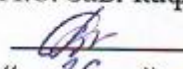


Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет Энергетический
Кафедра Автоматизации производственных процессов и электротехники
Направление подготовки 15.03.04 Автоматизация технологических
процессов и производств
Направленность (профиль) образовательной программы Автоматизация
технологических процессов и производств в энергетике

ДОПУСТИТЬ К ЗАЩИТЕ
И.о. зав. кафедрой
 О.В. Скрипко
« 26 » 06 2021 г

БАКАЛАВРСКАЯ РАБОТА

на тему: Автоматизированная система управления освещённости
помещения (комплексная бакалаврская работа)

Исполнитель
студент группы 741 об


11.06.2021
(подпись, дата)


А.В. Левшин

Руководитель
доцент, канд.тех.наук


24.06.2021
(подпись, дата)

М.Д. Штыкин

Консультант:
по безопасности жизне-
деятельности
доцент, канд. физ.мат. наук


11.06.2021
(подпись, дата)

В.Н. Аверьянов

Нормоконтроль
профессор, д-р.тех.наук


26.06.2021
(подпись, дата)

О.В. Скрипко

Благовещенск 2021

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет Энергетический
Кафедра автоматизации производственных процессов и электротехники

УТВЕРЖДАЮ

И.о.зав. кафедрой

 О.В. Скрипко

« 26 » 06 2021 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента 741 группы Левшина
Александра Васильевича

1. Тема выпускной квалификационной работы: Автоматизированная
система управления освещенности помещения (комплексная бакалаврская
работа)

(утверждена приказом от 23.04.21. № 812–уч)

2. Срок сдачи студентом законченной проекта: 15 июня 2021 года.

3. Исходные данные к курсовому проекту: Автоматизированная система
управления наружного освещения помещения

4. Содержание выпускной квалификационной работы: 1) разработка системы
управления естественным и искусственным освещением; 2) выбор
комплектующих системы; 3) разработка программно–аппаратного управления
естественным и искусственным освещением

5. Перечень материалов приложения (наличие чертежей, графиков, схем,
программных продуктов, иллюстративного материала и т.п.):

Лист 1: Электрическая схема управления наружным освещением помещения

Лист 2: Светотехнический расчёт наружного освещения

Лист 3: Схема управления наружным освещением

Лист 4: Принципиальная схема Arduino UNO

Лист 5: Код программы для теста Bluetooth-модуля

Лист 6: Код программы для работы устройства


6. Дата выдачи задания: 02.11.2020

Руководитель выпускной квалификационной работы: Штыкин Михаил

Дмитриевич, доцент, канд. тех. наук

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению (дата): 02.11.2021


(подпись студента)

РЕФЕРАТ

Выпускная квалификационная работа содержит 72 страницы, 6 приложений, 21 рисунок, 1 таблицу, 22 источника.

УМНЫЙ ДОМ, СИСТЕМА, УПРАВЛЕНИЕ, КОНТРОЛЬ, ОСВЕЩЕНИЕ, ARDUINO, ДАТЧИК, МИКРОКОНТРОЛЛЕР.

Решаемые задачи в рамках выпускной квалификационной работы:

- разработать систему управления естественным и искусственным освещением;
- провести выбор комплектующих системы
- основные требования к разрабатываемой конфигурации системы:
- обеспечить программно – аппаратное управление естественным и искусственным освещением;
- спроектировать систему для конкретного сооружения.

По ходу выполнения выпускной квалификационной работы был произведен анализ существующих систем, выявлены их плюсы и минусы, обоснован выбор принятого решения. Созданы схемы подключения датчиков и исполнительных устройств к микроконтроллеру (МК). Разработана программа, регулирующая автоосвещение.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1. История появления микропроцессоров	9
1.1 Микроконтроллеры Arduino	12
1.2 Характеристика микропроцессора	20
1.3 Графические пакеты проектирования устройств с микроконтроллером	23
2. Светотехнический расчет наружного освещения	27
2.1 Электротехнический расчет освещения	33
3. Методы управления освещением	38
3.1 Удалённое управление светом	38
3.2 Автоматическое управление	38
3.3 Ручное управление	40
4. Контроль освещения	41
4.1 Устройства контроля	41
4.2 Методы дистанционного управления светом	42
5. Разработка схемы управления освещением помещения	45
5.1 Средства разработки и адаптации	46
5.2 Графические пакеты разработки устройств на микроконтроллерах	47
5.3 Код программы для теста Bluetooth-модуля	49
6. Разработка программы управления освещением	52
6.1 Приложение по управлению на Android	57
ЗАКЛЮЧЕНИЕ	63
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	66
ПРИЛОЖЕНИЕ А	64
ПРИЛОЖЕНИЕ Б	65
ПРИЛОЖЕНИЕ В	66
ПРИЛОЖЕНИЕ Г	67
ПРИЛОЖЕНИЕ Д	68
ПРИЛОЖЕНИЕ Е	69

СПИСОК СОКРАЩЕНИЙ

ЭУ – Электронное устройство

ЭВМ – Электронная вычислительная машина

МП – Микропроцессор

ТЗ – Техническое задание

ОЗУ – Оперативная память

ПЗУ – Постоянное запоминающее устройство

ПДУ – Пульт дистанционного управления

АВП – Аппаратно–вычислительная платформа

CPU – Центральный процессор или Центрально обрабатывающей устройство

Arduino – Электронные платы или Микроконтроллеры

IDE – программная оболочка Arduino

Fritzing и AutoCAD – графические среды

ШИМ – широтно–импульсная модуляция – процесс управления мощностью, подводимой к нагрузке, путём изменённой скважности импульсов

АЦП – аналого–цифровой преобразователь – устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал).

ИМС – интегральная микросхема

XRef – механизм внешних ссылок

ВВЕДЕНИЕ

Тема выпускной квалификационной работы «Автоматизированная система управления освещением помещения». Выполнение подобной работы включает обязательную разработку технического задания, проектирование схем в различных графических средах, сборку прототипа ЭУ, написание и отладку кода программы и сборку самого устройства, что может являться полезным опытом и позволит проверить знания в разных областях.

Актуальность темы: увеличивающаяся с каждым годом потребность в использовании микропроцессорных систем для выполнения разных задач.

Объектом исследования является использование электронных устройств в повседневной жизни.

Предметом исследования является использование микроконтроллеров Arduino для управления светом в помещении.

В данной работе исследуется разработка системы управления освещением помещения на базе микроконтроллера Arduino, будет исследована и проанализирована аппаратная и программная часть. На базе исследования аппаратной части будут подобраны оптимальные компоненты и расширения для разработки устройства приёма сигнала. В исследовании программной части будут использованы программные и графические среды с целью создания проекта.

Задачи:

- разработать электронное устройство, соответствующее требованиям задания;
- произвести работу в среде IDE, написать программу для микроконтроллера;
- создать схемы в графической среде Fritzing и AutoCAD.

В теоретической части проводится изучением используемых источников по платформе Arduino UNO и Arduino Nano. На основе полученных в ходе исследований данных исследуются элементы платформы Arduino UNO и Arduino Nano.

Микроконтроллеры Arduino ориентированы на необученных пользователей, но и опытные разработчики используют эти микроконтроллеры для построения простых систем автоматики и робототехники, а также в различных проектах. Помимо микроконтроллера Arduino существует большое количество других семейств микроконтроллеров от других разработчиков, которые или сами создавали свои версии микроконтроллеров, или могли скачать с официального сайта компании Arduino чертежи продукции и выпустить микроконтроллер по собственному образцу. В компании Arduino оставили полностью открытую архитектуру, что позволяет дополнять ряд совместимых печатных плат для расширения функционала микроконтроллера Arduino. Микроконтроллеры Arduino используются во многих направлениях. Можно создать радиоуправляемого робота или умный дом. И все это благодаря платам расширения функционала, количество которых постоянно растет. Исследуя микроконтроллер Arduino UNO и Nano, можно определить поддерживаемые интерфейсы и архитектуру микропроцессора, а также на каком языке программируется микропроцессор в микроконтроллерах Arduino UNO и Nano. Проанализировав возможные варианты, выберем несколько отличающихся друг от друга устройств для более оптимального использования.

В практической части с помощью полученных в ходе исследования данных будет разработана схема устройства, удалённого управления освещением помещения на базе микроконтроллеров Arduino UNO и Arduino Nano и будет написана программа, в которой будет осуществляться управление устройством через микроконтроллер, который в свою очередь будет получать сигналы с дополнительного устройства (смартфона), и будет преобразовывать поступающую информацию с каналов управления, для выполнения предписанных функций устройства.

Результатами исследования являются принципиальные схемы, программы управления и смонтированное ЭУ.

1 ИСТОРИЯ ПОЯВЛЕНИЯ МИКРОПРОЦЕССОРОВ

В наше время микропроцессоры занимают важную позицию в жизни человечества. Они почти везде, в компьютерах, в программируемых станках, в автомобилях и так далее.

Начало развития микропроцессоров относится к 1959 году. Инженеры фирмы «Texas Instruments» разработали способ, как разместить внутри одного полупроводникового кристалла несколько транзисторов и соединить их между собой – так появилась первая ИМС. По сравнению с функционально теми же устройствами, собранными из отдельных транзисторов, резисторов и тому подобного, ИМС обладает значительными преимуществами: малыми габаритами, более высокой надежностью и так далее. Неудивительно, что количество выпускаемых микросхем стало возрастать, а их ассортимент расширяться. Последнее обстоятельство создавало ряд трудностей для потребителей. Важно даже не столько то, что стремительно возраставшее количество типов ИМС затрудняло ориентацию в море наименований. Значительно большим недостатком была узкая специализация ИМС, из-за которой объем их выпуска не мог быть большим, а значит стоимость одной микросхемы оставалась высокой. Улучшить ситуацию позволило бы создание универсальной логической ИМС, специализация которой определялась бы не заложенной на заводе внутренней структурой, а заданной непосредственно самим потребителем программой работы.

Первые МП появились в целях создания более дешевой логической микросхемы, легко адаптируемой к потребностям пользователя.

Летом 1969 г. японская компания «Busicom», разрабатывавшая новое семейство калькуляторов, обратилась за помощью в фирму «Intel». К тому времени «Intel» просуществовала всего около года, но уже проявила себя созданием самой емкой на тот момент микросхемы памяти. Фирме «Busicom» как раз и требовалось изготовить микросхемы, содержащие несколько тысяч

транзисторов. Для реализации совместного проекта был привлечен инженер фирмы «Intel» М.Хофф. Он познакомился с разработками «Busicom» и предложил альтернативную идею: вместо 12 сложных специализированных микросхем создать одну программируемую универсальную – микропроцессор. Проект Хоффа победил и фирма «Intel» получила контракт на производство первого в мире микропроцессора.

В начале 1970 г. к работе подключился Ф.Фаджин, который за 9 месяцев довел процессор от описания до кристалла (позднее Ф.Фаджин основал фирму «Zilog», создавшую замечательный 8–разрядный процессор Z80, который успешно работал во многих домашних компьютерах). В 1971 году был создан микропроцессор, вместе с ним появилось четвёртое поколение компьютеров.

Центральный процессор (CPU) – электронный блок интегральных схем, выполняющий машинные инструкции. Иногда его называют микропроцессором или просто процессором.

Главной характеристикой CPU являются: тактовая частота, энергопотребление, производительность и архитектура.

Ранние CPU создавались в виде уникальных составленных частей для единственных в своём роде компьютеров. А когда производителю потребовалось выполнение одной единственной программы, производители перешли к серийному производству типовых классов процессоров.

Создание микросхем позволило ещё сильнее усложнить CPU с одновременным уменьшением физических габаритов.

Фирма «Intel» первой разработала 4–разрядный микропроцессор (Intel 4004) в 1971 году.

Потом его сменили более мощные 8–разрядные Intel 8080 и 16–разрядные Intel 8086, которые заложили основы архитектуры всех нынешних персональных компьютеров.

Развитие пошло дальше и не стало стоять на месте, на смену 8 и 16–разрядным процессорам пришли 32–битные процессоры такие как: PentiumPro, Pentium 2, Pentium 3, IntelCore, Pentium 4 и Xeon.

Ну а уже 32–битные процессоры потеснились 64–х битными процессорами: Itanium, Itanium 2, Xeon, IntelCore 2, PentiumDualCore, Celeron DualCore, IntelCore 2 Duo, IntelCorei3, IntelCorei5, IntelCorei7, IntelCorei9, IntelXeonE3. И спустя достаточное количество лет микропроцессоры стали компактными. Одним из активно развивающихся направлений и являются «Микроконтроллеры».

Микропроцессор – процессор (устройство, отвечающее за выполнение арифметических, логических операций и операций управления, записанных в машинном коде.

Вся схемотехника разделяется на две большие области: аналоговую и цифровую. Преимущества и недостатки этих технологий известны. Аналоговая схемотехника характеризуется максимальным быстродействием, малым потреблением энергии и малой стабильностью параметров. Цифровая схемотехника обладает прекрасной повторяемостью параметров. Это привело к её развитию в последние годы.

В современном мире трудно найти область техники, где не применялись бы микропроцессоры. Они применяются при вычислениях, они выполняют функции управления, они используются при обработке звука и изображения. В зависимости от области применения микропроцессора меняются требования к нему. Это накладывает отпечаток на внутреннюю структуру микропроцессора. В настоящее время определилось три направления развития микропроцессоров:

- универсальные микропроцессоры;
- микроконтроллеры;
- сигнальные микропроцессоры.

Универсальные микропроцессоры используются для построения вычислительных машин. В них используются самые передовые решения по повышению быстродействия, не обращая особого внимания на габариты, стоимость и потребляемую энергию. В технике связи компьютеры используются для управления системами связи или устройствами связи,

обладающими большими габаритами и стоимостью. Такие компьютеры называются контроллерами.

Для управления малогабаритными и дешёвыми устройствами связи используются однокристалльные ЭВМ и микро – ЭВМ, которые в настоящее время называются микроконтроллерами. В микроконтроллерах наоборот, максимальное внимание уделяется именно габаритам, стоимости и потребляемой энергии.

Еще один класс микропроцессоров решает задачи, которые традиционно решала аналоговая схемотехника. Это сигнальные процессоры. К сигнальным процессорам предъявляются специфические требования. От них требуются максимальное быстродействие, малые габариты, легкая стыковка с аналого – цифровыми и цифро – аналоговыми преобразователями, большая разрядность обрабатываемых данных и небольшой набор математических операций, обязательно включающий операцию умножения–накопления и аппаратную организацию циклов.

1.1 Микроконтроллеры Arduino

Arduino – это удобная платформа для проектирования электронных устройств, которая плотно взаимодействует с окружающей физической средой. Платформа пользуется популярностью во всём мире благодаря своей простоте, а также открытой архитектуре и программному коду. Устройство программируется по средствам подключения USB без использования сторонних устройств (программаторов). Arduino позволяет компьютеру выйти за рамки цифровой среды и взаимодействовать с физической средой. Устройства на базе Arduino могут получать информацию об окружающей среде при помощи датчиков, также могут управлять исполнительными устройствами.

Микроконтроллеры Arduino отличаются наличием прошивочника (bootloader). Загрузчик подключается к компьютеру через интерфейс USB (если она имеется на плате Arduino) или при помощи переходника UART–USB. Организация поддержки загрузчика вшита в Arduino и выполняется в один клик мышки.

Микроконтроллер программируется языком Arduino (основан на базе языка Wiring—специальный язык программирования, основанной на C/C++) и средой разработки Arduino (основанной на среде Processing). Устройства, основанные на Arduino, могут работать самостоятельно, либо же взаимодействовать с ПО компьютера (например: Processing, MaxMSP). Платы могут быть собраны пользователями или куплены в собранном виде. ПО доступно для бесплатного скачивания.

Существует несколько видов платформ Arduino:

— последняя изданная версия на данный момент Leonardo основана на микроконтроллере ATmega32u4. Arduino UNO, построена на основе микроконтроллера Atmel ATmega328;

— более ранние версии основаны на ATmega8.

Основные форм-факторы:

— dinoxxx — стандартный размер, 20 входов-выходов, полная вместимость;

— dinoMegaxxx — увеличенный размер, 70 входов-выходов;

— dinoNanoxxx — уменьшенный размер, 22 вход-выход;

— dinoMinixxx — ещё меньший размер, 22 входа-выхода, но не имеет USB.

Есть серийные модели, производимые компаниями:

— ArduinoUSB 9 моделей;

— ArduinoSerial 2 модели;

— ArduinoMega 1 модель;

— ArduinoNano 2 модели;

— ArduinoBluetooth 1 модель;

— ArduinoLilypad 4 модели.

Есть множество различных дополнительных расширений и компонентов для микроконтроллеров Arduino, которые расширяют возможности микроконтроллера. Самым главным считается, макетная плата по которой через провода передаются сигналы с микроконтроллера Arduino, что позволяет использовать микроконтроллер с неограниченным удобством, увеличивая при

этом количество используемых pin во время работы устройства. Shield–плата – это решение для частных задач, которые встают перед разработчиками аппаратного обеспечения. В пример таких задач, можно взять, передачу данных и работу через Ethernet. Новые компоненты с лёгкостью устанавливаются на Arduino, через контакты pin, образуя при этом крепкую конструкцию. Конструкция, воссозданная в процессе установки дополнительных расширений, повышает мощность Arduino по её характеристикам.

В процессе обучения обычно используются: светодиоды, резисторы и кнопка. Так же понадобятся резисторы, которые необходимы для снятия напряжения на компоненты, так как светодиоды и прочие компоненты могут сгореть, если питание на них будет выше их технических характеристик. Следовательно, компоненты расширения для Arduino подбираются под индивидуальное задание.

Для того чтобы создать систему удалённого управления светом при помощи Arduino, нам потребуются: программный код и компоненты расширения для Arduino, и сама Arduino.

Использование макетной платы (рисунок 1.1) значительно упростит сборку устройства.

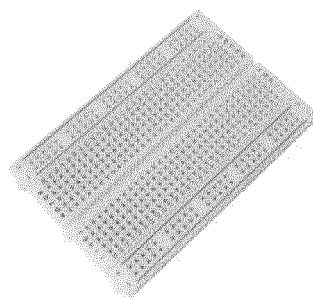


Рисунок 1.1 – Макетная плата

Программное обеспечение микроконтроллеров Arduino состоит из программной оболочки IDE для записи программ, которая осуществляет компиляцию и программирование разработанных устройств. Компиляция – это транслирование программы, составленной на языке программирования высокого уровня, в эквивалентную программу низкоуровневого языка

программирования, который близок к машинному коду (языку ассемблера), выполняемое компилятором. Полный цикл программирования проводится через ардуиновскую программную оболочку (рисунок 1.2), которая находится в бесплатном доступе на сайте разработчиков Arduino.

В этой оболочке имеется: текстовый редактор, процессор, менеджер проектов, инструменты загрузки программы и компилятор. Оболочки, написанные на Java на основе проекта Processing, работает под Windows, MacOSX и Linux. Язык программирования Arduino— это уже знакомый и стандартный C++ (используемый им компилятор AVR–GCC).

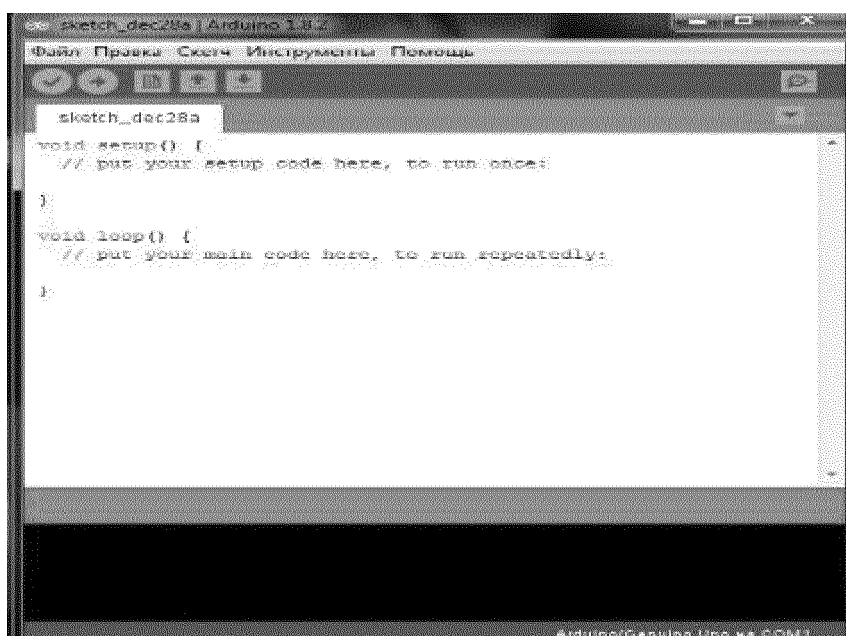


Рисунок 1.2 – Arduino IDE

Имеются особенности, облегчающие новичкам работу со средой программирования, для написания своих первых программных кодов. Программы, которые написаны для Arduino, называются «набросками». Эти программы, перед тем как отправиться на компиляцию, проходят обработку в процессоре Arduino. Обязательную в C++ функцию main () процессор Arduino создаёт сам, и записывает в неё «черновые» действия.

При написании программы для Arduino необходимо написать две

обязательные функции `–setup ()` и `loop ()`. Первая вызывается единожды при запуске, вторая выполняется в бесконечном цикле.

- `if` – используется в сочетании с операторами сравнения, проверяется достигнута ли истинность условия;

- `if else` – предоставляет более больший контроль над процессом выполнения кода, чем базовый оператор `if`, позволяя при этом проводить несколько проверок, будучи объединенными вместе;

- `while`– будет вычислять в цикле непрерывно и бесконечно до тех пор, пока выражение в скобках `()` не станет «ложным».

- `dowhile` – работает так же, как цикл `while`, исключение только в том, что условие выполняется в конце цикла, и цикл `do` всегда будет выполняться хоть один раз;

- `for` – используется для повторения блока операторов, заключённых в фигурных скобках `{}`;

- `break` – используется для принудительного выхода из цикла `do`, `for` или `while`;

- `continue` – пропускает оставшиеся операторы в текущем цикле, вместо операторов выполняется проверка условного выражения цикла;

- `goto` – условное «перемещение», выполняет переход к определённой метке в программе.

Операторы сравнения и арифметические операторы:

- «`if`» (оператор сравнения `XY`);

- «`==`» (`equalto`) – равно;

- «`!=`» (`notequalto`) – неравно;

- «`<`» (`lessthan`) – меньше;

- «`<=`» (`less than or equal to`) – меньше ИЛИ равно;

- «`>`» (`greater than`) – больше;

- «`>=`» (`greater than or equal to`) – больше ИЛИ равно;

- «`=`» (`assignment`) – присваивает значению правой переменной от оператора, левой переменной;

- «+» (addition) – сложение;
- «-» (subtraction) – вычитание;
- «*» (multiplication) – умножение;
- «/» (division) – деление;
- «%» (modulo) – возвращает остаток от деления одного целого операнда на другой.

Временные, математические и тригонометрические функции:

- millis() – возвращает количество миллисекунд с момента начала выполнения программы на плате Arduino;
- micros() – возвращает количество микросекунд с момента начала выполнения программы на плате Arduino;
- delay() – останавливает выполнение программы на заданное количество миллисекунд;
- delayMicroseconds() – останавливает выполнение программы на заданное количество микросекунд;
- min() – возвращает меньшее из двух сравниваемых значений;
- max() – возвращает большее из двух сравниваемых значений;
- abs() – возвращает модуль числа;
- map() – пропорционально переносит значение из текущего диапазона значений в новый диапазон;
- pow() – вычисляет значение, возведённое в заданную степень;
- sq() – вычисляет квадрат числа;
- sqrt() – вычисляет квадратный корень числа;
- sin() – вычисление синуса угла, заданного в радианах;
- cos() – вычисление косинуса угла, заданного в радианах;
- tan() – вычисление тангенса угла, заданного в радианах.

Функции передачи данных:

- serial – служит для связи собранного устройства на базе Arduino с компьютером или другим устройством, которое поддерживает последовательный интерфейс обмена данными.

Заголовочные файлы не надо вставлять в программу, эти заголовки используются в стандартных библиотеках. Эти файлы добавит процессор Arduino в соответствии с конфигурацией и параметрами проекта.

Но использование библиотек нужно указывать заранее. Библиотеки представлены в исходном виде текстов на стандартном языке C++, они добавляются в специальную папку в рабочем каталоге IDE и при этом названия библиотек добавляются в список библиотек в меню IDE.

Необходимо отмечать нужные библиотеки, и они будут вноситься в список компиляции. Arduino IDE не предоставляет никаких настроек компилятора и урезает многие настройки, но это защищает новичков от лишнего риска возникновения проблем в работе.

Самая простая схема на Arduino – это подключение внешнего светодиода, управление которым осуществляется с помощью программы написанной в среде IDE. Для написания программы нужно только открыть IDE и начать писать программный код, когда программа написана нужно записать её на Arduino. Для записи Arduino должна быть подключена к компьютеру с установленными на него драйверами, если все условия предустановки выполнены, то программа запишется на микроконтроллер и можно приступить к выполнению всех поставленных задач на плате.

Программирование микроконтроллера осуществляется по средствам среды программирования и программатора. Программатор – это аппаратно–программное устройство, предназначенное для записи и считывания информации в постоянное запоминающее устройство (ПЗУ). Для исследуемого нами микропроцессора программатор не требуется. Но если есть необходимость прошить микроконтроллер, то можно прошить программу в отдельный чип Atmegf328 с помощью ArduinoISP.

Для прошивки микропроцессора надо:

- записать на Arduino программу ArduinoISP;
- подключить Arduino и чип Atmega328;
- включить Arduino;

- включить autoreset на Arduino;
- соединить RST и +5V через резистор на 1150м;
- выбрать соответствующую плату в настройках;
- нажать Burnbootloader;
- написать программу;
- нажать Upload using programmer.

В итоге получится, что программа записана вместо Bootloader.

Микропроцессор выполняет только то, что описывают команды, которые составляют программу, именно по этой причине, для решения какой-либо задачи необходимо детально её проработать. Поскольку каждый МП имеет свою систему команд, то язык низкого уровня ориентирован на конкретные МП и поэтому называется машинно – ориентированным языком.

Язык низкого уровня, использующий команды в машинах кодах, называется машинным языком. Программа, написанная на данном языке, называется объектной программой. Это единственная форма программы доступная пониманию МП. Однако писать программы на машинном языке неудобно и сложно. Язык низкого уровня, в котором используются символические обозначения имён адресов и операнды, называется языком ассемблера. При работе с микропроцессором программирование на языке ассемблера является наиболее распространенным.

Помимо языков низкого уровня существуют также языки высокого уровня. В этих языках команды, представляемые условными обозначениями, определяют действия, ориентированные на самое эффективное решение определённых задач. Например, программирование на электронно-вычислительных машинах (ЭВМ), используется язык высокого уровня БЕЙСИК. Этот язык предназначен для решения задач направленных на понимание математических функций – то есть на инженерные задачи. Для записи программы в микропроцессор Arduino используется язык СИ. Этот язык удобен для создания прикладных программ с учётом особенностей аппаратных средств. После написания программы, она компилируется и загружается в

память микропроцессора.

1.2 Характеристика микропроцессора

Atmel 8–разрядный AVR RISC –На микроконтроллер объединяет 32 кБ ISP флэш – памяти с возможностями чтения во время записи,–1 кБ EEPROM , 2 кБ SRAM , 23 общего назначения линий ввода / вывода, 32 общего назначения рабочих регистров, три гибких таймера / счетчики с режимами сравнения, внутренние и внешние прерывания, последовательный программируемый USART, 2–проводной последовательный интерфейс байт–ориентированный, SPI последовательный порт, 6–канальный 10–разрядный аналого–цифровой преобразователь (8– ми каналов в TQFP и QFN пакетов), программируемый сторожевой таймер с внутренним генератором и пять программных режимов экономии по выбору мощности. Устройство работает в диапазоне 1.8–5.5 вольт. Устройство обеспечивает пропускную способность, приближающуюся к 1 MIPS на МГц. Его характеристики:

- высокая производительность;
- низкое энергопотребление;
- улучшенная RISC–Архитектура;
- 131 мощная команда большинство которых выполняется за один такт ЦПУ;
- 32– х 8 регистра общего назначения;
- полностью статическая операция;
- производительность до 20 миллионов команд в секунду на 20 мГц ЦПУ;
- внутрикристальный 2 цикловой множитель;
- энергонезависимая память данных и программ;
- 4/8/16/32 кБ внутрисистемной энергонезависимой ФЛЭШ памяти программ;
- 256/512/512/1 кБ EEPROM ПЗУ;
 - количество циклов запись / стирание: 10000 Flash / EEPROM 100000;
- хранение данных: 20 лет при температуре 85 ° С / 100 лет при

температуре 25 ° C;

- дополнительный загрузочный раздел независимыми блокировочными битами;
- в системе программирования внутренних загрузочных программ;
- истинность Read– While;
- Write операции;
- программная блокировка для обеспечения безопасности;
- поддержка библиотеки Atmel® QTouch®;
- ёмкостные сенсорные кнопки, слайдеры и колеса прокрутки;
- технологии QTouch и QMatrix®;
- до 64 сенсорных канала;
- два 8 – битных таймера/счетчика с отдельным предделителем частоты и режимом сравнения;
- 16 – битный таймер/счетчик с отдельным предделителем частоты, режимом сравнения и режимом захвата;
- счетчик реального времени с отдельным генератором;
- 6 ШИМ–каналов;
- 8–канальный 10–разрядный АЦП в корпусах TQFP и QFN / MLF;
- измерение температуры;
- 6 канальный 10–разрядный АЦП в корпусе PDIP;
- измерение температуры;
- программируемый последовательный интерфейс USART;
- последовательный SPI Интерфейс Master / Slave;
- байтно–ориентированный последовательный интерфейс (совместим с I2C Philips);
- программируемый сторожевой таймер со встроенным генератором;
- встроенный аналоговый компаратор;
- прерывание и пробуждение по изменению на выводах;
- схема сброса при подаче питания и программируемое обнаружение провалов по напряжению;

- внутренний калиброванный генератор;
- 6 режимов сна: холостой ход, снижение шумов АЦП, экономии энергии, выключение питания, режим ожидания и расширенный режим ожидания;

Ввод / вывод и типы корпусов:

- 23 программируемые линии ввода / вывода;
- корпус PDIP 28 выводов, корпус TQFP 32 вывода, корпус QFN /MLF с 28

и 32 выводами.

Рабочее напряжение:

- от 1,8 до 5,5 В.

Температурный диапазон:

- от -40°C до 85°C .

Производительность:

– 0 – 4 МГц ПРИ 1.8 – 5.5 В, 0 – 10 МГц ПРИ 2,7 – 5,5 В, 0 – 20 МГц при 4,5;

- потребляемый ТОК ПРИ 1 МГц, 1,8 В, 25°C ;

– активный режим: 0,2 мА;

–режим отключения: 0,1 мкА;

- режим энергосбережения: 0,75 мкА (Включая 32 кГц RTC).

Принцип работы микропроцессора в Arduino Nano и Arduino UNO

В рамках исследования были взяты платы Arduino Nano и Arduino UNO.

На этих платах микроконтроллеров, в которых будут записываться программные коды, это ATmega328 (ПРИЛОЖЕНИЕ В). Архитектура микропроцессора имеет сокращённый набор команд.

Для резервного проекта взят микроконтроллер Arduino UNO. На этой плате микроконтроллер, в который будут записываться программные коды, так же, как и у Arduino Uno является ATmega328. Архитектура микропроцессора имеет сокращенный набор команд.

RISC – это так же архитектура процессора, в которой быстроедействие, взаимно пропорциональная меньшему времени исполнения, что облегчает повышения тактовой частоты и делает более эффективной.

На (рисунке 1.3) изображены пины МП микроконтроллера АТmega328. На ней можно увидеть на какой пин какой сигнал подаётся и откуда сигнал выходит. 7 и 20 это pin питания микропроцессора, а 8 и 22 это pin заземления.

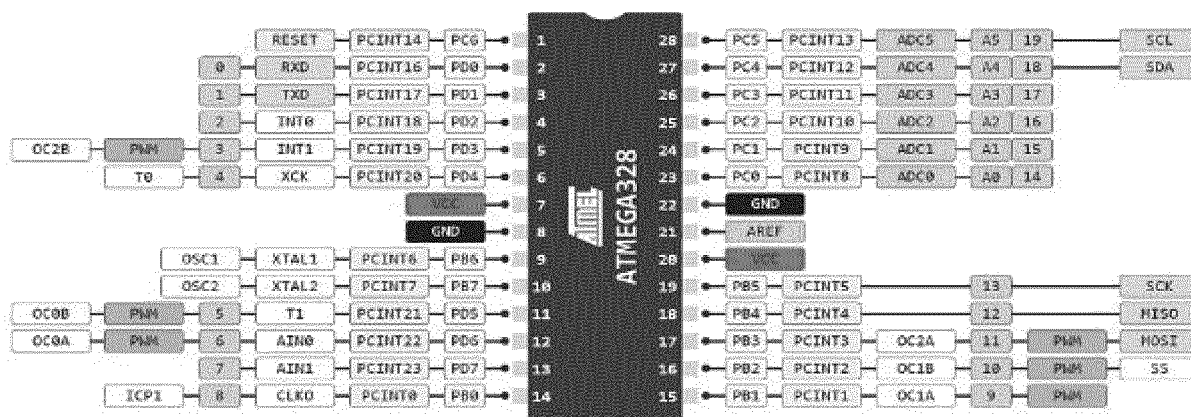


Рисунок 1.3– Pin МП

Большая часть контактов предназначена для ввода и вывода информации цифровой и аналоговой PC6/RESET.

Если RSTDISBL является запрограммированным предохранителем, PC6 в этом случае используется как линия ввода/вывода. Следует отметить, что электрические характеристики.

PC6 отличаются от других штырей Портов С.

Характеристики:

- тактовая частота: 0–20 МГц;
- объём SRAM–памяти: 2 кб;
- объём Flash–памяти: 32 кб;
- напряжение питания: 1,8–5,5 В;
- потребляемый ток в режиме работы: 0,2 мА (1 МГц, 1,8 В);
- потребляемый ток в режиме сна: 0,675 мкА (1 МГц, 1,8 В);
- общее количество портов: 23;
- количество ШИМ (PWM) выходов: 6;

ШИМ – широтно–импульсная модуляция – процесс управления мощностью, подводимой к нагрузке, путём изменения скважности импульсов.

- количество каналов АЦП входов (аналоговые): 6;

АЦП – аналого–цифровой преобразователь – устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал).

– количество аппаратных USART: 1;– разрешение АЦП: 10 бит.

USART/serial – универсальный асинхронный приёмопередатчик – узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами.

– количество аппаратных SPI: 1;

SPIMaster/slave – последовательный синхронный стандарт передачи данных в режиме полного дуплекса, предназначенный для высокоскоростного сопряжения микроконтроллеров.

– количество аппаратных PC/SPI: 1.

PC–шина – последовательная асимметричная шина – предназначенная для связи между интегральными схемами внутри электронных приборов.

1.3 Графические пакеты проектирования устройств с микроконтроллером

AutoCAD – двух и трёхмерная система автоматизированного проектирования и черчения, разработанная компанией Autodesk. Первая версия системы была выпущена в 1982 году. AutoCAD и специализированные приложения на ее основе нашли широкое применение в машиностроении, строительстве, архитектуре и других отраслях промышленности. Программа выпускается на 18 языках. Уровень локализации варьирует от полной адаптации до перевода только справочной документации. Русскоязычная версия локализована полностью, включая интерфейс командной строки и всю документацию, кроме руководства по программированию.

Интерфейс AutoCAD (рисунок 1.4)

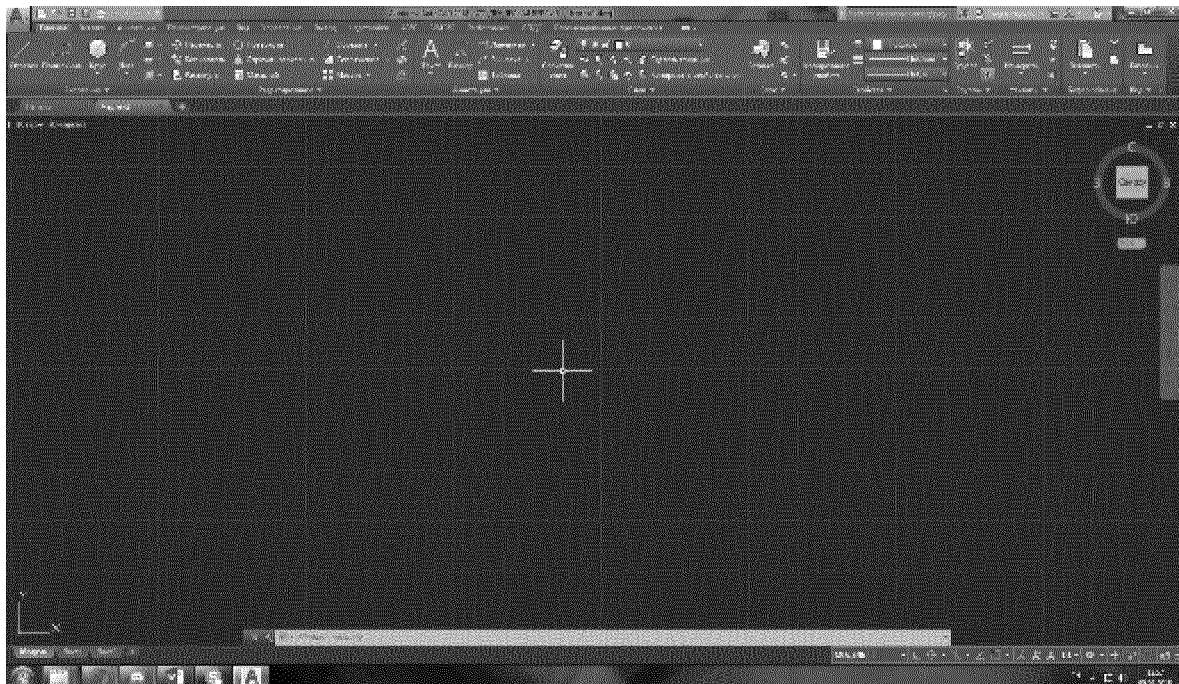


Рисунок 1.4– Интерфейс AutoCAD

Ранние версии AutoCAD оперировали небольшим числом элементарных объектов, такими как круги, линии, дуги и текст, из которых составлялись более сложные. В этом качестве AutoCAD заслужил репутацию «электронного кульмана», которая остаётся за ним и поныне. Однако на современном этапе возможности AutoCAD весьма широки и намного превосходят возможности, которые были у него ранее.

В области двухмерного проектирования AutoCAD по-прежнему позволяет использовать элементарные графические примитивы для получения более сложных объектов. Кроме того, программа предоставляет обширные возможности работы со слоями и аннотативными объектами: размерами, текстом, обозначениями. Использование механизма внешних ссылок XRef позволяет разбивать чертёж на составные файлы, за которые ответственны различные разработчики, а динамические блоки расширяют возможности автоматизации 2D-проектирования обычным пользователем без использования программирования.

Версия программы AutoCAD 2014 включает в себя полный набор

инструментов для комплексного трёхмерного моделирования, поддерживается твердотельное, поверхностное и полигональное моделирование. AutoCAD позволяет получить высококачественную визуализацию моделей с помощью системы рендеринга mentalray. Также в программе реализовано управление трёхмерной печатью, результат моделирования можно отправить на 3D-принтер, а поддержка облаков точек позволяет работать с результатами 3D-сканирования. Следует отметить, что отсутствие трёхмерной параметризации не позволяет AutoCAD напрямую конкурировать с машиностроительными САПР среднего класса, такими как Inventor, SolidWorks и другими. В состав AutoCAD 2012 включена программа InventorFusion, реализующая технологию прямого моделирования.

2 СВЕТОТЕХНИЧЕСКИЙ РАСЧЕТ НАРУЖНОГО ОСВЕЩЕНИЯ

Расчёт освещения на прямых участках дороги

Для проезжей части улиц и площадей категорий Б и В рекомендуются осветительные приборы с лампами НЛВД. Выбор светильников, НО производится в соответствии с техническим заданием на проектирование и с учётом генплана объекта. В зависимости от значимости освещаемого объекта и интенсивности движения определяется нормируемое значение L_{cp} , с учётом размеров освещаемой поверхности и выбранного способа размещения осветительных приборов (ОП) устанавливаются их светораспределение, тип и мощность используемого источника света. При выборе типа и мощности источника света для НО, необходимо учитывать спектральный состав излучения, а также определять экономическую целесообразность его применения в ОУ.

При нормировании установок, НО по средней освещённости определение расстояния между ОП (шага d , м) производится по формуле

$$d = \frac{1}{E_n b K_3} \sum_{i=1}^M u_{Ei} \Phi_{ли} m_i, \quad (2.1)$$

где: E_n – нормируемая средняя освещённость, лк;

b – ширина проезжей части, м;

K_3 – коэффициент запаса равный 1,3 для ОП с ЛН и 1,5 с ГЛ;

M – число рядов ОП вдоль освещаемой полосы (каждый ряд должен состоять из однотипных ОП);

u_{Ei} – коэффициент использования по яркости ОП 1–го ряда;

$\Phi_{ли}$ – световой поток ламп ОП i –го ряда;

m – число ОП на опоре, относящейся к 1–му ряду.

Дороги имеют асфальтное покрытие, относятся к разряду дорог местного значения с наибольшей интенсивностью движения транспорта в обоих направлениях 500 и более, для которых среднее значение горизонтальной

освещённости составляет $E_H=6$ лк. Ширина проезжей части составляет $b=7,5$ м, тротуара $2t=2$ м, рис. 2.1.

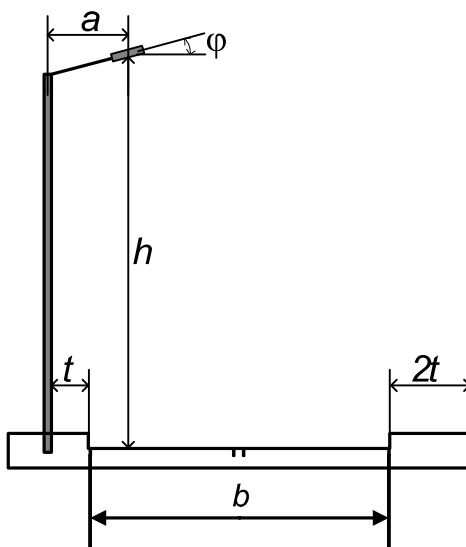


Рисунок 2.1 – Схема размеров дороги

Для наружного освещения используем светильник ЖКУ12–100–101 с натриевой лампой высокого давления ДНАТ–100, мощностью $P_{\text{л}}=100$ Вт и световым потоком $\Phi_{\text{л}}=10000$ лм. Учитывая ширину дорог, по одному светильнику $m=1$ устанавливается на опорах с одной стороны проезжей части, число рядов $M=1$. Высота опор составляет $H=10$ м, вылет светильника $a=2$ м, отношение $b/H=7,5/10=0,75$, при этом значении величина коэффициента использования для выбранного светильника $u_E=0,27$.

Затем по (2.1) определяем расстояние между опорами

$$d = \frac{0,27 \cdot 10000 \cdot 1}{6 \cdot 10 \cdot 1,5} = 40 \text{ м}$$

Принимаем расстояние между опорами $d=40$ м.

Используем точечный метод расчёта для проверки средней освещённости проезжей части. Определение равномерности распределения освещённости производится расчётом средней мощности на участке размерами $d \times b$ в n контрольных точках, находящихся на расстоянии не более 5 м одна от другой и равномерно расположенных на этом участке, тогда средняя освещённость определяется как

$$E_{\text{ср}} = \frac{\sum_{i=1}^n E_i}{n}, \quad (2.2)$$

где E_i – освещённость в контрольной точке, лк.

Освещённость в контрольной точке от одного светильника находится по выражению

$$E_1 = \frac{e_{\alpha 100} I'_{\alpha} \Phi_{\text{св}}}{1000 \text{ лм} \cdot 100 \text{ кд} \cdot K_3}, \quad (2.3)$$

где $e_{\alpha 100}$ – освещённость, создаваемая силой света $I_{\alpha}=100$ кд, действующей по направлению α ;

I'_{α} – значение силы света светильника по его типовой кривой либо по каталогу для угла α ;

$\Phi_{\text{св}}$ – световой поток светильника.

Пример расчёта освещённости в контрольной точке А, рис. 2.2.

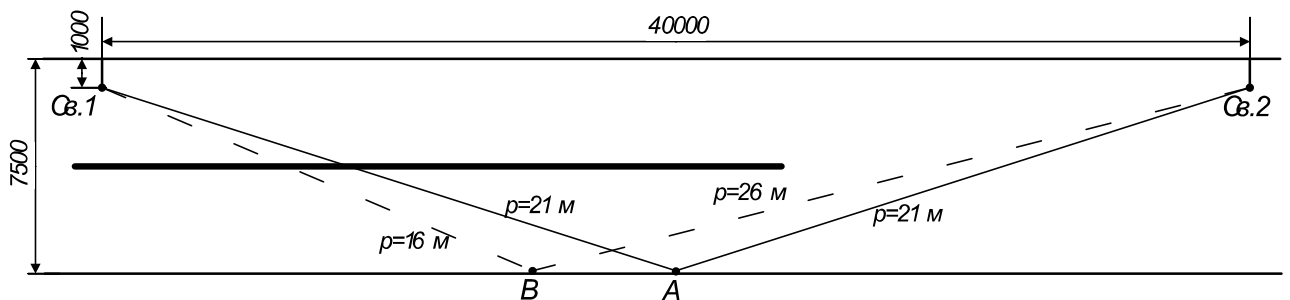


Рисунок 2.2 – Расположение контрольных точек

1. Определяем расстояния h и r от 1–го светильника: $h=10$ м, $r=21$ м.
2. Находим угол $\alpha=64,5^\circ$ и освещённость, создаваемой силой света $I_{\alpha}=100$ кд, $e_{\alpha 100}=0,08$ лк.
3. Для угла $\alpha=64,5^\circ$ по типовой кривой светильника, рис. 2.3, находим значение силы света $I'_{\alpha}=365$ кд.

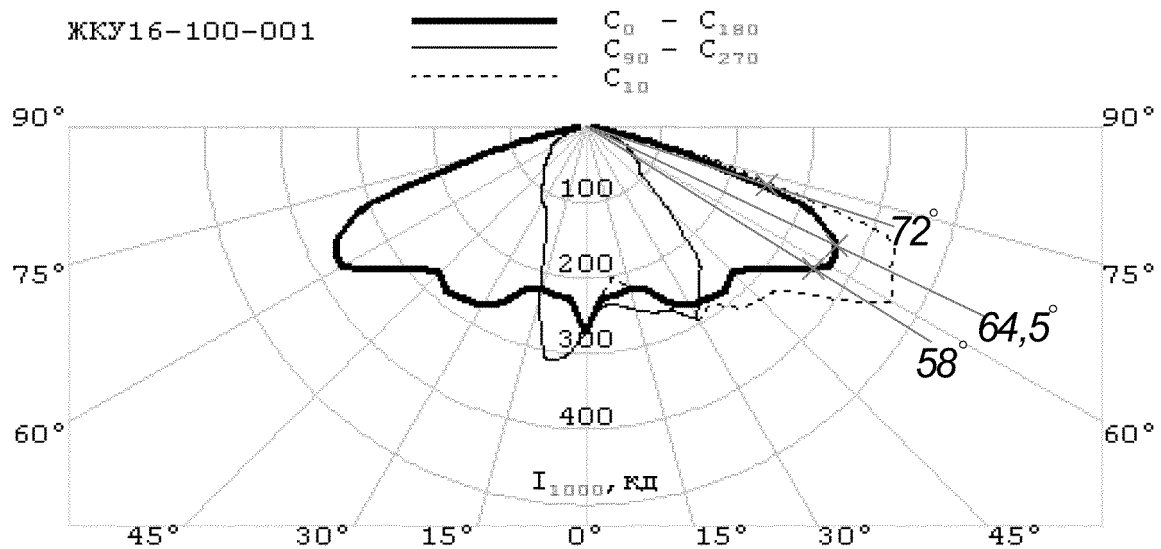


Рисунок 2.3 – Кривая силы света светильника

4. Определяем по (2.3) освещённость в контрольной точке А от Св.1:

$$E_{A1} = \frac{0,08 \cdot 365 \cdot 10000}{1000 \cdot 100 \cdot 1,5} = 1,95 \text{ лк}$$

5. Находим освещённость в контрольной точке А от светильника Св.2, которая составит $E_{A2} = 1,95$ лк.

6. Освещённость в контрольной точке А имеет значение:

$$E_A = E_{A1} + E_{A2} = 1,95 + 1,95 = 3,9 \text{ лк.}$$

Аналогично подсчитываем освещённость в контрольной точке В, от светильников Св.1 и Св.2, рис.2 и 3:

– освещённость в точке В от Св.1: $h=10$ м, $p=16$ м, $\alpha=58^\circ$, $e_{\alpha 100}=0,149$ лк, $I'_\alpha=350$ кд, $E_{B1}=3,5$ лк;

– освещённость в точке В от Св.2: $h=10$ м, $p=26$ м, $\alpha=72^\circ$, $e_{\alpha 100}=0,046$ лк, $I'_\alpha=250$ кд, $E_{B2}=0,8$ лк;

– освещённость в точке В от светильников: $E_B = 3,5 + 0,8 = 4,2$ лк.

Таким же образом находятся освещённости в остальных контрольных точках (узлах расчётной сетки), табл.1.

Таблица 1 – Таблица значений освещённости в узлах расчётной сетки, лк

		Расстояние от проекции светильника на горизонтальную плоскость до контрольной точки по оси X, м													
		2	4,5	7	10	13	16	19	21	24	27	30	33	36	39
по оси Y, м	6,5	9,1	8,4	5,6	3,5	2,2	2,5	2,6	2,6	2,5	2,2	3,5	5,6	8,4	9,1
	5,5	13,4	10,6	7,6	5,2	4,3	4,1	3,9	3,9	4,1	4,3	5,2	7,6	10,6	13,4
	4,5	15,1	12,3	9,7	6,7	5,4	4,7	4,4	4,4	4,7	5,4	6,7	9,7	12,3	15,1
	3	11,4	11,3	9,5	6,8	5,3	4,7	4,4	4,4	4,7	5,3	6,8	9,5	11,3	11,4
	2	11,1	10,3	8,8	7,3	5,5	5	4,7	4,7	5	5,5	7,3	8,8	10,3	11,1
	0,5	9,4	8,3	7,4	6,5	5,3	5,1	4,9	4,9	5,1	5,3	6,5	7,4	8,3	9,4

Затем используя значения освещённости в контрольных точках по (2) подсчитывается средняя освещённость. Из табл.1 сумма освещённости в

контрольных точках $\sum_{i=1}^n E_i = 588,6$ лк, число точек $n = 14 \cdot 6 = 124$, средняя

освещённость $E_{cp} = 588,6 / 124 = 7,01$ лк. Средняя освещённость установки наружного освещения не ниже нормируемой: $E_{cp} = 7,01 \text{ лк} \geq E_n = 6 \text{ лк}$.

Освещённость в контрольных точках показана на рис. 2.4, рассчитанном в компьютерной программе.

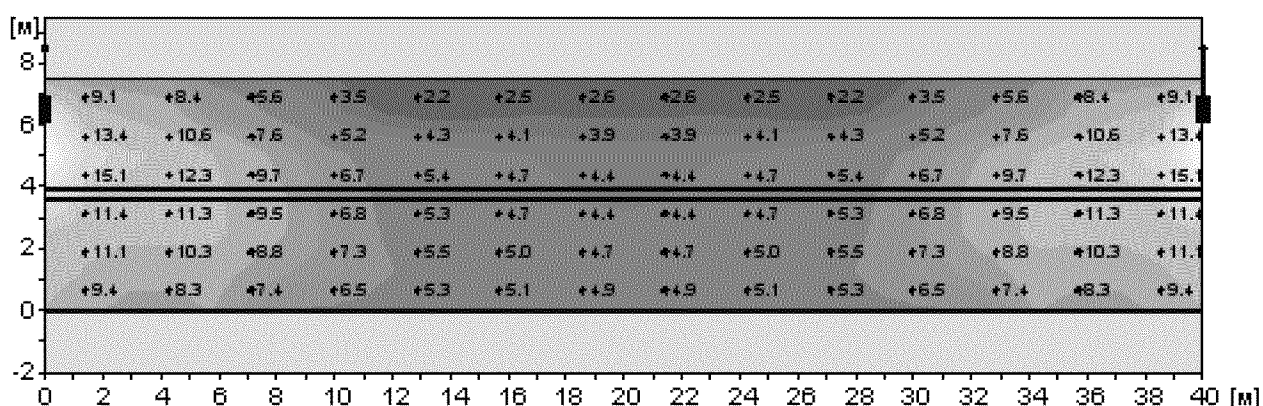


Рисунок 2.4 – Освещённость в контрольных точках

Расчёт освещения на закруглениях и пересечениях дорог

Закругления и пересечения дорог являются конфликтными участками, поэтому при проектировании освещения необходимо использовать все возможности для повышения безопасности.

На перекрёстках улиц, пешеходных переходах и железнодорожных переездах для обеспечения безопасности движения светильники рекомендуется располагать в соответствии с рис. 2.5а. При одностороннем расположении светильников на обеих пересекающихся улицах в зоне перекрёстка необходимо устанавливать не менее двух дополнительных светильников.

На примыканиях улиц один из светильников основной улицы или дороги независимо от схемы расположения светильников на ней должен размещаться на противоположной стороне перекрёстка по линии зрения подъезжающего водителя, рис.5б.

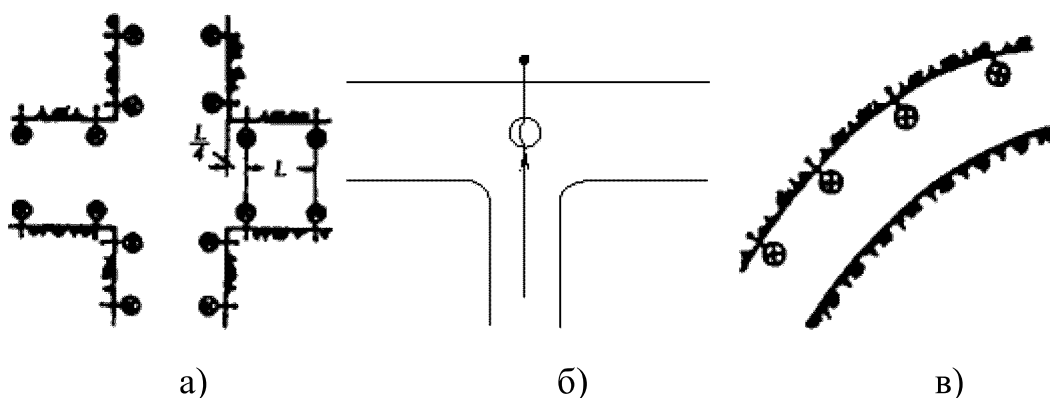


Рисунок 2.5 – Расположение светильников

На закруглениях улиц и дорог с радиусом кривизны в плане по оси проезжей части 60–250 м осветительные приборы (ОП) при одностороннем расположении должны размещаться по внешней стороне улицы, рис. 5в, а в случае необходимости размещения опор по внутреннему радиусу они должны устанавливаться с шагом $d=0,55h$.

Расположение светильников с учётом особенностей размещения на перекрёстках, примыканиях и закругления дорог показаны на листе 2.

2.1 Электротехнический расчет

Освещение различных по назначению участков территорий рекомендуется питать самостоятельными линиями, начиная от магистральных щитков, размещаемых в помещениях трансформаторных подстанций. Линии уличного освещения рекомендуется выполнять трёхфазными трёхпроводными.

Для питания осветительной сети применяется система трёхфазного

переменного тока с глухим заземлением нейтрали напряжением 380/220 В, линии и отдельные участки выполняются четырёхпроводными (три фазы и нуль).

Допустимые потери напряжения в сети:

Величина допустимых потерь напряжения $\Delta U_{\text{доп}}$ в сети определяется по выражению

$$\Delta U_{\text{доп}} = U_{\text{х.х}} - U_{\text{мин}} - \Delta U_{\text{т}}, \quad (2.4)$$

где $U_{\text{х.х}}$ – номинальное напряжение при холостом ходе трансформатора, %;

$U_{\text{мин}}$ – допускаемое напряжение у наиболее удалённых ламп, %;

$\Delta U_{\text{т}}$ – потеря напряжения в трансформаторе, приведённая ко вторичному напряжению, %.

Потеря напряжения $\Delta U_{\text{т}}$ зависит от мощности трансформатора, его загрузки β , коэффициента мощности $\cos\varphi$ питаемых электроприёмников и приближённо определяется по формуле

$$\Delta U_{\text{т}} = \beta(U_{\text{а.т}} \cos\varphi + U_{\text{р.т}} \sin\varphi). \quad (2.5)$$

Значения $U_{\text{а.т}}$ и $U_{\text{р.т}}$ определяются по формулам

$$U_{\text{а.т}} = \frac{P_{\text{к}}}{S_{\text{н}}} \cdot 100; \quad (2.6)$$

$$U_{\text{р.т}} = \sqrt{U_{\text{к}}^2 - U_{\text{а.т}}^2}, \quad (2.7)$$

где $P_{\text{к}}$ – потери короткого замыкания, кВт;

$S_{\text{н}}$ – номинальная мощность трансформатора, кВА;

$U_{\text{к}}$ – напряжение короткого замыкания, %.

Параметры трансформатора питающего осветительную установку $S_{\text{н}}=400$ кВА, $P_{\text{к}}=5,5$ кВт, $U_{\text{к}}=4,5$ %, активная составляющая напряжения короткого замыкания:

$$U_{\text{а.т}} = \frac{5,5}{400} \cdot 100 = 1,37 \%;$$

реактивная составляющая:

$$U_{\text{р.т}} = \sqrt{4,5^2 - 1,37^2} = 4,29 \%.$$

Потеря напряжения в трансформаторе КТП при загрузке $\beta=0,8$ и коэффициенте мощности нагрузки $\cos\varphi=0,97$:

$$\Delta U_T = 0,8(1,37 \cdot 0,97 + 4,29 \cdot 0,24) = 1,9 \%$$

Принимаем $U_{x.x} = 105 \%$, рассчитаем допустимую потерю напряжения в осветительной сети для $U_{\min}=97,5 \%$

$$\Delta U_{\text{доп}} = 105 - 97,5 - 1,9 = 5,6 \%$$

Выбор сечений проводов осветительной сети:

Расчёт сечения проводов осветительной сети проводим на минимум металла. Учитывая, малую мощность светильников и небольшие длины участков осветительная сеть выполняется двухпроводной. Прокладка алюминиевого кабеля осуществляется в земле. По плану расположения светильников составим схему моментов однофазных групп наружного освещения для выбора сечения линий. Рассчитаем моменты нагрузки участков групповых линий и выберем сечение проводов.

Момент нагрузки представляет собой произведение мощности P на длину l линии:

$$M = P \cdot l \tag{2.8}$$

Сечение провода на участке рассчитывается по формуле, мм:

$$s = \frac{\sum M}{K_c \Delta U_{\text{доп}}}, \tag{2.9}$$

где $\sum M$ – сумма моментов данного и всех последующих по направлению энергии участков с тем же числом проводов в линии, как и на данном участке, кВт·м;

K_c – коэффициент, зависящий от схемы питания (трёх-, двух- или однофазная) и материала проводника;

$\Delta U_{\text{доп}}$ – допустимая потеря напряжения в осветительной сети, %.

Выбор сечения проводников на примере групповой линии 1, установленная мощность светильников $P=100$ Вт, длины линий определяются согласно плана.

Рассчитаем суммарные моменты на участках:

$$\begin{aligned}
 & - \text{участок } 0-1: \Sigma M_{01} = 6P \cdot l_{01} + 5P \cdot l_{12} + 4P \cdot l_{23} + 3P \cdot l_{34} + 2P \cdot l_{45} + P \cdot l_{56} = 6 \cdot 100 \cdot 18 + \\
 & + 500 \cdot 34 + 400 \cdot 20 + 300 \cdot 35 + 200 \cdot 30 + 100 \cdot 20 = 54 \text{ кВт}\cdot\text{м}.
 \end{aligned}$$

На остальных участках суммарные моменты находятся аналогично, и имеют значение: $\Sigma M_{12} = 17$ кВт·м; $\Sigma M_{23} = 8$ кВт·м; $\Sigma M_{34} = 10$ кВт·м; $\Sigma M_{45} = 6$ кВт·м; $\Sigma M_{56} = 2$ кВт·м.

Определяем сечение проводов участка 0–1 (значение $K_c = 7,4$ для однофазной осветительной сети 220 В выполненной алюминиевым проводом). Допустимую потерю напряжения на участках находим по выражению

$$\Delta U_{\text{д.уч}} = \frac{\Delta U_{\text{дон}}}{n}, \quad (2.10)$$

где n – число участков.

Итак, допустимая потеря напряжения на участках не должна быть выше

$$\Delta U_{\text{д.уч}} = \frac{5,6}{6} = 0,9 \%.$$

Сечение провода на участке 0–1:

$$s_{01} = \frac{\Sigma M_{01}}{K_c \Delta U_{\text{д.уч}}} = \frac{54}{7,4 \cdot 0,9} = 7,7 \text{ мм}^2.$$

Выбираем ближайшее стандартное сечение, равное $s_{01} = 6 \text{ мм}^2$.

Действительная потеря напряжения на участке 0–1 составляет:

$$\Delta U_{01} = \frac{\Sigma M_{01}}{K_c s_{01}} = \frac{54}{7,4 \cdot 6} = 1,2 \%.$$

Аналогично подсчитываем, суммарные моменты, выбираем сечения проводов на этих участках и определяем действительную потерю напряжения:

- участок 1–2: $s_{12} = 6 \text{ мм}^2$; $\Delta U_{12} = 1 \%$;
- участок 2–3: $s_{23} = 4 \text{ мм}^2$; $\Delta U_{23} = 0,9 \%$;
- участок 3–4: $s_{34} = 2,5 \text{ мм}^2$; $\Delta U_{34} = 1 \%$;
- участок 4–5: $s_{45} = 2,5 \text{ мм}^2$; $\Delta U_{45} = 0,4 \%$;
- участок 5–6: $s_{56} = 2,5 \text{ мм}^2$; $\Delta U_{56} = 0,1 \%$;

Подсчитываем наибольшую потерю напряжения в конце линии:

$$\Delta U_{06}=1,2+1+0,9+1+0,4+0,1=4,6 \text{ \%}.$$

Проверка выбранных сечений проводников осветительной сети по нагреву. Нагрев проводников вызывается прохождением по ним тока I , величина которого для однофазной сети определяется по формуле:

$$I = \frac{P}{U_{\phi} \cos\varphi}; \quad (2.11)$$

где P – активная мощность нагрузки (включая потери в ПРА газоразрядных ламп – 10 %);

$\cos\varphi$ – коэффициент мощности нагрузки (для газоразрядных ламп с компенсированными ПРА, $\cos\varphi=0,85$);

U_n – номинальное напряжение сети.

Проверим выбранное сечение участка 0–1 осветительной сети по длительно допустимому току.

Активная мощность протекающая по головному участку с учётом потерь мощности в ПРА ($\eta_{\text{ПРА}}=1,1$) газоразрядных ламп:

$$P_{01}=6P \cdot \eta_{\text{ПРА}}=6 \cdot 100 \cdot 1,1=660 \text{ Вт}.$$

Ток, протекающий по проводнику:

$$I = \frac{660}{220 \cdot 0,85} = 3,5 \text{ А}.$$

Участок выполнен алюминиевым кабелем АВВБ 2х6, сечением 6 мм² и длительно допустимым током 55 А при прокладке в земле:

$$I_d=55 \text{ А} \geq I=3,5 \text{ А},$$

выбранное сечение проходит по нагреву длительно допустимым током.

Выбор и проверка аппаратов защиты:

При защите осветительной сети с газоразрядными лампами, автоматами имеющими тепловой или комбинированный расцепитель, отношение тока уставки расцепителя к рабочему току линии должно быть не менее $I_{\text{расц}}/I_p \geq 1,4$ с уставкой менее 50 А:

Расчёт тока срабатывания расцепителя автомата осветительной линии :

$$I_{\text{расц}} > 1,4 \cdot 3,5 = 4,9 \text{ А},$$

выбираем автомат марки ВА47–29В 1Р 5А, $I_{расц}=5$ А.

Автоматы устанавливаются на отдельном щитке освещения в РУНН трансформаторной подстанции.

В установках с глухим заземлением нейтрали проводники должны быть выбраны таким образом, чтобы при коротком замыкании возник ток, превышающий не менее чем в три раза номинальный ток расцепителя автомата, имеющего обратно зависимую от тока характеристику. Допускается не выполнять расчётной проверки кратности тока короткого замыкания, если обеспечено соотношение между длительно допустимым током проводника $I_{п}$ и номинальным током I_3 защитного аппарата: $I_{п}/I_3 \geq 1$.

Отношение между длительно допустимым током линии 1 и номинальным током защитного аппарата составляет:

$$I_{п}/I_3 = 55/5 = 11 \text{ A} \geq 1,$$

что соответствует условию согласования выбранного автомата с сечением проводника.

3 МЕТОДЫ УПРАВЛЕНИЯ ОСВЕЩЕНИЕМ

3.1 Удалённое управление светом

Удалённое управление светом – дает возможность создавать световые сценарии из неограниченного количества источников света с различной яркостью, включать их с задержкой или одновременно, имитируя, например, эффект "бегущих огней".

Используя специальные световые регуляторы можно не только менять яркость, с которой горит лампа при включении, но и время, за которое будет достигаться эта яркость.

Функция контроля освещенности, предназначена в основном для жилых или офисных помещений – дает возможность поддерживать заданную освещенность рабочей поверхности, или управлять освещением в тёмных помещениях. Так же контроль освещения регулирует автоматическое включение наружного освещения в зависимости от времени суток и присутствия людей, это обеспечит дополнительный комфорт и экономию электроэнергии.

3.2 Автоматическое управление

Домашняя автоматизация управления освещением может быть реализована двумя основными способами. Первый – оборудование каждой комнаты пультом с несколькими кнопками. В результате управлять светом можно прямо с дивана или из кресла. Второй способ – полная автоматизация с привлечением специальных датчиков. Свет включается сразу, как только человек входит в помещение, а спустя некоторое время после его выхода выключается. Удобство такой автоматизации ощущается с первого же раза, особенно если обе руки заняты чем-либо и для того, чтобы дотянуться до выключателя, потребуются дополнительные усилия. Впрочем, выключатели при автоматическом управлении совсем не отменяются: если нужно выключить свет, оставаясь в комнате, это вполне можно сделать вручную.

Управление яркостью освещения:

Это система, дающая возможность регулировать световой поток путем добавления или убавления яркости. Подобные действия можно осуществлять с пульта дистанционного управления. Однако регулирование яркости может быть вынесено на отдельную панель, а может стать частью программы управления освещением по таймеру.

Управление освещением по таймеру:

Возможность программирования времени включения и выключения света, а также автоматическое изменение настроек. Так вечером после установленного ими часа и ночью освещение будет включаться только на половинную мощность, достаточную чтобы пройти в нужном направлении и взять необходимую вещь. А проживающим в загородном доме не придется самостоятельно включать свет на улице: управление по таймеру позволит решить эту задачу без вмешательства людей.

Управление освещением по времени суток:

Это более сложная, но очень удобная система. Даже летним днем, особенно в пасмурную погоду, естественного света может оказаться недостаточно для освещения комнаты или офиса. Тут поможет специальный датчик, фиксирующий уровень освещенности и при необходимости включающий лампы. Такая составляющая системы «умного дома» актуальна и в домах, где имеется зимний сад или оранжерея. Хозяевам не придется по несколько раз в день включать и выключать свет, а при отъезде волноваться за любимые растения.

Имитация присутствия:

Тем, чей образ жизни связан с частыми путешествиями или командировками, интеллектуальная система управления светом просто необходима. Именно она поможет создать эффект присутствия хозяев дома. Свет в различных помещениях будет включаться и отключаться, а его яркость будет регулироваться системой «умного здания». Такая имитация часто помогает предотвратить вторжение посторонних лиц во время отсутствия хозяев.

Автоматические системы управления освещением удобны и эффективны как в многоэтажных загородных коттеджах, так и в небольших типовых квартирах. Они позволяют сделать жизнь более комфортной, а затраты на приобретение и установку специального оборудования окупаются куда быстрее, чем это может показаться на первый взгляд.

3.3 Ручное управление

Включение и выключение уличных светильников осуществляется в ручном режиме. Каждый источник света или их группа контролируется оператором непосредственно на месте. Данный способ является наиболее проработанным, так как он применяется уже много лет.

4 КОНТРОЛЬ ОСВЕЩЕНИЯ

4.1 Устройства контроля

Перечислим устройства контроля для микроконтроллеров на базе Arduino:

Компьютер – установив на домашнем компьютере программу визуализации и используя модем, можно с портативного компьютера как отключить оставленный включённым свет, так и регулировать освещение, и настраивать сценарии работы света как вам удобнее.

Телефон – обеспечивает надёжную связь с микроконтроллером даже тогда, когда нет под рукой компьютера, для этого достаточно набрать номер домашнего телефона, подключённого к микроконтроллеру.

Сотовый телефон – современные WAP\GPRS\EDGE\3G\LTE–технологии позволяют использовать сотовый телефон в качестве инструмента или же пульта управления, контроля и диагностики системой, при помощи сторонних приложений.

Дистанционное управление светом, находясь в доме с помощью телефона и модуля– Bluetooth связи, легкий способ управлять освещением, не вставая с дивана.

4.2 Методы дистанционного управления светом

Управление освещением можно разделить на 5 типов:

- дистанционное управление светом – когда человек (или центральный пульт управления) управляет освещением на расстоянии, не прикасаясь к выключателям;
- локальное управление освещением – когда человек управляет светом, нажимая на клавишу выключателя;
- управление освещением по предписанию – когда управление светом происходит при наступлении какого–либо события;
- управление освещением по сценарию – когда управление светом

происходит при наступлении определенного момента времени;

– управление светом по датчику движения – когда управление светом происходит при срабатывании датчика движения.

Суть дистанционного управления с помощью телефона и модуля Bluetooth управления заключается в том, что, человек, нажимая на кнопки, записанные в программе на телефоне, может включать и выключать свет. Например, выходя из дома, человек может нажать одну кнопку на телефоне и как с пульта дистанционного управления, цепь напряжения разомкнется у всех связанных с ним выключателей. Одна кнопка пульта может управлять светом одной группы настроенных на неё устройств. В данной группе может находиться от 1 до 232 устройств. Количество групп управления напрямую зависит от программы, написанной для дистанционного управления.

Телефон считается пультом с дисплеем, именно поэтому с него можно управлять группами на программном уровне. И количество групп тут не зависит от физических свойств телефона (пульта) управления и может быть расширено до предельно большого количества.

Например, при подходе к дому, человек может достать телефон и нажимая на кнопку в приложении, включается свет перед дверью, чтоб её открыть. Для дистанционного управления светом можно использовать любые подходящие под это ПДУ (пульт дистанционного управления) помимо телефонов.

Основная идея дистанционного управления освещением с помощью портативного компьютера или ноутбука, который находится дома, заключается в том, что, человек, с помощью специального ПО, может включать, выключать свет. Например, рано утром сотрудник фирмы спускается с одного этажа рабочего помещения на другой в цех разработок, находящийся на другом этаже. Сев за рабочий компьютер, он забыл выключил ли он свет в помещении выше этажом. При себе нет ПДУ (пульта дистанционного управления), позволяющего управлять освещением на верхнем этаже. В этом случае он заходит в специальную программу через свой ПК, находит помещение, в

которой не выключен свет и выключает его одним кликом мыши. Для дистанционного управления с помощью ПК, можно использовать типовые решения.

Инновационным способом дистанционного управления является управление через смартфон на базе операционных систем iOS или Android. Все сервисы предоставляют клиентские приложения для мобильных устройств на базе ОС iOS или Android. Для этого нужно всего лишь установить любое из этих приложений на свой телефон и подключиться к микроконтроллеру, управляющему освещением, например, через bluetooth расширение.

Система дистанционного управления с помощью контроллера или же настенного контроллера. Настенный контроллер внешне похож на обычный выключатель, но к нему не подводится сетевая нагрузка. Он лишь передает команды на включение или выключение света на выключатели, которые управляют освещением. Настенный контроллер питается от батареи. Он управляет одним устройством или группой устройств. Также кнопки включения и выключения могут быть связаны с разными устройствами. Например, при входе в дом, человек нажимает на клавишу включения, которая связывает все устройства управления светом в доме. А при уходе из дома, человек нажмет на клавишу выключения и все устройства в прихожей, кухне и так далее выключатся.

Система управления освещением по предписанию (событиям). В качестве событий могут выступать срабатывание различных датчиков. Например, человек желает, чтобы при спуске по лестнице в подвальное помещение включался свет. Для этого можно использовать датчик двери, который срабатывает при открытии двери в подвальное помещение и срабатывая, посылает команду на включение света. Тем самым человек уже спускается по освещённой лестнице.

Изучив все технологии, было принято решение остановиться на управлении освещением при помощи дистанционных устройств, а именно мобильных устройств с использованием Bluetooth модулей. Это полное ручное

управление. Например, человек хочет, чтобы свет включался по нажатию иконки в приложении смартфона в диапазоне от 10 до 100 метров от устройства. Тут будет разработано ЭУ принимающее сигнал с запрограммированного приложения, установленного на смартфоне (на базе Android), принимающее ЭУ – это устройство, которое обрабатывает и исполняет полученные команды, телефон будет выступать устройством, которое подаёт сигналы управления на принимающую часть по средствам Bluetooth соединения. При получении сигнала принимающая часть напрямую включает подключенное к ней освещение или отключает. Было решено, что данный способ является более действенным и актуальным, и менее сложным, не нужно много датчиков и ретрансляторов, вся система находится полностью под управлением человека, также данная система при желании и наличии средств, в любой момент может быть модернизирована. В данной работе используются модули реле, модули–Bluetooth HC-06, для облегчения разработки устройства и уменьшения затрат.

По статистке, распространяемой в интернете, риск получения травмы при выполнении каких–то либо действий в не освящённом помещении за последний год, более 10000 людей, как в частных домах, так и в производственных помещениях. Этот проект направлен на упрощение удалённого управления освещением и уменьшение возможных рисков, и улучшение качества жизни людей с ограниченными возможностями.

5 РАЗРАБОТКА СХЕМЫ УПРАВЛЕНИЯ ОСВЕЩЕНИЕМ ПОМЕЩЕНИЯ

Лучше всего для разработки устройства использовать удобную среду разработки проектов для Arduino — Fritzing.

Fritzing – этот инструмент разработчика с открытым исходным кодом для обучения, прототипирования и обмена проектами на базе Arduino. Он работает на Windows, MacOS и Linux.

Fritzing позволит разработать принципиальную схему устройства, и создать ее представление в виде соединения макетов элементов расширения. Также дает возможность разработать печатную плату для ее дальнейшего изготовления. У Fritzing простой интерфейс (рисунок 5.1), который делает разработку электронных схем понятной.

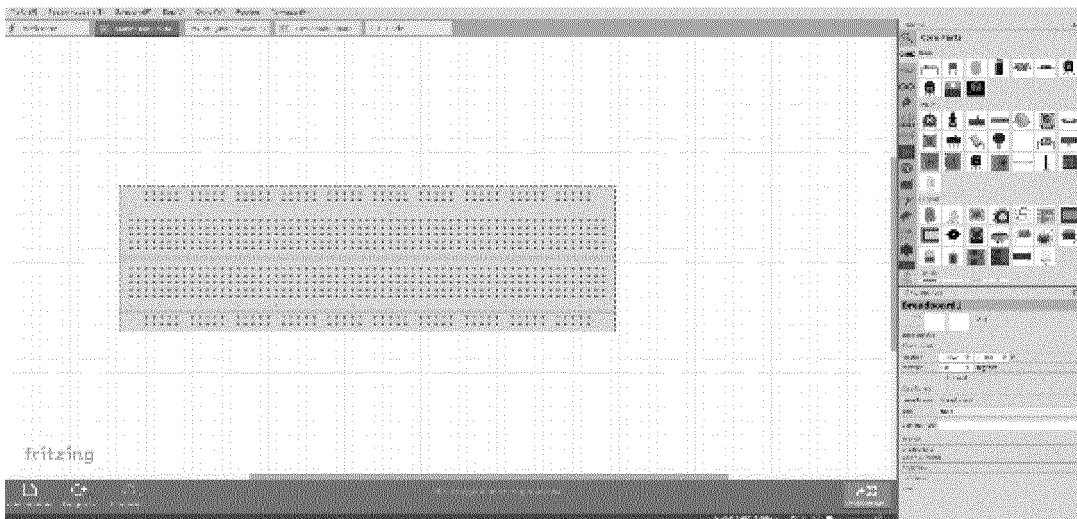


Рисунок 5.1 – Интерфейс Fritzing

Разработка устройства – станции принимающей сигнал на базе Arduino Nano (рисунок 5.2).

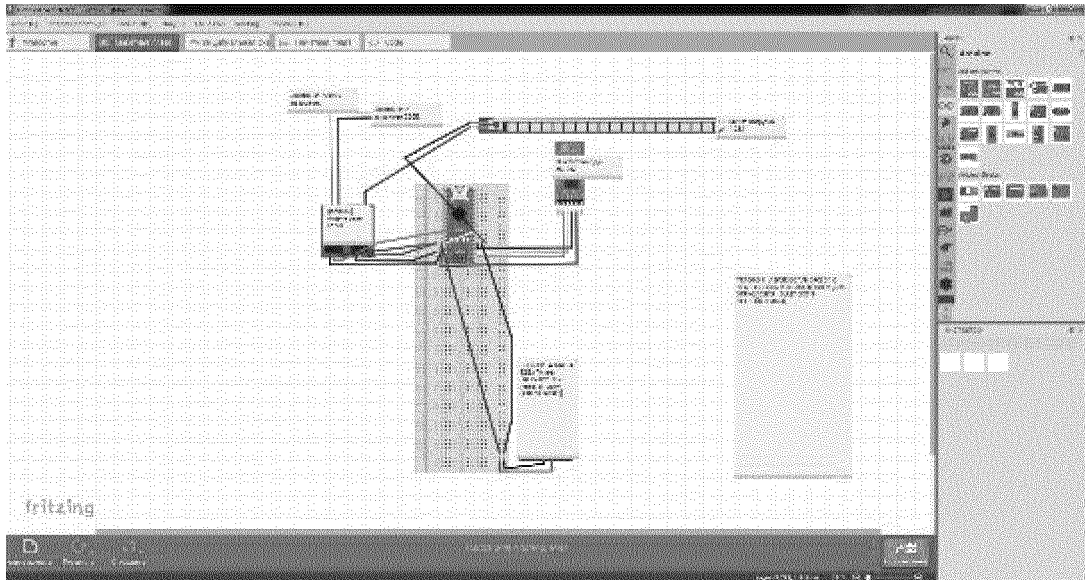


Рисунок 5.2 – Схема устройства для приёма сигнала с телефона на базе Arduino Nano

Разработка устройства – станции принимающей сигнал на базе Arduino UNO (рисунок 5.3).

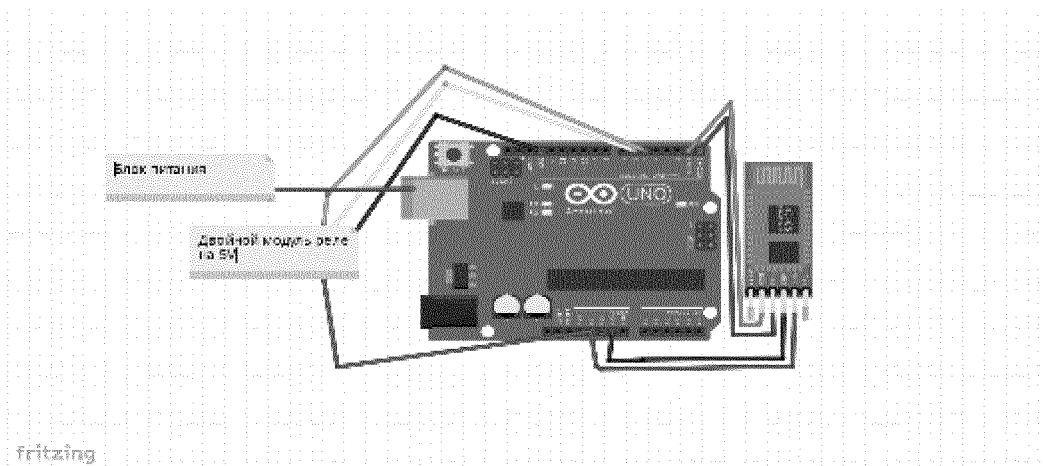


Рисунок 5.3 – Схема устройства для приёма сигнала с смартфона на базе Arduino UNO

5.1 Средства разработки и адаптации

Широкое распространение AutoCAD в мире обусловлено не в последнюю очередь развитыми средствами разработки и адаптации, которые позволяют настроить систему под нужды конкретных пользователей и значительно расширить функциональность базовой системы. Большой набор инструментальных средств, для разработки приложений делает базовую версию

AutoCAD универсальной платформой для разработки приложений. На базе AutoCAD самой компанией Autodesk и сторонними производителями создано большое количество специализированных прикладных приложений, таких как AutoCADMechanical, AutoCADElectrical, AutoCADArchitecture, GeoniCS, Promis-e, PLANT-4D, AutoPLANT, СПДС GraphiCS, MechaniCS, GEOBRIDGE, САПР ЛЭП, RubiusElectricSuite и других.

5.2 Графические пакеты разработки устройств на микроконтроллерах

Процесс разработки принципиальной схемы устройств на базе Arduino Nano и Arduino UNO в среде проектирования AutoCAD (рисунок 5.4 рисунок 5.5). Готовый вид схем разработанных устройств изображен на (рисунок 5.6).

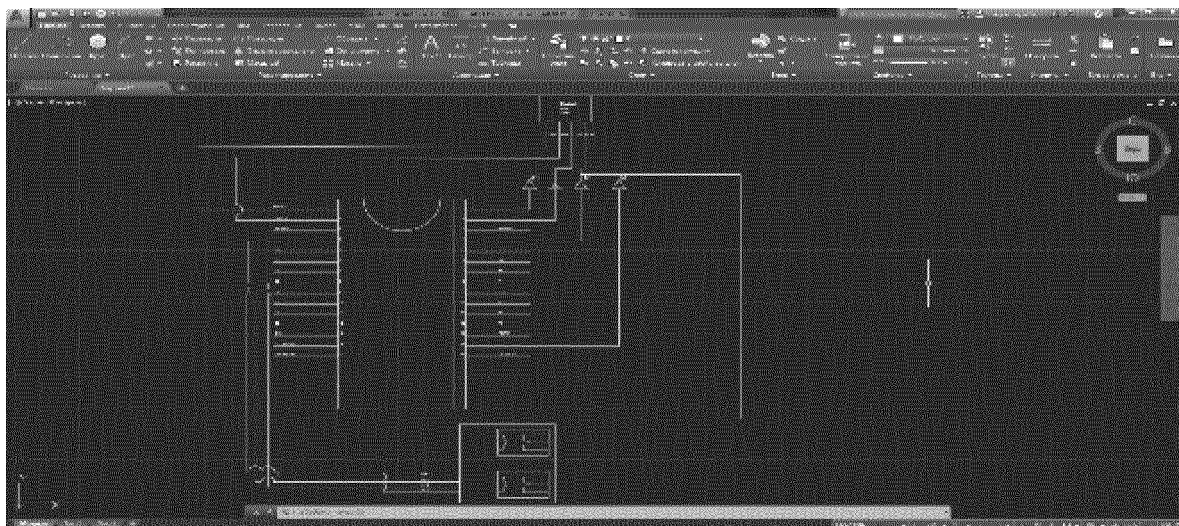


Рисунок 5.4 – Разработка принципиальной схемы устройства на базе Arduino Nano в среде проектирования AutoCAD.

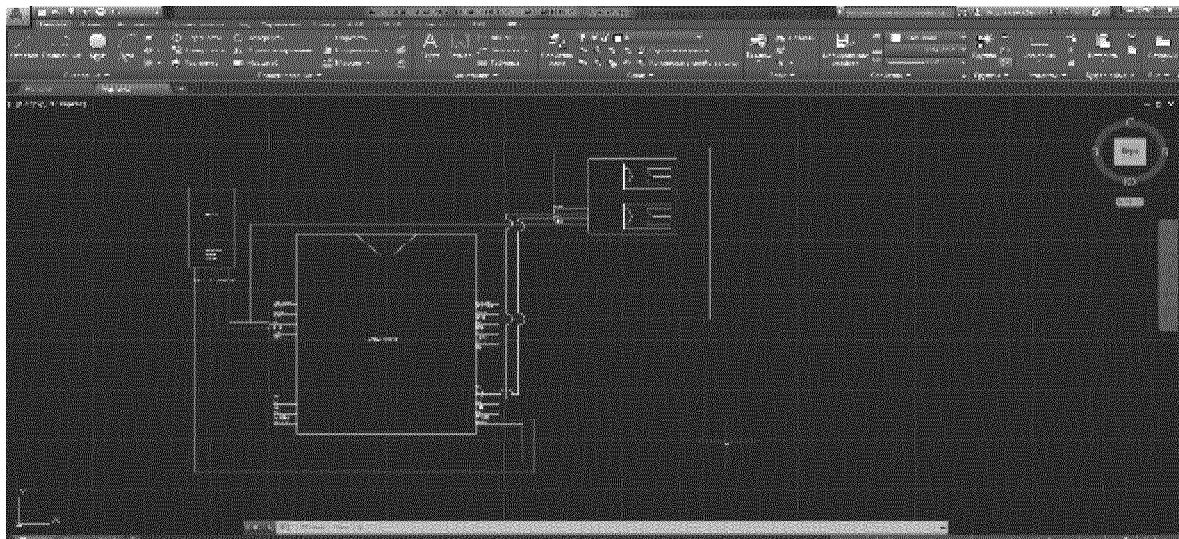


Рисунок 5.5 – Разработка принципиальной схемы устройства на базе Arduino UNO в среде проектирования AutoCAD.

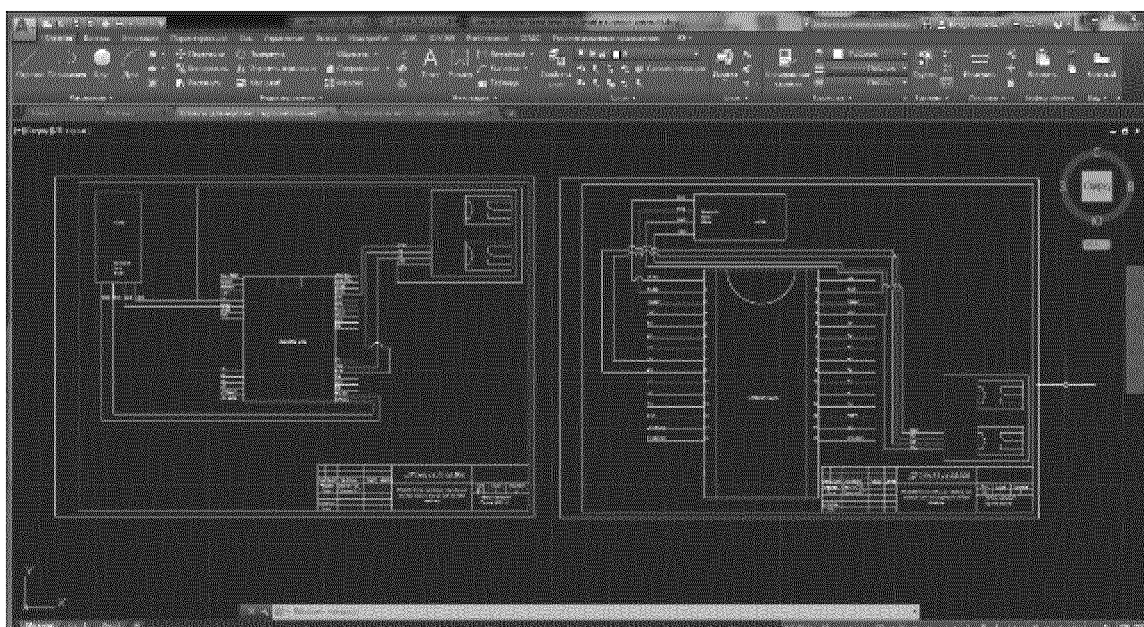


Рисунок 5.6 – Принципиальные схемы устройств

До того, как начать сборку тестового устройства, необходимо заранее записать скетч на Arduino Nano, или иначе будут возникать ошибки пинов микроконтроллера при записи программы на устройство.

5.3 Код программы для теста Bluetooth-модуля

```
char incomingByte; // переменная для хранения входящих данных
```

```
int LED = 13; // LED подключен к 13 выводу (используем встроенный светодиод)
```



```

void setup() {
    Serial.begin(9600); // инициализация порта на скорость 9600 бод для связи
с Bluetooth-модулем

    pinMode(LED, OUTPUT); // вывод LED настраиваем как выход
}

void loop() {
    if (Serial.available() > 0) { // если пришли данные
        incomingByte = Serial.read(); // считываем байт
        if(incomingByte == '0') // если пришел символ "0",
            digitalWrite(LED, LOW); // то выключаем вывод "LED"
        if(incomingByte == '1') // если пришел символ "1",
            digitalWrite(LED, HIGH); // то включаем вывод "LED"
    }
}

```

Успешная запись программы видна на рисунке 2.7

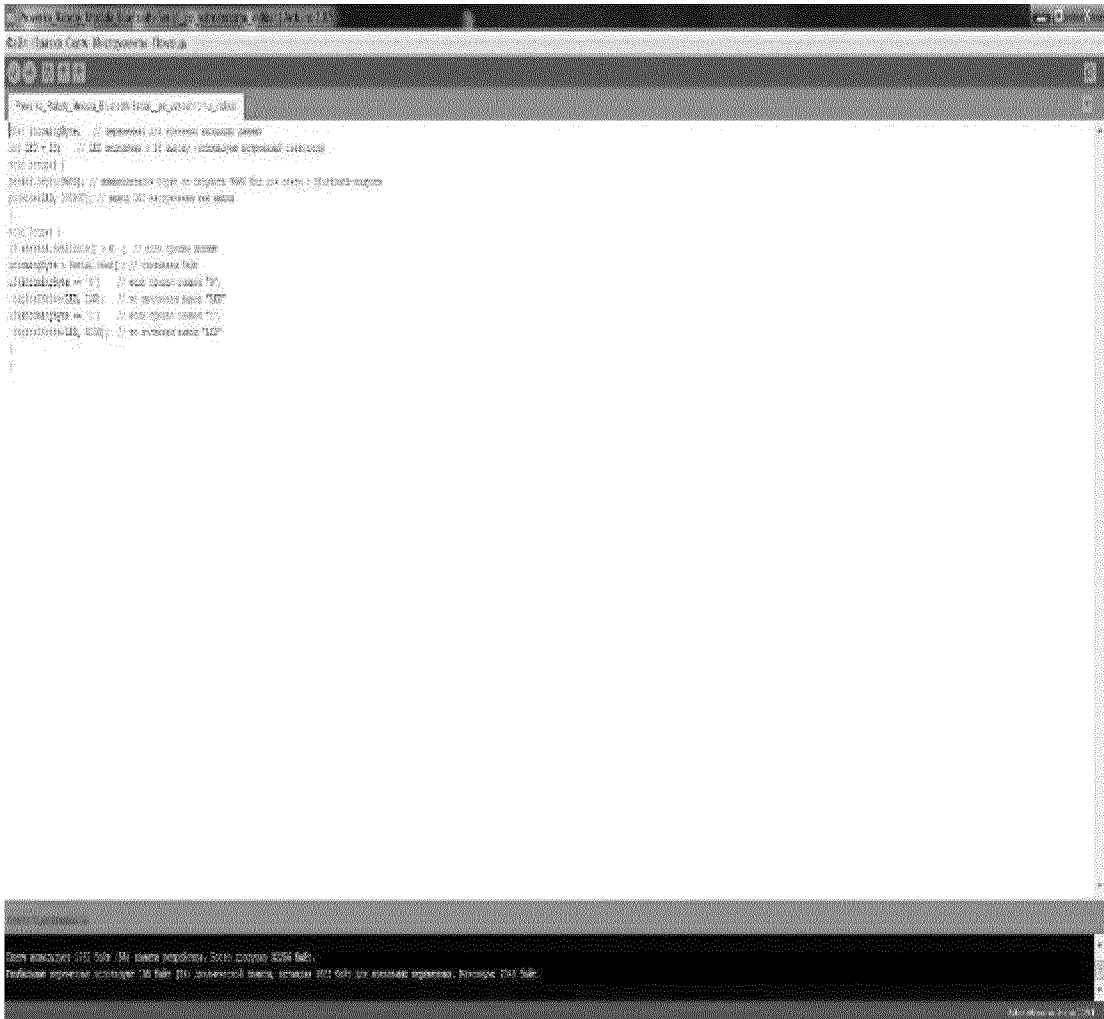


Рисунок 5.7 – Успешная запись программы на микропроцессор

Схема первого тестового устройства изображена на рисунке 5.8, подключаем согласно схеме:

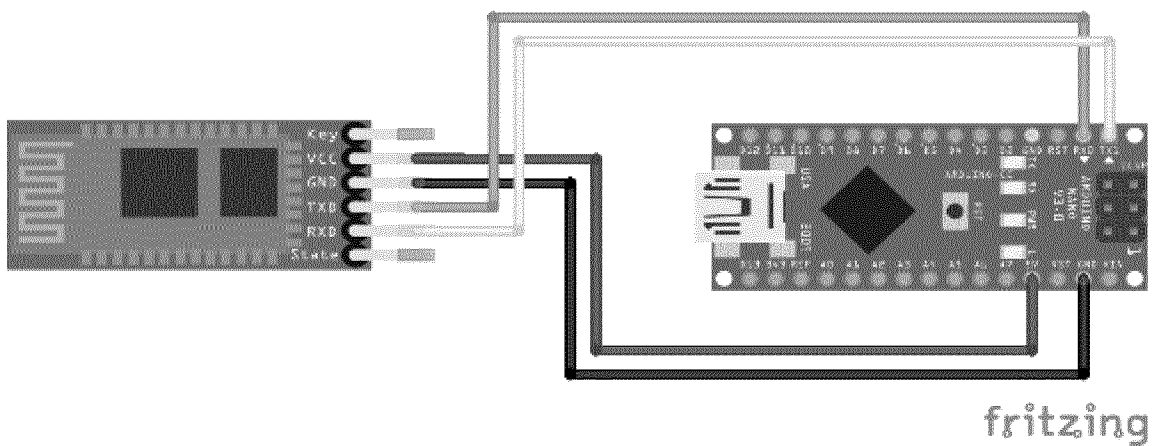


Рисунок 5.8 – Схема первого тестового устройства

VCC – 5V

GND – GND

TXD – RX

RXD–TX

Мы удалённо управляем незадействованным светодиодом на самом микроконтроллере, а именно, включаем и выключаем не задействованный светодиод.

Разработка тестового устройства, более мощной версии.

Схема тестового устройства(рисунок 5.9). Подключение второго устройства, разработанного с более оптимальными модулями:

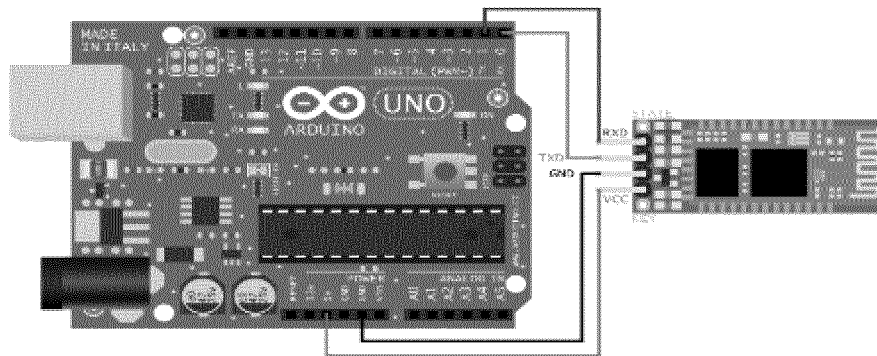


Рисунок 5.9 – Схема тестового устройства

До того, как начать сборку устройства, необходимо заранее записать скетч на Arduino Nano или Arduino UNO в зависимости от выбранной версии устройства, иначе будут возникать ошибки пинов микроконтроллера при записи программы на устройство.

Печатаем в среде программирования Arduino IDE, данную программу. Данная программа подходит для обоих микроконтроллеров, ничего исправлять не придётся т.к. все разъёмы указанные в программе есть на обоих платах.

6. РАЗРАБОТКА ПРОГРАММЫ УПРАВЛЕНИЯ ОСВЕЩЕНИЕМ

Код программы для работы устройства:

```
char incomingByte; // переменная для хранения входящих данных

int load1 = 5; // нагрузка 1 подключена к 5 выводу

int load2 = 6; // нагрузка 2 подключена к 6 выводу

void setup() {

  Serial.begin(9600); // инициализация порта на скорость 9600 бод для связи
с Bluetooth-модулем

  pinMode(5, OUTPUT); // вывод load1 настраиваем как выход
digitalWrite(5, HIGH); // сначала реле выключено

  pinMode(6, OUTPUT); // вывод load2 настраиваем как выход
digitalWrite(6, HIGH); // сначала реле выключено

}

void loop() {

  if (Serial.available() > 0) { // если пришли данные
incomingByte = Serial.read(); // считываем байт

  switch (incomingByte) { // перебираем входящие данные

    case '0': // если пришел "0" (задается в настройках Android-приложения)
      digitalWrite(5, LOW); // то выключаем вывод "load1"

      break;

    case '1': // если пришел "1"
```

```

digitalWrite(5, HIGH); // то включаем вывод "load1"

break;

case 'B': // если пришел "B"

digitalWrite(6, LOW); // то выключаем вывод "load2"

break;

case 'A': // если пришел "A"

digitalWrite(6, HIGH); // то включаем вывод "load2"

break;

} }

```

Успешная запись программы видна на рисунке 6.1.

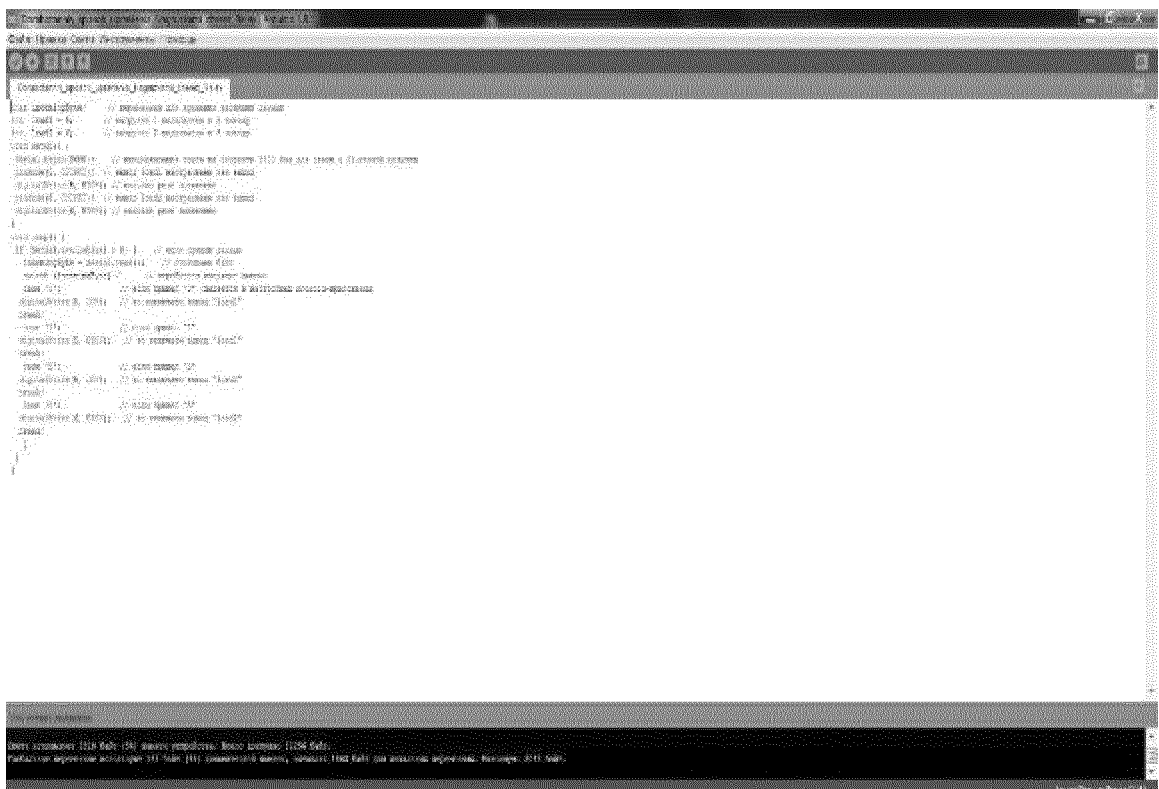


Рисунок 6.1 – Успешная запись программы на микроконтроллер.

– Подключаем всё согласно разработанной схеме №1 (рисунок 6.2)

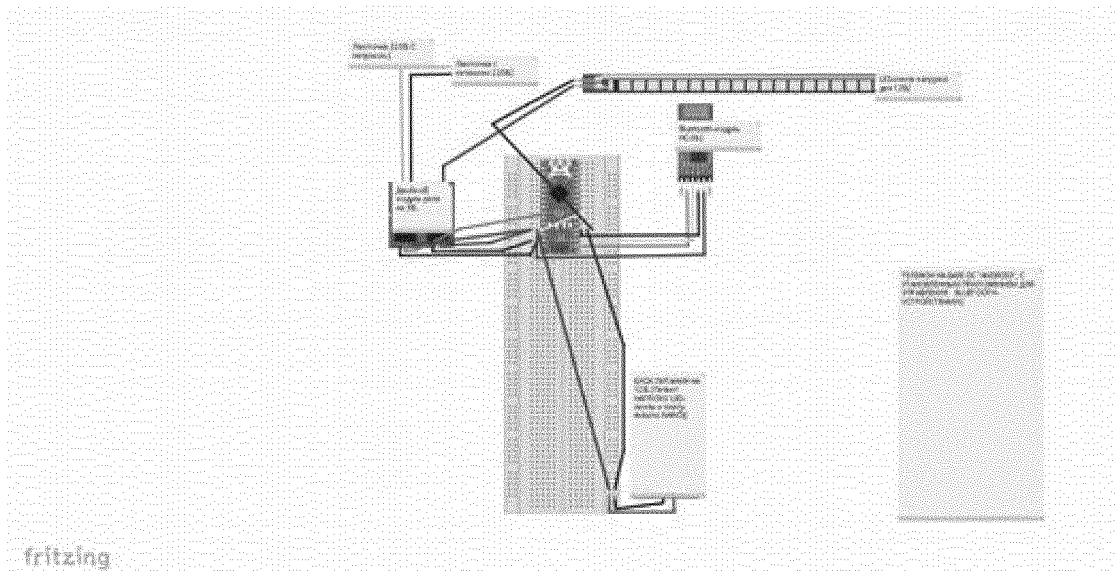


Рисунок 6.2 – Схема № 1, устройства на базе Arduino Nano

– Подключение Bluetooth–модуля:

VCC – 5V

GND – GND

TXD – RX

RXD – TX

– Подключение реле–модуля:

GND – GND

IN1 – D5

IN2 – D6

VCC – 5V

– Подключение БП 12В:

Подключаем + и –, к Шине питания 12В, от шины питания + подключаем к 5V, а – подключаем к GND.

– Подключение нагрузок к реле:

Лампу подключаем через разрыв проводов к источнику питания 220В и реле, а LED–ленту или её аналог LED–лампочку, так как, и у этой, и у другой нагрузка 12В, подключаем к контакту реле и GND на макетной плате.

Сборка устройства на базе Arduino Nano последовательно представлена

на рисунках 6.3, 6.4, 6.5, 6.6

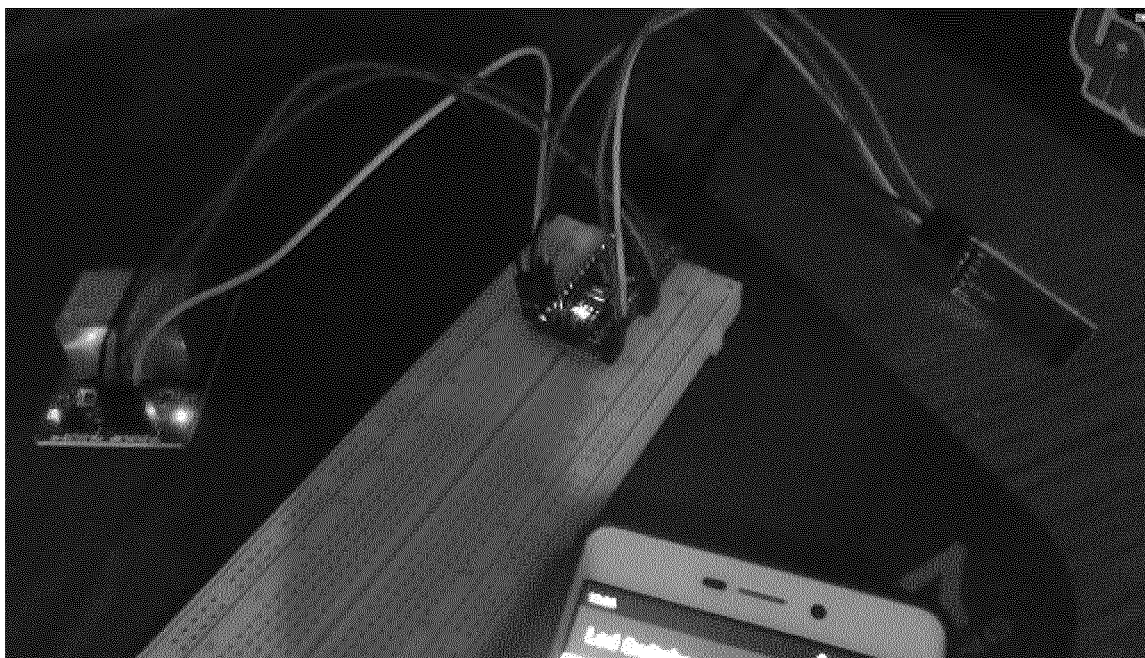


Рисунок 6.3 – Устройство без подключённой нагрузки на базе ArduinoNano.
Готовый вид устройства подключаемого в сеть освещения.

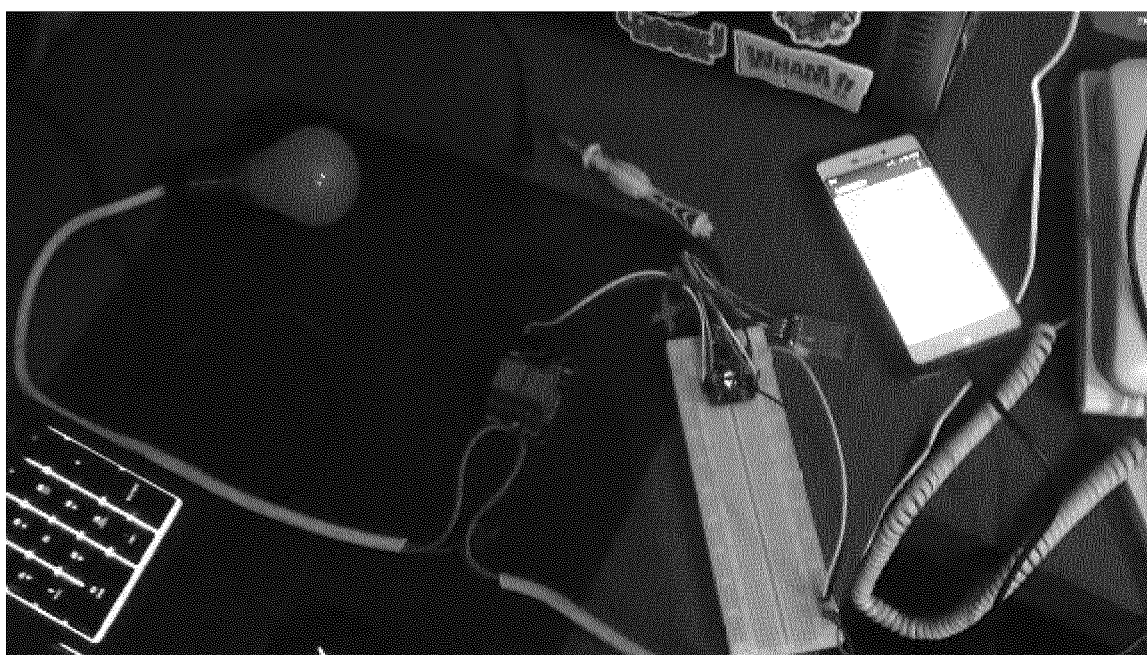


Рисунок 6.4 – Устройство с настроенным Bluetooth соединением на базе
Arduino Nano.

Подключаем всё согласно разработанной схеме №2 (рисунок 6.5)

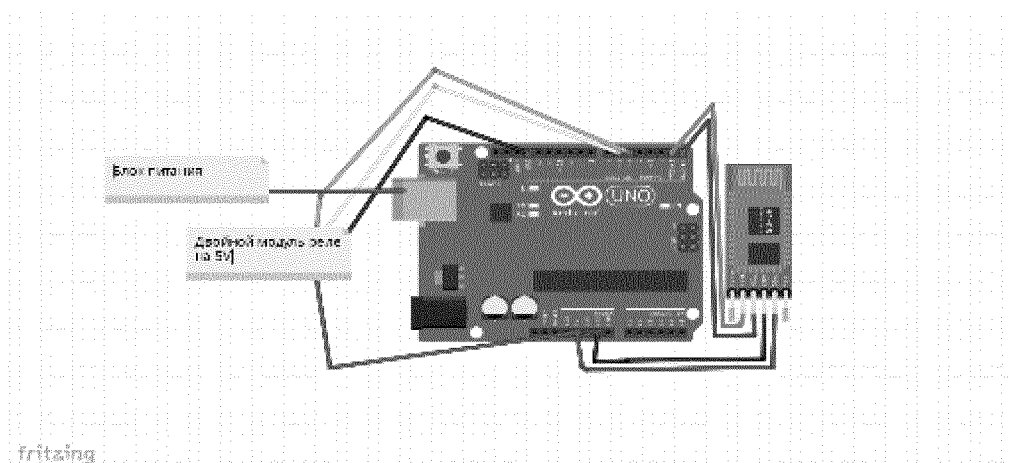


Рисунок 6.5 – Схема №2, устройства на базе Arduino UNO

Сборка устройства на базе Arduino UNO последовательно представлена на рисунке 6.6.

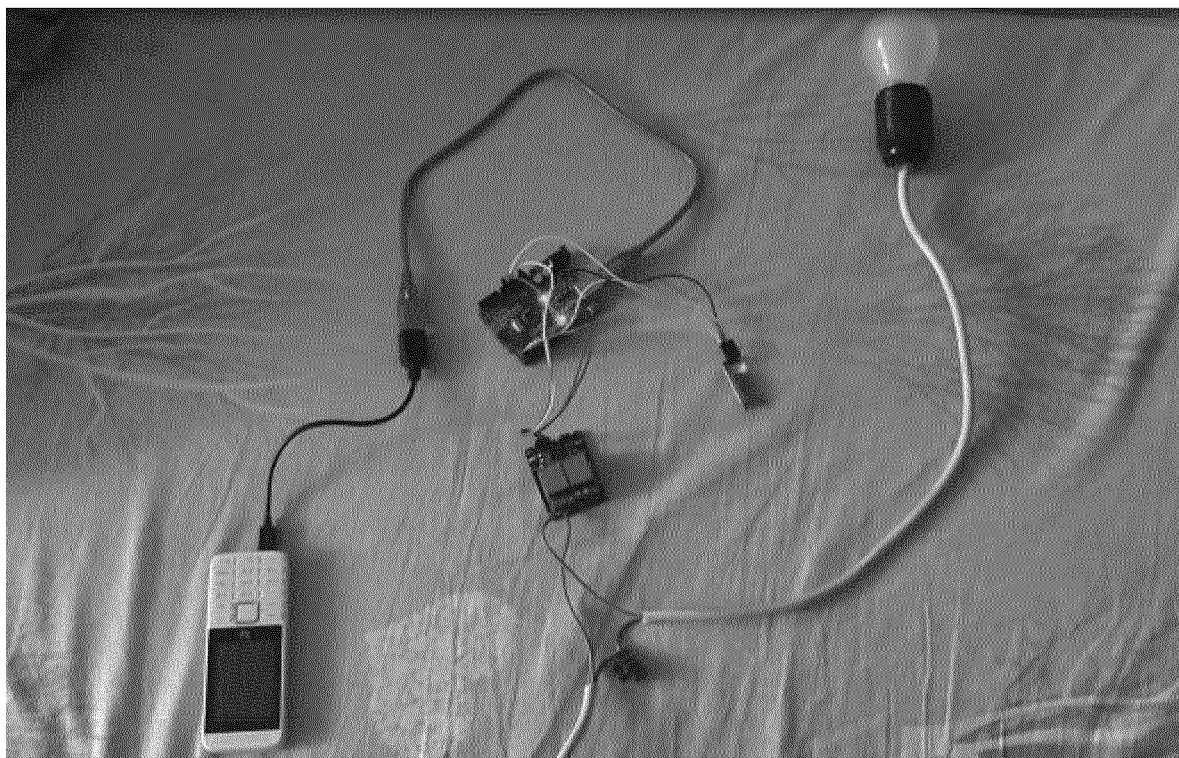


Рисунок 6.6 – Устройство с настроенным Bluetooth соединением на базе Arduino UNO



Рисунок 6.7 – Демонстрация работы устройства

6.1 Приложение по управлению на Android

Крайне простой интерфейс, выполняющий простенькие функции: именно включение и выключение (рисунки 6.8 – 6.11)

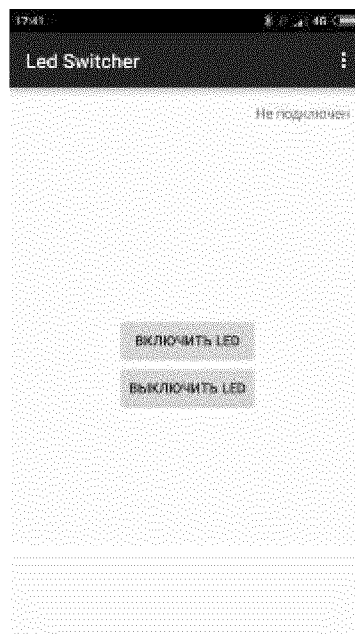


Рисунок 6.8 – Интерфейс программы управления

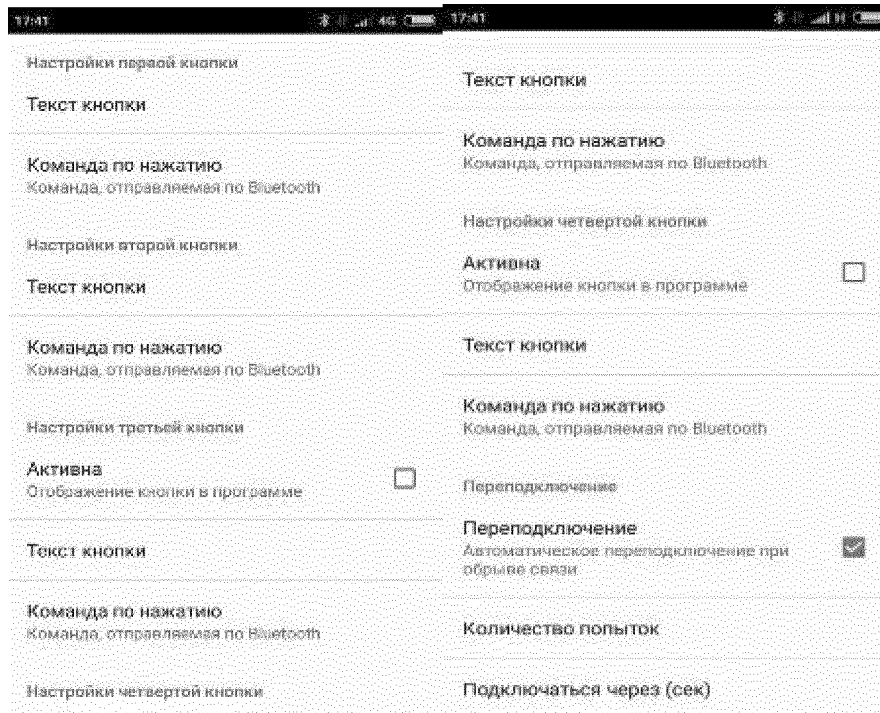


Рисунок 6.9 – Перечень возможных настроек данного приложения

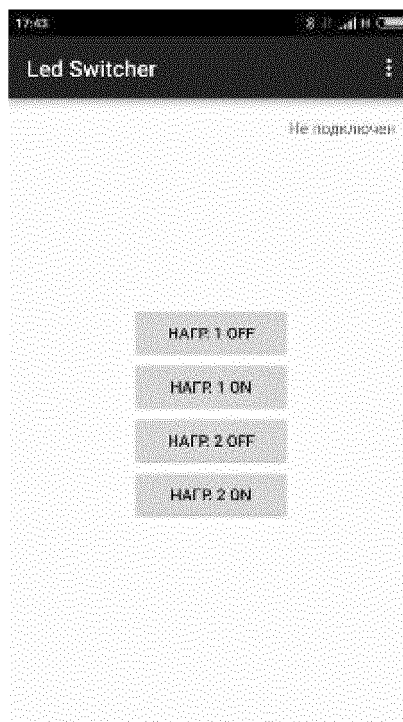


Рисунок 6.10 – Настройка клавиш под устройство

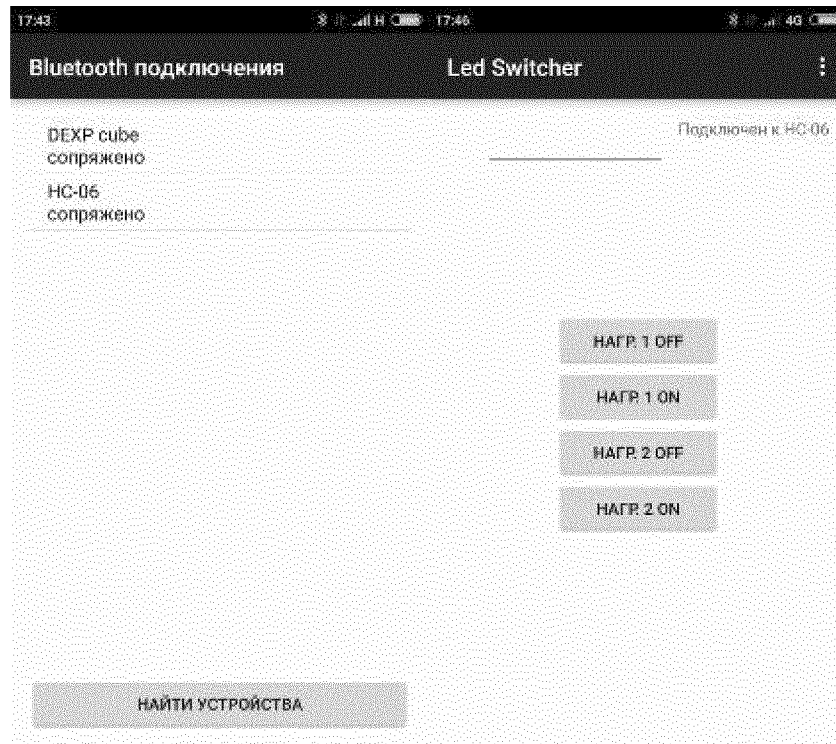


Рисунок 6.11 – Подключение устройств друг к другу

Крайне простое и удобное приложение, аналог можно написать с помощью шаблонов в веб-среде для программирования.

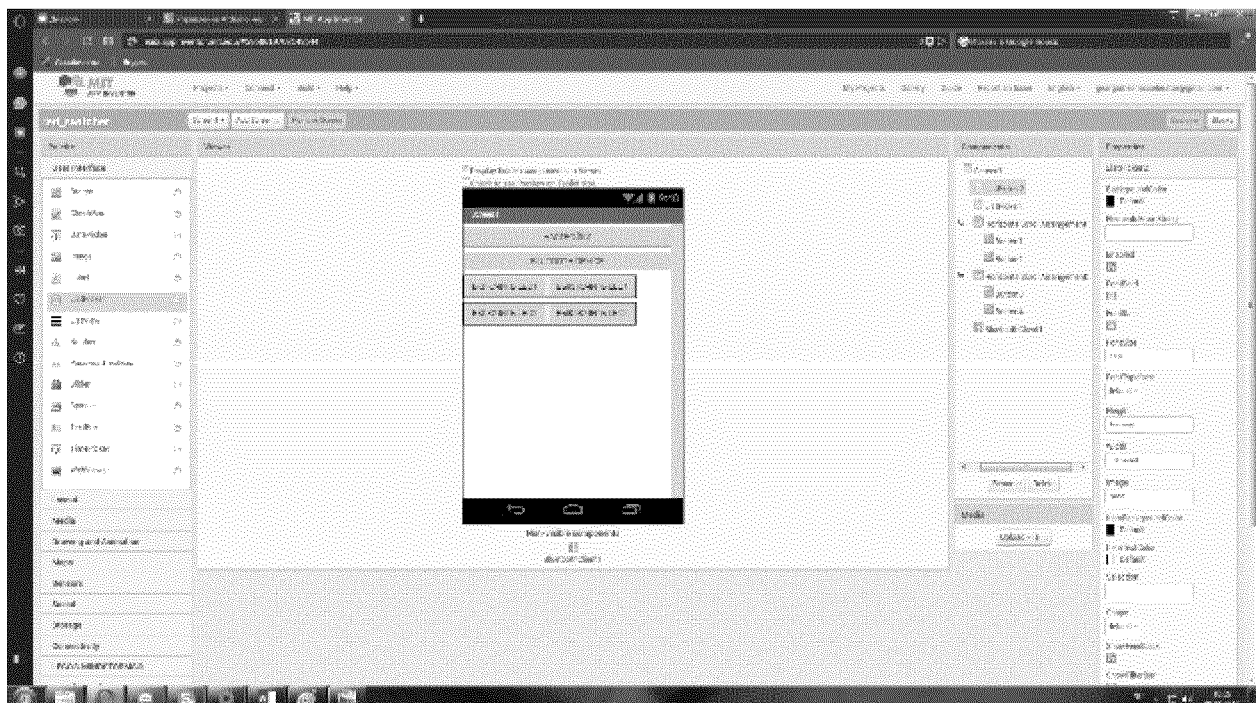


Рисунок 6.12 – Первая версия приложения управления для Android

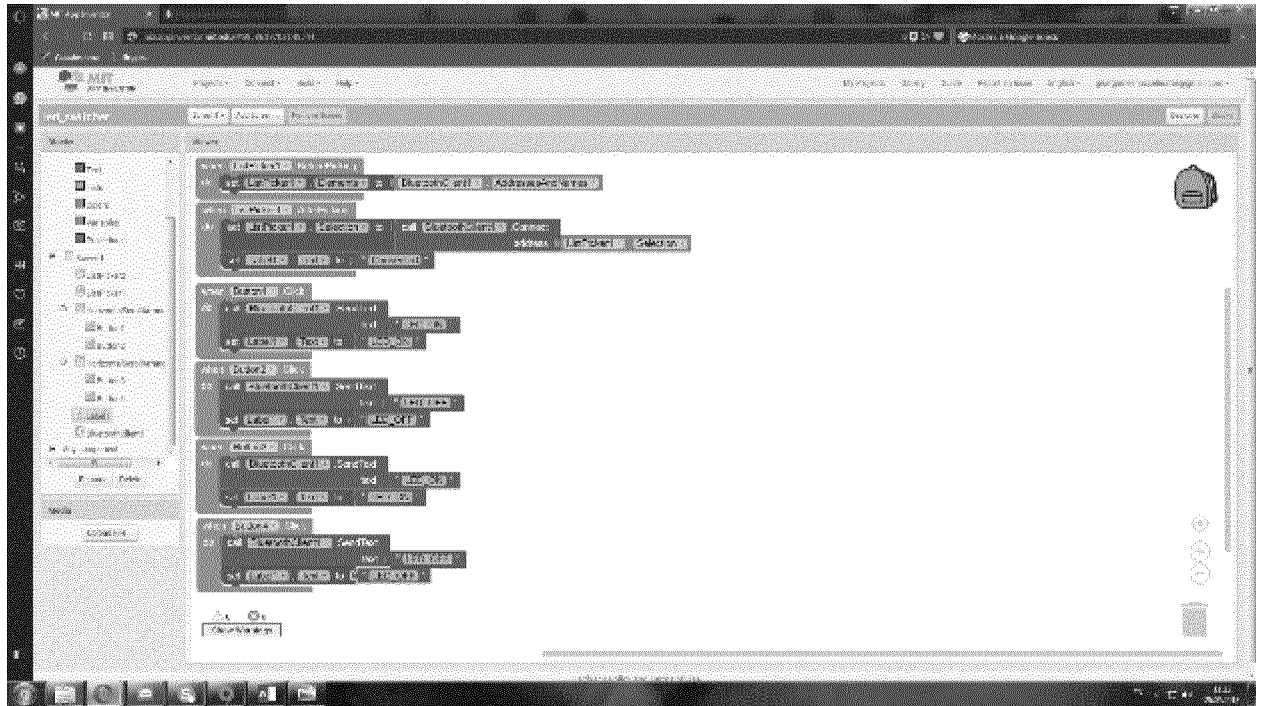


Рисунок 6.13 – Программирование приложения блочными командами в веб-интерфейсе AppInventor

ЗАКЛЮЧЕНИЕ

В ходе работы выполнены следующие задачи:

– разработаны схемы устройства принимающих сигнал, разработан алгоритм для удалённого управления, написан программный код, запрограммирован Arduino;

Созданное устройство поможет внести на рынок более дешёвые, более открытые к модификациям и ничем не уступающие своим конкурентам продукты. Любой покупатель сможет улучшить качество своей жизни и обезопасить ее от рисков. А люди с ограниченными возможностями этот проект окажет ещё более большую помощь, им не потребуется тянуться к выключателям, когда всё под рукой, управляется вручную. Для каждого клиента данное устройство можно будет подстроить под необходимые нужды, что делает данный продукт крайне мобильным для современного рынка.

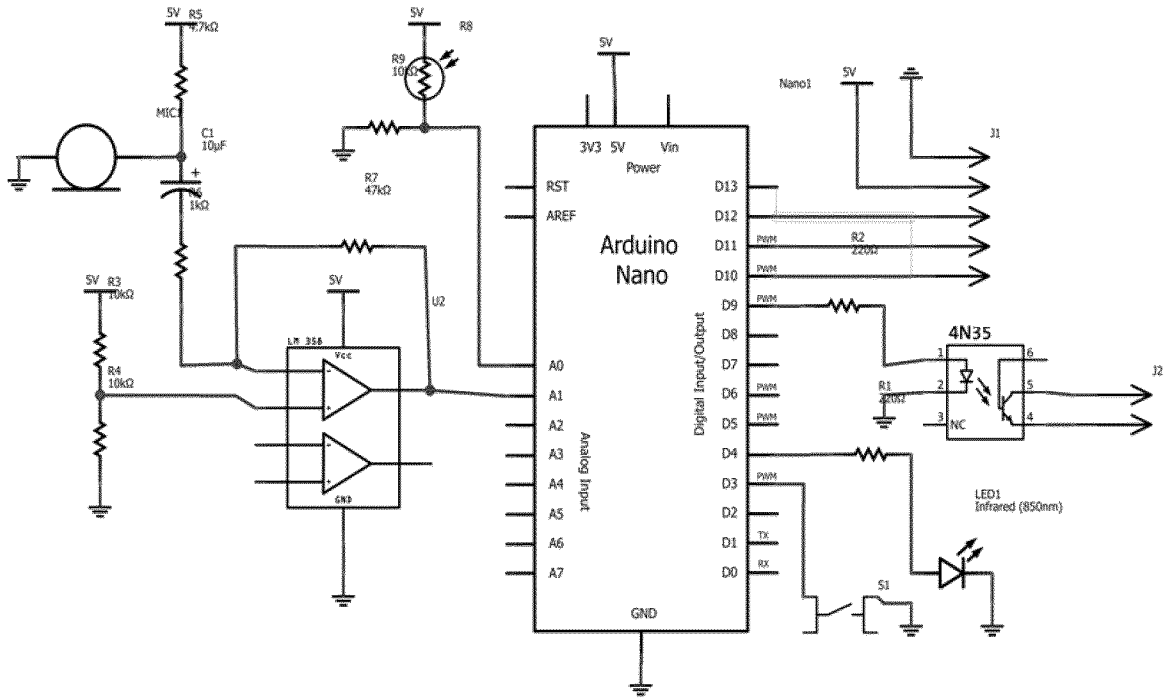
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальный сайт компании Arduino [Электронный источник]. – Режим доступа: <http://arduino.cc>. – 03.04.2021
2. Проекты с использованием контроллера Arduino. — СПб.: БХВ–Петербург, 2014 – 43 с.
3. Первые шаги по освоению Arduino [Электронный источник]. – Режим доступа: <http://maxkit.ru>. – 05.04.2021
4. Модули Последовательного Интерфейса Bluetooth. Руководство Пользователя, 2016. – 19 с.
5. Компоненты и модули для Arduino [Электронный источник]. – Режим доступа: <https://geektimes.ru/post/255798/>. – 05.04.2021
6. Технические характеристики ArduinoUno [Электронный источник].– Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardUno>. – 11.04.2021
7. Блог про Arduino / Freeduino [Электронный источник]. – Режим доступа: <http://arduino-ru.blogspot.ru/>. – 20.04.2021
8. Технические характеристики Arduino Nano [Электронный источник]. – Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardNano/>. – 21.04.2021
9. Проекты для Arduino [Электронный источник]. – Режим доступа: <https://create.arduino.cc/projecthub/>. – 5.05.2021
10. Джереми, Б. Изучаем Arduino Книга / Блюман Джереми. – СПб.: БХВ – Петербург, 2015.– 336с
11. Сайт разработчика Fritzing [Электронный источник]. – Режим доступа: <http://fritzing.org/home/>. – 5.05.2021
12. История развития микроконтроллеров [Электронный источник]. – Режим доступа: <https://vuzlit.ru/>. – 6.05.2021

13. Википедия – свободная энциклопедия. Клиент – сервер [Электронный источник]. – Режим доступа: <http://ru.wikipedia.org/wiki/Arduino/>. 7.05.2021
14. Группа документирования Atmega328 [Электронный источник]. – Режим доступа: <http://www.atmel.com/ru/ru/devices/atmega328.aspx/>. – 7.05.2021
15. Технические данные Arduino [Электронный источник]. – Режим доступа: <http://www.arduino.ua/ru/hardware/>. – 08.05.2021
16. Блог любителей робототехники "RoboCraft" [Электронный источник]. – Режим доступа: <http://robocraft.ru/page/about/>. – 08.05.2021
17. Островский М.А. Техника наружного освещения городов. – Итоги науки и техники. Сер. Светотехника и инфракрасная техника. 1973, т.3, 147 с.
18. Справочная книга по светотехнике/Под ред. Ю.Б. Айзенберга.–М.: Энергоатомиздат, 1983, – 472 с.
19. Электротехнический справочник: В 4 т. Т.4. Производство, передача и распределение электрической энергии / Под общ.ред. профессоров МЭИ В.Г. Герасимова и др. (гл. ред. И.Н. Орлов) – 8–е изд., испр. и доп. – М.: Издательство МЭИ, 2001 г. 964 с
20. Г.М. Кнорринг Справочная книга для проектирования электрического освещения, 1976, – 300 с.
21. Справочник по проектированию электроснабжения: Электроустановки промышленных предприятий/ Под общ. ред. Ю.Н. Тищенко и др. – М.: Энергоатомиздат, 1990, – 576 с.
22. Федоров, А.А. Учебное пособие для курсового и дипломного проектирования по электроснабжению промышленных предприятий – М.: Энергоатомиздат, 1987, – 420 с. А.А. Федоров, Л.Е. Старкова.

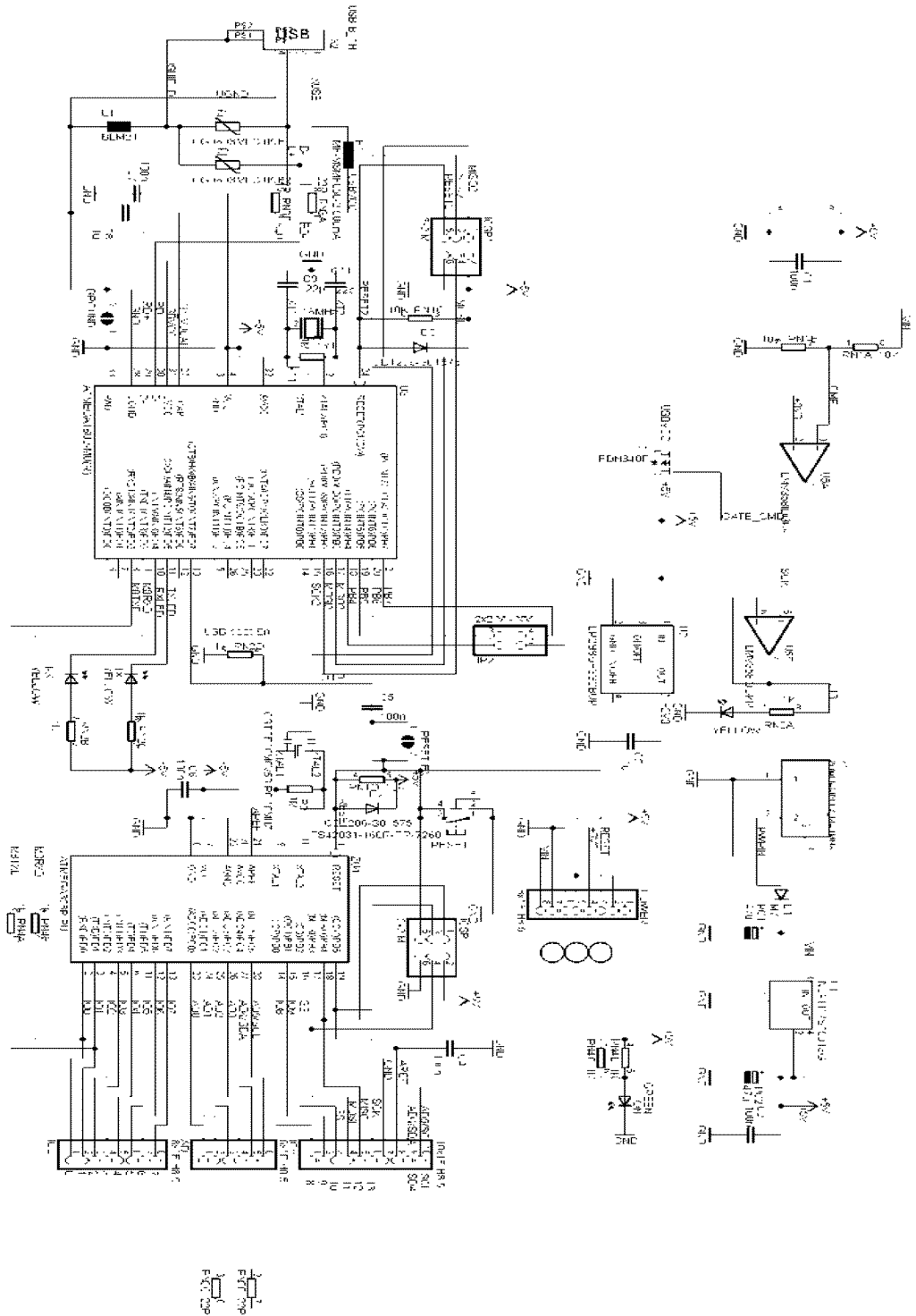
ПРИЛОЖЕНИЕ А

Принципиальная схема Arduino Nano



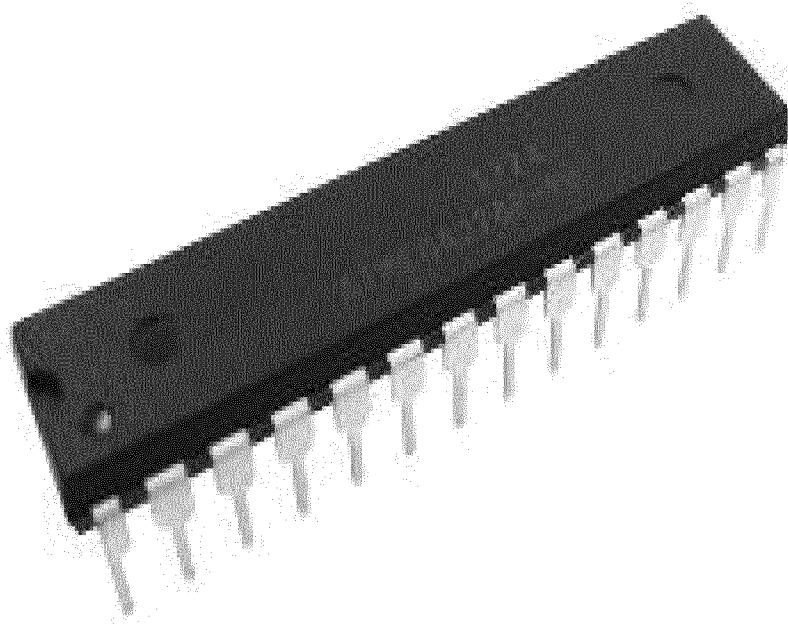
ПРИЛОЖЕНИЕ Б

Принципиальная схема Arduino UNO



ПРИЛОЖЕНИЕ В

Atmega328



ПРИЛОЖЕНИЕ Г

Код программы для теста Bluetooth–модуля:

```
char incomingByte; // переменная для хранения входящих данных

int LED = 13; // LED подключен к 13 выводу (используем встроенный
светодиод)

void setup() {

  Serial.begin(9600); // инициализация порта на скорость 9600 бод для связи
с Bluetooth–модулем

  pinMode(LED, OUTPUT); // вывод LED настраиваем как выход
}

void loop() {

  if (Serial.available() > 0) { // если пришли данные
incomingByte = Serial.read(); // считываем байт

  if (incomingByte == '0') // если пришел символ "0",
digitalWrite(LED, LOW); // то выключаем вывод "LED"

  if (incomingByte == '1') // если пришел символ "1",
digitalWrite(LED, HIGH); // то выключаем вывод "LED"

  }

}
```

ПРИЛОЖЕНИЕ Д

Код программы для работы устройства:

```
char incomingByte; // переменная для хранения входящих данных
int load1 = 5; // нагрузка 1 подключена к 5 выводу
int load2 = 6; // нагрузка 2 подключена к 6 выводу
void setup() {
  Serial.begin(9600); // инициализация порта на скорость 9600 бод для связи
  с Bluetooth-модулем
  pinMode(5, OUTPUT); // вывод load1 настраиваем как выход
  digitalWrite(5, HIGH); // сначала реле выключено
  pinMode(6, OUTPUT); // вывод load2 настраиваем как выход
  digitalWrite(6, HIGH); // сначала реле выключено
}
void loop() {
  if (Serial.available() > 0) { // если пришли данные
    incomingByte = Serial.read(); // считываем байт
    switch (incomingByte) { // перебираем входящие данные
      case '0': // если пришел "0" (задается в настройках Android-приложения)
        digitalWrite(5, LOW); // то выключаем вывод "load1"
        break;
      case '1': // если пришел "1"
        digitalWrite(5, HIGH); // то включаем вывод "load1"
        break;
      case 'B': // если пришел "B"
        digitalWrite(6, LOW); // то выключаем вывод "load2"
        break;
      case 'A': // если пришел "A"
        digitalWrite(6, HIGH); // то включаем вывод "load2"
        break;
    }
  }
}
```

ПРИЛОЖЕНИЕ Е

Листинг программы

```
#define MaxLightHome 450
#define MaxLightStreet 400
# define LightStreetPin 42 // выход освещения улицы
# define LightSensorStreetPin A4
# define NightMode 31
# define maxlight 450
# define minlight 400

int lightWindPin[4] = {38, 39, 40, 41}, // Выходы освещения с окнами
  MoveSensPinWind[4] = {8, 9, 10, 11}, // Датчики движения комнат с окнами
  LightPin[2] = {43, 44}, // Выходы освещения без окон
  MoveSensPin[2] = {12, 13}, // Датчики движения комнат без окон
  LightSensorPin[4] = {A0, A1, A2, A3};

int motorOpen[3] = {32, 34, 36},
  motorClose[3] = {33, 35, 37},
  Switcher[3] = {28, 29, 30},
  SwitcherON[3] = {0, 1, 2},
  StopOpen[3] = {22, 24, 26},
  StopClose[3] = {23, 25, 27},
  SensPin[3] = {A5, A6, A7};

void setup() {
  Serial.begin(9600);
  delay(100);
  pinMode(LightStreetPin, OUTPUT);
  pinMode(NightMode, INPUT);
  digitalWrite(NightMode, HIGH);
  for (inti = 0; i < 4; i++)
  {
    pinMode(lightWindPin[i], OUTPUT);
    pinMode(LightSensorPin[i], INPUT);
  }
  pinMode(LightStreetPin, OUTPUT);
  pinMode(LightSensorStreetPin, INPUT);
  pinMode(LightPin[0], OUTPUT);
  pinMode(LightPin[1], OUTPUT);

  for (inti = 0; i < 3; i++)
  {
    pinMode(motorOpen[i], OUTPUT);
```

```

pinMode(motorClose[i], OUTPUT);
pinMode(Switcher[i], INPUT);
pinMode(SwitcherON[i], INPUT);
pinMode(StopOpen[i], INPUT);
pinMode(StopClose[i], INPUT);

// while (digitalRead(StopOpen[i]) == LOW)
// {
//   digitalWrite(motorOpen[i], HIGH);
//   digitalWrite(motorClose[i], LOW);
// }
  digitalWrite(motorOpen[i], LOW);
  digitalWrite(motorClose[i], LOW);
}
pinMode(NightMode, INPUT);
digitalWrite(NightMode, HIGH);
delay(100);
}

//Управление светом в комнатах без окон
voidCheckLight()
{
  for (inti= 0; i< 2; i++)
  {
    if (digitalRead (MoveSensPin[i]) == HIGH)
      digitalWrite (LightPin[i], HIGH);
    elsedigitalWrite (LightPin[i], LOW);
  }
}

voidCheckLightWindow()
{
  for( inti= 0; i< 4; i++)
  {
    if ((analogRead(LightSensorPin[i]) <MaxLightHome) &&(digitalRead
(MoveSensPinWind[i])
== HIGH))
      digitalWrite(lightWindPin[i], HIGH);
    elsedigitalWrite(lightWindPin[i], LOW);
  }
}

// Функция управления шторами от выключателя

```

```

voidwinSwitch()
{
  for (inti= 0; i< 3; i++) {
    // Открытештор
    if (digitalRead(SwitcherON[i]) == LOW) {
      if (digitalRead(Switcher[i]) == HIGH &&digitalRead(StopOpen[i]) == LOW) {
        digitalWrite(motorOpen[i], HIGH);
        digitalWrite(motorClose[i], LOW);
      }
      // Закрытештор
      else if (digitalRead(Switcher[i]) == LOW &&digitalRead(StopClose[i]) == LOW)
      {
        digitalWrite(motorOpen[i], LOW);
        digitalWrite(motorClose[i], HIGH);
      }
      if (digitalRead(StopClose[i]) == HIGH || digitalRead(StopOpen[i]) == HIGH) //
      Есликонцевойвыключательсработал
      {
        digitalWrite(motorClose[i], LOW);
        digitalWrite(motorOpen[i], LOW);
      }
    }
  }
}

// Функция автоматического управления шторами
voidwinAutoSwitch() {
  for (inti = 0; i< 3; i++) {
    if (digitalRead(SwitcherON[i]) == HIGH){
      // Если освещение улицы выше максимального установленного значения
      if (analogRead(SensPin[i]) >maxlight)
      {
        digitalWrite(motorOpen[i], LOW);
        digitalWrite(motorClose[i], HIGH);
      }
      // Если освещение улицы ниже минимального установленного значения
      else if (analogRead(SensPin[i]) <minlight){
        digitalWrite(motorOpen[i], HIGH);
        digitalWrite(motorClose[i], LOW);
      }
    }
  }
  if (digitalRead(StopClose[i]) == HIGH || digitalRead(StopOpen[i]) == HIGH)
  {
    digitalWrite(motorClose[i], LOW);
    digitalWrite(motorOpen[i], LOW);
  }
}

```

```

    }
  }
}

void OnNightMode()
{
  delay(100);
  for ( int i = 0; i < 4; i++)
    digitalWrite(lightWindowPin[i], LOW);
  digitalWrite(LightStreetPin, LOW);
  for (int i = 0; i < 3; i++) {
    if (digitalRead(StopClose[i]) == LOW)
    {
      digitalWrite(motorOpen[i], LOW);
      digitalWrite(motorClose[i], HIGH);
    }
    if (digitalRead(StopClose[i]) == HIGH) // Если концевой выключатель сработал
    {
      digitalWrite(motorClose[i], LOW);
      digitalWrite(motorOpen[i], LOW);
    }
  }
}

void loop() {
  CheckLight();
  if ((digitalRead(NightMode)))
    OnNightMode();
  else{
    CheckLightWindow();
    winSwitch();
    winAutoSwitch();
    if (analogRead(LightSensorStreetPin) < MaxLightStreet)
      digitalWrite(LightStreetPin, HIGH);
      else digitalWrite(LightStreetPin, LOW);
  }
  delay(100);
}

```