

«Шаг в будущее 2020»

ИУ-6

***Восстановление природных биоритмов
человека путем создания эффекта
искусственного рассвета***

Автор: Чичунов Михаил Эдуардович

ГБОУ школа №2006, 11 класс

Восстановление природных биоритмов человека путем создания эффекта искусственного рассвета

Аннотация

Наиболее важные сведения о работе:

Цель работы: восстановление и сохранение естественных биоритмов человека.

Использованные методы и приёмы:

1. Изучение причин возникновения проблемы
2. Поиск способов решения проблемы и выбор оптимального из них
3. Составление плана работы
4. Использование микроконтроллера Arduino для управления освещением
5. Взаимодействие мобильного устройства и микроконтроллера через Bluetooth
6. Разработка мобильного приложения на языке Java

Полученные результаты: стабильная работа устройства и выполнение им поставленной задачи.

Выводы: Данный проект работоспособен и обладает практически неограниченным количеством возможностей улучшений и доработок:

- большое количество свободных пинов Arduino NANO;
- простота программной обработки передаваемых по bluetooth пакетов;
- малый объём занятой памяти Arduino NANO;
- открытый исходный код проекта;
- бесконечное многообразие датчиков и исполнительных устройства;
- всё это даёт проекту огромный потенциал для улучшений.

Содержание

•	Исследование.....	4
•	Проектирование.....	5
•	Разработка.....	7
	1. Световой элемент.....	8
	2. Источник питания.....	9
	3. Управление яркостью светодиодной ленты.....	10
	4. Канал связи Bluetooth.....	12
	5. Модуль часов реального времени.....	13
	6. Микроконтроллер.....	14
	7. Скетч Arduino.....	16
	8. Принцип работы светильника.....	16
•	Тестирование.....	20
•	Сравнение с аналогами.....	22
•	Предложения по практическому использованию результатов....	24
•	Собственные выводы, имеющие научное и практическое.....	24
	значение	
•	Заключение.....	25
•	Список источников.....	26
•	Приложение «Скетч Arduino NANO».....	27

Исследование

По данным Lenta.ru на 2019 год, среди работающих россиян 47 процентов признались, что не высыпаются в будние дни, 32 процента из них не могут выспаться из-за переработок или учебы. Это приводит к низкой производительности труда. Одной из главных причин хронического недосыпа может быть нарушение циркадных ритмов – это циклические колебания интенсивности различных биологических процессов, связанные со сменой дня и ночи. Биоритмы могут быть нарушены по множеству причин, но одна из основных причин сбоя внутренних биологических часов — нестабильность или полное отсутствие фактора, который задает эти самые биоритмы — Солнца, в зимнее время года. Учитывая данные современной биоритмологии, для качественного пробуждения необходимо воздействие утреннего солнечного света. Это не только простимулирует обмен веществ, но также поможет откалибровать биоритмы. Наиболее существенным раздражителем, влияющим на естественные биоритмы, является свет. В силу отсутствия естественного солнечного освещения во многих регионах нашей страны в последние годы, хорошим способом сохранить или восстановить циркадные ритмы станет искусственное освещение, имитирующее рассвет.

В наше время подавляющее большинство будильников используют громкий звук, что вводит человека в стрессовую ситуацию, а если будильник прозвонит во время фазы глубокого сна — весь день человек будет чувствовать усталость, сонливость и напряжение. Если же перед звонком будильника вывести человека из фазы глубокого сна с помощью искусственного рассвета, то ему удастся не только быстрее проснуться, но и снизить до минимума “сонную инерцию” – заторможенность и снижение когнитивных функций после пробуждения. В качестве ещё одного бонуса световых будильников исследователи Гронингенского университета называют лечение зимней хандры – сезонного аффективного расстройства.

Проектирование

Как известно, искусственный рассвет выигрывает у обычных будильников: плавное увеличение яркости в течение относительно большого промежутка времени, имитирующее рассвет, не только отличная идея для освещения, но и простой и безболезненный способ вывести спящего человека из фазы глубокого сна, тем самым подготовив его к пробуждению.

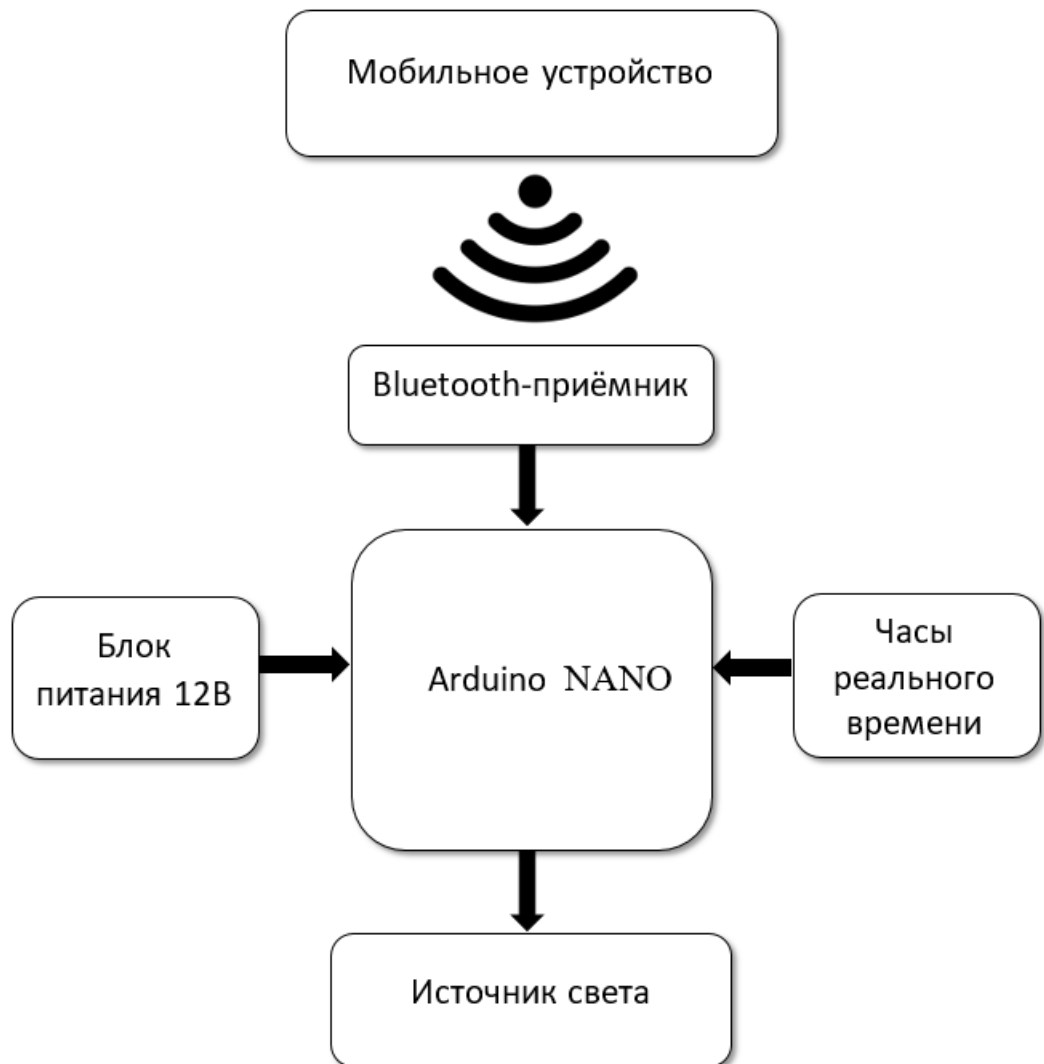
Я планирую создать устройство с функцией искусственного рассвета, управляемое с android-устройства — больше не нужно будет вставать и идти к выключателю, чтобы выключить или включить свет.

Это устройство будет полезно в первую очередь студентам, школьникам, работающим людям, да и просто тем, кто следит за своим режимом.



Для реализации своего проекта я планирую использовать микроконтроллер семейства Arduino из-за большого количества всевозможных модулей и простоты использования, так же мне понадобится светодиодная лента, блок питания, модуль часов реального времени и, конечно-же, Bluetooth-приёмник.

Разработка



В ходе разработки устройства я составил план работы и разбил его на следующие пункты:

1. Световой элемент — светодиодная лента тёплого свечения.



Рисунок 1 — светодиодная лента

Характеристики светодиодной ленты:

Длина ленты	0.5 м
Плотность светодиодов	60 шт/м
Потребляемая мощность	14 Ватт/м
Рабочее напряжение	12 Вольт

Световой поток ленты будет имитировать солнечный рассвет, исходя из этого, была выбрана лента тёплого света, чтобы максимально приблизиться к естественному солнечному свету.

2. Источник питания — импульсный блок питания с выходным напряжением, равным рабочему напряжению светодиодной ленты — 12 Вольт.



Рисунок 2 — Блок питания 12 В

Чтобы рассчитать требуемую мощность блока питания, нужно знать максимальную затрачиваемую мощность всех элементов цепи:

светодиодной ленты и микроконтроллера Arduino NANO.

Мы знаем номинальную мощность (P_n) метра светодиодной ленты, а так же знаем длину ленты (L), чтобы вычислить требуемую мощность пользуемся формулой:

$$P = P_n \times L$$

В моём случае лента при максимальной яркости требует 5,5 Ватт.

Так же в цепи присутствует микроконтроллер Arduino NANO, который в рабочем режиме потребляет не более 25 мВатт.

Суммарная требуемая мощность составит 5,525 Ватт. Чтобы узнать требуемый ток БП, разделим мощность на напряжение в цепи и получим 0,46 Ампер.

В итоге выбираем БП со следующими характеристиками:

Входное напряжение	АС 100-240 Вольт
Выходное напряжение	12 Вольт
Макс. ток	0,5 Ампер

3. Транзистор — устройство, способное от небольшого входного сигнала управлять значительным током в выходной цепи. Для управления яркостью свечения светодиодной ленты в цепь включен n-канальный транзистор IRF3205PBF

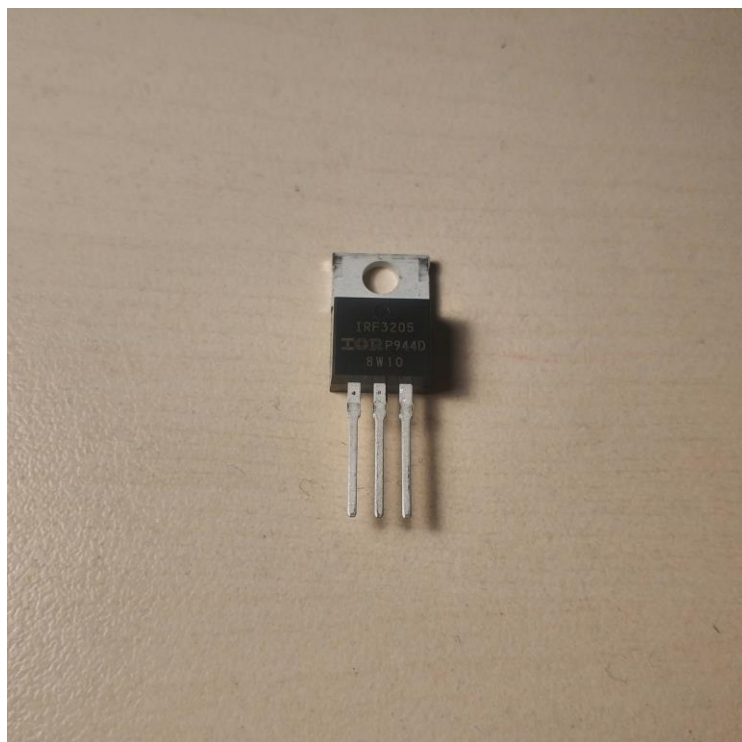


Рисунок 3 — транзистор IRF3205

Характеристики транзистора IRF3205PBF:

Макс. Напряжение сток-исток	55 Вольт
Макс. ток сток-исток	110 Ампер
Пороговое напряжение на затворе	2..4 Вольт
Сопротивление в открытом состоянии	8 мОм
Структура	n-канал

Транзистор, или же MOSFET включен в разрыв цепи питания светодиодной ленты и управляется ШИМ сигналом с микроконтроллера.

4. Для осуществления связи микроконтроллера с приложением используется bluetooth - модуль HC-06.



Рисунок 4 — HC-06

Это устройство предназначено для двусторонней передачи данных по одноимённому протоколу. Сфера применения технологии bluetooth ограничена лишь небольшим радиусом действия и фантазией конструктора. Bluetooth – идеальный способ быстрого управления устройствами на небольшом расстоянии.

Характеристики модуля НС-06:

Диапазон частот радиосвязи	2.4-2.48ГГц
Радиус действия	10 м
Максимальная скорость обмена информацией	115300 бод
Температура хранения	-40 .. +85 °С
Температура работы	-20 .. +70 °С
Рабочее напряжение	3.3 Вольт
Энергопотребление в рабочем режиме	20 мА
Энергопотребление в спящем режиме	4 мА

5. Для определения точного времени используется модуль часов реального времени — RTC DS1302.



Рисунок 5 — RTC DS3205

Этот модуль имеет встроенный источник питания на 3 Вольта и считает время независимо от внешних условий. Подключив его к микроконтроллеру можно получить следующие данные: текущие год, месяц, число, день недели, час, минута, секунда.

С помощью данных с этого модуля микроконтроллер сможет определить наступление заданного времени рассвета.

Характеристики модуля RTC DS1302:

Рабочее напряжение	2 .. 5 Вольт
Энергопотребление	300 нА
Рабочая температура	-40 .. +85 °C
Шина	SPI

6. Микроконтроллер — это микросхема, которая обрабатывает входящие данные и управляет подключенными устройствами в соответствии с программой, записанной во внутреннюю память (скетч).

В моём проекте использован микроконтроллер Arduino NANO.

Arduino NANO представляет плату с микроконтроллером ATmega328 и со всеми необходимыми для стабильной работы микроконтроллера элементами, такими как стабилизатор напряжения, системные светодиоды, кнопка RESET, входы для питания и прошивки микроконтроллера и удобно выведенные пины микроконтроллера.

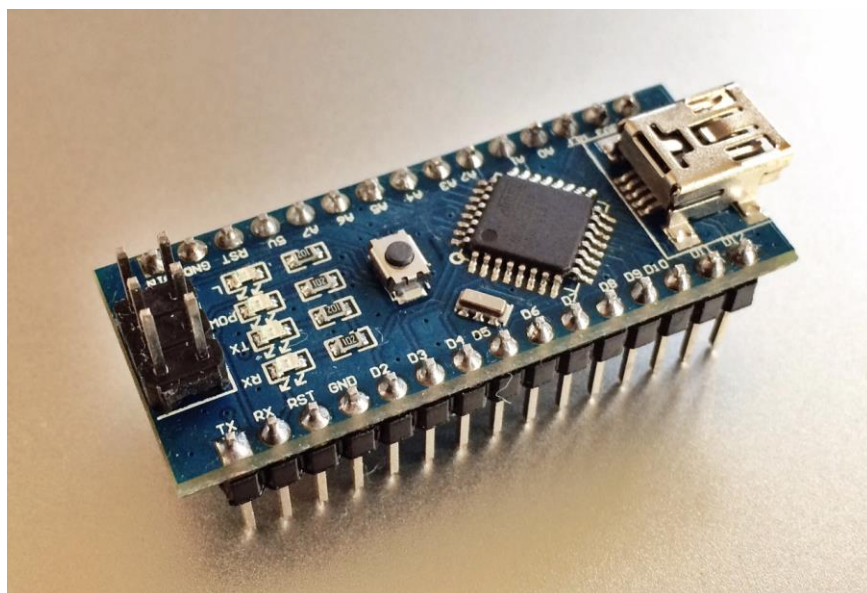


Рисунок 6 — Arduino NANO

Характеристики Arduino NANO:

Рабочее напряжение	5 Вольт
Цифровые входы/выходы	14
Аналоговые входы	8
Макс. ток одного входа/выхода	40 мА
Flash-память	16 КБ
EEPROM (энергонезависимая память)	512 Байт
Тактовая частота	16 МГц

7. Принцип работы скетча, записанного во внутреннюю память Arduino NANO (скетч прикреплён в приложении 1):

Микроконтроллер получает пакет данных по bluetooth, пакеты делятся на 3 типа:

- время начала рассвета;
- длительность рассвета;
- яркость лампы

При получении пакетов первого и второго типа данные записываются во внутреннюю память, а при получении пакета третьего типа микроконтроллер непосредственно меняет напряжение на затворе транзистора в соответствие с содержанием пакета.

Одновременно с этим микроконтроллер получает текущее время от модуля часов реального времени, и в случае наступления времени рассвета увеличивает напряжение на затворе транзистора пропорционально прошедшему с начала рассвета времени. Когда с начала рассвета проходит время, принятое в пакете второго типа, яркость свечения ленты становится максимальной.

8. Принцип работы светильника:

При запуске android-приложения инициализируется bluetooth и выполняется поиск светильника и подключение к нему. Далее открывается главный экран приложения, содержащий кнопки включения и выключения лампы, поле для ввода времени рассвета и длительности рассвета и ползунок для регулировки яркости светильника. В зависимости от действий пользователя, на bluetooth-приёмник НС-06 светильника отправляется пакет данных, содержащий соответствующую команду.

Модуль HC-06 передаёт полученные данные на serial-порт (2 контакта: RX и TX) микроконтроллера:

```
while (Serial.available() > 0) {  
    val += (char)Serial.read();  
    recieved = true;  
    delay(2);  
}
```

Если была получена команда включения или выключения, микроконтроллер соответственно включает или выключает напряжение на затворе транзистора IRF3205:

```
if (val == "off") {  
    digitalWrite(LED, LOW);  
}  
  
if (val == "on") {  
    digitalWrite(LED, HIGH);  
}
```

А транзистор, в свою очередь, полностью открывается, становясь проводником, или же полностью закрывается, становясь диэлектриком и разрывая цепь питания светодиодной ленты.

Если была отправлена команда изменить яркость, то на затвор транзистора подаётся не постоянное напряжение, а ШИМ сигнал, представляющий из себя чередующиеся с высокой частотой фазы напряжения 5V и отсутствия

напряжения. И отношение длительности «высокой» фазы к длительности «низкой» фазы называется скважностью и определяет уровень сигнала на затворе.

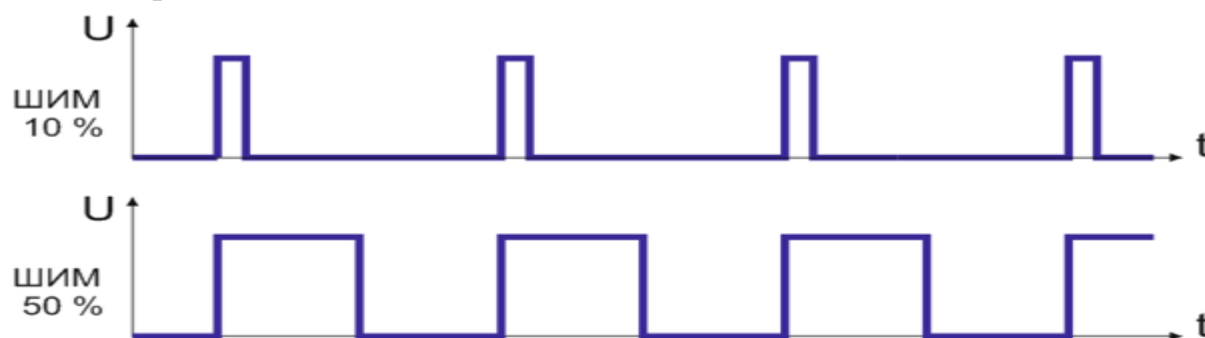


Рисунок 7 – принцип работы ШИМ-сигнала

В случае Arduino NANO выходной ШИМ-сигнал 8 битный, то есть функция `analogWrite()` может принимать значения от 0 до 255, где 0 — отсутствие «высоких» фаз, а 255 — отсутствие «низких» фаз.

Участок кода, ответственный за изменение яркости свечения светодиодной ленты:

```
if (val[0] == 't') {
    sunrise_hour = val.substring(1, 3).toInt();
    sunrise_minute = val.substring(3, 5).toInt();
}

if (val[0] == 'd') {
    sunrise_duration = val.substring(1).toInt();
    sunrise_duration = sunrise_duration * 60000;
}
```

Функция `map()` пропорционально переносит значение `val` из диапазона от 0 до 100 в диапазон от 0 до 255. Нужно это потому, что пакет данных содержит процент от полной яркости светодиодной ленты, соответственно, если мы хотим поставить яркость 100%, то функция `map()` переведет это значение в 255, что соответствует максимальному уровню ШИМ сигнала.

Если же был отправлен пакет «t» (время начала рассвета) или пакет «d» (длительность рассвета), то данные, содержащиеся в этих пакетах, записываются в соответствующие переменные и используются позже:

```
if (val[0] == 'b') {
    brightness = map(val.substring(1).toInt(), 0, 100, 0, 255);
    analogWrite(LED, brightness);
}
```

Умножение `sunset_time` на 60000 нужно потому, что Arduino NANO считает время в миллисекундах, а пакет «d» содержит количество минут.

Дальше остается лишь вовремя и равномерно, в течение указанного времени плавно увеличивать яркость светодиодной ленты, имитируя рассвет. Текущие час и минута сравниваются с указанными в пакете «t» данными, и если данные совпадают, то есть наступило время рассвета, то напряжение на затворе транзистора начинает равномерно расти, и полностью открывает транзистор спустя количество минут, указанное в пакете данных «t»:

```
if ((time.Hours == sunrise_hour) && (time.minutes == sunrise_minute) && (!sunrise)){
    Serial.println("SUNRISE STARTED");
    timer = millis();
    sunrise = true;
}

if ((millis()-timer >= sunrise_duration) && sunrise) {
    Serial.println("SUNRISE ENDED");
    sunrise = false;
}

if (sunrise) {
    sunrise_time = millis()-timer;
    analogWrite(LED, map(sunrise_time, 0, sunrise_duration, 0, 255));
}
```

Здесь используется функция `millis()`, она возвращает количество миллисекунд, прошедших с начала работы Arduino NANO.

Сначала микроконтроллер запоминает момент времени, когда начался рассвет, а затем, в функции `map()`, на затвор транзистора подается ШИМ сигнал, пропорциональный прошедшему с момента начала рассвета времени. Когда время рассвета достигает `sunset_time`, яркость становится максимальной.

Тестирование

В ходе тестирования мной были проверены такие параметры устройства, как:

1. Точность времени срабатывания:

Рассвет начинается в установленное время с погрешностью < 2 сек. Такая погрешность обусловлена тем, что время загружается в модуль вручную, через Bluetooth-интерфейс.

2. Плавность включения освещения:

Проверяя плавность увеличения яркости, я установил длительность рассвета 15 минут и наблюдал за тем, как яркость ленты постепенно растёт



РИСУНОК 8

60 СЕКУНД ПОСЛЕ НАЧАЛА РАС-
СВЕТА



РИСУНОК 9

5 МИНУТ ПОСЛЕ НАЧАЛА РАССВЕТА



РИСУНОК 10

15 МИНУТ ПОСЛЕ НАЧАЛА РАС-
СВЕТА

3. Яркость свечения:

Светодиодная лента вышла на 100% яркости. Этой яркости достаточно, чтобы осветить комнату среднего размера и разбудить человека.

4. Логика работы устройства:

В ходе тестов мной было замечено несколько логических ошибок – багов, например, процесс рассвета нельзя было никак прервать или остановить, эта ошибка была исправлена введением дополнительного флага в программу, который прекращал рассвет при изменении яркости, включении или выключении света через android-приложение.

Все выявленные в процессе тестирования ошибки и недочёты были исправлены на этапе отладки устройства

Используя этот способ пробуждения в течение недели, я и сам заметил, что просыпаться стало легче и комфортней. Вывод – устройство работает и выполняет поставленную задачу.

Теперь даже зимой или в пасмурный день у меня есть возможность просыпаться от света, а не от звука.

Сравнение с аналогами

Световые будильники Phillips



Рисунок 11 – световой будильник Phillips

Будильники компании Phillips одни из самых популярных на рынке и они имеют такие функции, как:

- Встроенное радио
- Искусственный закат
- Циферблат на корпусе
- Функция расслабления дыхания
- Встроенный будильник со звуками природы
- Сенсорное управление

Эти устройства снабжены большим количеством дополнительных функций, но, как мне показалось, они не решают одной из главных проблем, а именно - ручного управления. Если сравнивать мой вариант рассвета и устройство

компании Phillips, то мой вариант дешевле, не перегружен не нужными функциями, управляется с мобильного устройства, а не с экрана будильника.

На российском рынке будильников с функцией искусственного рассвета Phillips не имеет конкурентов, и в поисках аналогов я пошел на AliExpress.

Китайский будильник с функцией искусственного рассвета



Рисунок 12 – китайский световой будильник

Светильник с AliExpress так же предлагает пользователю несколько звуков природы и радио, но, в отличие от Phillips, этот светильник способен светиться разными цветами. Управление им выполнено в виде кнопок на корпусе. Данный будильник обладает теми же недостатками.

Рассмотрев несколько аналогов моего устройства, я пришёл к выводу, что на данный момент в продаже нет устройств с функцией искусственного света, управляемых по bluetooth, или любым другим способом беспроводной передачи данных.

Предложения по практическому использованию результатов

На данный момент устройство выполняет задачу восстановления и сохранения человеческих биоритмов, но так же имеет множество путей доработок и нововведений, таких как: пробуждение и оповещение пользователя о задымлении, утечки газа, затоплении, коротком замыкании и других ЧС, возможность работы в режиме гирлянды или светомузыки, возможность добавления свечения разного цвета, или же взаимодействие с устройствами концепции «умный дом», например, с Алисой от Яндекса или с Siri от компании Apple.

Собственные выводы, имеющие научное и практическое значение

Разработанная схема и программа осуществляют взаимодействие нескольких устройств через bluetooth под конкретную задачу – «Искусственный свет». Однако возможности данной схемы практически безграничны, можно использовать другие типы беспроводной связи, например мобильную связь или WI-FI, неограниченное количество датчиков, анализаторов и исполнительных устройств, написание программ под каждую конкретную задачу, а принципиальная схема остается та же. Т.е. при той же схеме мы можем создать устройство под конкретную задачу.

Заключение

Работать над созданием данной системы было интересно и познавательно. В итоге у меня получилось создать полноценное устройство, способное удовлетворить желания пользователя. Были добавлены такие функции как: искусственный свет, дистанционное включение и выключение и управление яркостью лампы. Лампа работает в связке с мобильным приложением, именно это дает возможность удобного дистанционного управления освещением, которого так не хватает, когда ты уже лёг спать, но забыл выключить свет. В будущем хотелось бы расширить возможности данного прибора: помимо искусственного света устройство может обладать функциями пробуждения и предупреждения человека в экстренных ситуациях, таких как: утечка газа, задымление, короткое замыкание, физиологические проблемы, и другие ЧС. Также ясно, что увлеченному пользователю всегда будет мало предоставленных функций, поэтому появится возможность создания собственных алгоритмов работы прибора.

В ходе данной исследовательской работы были рассмотрены и проанализированы разные алгоритмы беспроводной передачи данных, налажена отправка и получение запросов по каналу bluetooth и обеспечена стабильная работа программы.

Список источников

1. Саймон Монк – Програмируем Arduino. Профессиональная работа со скетчами
2. Рудольф Сворень — электроника шаг за шагом
3. Хуанг Б., Ранберг Д. - Arduino для изобретателей
4. Петин В. - Arduino и Raspberry Pi в проектах Internet of Things, 2-е издание
5. Бьерн Страуструп - Программирование. Принципы и практика с использованием C++
6. Липпман С., Лажоие Ж., Му Б. - Язык программирования C++. Базовый курс

«Скетч Arduino NANO»

```
#include <DS3231.h>

DS3231 rtc(SDA, SCL);
Time t;

String val;
int h, m;
int LED = 3;
int sunrise_hour = 7;
int sunrise_minute = 0;
unsigned long sunrise_duration = 20*60000, timer, sunrise_time, rtc_timer;
int sun_min = 0;
int sun_max = 255;
int brightness = 0;
bool sunrise = false;
bool recieved, started, budilnik;

void setup() {
    delay(300);
    Serial.begin(9600);
    pinMode(LED, OUTPUT);
    rtc_timer = 0;
    rtc.begin();
}

void loop() {
    t = rtc.getTime();
    h = t.hour;
```

```
m = t.min;
if (millis() - rtc_timer >= 5000){
    Serial.print("Текущее время: ");
    Serial.print(h);
    Serial.print(":");
    Serial.println(m);
    rtc_timer = millis();
}
```

```
while (Serial.available() > 0) {
    val += (char)Serial.read();
    recivied = true;
    delay(2);
}
```

```
if (recivied) {
    Serial.println();
    Serial.print("Получена строка: ");
    Serial.println(val);
    Serial.print("Текущее время: ");
    Serial.print(h, DEC);
    Serial.print(":");
    Serial.println(m, DEC);
}
```

```
if (val[0] == 'x') {
    Serial.println("Текущие дата и время: ");
    Serial.print(t.date);
    Serial.print(".");
    Serial.print(t.mon);
    Serial.print(".");
    Serial.print(t.year);
}
```

```

Serial.print(" ");
Serial.print(t.hour);
Serial.print(":");
Serial.print(t.min);
Serial.print(":");
Serial.print(t.sec);
Serial.print(" ");
Serial.print(t.dow);
}

if (val[0] == 'z') {
    rtc.setDOW(val.substring(15, 16).toInt()); // Set Day-of-Week to SUNDAY
    rtc.setTime(val.substring(9, 11).toInt(), val.substring(11, 13).toInt(), val.substring(13,
15).toInt());
    rtc.setDate(val.substring(7, 9).toInt(), val.substring(5, 7).toInt(), val.substring(1,
5).toInt());
}

if (val[0] == 't') {
    sunrise_hour = val.substring(1, 3).toInt();
    sunrise_minute = val.substring(3, 5).toInt();
}

if (val[0] == 'd') {
    sunrise_duration = val.substring(1).toInt();
    sunrise_duration = sunrise_duration * 60000;
}

if (val[0] == 'b') {
    if (sunrise) {
        sunrise = false;
    }
}

```

```

    Serial.println(val.substring(1).toInt());
    brightness = map(val.substring(1).toInt(), 0, 100, 0, 255);
    analogWrite(LED, brightness);
}

if (val[0] == 's') {
    if (val.substring(1).toInt() == 1) {
        budilnik = true;
        Serial.println("будильник включен");
    }
    if (val.substring(1).toInt() == 0) {
        budilnik = false;
        Serial.println("Будильник выключен");
    }
}

if (val == "off") {
    digitalWrite(LED, LOW);
}

if (val == "on") {
    digitalWrite(LED, HIGH);
}

val = "";
recivied = false;
}

if ((h == sunrise_hour) && (m == sunrise_minute) && (!sunrise) && (budilnik) && (t.sec < 5)){
    Serial.println("SUNRISE STARTED");
}

```

```
    timer = millis();  
    sunrise = true;  
}  
  
if ((millis()-timer >= sunrise_duration) && sunrise) {  
    Serial.println("SUNRISE ENDED");  
    sunrise = false;  
}  
  
if (sunrise) {  
    sunrise_time = millis()-timer;  
    analogWrite(LED, map(sunrise_time, 0, sunrise_duration, 0, 255));  
}  
  
}
```