

Проектная работа

Физика

**«Создание версии розетки, управляемой звуком, на базе  
микроконтроллера Arduino»**

*Выполнила:*

***Сидоренко Дарья Дмитриевна***

*учащаяся 11 «А» класса*

*МБОУ СОШ № 46, Россия, г. Калуги*

*Руководитель:*

***Иванова Татьяна Анатольевна***

*Учитель физики*

*МБОУ СОШ № 46, Россия, г. Калуги*

*Я понял одну нехитрую истину. Она в том,  
чтобы делать чудеса своими руками.*

А.Грин

### **Введение.**

Система умный и интеллектуальный дом, представляющая собой автоматическую систему управления всеми автоматическими средствами, находящимися в доме, становится очень **актуальной** в современном мире, динамичном и требующим все новых и новых технологических новинок. Основная задача этого управления – обеспечить надежность и качество управления. Система умный и интеллектуальный дом дает возможность управлять всеми видами электропроводки: освещение, отопление, кондиционирование, сигнализация, электрические сети и прочее оборудование, имеющееся в доме. Выключатели, активируемые с помощью хлопка, стали популярными благодаря удобству применения. Они полезны для пожилых людей, лежачих больных. Также **эта система безопасна** в использовании даже ребенком, потому что нет непосредственного контакта с электроприбором. Она относится к новым современным технологиям, отличающимся своими эффективными и экономичными качествами. Розетка, управляемая звуком, которую я хочу создать, входит в систему «Умный» дом. Такая система представляется пока как неким элементом чуда. Вот и я, вслед за А. Грином, подумала, что «чудеса» можно делать своими руками.

### **Проблема**

Устройство Клаппер (от английского - хлопать в ладоши), которое позволяло включать и выключать приборы простым хлопаньем в ладоши, становится популярным в России с конца прошлого века. Это было полезное изобретение, но оно имело ограничение: при очень громком шуме устройство могло случайно сработать и отключить свет.<sup>1</sup>В настоящее время есть возможность купить подобное устройство (в основном, китайского производства), запрограммированное на выполнение конкретных действий. Стоят такие розетки относительно дорого. Поэтому я решила создать версию подобного устройства на базе микроконтроллера Arduino. Я думаю, устройство позволит устанавливать код для каждой розетки, исключая ложные срабатывания. Arduino также позволит запрограммировать продолжительность включения и выключения розетки.

**Цель:** создать версию розетки на базе микроконтроллера Arduino.

**Гипотеза:** «умную» розетку можно создать на базе Arduino и запрограммировать ее, разработав самостоятельно программу, исключив ложные срабатывания.

---

<sup>1</sup>Умный» дом по хлопку. Интернет источник/ <http://www.smarthomegroup.ru/Smart-home-history.html>

## Глава 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

### 1.1.«Умная» розетка как часть системы «умного» дома.

Мы живём в очень интересное время. Новые технологии постоянно внедряются в нашу жизнь, улучшая и облегчая ее. Например, технология «Умного» дома. Это «умное» управление освещением, вентиляцией, датчиками температуры и влажности, «умные» розетки.«Умная» розетка - это довольно недавнее новшество, которое получило огромную популярность у пользователей.

Что это такое? «Умная» розетка – это девайс, что представляет собою усовершенствованную электрическую розетку, которая контролирует состояние подключенной к ней электротехники. Она позволяет совершать удалённое управление электроприборами на расстоянии благодаря смартфону и специальному программному обеспечению, задавать время включения и выключения устройства, контролировать напряжение, отслеживать длительность работы розетки и многие другие функциональные возможности.<sup>2</sup>

Смарт-розетки предоставляют огромный спектр всевозможных функций и задач для их применения. К «умным» розеткам также относятся розетки, управляемые звуком. Главная особенность звукового выключателя заключается в том, что они реагируют на звук. К примеру, самый простой вариант — на хлопок. То есть, человеку достаточно воспроизвести звуковую команду для того, чтобы управлять светом в комнате или даже во всей квартире. Звуковой выключатель, или так называемый хлопковый выключатель, практичен и может всегда приятно удивить ваших гостей.

### 1.2.Виды звуковых выключателей.

Ассортимент данных коммутационных аппаратов на сегодняшний день является достаточно обширным. Это распространяется не только на виды, но и на ценовые категории и некоторые особенности отдельных вариантов. Более того, существуют даже дизайнерские модели, которые предназначены для интерьеров все тех же «умных» домов.

Но, в целом, все современные выключатели делятся на следующие виды:

Хлопковые;

---

<sup>2</sup>«Умные» розетки: назначение, разновидности, выбор и применение. Интернет источник/ <https://stroy-podskazka.ru/umnye-rozetki-i-vyklyuchateli/modeli/>

Акустические;

С датчиком движения и звука.

Хлопковый выключатель, как уже было сказано, срабатывает, улавливая звуки определенного количества хлопков.

Акустический выключатель реагирует на голос, какую-либо команду. Кодовые фразы могут быть стандартными: «включить», «выключить», но некоторые обладатели звуковых моделей предпочитают задавать свои варианты, которые точно не выкрикнешь случайно. Правда, аппараты, поддерживающие функцию выбора кодового слова, стоят несколько дороже.

Последний вариант из общего ассортимента является наиболее технологичным. Его конструкция устроена таким образом, чтобы реагировать на движения человека и его голос одновременно. Они существуют для того, чтобы не было сбоев в работе розетки.

Акустические выключатели могут реагировать на голосовые команды и хлопки. Существуют следующие **типы акустических выключателей**:

- Приборы, реагирующие на хлопок. Заранее программируется, при каком количестве хлопков будет выполняться та или иная команда.
- Приборы, реагирующие на голос или заранее поставленную команду.
- Комбинированные устройства (например, светоакустический выключатель). В них могут устанавливаться звуковые, световые датчики или сенсоры движения. Это наиболее технологичные устройства, у которых риск ложного срабатывания максимально снижен.
- Устройства для слаботочных систем. Используются для подключения видеокамеры или передачи команды охране.

Акустические выключатели стали популярными благодаря удобству применения. Они полезны для пожилых людей, лежачих больных и маленьких детей. Приборы такого класса уже активно используются в системе «Умный дом».<sup>3</sup>

**Выключатель по хлопку с микроконтроллером.** Микроконтроллер — это маленький кусочек кремния, покрытый пластиком и имеющий металлические выводы, который не выполняет никаких функций без программного обеспечения. Применение его в любой технике делает её умной и предполагает использование в системе «Умный дом».

---

<sup>3</sup> Хлопковый выключатель: принцип работы, схема подключение и монтаж. Источник: <https://electric-220.ru/hlopkovuj-vykljuchatel-princip-raboty-shema>

### 1.3. Принцип работы устройства.

Устройство функционирует за счет установленного в нем микроконтроллера. Контроллер санкционирует включение и выключение света по хлопку. Данный прибор при желании можно использовать для контроля за другими электробытовыми устройствами (кондиционерами, вентиляторами и т. п.).

В составе обычного звукового выключателя света имеется электронный микрофон с предусилителем. Этот компонент усиливает поступающий в устройство звук, что позволяет фиксировать даже самые негромкие хлопки. Деятельность усилителя управляется с помощью транзисторов VT1 и VT2. Схема контролируется парой резисторов R2, а для выравнивания сигнала используются диоды VD1 и VD2.

Звук от хлопка проходит через микрофон, там усиливается и трансформируется электрический импульс. Далее происходит выравнивание звука благодаря работе выпрямляющих диодов. Звук находится под контролем резистора (если уровень громкости звука, ниже заданного, резистор не допустит срабатывания прибора). Когда выравнивается сигнал на конденсаторе, возрастает напряжение, открывается транзисторный ключ VT3.

Свет отключается и включается после последовательной зарядки и разрядки конденсаторов. По окончании полного цикла деятельности (по еще одному хлопку) резистор и конденсатор C10 разряжаются за четыре секунды. Это приводит устройство к выключению.

### 1.4. Возможно ли сделать «умную» розетку своими руками?

Этот вопрос меня заинтересовал. Опытный знакомый инженер мне сказал, что своими руками сделать звуковой коммутационный аппарат сможет только тот, кто уже имеет хоть какой-то опыт в работе с радиоаппаратурой. Новичку в этом деле лучше всего будет использовать самые простые схемы, найти которые тоже не так-то просто. Поэтому, чтобы научиться создавать элементарные модели, используя радиосхемы, я создала несколько проектов на базе Arduino. Это – один из популярнейших микроконтроллеров для создания разнообразных автоматизированных систем. Сейчас можно увидеть большое разнообразие изделий на основе Arduino и столько же инструкций по тому, как и что делать. Что же собой представляет данный микроконтроллер и к чему стоит быть готовым

новичкам? Достоинствами Arduino являются: библиотеки, которые создаются не только авторами платы, но и сообществом. Благодаря этому можно найти подходящий инструментарий под любую задачу. Но здесь кроется и главный недостаток. Никто не контролирует качество кода, и в результате большую часть библиотек из свободного доступа вам придётся вручную модифицировать или переписывать десятки раз. Поэтому многие предпочитают самостоятельно написать код и базовый функции, если есть такая возможность. Я вначале тоже взяла готовую программу, но потом, когда поняла, что есть сбои, написала свою. Конечно, если вы никогда раньше не имели опыта с программным кодом и не знаете базовых алгоритмов – лучше пользоваться заготовленными библиотеками.

### 1.5. Микроконтроллер Arduino.

Итак, что собой представляет микроконтроллер? Это небольшое устройство, к которому подключаются все остальные элементы системы. Это лишь инструмент, который позволяет координировать работу всей системы. А делает он это при помощи встроенных в него библиотек, которые можно устанавливать в систему дополнительно, по необходимости. Вплоть до того, что вы можете поставить вспомогательную карту памяти, если не хватает места.<sup>4</sup>

Почему я советую воспользоваться Arduino тем, кто только начинает осваивать программирование? Arduino — не просто плата, позволяющая подключить различные устройства. Это мощная база, которую можно использовать для создания «Умного дома». При этом нет нужды тратить большие деньги за дорогостоящие устройства, стоимость которых в 5-10 раз больше. Управление автоматикой возможно через Интернет или посредством сообщений. Так что Arduino отлично подходит для создания устройств повышенной сложности.

---

<sup>4</sup>Что такое микроконтроллеры - назначение, устройство, софт. Интернет источник/<http://elektrik.info/main/automation/549-cto-takoe-mikrokontrollery-naznachenie-ustroystvo-princip-raboty-soft.html>



## Глава 2. ПРАКТИЧЕСКАЯ ЧАСТЬ.

### 2.1. План-график работы над проектом

Сроки	Мероприятие	Результат
Сентябрь-октябрь	Изучение материала по теме Изучение мнения по данной теме	Создание плана работы, руководство к действию Анализ анкет.
Ноябрь	Изучение возможностей микроконтроллера Arduino	Подготовка всех необходимых материалов для создания розетки
Декабрь	Работа по созданию розетки	Создание розетки
Январь	Программирование работы розетки	Создание версии управляемой розетки на базе микроконтроллера Arduino. Проведение эксперимента.
Февраль	Обобщение работы.	Создание презентации.
Апрель-май	Участие в защите проекта	Получение «зачета» по проекту

### 2.2. Предварительный этап. Изучение материала по теме и опрос мнений

Перед тем, как приступить к созданию розетки, я изучила материал по теме и поняла, как собрать умную розетку, а также ответила на несколько вопросов.

#### Какими бывают умные розетки?

Существует два вида умных розеток: наружные и встраиваемые. Наружные розетки не требуют монтажа: сзади имеется обычная электрическая вилка, поэтому их можно подключить в любую розетку или сетевой фильтр как переходник. Встраиваемые смарт-розетки внешне практически не отличаются от обычных, их может выдать разве что индикатор или кнопка на корпусе. Задача у смарт-розеток одинаковая. Они подают питание по команде или по расписанию.

#### Для чего нужны умные розетки?

Умные розетки используют для экономии электроэнергии и ее учета, автоматизации работы устройств и дистанционного управления приборами. Например, с помощью умной розетки можно автоматизировать освещение дачного участка или мини-сада на подоконнике. Устройство будет по графику подавать питание к подключенному прибору.

Самое простое, что можно подключить, это обогреватель, кондиционер или даже небольшой электрод котел отопления. В конце рабочего дня нужно удаленно активировать розетку, тогда по приходу домой температура будет оптимальной, особенно если розетка умеет работать в качестве термостата.

Также к умной розетке можно подключить вентиляцию, рекуператор или оконный электропривод, чтобы проветривать квартиру перед вашим возвращением. Все эти проекты можно создать, используя Arduino.

Я решила спросить близких мне людей, видят ли они практическую значимость в создании умной розетки. В опросе участвовало 27 человек разного возраста (родители, родственники, друзья) (рис. 1.1).

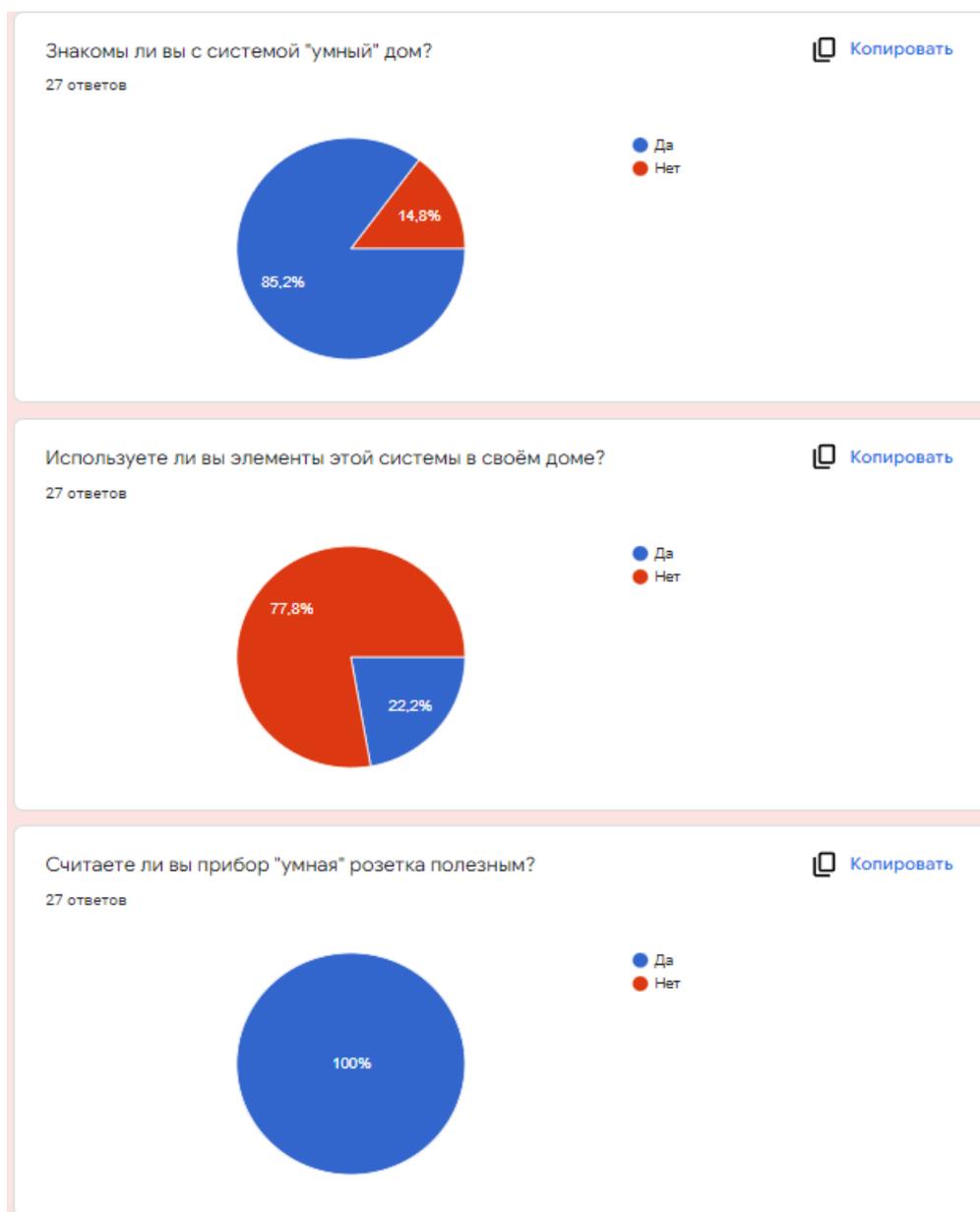


рис. 1.1 - Результаты опроса

После того, как я объяснила, что можно сделать, используя умную розетку, все 27 человек утвердительно ответили на вопрос: видят ли они практическую значимость проекта?

### 2.3. Подготовка к сборке

Для того чтобы приступить к сборке розетки, я подготовила все необходимые материалы(рис.1.2). Часть из них я нашла в наборе Arduino, некоторые детали я купила в магазине: резистор на 100 кОм, конденсатор на 0.1 мкФ, микрофон, печатная плата универсальная GSMIN PCB1 (7 см на 9 см)и аккумулятор на 5В.Также я рассчитала затраты на изготовление розетки.



#### 2.4. Список необходимых деталей и экономический расчет.

Тип	Номинал	Количество	Цена (руб)
Плата Arduino	ArduinoUno	1	450
Полевой транзистор		1	75
Диод		1	5
Конденсатор	0.1 мкФ	1	68
Резистор	10 кОм	2	25
Резистор	100 кОм	1	15
Микрофон		1	35
Реле		1	118
Выключатель		1	62
Розетка		1	130
Вилка		1	124
Универсальная печатная плата РСВ 7х9 см		1	140
<b>Итого</b>			<b>1 247</b>

В магазинах города Калуги можно найти практически все необходимые детали для создания умной розетки.<sup>5</sup>

Если все покупать в магазине, сумма выходит большая. Но для меня этот «набор» деталей получился гораздо дешевле, т.к. часть из них уже были куплены ранее: выключатель, розетка, вилка, а также плата Arduino. Таким образом, мне пришлось затратить **481 рубль**.

Если сравнить цены в магазинах на «умные» розетки, они будут значительно выше.

<sup>5</sup> Ссылка на адреса магазинов радиоэлектроники в Калуге <https://2gis.ru/kaluga/search/Магазины%20радиодеталей%20и%20приборов/rubricId/463>

Название модели	Цена
Умная розетка HIPER Outlet W02 Duo, белый	3 190 р.
Хлопковый выключатель «CLAPS MAX»	2 470 р.
Хлопковый выключатель «CLAPS MAX Free»	2 370р.

## 2.5. Сборка розетки по схеме

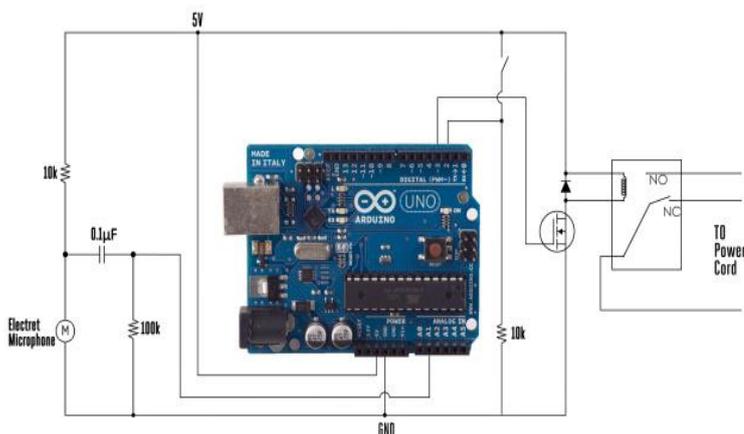


рис. 1.2 - Полная схема

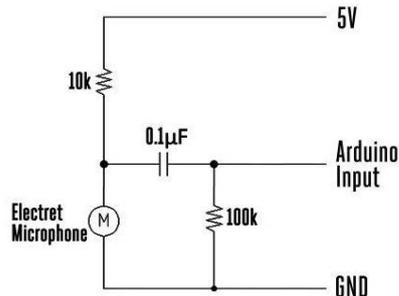


рис 1.3

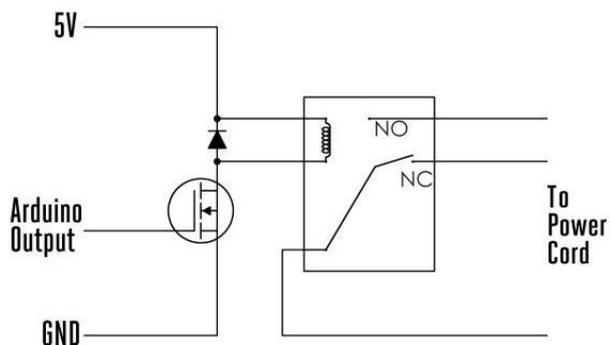


рис 1.4

Схема управления электретного микрофона в сборе состоит из самого электретного микрофона, двух резисторов и конденсатора. Когда микрофон улавливает звуковые

вибрации, то соответственно меняется выходное напряжение. Данный сигнал подается на один из аналоговых входов микроконтроллера Arduino.

Релейный модуль состоит из МОП-транзистора, диода и реле. Силовой транзистор добавлен в схему, поскольку для управления реле необходимо обеспечить больший ток, чем выдает цифровой выход микроконтроллера Arduino. Поэтому сигнал, выдаваемый цифровым выходом, активирует транзистор, который в свою очередь подает питание на реле и включает его. Реле коммутирует напряжение переменного тока и соответственно включает, и выключает бытовое устройство. Диод действует как блокирующий элемент и защищает микроконтроллер Arduino от всплесков напряжения, которые происходят при отключении устройства. Если мне нужно будет перепрограммировать последовательность хлопков без модификации кода, тогда подключу выключатель к цифровому выводу 2. Данный выключатель подключает вывод 2 к GND в обычном рабочем режиме, и к источнику питания напряжением 5В для внесения изменений в программном режиме.

GND (GROUND, перевод — земля) — точка нулевого потенциала микросхемы<sup>[14]</sup>.

Электретный микрофон - один из подвидов конденсаторных устройств<sup>[13]</sup>.

### **2.6.Разработка и загрузка программного кода**

Мной был разработан программный код для «умной» розетки в среде разработки ArduinoUno на языке программирования C++, который я изучала самостоятельно. Этот язык является низкоуровневым, поэтому считается сложным и имеет высокий порог вхождения. Но для программирования Arduino используется **упрощенная версия этого языка программирования**. Также для упрощения разработки прошивок существует множество функций, классов, методов и библиотек. Благодаря этому работать с этими микроконтроллерами очень удобно и несложно. Хотя требуется достаточно времени, чтобы разобраться со всеми функциями. (Приложение № 2)

( Код представлен в приложении №6 )

## 2.7. Пайка элементов схемы на печатной плате

Как только я спроектировала схему, приступила к ее пайке на печатной плате (рис.1.6). Я узнала, что существуют определенные правила при пайке электронных модулей радиоэлектронных средств, которые прописаны в национальном стандарте РФ.<sup>6</sup>

Пайка является основой сборки печатного узла. Пайка объединяет две или несколько металлических поверхностей в одно металлургическое соединение.



Процесс пайки – это нанесение расплавленного припоя на обработанные флюсом поверхности, флюс наносится с целью смачивания припоем паяемых поверхностей, смачивание требуется для получения при пайке металлургического соединения.

Смачиваемость определяется как образование однородной, гладкой, не имеющей разрывов и прилипающей пленки припоя на основном металле. Паяемость — свойство металлической поверхности, позволяющее смачивание ее припоем. Процесс пайки печатного узла заключается в одновременной подаче тепла и припоя.<sup>7</sup>

При пайке я использовала правила безопасности.(Приложение №2).

После того, как паяльные работы были выполнены, я подсоединила плату к микроконтроллеру Arduino с помощью проволочных перемычек.

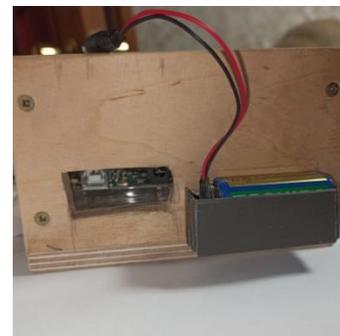
---

<sup>6</sup>Электронный фонд правовой и нормативно-технической информации - docs.cntd.ru

<sup>7</sup> Технология ручной пайки и лужения. Источник: <https://gostost.ru/paika-pechatnie-platy>

## 2.8. Установка компонентов внутри изолированного корпуса

Установила компоненты внутри изолированного корпуса, созданного мной из фанеры. Фанера - это один из самых простых в обработке материалов. Качественная фанера великолепно обрабатывается ручным и электрическим инструментом и ее несложно прикрепить гвоздями, как другому листу фанеры, так и к массиву. Вместе с тем при обработке следует соблюдать ряд правил. (Приложение № 3)



Я использовала горячий клей для прикрепления платы внутри корпуса. Далее я вырезала несколько отверстий для кабелей. Сделала прорезь для выключателя, подключила провода и выключатель в прорезанные посадочные места, и зафиксировала на месте горячим клеем.

## 2.9. Управление устройством с помощью простого хлопка в ладоши

Звуковой датчик включения работает по двум основным принципам. В первом случае он нужен, чтобы сразу после раздавшегося звука, зажечь отключенное в тишине освещение. Команда, поданная на реле раздавшимся шумом, включает свет на время, достаточное для прохождения определенного расстояния. По такому принципу работает датчик на лестничных пролетах, в темных местах двора, на участках земельных владений, где нет стационарного освещения или просто в нем нет особенной необходимости.

Второй принцип работы обеспечивает реле задержки. Он позволяет устройству срабатывать только от определенного шума – характерного или продолжительного. Такое реле включения света позволяет не тратить электроэнергию на движение более мелкого объекта, чем человек домашнего животного, грызуна на приусадебном участке, не реагировать на посторонние шумы – природные или механического происхождения.

Чтобы проконтролировать работу устройства с помощью хлопка в ладоши, подключаю устройство и хлопаю закодированную последовательность. Можно также создать и другой шум (постучать по столу, например) (Приложение 4). Для данного кода устройство должно включиться на десять секунд и затем автоматически выключиться. Изменить длительность работы можно, скорректировав код.



### Глава 3. ОЦЕНКА ЭФФЕКТИВНОСТИ ПРОЕКТА.

#### 1.1. Индикаторы и практическая направленность продукта.

1. Для оценки эффективности и полезности своего проекта я обратилась к заместителю директора по воспитательной работе школы с вопросом: найдется ли применение моему проекту на новогодних мероприятиях.

Ответ: забираться под Рождественскую елку, чтобы каждый раз включать и выключать свет, может быть непросто. Автоматизация праздничных огней с помощью умных розеток - самое удобное решение. Я бы использовала это устройство на новогоднем мероприятии и у себя дома.

Новогодние праздники рано или поздно закончатся, и ёлку с игрушками придётся убрать до следующего года. Умную розетку отключать не стоит: она рассчитана и под более мощные электроприборы. К ней можно подключить обогреватель или торшер с лампой, а потом управлять их работой.

2. Я поместила в Google Формы опрос:

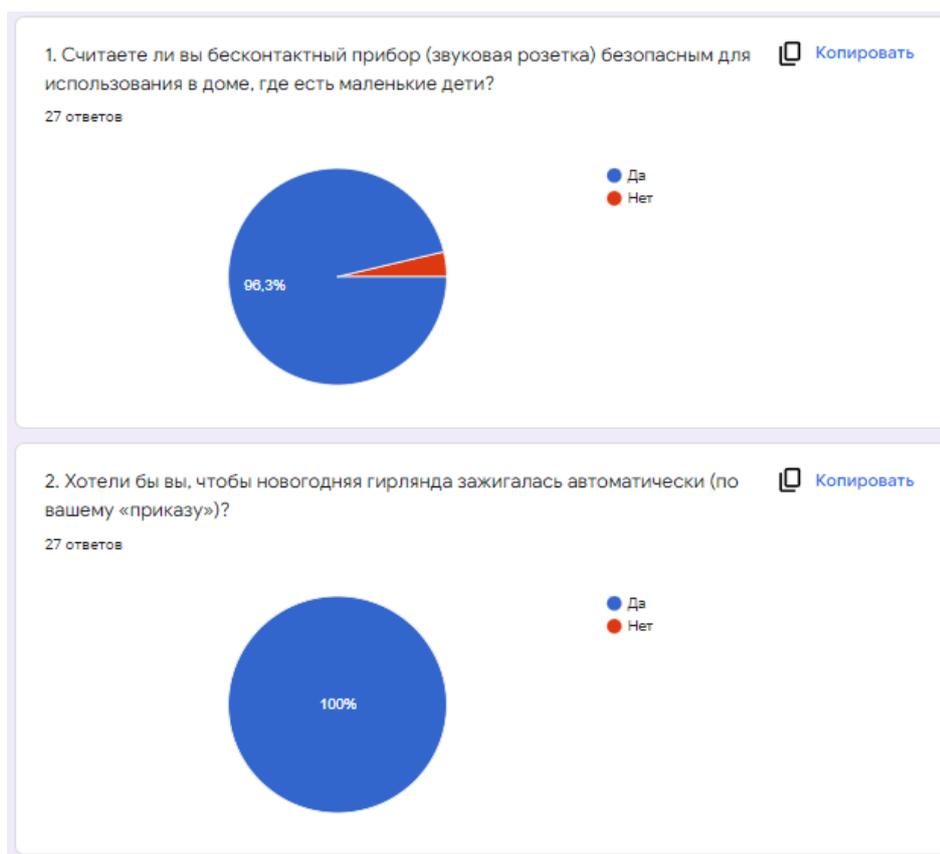


рис. 1.5 - Результаты опроса

Таким образом, я сделала вывод, что прибор, изготовленный мною, будет полезен, и ему найдется применение.

## Заключение

Работа над проектом позволила мне создать умную розетку, управляемую звуком. Я выполнила все намеченные задачи и внесла некоторые изменения. С целью усиления значимости проекта я подключила к розетке новогоднюю гирлянду и придумала новогодний «сигнал» для запуска работы розетки. Таким образом, данный продукт можно использовать на новогодних мероприятиях. Задача подготовить «чудо» своими руками выполнена.

### Чему я научилась за время работы над проектом:

1. Систематизировала и расширила знания работы над проектом.
2. Познакомилась с работой и возможностями Arduino. На его основе я собрала несколько простых устройств. Например, маячок и мерзкое пианино.
3. Познакомилась с синтаксисом языка программирования C++.
4. Впервые попробовала работать с паяльником.

Обобщая работу над проектом, я создала презентацию, куда включила фото и видео, подтверждающие самостоятельность работы над проектом.

В заключение хочется отметить, что, как правило, построение интеллектуального дома начинается именно с функции управления освещением, поскольку одна эта функция способна значительно преобразить наш опыт пребывания в своем доме.

В дальнейшем я бы хотела продолжить совершенствование системы «умного» дома и создать установку автоматического полива комнатных растений с помощью датчика влажности почвы. Также хочу поучаствовать в конференциях и конкурсах по направлению «Умный» дом.

### Список используемой литературы.

1. Игнатьева, Е. А. Умный дом – технология будущего / Е. А.
  2. Детлаф, А.А. Курс общей физики / А.А. Детлаф, Б.М. Яворский. - М. Высшая школа, 2017. - 245 с.
  3. Березин Б.И., Березин С.Б. Начальный курс С и С++. — М.: ДИАЛОГ-МИФИ, 1996.
  4. Ван Тассел Д. Стилль, разработка, эффективность, отладка и испытание программ. — М.: Мир, 1981.
  5. Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989.
- Интернет-ресурсы:
6. «Умные» розетки: назначение, разновидности, выбор и применение[Электронный ресурс]<https://stroy-podskazka.ru/umnye-rozetki-i-vyklyuchateli/modeli/>
  7. Звуковой выключатель в обычный дом: нужно ли? [Электронный ресурс]<http://electry.ru/vyiklyuchateli/zvukovoy-vyiklyuchatel.html>
  8. История «умного дома». [Электронный ресурс] [http://efarostov.ru/istoriya\\_ymnogo\\_doma.htm](http://efarostov.ru/istoriya_ymnogo_doma.htm)
  9. История возникновения и развития «умного» дома. [Электронный ресурс] <https://progress.online/tehnologii/1686-istoriya-vozniknoveniya-i-razvitiya-umnogo-doma>
  10. Умная розетка: что можно подключить и как автоматизировать свой дом[Электронный ресурс]<https://club.dns-shop.ru/blog/t-238-drugoe-dlya-umnogo-doma/53612-umnaya-rozetka-chto-mojno-podkluchit-i-kak-avtomatizirovat-svoi/>
  11. Умный дом и бытовая техника: дистанционное управление или нечто большее?[Электронный ресурс]<https://www.ixbt.com/home/smart-home-2021.html>
  12. Умный дом на основе Arduino. [Электронный ресурс] <https://future2day.ru/umnyj-dom-na-osnove-arduino/>
  13. Электретные микрофоны: что это такое и как это подключить? [Электронный ресурс]<https://stroy-podskazka.ru/mikrofony/elektretnyye/>
  14. GND - что это такое на схеме? (или на материнской плате) [Электронный ресурс]<http://virtmachine.ru/gnd-na-materinskoj-plate-sheme-chto-eto-znachit.html>

## Приложение № 1.

### Общие требования к рабочему месту при работе с паяльником.

Основы безопасности <https://masterkit.ru/blog/lessons/urok-3-osnovy-montazha-i-pajki>

Для участка пайки и монтажа, сопровождающегося пайкой, следует отводить отдельное помещение. Рабочие поверхности оборудования необходимо покрывать легкообмываемыми материалами; стационарные места пайки необходимо оборудовать местными вытяжными устройствами, обеспечивающими скорость воздуха непосредственно на месте пайки не менее 0,6 м/с. Помещения, в которых производится пайка, должны обеспечиваться приточным воздухом, подаваемым в верхнюю зону. В процессе пайки пары и газы могут образовываться за счет сгорания загрязнений, имеющихся на поверхности паяемых деталей, а также при нагреве флюсов и припоев. Наиболее вредны испарения галоидных флюсов, особенно содержащих фториды. Все емкости с флюсами, содержащими соединения фтора, должны иметь наклейки с указанием мер предосторожности при их применении. Нельзя допускать образования паров или пыли кадмия, свинца и цинка. Вредное действие их усиливается, когда эти металлы применяются в чистом виде. Перегрев при пайке усиливает испарение отдельных компонентов припоев.

### Памятка.

- рабочее место (стол) не должен быть захламлён. На свободном столе работать приятнее и эффективнее. Кроме того, радиодетали не смогут легко потеряться в окружающем хламе;
- Так как радиодетали мелкие, во избежание излишнего перенапряжения глаз рабочее место должно быть хорошо освещено. Можно включить настольную лампу (Вместо настольной лампы я использовала специальные очки)
- во время пайки предусмотрите хорошую вентиляцию рабочего места. Необходимо открыть форточку, или включить настольный вентилятор, отгоняющий дым от паяльника в сторону;
- паяльник горячий! Надо держать только за его ручку. Не допускайте прикосновений пальцев к жалу;
- после пайки, как и после любой другой работы, необходимо вымыть руки.

## Приложение № 2.

### Программирование Arduino.

Программирование микроконтроллеров Arduino осуществляется на языке программирования C++. Этот язык является низкоуровневым, поэтому считается сложным и имеет высокий порог вхождения. Но для программирования Arduino используется **упрощенная версия этого языка программирования**. Также для упрощения разработки прошивок существует множество функций, классов, методов и библиотек. Благодаря этому работать с этими микроконтроллерами очень удобно и легко. Описание большинства необходимых функций и операторов вы найдете на этой странице. Этого хватит для написания прошивок под самые разнообразные устройства на базе Arduino. Для всех функций есть небольшие примеры, показывающие, как вы можете их использовать.

<p>Операторы</p> <ul style="list-style-type: none"> <li>• <a href="#">setup()</a></li> <li>• <a href="#">loop()</a></li> </ul> <p>Управляющие операторы</p> <ul style="list-style-type: none"> <li>• <a href="#">if</a></li> <li>• <a href="#">if...else</a></li> <li>• <a href="#">for</a></li> <li>• <a href="#">switchcase</a></li> <li>• <a href="#">while</a></li> <li>• <a href="#">do... while</a></li> <li>• <a href="#">break</a></li> <li>• <a href="#">continue</a></li> <li>• <a href="#">return</a></li> <li>• <a href="#">goto</a></li> </ul> <p>Синтаксис</p> <ul style="list-style-type: none"> <li>• <a href="#">; (точка с запятой)</a></li> <li>• <a href="#">{} (фигурные скобки)</a></li> <li>• <a href="#">// (однострочный комментарий)</a></li> <li>• <a href="#">/* */ (многострочный комментарий)</a></li> <li>• <a href="#">#define</a></li> <li>• <a href="#">#include</a></li> </ul> <p>Арифметические операторы</p> <ul style="list-style-type: none"> <li>• <a href="#">= (оператор присваивания)</a></li> </ul>	<p>Битовые операторы</p> <ul style="list-style-type: none"> <li>• <a href="#">&amp; (побитовое И)</a></li> <li>• <a href="#">  (побитовое ИЛИ)</a></li> <li>• <a href="#">^ (побитовое XOR или исключающее ИЛИ)</a></li> <li>• <a href="#">~ (побитовое НЕ)</a></li> <li>• <a href="#">&lt;&lt; (побитовый сдвиг влево)</a></li> <li>• <a href="#">&gt;&gt; (побитовый сдвиг вправо)</a></li> </ul> <p>Составные операторы</p> <ul style="list-style-type: none"> <li>• <a href="#">++ (инкремент)</a></li> <li>• <a href="#">-- (декремент)</a></li> <li>• <a href="#">+= (составное сложение)</a></li> <li>• <a href="#">-= (составное вычитание)</a></li> <li>• <a href="#">*= (составное умножение)</a></li> <li>• <a href="#">/= (составное деление)</a></li> <li>• <a href="#">&amp;= (составное побитовое И)</a></li> <li>• <a href="#"> = (составное побитовое ИЛИ)</a></li> </ul> <p>Данные</p> <p>Константы</p> <ul style="list-style-type: none"> <li>• <a href="#">HIGH   LOW</a></li> <li>• <a href="#">INPUT   OUTPUT   I</a></li> </ul>	<p>Функции</p> <p>Цифровой ввод/вывод</p> <ul style="list-style-type: none"> <li>• <a href="#">pinMode()</a></li> <li>• <a href="#">digitalWrite()</a></li> <li>• <a href="#">digitalRead()</a></li> </ul> <p>Аналоговый ввод/вывод</p> <ul style="list-style-type: none"> <li>• <a href="#">analogReference()</a></li> <li>• <a href="#">analogRead()</a></li> <li>• <a href="#">analogWrite()</a> — <i>PWM</i></li> </ul> <p>Расширенный ввод/вывод</p> <ul style="list-style-type: none"> <li>• <a href="#">tone()</a></li> <li>• <a href="#">noTone()</a></li> <li>• <a href="#">shiftOut()</a></li> <li>• <a href="#">shiftIn()</a></li> <li>• <a href="#">pulseIn()</a></li> </ul> <p>Время</p> <ul style="list-style-type: none"> <li>• <a href="#">millis()</a></li> <li>• <a href="#">micros()</a></li> <li>• <a href="#">delay()</a></li> <li>• <a href="#">delayMicroseconds()</a></li> </ul> <p>Математические вычисления</p> <ul style="list-style-type: none"> <li>• <a href="#">min()</a></li> <li>• <a href="#">max()</a></li> <li>• <a href="#">abs()</a></li> <li>• <a href="#">constrain()</a></li> <li>• <a href="#">map()</a></li> <li>• <a href="#">pow()</a></li> </ul>
---	--	--

<ul style="list-style-type: none"> <li>• <a href="#">+</a> (сложение)</li> <li>• <a href="#">—</a> (вычитание)</li> <li>• <a href="#">*</a> (умножение)</li> <li>• <a href="#">/</a> (деление)</li> <li>• <a href="#">%</a> (остаток от деления)</li> </ul> <p>Операторы сравнения</p> <ul style="list-style-type: none"> <li>• <a href="#">==</a> (равно)</li> <li>• <a href="#">!=</a> (не равно)</li> <li>• <a href="#">&lt;</a> (меньше чем)</li> <li>• <a href="#">&gt;</a> (больше чем)</li> <li>• <a href="#">&lt;=</a> (меньше или равно)</li> <li>• <a href="#">&gt;=</a> (больше или равно)</li> </ul> <p>Логические операторы</p> <ul style="list-style-type: none"> <li>• <a href="#">&amp;&amp;</a> (И)</li> <li>• <a href="#">  </a> (ИЛИ)</li> <li>• <a href="#">!</a> (НЕ)</li> </ul> <p>Указатели доступа</p> <ul style="list-style-type: none"> <li>• <a href="#">*</a> <u>указатель</u></li> <li>• <a href="#">&amp;</a> <u>ссылка</u></li> </ul>	<p><a href="#">NPUT_PULLUP</a></p> <ul style="list-style-type: none"> <li>• <a href="#">true</a>   <a href="#">false</a></li> <li>• <a href="#">целочисленные константы</a></li> <li>• <a href="#">константы с плавающей точкой</a></li> </ul> <p>Типы данных</p> <ul style="list-style-type: none"> <li>• <a href="#">void</a></li> <li>• <a href="#">boolean</a></li> <li>• <a href="#">char</a></li> <li>• <a href="#">unsignedchar</a></li> <li>• <a href="#">byte</a></li> <li>• <a href="#">int</a></li> <li>• <a href="#">unsignedint</a></li> <li>• <a href="#">word</a></li> <li>• <a href="#">long</a></li> <li>• <a href="#">unsignedlong</a></li> <li>• <a href="#">short</a></li> <li>• <a href="#">float</a></li> <li>• <a href="#">double</a></li> <li>• <a href="#">string</a> — массив символов</li> <li>• <a href="#">массивы</a></li> </ul> <p>Преобразование типов</p> <ul style="list-style-type: none"> <li>• <a href="#">char()</a></li> <li>• <a href="#">byte()</a></li> <li>• <a href="#">int()</a></li> <li>• <a href="#">word()</a></li> <li>• <a href="#">long()</a></li> <li>• <a href="#">float()</a></li> </ul> <p>Область видимости переменной и спецификаторы</p> <ul style="list-style-type: none"> <li>• <a href="#">Область видимости переменной</a></li> <li>• <a href="#">static</a></li> <li>• <a href="#">volatile</a></li> <li>• <a href="#">const</a></li> </ul> <p>Вспомогательная функция</p> <ul style="list-style-type: none"> <li>• <a href="#">sizeof()</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">sqrt()</a></li> <li>• <a href="#">sq()</a></li> </ul> <p>Тригонометрия</p> <ul style="list-style-type: none"> <li>• <a href="#">sin()</a></li> <li>• <a href="#">cos()</a></li> <li>• <a href="#">tan()</a></li> </ul> <p>Случайные числа</p> <ul style="list-style-type: none"> <li>• <a href="#">randomSeed()</a></li> <li>• <a href="#">random()</a></li> </ul> <p>Биты и байты</p> <ul style="list-style-type: none"> <li>• <a href="#">lowByte()</a></li> <li>• <a href="#">highByte()</a></li> <li>• <a href="#">bitRead()</a></li> <li>• <a href="#">bitWrite()</a></li> <li>• <a href="#">bitSet()</a></li> <li>• <a href="#">bitClear()</a></li> <li>• <a href="#">bit()</a></li> </ul> <p>Внешние прерывания</p> <ul style="list-style-type: none"> <li>• <a href="#">attachInterrupt()</a></li> <li>• <a href="#">detachInterrupt()</a></li> </ul> <p>Прерывания</p> <ul style="list-style-type: none"> <li>• <a href="#">interrupts()</a></li> <li>• <a href="#">noInterrupts()</a></li> </ul>
--	---	---

## ОСНОВЫ РАБОТЫ С ФАНЕРОЙ

**Фанера** - это один из самых простых в обработке материалов. Качественная фанера великолепно обрабатывается ручным и электрическим инструментом и ее несложно прикрепить гвоздями, как другому листу фанеры, так и к массиву. Вместе с тем при обработке следует соблюдать ряд правил.

### Распиловка фанеры

Для работы с фанерой используются ленточные и дисковые пилы с небольшими зубьями. Главное и непреложное правило - распил должен проводиться поперек волокна: в противном случае возможны многочисленные сколы.

### Сверление фанеры

Сверлить фанеру несложно: гораздо сложнее делать это так, чтобы отверстия были ровным, без сколов и задиров. С этой целью применяют подкладочный лист или ДСП, который плотно прижимается к обрабатываемому. Еще один немаловажный нюанс - качество сверла: оно должно иметь передний резак и хорошую заточку.

### Крепление в фанере при помощи гвоздей и шурупов

Фанера хорошо крепится при помощи любых гвоздей, однако лучше всего для работы с ней использовать резьбовые гвозди или фанерные шурупы. В ряде случаев используются проволочные гвозди, которые позволяют избежать выступов шляпки.

Считается, что длина гвоздя должна превышать толщину фанерного листа в два три раза. На самом деле это правило не является очень строгим: зачастую работать приходится с длинными гвоздями. Оптимальное расстояние между гвоздями зависит от толщины листа, типа и величины нагрузок, а также от типа используемых гвоздей. В среднем расстояние составляет около 20 см, хотя при подготовке основы для укладки паркета гвозди вбивают не ближе, чем в 40 см друг от друга. Что касается расстояния до края, то отступ может составлять всего полтора-два сантиметра: обычно этого бывает вполне достаточно.

### Резьбовые соединения фанеры

Прикрепить лист фанеры можно как с помощью обычного болта с гайкой, так и при помощи саморезного винта. И в том и в другом случае фанеру необходимо защитить при помощи шайбы, поскольку со временем головка болта или винта способна прорезать в тонком шпоне довольно глубокие отверстия, а при колебаниях влажности воздуха и вовсе расколоть лист.

### Склеивание фанер

Для склеивания можно использовать древесный клей любой марки, которая подойдет для конкретных условий эксплуатации изделия. Так, ПВА используется при проведении внутренних работ, а фенольные и эпоксидные смолы - для изготовления изделий, эксплуатируемых на открытом воздухе: кроме того, эпоксидные составы незаменимы при склеивании металла и фанеры. Клей необходимо наносить на сухую и чистую поверхность шпона. Именно шпона - если для приклеивания используется ламинированная фанера, то пленку необходимо счистить. При склеивании обязательно необходимо промазывать обе поверхности: для этого используется кисть или валик.

### Механическая шлифовка фанеры

Для шлифовки применяют наждачную бумагу, предназначенную для работы по дереву. Шлифуют поперек волокна: иногда можно использовать также шлифовку круговыми движениями.

**Приложение № 4.**

**Проверка на плагиат.**



## Отчет о проверке на заимствования №1



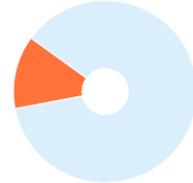
**Автор:** Сидоренко Дарья  
**Проверяющий:** Сидоренко Дарья  
 Отчет предоставлен сервисом «Антиплагиат» - <http://users.antiplagiat.ru>

### ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 2  
 Начало загрузки: 25.04.2022 21:04:12  
 Длительность загрузки: 00:00:00  
 Имя исходного файла: Проект Сидоренко.txt  
 Название документа: Проект Сидоренко  
 Размер текста: 41 кБ  
 Символов в тексте: 42079  
 Слов в тексте: 5170  
 Число предложений: 374

### ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 25.04.2022 21:04:13  
 Длительность проверки: 00:00:01  
 Комментарии: не указано  
 Модули поиска: Интернет Free



ЗАИМСТВОВАНИЯ  
12,53%

САМОЦИТИРОВАНИЯ  
0%

ЦИТИРОВАНИЯ  
0%

ОРИГИНАЛЬНОСТЬ  
87,47%

Оригинальность 87,47%

Заимствования 12,53%

Цитирования 0%

Самоцитирования 0%

ПОЛНЫЙ ОТЧЕТ

КРАТКИЙ ОТЧЕТ

ИСТОРИЯ ОТЧЕТОВ

РАСПЕЧАТАТЬ

ВЫГРУЗИТЬ

СОЗДАТЬ ССЫЛКУ

Свойства документа

Структура документа

Параметры проверки

Текстовые метрики

Статистика по документу

#### ИМЕНОВАННЫЕ СУЩНОСТИ

Чтобы загрузить список именованных сущностей, нажмите на кнопку справа.

ПОСМОТРЕТЬ СПИСОК

#### СЕМАНТИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Доля научной лексики: 100  
 Доля общей лексики: 0  
 Наличие аннотации: нет  
 Наличие библиографии: да  
 Наличие выводов: нет  
 Наличие введения: да  
 Наличие описания метода исследования: нет  
 Наличие описания результатов исследования: да  
 Оценка связности текста: 70.83333333333333

## Приложение №5

### Плюсы и минусы звуковых выключателей.

Звуковой или хлопковый выключатели сами по себе являются довольно уникальными аппаратами. Их главное преимущество перед стандартными моделями заключается в том, что человеку не нужно каждый раз прикладывать усилия, чтобы отключить или выключить свет. Кроме того, сделать это можно, находясь в любой части комнаты.

Их основной минус для радиолюбителей будет в трудновоспроизводимой схеме. Все-таки, конструкция таких аппаратов куда сложнее, а значит, и сделать их своими руками далеко не так просто.

Кроме того, хлопковый выключатель, также, как и звуковой, имеет ряд небольших недостатков, которые будут заметны абсолютно всем: они могут реагировать на похожие звуки, особенно этим грешит голосовой вариант. Естественно, если произнести кодовую фразу в контексте, к примеру, обычного разговора, то свет в помещении среагирует на это. Устройства подобного типа рекомендуется ставить только в нешумных помещениях. «К помещениям такого типа относят спальни, кладовки, подсобные комнаты, подвалы. Нерационально устанавливать хлопковые устройства в помещениях, где бывает множество людей (офисы, гостиные, производственные территории), так как посторонние шумы приводят к ложным реакциям устройств. Наиболее распространенное место для установки выключателя света по хлопку — спальня. Очень удобно управлять освещением, не вставая с постели. Популярны хлопковые системы в семьях с маленькими детьми, так как для включения или выключения света теперь не нужно дотягиваться до высоко расположенного выключателя.

Мелкие недоработки могут стать причиной «заторможенности» реакции у данных аппаратов. Такое часто встречается, когда акустический выключатель, у которого конструкция несколько сложнее, был изготовлен своими руками. Так что, на этот фактор следует обратить особое внимание, прежде чем начать воспроизводить какие-либо схемы.

Если сигнал — хлопок или команда, будут недостаточно громкими, то аппарат, скорее всего, не среагирует. Низкая чувствительность у подобных выключателей нужна для того, чтобы человек реже сталкивался с первой проблемой этого списка.

В принципе, бороться с этими минусами можно, но полностью устранить их вряд ли сможет даже профессионал — поскольку, если, к примеру, повысить чувствительность, акустический выключатель начнет реагировать на «ложные» команды куда чаще. Но, как я думаю, любители подобных розеток «с эффектами» согласятся мириться с некоторыми недостатками. В данном случае главное, не включить, а как включить.

Если опираться на технические данные, то стандартный хлопковый и акустический выключатель работает при таких параметрах:

Напряжение сети должно быть 220 (В) — то есть, подойдет обычная лампа накаливания;

Мощность нагрузки не должна превышать 300 (Вт);

Температура в помещении не ниже -20 градусов, и не выше +45;

Звук регулируется от 30 до 150 децибелов;

Защита корпуса IP-30.

Сделать своими руками управление освещением голосом – задача не из простых. Для ее решения мало одного желания. Необходимо тщательно продумать схему, рассчитать электрические параметры, подобрать комплектующие, определиться, какое программное обеспечение будет использовано, не потребуется ли его модификация, что может подойти из уже существующих разработок, что можно модифицировать. Желательно уметь работать с паяльником, с тонкой электроникой.

```

//Пины подключения датчиков и устройств
constintNoiseSensor = 1;           //Датчик шума подключается в пин 1
constintprogramSwitch = 2;         //Программный переключатель
constintOutput = 3;                //Выходное реле

//Настройка констант
constintthreshold = 8;             //Минимальное значение шума, которое будет
распознавать датчик
constintrejectValue = 25;          //Погрешность значения шума
constintaverageRejectValue = 15;   //Погрешность значения времени шума
constintknockFadeTime = 150;       //Таймер устранения дребезга (время в миллисекундах
между отдельными сигналами)
constintmaximumKnocks = 20;        //Максимальное количество распознаваемых сигналов
constintknockComplete = 1200;     //Самое долго время ожидания звука, прежде чем мы
предположим, что он закончился

//Переменные
//Задаем шаблон сигналов
intsecretCode[maximumKnocks] = {50, 25, 25, 50, 100, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
intknockReadings[maximumKnocks]; //При распознавании звука массив заполняется задержками между
сигналами
intNoiseSensorValue = 0;           //Последнее показание датчика шума
intprogramButtonPressed = false;   //Флаг для программирования кнопки

void setup() {
pinMode(Output, OUTPUT);
pinMode(redLED, OUTPUT);
pinMode(greenLED, OUTPUT);
pinMode(programSwitch, INPUT);
}

void loop()
{
NoiseSensorValue = analogRead(NoiseSensor); //Считывание звука

if (digitalRead(programSwitch)==HIGH)       //Если кнопка нажата
{
programButtonPressed = true;               //Если да, то сохраняем состояние флага
}
else
{
programButtonPressed = false;
}

if (NoiseSensorValue>=threshold)
{
listenToSecretKnock();
}
}

voidlistenToSecretKnock()                  //Записываем время звуков
{
Serial.println("knock starting");

inti = 0;
//Сначала сбрасываю прослушивающий массив
for (i=0;i<maximumKnocks;i++)
{
knockReadings[i]=0;
}

intcurrentKnockNumber=0;                  //Текущий номер звука в массиве
intstartTime=millis();                    //Время начала звука
intnow=millis();

```

```

delay(knockFadeTime); //Ждем пока начнется новый сигнал
digitalWrite(greenLED, HIGH);

do //Ждем появления нового звука или завершаем работу
{
NoiseSensorValue = analogRead(NoiseSensor);
if (NoiseSensorValue>=threshold) //Записываем время задержки между сигналами
{
Serial.println("knock.");
now=millis();
knockReadings[currentKnockNumber] = now-startTime;
currentKnockNumber ++; //Увеличиваем счетчик массива на +1
startTime=now; //Меняем значение начала звука на текущее
время

delay(knockFadeTime); //Ждем, пока начнется новый сигнал
}

now=millis();
}
while ((now-startTime<knockComplete) && (currentKnockNumber<maximumKnocks));
//Проверяем, что звуки закончились

//Проверяем, что последовательность сигналов была получена
if (programButtonPressed==false)
{
if (validateKnock() == true)
{
triggerOutput();
}
else
{
Serial.println("Secret knock failed.");
Serial.println("");
}
}
else
{
validateKnock();
Serial.println("New code stored.");
}
}

//Проверка соответствия звука шаблону
//Возвращает true, если сигнал совпадает, false если нет.
boolean validateKnock(){
inti=0;
intcurrentKnockCount = 0;
intsecretKnockCount = 0;
intmaxKnockInterval = 0; //Максимальный интервал между звуками

for (i=0;i<maximumKnocks;i++){
if (knockReadings[i] > 0){
currentKnockCount++;
}
if (secretCode[i] > 0){
secretKnockCount++;
}
}

if (knockReadings[i] >maxKnockInterval){ //Меняем значение максимального звука на
текущий
//Если он превышает максимальное значение
maxKnockInterval = knockReadings[i];
}
}

```

```

    }
  }

  //Если появляется новый звук, записываем его
  if (programButtonPressed==true){
  for (i=0;i<maximumKnocks;i++){
  secretCode[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
  }
  delay(50);
  returnfalse; //Если был новый звук - не запускается выход
  }

  if (currentKnockCount != secretKnockCount) //Проверяем, что мы получили правильное количество
  сигналов
  {
  returnfalse;
  }

  /* Теперь мы сравниваем относительные интервалы наших ударов, а не абсолютное время между
  ними.
  (т. е. если вы делаете один и тот же шаблон медленно или быстро, он все равно должен сработать.)
  */
  inttotaltimeDifferences=0;
  inttimeDiff=0;
  for (i=0;i<maximumKnocks;i++)
  {
  knockReadings[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
  timeDiff = abs(knockReadings[i]-secretCode[i]);
  if (timeDiff>rejectValue) //Индивидуальное значение слишком далеко от
  нормы
  {
  return false;
  }
  totaltimeDifferences += timeDiff;
  }

  if (totaltimeDifferences/secretKnockCount>averageRejectValue) //Проверяем. Что неточностей не очень
  много
  {
  returnfalse;
  }
  returntrue;
  }

```