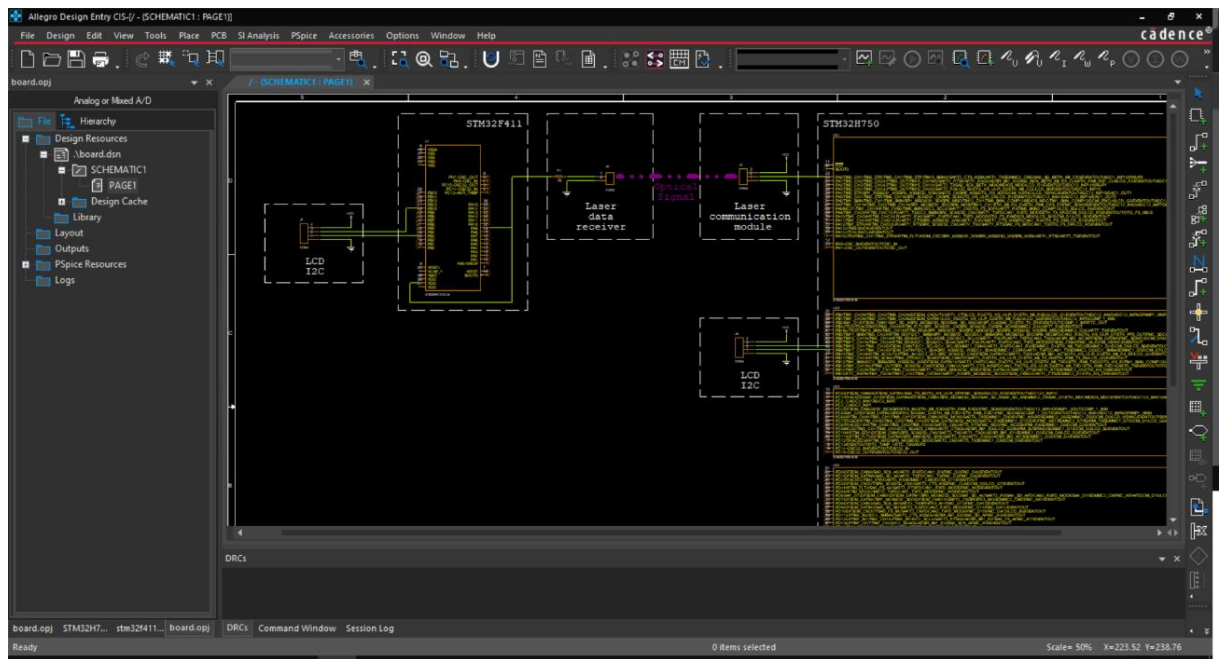


Рисунок 11 – Электрическая схема устройства спроектированная в Cadence Allegro Design CIS

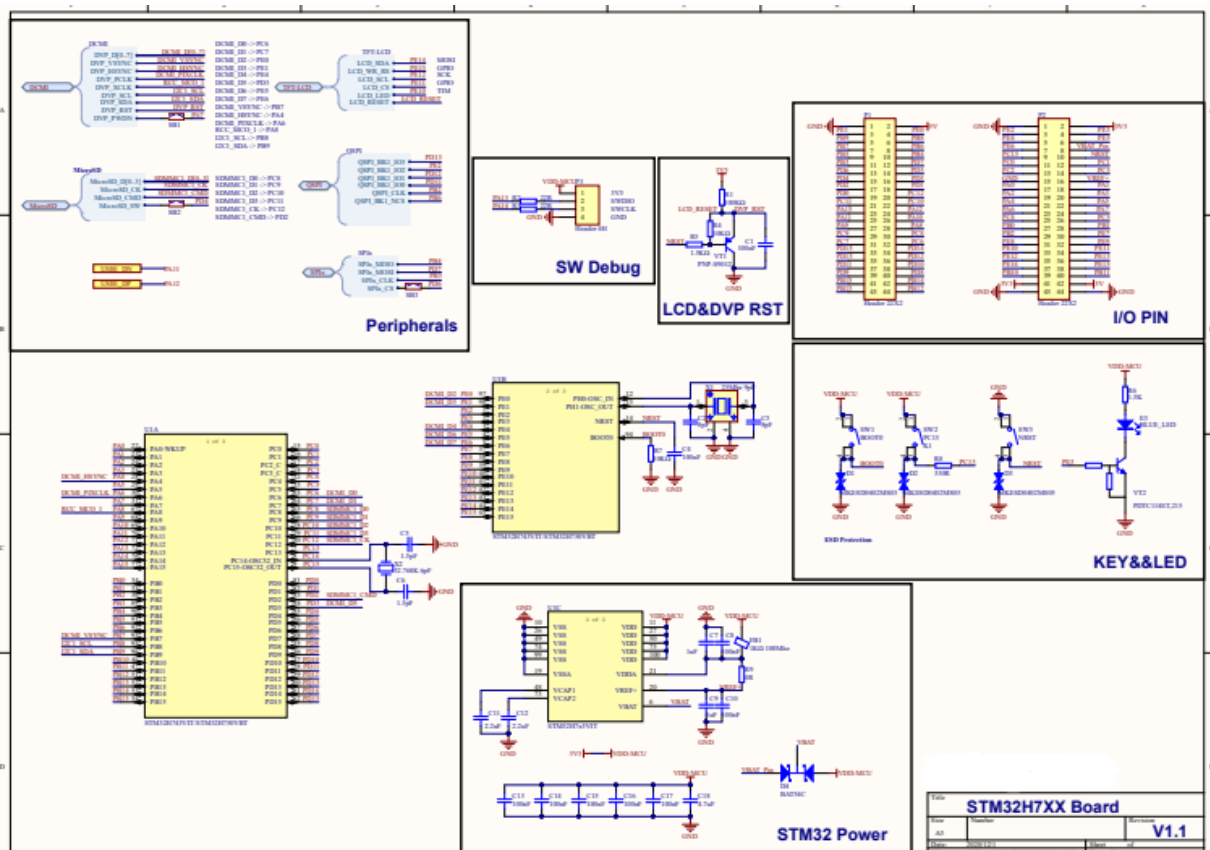


Рисунок 12 – Чертежно-конструкторская документация устройства, спроектированного в Cadence Allegro Design CIS

Таблица 3 – Список компонентов для устройства №1 и №2

№	Имя	Количество	Шелкография	Корпус
1	CON2	7	CON2n	Header_Pin_2.54mm
2	R, 10K	1	R1	1206_SMD
3	U1	1	STM32F411CEU6	QFN50
4	U2	1	STM32H750VBT6	QFN100
5	KY-008	1	LDR	Pin_2.54mm
6	PHRES	1	PHRES	Pin_2.54mm

Использование комплексной среды проектирования печатных плат и электронных компонентов Cadence Allegro позволяет проектировать печатные платы высокой топологии с использованием сигналов разной природы, характеристик, частоты и модуляции. Благодаря использованию принципа модульности и взаимозаменяемости, среда проектирования Cadence Allegro успешно зарекомендовала себя в области проектирования высокоскоростных и высокочастотных печатных плат и электронных периферийных компонентов для печатных плат и электронных устройств [9].

Проектирование программного обеспечения производится с помощью специальной интегрированной среды программирования для микроконтроллерных платформ – STM32CubeIDE с использованием языка программирования C++ и специальных программных директив микроконтроллера. Отладка программного кода и устранение неисправностей производится с помощью встроенного программного отладчика и профилирования программного кода для микроконтроллеров [8].



## 2 ПРОЕКТИРОВАНИЕ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА

### 2.1 Проектирование аппаратного обеспечения комплекса

Начиная с конца 1950-х годов, крупные промышленные компании, специализирующиеся на производстве специальных инженерных или транспортных систем, например аэрокосмическая компания Boeing или электротехнические компании Westinghouse Electric Company, Toshiba задумались о создании компактных встраиваемых системах в своих продуктах. Впервые такие встраиваемые системы появились в конце 1960-х годов с развитием космической программы США и СССР, где первые встраиваемые системы управляли множеством подсистем космических ракет-носителей и даже помогали людям дистанционно исследовать планеты Марс и Венера с помощью автоматических межпланетных станций [6].

Встраиваемые системы – информационно-коммуникационная система, состоящая из двух групп компонентов – аппаратного и программного обеспечения, чаще всего такие системы разработаны специально для выполнения конкретного прикладного или фундаментального применения. Например, блок управления навигацией и ориентированием самолета Airbus A350 (рисунок 13).



Рисунок 13 – Блок управления навигацией и ориентированием самолета Airbus A350



Область применения встраиваемых систем не ограничивается только массивными самолетами и кораблями, встраиваемые системы являясь одной из областей компьютерных наук, постоянно развивается, находя все новые и новые сферы применения (рисунок 14). На сегодняшний день, встраиваемые системы используются в следующих сферах:

- Разработка измерительного оборудования (осциллограф, анализатор сигналов, спектрометр, импульсные блоки питания).
- Разработка бортовых управляющих систем (колесный автотранспорт, железнодорожный транспорт, аэрокосмический и водный транспорт).
- Разработка медицинского оборудования (высокоточное хирургическое, лабораторное, терапевтическое, диагностическое, физиотерапевтическое оборудование).
- Разработка телекоммуникационных станций (базовые станции различных телекоммуникационных стандартов, например 4G, 5G, GPS, LoRaWAN).
- Разработка мехатронных и робототехнических станций (станки с числовым программным управлением, автомобили с автономным управлением, высокоточные 3D-принтеры).
- Разработка системы безопасности и сигнализации (блоки управления видеонаблюдением, доступом и полноценные системы управления доступом).



Рисунок 14 – Встраиваемая система на основе микроконтроллера ARM промышленного стандарта

Проектирование аппаратного обеспечения выполняется в системах автоматизированного проектирования специального назначения (на англ. Electronic design automation) для размещения, трассировки и соединения электронных устройств, компонентов и сигнальных дорожек.

Системы автоматизированного проектирования, предназначенные для проектирования электронных устройств, печатных плат и микросхем имеют свои

различия и сходства. Главная задача таких систем автоматизированного проектирования заключается в изначально правильном проектировании электронных устройств и изделий для последующей передачи конструкторско-чертежной документации на производство и организации процесса промышленного производства спроектированного устройства [5].

Системы автоматизированного проектирования, предназначенные для проектирования электронных устройств, печатных плат и микросхем различаются по имеющемуся функционалу, лицензирования и наличию интегрированной среды математического и физического моделирования, что бывает полезно при проектировании комплексных устройств и систем высокой топологии (таблица 4).

Таблица 4 – Сравнение САПР для проектирования печатных плат

№	Класс	САПР	Описание
1	Любительские	KiCAD	Бесплатная САПР для разработки электронных схем и печатных плат легкой и средней сложности без возможности моделирования
		Eagle CAD	Платная САПР для разработки электронных схем и печатных плат легкой и средней сложности без возможности моделирования
2	Полупрофессиональные	Altium Designer	Платная САПР для проектирования электронных схем и печатных плат средней и высокой сложности с удобным интерфейсом и большим набором функций
		P-CAD	Платная САПР для проектирования электронных схем и печатных плат высокой сложности, предназначенная для работы с многослойными платами
		Proteus	Платная САПР для проектирования электронных схем и симуляции работы устройств
3	Профессиональные	Mentor Graphics PADS	Платная САПР для проектирования электронных схем и печатных плат, предназначенная для работы с многослойными платами и высокочастотными сигналами
		Cadence Allegro	Платная САПР для проектирования электронных схем и печатных плат, предназначенная для работы с многослойными платами и высокочастотными сигналами.
		ANSYS Electronics Suite	Платная САПР для проектирования электронных схем и моделирования работы устройств в условиях высоких частот и электромагнитных помех

Дальнейшее проектирование устройства, печатной платы, а также трассировка дорожек и размещения компонентов при помощи программного комплекса САПР от компании Cadence Design System, USA:

- Среда разработки и проектирования печатной платы – Cadence Allegro 2022 (рисунок 15).
- Среда анализа и моделирования печатной платы – Cadence Sigrity 2022 (рисунок 16).

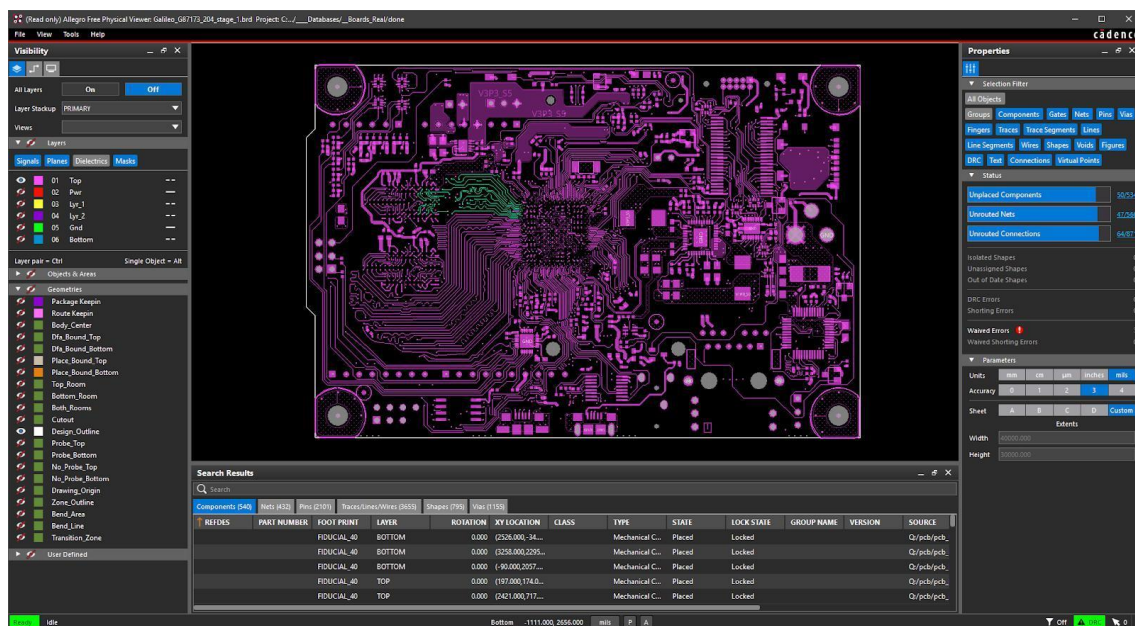


Рисунок 15 – Внешний вид Cadence Allegro PCB Designer 2022

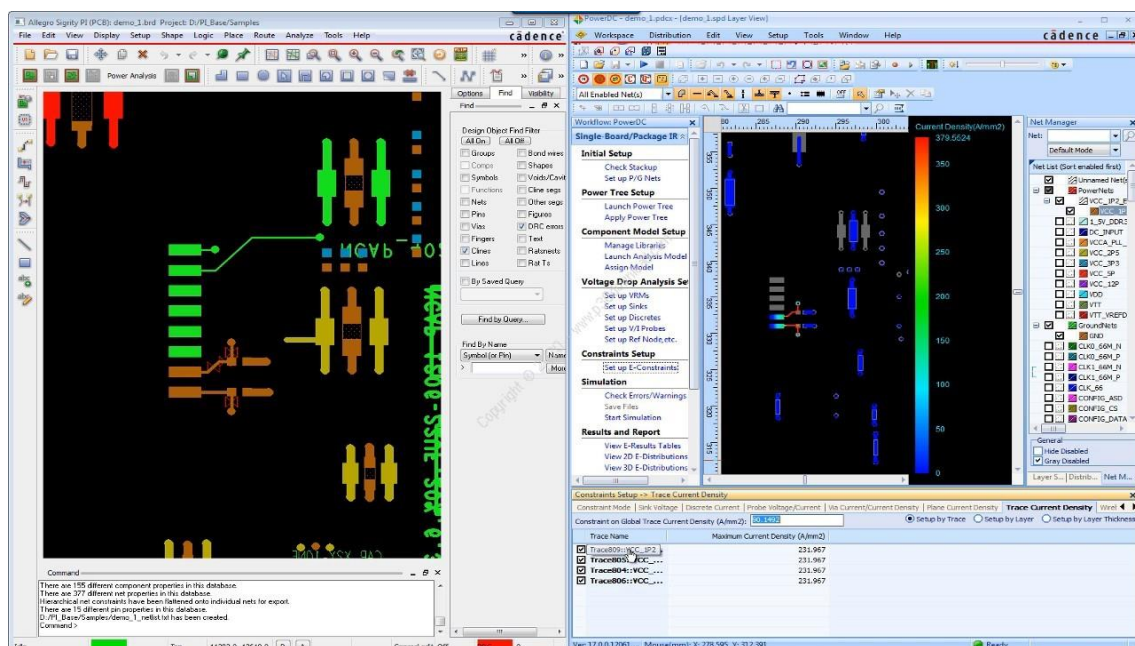


Рисунок 16 – Внешний вид Cadence Sigrity 2019

Cadence Allegro 2022 – профессиональная САПР для проектирования печатных плат и узлов высокой сложности и топологии, включает в себя программы для проектирования электрических принципиальных схем, топологии печатных плат, создания корпусов микросхем и простейшего анализа цепей питания и сигнальной передачи информации [4].

Проектирование, разработка и создание устройств производится в Cadence Allegro 2022. Данная САПР обладает принципом модульности и взаимозаменяемости, поэтому системное проектирование устройства производится в следующих программах входящих в САПР Cadence Allegro 2022:

- Allegro Design CIS – программа для проектирования электрических принципиальных схем и создания электрических цепей с использованием компонентов доступных в интегрированных библиотеках (рисунок 17) [3].

- Allegro PCB Designer – программа для проектирования печатной платы на основе уже созданной виртуальной модели устройства созданной на основе электрических цепей с использованием компонентов доступных в интегрированных библиотеках (рисунок 18).

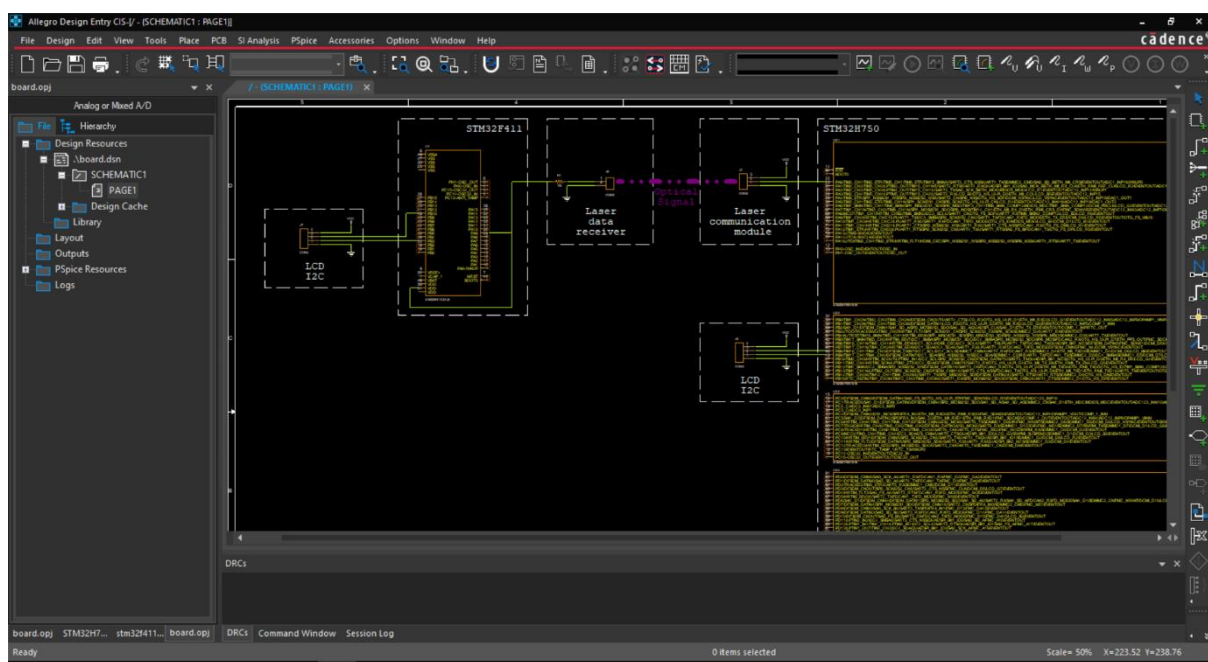


Рисунок 17 – Электрическая схема устройства спроектированная в Cadence Allegro Design CIS

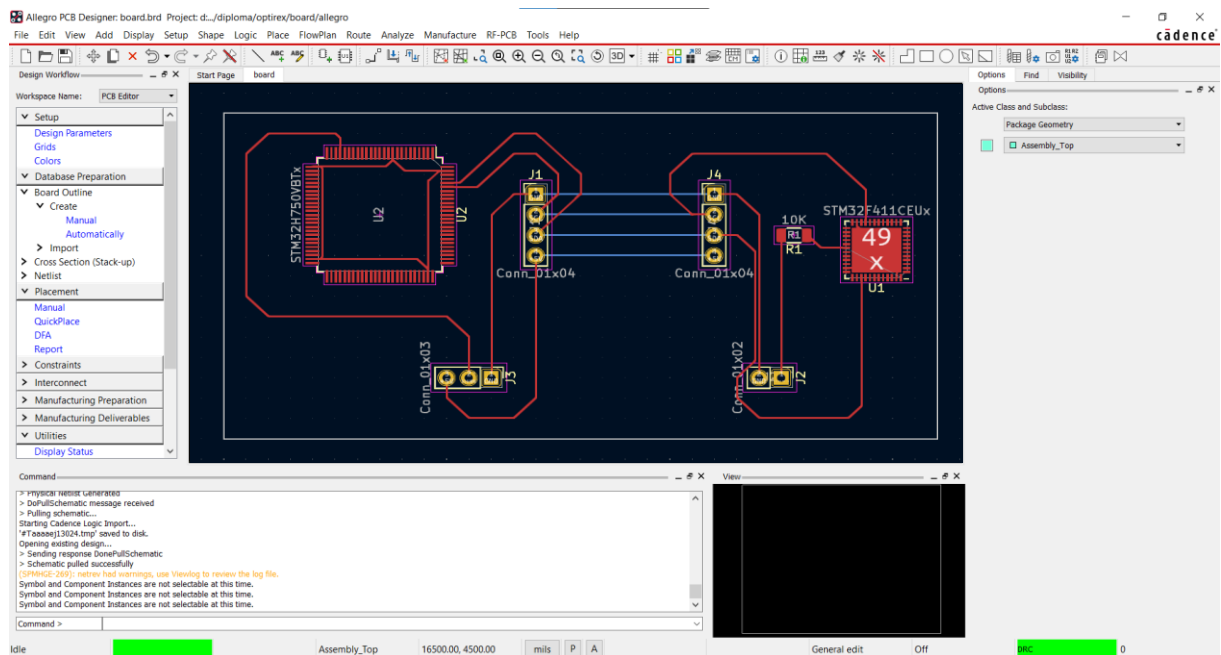


Рисунок 18 – Печатная плата на основе созданной электрической схемы  
Cadence Allegro PCB Designer

Выполняя проектирование аппаратного обеспечения комплекса, сохранилась модульная архитектура позволяющая подключать все необходимые компоненты с помощью механических разъёмов для быстрой замены неисправных элементов и усовершенствования уже выпущенных устройств [2].

## 2.2 Проектирование программного обеспечения комплекса

Разработка встраиваемых приложений для микроконтроллеров серии STM32, осуществляется с помощью интегрированной среды разработки STM32CubeIDE компании STMicroelectronics с бесплатной лицензией и пожизненным сроком обновления и сервисной поддержки. STM32CubeIDE в отличие от других интегрированных сред программирования микроконтроллерных платформ обладает следующими преимуществами:

- STM32CubeIDE доступна для загрузки бесплатно и предлагает множество функций, облегчающих разработку программного обеспечения для микроконтроллеров семейства STM32. Она объединяет в себе различные инструменты и ресурсы, включая интегрированную среду разработки, компилятор, отладчик и конфигурационные файлы для микроконтроллеров STM32.

- STM32CubeIDE поддерживает различные языки программирования, включая C и C++. Это позволяет использовать предпочитаемый язык программирования при работе с микроконтроллерами семейства STM32.

- STM32CubeIDE предлагает интуитивно понятный пользовательский интерфейс, что делает процесс разработки более эффективным и удобным. Его



пользовательский интерфейс содержит различные панели, окна и инструменты, которые обеспечивают простую навигацию и управление проектами (рисунок 19).

- STM32CubeIDE обеспечивает интеграцию с другими инструментами, такими как STM32CubeMX и STM32CubeProfiler. STM32CubeMX позволяет генерировать и настраивать исходный код для периферийных устройств микроконтроллеров STM32, а STM32CubeProfiler предоставляет средства для профилирования и анализа производительности приложений.

- STM32CubeIDE обеспечивает интеграцию с отладчиками и эмуляторами, позволяя разработчикам выполнять отладку кода, наблюдать переменные, проверять и исправлять ошибки и анализировать работу программы на микроконтроллере.

- STM32CubeIDE поставляется с обширной библиотекой периферийных драйверов и примеров кода для микроконтроллеров STM32. Это упрощает разработку, так как разработчики могут использовать готовые решения и примеры для быстрой разработки своих приложений.

STM32CubeIDE предоставляет разработчикам мощный инструмент для разработки приложений на микроконтроллерах STM32, упрощая процесс разработки, отладки и тестирования программного обеспечения. Она является широко применяемой интегрированной средой разработки и пользуется популярностью среди профессиональных разработчиков, работающих с микроконтроллерами STM32.

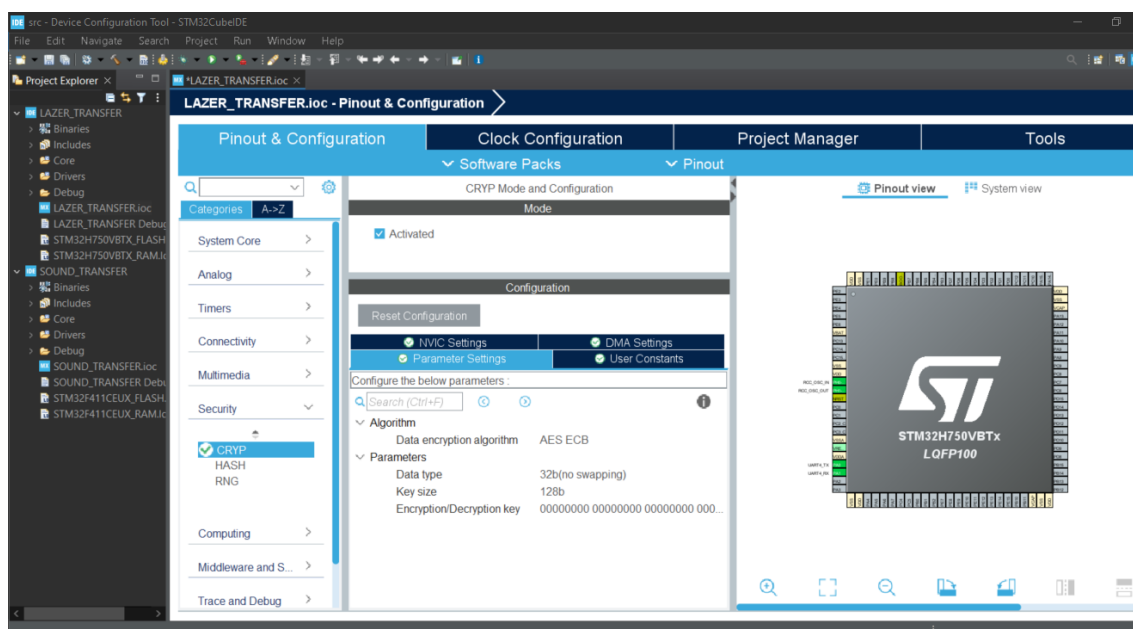


Рисунок 19 – Интерфейс интегрированной среды разработки STM32CubeIDE

### 3 ОПТИМИЗАЦИЯ КРИПТОГРАФИЧЕСКОГО АЛГОРИТМА

#### 3.1 Криптографический алгоритм AES

В период с 1972 по 1977 года в ходе открытого конкурса и общественных слушаний по созданию криптографического алгоритма устойчивого к криптоанализу и различным атакам, Национальным бюро стандартизации Соединенных Штатов Америки был принят в качестве финального алгоритма – криптографический алгоритм DES (Data Encryption Standard, с англ. – стандарт шифрования данных), разработанный компанией IBM [23].

Архитектура алгоритма DES основывалась на использовании принципов симметричного шифрования информации – использовании блоков открытого и секретного текста. Блоки открытого и секретного текста имели размерность в 64 бита каждый, при этом длина ключа необходимого для совершения операции шифрования и расшифрования информации первоначально составлял 56 бит.

Алгоритм DES основан на использовании ячеек Фейстеля и таблиц расширения с возможностью перестановки битов текста. Использовании архитектуры сети Фейстеля с раундовой обработкой текста позволило существенно уменьшить зависимость от прямого криптоанализа и имело преимущество в виде аппаратной реализации криптографического алгоритма с помощью логических интегральных микросхем [22].

Для повышения стойкости криптографического алгоритма и замены стандарта DES, в 1997 был объявлен открытый конкурс по созданию криптографического алгоритма устойчивого к атаке «грубой силой». Алгоритм DES кроме неоспоримых преимуществ имел фундаментальные недостатки, заложенные еще при разработке и имплементации криптографического алгоритма, а именно:

- Невозможность использовать поточное шифрование информации;
- Длина ключа в 56 бит позволяла со временем реализовать атаку методом «грубой силы», при которой возможно осуществить полный перебор всех ключей необходимых для шифрования информации на всех этапах передачи по линиям связи.

Победителем конкурса в 1998 году стал алгоритма «Rijndael», названный по инициалам его авторов – бельгийскими криптографами Джоаном Даменом и Винсентом Рейменом. При разработке криптографического алгоритма AES разработчики предусмотрели все слабые места алгоритма DES, а также его усовершенствованных вариаций и алгоритмов на основе сетей Фейстеля – TDES, Blowfish, DEA [25].

Криптографический алгоритм AES является блочным шифром с раундовым шифрованием информации с переменной длиной ключа и фиксированными длинами входного и выходного блоков шифрования. Длина ключа зависит от применения криптографического алгоритма – вычислительные системы с малой степенью интеграции и производительности имеют длину

ключа в 128 бит, и напротив – вычислительные системы с высоким уровнем производительности имеют длину ключа в 192 либо 256 бит. Фиксированная длина входного и выходного блоков шифрования имеет длину в 128 бит для упрощения аппаратной и программной имплементации криптографического алгоритма на разных вычислительных системах в независимости от цифровой архитектуры центрального процессора или операционной системы.

AES-256 (Advanced Encryption Standard) — это криптографический алгоритм симметричного шифрования, который используется для защиты данных и обеспечения их конфиденциальности. Он был разработан Национальным институтом стандартов и технологий (на англ. NIST) США и является одним из самых безопасных алгоритмов шифрования на сегодняшний день. Криптографический алгоритм AES-256 использует в своей основе 256-битный ключ для шифрования данных, что обеспечивает высокий уровень защиты по сравнению с другими алгоритмами, например DES и TDES. Каждый блок данных обрабатывается отдельно, что делает алгоритм надежным и защищенным от атак типа "человек посередине" (с англ. – man-in-the-middle).

Разработчики криптографического алгоритма заложили в структура алгоритма понятие высокой безопасности с помощью табличных массивов зашифрованных и не зашифрованных данных, при этом, злоумышленник, который располагает большой вычислительной мощностью может использовать атаку на криптографический алгоритм с помощью метода перебора ключа (с англ. – brute force attack) или атака по времени выполнения (с англ. – timing attack). Поэтому особенно важно использовать дополнительные меры безопасности, такие как двухфакторная аутентификация и защита от несанкционированного аппаратного считывания прошивки и данных из Flash-памяти.

На сегодняшний день, AES-256 является одним из самых надежных алгоритмов шифрования в мире, который обеспечивает высокий уровень защиты данных. Он используется в различных сферах, где требуется защита конфиденциальных данных, и является важным инструментом для обеспечения безопасности в современном мире.

Криптографический алгоритм AES-256 может быть реализован на микроконтроллере STM32 с помощью специальных библиотек, таких как STM32 X-CUBE CRYPTOLIB. Эта библиотека содержит набор функций для шифрования и расшифровки данных с использованием AES-256.

Однако, при использовании алгоритма AES-256 на микроконтроллере STM32 любой серии, необходимо учитывать ограниченные ресурсы микроконтроллер – объем оперативной и постоянной памяти, а также производительность микроконтроллера в вычислениях операций с плавающей точкой. Поэтому важно оптимизировать код и выбирать подходящие параметры алгоритма для конкретной прикладной задачи.



### 3.2 Аппаратные ускорители шифрования и дешифрования информации

Микроконтроллеры STM32 являются важным элементом современных систем автоматизации и управления. Однако, при использовании этих устройств возникает необходимость обеспечения безопасности программного кода и данных, хранящихся во встроенной Flash-памяти. Для этого разработчики микроконтроллера STM32 предусмотрели механизмы защиты на уровне аппаратуры и программного обеспечения [25].

Один из таких механизмов – защита области памяти (Memory Protection Unit, MPU), который позволяет разделить память на несколько областей и задать права доступа к каждой из них. Это позволяет защитить программный код и данные от несанкционированного доступа и изменения (рисунок 20).

Memory Protection Unit (MPU) — это механизм защиты памяти на уровне аппаратуры, который позволяет разделить память на несколько областей и задать права доступа к каждой из них. Это позволяет защитить программный код и данные от несанкционированного доступа и изменения.

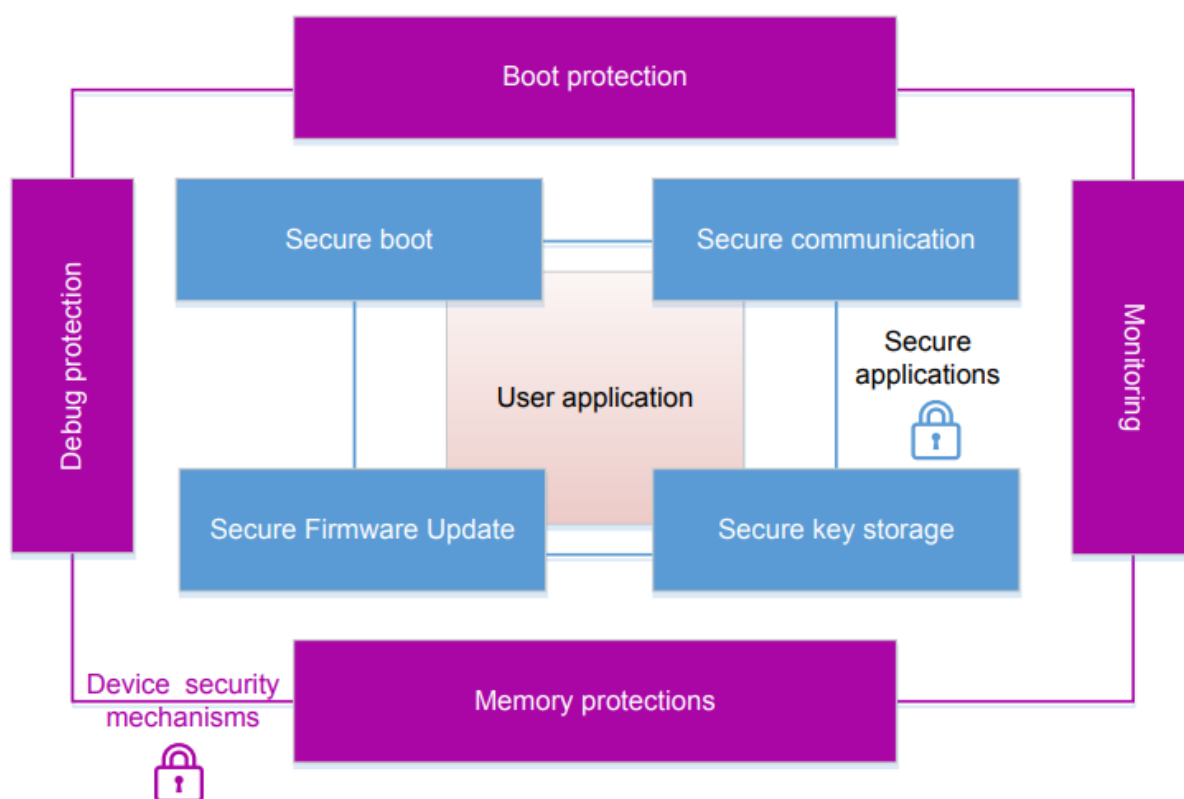


Рисунок 20 – Уровни аппаратно-программной защиты информации в микроконтроллерах серии STM32

MPU работает на уровне железа и может быть настроен для ограничения доступа к определенным областям памяти для чтения, записи или выполнения. Например, можно разрешить чтение программного кода, но запретить его запись или выполнение из области данных. Для настройки MPU используются

специальные регистры, которые задают параметры защиты для каждой области памяти. Кроме того, в микроконтроллерах STM32 есть возможность использовать MPU в сочетании с механизмом защиты от записи, что позволяет еще более усилить защиту данных Flash-памяти (рисунок 21).

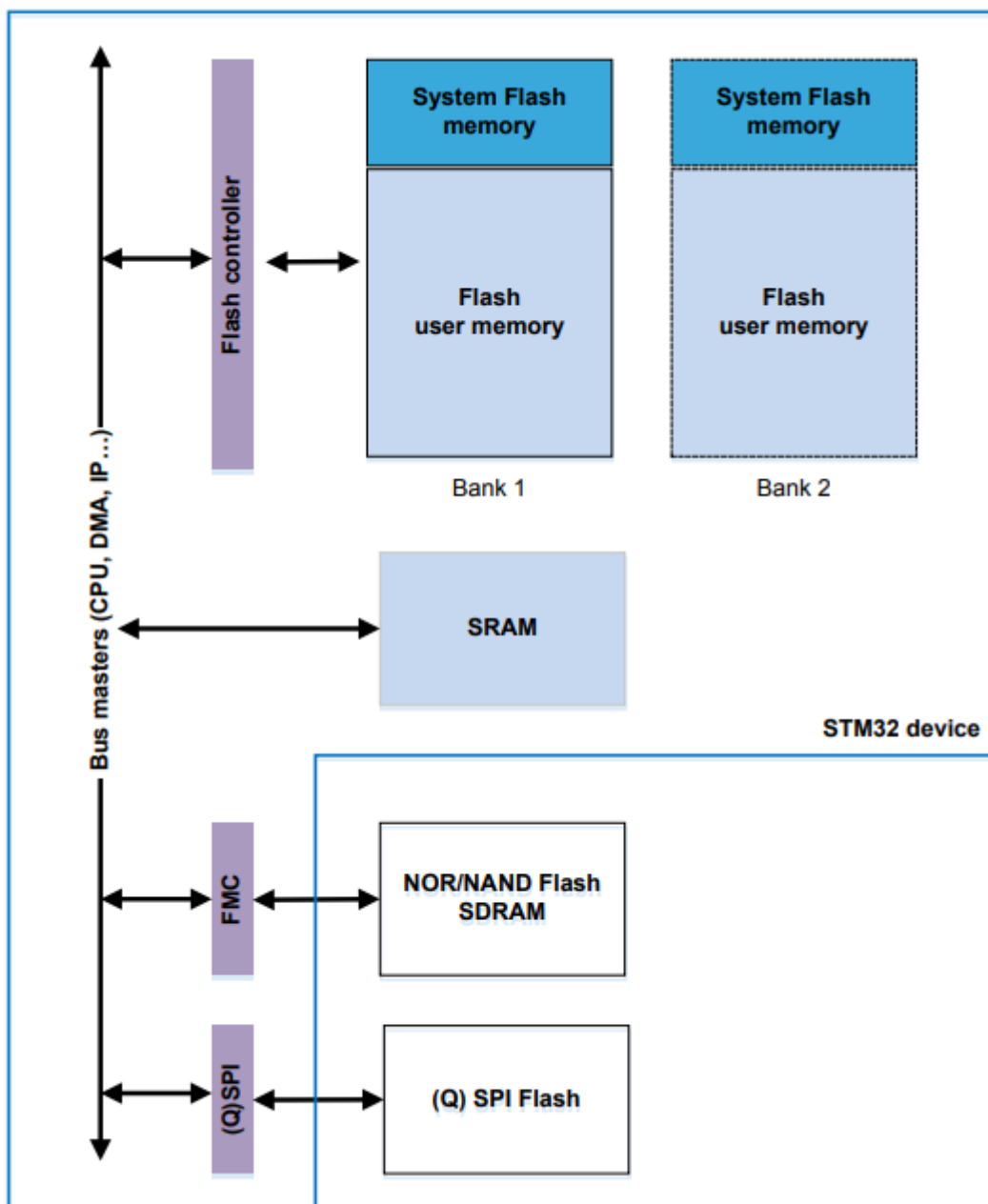


Рисунок 21 – Защита информации с использованием механизмов защиты Flash-памяти

Кроме того, микроконтроллеры STM32 имеют механизм защиты от записи (Write Protection), который позволяет запретить запись в определенные области памяти. Это может быть полезно для защиты программного кода от изменений после его загрузки в устройство. Для настройки защиты Flash-памяти в STM32 используется специальный инструментальный – STM32CubeMX и

STM32CubeProg. С их помощью можно задать параметры защиты, такие как права доступа к областям памяти и уровень защиты от записи.

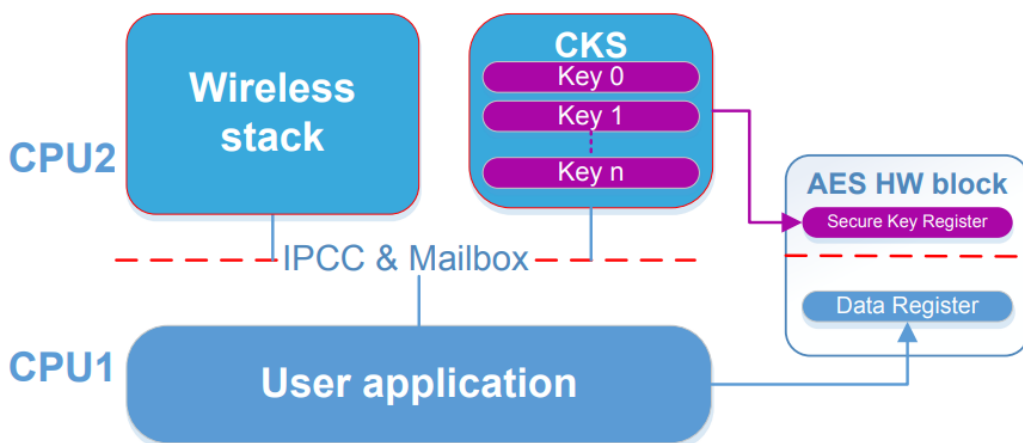


Рисунок 22 – Защита информации с помощью аппаратного блока ускорения криптографических операций

Защита данных Flash-памяти в микроконтроллере STM32 является важным аспектом обеспечения безопасности программного кода и данных. Она особенно важна для устройств, работающих в критических условиях или хранящих конфиденциальную информацию как в случае с трансляцией зашифрованной информации по инфракрасному оптическому каналу связи (рисунок 22).

Рассмотренные на практике средства защиты компьютерной информации касательно применения таких средств на микроконтроллерных платформах семейства ARM и STM32, конечно же, не закрывают весь спектр угроз безопасности информации.

### 3.3 Оптимизация криптографического алгоритма AES

Молниеносное развитие сетевых технологий, сподвигло многих компаний и людей задуматься над безопасностью передаваемой, отправляемой и хранимой информацией. Существует огромное количество приложений, которые по определению должны обеспечивать высокую степень защиты данных. Это касается Интернета вещей (рисунок 23), терминалов оплаты, банкоматов, счетчиков коммунальных услуг, систем безопасности и многих других. При этом в каждом из перечисленных случаев необходимо защитить данные не только от кражи, но и от вредоносного изменения. Это достаточно сложная задача даже для статистической математики.

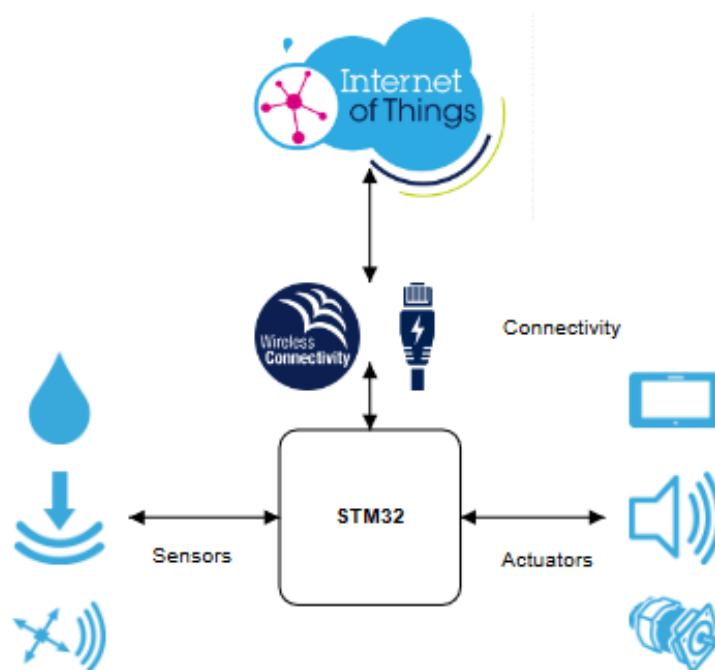


Рисунок 23 – Составная схема базового устройства Интернета вещей на базе микроконтроллера STM32

В июле 2016 года компания STMicroelectronics презентовала свою фирменную библиотеку шифрования для надежного хранения и защиты информации непосредственно на микроконтроллерах. Библиотека была выпущена в качестве программного расширения для фирменной интегрированной среды разработки STM32CubeIDE. Библиотека имеет сертифицированную поддержку следующих алгоритмов:

- Криптографические алгоритмы серии AES с разрядностью 128, 192, и 256 в режимах ECB, CBC, CTR, CFB, OFB, CCM, GCM, CMAC, KEY WRAP, XTS.
- Хеш-функции с поддержкой режима HMAC серии SHA с разрядностью 1, 224, 256, 384, 512 бит.
- Программный генератор случайных чисел на базе DRBG-AES-128.

- Создание и хранение цифровых ключей по стандарту RSA с поддержкой протокола PKCS-1v1.5 с режимами кодирования/декодирования и хранения цифровой подписи.

- Генерация и хранения ключей по стандарту ECC с режимами Scalar multiplication и ECDSA.

Кроме того, библиотека имеет несертифицированную поддержку дополнительных алгоритмов ARC4, DES, TripleDES (ECB (Electronic Codebook Mode) и CBC (Cipher-Block Chaining)), хеш-функции (MD5 и HKDF-SHA-512), ChaCha20, Poly1305, CHaCHA20-POLY1305, ED25519, Curve25519.

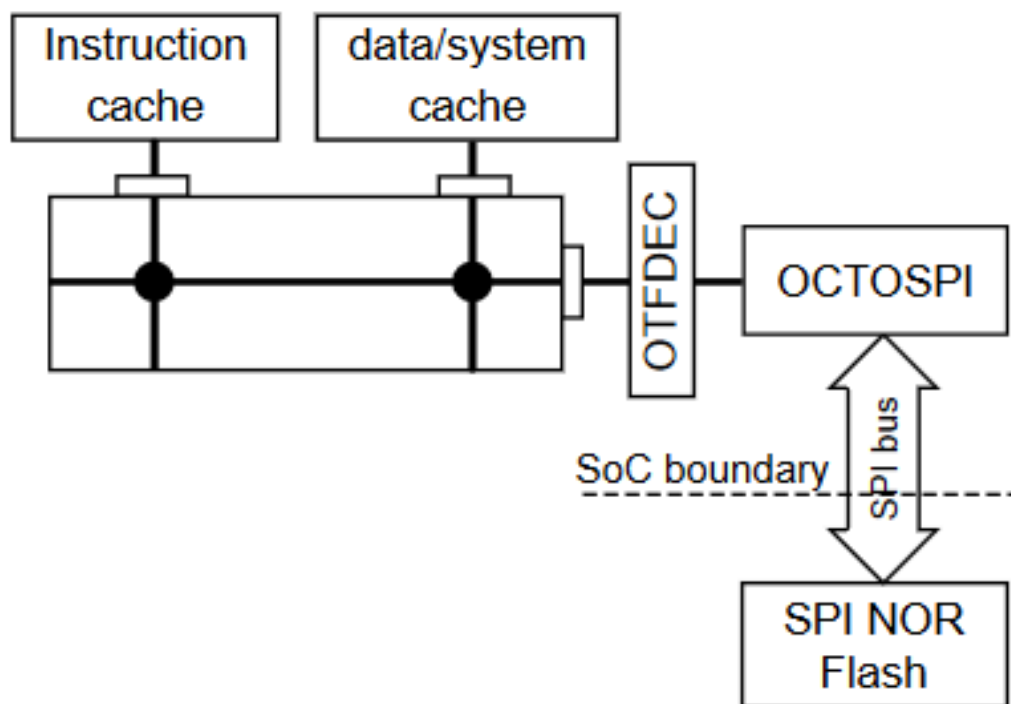


Рисунок 24 – Схема работы криптографического сопроцессора на базе микроконтроллера STM32

Проводя различные исследования и эксперименты с программным кодом, я выделил несколько пунктов об оптимизации криптографического алгоритма для использования в микроконтроллере STM32. Оптимизация алгоритма AES-256 для микроконтроллера STM32F1/F4/H7 может быть выполнена с помощью следующих методов:

- Выбор правильных параметров алгоритма. AES-256 имеет несколько параметров, например количество раундов и размер ключа, которые могут быть настроены для оптимизации производительности. В качестве примера скажу, что уменьшение количества раундов может ускорить процесс шифрования, но может снизить уровень безопасности.

- Использование аппаратного ускорения. Многие микроконтроллеры STM32 имеют аппаратное ускорение для криптографических операций, включая

AES-256. Использование аппаратного ускорения может значительно ускорить процесс шифрования и расшифровки данных (рисунок 22).

- Оптимизация кода. Оптимизация кода может включать в себя использование более эффективных алгоритмов или структур данных, уменьшение количества операций и использование более быстрых функций.

- Управление памятью. Управление памятью может быть оптимизировано для уменьшения нагрузки на микроконтроллер и ускорения процесса шифрования. Например, использование буферов фиксированного размера может снизить количество операций выделения и освобождения памяти.

- Использование асинхронных операций. Использование асинхронных операций может позволить микроконтроллеру выполнять другие задачи во время процесса шифрования или расшифровки данных (рисунок 23).

- Оптимизация выборки ключей. Выборка ключей может быть оптимизирована для уменьшения нагрузки на микроконтроллер и ускорения процесса шифрования. Например, использование кэша ключей может снизить количество операций выборки ключей (рисунок 24).

## ЗАКЛЮЧЕНИЕ

Актуальность инфракрасных оптических систем в сфере телекоммуникаций специального и общего назначения не оспаривается ввиду наличия преимуществ по сравнению с другими типами передачи информации. Предприятия, объекты и организации, имеющие критическую информационную инфраструктуру, увеличиваются инвестиции, направленные на развитие защищенных телекоммуникационных систем специального и общего назначения для передачи зашифрованной информации.

Разработанный аппаратно-программный комплекс для безопасной передачи сообщения по оптическому каналу связи повышает защищенность телекоммуникационных систем специального назначения за счет использования аппаратно-программных криптографических алгоритмов, повышающих скорость потокового шифрования информации и уровня защищенности передаваемой информации с помощью инфракрасного оптического сигнала.

В ходе выполнения выпускной квалификационной работы был разработан аппаратно-программный стенд для безопасной передачи сообщения по инфракрасному оптическому каналу связи между двумя устройствами – приемником и передатчиком с использованием оптимизированных криптографических алгоритмов для аппаратно-программного ускорения потокового шифрования информации.

В данной работе была спроектирована, указана и продемонстрирована чертежно-конструкторская документация на разработанные устройства для безопасной передачи сообщения по инфракрасному оптическому каналу связи, а также программное обеспечение необходимое для потокового шифрования передающейся информации оптимизированного для микроконтроллерной платформы общего назначения.

Целью в данной выпускной квалификационной работе было создание аппаратно-программного комплекса для безопасной передачи информации по оптическому каналу связи и применение данного комплекса в обучении студентов учреждений технического и профессионального образования по специальностям и квалификациям, связанным с аппаратно-программными комплексами информационной безопасности и основам сигнальной передачи в высокоскоростной электронике.

Для реализации данного проекта был выбран язык C++, а в качестве аппаратной платформы – микроконтроллеры STM32 семейства H7. Для создания аппаратного и программного обеспечения комплекса были выбраны следующие системы автоматизированного проектирования:

- Cadence Allegro 2022 – для проектирования электрических схем и трассировки электронных компонентов и печатных плат;
- Cadence Sigrity 2022 – для системного анализа печатных плат по тепловым, сигнальным и физическим характеристикам;
- STM32CubeIDE – для проектирования программного обеспечения и оптимизации криптографических алгоритмов;

Цель выпускной квалификационной работы достигнута – разработан аппаратно-программный комплекс для безопасной передачи сообщения по оптическому каналу связи с использованием оптимизированных криптографических алгоритмов. Архитектура комплекса прошла все физические, аппаратные и программные тесты для соответствия всем заявленным характеристикам.

Мы показали, что создание архитектуры с нуля позволяет понять, как устроена модель электронного магазина, какова его структура, свойства, законы моделирования; научиться управлять данной архитектурой, развивать ее в необходимом русле, определять наилучшие способы управления проектом, прогнозировать последствия тех или иных действий в данной модели проекта.

Разработка аппаратно-программного комплекса для безопасной передачи сообщения по оптическому каналу связи позволяет дополнительно изучить аспекты проектирования высокоскоростной электроники и печатных плат, программирования встраиваемых систем на основе промышленных микроконтроллеров, основ передачи зашифрованной информации с помощью оптических систем и использование оптимизированных криптографических алгоритмов для микроконтроллерных платформ общего назначения.



## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Thaker N.B., Ashok R., Manikandan S., Nambath N., Gupta S., – «A Cost-Effective Solution for Testing High-Performance Integrated Circuits»; IEEE TRANSACTIONS ON COMPONENTS PACKAGING AND MANUFACTURING TECHNOLOGY, 2021. – 7 p.
- 2 Yi Z.C., Feng H.Q., Zhou X.F., Shui L.L., – "Design of an Open Electrowetting on Dielectric Device Based on Printed Circuit Board by Using a Parafilm M"; FRONTIERS IN PHYSICS, 2020. – 8 p.
- 3 Zhang Y., Liu Z.Y., Zhang X.L., Guo S.Y., – "Sandwich-Layered Dielectric Film with Intrinsically Excellent Adhesion, Low Dielectric Constant, and Ultralow Dielectric Loss for a High-Frequency Flexible Printed Circuit"; INDUSTRIAL & ENGINEERING CHEMISTRY RESEARCH, 2021. – 10 p.
- 4 Park H., Song J., Sim J., Choi Y., Choi J., Yoo J., Kim C., – "30-Gb/s 1.11-pJ/bit Single-Ended PAM-3 Transceiver for High-Speed Memory Links"; IEEE JOURNAL OF SOLID-STATE PHYSICS, 2021. – 9 p.
- 5 Pace L., Idir N., Duquesne T., De Jaeger J.C., – "Parasitic Loop Inductances Reduction in the PCB Layout in GaN-Based Power Converters Using S-Parameters and EM Simulations"; ENERGIES, 2021. – 15 p.
- 6 Mitzner K., «Complete PCB Design Using OrCAD Capture and PCB Editor» – Second Edition. – Elsevier Academic Press, 2019. – 600 p.
- 7 Митцнер К., Доу Б., Акулин А., Супонин А., Мюллер Д., «Проектирование печатных плат в OrCAD Capture и OrCAD PCB Editor», Второе издание. – Москва: Техносфера, 2022. – 592 с.
- 8 Труднов А.В., «Высокоскоростные печатные платы. Записки схемотехника», – Москва: Ridero, 2020. – 108 с.
- 9 Труднов А.В., «Высокоскоростные печатные платы. Теоретические рекомендации», – Москва: Ridero, 2019. – 156 с.
- 10 Труднов А.В., «Высокоскоростные печатные платы. Практические рекомендации», – Москва: Ridero, 2019. – 152 с.
- 11 Кечиев Л.Н., «Печатные платы и узлы гигабитной электроники», – Москва: Библиотека ЭМС, 2017. – 424 с.
- 12 Ardizoni J., «Practical guide to design high-speed printed circuit boards» – Cadence Publisher, 2019. – 20 p.
- 13 Bogatin E., «Practical guide to transmission line design and characterization for signal integrity applications» – Artech House, 2020. – 604 p.
- 14 Griffin B., «Cadence Sigrity 2019 Release Function» – Cadence Publisher, 2019. – 56 p.
- 15 Коберниченко В.Г., «Основы цифровой обработки сигналов», – Изд-во Урал. Ун-та, 2018. – 150 с.
- 16 Белоус А.И., Солодуха В.А., Шведов С.В., «Основы конструирования высокоскоростных электронных устройств», – Москва: Техносфера, 2017, 872 с.
- 17 Пош М., «Программирование встроенных систем на C++ 17» / пер. с англ. А.В. Снастина. – М.: ДМК Пресс, 2020. – 394 с.

- 18 Альфред, В. Ахо., «Компиляторы. Принципы, технологии и инструментарий» / Альфред В. Ахо и др. – Москва: Высшая школа, 2015 – 882 с.
- 19 Ашарина И.В., «Основы программирования на языках С и С++: Курс лекций для высших учебных заведений» / И.В. Ашарина. — М.: Гор. линия-Телеком, 2018. — 208 с.
- 20 Герберт, Шилдт «С++. Базовый курс» / Шилдт Герберт. – М.: Диалектика / Вильямс, 2022. – 564 с.
- 21 Дейтел, Пол «Как программировать на С» / Пол Дейтел, Харви Дейтел. – М.: Бином, 2022. – 858 с.
- 22 Владимиров С.М., Габидулин Э.М., Колыбельников А.И., Кшевецкий А.С., «Криптографические методы защиты информации.» – М.: Издательство Московского Государственного Университета, 2021. – 433 с.
- 23 Беляков С.Л., Боженюк А.В., Петряева М.В., «Основы разработки программы на языке С++ для систем информационной безопасности: учебное пособие». – М.: Издательство Южного Федерального Университета, Ростов-на-Дону, 2020. – 152 с.
- 24 Novello K., «Mastering STM32». – М.: Leanpub Publishing, 2022. – 910 p.
- 25 AN5156 Application Note – Introduction to STM32 microcontrollers security., 2019. – 55 с.

## ПРИЛОЖЕНИЕ

```
CRYP_HandleTypeDef hcryp;
__ALIGN_BEGIN static const uint32_t pKeyCRYP[4] __ALIGN_END = {
    0x00000000,0x00000000,0x00000000,0x00000000};

UART_HandleTypeDef huart4;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_UART4_Init(void);
static void MX_CRYP_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */
uint8_t buffer[] = "Hello world";
uint8_t msg[64];
unsigned long Time;
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */
```

```

/* USER CODE END Init */

/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_UART4_Init();
MX_Cryp_Init();
/* USER CODE BEGIN 2 */
Time = HAL_GetTick();
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    if (HAL_GetTick() - Time >= 1000) {
        Time = HAL_GetTick();
        HAL_UART_Transmit(&huart4, msg, sprintf(msg, "Hello"), 0xFFFF);
    }
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Supply configuration update enable
 */

```

```

HAL_PWREx_ConfigSupply(PWR_LDO_SUPPLY);

/** Configure the main internal regulator output voltage
 */

__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_
SCALE3);

while(!__HAL_PWR_GET_FLAG(PWR_FLAG_VOSRDY)) {}

/** Initializes the RCC Oscillators according to the specified parameters
 * in the RCC_OscInitTypeDef structure.
 */
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSISState = RCC_HSI_DIV1;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 4;
RCC_OscInitStruct.PLL.PLLN = 12;
RCC_OscInitStruct.PLL.PLLP = 2;
RCC_OscInitStruct.PLL.PLLQ = 2;
RCC_OscInitStruct.PLL.PLLR = 2;
RCC_OscInitStruct.PLL.PLLRGE = RCC_PLL1VCIRANGE_3;
RCC_OscInitStruct.PLL.PLLVCOSSEL = RCC_PLL1VCOWIDE;
RCC_OscInitStruct.PLL.PLLFRACN = 4096;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}

/** Initializes the CPU, AHB and APB buses clocks
 */
RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2

|RCC_CLOCKTYPE_D3PCLK1|RCC_CLOCKTYPE_D1PCLK1;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.SYSCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.AHBCLKDivider = RCC_HCLK_DIV1;
RCC_ClkInitStruct.APB3CLKDivider = RCC_APB3_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_APB1_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_APB2_DIV1;

```

```

RCC_ClkInitStruct.APB4CLKDivider = RCC_APB4_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) !=
HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief CRYPT Initialization Function
 * @param None
 * @retval None
 */
static void MX_CRYPT_Init(void)
{

    /* USER CODE BEGIN CRYPT_Init 0 */

    /* USER CODE END CRYPT_Init 0 */

    /* USER CODE BEGIN CRYPT_Init 1 */

    /* USER CODE END CRYPT_Init 1 */
    hcrypt.Instance = CRYPT;
    hcrypt.Init.DataType = CRYPT_DATATYPE_32B;
    hcrypt.Init.KeySize = CRYPT_KEYSIZE_128B;
    hcrypt.Init.pKey = (uint32_t *)pKeyCRYPT;
    hcrypt.Init.Algorithm = CRYPT_AES_ECB;
    if (HAL_CRYPT_Init(&hcrypt) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN CRYPT_Init 2 */

    /* USER CODE END CRYPT_Init 2 */

}

/**
 * @brief UART4 Initialization Function
 * @param None
 * @retval None
 */

```

```

static void MX_UART4_Init(void)
{

    /* USER CODE BEGIN UART4_Init 0 */

    /* USER CODE END UART4_Init 0 */

    /* USER CODE BEGIN UART4_Init 1 */

    /* USER CODE END UART4_Init 1 */
    huart4.Instance = UART4;
    huart4.Init.BaudRate = 9600;
    huart4.Init.WordLength = UART_WORDLENGTH_8B;
    huart4.Init.StopBits = UART_STOPBITS_1;
    huart4.Init.Parity = UART_PARITY_NONE;
    huart4.Init.Mode = UART_MODE_TX_RX;
    huart4.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart4.Init.OverSampling = UART_OVERSAMPLING_16;
    huart4.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart4.Init.ClockPrescaler = UART_PRESCALER_DIV1;
    huart4.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    if (HAL_UART_Init(&huart4) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_UARTEx_SetTxFifoThreshold(&huart4,
UART_TXFIFO_THRESHOLD_1_8) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_UARTEx_SetRxFifoThreshold(&huart4,
UART_RXFIFO_THRESHOLD_1_8) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_UARTEx_DisableFifoMode(&huart4) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN UART4_Init 2 */

    /* USER CODE END UART4_Init 2 */

}

```

```

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */

/* GPIO Ports Clock Enable */
__HAL_RCC_GPIOH_CLK_ENABLE();
__HAL_RCC_GPIOA_CLK_ENABLE();

/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
/* USER CODE BEGIN Error_Handler_Debug */
/* User can add his own implementation to report the HAL error return state */
__disable_irq();
while (1)
{
}
/* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number

```



```
* @retval None
*/
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```