

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Научный руководитель, старший
преподаватель департамента программной
инженерии факультета компьютерных наук

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

С.А.Шершаков
«__» _____ 2019 г.

В.В. Шилов
«__» _____ 2019 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

**МИКРОКОД ДЛЯ ESP32 ДЛЯ УПРАВЛЕНИЯ СВЕТОДИОДНОЙ
ЛЕНТОЙ**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729. 02.07-01 ПЗ 01-1

Исполнитель
студент группы БПИ 171
_____/ Н.А. Самойлов/
«__» _____ 2019 г.

Москва 2019

УТВЕРЖДЕН

RU.17701729.02.07-01 ПЗ 01-1

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

МИКРОКОД ДЛЯ ESP32 ДЛЯ УПРАВЛЕНИЯ СВЕТОДИОДНОЙ ЛЕНТОЙ

Пояснительная записка

RU.17701729.02.07-01 ПЗ 01-1

Листов 18

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «LedContoller» («Микрокод для esp32 для управления светодиодной лентой»). Данный документ содержит разделы «Введение», «Назначение и область применения программы», «Технические характеристики», «Ожидаемые технико-экономические показатели», «Источники, использованные при разработке»

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы;
- описание алгоритма и функционирования программы
- описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [3];
- 2) ГОСТ 19.102-77 Стадии разработки [4];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [5];
- 4) ГОСТ 19.104-78 Основные надписи [6];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [7];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [8];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [9].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [10], ГОСТ 19.604-78 [11].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	4
1. ВВЕДЕНИЕ	6
1.1. Наименование программы	6
1.2. Документы, на основании которых ведется разработка	6
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	7
2.1. Функциональное назначение	7
2.2. Эксплуатационное назначение	7
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	8
3.1. Постановка задачи	8
3.2. Описание алгоритма и функционирования программы	8
3.2.1. Общая структура работы программы в паре с Android приложением	8
3.2.2. Схема подключения	9
3.2.3. Логика работы программы «LedController»	10
3.2.3.1. UML диаграмма	10
3.2.4. Инициализация в коде	11
3.2.5. Класс BTServer	12
3.2.5.1. GATT сервер	12
3.2.6. ModeDispatcher	13
3.2.6.1. Прерывание анимаций	13
3.2.7. Animator	13
3.2.8. Animation	13
3.3. Описание и обоснование выбора состава технических и программных средств	14
3.3.1. Состав технических и программных средств	14
3.3.2. Обоснование выбора состава технических и программных средств	14
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	15
4.1. Предполагаемая потребность	15
4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	15
5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «LedController».

1.2. Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1012-02 от 10.12.2018 «Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

Функциональным назначением программы является предоставление Bluetooth интерфейса для Android программы «TripleLed»

2.2. Эксплуатационное назначение

Программа предназначена для микроконтроллера eps32.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи

Разработать программу для удобного и понятного взаимодействия с Bluetooth устройствами, а также обеспечить расширяемость системы для последующих модификаций программы.

3.2. Описание алгоритма и функционирования программы

3.2.1. Общая структура работы программы в паре с Android приложением

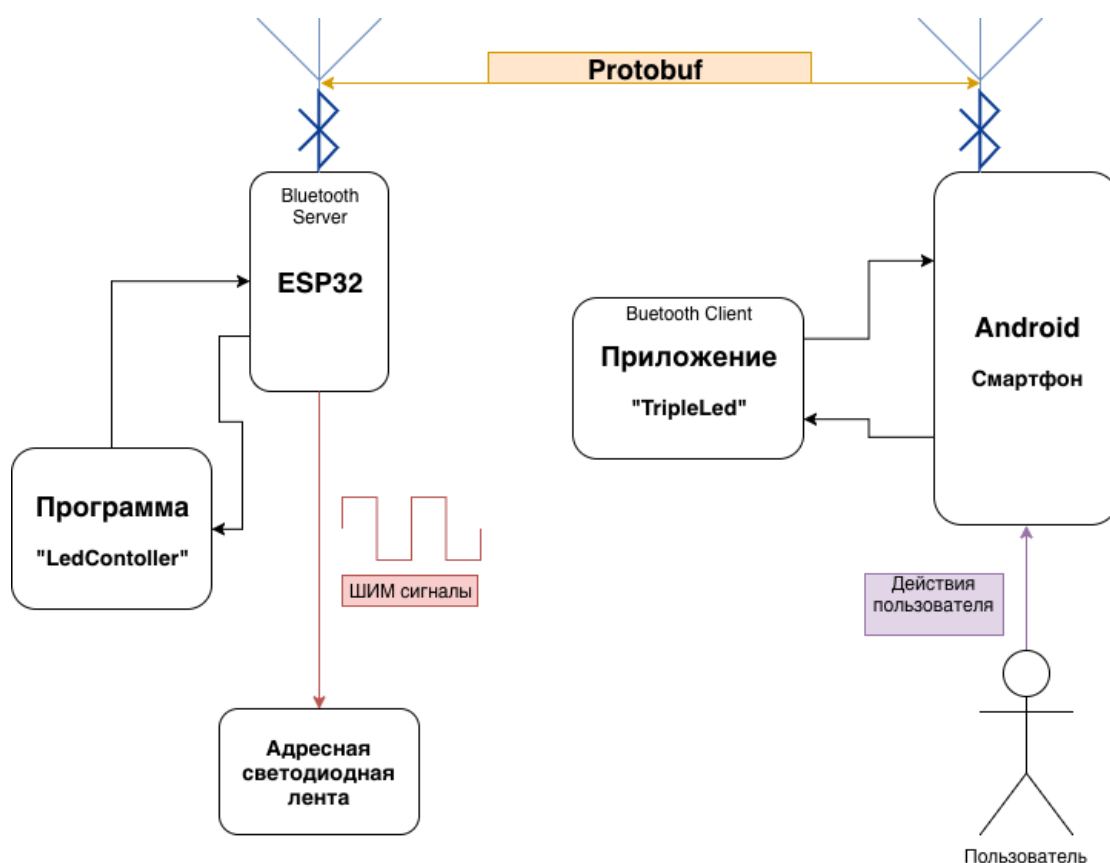


Рис. 1 Диаграмма работы системы из двух устройств

Работа системы, контролирующей работу светодиодной ленты, начинается с пользователя. Пользователь с помощью приложения «TripleLed» подключается к ESP32 посредством Bluetooth соединения и передает закодированные библиотекой Protobuf данные ESP32. ESP32 обрабатывает данные и через выходной пин под номером 18 посылает ШИМ сигналы для последующей обработки микроконтроллерами светодиодной ленты.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.2. Схема подключения

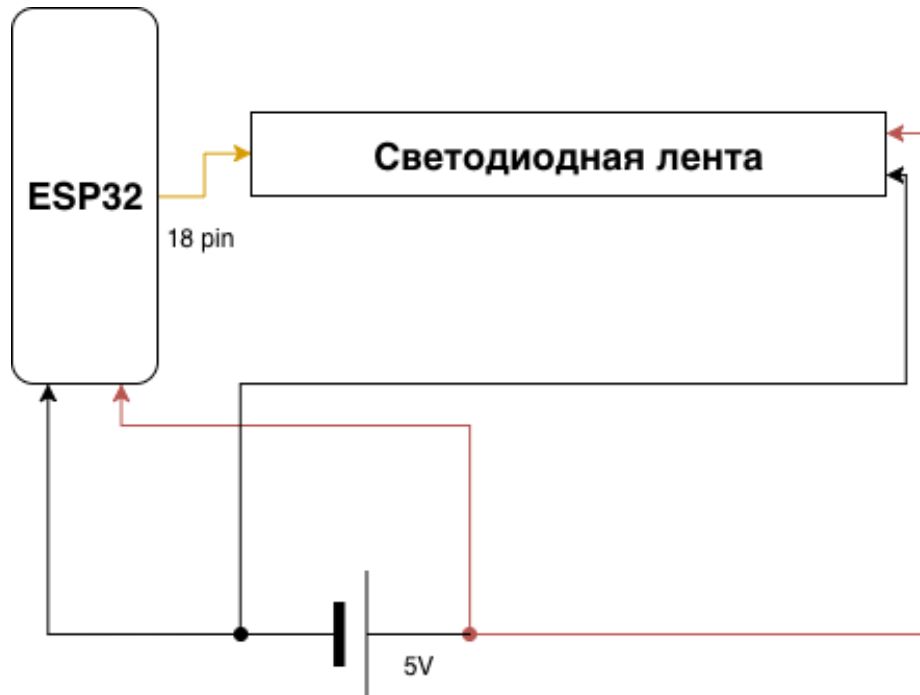


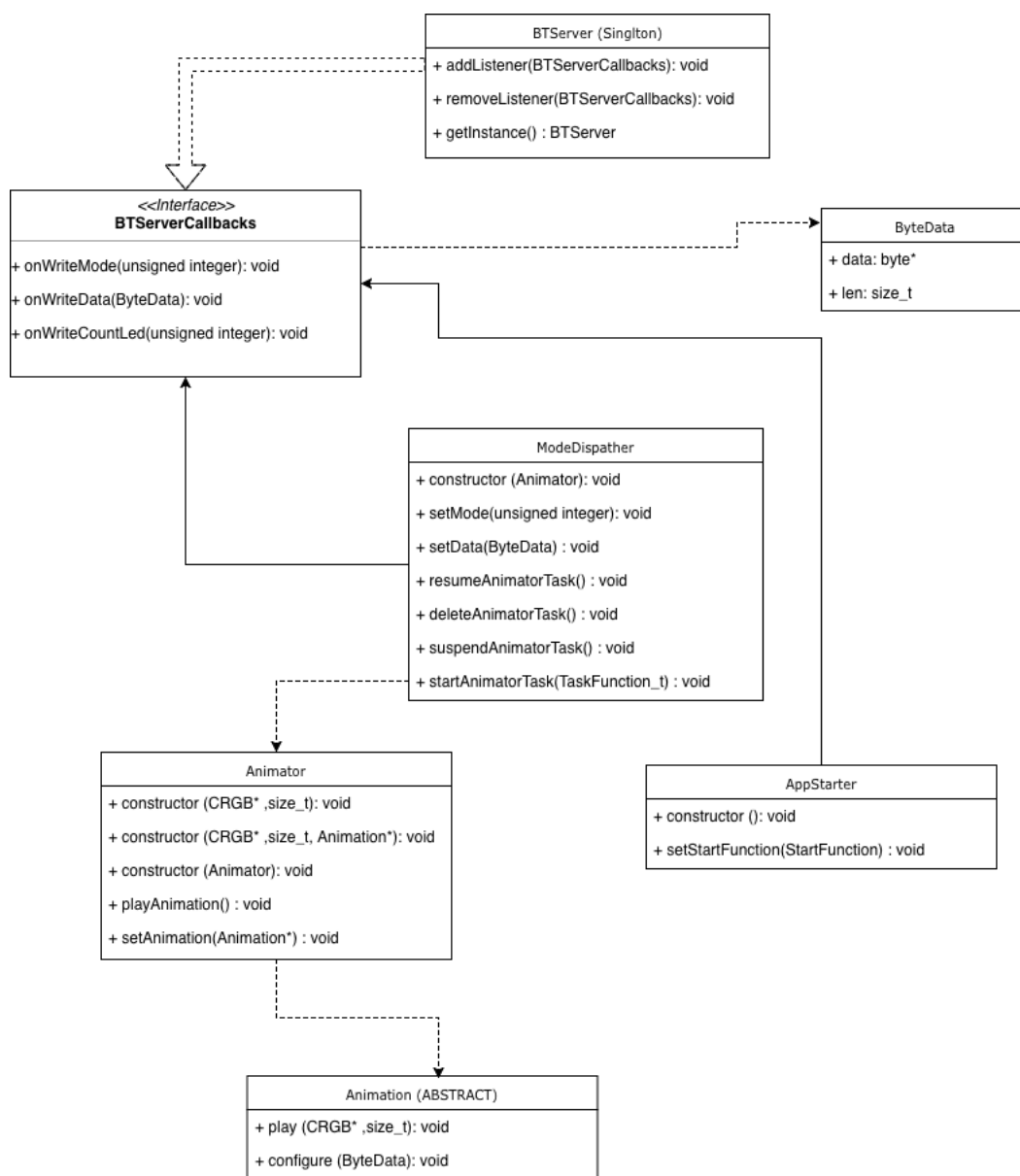
Рис. 2. Схема подключения

Для подключения необходимо питание напряжением 5v и мощностью равной потребляемой светодиодной лентой. Также необходимо подключить информационный вход адресной светодиодной ленты к 18 пину ESP32.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.3. Логика работы программы «LedController»

3.2.3.1. UML диаграмма



Работа программы начинается с инициализации GATT сервера [1]. Он представлен классом-синглтоном [2] BServer. В процессе инициализации создается сервис и его характеристики. Сервер ожидает данных о количестве светодиодов на ленте и как только он их получил, с помощью дочернего класса-интерфейса BServerCallbacks отправляет эти данные классу AppStarter, который занимается инициализацией всех остальных компонентов программы. А именно, создается класс ModeDispatcher, предназначенный для управления и настройки анимаций, представленных классами-наследниками Animation, с помощью класса-посредника Animator.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.4. Инициализация в коде

```
CRGB *leds;

Animator *animator;
ModeDispatcher *modeDispatcher;
AppStarter* appstarter;

void animationTask(void *xParam)
{
    for (;;)
    {
        LOG("Animation started!");
        animator->playAnimation();
        LOG("Animation has ended");
    }
}

void startApplication(unsigned int led_count)
{
    leds = new CRGB[led_count];
    animator = new Animator(leds, led_count);
    modeDispatcher = new ModeDispatcher(animator);
    modeDispatcher->setMode(ModeDispatcher::RAINBOW_ID);
    modeDispatcher->startAnimatorTask(animationTask);
    LOG("App has been started!");
}

void setup()
{
    Serial.begin(115200);
    appstarter = new AppStarter();
    appstarter->setStartFunction(startApplication);

    LOG("Initialization complete!");
}
```

В системе Arduino [3] стартовой точкой программы является ф-я setup(). В ней нам необходимо создать класс AppStarter и передать нашу ссылку на функцию с которой начнется инициализация, в случае если пользователь задаст количество светодиодов. В нашем случае это функция startApplication(). В ней необходимо создать массив CRGB[], который дает возможность задавать каждому светодиоду свой цвет в RGB формате [4]. Далее передаем этот массив классу Aninimator, который предоставляет интерфейс для запуска и остановки анимаций. Создаем ModeDispatcher, который будет связующим звеном между сервером и классом Animator, принимая информацию и производя соответствующие операции и задаем поток в виде Task [5], в котором будет проигрываться анимация.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.5. Класс BtServer

Класс BtServer представлен в виде синглтона дабы обеспечить легкий доступ и единственность объекта этого класса. Компонент в паре с BtCallbacks предоставляет упрощенный интерфейс для работы с GATT сервером. Слушателям этого класса необходимо наследоваться от BtCallbacks, реализовать его виртуальные функции, через которые будут поступать данные, и подписаться на вызов этих функций.

3.2.5.1. GATT сервер

GATT (Generic Attribute Profile) – один из протоколов стека протоколов Bluetooth LE [1]. Программа в данном случае создает таблицу, также называемую как GATT Table. В ней содержатся сервисы, а в них атрибуты, с помощью которых можно считывать и писать данные. В контексте этого проекта в ней содержится два стандартных сервиса, предоставляющие системную информацию и сервис, с которым работает данная программа. Он включает в себя следующие атрибуты:

- Version – отображает информацию о текущей версии прошивки в виде строки
- LedCount – отображает выставленное количество светодиодов в виде беззнакового целого.
- Mode – предоставляет данные о номере выставленной анимации в виде беззнакового целого.
- Data – содержит конфигурацию текущей анимации в виде байтовой информации, закодированной библиотекой Protobuf.

Логика получения и отправки информации через эти атрибуты, следующая:

1. В начале работы программа находится в состоянии ожидания поступления данных в атрибут LedCount. Как только они пришли, программа проводит инициализацию и может принимать информацию в другие два атрибута: Mode и Data.
2. При выставлении значения атрибуту Mode клиентом происходит смена анимации.
3. При выставлении значения атрибуту Data программа будет производить соответствующую конфигурацию анимации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.6. ModeDispatcher

Класс ModeDispatcher включает в себя все анимации, представленные наследниками класса Animation, их идентификаторы и интерфейс воздействия на класс Animator. В процессе инициализации подписывается на получение данных от класса BTServer, а также управляет логикой прерывания текущей анимации и обеспечивает потокобезопасность вызовов своих функций с помощью семафоров.

3.2.6.1. Прерывание анимаций

Так как прервать выполнение функции, проигрывающей анимацию, без помощи системных вызовов очень трудно, да и не эффективно, воспользуемся ими и будем запускать анимацию в отдельном процессе операционной системы FreeRTOS [5]. Как только нам понадобится остановить выполнение, вызовем системную функцию xTaskDelete() и остановим выполнение нашего процесса. Далее создадим новый вызовом xTaskCreate(). Так мы обеспечим функционирование переключения анимаций.

3.2.7. Animator

Предоставляет интерфейс задания анимаций и их проигрывания.

3.2.8. Animation

Абстрактный класс, наследник которого должен реализовать две функции: play() и configure(). В методе play() производятся операции над массивом CRGB[] и тем самым меняются состояния светодиодов на ленте. Метод configure() принимает информацию о выставленной конфигурации в виде массива байтов. Методу необходимо распаковать информацию с помощью библиотеки Protobuf и произвести изменения в соответствии с ней.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование выбора состава технических и программных средств

3.3.1. Состав технических и программных средств

Для работы программы необходим микроконтроллер ESP32, питание 5V, адресная светодиодная лента типа WS2812, устройство на базе операционной системы Android с установленным на нем приложением «TripleLed».

3.3.2 Обоснование выбора состава технических и программных средств

В качестве контроллера был выбран ESP32, так как он обладает рядом преимуществ по сравнению с аналогами, а именно: операционная система с открытым кодом, популярность устройства (что обеспечит большое количество пользователей), большое количество поддерживаемых на физическом уровне протоколов (Bluetooth, WIFI и т.д.), мощный процессор с 2 ядрами и т.д. [6]

В качестве платформы для разработки была выбрана платформа Arduino, так как обладает огромной базой библиотек и использует C++ подобный язык.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Программа предназначена для работы с Android приложением «TripleLed».

4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Нет.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Беспроводная технология Bluetooth с низким энергопотреблением [Электронный ресурс] / Электрон. дан. - 2019 – Режим доступа: https://ru.wikipedia.org/wiki/Bluetooth_с_низким_энергопотреблением , свободный. – Загл. С экрана.
2. Singleton [Электронный ресурс] / Rick Johnson – Электрон. дан. - 2016 – Режим доступа: [https://ru.wikipedia.org/wiki/Одиночка_\(шаблон_проектирования\)](https://ru.wikipedia.org/wiki/Одиночка_(шаблон_проектирования)), свободный. – Загл. С экрана.
3. Arduino [Электронный ресурс] / Rick Johnson – Электрон. дан. - 2019 – Режим доступа: <https://www.arduino.cc> свободный. – Загл. С экрана.
4. RGB [Электронный ресурс] / Rick Johnson – Электрон. дан. - 2019 – Режим доступа: <https://ru.wikipedia.org/wiki/RGB> свободный. – Загл. С экрана.
5. FreeRTOS [Электронный ресурс] / Rick Johnson – Электрон. дан. - 2019 – Режим доступа: <https://www.freertos.org> свободный. – Загл. С экрана.
6. ESP32 [Электронный ресурс] / Rick Johnson – Электрон. дан. - 2019 – Режим доступа: <https://www.espressif.com> свободный. – Загл. С экрана.
7. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
8. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
9. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
10. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
11. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
12. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

13. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
14. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
15. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата