

СОДЕРЖАНИЕ

Введение	8
1 Литературный обзор	10
1.1 Базовое понятие сенсорной сети	10
1.2 Классификация существующих систем умный дом	13
1.3 Анализ аналогичных разработок	15
2 Анализ исходных данных и основных технических требований к разрабатываемому устройству	18
3 Предварительное проектирование системы	22
3.1 Разбиение системы на функциональные модули	22
3.2 Выбор соотношения между аппаратными и программными средствами	25
3.3 Построение структурной схемы аппаратной части системы	25
3.4 Описание структурной схемы	26
4 Аппаратно-программная реализация с узла сенсорной сети системы умный дом на базе микроконтроллера.	27
4.1 Проектирование аппаратных средств	27
4.1.1 Выбор микроконтроллера	27
4.1.2 Выбор радиомодуля	29
4.1.3 Выбор и подключение LCD-дисплея к ATmega328P	30
4.1.4 Выбор и подключение клавиатуры к ATmega328P	31
4.1.5 Подключение двигателя постоянного тока с помощью ATmega328P	34
4.1.6 Подключение динамика к ATmega328P	35
4.1.7 Подключение реле к ATmega328P	36
4.1.8 Подключение фоторезистора к ATmega328P	37
4.1.9 Подключение датчика температуры относительной влажности к ATmega328P	37
4.1.10 Подключение часов реального времени к ATmega328P	38
4.1.11 Подключение диммера к ATmega328P	39

4.1.12	Подключение микрофона к ATmega328P	41
4.1.13	Описание работы системы по принципиальной схеме	42
4.2	Проектирование программного обеспечения.....	43
4.2.14	Разработка схемы алгоритма работы системы и программы	43
4.2.15	Описание работы системы и программы.....	43
4.3	Моделирование работы системы.....	45
4.3.16	Выбор среды разработки	45
4.3.17	Описание работы процесса	46
5	Разработка печатной платы узла сенсорной сети системы умный дом	53
5.1	Выбор и обоснование компоновочной схемы и метода конструирования	53
5.2	Оценка потребляемой мощности и токов, протекающих в отдельных цепях	55
5.3	Выбор применяемого материала печатной платы	60
5.4	Выбор типоразмера печатной платы.....	60
5.5	Выбор класса точности печатной платы.....	65
5.6	Расчёт диаметров монтажных отверстий	67
5.7	Рассчитаем диаметры контактных площадок	68
5.8	Расчет наименьшего номинального значения ширины печатного проводника.....	70
5.9	Расчет наименьшего номинального расстояния между соседними элементами печатного рисунка.....	71
5.10	Трассировка печатной платы	71
5.1	Оценка качества разработанной конструкции	72
6	Технико-экономическое обоснование эффективности разработки и использования узла сенсорной сети системы умный дом	74
6.1	Характеристика нового изделия	74
6.2.1	Расчёт себестоимости и отпускной цены нового изделия	74
6.2.2	Расчёт чистой прибыли.....	79
6.3	Расчёт инвестиций в производство нового изделия.....	79
6.4	Расчёт показателей экономической эффективности проекта.....	80

Заключение	84
Список использованных источников	85
Приложение А	87
Приложение Б.....	99

ВВЕДЕНИЕ

Мы живём в мире, где большинство повседневных задач автоматизированы или максимально упрощены. Сегодня сложно представить мир без автоматизации. В квартире или офисе совершается огромное число действий, которые могли бы выполняться без нашего участия, к примеру такими как управление бытовыми приборами. Огромный вклад в автоматизацию вносят электро-вычислительных машины (ЭВМ), область применения ЭВМ не ограничивается отдельными устройствами, а распространяется вплоть до целого производства, благодаря этому один человек при помощи ЭВМ может управлять целым предприятием. Развитие и упрощение ЭВМ привело к созданию систем Умного дома.

Первой концепцией умного дома является «Дом с кнопками» который был разработан в 1950 году. Система была создана американским инженером Эмилем Матиасом. Он разместил по всему дому кнопки, которые автоматизировали выполнение основных бытовых задач. Система могла включать и выключать некоторые приборы и распечатывать списки покупок. Но для своего времени система выглядела слишком футуристической и практического распространения не получила [1].

Умного дома – это интеллектуальной системы управления, которая обеспечивает автоматическую работу всех инженерных сетей дома с минимальным участием человека.

Основные функций умного дома включают в себя управление следующими системами:

- энергосистемой (управление электроприборами);
- вентиляцией (очистки и увлажнения воздуха);
- обеспечение безопасности (видео наблюдение, охранные системы);
- климат-контроль, отопление (системы кондиционирование, управление теплыми полами и нагревательными приборами);
- дистанционного управление;
- система оповещения (интернет, телефонная и сотовая связь);
- мониторинг поломок (утечек газа или протечек воды);
- освещением (системы управление естественным и искусственным освещением).

Подобные решения позволяют сэкономить человеку время и деньги, значительной степени облегчают жизнь, а также повышают уровень комфорта. Эти технологии помогают не только экономить время, но и позволяют не зависеть от местонахождения. Данные системы предоставляют возможность

удаленно управлять теми или иными элементами системы умного дома. Система видео наблюдения позволит мониторить состояния квартиры или домашних питомцев в ваше отсутствие. Система климат контроля будет самостоятельно реагировать на изменения температуры и влажности. Автоматизация освещения снизит расходы на электроэнергию и уменьшит выгорание мебели в ваше отсутствие.

«Умные дома» покупают не только энтузиасты, которые хотят повысить уровень комфорта жилья, но и крупные компании для автоматизаций помещений и экономии ресурсов. Благодаря универсальным системам «умный дом», обладающих широким выбором устройств. Имеется возможность модернизации, что позволит использовать новые устройства, не меняя всей системы, а заменяя или добавляя лишь те устройства, которые дадут наибольшую эффективность.

Целью данной дипломной работы является разработать сенсорный узел системы умного дома позволяющая обрабатывать показания с датчиков, и управлять устройствами на базе микроконтроллера.

Данная сенсорная сеть будет состоять из трех узлов:

- контролер естественного освещения;
- контролер искусственного освещения;
- консоли.

1 ЛИТЕРАТУРНЫЙ ОБЗОР

1.1 Базовое понятие сенсорной сети

В данном дипломном проекте требуется разработать сенсорный узел системы умный дом на микроконтроллере с использованием радиомодуля.

Беспроводная сенсорная сеть — это распределённая, самоорганизующаяся сеть множества датчиков и исполнительных устройств, объединённых между собой посредством радиоканала. Причем область покрытия подобной сети может составлять от нескольких метров до нескольких километров за счет способности ретрансляции сообщений от одного устройства к другому. Устройства могут измерять такие параметры, как температуры, влажность, звук, уровень света, вибрации, давление, движение объектов или воздуха. Развитие беспроводных сенсорных сетей началось в начале 1980-х годов, изначально было мотивировано военными задачами и предполагало наличие множество распределённых недорогих узлов, взаимодействующих друг с другом, но управляемых автономно, при этом информация направляется в любой узел, где она может использоваться наилучшим образом. Несмотря на это технология не представляла готовый продукт. Если более конкретно, то датчики были довольно объёмными, а возможности практического применения были весьма ограничены. Более того, первые сенсорная сеть не были тесно связаны с беспроводной связью [2].

Достижения в области вычислительной техники и связи привели к уменьшению размеров сенсорных сетей и удешевлению их производства. Благодаря этому расширились возможности практического применения сенсорных сетей, таких как мониторинг окружающей среды, сеть автомобильных датчиков и сети датчиков тела.

Основной частью сенсорных сети является сенсорный узел. Каждый узел сети как правило состоит из четырех частей (см. рисунок 1.1):

- модуля управления энергопотреблением;
- датчика;
- микроконтроллера;
- приемопередатчика или другим устройством беспроводной связи.



Рисунок 1.1 – Аппаратная структура узла сенсорной сети

Основной проблемой сенсорных сетей было то, что информация между соседними узлами, имеет некоторую избыточности, из-за чего передаче данных по отдельности в каждом узле осуществляет чрезмерную перегрузку каналов связи и повышенное потребление энергии всей сенсорной сети, тем самым сокращая срок службы такой сети. Для устранения этой проблемы стали использовать методы группировки данных. Группирование данных — это процесс объединение нескольких копий информации в один экземпляр. Производительность протокола группирования данных тесно связана с топологией сети [3].

Можно выделить следующие топологии беспроводных сенсорных сетей:

1 Точка-точка – связь между двумя узлами сети, имеют максимально низкое энергопотребление, обладает минимальной стоимостью (см. рисунок 1.2).

2 Звезда – связь узлов сети с одним главным узлом, позволяет использовать стратегию множественного доступа, в сети с топологией звезда имеется один координатор сети (см. рисунок 1.3).

3 Одноранговая многоячейковая сеть – не имеет явно выраженного центрального узла и любой сетевой узел может выполнять функцию координатора, однако при таком типе топологий срок службы узлов уменьшается, за счёт увеличенной нагрузки и частоты использования (см. рисунок 1.4).

4 Кластерное дерево – как правило включает в себя несколько подсетей с топологией звезда, что обеспечивает расширенную зону покрытия и масштабируемость сети (см. рисунок 1.5).

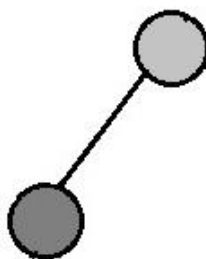


Рисунок 1.2 – Топология точка-точка

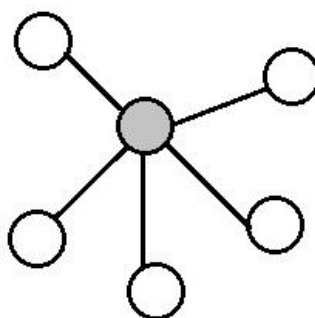


Рисунок 1.3 – Топология звезда

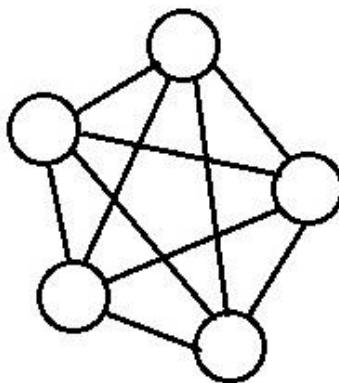


Рисунок 1.4 – Топология одноранговая многочейковая сеть

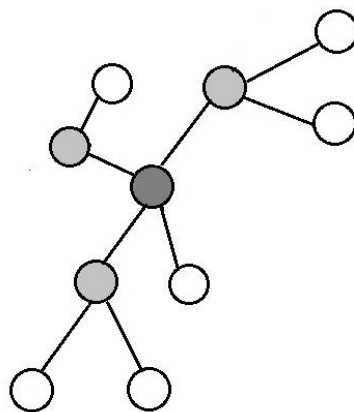


Рисунок 1.5 – Топология кластерное дерево

Проанализировав топологий беспроводных сенсорных сетей, можно сказать что наилучшим вариантом для созданий системы умный дом будет использования топологий типа звезды, кластерного дерева. Данные типы топологии: имеет высокую производительность, увеличенный срок службы, выход из строя одной рабочей станции не повлияет на работу всей сети, проста в настройке, в сеть легко масштабируемая.

1.2 Классификация существующих систем умный дом

В этом разделе будут рассмотрены существующие классификаций систем “умный дом, выявим достоинства и недостатки.

По технологии объединения и управления системами «умного дома» разделяются на:

1 LanDriver универсальная платформа построения шинных систем управления, используемая в автоматизации зданий. Предназначена для управления внутренними и внешними системами. Система LanDriver состоит из центрального контроллера и модулей, подключенных между собой шиной (стандарт RS-485). К модулям подключается управляемое оборудование [4].

2 EIB/KNX система EIB распределенная, управление осуществляется в пределах устройств. Устройства обмениваются информацией по шине EIB в соответствии с собственным протоколом. Система, построенная на EIB, автономна и не зависит от работоспособности центрального контроллера [4].

3 AMX разрабатывает программно аппаратные средства удаленного управления, медиа системой, системой видеонаблюдения и широкого спектра датчиков. Протоколы передачи данных закрыты. Изначально применялась собственная шина передачи данных, в новую линейку оборудования

применяются стандартные протоколы Ethernet, Wi-Fi, а также имеет шлюзы сопряжений с системами EIB, LON и др [4].

4 Z-wave, технология беспроводной передачи данных, разработанная для домашней автоматизации. В технологии Z-wave применяются маломощные и миниатюрные радио модули, встраиваемые в бытовую технику. В основе технологии лежит ячеистая технология, в которой каждый узел является приемником и передатчиком, т.е. при возникновении препятствия сигнал пойдет через соседние узлы сети, находящиеся в радиусе действия. Еще одним преимуществом является малое энергопотребление, что вместе с малыми размерами, позволяет встраивать Z-wave в различные бытовые прибор [4].

Каждую систему умный дом можно разделить на две архитектуры управления централизованная и распределенная. Централизованная система состоит из центрального контроллера с подключенными к нему модулями. К централизованным системам относятся AMX. В распределенных системах управления устройства не зависят друг от друга. К распределенным системам относятся EIB/KNX.

Достоинства централизованных систем:

1 Позволяет строить сложные системы управления. Центральный контролер обладает достаточной производительностью и несет актуальную информацию о подключенных к нему модулей.

2 Скорость обработки информации выше у централизованных систем, так как сбором информации с модулей, центральный контроллер занимается единолично минуя модульную обработку.

3 Модули (датчики, исполняемые устройства) компактны, дешевы и имеют простую техническую реализацию.

Недостатки централизованных систем:

1 Основным недостатком является ненадежность таких систем. При выходе из строя центрального контроллера, вся система перестает функционировать.

2 Высокая цена центрального модуля.

Достоинства распределенных систем:

1 Из-за отсутствия центрального контроллера, выход из строя одного или нескольких модулей, существенно не повлияет на работу системы в целом. Распределенная система отличается повышенной надежностью.

2 Распределенные системы просты в расширении. На имеющуюся шину добавляются новые модули, поддерживающие протокол передачи данных данной шины.

Недостатки распределенных систем:

1 Модули (датчики, исполняемые устройства), имеют собственные контроллеры обработки данных из-за чего становятся технически сложными, громоздкими. Стоимость таких модулей выше, чем в централизованных системах.

2 Скорость работы таких систем ниже из-за обработки данных в разных модулях.

1.3 Анализ аналогичных разработок

В этом разделе будут рассмотрены существующие аналоги, которые выполняют такие же или схожие задачи. В проанализированных системах выявим достоинства и недостатки.

Отечественная компания “z-wave” является разработчиком и производителем устройства мониторинга и контроля дома (см. рисунок 1.6). Основная сфера применения – удаленный контроль и мониторинг устройств в доме и коттедже [5].

Задачи, решаемые при помощи устройств z-wave:

- контроль освещение;
- климат-контроль;
- управление бытовыми приборами.



Рисунок 1.6 – Общая структура взаимодействия устройств z-wave

Z-Wave устройства — это не только контроллеры, выключатели, датчики. Каждое Z-Wave устройство является ретранслятором, благодаря чему существенно повышая надёжность сети и увеличивая её радиус. Благодаря «ячеистости» размер сети в целом может достигать 150 метров в диаметре. Основным преимуществом является персонализированная разработка и наличие большого количества устройств на выбор, а недостаток — большая стоимость.

Умный дом от «Белтелеком» представляет собой услугу системы мониторинга отдельных объектов в помещениях, рамках услуги абоненту предоставляется [6]:

1 Базовый комплект включает в себя абонентское устройство и три устройства датчик задымленности, датчик движения и датчик открытия дверей/окон (см. рисунок 1.7).

2 Дополнительное оборудование видеокамеру, сирену, умную розетку, датчики температуры и влажности, датчики протечки воды.

3 Программный продукт, который абонент самостоятельно скачивает из магазина приложений и устанавливает на свое мобильные.

Основным преимуществом является возможность брать систему в пользование в рамках абонентской платы, а недостаток — ограниченный набор блоков и функционал системы.



Рисунок 1.7 – Базовый комплект умного дома от Белтелекома

Система от китайской фирмы “Xiaomi” состоит из необходимого минимума независимых устройств, работающих по ZigBee-протоколу. Благодаря большому ассортименту недорогих устройств пользователь может собрать персонализированную систему под собственные нужды (см. рисунок 1.7). Такие как настроить комфортную температуру в комнате, нужное именно в данный момент освещение, а также включить дежурное "освещение" ночью, или запрограммировать несколько возможных сценариев для поддержания заданной температуры и влажности в помещении.

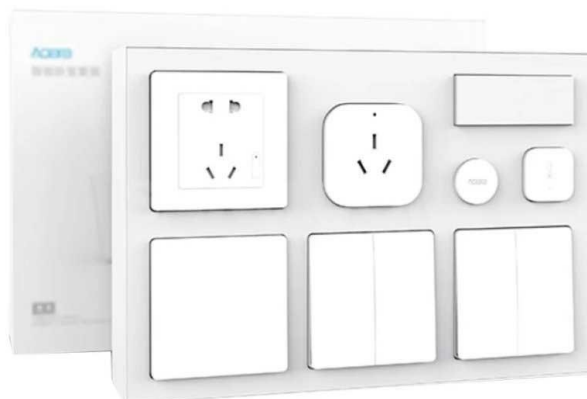


Рисунок 1.7 – Базовый комплект умного дома от Xiaomi

Проанализировав рынок и доступные модели можно сказать, что разрабатываемое устройства не уступает аналог, а также имеет преимущество в стоимости.

2 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ОСНОВНЫХ ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ К РАЗРАБАТЫВАЕМОМУ УСТРОЙСТВУ

В данном дипломном проекте требуется реализовать сенсорный узел системы умный дом на микроконтроллере.

Умного дома – системы управления, которая обеспечивает автоматическую работу систем по средствам обработки данных и отправки управляющего сигнала на устройства. Получение данных происходит по средствам сигналов от датчиков температуры и влажности, датчиков освещённости, датчика холла, клавиатуры, а также приема данных с радиомодуля. В качестве исполнительных механизмов используются двигатель переменного тока, динамик, реле и диммер.

Сенсорный узел системы умный дом должен обеспечивать управление яркостью естественного и искусственного освещения, то есть должен управлять подачей напряжения на светильник, обеспечивающего открытия и закрытие штор, а иметь возможность обмениваться данными с управляющим устройством.

Взаимодействие пользователя с устройством производится по средствам тактовых кнопок, расположенных на лицевой панели корпуса. Вывод информации будет производиться на LCD дисплей.

Эффективность системы определяется надежностью и простотой использования, небольшими габаритами и малым электропотреблением.

Функциональная спецификация определяет, какие функции должны выполняться системой и как выглядит интерфейс между системой и окружением. Таким образом, функциональная спецификация включает в себя два основных компонента:

- список функций, выполняемых системой;
- описание интерфейса между системой и пользователем.

По отношению к пользователю системы умного дома, функциональная спецификация должна дать ответы на следующие вопросы:

- 1 Что необходимо предусмотреть для включения дисплея?
- 2 Что необходимо предусмотреть для ввода информации от пользователя в микроконтроллер?
- 3 Что необходимо предусмотреть для вывода введенной пользователем информации?
- 4 Что необходимо предусмотреть для настройки режимов работы?

5 Что необходимо предусмотреть для определения приоритета управления?

6 Что необходимо предусмотреть для оповещения пользователя об неисправности системы?

7 Что необходимо предусмотреть для ручного управления системой в случае невозможности управлять с консоли?

Ответив на все эти вопросы, можно перейти к составлению функциональной спецификации устройства

1 Функцию включения дисплея будет выполнять специальная кнопка, расположенная на клавиатуре.

2 Для представления пользователю возможности ввода данных, используется клавиатура на тактовых кнопках. На клавиатуре располагаются кнопки выбора режима, кнопки изменения значения и подтверждения.

3 Для вывода введенной пользователем информации используются LCD дисплей.

4 Для настройки режима работы, необходимо выбрать режим работы, находящийся на верхнем ряду клавиатуры и нажать клавишу “Ok” после чего используя клавиши “<” и “>” задать нужное значение, по окончании нажать повторно клавишу “Ok” после чего дублирующая информация введенных значений выведется на дисплей.

5 Оповещение пользователя об неисправности будет использоваться устройство звуковой сигнализации и дисплей. Дисплей будет выводить сообщение об ошибке, а звуковая сигнализация будет привлекать внимания, выключение будет производиться по средствам нажатие кнопки “Ok”, что позволит удостовериться в том, что пользователь информирован об неисправности системы.

6 Для определения приоритета ручного управления. На устройстве управление светом будет добавлена клавиша manual отключающая прием данных с консоли.

7 В случае невозможности использовать консоль для управления, каждое устройство сенсорной сети имеет возможность работать независимо от консоли посредством дополнительных клавиш управление на лицевой панели.

Учёт человеческих факторов при проектировании должен приводить к простоте системы и легкости ее использования. Эти цели достигаются посредством проектирования надлежащего системного интерфейса.

На панели модуля управления искусственным освещением изображены: клавиша manual, предназначенная для отключения дистанционного режима управления, клавиша Light отвечающая за включение и выключение света, правее от нее регулятор, отвечающий за ручное управление яркостью света.

Внешний вид панели управления искусственным освещением представлен на рисунке 2.1.

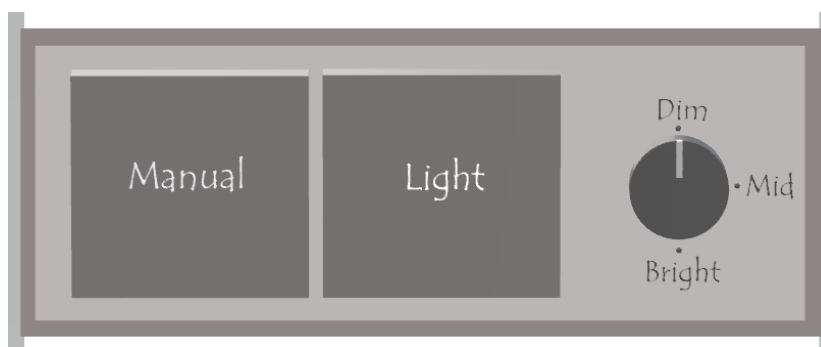


Рисунок 2.1 – Лицевая панель модуля управления искусственным освещением

На панели модуля управления естественным освещением изображены клавиша Up, отвечающая за ручное включение подъёма роллета и клавиша Down, отвечающая за ручное включение реверса.

Внешний вид панели управления естественным освещением представлен на рисунке 2.2.

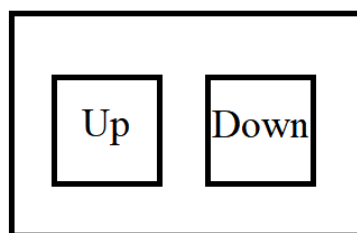


Рисунок 2.2 – Лицевая панель модуля управления естественным освещением

На панели управляющей консоли изображены: отверстия для вывода звука из динамика, служащий для звукового предупреждения, lcd дисплей, служащий для вывода введенной пользователем информации, под ним клавиатура на тактовых кнопках и регуляторы, предназначенная для ввода пользователем информации.

Верхний ряд кнопок отвечает за шесть основных режимов. Каждый режим имеет определенный функционал:

1 SET является пользовательским режимом с занесёнными предпочтениями, при его нажатии система автоматически приведет все значение к нужным.

2 ECO это энергосберегающий режим предназначенный для максимальной экономии энергий путем максимизации использования естественного освещения, а также уменьшение частоты опроса и передачи данных.

3 REAL режим реального времени предназначен для отображение текущего состояния системы и изменений в реальном времени.

4 TIME режим, предназначенный для защиты интерьера от выгорания и нежелательного нагрева, а также для увеличения комфортного уровня сна путём избавления от лишнего света ночью. Пользователь задает время закрытия и желаемый уровень затемнения, а система автоматически при наступление нужного интервала времени опускает или поднимает роллет.

5 INFO режим мониторинга значений с датчиков температуры и влажности, а также в этом режиме происходит отображение значения текущего времени и отправка информации на телефон.

6 OF режим отключение подсветки дисплея, функция необходима для избавления от нежелательного свечения дисплея ночью.

Внешний вид управляющей консоли представлен на рисунке 2.3.

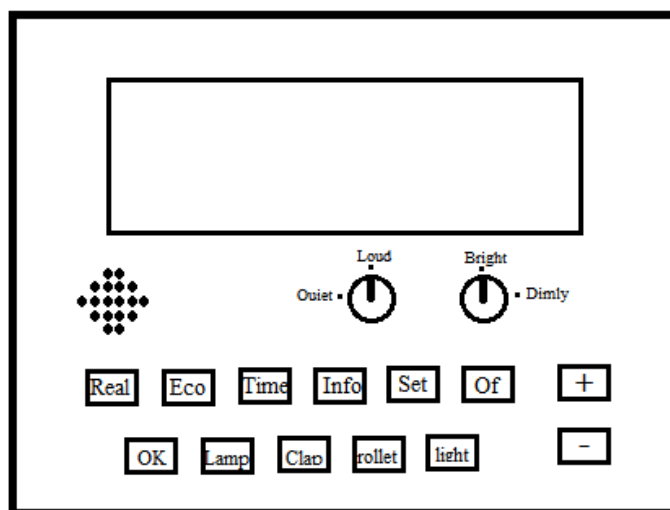


Рисунок 2.3 – Лицевая панель управляющей консоли

Пример развёртки узла сенсорной сети умного дома представлен на чертеже ГУИР484434.001 ПЛ.

3 ПРЕДВАРИТЕЛЬНОЕ ПРОЕКТИРОВАНИЕ СИСТЕМЫ

3.1 Разбиение системы на функциональные модули

На основе функциональной спецификации определим набор модулей реализации функций в системе. Следующим шагом будет разбиение модулей на аппаратные и программные.

В результате получим модульную структуру аппаратных средств устройства.

Основу аппаратных средств системы составляет управляющая микро-ЭВМ, которая включает в себя:

- 1 Процессорный модуль, предназначенный для обработки информации.
- 2 Модель генератора тактовых импульсов, предназначенный для синхронизации работы системы.
- 3 Модуль памяти, предназначенный для хранения информации.
- 4 Модуль интерфейса ввода и модуль интерфейса вывода, содержащие интерфейсные компоненты, необходимые для связи процессорного модуля с другими модулями системы.
- 5 Модуль таймера.
- 6 Модуль преобразования входного сигнала.
- 7 Модуль преобразования входного сигнала, которые содержат компоненты, необходимые для обмена входными и выходными сигналами с внешним окружением.

На рисунке 3.1 представлена структура аппаратных средств сенсорного узла системы умный дом.

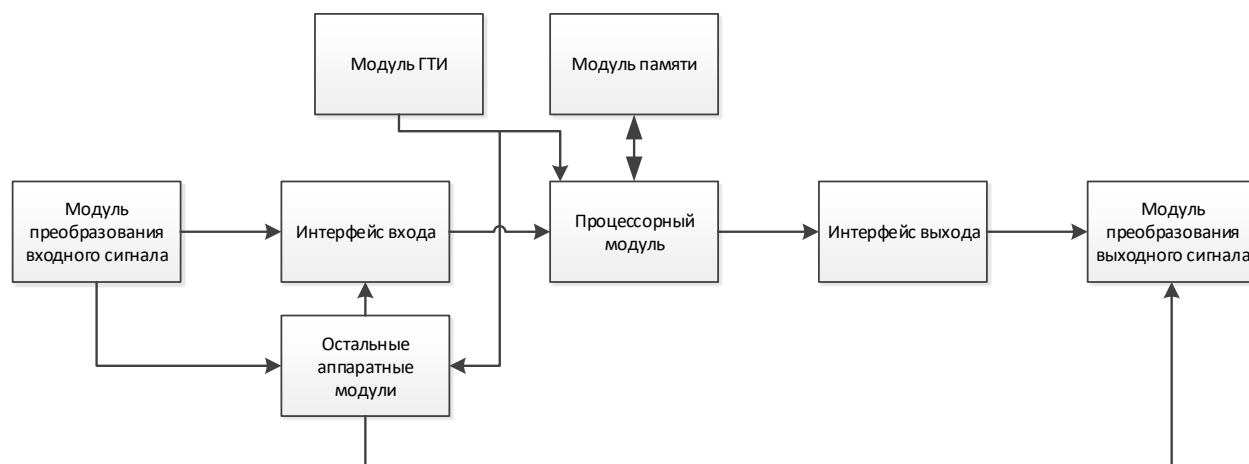


Рисунок 3.1 – Модульная структура аппаратных средств микропроцессорной системы

Разобьем систему на функциональные модули. Система состоит из 3 основных частей: вход, выход и обработка информации.

Входной модуль является клавиатурой состоящая из тактовых кнопок, регуляторы, датчики света, датчик температуры и влажности, датчик холла, микрофон, модуль реального времени, а также радиомодуль для приема информации.

Выходной модуль состоит из двигателя постоянного тока, радиомодуль, светильника, светодиодов, динамика и lcd дисплея.

Модуль обработки информации состоит из подмодулей:

- модуль проверки состояния датчика холла;
- модуль считывания информации с клавиатуры;
- модуль считывания информации с датчика света;
- модуль считывания информации с датчика температуры и влажности;
- модуль проверки состояния реле;
- модуль детектора нуля;
- модуль вывода информации на дисплей;
- модуль смены настроенных значений;
- модуль обработки данных клавиатуры;
- модуль проверки неисправности.

Функционально-модульная структура, разбитая на слои по иерархии представлена на рисунке 3.2. Верхний уровень представлен исполнительным модулем, который имеет средства, необходимые для реализации управляющей функции. На среднем уровне располагаются модули: управление светом, шторами, смены значения. Дальше расположились модули считывания, проверки, ожидания. А на нижнем находятся входной и выходной модуль.

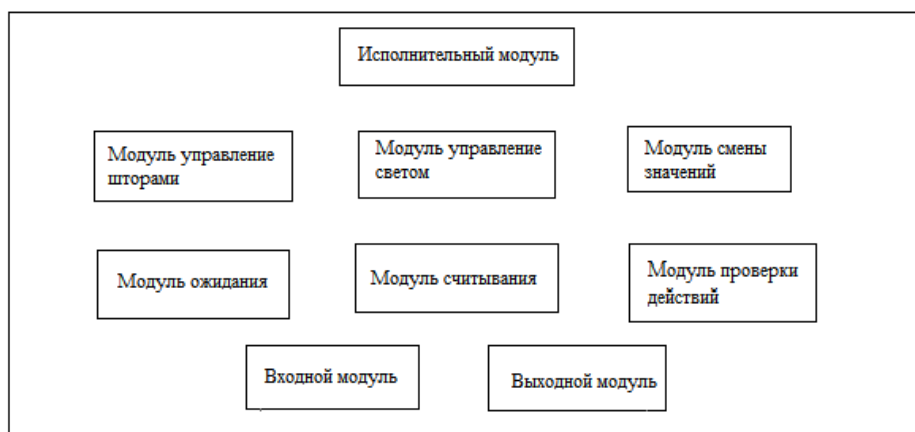


Рисунок 3.2 – Функционально-модульная структура сенсорного узла системы умный дом

Распределение функций сенсорного узла систем умный дом по модулям выглядит следующим образом:

а) Исполнительный модуль:

1) Управление системой.

б) Входной модуль:

1) Прием данных с радиомодуля;

2) Считывания состояния датчика холла;

3) Считывания информации с микрофона;

4) Считывания информации с датчика света;

5) Считывания информации с датчика температуры и влажности;

6) Проверка состояния реле;

7) Считывание данных с модуля реального времени.

в) Выходной модуль:

1) Вывод информации дисплея;

2) Воспроизведение звукового предупреждения;

3) Регулировка освещения;

г) Модуль ожидания выполнения действия:

1) Ожидание нажатия кнопки;

2) Ожидание сигнала активности с холла.

д) Модуль считывания:

1) Считывание значений с клавиатуры.

2) Считывание значений с регулятора.

е) Модуль проверки действия:

1) Проверка на нажатие клавиши с клавиатуры;

2) Проверка на нажатие клавиши «manual»;

3) Проверка на приём назначений с датчиков;

4) Проверка на выбранный режим.

ж) Модуль смены значений:

1) Смена настроенных значений.

з) Модуль управления штор:

1) Включение/выключение моторчика.

и) Модуль управлением света:

1) Детектор нуля;

2) Контроль скважности.

3.2 Выбор соотношения между аппаратными и программными средствами

Функция управления системы реализуется управляющей микро-ЭВМ (микроконтроллер) в результате выполнения основной (управляющей) программы путем последовательного вызова функций соответствующих программных модулей системы.

Связь между программными и аппаратными модулями представлена на рисунке 3.3.

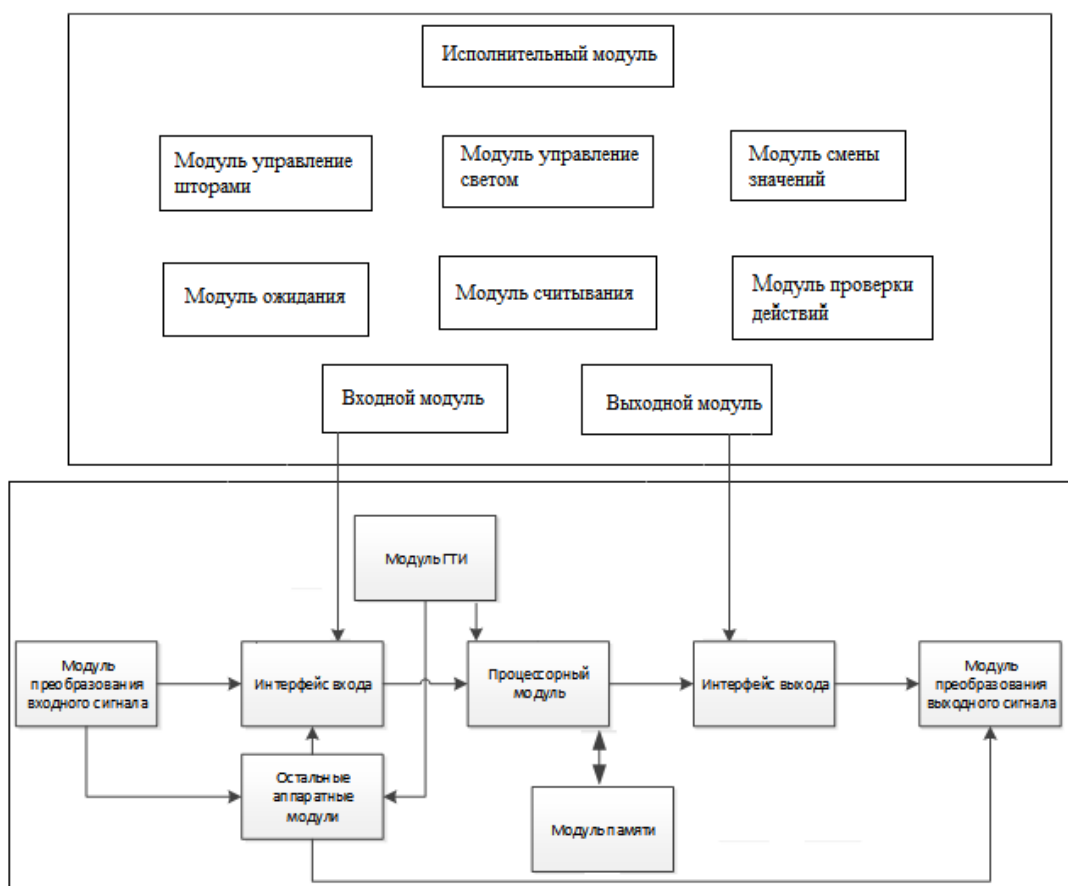


Рисунок 3.3 – Связь между аппаратными и программными средствами

3.3 Построение структурной схемы аппаратной части системы

На основе выполняемых системой функций, построим структурную схему аппаратной части.

Электрическая структурная схема проектируемой системы представлена на чертеже ГУИР484434.001 Э1.

3.4 Описание структурной схемы

Центральным модулем структурной схемы является процессорный модуль (микроконтроллер). Он выполняет функции управления процессом обмена данными с периферийными устройствами и обработку поступающей в него информации. В модуле памяти хранятся коды, константы и переменные программ процессорного модуля.

Входной модуль представлены портами микропроцессорного устройства, включает в себя в себя периферийные устройства, которые позволяют нам считывать информацию, поступающую в систему от пользователя. К входным периферийных устройствам относятся клавиатура на тактовых кнопках, регуляторы. Считанные с кнопок и регулятора данные поступают в микроконтроллер, где впоследствии проходят обработку и запускают необходимые процессы. К входному модулю также относится: датчик температур, датчик света, датчик холла, микрофон, детектор нуля, модуль реального времени данные с которого считываются и обрабатываются в микроконтроллере.

Модуль ГТИ обеспечивает синхронную работу нашего устройства.

Выходной модуль связывает микроконтроллер с выходными внешними устройствами: lcd дисплеем, светодиодом, динамиком, двигатель постоянного тока и светильником.

В отдельный блок можно выделить радиомодуль. Необходимый для обмена данными между устройствами.

4 АППАРАТНО-ПРОГРАММНАЯ РЕАЛИЗАЦИЯ С УЗЛА СЕНСОРНОЙ СЕТИ СИСТЕМЫ УМНЫЙ ДОМ НА БАЗЕ МИКРОКОНТРОЛЛЕРА.

4.1 Проектирование аппаратных средств

4.1.1 Выбор микроконтроллера

На этапе проектирования системы было принято решение использовать микроконтроллер ATmega328P. На сегодняшний день он является одним из самых популярных микроконтроллеров от компании Atmel Corporation. С ростом популярности ATmega все больше и больше производитель начали производить аппаратно-программных средства такие как платы Arduino.

ATmega328P — это высокопроизводительный 8-битный AVR-микроконтроллер с низким энергопотреблением, основанным на усовершенствованной RISC архитектуре.

Atmel Corporation — компания производитель полупроводниковых электронных компонентов, основана в 1984 году. Один из лидеров производства микроконтроллеров (MCS-51, ARM, AVR, AVR32). Также разрабатывает и производит небольшие модули энергонезависимой памяти для электронных изделий, ПЛИС, цифровые микросхемы-радиоприёмники и передатчики. Компания для своих клиентов может предложить систему на кристалле, объединяющую затребованные компоненты. Продукция Atmel широко применяется в компьютерных сетях, промышленности, медицине, связи, автомобилях, космосе, военных устройствах, а также кредитных картах [7].

Идея разработки нового RISC-ядра принадлежит двум студентам студентами Норвежского технологического института. В 1995 году Боген и Воллен решили предложить американской корпорации Atmel, которая была известна своими чипами с Flash-памятью, выпускать новый 8-битный RISC-микроконтроллер и снабдить его Flash-памятью для программ на одном кристалле с вычислительным ядром.

Идея была одобрена Atmel Corp., и было принято решение незамедлительно инвестировать в данную разработку. В конце 1996 года был выпущен опытный микроконтроллер AT90S1200, а во второй половине 1997 г. корпорация Atmel приступила к серийному производству нового семейства микроконтроллеров.

Микроконтроллеры AVR имеют гарвардскую архитектуру (программа и данные находятся в разных адресных пространствах) и систему команд, близкую к идеологии RISC. Процессор AVR имеет 32 8-битных регистра общего назначения, объединённых в регистровый файл [8].

ATmega328P получило широкое применение в отрасли создания электронных средств. Основными достоинствами микроконтроллер являются следующие показатели:

- низкая стоимость;
- простая и понятная среда программирования.

Согласно документации, микроконтроллер имеет следующие характеристики:

- 14 цифровых входов/выходов (6 из которых могут использоваться как выходы ШИМ);
- 6 аналоговых входов;
- Два 8-битных таймера/счетчика с модулем сравнения и делителями частоты;
- 16-битный таймер/счетчик с модулем сравнения и делителем частоты, а также с режимом записи;
- Счетчик реального времени с отдельным генератором;
- Программируемый последовательный порт USART;
- Последовательный интерфейс SPI;
- Интерфейс I2C;
- Внутренняя схема сравнения напряжений;
- Блок обработки прерываний и пробуждения при изменении напряжений на выводах микроконтроллера;
- флеш-память объемов 32 КБ;
- ОЗУ объемом 2 КБ;
- ПЗУ 1 КБ.

Расположение выводов микроконтроллера ATmega328P представлено на рисунке 4.1

(PCINT14/RESET)	PC6	Pin1	1	28	Pin28 PCS (ADCS/SCL/PCINT13)
(PCINT16/RXD)	PD0	Pin2	2	27	Pin27 PD4 (ADC4/SDA/PCINT12)
(PCINT17/TXD)	PD1	Pin3	3	26	Pin26 PD3 (ADC3/PCINT11)
(PCINT18/INT0)	PD2	Pin4	4	25	Pin25 PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1)	PD3	Pin5	5	24	Pin24 PC1 (ADC1/PCINTS)
	PD4	Pin6	6	23	Pin23 PC0 (ADCO/PCINT8)
	Vcc	Pin7	7	22	Pin22 GND
	GND	Pin8	8	21	Pin21 AREF
(PCINT6/XTAL1/TOSC1)	PB6	Pin9	9	20	Pin20 AVCC
(PCINT7/XTAL2/TOSC2)	PB7	Pin10	10	19	Pin19 PBS (SCK/PCINTS)
(PCINT21/OC0B/T1)	PD5	Pin11	11	18	Pin18 PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0)	PD6	Pin12	12	17	Pin17 PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1)	PD7	Pin13	13	16	Pin16 PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1)	PB0	Pin14	14	15	Pin15 PB1 (OC1A/PCINT1)

Рисунок 4.1 – Расположение выводов микроконтроллера ATmega328P

4.1.2 Выбор радиомодуля

В данном проекте применяется радиомодуль NRF24L01 (см. рисунок 4.2). NRF24L01 – это высоко интегрированная микросхема с пониженным потреблением энергии. При помощи модуля можно связать несколько устройств для передачи связи с подтверждением доставки пакетов данных. Можно объединить до семи приборов в одну общую радиосеть на частоте 2,4 ГГц. Так же присутствует возможность одновременного приема и передачи данных, что позволяет использовать его как ретранслятор значительно увеличивая радиус передачи данных. Радиомодуль имеет возможность беспрепятственно подключается к микроконтроллерам с интерфейсом SPI [9].

Невысокая цена, малое энергопотребление, существенный радиус действия и возможность связи с телефоном при помощи беспроводного стандарта BLE являются основные преимущества модели NRF24L01.



Рисунок 4.2 – Внешний вид радиомодуля NRF24L01

4.1.3 Выбор и подключение LCD-дисплея к ATmega328P

За вывод информации в проектируемой схеме будет отвечать LCD-1602 дисплей. Основным критерием выбора стала низкая стоимость по сравнению с другими типами дисплеев. LCD-1602 может отображать 2 строки по 16 символов.

Для подключения LCD дисплея будет применяться расширитель портов. Для уменьшения количества занятых портов микроконтроллера используется расширитель портов PCF8574. С его помощью можно увеличить количество портов входа-выхода микроконтроллера на 8 штук. Расширитель портов имеет три порта A, при изменении значений на которых от нуля до единицы, изменяется адрес для микроконтроллера. Благодаря этому можно одновременно подключить до 8 расширителей портов.

Подключение к микроконтроллеру происходит по интерфейсу I2C.

I2C – это протокол последовательной связи данные передаются по одному проводу (линия SDA). I2C является синхронным, поэтому вывод битов синхронизируется с выборкой битов тактовым сигналом, разделяемым между ведущим и ведомым устройством. Тактовый сигнал всегда контролируется мастером [10].

SDA (Serial Data) – линия для передачи и приема данных.

SCL (Serial Clock) – линия, несущая тактовый сигнал.

С помощью I2C данные передаются в сообщениях. Сообщения разбиты на фреймы данных. Каждое сообщение имеет кадр адреса, содержащий двоичный адрес ведомого устройства, и один или несколько кадров данных, содержащих передаваемые данные. Сообщение также включает в себя условия запуска и остановки, биты чтения / записи и биты ACK/NACK между каждым кадром данных (см. рисунок 4.3).

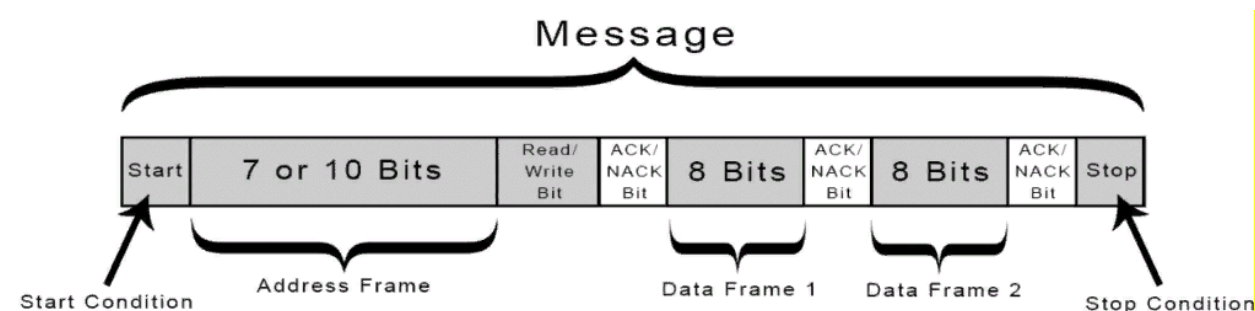


Рисунок 4.3 – Структура сообщения I2C

Для запуска линия SDA переключается с высокого уровня напряжения на низкий уровень напряжения, прежде чем линия SCL переключается с высокого на низкий.

Для остановки линия SDA переключается с низкого уровня напряжения на высокий уровень напряжения после того, как линия SCL переключается с низкого на высокий.

Для подключения расширителя к LCD дисплею необходима вход выбора направления передачи данных присоединить к пину P1, а вход для выбора инструкций к P0. Для ввода инструкции в дисплей необходимо тактирование, которое подключается к P2.

Схема подключения LCD дисплея к ATmega328P представлена на рисунке 4.4.

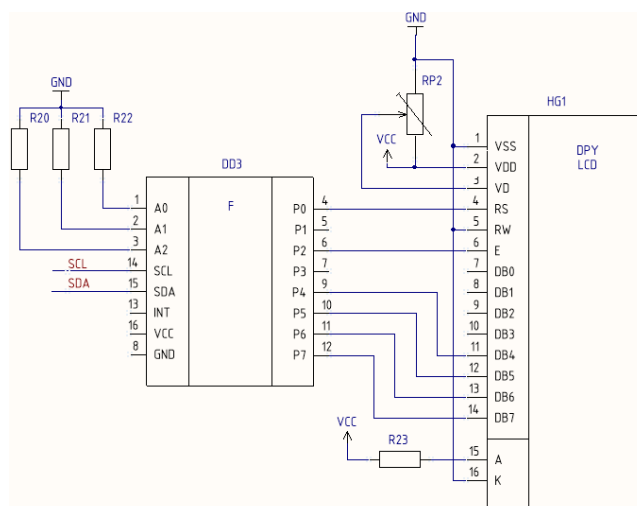


Рисунок 4.4 – Схема подключения LCD дисплея через расширитель портов

В качестве LCD дисплея для нашего устройства была выбран YJD1602A-1, а в качестве расширителя портов PCF8574.

4.1.4 Выбор и подключение клавиатуры к ATmega328P

Активация системы, выбор режима работы, а также настройка режимов осуществляется с помощью тактовых кнопок. Кнопка соединяется с микроконтроллером через подтягивающий резистор номиналом 10 кОм. Таким образом, при нажатии на кнопку, на соответствующем порте микроконтроллера появляется логическая единица, при отпускании кнопки – логический ноль.

Схема подключения тактовой кнопки к ATmega328P представлена на рисунке 4.5.

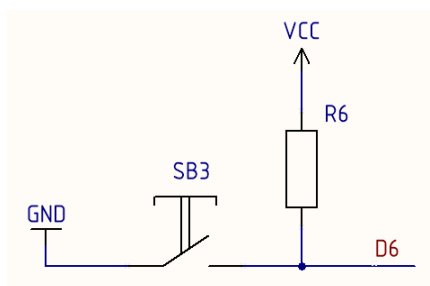


Рисунок 4.5 – Схема подключения кнопки к ATmega328P

При таком типе подключения для управление большое количество кнопок необходимо такое же количество портов микроконтроллера для уменьшения задействованных портов, было принято решение использовать клавиатуру. Из-за нестандартного количества клавиш и их расположение решено не использовать готовый вариант матричной клавиатуры, а создать собственную клавиатуру на тактовых кнопках. При создания собственной клавиатуры на тактовых кнопка, было решено подключать несколько кнопок на одни аналоговый пин микросхемы. Принцип основан на чтении и интерпретации аналого-цифровым преобразователем микроконтроллера индивидуального напряжения на каждой кнопке. Индивидуальное напряжение на каждой кнопке определяется путем классического делителя напряжения и вычисляется по формуле:

$$U_{R2} = U * \frac{R2}{R2 + R1}, \quad (4.1)$$

Принцип делителя напряжение представлен на рисунке 4.6.

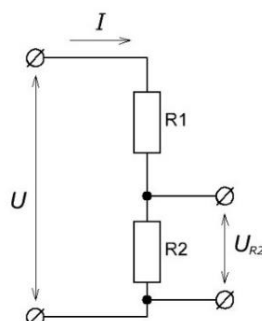


Рисунок 4.6 – Схема делителя напряжения

Номиналы резисторов для делителя напряжения подобраны таким образом, чтобы обеспечить равные интервалы при определении значений

напряжения АЦП. Такой подход очень удобен для количества кнопок 2^n . Удобство заключается в том, что результат работы АЦП при помощи битового сдвига можно округлить, при этом максимально уменьшить погрешности АЦП. Сдвигаем на 7 бит, потому что из 10-битного значения нужно оставить всего 3 бита, в которых закодированы значения от 0 до 7. К ним прибавляется единица, чтобы отделить полученные состояния от состояния спокойствия 0. Зная необходимое количество кнопок равно 8, подберем оптимальное падение напряжение на каждой кнопке равное 0,65 В. Поскольку первая кнопка замыкается без резистора, то резистор нулевого сопротивления опускаем, отсюда кнопка даст результат равный 5 В. Рассчитаем номиналы для приведенных восьми кнопок и семи резисторов по формуле:

$$R_i = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * (n - i)} - 10\text{Ком} - \sum R_i \quad (4.2)$$

Подставляя значение в формулу (4.2), получим:

$$R_1 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 7} - 10\text{Ком} = 1\text{Ком}$$

$$R_2 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 6} - 11\text{Ком} = 1,84\text{Ком}$$

$$R_3 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 5} - 12,84\text{Ком} = 2,54\text{Ком}$$

$$R_4 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 4} - 15,38\text{Ком} = 3,87\text{Ком}$$

$$R_5 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 3} - 19,25\text{Ком} = 6,41\text{Ком}$$

$$R_6 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 2} - 25,66\text{Ком} = 12,82\text{Ком}$$

$$R_7 = \frac{5\text{В} * 10\text{Ком}}{0,65\text{В} * 1} - 38,48\text{Ком} = 38,46\text{Ком}$$

Схема подключения клавиатуры на тактовых кнопках к ATmega328P представлена на рисунке 4.7.

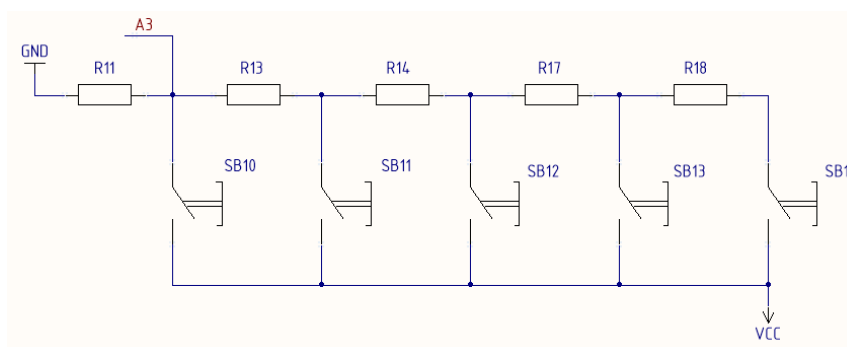


Рисунок 4.7 – Схема подключения клавиатуры на тактовых кнопках к ATmega328P

В качестве тактовой кнопки для нашего устройства были выбраны KLS7-TS1202.

4.1.5 Подключение двигателя постоянного тока с помощью ATmega328P

В качестве устройства для открытия и закрытия штор используется двигатель постоянного тока. Для его подключения необходима два цифровых пина микроконтроллера. Один цифровой пин отвечает за прямой ход, а другой за реверс. Из-за мелового тока с портов микроконтроллер необходимо подключить двигатель постоянного тока к проводу питания 5В через транзистор в ключевом режиме. Для обеспечения возможности реверса, схема была доработана до Н-моста [13].

Для сборки Н-моста понадобятся два транзистора pnp типа (VT3, VT5), способные выдержать напряжение 5В и ток 0,5А. И два транзистора их комплиментарной пары npn типа (VT2, VT4).

Зная коэффициент усиление транзистора, равный 40 можно найти управляющий ток.

$$I = \frac{0,5A}{40} = 12mA$$

Управляющий ток для транзисторов VT3-VT5 равен 12мА.

Рассчитываем резистор R6, полагая, что на p-n переходах БЭ транзисторов VT1, VT2, VT5 падает по 0,6 В. Зная падение напряжение на транзисторе можно рассчитать сопротивление.

$$R_{6,10} = \frac{5 - 3 * 0,6}{12} = 260 \text{ Ом}$$

Сопротивление R6 и R10 равно 260 Ом.

Так же в схеме подключения кроме транзисторов и резисторов необходимо используются шунтирующие диоды. Это сделано по причине того, что ток в двигателе не может изменяться скачками.

Схема подключение двигателя представлена на рисунке 4.8.

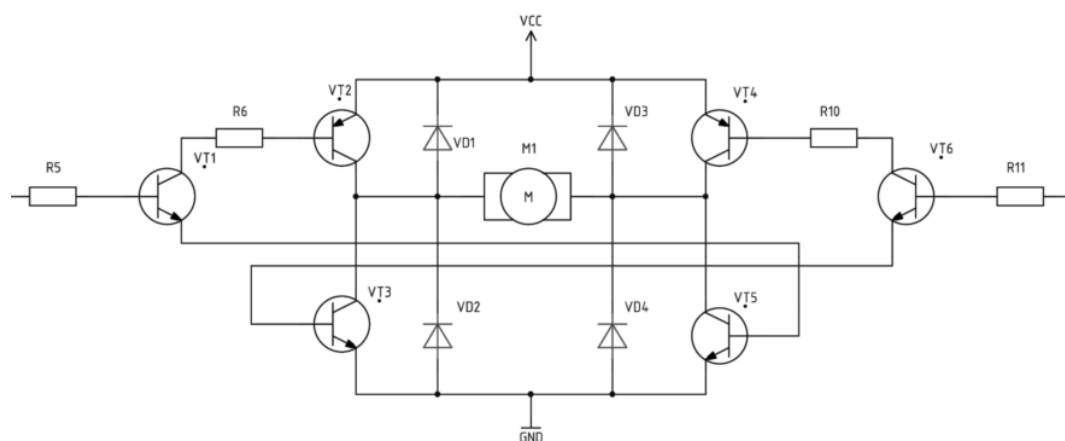


Рисунок 4.8 – Схема подключения двигателя постоянного тока к ATmega328P

Для нашего устройства будет использоваться двигатель постоянного тока F130-16155.

4.1.6 Подключение динамика к ATmega328P

Динамик применяются для звуковой сигнализации о неисправности системы. Для данного устройства подойдет стандартный 8 Ом динамик. Динамик подключается к цифровому пину микроконтроллера. Для получение хорошей громкости был использован транзистор в ключевом режиме. Резистор перед транзистором необходим для подачи корректного тока управления, а также совместно с конденсатором образует RC-фильтр низких частот. Подстроечный резистор используется в качестве регулятора громкости.

Схема подключение динамика представлена на рисунке 4.9.

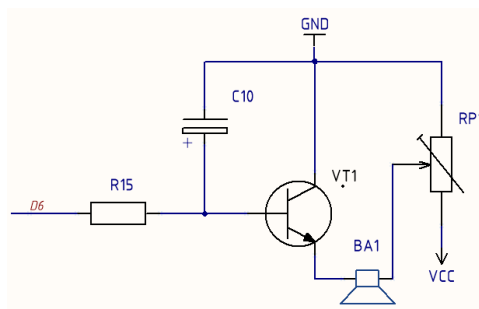


Рисунок 4.9 – Схема подключения динамика к ATmega328P

Для нашего устройства будет использоваться динамик KSSD1708.

4.1.7 Подключение реле к ATmega328P

В данном проекте в качестве дистанционного выключателя света используется электромагнитное реле.

Электромагнитное реле – это электрическое устройство, которое механическим путем замыкает или размыкает цепь нагрузки при помощи магнита. состоит из электромагнита, подвижного якоря и переключателя. Реле имеет всего три контакта, подключаются к микроконтроллеру следующим образом: GND – GND, VCC – +5V, In – к цифровому пину. Для срабатывания реле необходимо определённое напряжение, по этому реле подключается через транзистор в ключевом режиме. Для предотвращения электрических всплесков напряжения на катушке реле при выключении используется параллельный диод.

Схема подключение реле представлена на рисунке 4.10.

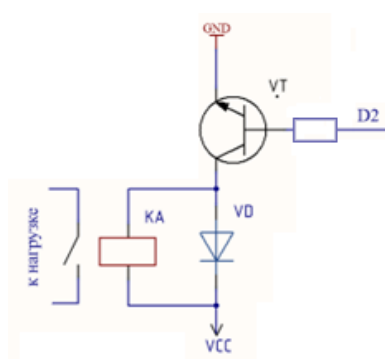


Рисунок 4.10 – Схема подключения реле к ATmega328P

Для нашего устройства будет использоваться реле RY610024.

4.1.8 Подключение фоторезистора к ATmega328P

Для определения уровня света используется фоторезистор.

Фоторезистор может менять свое сопротивление в зависимости от уровня окружающего освещения. Это означает, что в электронной схеме будут постоянно меняться напряжение, падающее на фоторезисторе. Фоторезистор подключается одной ногой к питанию, а другой к аналоговому входу микроконтроллера.

Единственное, что необходимо добавить из внешней дополнительной обвязки — это стягивающий резистор на 10 КОм.

Схема подключения фоторезистора представлена на рисунке 4.11.

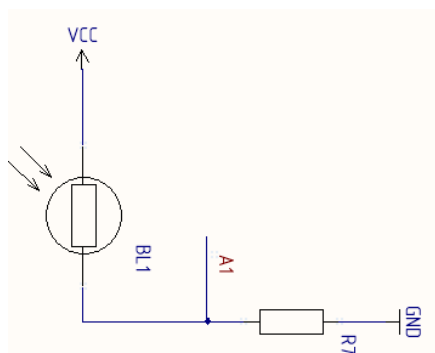


Рисунок 4.11 – Схема подключения фоторезистора к ATmega328P

Для нашего устройства будет использоваться фоторезистор MLG5506.

4.1.9 Подключение датчика температуры относительной влажности к ATmega328P

В данном проекте изменение температуры производится при помощи недорогого энергоэффективного датчика температуры относительной влажности DHT11. Датчик состоит из двух частей температура и гигрометра. Первый используется для измерения температуры, второй для влажности воздуха. DHT11 измеряет температуру в диапазоне $0-50\text{ }^{\circ}\text{C} \pm 2^{\circ}\text{C}$, влажности от 20% до 90%. Датчик подключается к микроконтроллеру через цифровой пин с добавлением подтягивающего резистора на 4,7 Ком между цифровым пином и питанием. Благодаря находящемуся внутри чипу, датчик может выполнять аналого-цифровые преобразования и выдавать цифровой сигнал [14].

DHT11 имеет собственный протокол, это выглядит следующим образом:

- контролер устанавливает сигнальную линию в ноль, а затем в единицу (хочет считать показания);
- датчик устанавливает сигнальную линию в ноль, а затем в единицу (подтверждение готовности передачи данных);
- датчик передает последовательность нулей и единиц, формирующих 5 байт;
- первые два байта передают значение температуры, вторые два байта относительную влажность воздуха, в пятом байте передается контрольная сумма.

Схема подключения датчика представлена на рисунке 4.12

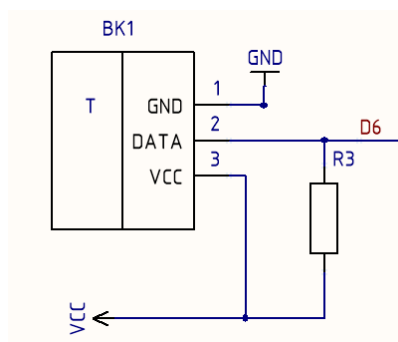


Рисунок 4.12 – Схема подключения датчика температуры относительной влажности к ATmega328P

Для нашего устройства будет использоваться датчик температуры относительной влажности DHT11.

4.1.10 Подключение часов реального времени к ATmega328P

В качестве часов реального времени используется микросхема DS3231. Это недорогая микросхема часов реального времени, подключаемая к различным устройствам по I2C интерфейсу. Для микросхемы не нужен внешний кварцевый резонатор, и несмотря на это она имеет очень высокую точность хода часов благодаря встроенному термокомпенсированному кварцевому генератору, из-за чего уход времени составляет всего ± 2 минуты за год. К микросхеме есть возможность подключить батарейку, которая необходима для поддержания точного времени даже когда питание системы отключается [15].

Для работы часов реального времени необходима подключить его к пинам SDA SCL микроконтроллера. Не стоит забывать, что для корректной работы необходимо добавить стягивающие резисторы и аккумуляторную батарею на 3.3В. Для зарядки батарей необходим тока ограничивающий резистор.

Схема подключение часов реального времени представлена на рисунке 4.13.

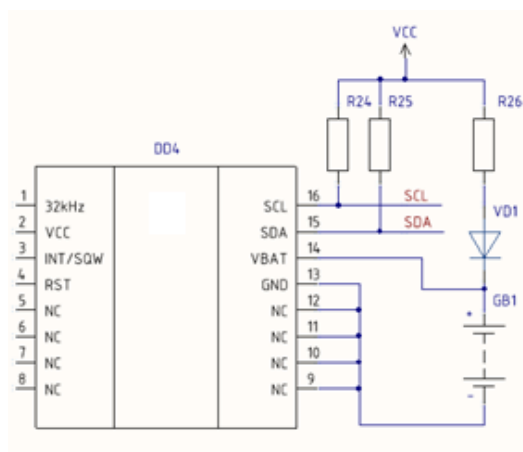


Рисунок 4.13 – Схема подключение часов реального времени к ATmega328P

Для нашего устройства будет использоваться часов реального времени DS3231.

4.1.11 Подключение диммира к ATmega328P

В качестве устройство регулирование яркости света используется самодельный диммер.

Диммер – электронное устройство, предназначенное для изменения мощности. Обычно применяется для регулировки яркости диодных лент, ламп накаливания. Для реализаций диммира можно использовать шим пины микроконтроллера, но в этом случае нельзя добиться равномерного свечения. Так как переменное напряжение синусоидальной формы, управление с помощью ШИМ будут пропускаться куски синусоиды, имеющие разную мощность рисунок 4.14. Как итог никакого плавного свечение и мерцания лампочки.

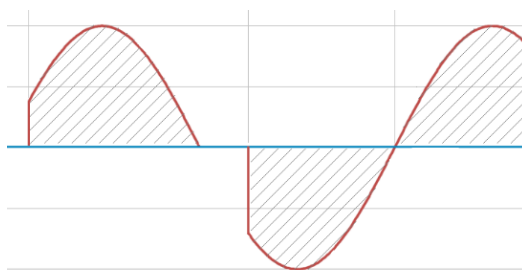


Рисунок 4.14 – Выходной сигнал под управление ШИМ

Для решение этой проблемы необходимо знать когда сигнал проходит через 0. Имея точку начало каждой полуволны, мы можем подавать напряжение в нужные моменты времени, тем самым отдавая нагрузке одинаковую мощность рисунок 4.15. А управляя слаженность мы можем изменять отдаваемую мощность.

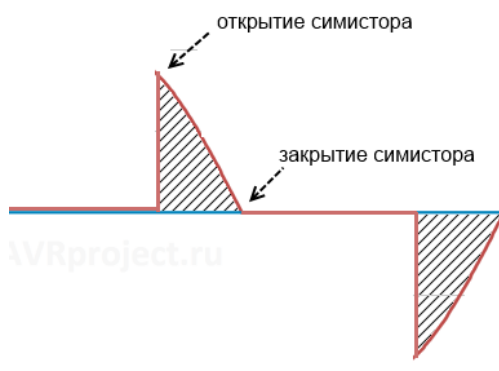


Рисунок 4.15 – Выходной сигнал под управление детектора нуля

Для обнаружения точки перехода был разработан детектор нуля. Детектор нуля подключается через подтягивающий резистор к цифровому пину микроконтроллера. Для корректной работы оптопары ставится диодный мост, преобразующий переменное напряжение постоянное. Так же схеме детектора нуля используются два гасящих резисторов перед диодным мостом [16].

Зная точки перехода синуситы через ноль и время скважности мы можем изменять уровень яркость. Для каретной работы надо рассчитать допустимое время скважности. Зная, что в Европе частота переменного тока 50 Гц по формуле:

$$T = \frac{1}{V} \quad (4.3)$$

Находим время одного периода, равное 20ms. Для того что бы пропускать одинаковые части синусоиды как в положительной, так и в отрицательной части нам нужно время полу периода, равное 10ms. Теперь если нам надо заглушить свет на половину мы проверяем детектор нуля, в случае прохождения нуля мы в течение 5ms подаем напряжение на лампу.

Схема подключения диммера представлена на рисунке 4.16.

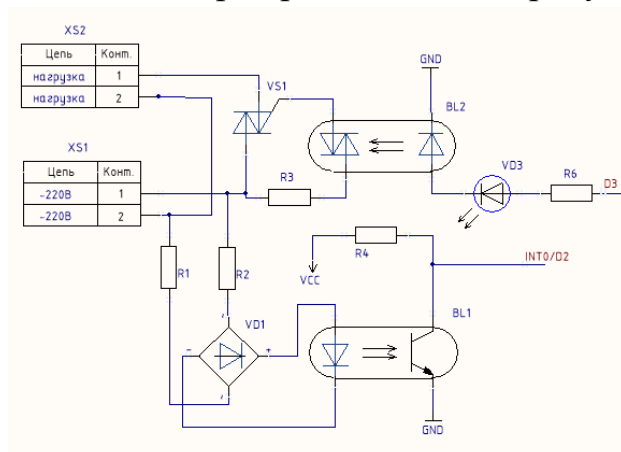


Рисунок 4.16 – Схема подключение диммера к ATmega328P

4.1.12 Подключение микрофона к ATmega328P

В проекте используется хлопковый выключатель. Для его реализации необходимо подключить микрофон к микроконтроллеру. Микрофон — это прибор, преобразующий звуковые колебания в электрические. Он подключается к аналоговому выходу микроконтроллера, так как звук после прохождения через микрофон представляется виде изменяющегося напряжения. Для подключения микрофона используется RC фильтр верхних частот, и усилитель на транзисторе

Схема подключение микрофона представлена на рисунке 4.17.

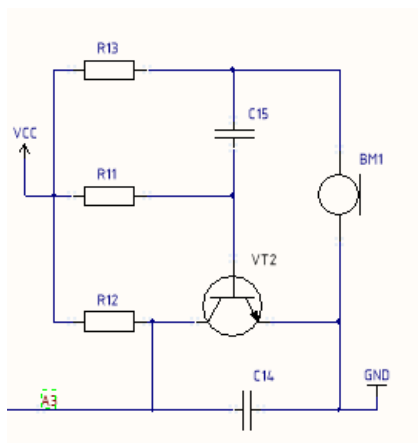


Рисунок 4.17 – Схема подключение микрофона к ATmega328P

Для нашего устройства будет использоваться микрофон PC1033.

4.1.13 Описание работы системы по принципиальной схеме

Выбор режима и управление осуществляется с помощью клавиатуры и тактовых кнопок. Клавиатура подключается к аналоговому порту микроконтроллера, а тактовые кнопки к цифровому. Так же важным элементом управления являются регуляторы яркости света и громкости звука, они подключаются к аналоговому порту микроконтроллера. К цифровым пинам системы подключен динамик, он необходим для оповещения о неисправности устройства. Отображение выбранного режима пользователем осуществляется при помощи lcd дисплея, подключённого через расширитель портов по I2C. После выбора режима пользователем мы считываем значение с датчика температуры, а также отправляем запрос на считывание значения с микрофона, датчика холла и свет при помощи радиомодуля. Затем идет проверка полученных данных, если значение совпадаю модуль уходит в режим ожидание если же нет, то по радиоканалу отправляет управляющий сигнал на диммер или двигатель постоянного тока.

Схема электрическая принципиальная сенсорного узла системы умный дом представлена на чертежах ГУИР.484434.001 ЭЗ, ГУИР.484434.002 ЭЗ, ГУИР.484434.003 ЭЗ.

Перечень элементов схемы электрической принципиальной приведён на чертежах ГУИР.484434.001 ПЭЗ, ГУИР.484434.002 ПЭЗ, ГУИР.484434.003 ПЭЗ.

4.2 Проектирование программного обеспечения

4.2.14 Разработка схемы алгоритма работы системы и программы

Разработка программного обеспечения включает в себя два основных этапа:

- разработку алгоритма работы системы;
- разработку прикладных программ на языке программирования с помощью системных средств поддержки проектирования.

Алгоритм работы системы описан с использованием всех программных доступных средств, а именно констант, переменных, ОЗУ и ПЗУ, а также портов ввода/вывода.

Подробная блок-схема алгоритма программного обеспечения устройства, включающая в себя все аспекты работы периферийных устройств и микроконтроллера приведена на чертеже ГУИР 484434.001 ПД, ГУИР 484434.002 ПД, ГУИР 484434.003 ПД.

Код программы с комментариями приведен в Приложении А.

4.2.15 Описание работы системы и программы

В данном курсовом проекте для обмена данными с внешними устройствами используются двунаправленные выводы микроконтроллера atmega328. Выводы ШИМ не используются.

Программа (далее «скетч») написана на языке программирования C++ в среде разработки Arduino IDE.

Скетч состоит из переменных и констант, настройки работы выводов платформы и бесконечный цикл выполнения алгоритма.

Перечень основных функций и процедур, используемых в коде и не упомянутых раньше, следующий:

- а) `init()` – для инициализации дисплея в программе;
- б) `backlight()` – для включения подсветки дисплея;
- в) `setCursor()` – для установления координат курсора;
- г) `clear()` – для отчистки дисплея от данных;
- д) `print(someString)` – для отображения заданной информации, начиная с координат курсора;
- е) `LiquidCrystal_I2C lcd(0x27, 16, 2);` настройка библиотеки `LiquidCrystal_I2C` с адресом 0x27, 16 столбцами и 2 строками;

ж) `isnan()`; проверка получения значения от датчика температуры;
 з) `readHumidity()`; считываем влажность;
 и) `readTemperature()`; считываем температуру в градусах цельсия;
 к) `gettime()`; получение времени;
 л) `settime()`; установка времени;
 м) `int button_2= 0`; объявление переменной целочисленного типа;
 н) `pinMode(button_p,INPUT)`; назначение направления пина на вход/выход;
 о) `setChannel ()`; указываем канал передачи данных (от 0 до 125);
 п) `radio.setDataRate (RF24_1MBPS)`; Указываем скорость передачи данных RF24_1MBPS - 1Мбит/сек;
 р) `radio.setPALevel (RF24_PA_MAX)`; Указываем мощность передатчика RF24_PA_MAX=0dBm;
 с) `radio.enableAckPayload()`; указываем что в пакетах подтверждения приёма есть блок с пользовательскими данными;
 т) `openWritingPipe()`; настройка адреса отправки;
 у) `write(&myData, sizeof(myData))`; отправка массива данных;
 ф) `float temp=1`;
 х) `nrf_service_data buf`; создание переменной buf типа NRF service data;
 ц) `buf.service_uuid = NRF_DEVICE_INFORMATION_SERVICE_UUID`; задание буферу UUI;
 ч) `buf.value = BTLE::to_nRF_Float(temp)`; преобразование float в NRF float;
 ш) `btle.advertise()`; отправка данных на телефон.

Упрощенный алгоритм работы устройства приведен ниже:

Шаг 1. Начальная инициализация состояния микроконтроллера.

Шаг 2. Ожидание нажатия кнопок выбора режима:

- 1) Если нажата SET, проверяем нажата ли клавиша ОК.
 - если да, то переходим к Шагу 3;
 - если нет, то переходим к Шагу 4.
- 2) Если нажата TIME, проверяем нажата ли клавиша ОК.
 - если да, то переходим к Шагу 3;
 - если нет, то переходим к Шагу 4.
- 3) Если нажата REAL, проверяем нажата ли клавиша нажата.
 - считываем заданные значение, переходим к Шагу 4.
- 4) Если нажата ECO, переходим к Шагу 4.
- 5) Если нажата INFO, считываем значение и переходим к Шагу 7;

- 6) Если нажата OF, выключаем подсветку и переедим к Шагу 2.
 - 7) Если не одна клавиша не нажата, переходим к Шагу 2.
- Шаг 3. Проводим настройку параметров, переходим к Шагу 7.
- Шаг 4. Считываем значение с датчиков и записываем в буфер.
- 1) Если произошла ошибка, переходим к Шагу.
- Шаг 5. Открываем функцию обработки данных.
- Шаг 6. Отправляем управляющий сигнал через радиоканал:
- Шаг 7. Выводим информации на дисплей, переходим к Шагу 2.
- Шаг 8. Проверка на нажатие клавиши ОК:
- 1) Если не нажата, вывести звуковую сигнализацию, атак же информацию об ошибке на дисплей и перейти к Шагу 8;
 - 2) Если нажата, перейти к Шагу 2.

Согласно среде разработки Arduino IDE, аппаратные затраты платформы Arduino Uno следующие:

Скетч использует 21832 байт (67%) памяти устройства. Всего доступно 32256 байт.

Глобальные переменные используют 1018 байт (49%) динамической памяти, оставляя 1030 байт для локальных переменных. Максимум: 2048 байт.

Учитывая тот факт, что для проекта был выбран не самый мощный микроконтроллер для разработки, и всем параметрам алгоритм занимает немного больше чем половину предоставляемых ресурсов, программа является легковесной, что свидетельствует о высоком быстродействии, а аппаратная составляющая будет иметь низкое энергопотребление.

4.3 Моделирование работы системы

4.3.16 Выбор среды разработки

Моделирование сенсорного узла системы умный дом осуществляется в системе автоматизированного проектирования (САПР) Proteus. Он позволяет виртуально смоделировать работу огромного количества аналоговых и цифровых устройств.

Программный пакет Proteus позволяет собрать схему любого электронного устройства и симулировать его работу, выявляя ошибки, допущенные на стадии проектирования и трассировки. Программа состоит из двух модулей:

1 ISIS – редактор электрических схем с последующей имитацией их работы.

2 ARES – редактор печатных плат, оснащенных автотрассировщиком Electra, встроенным редактором библиотек и автоматической системой размещения компонентов на плате. Кроме того, ARES может создать трехмерную модель печатной платы [17].

Proteus включает в себя более 6000 электронных компонентов со всеми справочными данными, а также демонстрационные ознакомительные проекты. Программа имеет инструменты USBCONN и COMPIR, которые позволяют подключить виртуальное устройство к портам USB и COM компьютера. При присоединении к этим портам любого внешнего прибора виртуальная схема будет работать с ним, как если бы она существовали в реальности. Proteus поддерживает следующие компиляторы: CodeVisionAVR, WinAVR (AVR), ICC (AVR, ARM7, Motorola), HiTECH (8051, PIC Microchip) и Keil (8051, ARM).

Кроме того, в Proteus можно добавлять библиотеки элементов. А если же необходимый компонент отсутствует, его можно создать самостоятельно.

4.3.17 Описание работы процесса

В процессе были использованы следующие библиотечные элементы:

1 LDR (см. рисунок 4.18). Это фоторезистор, используемый в качестве датчика света. Используется для определения уровня света.

2 VT93N2 (см. рисунок 4.19). Это микрофон. Используется для определения хлопка.

3 LCD 16x2 (см. рисунок 4.20). Это дисплей с отображением символов в 16 столбцов и 2 строки. Используется для отображения информации.

4 MOTOR-DC (см. рисунок 4.21). Двигатель постоянного тока. Используется для открытия и закрытия ролета.

5 PCF8574 (см. рисунок 4.22). I2C расширитель портов для подключения дисплея к микроконтроллеру. Основная цель этого элемента – экономия выводов микроконтроллера.

6 SOUNDER (см. рисунок 4.23). Динамик, используется для сигнализации пользователя.

7 Atmega328P (см. рисунок 4.24). Микроконтроллер, на которой проектируется система.

8 DS3232 (см. рисунок 4.25). Модуль реального времени. Применяется для определения точного времени.

9 DHT11 (см. рисунок 4.26). Датчик температуры и влажности.

10 SS49E (см. рисунок 4.27). Датчик холла, используется для определения положение ролета.

11 G2R-14-DC5 (см. рисунок 4.28). Электромагнитное реле, используется для дистанционного переключение света.

12 NRF24L01 (см. рисунок 4.29). Радиомодуль, используется для обмена данными между устройствами.

13 Для моделирования системы требуется загрузка прошивки в микроконтроллер. Для этого в среде Arduino IDE был скомпилирован код и получен hex-файл, который и является требуемой прошивкой.

14 Реализация схемы консоли в Proteus представлена на рисунке 4.30.

15 Реализация схемы модуля управление естественного освещения Proteus представлена на рисунке 4.31.

Реализация схемы модуля управление искусственного освещения Proteus представлена на рисунке 4.32.

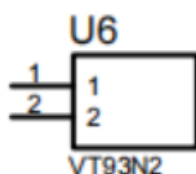


Рисунок 4.18 – Микрофон VT93N2



Рисунок 4.19 – Фоторезистор LDR

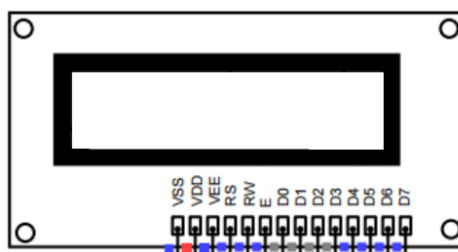


Рисунок 4.20 – Дисплей LCD 16x2

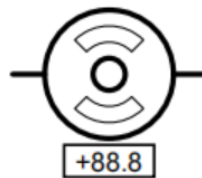


Рисунок 4.21 – Двигатель постоянного тока MOTOR-DC

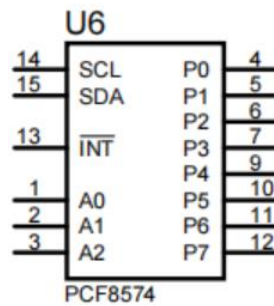


Рисунок 4.22 – Расширитель портов PCF8574

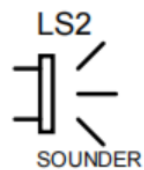


Рисунок 4.23 – Динамик SOUNDER

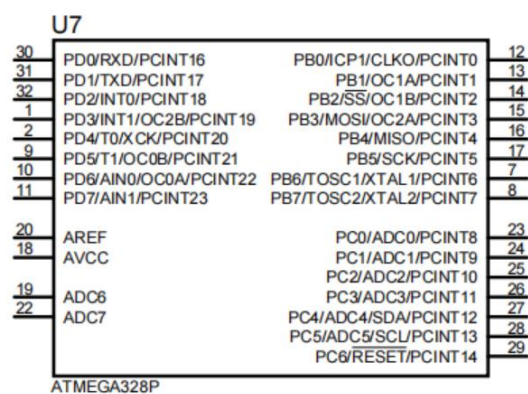


Рисунок 4.24 – Микроконтроллер Atmega328P

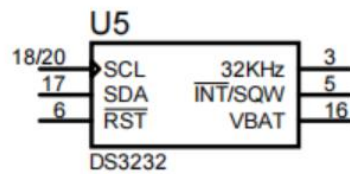


Рисунок 4.25 – Часы реального времени DS3232

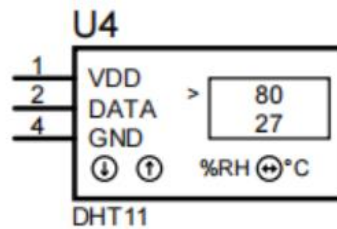


Рисунок 4.26 – Датчик температуры DHT11

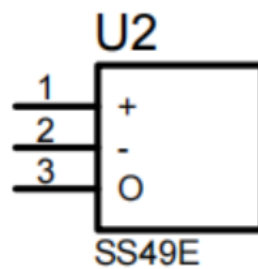


Рисунок 4.27 – Датчик холла SS49E

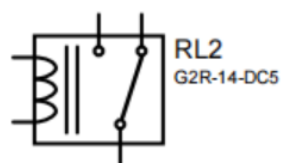


Рисунок 4.28 – Электромагнитное реле G2R-14-DC5

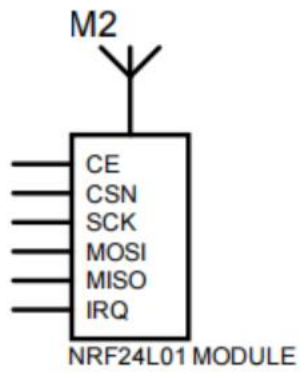


Рисунок 4.29 – Радиомодуль NRF24L01

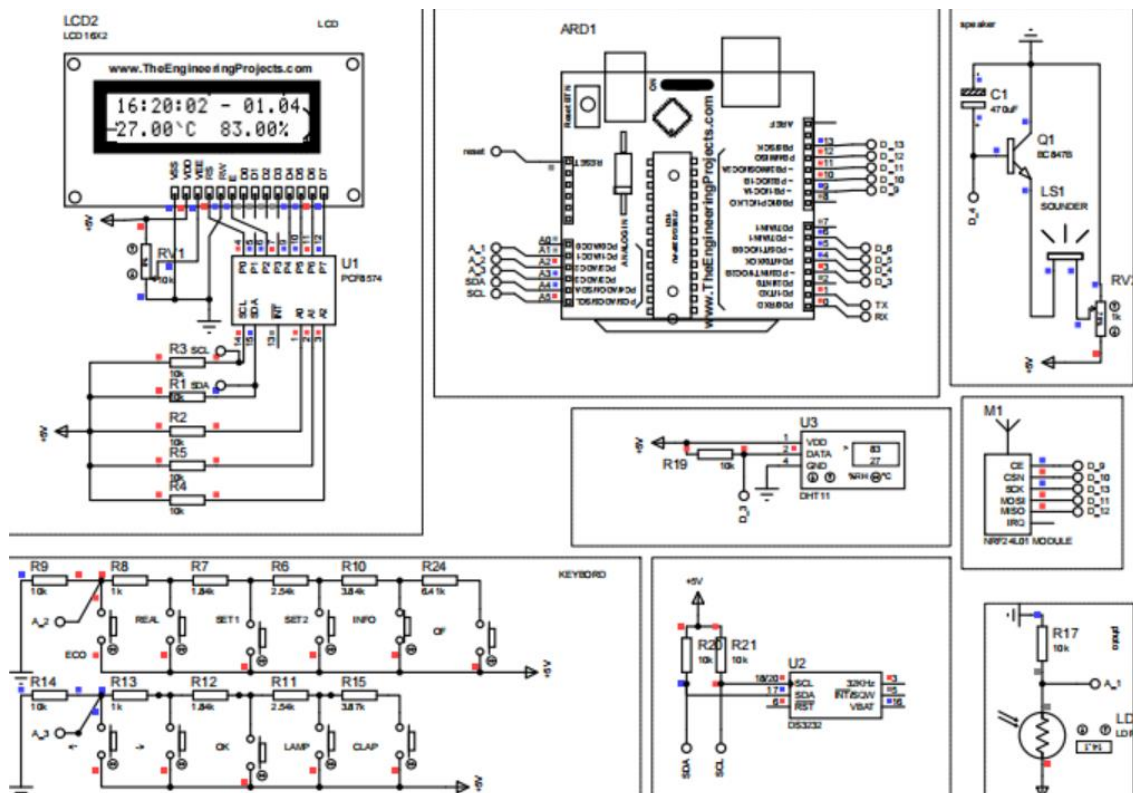
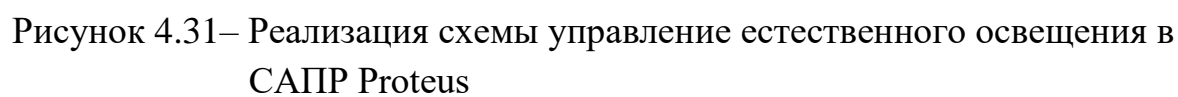


Рисунок 4.30 – Реализация схемы консоли САПР Proteus



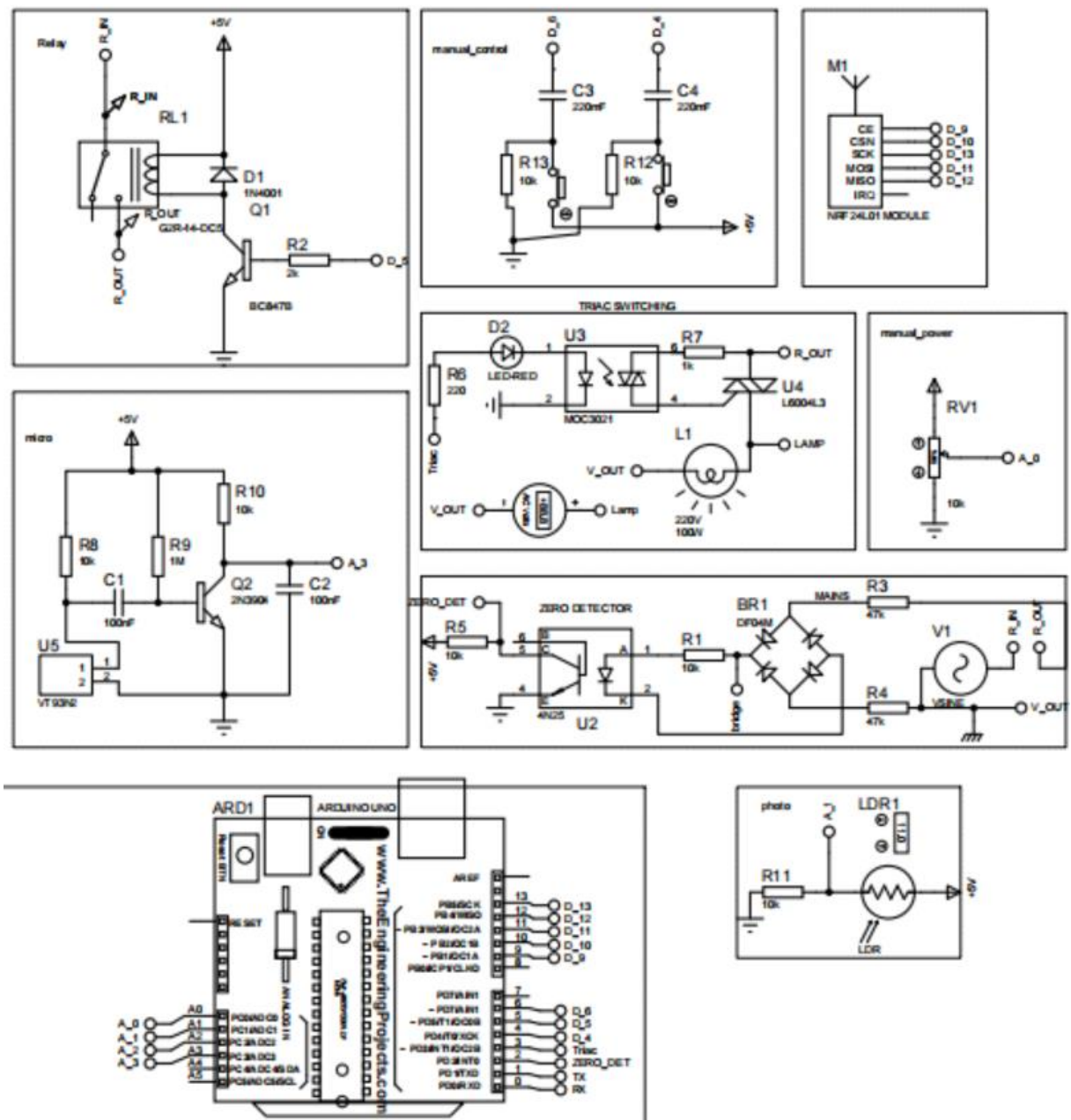


Рисунок 4.32— Реализация схемы управление искусственного освещения в САПР Proteus

5 РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ УЗЛА СЕНСОРНОЙ СЕТИ СИСТЕМЫ УМНЫЙ ДОМ

5.1 Выбор и обоснование компоновочной схемы и метода конструирования

Компоновочная схема блоков определяется количеством, видом составляющих элементов и их расположением.

Все компоновочные схемы делятся на два вида [18]:

- централизованная;
- децентрализованная.

Децентрализованная компоновочная схема устройства принимается для электронных средств, обладающих мощными выходными модулями, источниками помех и паразитных эффектов, тогда электронное средство разносится по нескольким корпусам.

В данном случае будем использовать централизованную компоновочную схему устройства, так как все его элементы располагаются в одном корпусе.

На компоновочные схемы электронных средств значительное влияние оказывают вспомогательные элементы. Это различные индикаторы, ручки управления, кнопки, экраны. В зависимости от их количества необходимо выбирать порядок их размещения на передней панели, что влияет на форму самого средства.

Предварительная компоновка электронных средств осуществляется уже на этапе технического предложения, поскольку необходимо учесть требования по габаритам и массе, которые определены в техническом задании на проектирование. На последующих стадиях проектирования происходит корректировка и уточнение компоновочных параметров. Как правило, при компоновке необходимо определить площадь и объем. В том случае, если результаты расчетов не будут соответствовать требованиям технического задания, то по согласованию с заказчиком в технически обоснованных случаях в техническое задание могут быть внесены соответствующие корректировки.

При компоновке сенсорного узла системы умный дом должны быть учтены следующие требования:

- оптимальность, устойчивость и стабильность функциональных межблочных связей;
- требования по жесткости и прочности;
- отсутствие помех;
- эргономика, удобство ремонта;

- оптимальное размещение комплектующих элементов в модулях всех уровней с учетом коэффициента заполнения по объему и удобству для осмотра и ремонта;

- сосредоточение центра тяжести ближе у опорной поверхности;

- наличие достаточного пространства для межблочных соединений.

При проектировании платы, необходимо учитывать влияние паразитных электромагнитных связей. Учет и анализ этих связей на ранней стадии проектирования позволит в значительной степени снизить затраты на производство всего изделия.

Способом решения этой проблемы является исключение с самого начала конструирования схемы дистанционной системы управления климатом причин, порождающих помехи

Для решения проблемы необходимо понять, какие виды помех наиболее вероятны в данной схеме и выбрать и разместить печатные платы, кабели и другие структурные составляющие системы таким образом, чтобы исключить как можно больше причин, вызывающих помехи.

Помехи бывают двух типов: постоянные и переживающиеся. Постоянные помехи имеют один и тот же характер. Поэтому можно легко выявить их причину. Однако могут возникнуть трудности при ее устранении, но если она устранена, то окончательно. Переживающиеся помехи появляются время от времени. Такой характер помех сильно затрудняет выявление их источника.

Проблемы возникновения помех можно свести к минимуму, изолируя чувствительные части схемы от источника помех, устраняя паразитные индуктивные и емкостные связи. Для этого необходимо:

- располагать маломощные (чувствительные) схемы поблизости от источника сигнала;

- размещать мощные схемы (в которых велика вероятность возникновения помех) вблизи нагрузок;

- располагать маломощные и мощные схемы как можно дальше друг от друга;

- стараться свести к минимуму длину проводников, использовать максимально короткие контуры прохождения тока.

5.2 Оценка потребляемой мощности и токов, протекающих в отдельных цепях

Мощность электрического тока – количество электричества, проходящего через поперечное сечение цепи в течение одной секунды, есть не что иное, как сила тока в цепи. Следовательно, мощность электрического тока будет прямо пропорциональна разности потенциалов (напряжению) и силе тока в цепи. Передача энергии по электрической цепи, рассеяние энергии, то есть переход электромагнитной энергии в тепловую, а также и другие виды преобразования энергии характеризуются интенсивностью, с которой протекает процесс, то есть тем, сколько энергии передается по линии в единицу времени, сколько энергии рассеивается в единицу времени.

Для расчёта потребляемой мощности всей схемы необходимо узнать мощность каждого отдельно взятого элемента. Некоторые значения максимальной рассеиваемой мощности взяты из официальной документации на соответствующие элементы. Будем считать, что конденсаторы, кварцевые резонаторы, предохранители и разъёмы являются идеальными и мощности не потребляют.

Потребляемые мощности элементов схемы модуля управления естественным освещением приведены в таблице 5.1.

Таблица 5.1 – Расчёт токов и мощностей, потребляемых элементной базой

Наименование	Количество, шт.	Мощность, Вт	Общая мощность, Вт
Микросхема Atmega 328	1	0,1	0,1
Микросхема NRF24L01	1	0,1	0,1
Диоды 1N4007	4	0,025	0,1
Транзисторы KT814	2	0,3	0,6
Транзисторы KT815	4	0,3	1,2
Катушки индуктивности LQW15AN3N9D10D	1	0,13	0,13
Катушки индуктивности B82496-C3829-J	1	0,2	0,2
Катушки индуктивности LQP18MN2N7C02D	1	0,13	0,13
Резисторы 0805	9	0,125	1,125

Продолжение таблицы 5.1

Наименование	Количество, шт.	Мощность, Вт	Общая мощность, Вт
Двигатель постоянного тока F130-16155	1	0,29	0,29
Датчик холла SS41	1	0,1	0,1
Фоторезистор MLG5506	1	0,2	0,2

Оценим потребляемую мощность всех элементов, подставив значения потребляемых мощностей всех элементов по формуле:

$$P_{\text{общ}} = \sum P_i * n \quad (5.1)$$

где – P_i потребляемая мощность i -го элемента, а n число элементов на схеме.

Общая потребляемая мощность составит:

$$P_i = 4,275 \text{ Вт.}$$

На основании полученных расчетов принимаем для схемы источник питания с $P = 4,275 \text{ Вт}$.

Для дальнейшего расчета ширины проводников, модуля управления естественным освещением необходимо оценить токи в наиболее нагруженных цепях. Определим максимальный протекающий ток в цепи 5В.

Через микросхемы DD1-DD2, двигатель постоянного тока и датчик холла протекают токи, максимальное значение которых 20 мА, 20 мА, 80 мА, 20 мА соответственно. Таким образом, максимальный ток в цепи 5 В равен 126 мА.

Определим минимальную ширину проводника для цепи 5 В по следующей формуле:

$$b_{\min} = \frac{I_{\max}}{J_{\text{доп}} * t} \quad (5.2)$$

где I_{\max} – максимальный постоянный ток, протекающий в проводниках;
 $J_{\text{доп}}$ – допустимая плотность тока;
 t – толщина проводника.

Подставив в формулу (5.2) имеющиеся данные, получаем минимальную ширину проводника для цепи 5В:

$$b_{min} = \frac{0,140}{20 * 0,035} = 0,20\text{мм}$$

Токи в остальных цепях пренебрежимо малы, поэтому специальной оценки производить не имеет смысла.

Потребляемые мощности элементов схемы консоли приведены в таблице 5.2.

Таблица 5.2 – Расчёт токов и мощностей, потребляемых элементной базой

Наименование	Количество, шт.	Мощность, Вт	Общая мощность, Вт
Микросхема Atmega 328	1	0,1	0,1
Микросхема NRF24L01	1	0,1	0,1
Микросхема PCF8574	1	0,07	0,07
Микросхема DS3231	1	0,027	0,027
LCD дисплей	1	0,3	0,3
Диоды 1N4148	1	0,025	0,025
Динамик KSSG1708	1	0,2	0,2
Транзисторы KT815	1	0,3	0,3
Катушки индуктивности LQW15AN3N9D10D	1	0,13	0,13
Катушки индуктивности B82496-C3829-J	1	0,2	0,2
Катушки индуктивности LQP18MN2N7C02D	1	0,13	0,13
Резисторы 0805	25	0,125	3,125
Датчик температуры DHT11	1	0,1	0,1
Фоторезистор MLG5506	1	0,2	0,2

Исходя из данных таблицы 5.2, была рассчитана общая потребляемая мощность

$$P_i = 5.207 \text{ Вт.}$$

На основании полученных расчетов принимаем для схемы источник питания с $P = 5.207$ Вт.

Для дальнейшего расчета ширины проводников консоли необходимо оценить токи в наиболее нагруженных цепях. Определим максимальный протекающий ток в цепи 5В.

Через микросхемы DD1-DD4, динамик и датчик температуры протекают токи, максимальное значение которых 60 мА, 100 мА, 20 мА соответственно. Таким образом, максимальный ток в цепи 5 В равен 180 мА.

Подставив в формулу (5.2) имеющиеся данные, получаем минимальную ширину проводника для цепи 5В:

$$b_{min} = \frac{0,180}{20 * 0,035} = 0,25\text{мм}$$

Токи в остальных цепях пренебрежимо малы, поэтому специальной оценки производить не имеет смысла.

Потребляемые мощности элементов схемы модуля управление искусственным освещением приведены в таблице 5.3.

Таблица 5.3 – Расчёт токов и мощностей, потребляемых элементной базой

Наименование	Количество, шт.	Мощность, Вт	Общая мощность, Вт
Микросхема Atmega 328	1	0,1	0,1
Микросхема NRF24L01	1	0,1	0,1
Микрофон PC1033	1	0,1	0,1
Светодиод	1	0,06	0,06
Диодный мост 2W10G-E4 / 51	1	0,2	0,2
Диоды 1N4007	1	0,025	0,025
Оптопара PC817	1	0,15	0,15
Оптопара МОС3021М	1	0,1	0,1
Транзисторы KT815	2	0,3	0,6
Катушки индуктивности LQW15AN3N9D10D	1	0,13	0,13
Катушки индуктивности B82496-C3829-J	1	0,2	0,2
Катушки индуктивности LQP18MN2N7C02D	1	0,13	0,13

Продолжение таблицы 5.3

Наименование	Количество, шт.	Мощность, Вт	Общая мощность, Вт
Резисторы 0805	14	0,125	1,75
Фоторезистор MLG5506	1	0,2	0,2

Исходя из данных таблицы 5.3, была рассчитана общая потребляемая мощность

$$P_i = 3,845 \text{ Вт.}$$

На основании полученных расчетов принимаем для схемы источник питания с $P = 3,845 \text{ Вт}$.

Для дальнейшего расчета ширины проводников модуля искусственного освещения необходимо оценить токи в наиболее нагруженных цепях. Определим максимальный протекающий ток в цепи 5В.

Через микросхемы DD1-DD2, оптопары и микрофон протекают токи, максимальное значение которых 41 мА, 65 мА, 50 мА соответственно. Таким образом, максимальный ток в цепи 5 В равен 156 мА.

Подставив в формулу (5.2) имеющиеся данные, получаем минимальную ширину проводника для цепи 5В:

$$b_{min} = \frac{0,180}{20 * 0,035} = 0,22 \text{ мм}$$

Через лампу протекает ток, максимальное значение которого 350 мА. Таким образом, максимальный ток в цепи ~220 В равен 350 мА.

Подставив в формулу (5.2) имеющиеся данные, получаем минимальную ширину проводника для цепи 5В:

$$b_{min} = \frac{0,35}{20 * 0,035} = 0,5 \text{ мм}$$

Токи в остальных цепях пренебрежимо малы, поэтому специальной оценки производить не имеет смысла.

5.3 Выбор применяемого материала печатной платы

Толщину основания печатной платы, влияющую на механическую прочность и жесткость, в основном определяют в зависимости от механических нагрузок на печатную плату и от ее конструктивных особенностей. Толщина печатной платы также определяет технологические возможности металлизации отверстий, а также зависит от материала ПП.

Поскольку была выбрана двухсторонняя печатная плата, никаких дополнительных расчётов проводить не нужно, т.к. толщина ДПП определяется толщиной материала основания с учетом толщины фольги:

$$H = H_C + H_{\Phi} \quad (5.3)$$

где H_C – толщина слоя;

H_{Φ} – толщина фольги с двух сторон;

Толщина слоя равна 1,43 мм, толщина слоя фольги 35 мкм, подставив значения в формулу (6.3) получим следующее значение:

$$H = 1,43 + 2 * 0,035 = 1,5 \text{ мм.}$$

В качестве материала для изготовления печатной платы выберем стеклотекстолит фольгированный СФ-2-35Г-1,5 толщиной 1,5 мм.

Стеклотекстолит фольгированный СФ-2-35Г-1,5 мм – листовой материал, изготовленный на основе стеклотканей с пропиткой связующим на основе эпоксидных смол и облицованный с двух сторон медной электролитической гальваностойкой фольгой толщиной 35 мкм.

5.4 Выбор типоразмера печатной платы

В зависимости от характера устройства выполняется компоновка различных ее элементов. Основной задачей, при компоновке элементов электронного средства, является расположение в пространстве различных элементов или изделий электронных средств.

Для определения размеров печатной платы электронного средства выполняются компоновочные расчеты. Рассчитываются установочные площади элементов, учитывая зазоры, которые служат для работы укладочного инструмента, поэтому рассчитанные площади умножают на коэффициент 1,3.

Формула для расчёта установочной площади элементов имеет следующий вид:

$$S_{\text{уст}} = 1,3 * b * l, \quad (5.4)$$

где b – ширина устанавливаемого элемента;

l – длина устанавливаемого элемента.

Установочная площадь элементов разрабатываемой платы консоли представлена в таблице 5.4.

Таблица 5.4 – Установочная площадь элементов

Наименование	Количество, шт.	Площадь, мм
Микросхема Atmega 328	1	468
Микросхема NRF24L01	1	32
Микросхема PCF8574	1	185
Микросхема DS3231	1	205
LCD дисплей	1	7566
Тактовая кнопка KLS7-TS1202	14	1300,9
Диоды 1N4007	1	29,5
Динамик KSSG1708	1	375,5
Транзисторы KT815	1	32,5
Катушки индуктивности LQW15AN3N9D10D	1	58
Катушки индуктивности B82496-C3829-J	1	50
Катушки индуктивности LQP18MN2N7C02D	1	58
Резисторы 0805	25	65
Кварцевый резонатор HC-49SM	2	170
Отсек под батарейку FC1-5023	1	682,5
Разъем VHR-2N	1	68,2
Разъем VHR-1N	1	15
Керамический конденсатор	15	345
Конденсатор электролитический ECAP SMD	2	73,6

Продолжение таблицы 5.4

Наименование	Количество, шт.	Площадь, мм
Переменный резистор L-20КС	2	370
Датчик температуры DHT11	1	330
Фоторезистор MLG5506	1	80

Суммарная установочная площадь всех элементов $S_{уст}$, мм², вычисляется по формуле:

$$S_{уст} = \sum_{i=1}^n (S_i \cdot n), \quad (5.5)$$

где S_i – значение установочной площади i -го элемента;
 n – количество элементов.

Для получения суммарной установочной площади платы, значения установочной площади в мм, подставляются в формулу (5.5).

$$S_{уст} = 12544,7 \text{ мм}^2$$

Площадь печатной платы $S_{пп}$, мм², вычисляется по формуле:

$$S_{пп} = \frac{S_{уст}}{m \cdot k_3}, \quad (5.6)$$

где k_3 – коэффициент заполнения платы;
 m – количество сторон монтажа.

Подставляем значение в формулу (5.6).

$$S_{пп} = \frac{12544,7}{1 \cdot 0,6} = 20907,8 \text{ мм}^2$$

Исходя из полученной площади и с учётом крепёжных отверстий для платы, а также с учётом отступа от края плата выбираем размеры платы 145 x 145 мм. Площадь такой печатной платы равна 21025 мм², что полностью удовлетворяет запросам по размеру монтажной зоны. Окончательные габариты получи в результате размещения элементов и трассировки печатной платы.

Установочная площадь элементов разрабатываемой платы управления естественным освещением представлена в таблице 5.5.

Таблица 5.5 – Установочная площадь элементов

Наименование	Количество, шт.	Площадь, мм
Микросхема Atmega 328	1	468
Микросхема NRF24L01	1	32
Тактовая кнопка KLS7-TS1202	3	278,7
Диоды 1N4007	4	118
Транзисторы KT814	2	65
Транзисторы KT815	4	130
Катушки индуктивности LQW15AN3N9D10D	1	58
Катушки индуктивности B82496-C3829-J	1	50
Катушки индуктивности LQP18MN2N7C02D	1	58
Резисторы 0805	10	26
Кварцевый резонатор HC-49SM	2	170
Разъем VHR-2N	2	136,4
Разъем VHR-1N	1	15
Керамический конденсатор	12	276
Конденсатор электролитический ECAP	1	46,8
Датчик холла SS41	1	50
Фоторезистор MLG5506	1	80

Исходя из данных таблицы 5.5, была рассчитана суммарная установочная площадь всех элементов.

$$S_{\text{уст}} = 2001,9 \text{ мм}^2$$

Рассчитываем площадь платы, подставляя данных установочной площади всех элементов в формулу (5.5).

$$S_{\text{пп}} = \frac{2001,9}{1 \cdot 0,6} = 3335,8 \text{ мм}^2$$

Исходя из полученной площади и с учётом крепёжных отверстий для платы, а также с учётом отступа от края плата выбираем размеры платы 75 x 80 мм. Площадь такой печатной платы равна 6000 мм, что полностью удовлетворяет запросам по размеру монтажной зоны. Окончательные габариты получи в результате размещения элементов и трассировки печатной платы.

Установочная площадь элементов разрабатываемой платы управления естественным освещением представлена в таблице 5.6.

Таблица 5.6 – Установочная площадь элементов

Наименование	Количество, шт.	Площадь, мм
Микросхема Atmega 328	1	468
Микросхема NRF24L01	1	32
Микрофон PC1033	1	80
Светодиод	1	30
Диодный мост 2W10G-E4 / 51	1	80
Оптопара PC817	1	95
Оптопара МОС3021М	1	100
Тактовая кнопка KLS7-TS1202	3	278,7
Диоды 1N4007	1	29,5
Транзисторы KT815	2	130
Катушки индуктивности LQW15AN3N9D10D	1	58
Катушки индуктивности B82496-C3829-J	1	50
Катушки индуктивности LQP18MN2N7C02D	1	58
Резисторы 0805	10	26
Кварцевый резонатор HC-49SM	2	165
Разъем VHR-2N	2	136,4
Разъем VHR-1N	1	15
Керамический конденсатор	12	276
Конденсатор электролитический ECAP SMD	1	46,8
Переменный резистор L-20KC	2	370
Реле RY610024	1	330
Фоторезистор MLG5506	1	157

Исходя из данных таблицы 5.6, была рассчитана суммарная установочная площадь всех элементов.

$$S_{\text{уст}} = 2964,6 \text{ мм}^2$$

Рассчитываем площадь платы, подставляя данных установочной площади всех элементов в формулу (6.6).

$$S_{\text{пп}} = \frac{2964,6}{1 \cdot 0,6} = 4940,5 \text{ мм}^2$$

Исходя из полученной площади и с учётом крепёжных отверстий для платы, а также с учётом отступа от края плата выбираем размеры платы 85 x 100 мм. Площадь такой печатной платы равна 8500 мм, что полностью удовлетворяет запросам по размеру монтажной зоны. Окончательные габариты получи в результате размещения элементов и трассировки печатной платы.

Сборочный чертеж печатной платы сенсорного узла системы умный дом представлена на чертежах ГУИР.484434.001 СБ, ГУИР.484434.002 СБ, ГУИР.484434.003 СБ.

5.5 Выбор класса точности печатной платы

В данном дипломантом проекте будет использоваться смешанная технология монтажа печатных плат, которая представляет собой сочетание технологии монтажа в сквозные отверстия и поверхностного монтажа на одной плате. Так как используются элементы поверхностного монтажа и выводные, необходимо произвести ряд конструкторских расчётов для определения параметров, таких как ширину проводников, диаметр контактных площадок, диаметр отверстий.

Будет использоваться печатная плата 1-ой группы жесткости. В нашем случае печатная плата должна удовлетворять условиям эксплуатации по ГОСТ 15.150-69 УХЛ 1.1 и 4.2.

Печатные платы 1-ой группы жесткости имеют диапазон рабочих температур от -25 до +25 и обеспечивают надежную работу при относительной влажности до 75%. Помимо этого, характеризуются высокой стойкостью к повышенному давлению [19].

Класс точности определяет наименьшее минимальное значение основных размеров конструктивных элементов ГОСТ 23751-86 определяет пять классов точности (Таблица 5.7 – Классы точности печатных плат) [20].

Таблица 5.7 – Классы точности печатных плат

Параметр	Класс точности				
	1	2	3	4	5
Мин. ширина проводника, t , мм	0,75	0,45	0,25	0,15	0,10
Мин. расстояние между центрами проводников, S , мм	0,75	0,45	0,25	0,15	0,10
Мин. ширина гарантийного пояса, B , мм	0,30	0,20	0,1	0,05	0,025
Отношение диаметра мин. отверстия к толщине ПП (γ)	1:2,5	1:2,5	1:3	1:4	1:5

Разработка печатной платы ведется по пятому классу точности. В данном случае выбор класса точности обусловлен тем, что печатная плата имеет штыревые и планарные выводы, расстояние между которыми достаточно мало. Допуски на расположение отверстий и контактных площадок для пятого класса точности ПП приведены в таблицу 5.8 .

Таблица 5.8 – Допуски на расположение отверстий и контактных площадок для 5-го класса точности ПП

Параметры, мм	Значение допуска
Допуск на отверстие Δd без металлизации, $d < 1$ мм	$\pm 0,025$
То же, $d > 1$	$\pm 0,10$
Допуск на отверстие Δd с металлизацией и оплавлением, $d < 1$ мм	+0 -0,13
То же, $d > 1$	+0,05 -0,18
Допуск на ширину проводника Δb без покрытия	$\pm 0,03$
То же, с покрытием	$\pm 0,03$
Допуск на расположение отверстий δd , при размере платы менее 180	0,05
Допуск на расположение контактных площадок δp , при размере платы менее 180	0,05

Продолжение таблицы 5.8

Параметры, мм	Значение допуска
Допуск на расположение проводников δl , при размере платы менее 180	0,02
Расстояние от края просверленного отверстия до края контактной площадки, b_M	0,025
То же, с покрытием	$\pm 0,05$

5.6 Расчёт диаметров монтажных отверстий

Диаметры монтажных и переходных отверстий металлизированных и не металлизированных должны соответствовать ГОСТ 10317-79. Предпочтительные размеры монтажных отверстий выбирают из ряда 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0; 1,1; 1,2; 1,3; 1,4; 1,5; 1,6; 1,7; 1,8; 2,0 мм [21].

Определяем номинальное значение диаметров монтажных отверстий по формуле:

$$d = d_{\text{э}} + |d_{\text{н.о.}}| + r, \quad (5.7)$$

где $d_{\text{э}}$ – максимальное значение диаметра вывода навесного ЭВС, устанавливаемого на печатную плату. Для прямоугольного вывода за диаметр берется диагональ его сечения;

$d_{\text{н.о.}}$ – нижнее предельное отклонение от номинального диаметра монтажного отверстия (определяется классом точности печатной платы и диаметром отверстия);

r – разница между минимальным диаметром отверстия и максимальным диаметром вывода ЭВС.

Подставим данные о диаметрах выводов элементов в формулу (5.7): для lcd дисплея YJD1602A-1, отсека батарейки V335:

$$d_1 = 1.07 + 0,05 + 0,1 = 1,22 \sim 1,2 \text{ мм},$$

для керамических дисковых конденсаторов, транзисторов, датчиков:

$$d_2 = 0,50 + 0,05 + 0,1 = 0,65 \sim 0,7 \text{ мм},$$

для микросхемы atmega328P, светодиода GNL-3012LED:

$$d_3 = 0,66 + 0,05 + 0,1 = 0,81 \sim 0,8 \text{ мм},$$

для разъёма VHR-2N, регулятора L-20КС:

$$d_4 = 0,8 + 0,05 + 0,1 = 0,95 \sim 1 \text{ мм},$$

для разъёма VHR-1N:

$$d_5 = 1 + 0,05 + 0,1 = 1,15 \sim 1,1 \text{ мм},$$

для динамика KSSG1708:

$$d_6 = 0,99 + 0,05 + 0,1 = 1,14 \text{ мм} \sim 1,1 \text{ мм},$$

для разъёма двигателя постоянного тока F130-16155:

$$d_7 = 1,3 + 0,05 + 0,1 = 1,45 \sim 1,5 \text{ мм},$$

для диодов 1N4148, 1N4001:

$$d_8 = 1 + 0,05 + 0,1 = 1,15 \sim 1,2 \text{ мм},$$

для крепления lcd дисплея YJD1602A-1:

$$d_{10} = 2,3 + 0,1 + 0,1 = 2,5 \text{ мм},$$

для микрофона PC1033:

$$d_{11} = 0,4 + 0,5 + 0,1 = 0,95 \sim 0,9 \text{ мм},$$

для реле RY610024:

$$d_{12} = 1,15 + 0,5 + 0,1 = 1,75 \text{ мм}.$$

5.7 Рассчитаем диаметры контактных площадок

Наименьшее номинальное значение диаметра контактной площадки D_{\min} , мм под выбранное отверстие рассчитывается по формуле:

$$D_{min} = (d + \Delta d_{\text{вo}}) + 2 * b + \Delta t_{\text{вo}} + 2 * \Delta d_{\text{тр}} + \sqrt{T_d^2 + T_D^2 + \Delta t_{\text{но}}^2}, \quad (5.8)$$

где d – номинальное значение монтажного отверстия; b – размер гарантийного пояса;

$\Delta d_{\text{вo}}$ – верхнее предельное отклонение диаметра отверстия;

$\Delta d_{\text{тр}}$ – величина протравливания диэлектрика, которая для МПП принимается равной 0,03 мм, для ОПП, ДПП, ГПК – нулю;

T_d – позиционный допуск расположения оси отверстия;

T_D – позиционный допуск расположения центра контактной площадки;

$\Delta t_{\text{вo}}$ – верхнее предельное отклонение диаметра контактной площадки;

$\Delta t_{\text{но}}$ – нижнее предельное отклонение диаметра контактной площадки.

Подставляя значение отверстий в формулу (5.8), находим минимальные значения диаметров контактных площадок, соответствующих данным отверстиям:

$$D_{\text{min1}} (0,5) = 0,74 \text{ мм.}$$

$$D_{\text{min2}} (0,7) = 0,94 \text{ мм,}$$

$$D_{\text{min3}} (0,8) = 1,04 \text{ мм,}$$

$$D_{\text{min4}} (1,0) = 1,24 \text{ мм,}$$

$$D_{\text{min4}} (1,1) = 1,34 \text{ мм,}$$

$$D_{\text{min6}} (1,2) = 1,44 \text{ мм,}$$

$$D_{\text{min7}} (1,3) = 1,54 \text{ мм,}$$

$$D_{\text{min8}} (1,5) = 1,74 \text{ мм.}$$

Максимальный диаметр контактной площадки определяется по формуле:

$$D_{\text{max}} = D_{\text{min}} + (0,02..0,06). \quad (5.9)$$

Выбираем значение допуска 0,06 и находим максимальные значения

диаметров контактных площадок:

$$D_{\max 1}(0,5) = 0,8 \text{ мм},$$

$$D_{\max 2}(0,7) = 1 \text{ мм},$$

$$D_{\max 3}(0,8) = 1,1 \text{ мм},$$

$$D_{\max 4}(1,0) = 1,3 \text{ мм},$$

$$D_{\max 5}(1,1) = 1,4 \text{ мм},$$

$$D_{\max 6}(1,2) = 1,5 \text{ мм},$$

$$D_{\max 7}(1,3) = 1,6 \text{ мм},$$

$$D_{\max 8}(1,5) = 1,8 \text{ мм}.$$

5.8 Расчет наименьшего номинальное значение ширины печатного проводника

Определение наименьшего номинального значение ширины печатного проводника рассчитывают по формуле:

$$t = t_{\text{мд}} + |t_{\text{но}}|, \quad (5.10)$$

где $t_{\text{мд}}$ – минимально допустимая ширина печатного проводника исходя из токовой нагрузки;

$\Delta t_{\text{но}}$ – нижнее предельное отклонение ширины печатного проводника.

Предельные отклонения ширины печатного проводника Δt контактной площадки, концевой печатного контакта, экрана для узкого места должны соответствовать указанным в таблице 5.8.

Подставив значения в формулу (5.10), получим:

$$t = 0,1 - 0,03 = 0,097 \text{ мм}$$

5.9 Расчет наименьшего номинального расстояния между соседними элементами печатного рисунка

Расстояние между соседними элементами печатного рисунка устанавливают в зависимости от электрических, конструктивных, технологических требований.

Наименьшее номинальное расстояние между соседними элементами печатного рисунка S , мм определяют по формуле:

$$S = S_{\text{мд}} + |t_{\text{во}}|, \quad (5.11)$$

где $S_{\text{мд}}$ – минимально допустимое расстояние между соседними элементами печатного рисунка;

$\Delta t_{\text{ВО}}$ – верхнее предельное отклонение ширины элемента печатного рисунка.

Подставив значения в формулу (5.11), получим:

$$S = 0,1 + 0,03 = 0,13 \text{ мм}$$

5.10 Трассировка печатной платы

Основная задача трассировки формулируется следующим образом: по заданной схеме соединений проложить необходимые проводники на плате, чтобы реализовать заданные технические соединения с учётом заранее заданных ограничений. Основными являются ограничения на ширину проводников и минимальные расстояния между ними.

Трассировка разрабатываемой платы была проведена с помощью САПР Altium Designer. Для этого необходимо задать определенные правила трассировки, выставив необходимые приоритеты.

Таким образом были установлены следующие ограничения:

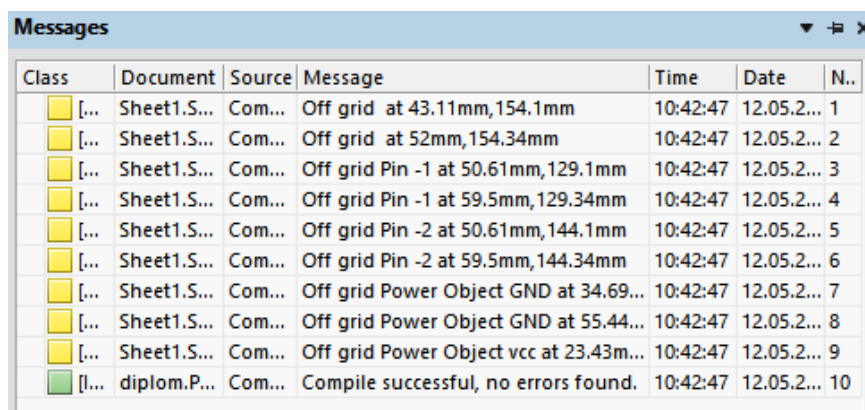
- Ширина проводника: 0,10 - 0,15 мм;
- Ширина проводника питания и земли: 0,20 - 0,5 мм;
- Расстояние между проводниками: 0,1 мм;
- Минимальный диаметр переходного отверстия: 0,5 мм;
- Топология трассировки: Shortest – соединяет все узлы по кратчайшей траектории Routing Topology.

Далее была выполнена автоматическая трассировка стратегией Default 2 Layer With Edge Connectors.

Результат трассировки печатной платы сенсорного узла системы умный дом представлена на чертежах ГУИР.758726.001, ГУИР.758726.002, ГУИР.758726.003.

5.1 Оценка качества разработанной конструкции

После окончания проектирования печатного узла в САПР Altium Designer были произведены проверки выполнения установленных правил, которую проект успешно прошёл. Проверка электрической принципиальной схемы может быть выполнена путем компиляции проекта с помощью команды Compile PCB Project. Сообщения по результатам компиляции проекта представлены на рисунке 5.1.



Class	Document	Source	Message	Time	Date	N..
[...]	Sheet1.S...	Com...	Off grid at 43.11mm,154.1mm	10:42:47	12.05.2...	1
[...]	Sheet1.S...	Com...	Off grid at 52mm,154.34mm	10:42:47	12.05.2...	2
[...]	Sheet1.S...	Com...	Off grid Pin -1 at 50.61mm,129.1mm	10:42:47	12.05.2...	3
[...]	Sheet1.S...	Com...	Off grid Pin -1 at 59.5mm,129.34mm	10:42:47	12.05.2...	4
[...]	Sheet1.S...	Com...	Off grid Pin -2 at 50.61mm,144.1mm	10:42:47	12.05.2...	5
[...]	Sheet1.S...	Com...	Off grid Pin -2 at 59.5mm,144.34mm	10:42:47	12.05.2...	6
[...]	Sheet1.S...	Com...	Off grid Power Object GND at 34.69...	10:42:47	12.05.2...	7
[...]	Sheet1.S...	Com...	Off grid Power Object GND at 55.44...	10:42:47	12.05.2...	8
[...]	Sheet1.S...	Com...	Off grid Power Object vcc at 23.43m...	10:42:47	12.05.2...	9
[I...]	diplom.P...	Com...	Compile successful, no errors found.	10:42:47	12.05.2...	10

Рисунок 5.1 – Результат выполнения компиляции всего проекта

Разработанная плата должна быть проверена на соответствие конструкторско-технологическим ограничениям, а также некоторым другим ограничениям. Проверка осуществляется с помощью команды *Tools* и выбора в диалоговом окне команды Design Rule Checks (см. рисунок 5.2).

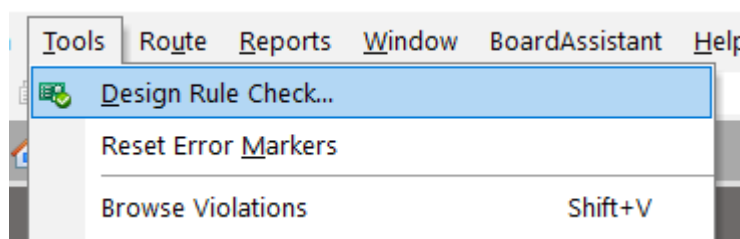


Рисунок 5.2 – Диалоговое окно инструментов в Altium Designer

Результат выполнения команды Design Rule Checks представлен на рисунке 5.3.

Summary	
Warnings	Count
	Total 0
Rule Violations	Count
Clearance Constraint (Gap=0.1mm) (All)(All)	0
Short-Circuit Constraint (Allowed=No) (All)(All)	0
Un-Routed Net Constraint (All)	0
Modified Polygon (Allow modified: No) (Allow shelved: No)	0
Width Constraint (Min=0.25mm) (Max=0.5mm) (Preferred=0.25mm) (InNetClass('VCC_GND'))	0
Width Constraint (Min=0.1mm) (Max=0.25mm) (Preferred=0.1mm) (NOT InNetClass('VCC_GND'))	0
Routing Layers(All)	0
Routing Via (MinHoleWidth=0.15mm) (MaxHoleWidth=0.3mm) (PreferredHoleWidth=0.3mm) (MinWidth=0.3mm) (MaxWidth=0.6mm) (PreferredWidth=0.6mm) (All)	0
Power Plane Connect Rule(Relief Connect)(Expansion=0.508mm) (Conductor Width=0.254mm) (Air_Gap=0.254mm) (Entries=4) (All)	0
Hole Size Constraint (Min=0.025mm) (Max=2.54mm) (All)	0
Hole To Hole Clearance (Gap=0.1mm) (All)(All)	0
Silk To Solder Mask (Clearance=0.025mm) (IsPad)(All)	0
Silk to Silk (Clearance=0.025mm) (All)(All)	0
Net Antennae (Tolerance=0mm) (All)	0
Component Clearance Constraint (Horizontal Gap = 0.1mm , Vertical Gap = 0.1mm) (All)(All)	0
Height Constraint (Min=0mm) (Max=25.4mm) (Preferred=12.7mm) (All)	0
	Total 0

Рисунок 5.3 – Результат выполнения команды Design Rule Checks

СЕНСОРНОЙ СЕТИ СИСТЕМЫ УМНЫЙ ДОМ

6.1 Характеристика нового изделия

Разрабатываемый в дипломном проекте узел сенсорной сети системы “умный дом” представляет собой набор комплектующих и главную печатную плату с предустановленным ПО для работы продукта.

Предполагаемый тип производства будет единичным и будет основываться на заказе конечного потребителя. Потребитель устройства – это компания, частные лица цель которой является автоматизация процессов контроля освещения.

Пользователь получает главную панель, устройства управление естественным и искусственным освещением. Сенсорный узел имеет модули ввода, модулями вывода. Модули ввода входят набор датчиков, клавиатура и регулятор. Модули вывода входит дисплей, динамик и набор светодиодов, которые будут извещать о действиях пользователя для большего удобства.

В устройстве используются радиочастотный модуль для передачи данных, что позволяет без проблем связывать модули на расстояние до 100м. Так же имеется возможность использовать каждое устройства как ретранслятор, что позволяет увеличить расстояние передачи. Это может быть полезно использование устройств, находящихся в офисах, частных домах, торговых центрах.

Преимущество системы в том, что пользователь получает недорогое специализированное устройство. Также преимуществом является то, что модули имеют возможность работать совместно, так и независимо друг от друга. Благодаря этому пользователи могут заменить или же докупить дополнительные блоки.

6.2 Расчет стоимостной оценки результата

6.2.1 Расчёт себестоимости и отпускной цены нового изделия

Расчёт затрат по статье «Покупные комплектующие изделия, полуфабрикаты и услуги производственного характера».

В таблице 6.1 провалены стоимости комплектующих.

Таблица 6.1 – Расчёт затрат на комплектующие и полуфабрикаты

Наименование комплектующего	Кол-во на единицу, шт.	Стоимость единицы, руб.	Суммарная стоимость, руб.
1. Микроконтроллер Atmega 328	3	8	24
2. Радиочастотный модуль NRF24L01	3	5	15
3. I2C расширитель PCF8574	1	3	3
4. Кварцевый резонатор	6	1,17	7,2
5. Дисплей LCD-1602	1	5,75	5,75
6. Датчик температуры DHT11	1	2,25	2,25
7. Часов реального времени DS3231	1	6,5	6,5
8. Динамик	1	5	5
9. Прочие материалы		30	30
Всего			106,52
Стоимость доставки комплектующих, р.			23
Итого			129,52

Расчёт затрат по статье «Основная заработная плата производственных рабочих»

В данную статью включаются расходы на оплату труда производственных рабочих, непосредственно связанных с изготовлением продукции, выполнением работ и услуг. Расчёт основной заработной платы представлен в таблице 6.2.

Часовая тарифная ставка определяется путем деления месячной базовой тарифной ставки на среднемесячное количество рабочих часов. Базовая тарифная ставка с 01.05.2022 составляет 210 руб. Среднемесячное количество расчетных рабочих часов – 168. Часовая тарифная ставка получается равной 1.25 руб.

Таблица 6.2 – Расчёт основной заработной платы

Вид работ (операции)	Разряд работ	Коэффициент разряда работ	Часовая тарифная ставка, руб./ч	Норма времени по операции, ч	Прямая зарплата (расценка), руб.
Заготовительные	3	1,14	1,34	1	1,34

Продолжение таблицы 6.2

Вид работ (операции)	Вид работ (опера ции)	Вид работ (операц ии)	Вид работ (операц и)	Вид работ (операции)	Вид работ (операц и)
Установка элементов на печатную плату	5	1,29	1,517	2	3,03
Монтажная	6	1,38	1,623	2	3,24
Контрольная	9	1,68	1,976	2	3,95
Сборочные	4	1,21	1,423	1	1,44
Премия (60%)					7,8
Всего					20,8

Дополнительная зарплата ($З_д$) определяется следующим образом:

$$З_д = \frac{З_0 * Н_д}{100} \quad (6.1)$$

где $Н_д = 20\%$ – норматив дополнительной заработной платы, установленный предприятием.

$$З_д = \frac{20,8 * 20}{100} = 4,6 \text{ руб.}$$

Сумма на накладные расходы рассчитывается по формуле:

$$P_н = \frac{З_0 * Н_н}{100}, \quad (6.2)$$

где $Н_н$ – норматив накладных расходов (150-200%), $Н_н = 150\%$.

$$P_н = \frac{З_0 * Н_н}{100} = 20,8 * 1,5 = 31,2 \text{ руб.}$$

Производственная себестоимость рассчитывается по формуле:

$$C_{\text{пр}} = P_{\text{м}} + P_{\text{к}} + Z_{\text{о}} + Z_{\text{д}} + P_{\text{соц}} + P_{\text{н}} \quad (6.3)$$

Отчисления на социальные нужды ($P_{\text{соц}}$) определяются следующим образом:

$$P_{\text{соц}} = \frac{(Z_{\text{о}} + Z_{\text{д}}) * H_{\text{соц}}}{100} \quad (6.4)$$

где $H_{\text{соц}}$ – страховые взносы на обязательное социальное страхование наёмных работников (34%) и обязательное страхование от несчастных случаев на производстве (1%).

$$P_{\text{соц}} = \frac{(20,8 + 4,6) * (34 + 1)}{100} = 8,89 \text{ руб.}$$

$$C_{\text{пр}} = 129,52 + 20,8 + 4,6 + 8,89 + 31,2 = 195,01 \text{ руб.}$$

Коммерческие расходы вычисляются по формуле (при $H_{\text{ком}}=5\%$):

$$P_{\text{ком}} = \frac{C_{\text{пр}} * H_{\text{ком}}}{100} \quad (6.5)$$

$$P_{\text{ком}} = \frac{195,01 * 5}{100} = 9,75 \text{ руб.}$$

Полная себестоимость вычисляется по формуле:

$$C_{\text{п}} = C_{\text{пр}} + P_{\text{ком}} \quad (6.6)$$

$$C_{\text{п}} = 195,01 + 9,75 = 204,76 \text{ руб.}$$

Плановая прибыль на единицу продукции при $H_{\text{ре}} = 31\%$ составит:

$$П_{\text{ед}} = \frac{C_{\text{п}} * H_{\text{ре}}}{100} \quad (6.7)$$

$$П_{ед} = \frac{204,76 * 31}{100} = 63,4 \text{ руб.}$$

Отпускная цена рассчитывается по формуле:

$$Ц_{отп} = C_{п} + П_{ед} = 195,01 + 63,4 = 258,41 \text{ руб.}$$

Результаты расчётов себестоимости и отпускной цены сенсорного узла системы умный дом в таблице 6.3.

Таблица 6.3– Расчёт себестоимости и отпускной цены единицы продукции

Наименование статьи затрат	Условное обозначение	Значение, руб.
1. Покупные комплектующие	P_k	129,52
2. Основная заработная плата производственных рабочих	$З_o$	20,8
3. Дополнительная заработная плата производственных рабочих	$З_d$	4,6
4. Отчисления на социальные нужды	$P_{соц}$	8,89
5. Накладные расходы	P_n	31,2
7. Производственная себестоимость	$C_{пр}$	195,01
8. Коммерческие расходы	$P_{ком}$	9,76
9. Полная себестоимость	$C_{п}$	204,76
10.Плановая прибыль на единицу продукции	$П_{ед}$	63,4
11. Отпускная цена	$Ц_{отп}$	258,41

В результате расчётов получаем, полная себестоимость сенсорного узла системы умный дом– 204 рубля 76 копеек, а отпускная цена – 258 рублей 41 копейка. Основными конкурентами раздаваемой системы является компании Z-Wave производящие систему под ключ и имеющую стоимость за базовый комплект 830 рублей. А компания Xiaomi имеющая множество универсальных блоков с возможность управление с телефона, подсчитав цену похожих блоков от компании Xiaomi, мы получили цену в 346 рублей. Проанализировав рынок, можно с уверенностью сказать, что цена разрабатываемой системы заметно меньше стоимости продукции конкурентов.

6.2.2 Расчёт чистой прибыли

На основе маркетинговых исследований и заказов потребителей было установлено количество производимых изделий как 300 изделий в первый год. В последующие годы планируется увеличение выпуска продукции с каждым годом на 30% (второй год – 390, третий год – 507, четвертый год – 659). Исходя из этого, высчитываем чистую прибыль. Ставка налога на прибыль $\Pi_{\text{п}}$ составляет 18%.

Расчёт чистой прибыли за 4 года производства:

$$\text{Пч1} = N1 * \text{П}_{\text{ед}} * \left(1 - \frac{\Pi_{\text{п}}}{100}\right) = 300 * 63,4 * 0,82 = 15596,4 \text{руб.}$$

$$\text{Пч2} = N2 * \text{П}_{\text{ед}} * \left(1 - \frac{\Pi_{\text{п}}}{100}\right) = 390 * 63,4 * 0,82 = 20275,3 \text{руб.}$$

$$\text{Пч3} = N3 * \text{П}_{\text{ед}} * \left(1 - \frac{\Pi_{\text{п}}}{100}\right) = 507 * 63,4 * 0,82 = 26357 \text{руб.}$$

$$\text{Пч4} = N4 * \text{П}_{\text{ед}} * \left(1 - \frac{\Pi_{\text{п}}}{100}\right) = 659 * 63,4 * 0,82 = 34260 \text{руб.}$$

6.3 Расчёт инвестиций в производство нового изделия

Инвестиции в производство нового изделия включают:

- инвестиции на разработку нового изделия (Ираз);
- инвестиции в основной и оборотный капитал.

Инвестиции на разработку (Ираз) сенсорного узла системы умный дом, согласно сметной документации предприятия, составят 3250 рублей.

Производство продукции предполагается осуществлять на действующем оборудовании на свободных производственных мощностях, поэтому инвестиции в основной капитал не требуются.

Для производства нового вида продукции требуется прирост инвестиций в собственный оборотный капитал в размере 30% общей годовой потребности в материальных ресурсах.

Годовая потребность в комплектующих изделиях определяется по формуле:

$$\Pi_{\text{к}} = P_{\text{к}} * N = 129,52 * 300 = 38856 \text{руб.}$$

Инвестиции в прирост собственного оборотного капитала вычисляются по формуле:

$$I_{об} = \left(\frac{30}{100}\right) * P_k = 0,3 * 38856 = 11656,8 \text{ руб.}$$

Таким образом, инвестиции в производство нового изделия составят:

$$I = I_{раз} + I_{об} = 3250 + 11656,8 = 14906,8 \text{ руб.}$$

Инвестиции в собственный оборотный капитал изменяются по годам из-за увеличения производимых устройств на 20% каждый год в отношении предыдущего. Из этого следует формула расчёта инвестиций в оборотный капитал (при учёте прироста собственного оборотного капитала на 30%):

$$I_{об} = 0,2 * (P_k + P_m), \quad (6.8)$$

где P_k – годовая потребность в комплектующих;

P_m – годовая потребность в расходных материалах.

$$I_{об1} = I_{об} = 11656,8 \text{ руб.}$$

$$I_{об2} = 0,3 * P_k = 0,3 * 190 * 129,52 = 7382,64 \text{ руб.}$$

$$I_{об3} = 0,3 * P_k = 0,3 * 207 * 129,52 = 8043,19 \text{ руб.}$$

$$I_{об4} = 0,3 * P_k = 0,3 * 359 * 129,52 = 13949,30 \text{ руб.}$$

6.4 Расчёт показателей экономической эффективности проекта

При оценке эффективности инвестиционных проектов необходимо осуществить приведение затрат и результатов, полученных в разные периоды времени, к расчётному году путём умножения затрат и результатов на коэффициент дисконтирования, который определяется следующим образом:

$$at = \frac{1}{(1 + E_n)^{t-tp}}$$

где E_n – требуемая норма дисконта (15%),

t – порядковый номер года, затраты и результаты которого приводятся к расчётному году;

tp – расчётный год (в качестве расчётного года принимается год вложения инвестиций, т.е. $tp=1$).

Для первых четырех лет, при $E_H=15\%$, коэффициенты дисконтирования равны соответственно:

$$at_1 = \frac{1}{(1+E_H)^{t-tp}} = 1 \quad (6.9)$$

$$at_2 = \frac{1}{(1+E_H)^{t-tp}} = \frac{1}{1+0,15} = 0,8695$$

$$at_3 = \frac{1}{(1+E_H)^{t-tp}} = \frac{1}{(1+0,15)^2} = \frac{1}{1,3225} = 0,7561$$

$$at_4 = \frac{1}{(1+E_H)^{t-tp}} = \frac{1}{(1+0,15)^3} = \frac{1}{1,521} = 0,6575$$

Расчёт чистого дисконтированного дохода за четыре года реализации проекта и срока окупаемости инвестиций представлены в таблице 6.4.

Таблица 6.4 – Экономические результаты работы предприятия

Наименование показателей	Обозна чение	По годам производства			
		2022	2023	2024	2025
Результат					
Выпуск изделий, шт.	N _t	300	390	507	659
Прирост чистой прибыли, руб.	П _ч	15596,4	20275,3	26357	34260
Коэффициент дисконтирования	a _t	1	0,8695	0,7561	0,6575
Прирост чистой прибыли с учётом фактора времени, руб.	П _{чт}	15596,4	17629,3	19928,5	22525,9
Затраты (инвестиции)					
Инвестиции в разработку нового изделия, руб.	И _{раз}	3250	0	0	0

Продолжение таблицы 6.4

Наименование показателей	Обозначение	По годам производства			
		2022	2023	2024	2025
Инвестиции в собственный оборотный капитал, руб.	И _{об}	11656,8	7382,64	8043,19	13949,3
Общая сумма инвестиций, руб.	И	14906,8	7382,64	8043,19	13949,3
Инвестиции с учётом фактора времени, руб.	И _t	14906,8	6419,20	6081,41	9171,66
Чистый дисконтный доход по годам, руб.	ЧДД _t	689,6	13176,23	14895,1	16836,1
Чистый дисконтный доход с учетом времени, руб.	ЧДД	689,6	13865,83	28760,93	45597,83

Расчёт рентабельности инвестиций производится по формуле:

$$P_{\text{и}} = \frac{П_{\text{чср}}}{\sum_1^4 И_{\text{т}}} * 100\% = \frac{18920}{14906,8 + 6419,2 + 6081,41 + 9171,66} * 100\% = 51\%$$

где $П_{\text{чср}}$ – среднегодовая величина чистой прибыли за расчётный период, которая определяется по формуле

$$П_{\text{чср}} = \frac{\sum_{t=1}^n П_{\text{чт}}}{n} = \frac{15596,4 + 17629,3 + 19928,5 + 22525,9}{4} = 18920 \text{ руб.}$$

где $П_{\text{чт}}$ – чистая прибыль, полученная в году t , руб.

В результате технико-экономического обоснования инвестиций по производству нового изделия были получены следующие значения показателей их эффективности:

1. Чистый дисконтированный доход за четыре года производства продукции составит 45597,83 руб.

2. Все инвестиции окупаются на конец первого года.

3. Рентабельность инвестиций составляет 51%.

Таким образом, производство нового вида изделия является эффективным и инвестиции в его производство целесообразны.

ЗАКЛЮЧЕНИЕ

В данном дипломном проекте была разработана и описана с конструкторской, технологической и экономической части сенсорный узел системы умный дом.

В итоге дипломного проекта было выполнено описание разработки устройства, его конструирование в САПР Altium Designer и AutoCAD, включая плату устройства, выполнено моделирование с использованием САПР Proteus для тестирования взаимодействия модулей в совокупности с запрограммированной прошивкой.

По результатам технико-экономического обоснования инвестиций по производству нового изделия можно сделать вывод, что все инвестиции окупятся по окончании первого года производства, что свидетельствует о эффективных разработке и производстве данного решения.

Отчет о проверке на заимствование представлен в приложении Б. Результат проверки на заимствование составил 90,68%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] История умного дома [Электронный ресурс]. – Режим доступа: <https://tech-house.su/istoriya-poyavleniya-umnogo-doma/>.

[2] “Internet of Things” Wireless Sensor Networks : Databook / IEC. – Peter Lanctot, Kang Lee, 2014.

[3] Методы увеличения времени жизни беспроводных сенсорных сетей: справочник / Н.М. Коплин [и др.]. – М. : Московский государственный технический университет им. Н.Э. Баумана, 2020. –256с

[4] Протоколы связи для "умного дома" [Электронный ресурс]. – Режим доступа: <https://www.ferra.ru/review/smarthome/SmartHome-Protocols.htm>.

[5] Z-WARE [Электронный ресурс]. – Режим доступа: <https://z-wave.by/2013/08/30/kak-rabotaet-z-wave/>.

[6] Белтелеком [Электронный ресурс]. – Режим доступа: <https://beltelecom.by/private/smart-home>

[7] Компания Atmel [Электронный ресурс]. – Режим доступа: <https://bit.ly/34dqO7A>.

[8] МК типа PIC [Электронный ресурс]. – Режим доступа: <https://studbooks.net/2355507/tehnika/tipa>.

[9] Модуль приемопередачи NRF24L01 [Электронный ресурс]. – Режим доступа: <https://www.quartz1.com/price/PIC/405N0472616.pdf>.

[10] Протокол шины I2C [Электронный ресурс]. – Режим доступа: <https://russianblogs.com/article/4584112848/>.

[11] Расширитель PCF8674 [Электронный ресурс]. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/18212/PHILIPS/PCF8574.html>.

[12] Дисплей YJD1602A-1 [Электронный ресурс]. – Режим доступа: <https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf>.

[13] Принцип работы Н-моста [Электронный ресурс]. – Режим доступа: <https://arduino-master.ru/uroki-arduino/shema-raboty-n-mosta-dlya-upravleniya-dvigatel-yami/>.

[14] Датчик DHT11 [Электронный ресурс]. – Режим доступа: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>.

[15] Часы реального времени DS3231 [Электронный ресурс]. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/112132/DALLAS/DS3231.html>.

[16] Детектор нуля [Электронный ресурс]. – Режим доступа: <https://radio-samodel.ru/detektor%20nulya.html>

[17] Proteus [Электронный ресурс]. – Режим доступа: <https://all-audio.pro/c14/shemi/shemi-v-proteuse.php>

[18] Базовый принцип конструирования РЭА / Е. М. Парфенов, В. Ф. Афанасенко, В. И. Владимиров, Е. В. Саушкин; Под ред. Е. М. Парфенова. - М.: Радио и связь, 1981.

[19] ГОСТ 23752-79 Платы печатные общие технические условия. – Издательство стандартов, 1991

[20] ГОСТ 23751-86 Платы печатные общие параметры конструкции. – Издательство стандартов, 1986

[21] ГОСТ 10317-79 Платы печатные основные размеры. – Издательство стандартов, 1985

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программы

```
// код консоли
#include <SPI.h>; // Библиотека для работы с шиной SPI
#include <nRF24L01.h>; // Файл конфигурации для библиотеки RF24
#include <RF24.h>; // Библиотека для работы с модулем NRF24L01
#include <BTLE.h>
#define PIN_CE 9 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define PIN_CSN 10 // Номер пина Arduino, к которому подключен вывод

CSN радиомодуля
RF24 radio(PIN_CE, PIN_CSN); // Создаём объект radio с указанием выводов CE и CSN
BTLE btle(&radio);

int myData[3]={1,1,1}; // Объявляем массив для передачи данных в пакете подтверждения приёма
(до 32 байт включительно).
int ackData[3]={1,1,1}; // Объявляем массив для приёма и хранения данных (до 32 байт
включительно).

#include <DHT.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Include LiquidCrystal_I2C library
#include <iarduino_RTC.h> // подключаем библиотеку для часов
iarduino_RTC time(RTC_DS1307); // создаем объект time
LiquidCrystal_I2C lcd(0x27, 16, 2); // Настройка библиотеки LiquidCrystal_I2C с адресом 0x27, 16
столбцами и 2 строками

int DHTPIN=3;
DHT dht(DHTPIN, DHT11);
int speaker=4;
int back=1;
int set_1[5] = {1,1,1,50,50};
int eco_1[3]={50,50,1};
int eco=90;
int time_1[4]={15,30,14,1};
int real_1[4]={1,1,50,50};
int PIN_PHOTO_SENSOR=A1;
int butten_m=5;
int butten_p=6;
int butten_1 = A3;
int keyValue_1 = 0; // Состояние покоя
int oldKeyValue_1=0; // Переменная для хранения предыдущего значения состояния кнопок
int butten_2 = A2;
int keyValue_2 = 0; // Состояние покоя
int oldKeyValue_2=0;

void setup() {
pinMode(speaker,OUTPUT);
```

```

pinMode(butten_p,INPUT);
pinMode(butten_m,INPUT);
lcd.init(); // Инициализация ЖК-модуля I2C
lcd.backlight();// Включить подсветку
lcd.setCursor(0,0 ); // Перейти к столбцу 0, строке 0
lcd.print("HELLO");
dht.begin();
Serial.begin(9600);//port
radio.begin(); // Инициализируем работу nRF24L01+
radio.setChannel (27); // Указываем канал передачи данных (от 0 до 125), 27 - значит передача
данных осуществляется на частоте 2,427 ГГц.
radio.setDataRate (RF24_1MBPS); // Указываем скорость передачи данных (RF24_250KBPS,
RF24_1MBPS, RF24_2MBPS), RF24_1MBPS - 1Мбит/сек.
radio.setPALevel (RF24_PA_MAX);// Указываем мощность передатчика (RF24_PA_MIN=-18dBm,
RF24_PA_LOW=-12dBm, RF24_PA_HIGH=-6dBm, RF24_PA_MAX=0dBm).
radio.enableAckPayload();// Указываем что в пакетах подтверждения приёма есть блок с
пользовательскими данными.
time.begin();
time.settime(0,20,16,1,4,22,25); // сек, мин, часы, дата, мес, год, день недели
btle.begin("data");
delay(2000);
}

// Функция устраняющаядребезг
int GetKeyValue(int butten, int oldKeyValue) {
static long lastChange; // Переменная для хранения времени последнего изменения состояния
int actualKeyValue = analogRead(butten); // Получаем актуальное состояние
//actualKeyValue = ((actualKeyValue - 16) >> 5) + 1; // Для 32 кнопок
//actualKeyValue = ((actualKeyValue - 32) >> 6) + 1; // Для 16 кнопок
//actualKeyValue = ((actualKeyValue - 64) >> 7) + 1; // Для 8 кнопок
//actualKeyValue = ((actualKeyValue - 128) >> 8) + 1; // Для 4 кнопок
actualKeyValue = ((actualKeyValue - 64) >> 7) + 1; // Для 8 кнопок
if ((actualKeyValue != oldKeyValue) && (millis() - lastChange > 200)) { // Пришло новое значение, и с
последнего изменения прошло достаточно времени (и &)

oldKeyValue = actualKeyValue; // Запоминаем новое значение
lastChange = millis(); // Обнуляем таймер
}
return oldKeyValue; // Отправляем старое, либо уже модифицированное новое значение
}

void SET(int *mas){
int newKeyValue = GetKeyValue(butten_1,oldKeyValue_1); // Получаем актуальное состояние кнопок с
коррекциейдребезга

if(mas[0]==1){
switch(newKeyValue){

case 8:
mas[0]!=1?mas[0]=1:mas[0]=0;
lcd.setCursor(0, 0);
lcd.println("SET CUSTOMIZ");

```

```

break;

case 7:
mas[1]!=1?mas[1]=1:mas[1]=0;
lcd.setCursor(0, 1);
mas[1]!=1? lcd.print("LAMP OF"): lcd.print("LAMP ON");
break;

case 6:
mas[2]!=1?mas[2]=1:mas[2]=0;
lcd.setCursor(0, 1);
mas[2]!=1? lcd.print("CLAP OF"): lcd.print("CLAP ON");
break;

case 5:
1==digitalRead(butten_m)?mas[3]=mas[3]-5:(
1==digitalRead(butten_p)?mas[3]=mas[3]+5:mas[3]=mas[3] );
lcd.setCursor(0, 1);
lcd.print("light"+String(mas[3]));
break;

case 4:
1==digitalRead(butten_m)?mas[4]=mas[4]-10:(
1==digitalRead(butten_p)?mas[4]=mas[4]+10:mas[4]=mas[4] );
lcd.setCursor(0, 1);
lcd.print("roller"+String(mas[4]));
break;

default:
break;
}}

else{
lcd.setCursor(0, 0);
lcd.println("SET_1");
lcd.setCursor(0, 1);
lcd.print(time.getTime("H:i:s - d.m"));
Serial.println(time.getTime("H:i:s - d.m"));
radio.openWritingPipe (0x1616161616LL);
myData[0]=mas[4];
radio.write(&myData, sizeof(myData[1])); // Отправляем данные из массива myData указывая
сколько байт массива мы хотим отправить.
radio.openWritingPipe (0x1515151515LL);
myData[0]=mas[2];
myData[1]=mas[1];
myData[2]=mas[3];
radio.write(&myData, sizeof(myData)); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
}
}

```

```

void ECO(int *mas){
radio.openWritingPipe (0x1616161616LL);
myData[0]=eco_1[0];
radio.write(&myData, sizeof(myData)); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
if( radio.isAckPayloadAvailable() ){ // Если в буфере имеются принятые данные из пакета
подтверждения приёма, то ...
radio.read(&ackData, sizeof(ackData)); // Читаем данные из буфера в массив ackData указывая сколько
всего байт может поместиться в массив.
mas[0]=ackData[1];
}
delay(50);
int val = analogRead(PIN_PHOTO_SENSOR);

if(val-eco<0){
mas[0]=mas[0]-10;
myData[0]=mas[0]-10;
radio.write(&myData, sizeof(myData));
if(mas[0]!=0){
Serial.println("rollet-: " + String(mas[0]));
ECO(mas);
}
else{
radio.openWritingPipe (0x1515151515LL);
myData[2]=eco_1[1];
radio.write(&myData, sizeof(myData)); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
if( radio.isAckPayloadAvailable() ){ // Если в буфере имеются принятые данные из пакета
подтверждения приёма, то ...
radio.read(&ackData, sizeof(ackData)); // Читаем данные из буфера в массив ackData указывая
сколько всего байт может поместиться в массив.
mas[1]=ackData[0];
mas[1]=mas[1]+30;
myData[2]=mas[1]+30;}

radio.write(&myData, sizeof(myData));
Serial.println("lamp+: " + String(mas[1]));
ECO(mas);
}
}
else{
if(val-eco>0){
radio.openWritingPipe (0x1515151515LL);
myData[2]=eco_1[3];
radio.write(&myData, sizeof(myData)); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
if( radio.isAckPayloadAvailable() ){ // Если в буфере имеются принятые данные из пакета
подтверждения приёма, то ...
radio.read(&ackData, sizeof(ackData)); // Читаем данные из буфера в массив ackData указывая
сколько всего байт может поместиться в массив.
mas[1]=ackData[0];
}
}
}
}

```

```

    mas[1]=mas[1]-30;
    myData[2]=mas[1]-30;
    if(mas[1]!=0){
        Serial.println("lamp-: " + String(mas[1]));
        ECO(mas);
    }
    }
    else{
        radio.openWritingPipe (0x161616161616LL);
        myData[0]=eco_1[0];
        radio.write(&myData, sizeof(myData)); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
        if( radio.isAckPayloadAvailable() ){ // Если в буфере имеются принятые данные из пакета
подтверждения приёма, то ...
            radio.read(&ackData, sizeof(ackData)); // Читаем данные из буфера в массив ackData указывая
сколько всего байт может поместиться в массив.
            mas[0]=ackData[1];
        }
        myData[0]=mas[0]+10;
        mas[0]=mas[0]+10;
        Serial.println("rollet+: " + String(mas[0]));
        ECO(mas);
    }
}

}

void TIME(int *tim){
//int newKeyValue = GetKeyValue(butten_1,oldKeyValue_1); // Получаем актуальное состояние кнопок с
коррекцией дребезга
if (8==GetKeyValue(butten_1,oldKeyValue_1)){
    tim[3]!=1?tim[3]=1:tim[3]=0;
}

if(tim[1]==1){
    lcd.setCursor(0, 0);
    lcd.println("TIME CUSTOMIZ");
    if(4== GetKeyValue(butten_1,oldKeyValue_1)){// Получаем актуальное состояние кнопок с коррекцией
дребезга
        1==digitalRead(butten_m)?tim[2]=tim[2]-10:( 1==digitalRead(butten_p)?tim[2]=tim[2]+10:tim[2]=tim[2]
);
        lcd.setCursor(0, 1);
        lcd.print("roller"+String(tim[2]));
    }

if(digitalRead(butten_m)==1){
    tim[1]=tim[1]+1;
    if(tim[1]>24)
        tim[1]=tim[1]-24;
    }
    if(digitalRead(butten_p)==1){
        tim[0]=tim[0]+1;
    }
}

```

```

    if(tim[0]>24)
        tim[0]=tim[0]-24;
    }
    lcd.setCursor(0, 1);
    lcd.print("FROM "+String(tim[0])+"ON "+String(tim[1]));
}
else{
    lcd.setCursor(0, 0);
    lcd.println("TIME OF ROLLET");
    lcd.setCursor(0, 1);
    lcd.print(time.getTime("H:i:s - d.m"));
    if(time.getTime("H")>tim[1]&&time.getTime("H")<tim[2]){
        radio.openWritingPipe (0x1616161616LL);
        myData[0]=tim[2];
        radio.write(&myData, sizeof(myData[1]));          // Отправляем данные из массива myData
        указывая сколько байт массива мы хотим отправить.
    }
}
}
}

```

```

void REAL(int *mas){
int newKeyValue = GetKeyValue(butten_1,oldKeyValue_1); // Получаем актуальное состояние кнопок с
коррекцией дребезга
switch(newKeyValue){

    case 7:
        mas[0]!=1?mas[0]=1:mas[0]=0;
        lcd.setCursor(0, 1);
        mas[0]!=1? lcd.print("LAMP OF"): lcd.print("LAMP ON");
        break;

    case 6:
        mas[1]!=1?mas[1]=1:mas[1]=0;
        lcd.setCursor(0, 1);
        mas[1]!=1? lcd.print("CLAP OF"): lcd.print("CLAP ON");
        break;

    case 5:
        1==digitalRead(butten_m)?mas[2]=mas[2]-30:(
        1==digitalRead(butten_p)?mas[2]=mas[2]+30:mas[2]=mas[2] );
        lcd.setCursor(0, 1);
        lcd.print("light"+String(mas[2]));
        break;

    case 4:
        1==digitalRead(butten_m)?mas[3]=mas[3]-10:(
        1==digitalRead(butten_p)?mas[3]=mas[3]+10:mas[3]=mas[3] );
        lcd.setCursor(0, 1);
        lcd.print("roller"+String(mas[3]));
        break;
}
}

```



```

    default:
    break;
}

radio.openWritingPipe (0x1616161616LL);
myData[0]=mas[3];
radio.write(&myData, sizeof(myData[1])); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
radio.openWritingPipe (0x1515151515LL);
myData[0]=mas[1];
myData[1]=mas[0];
myData[2]=mas[2];
radio.write(&myData, sizeof(myData)); // Отправляем данные из массива myData указывая сколько
байт массива мы хотим отправить.
delay(50); // Устанавливаем задержку на 50 мс.
}

void EROR(){
    int actualKeyValue = analogRead(A3);
    actualKeyValue = ((actualKeyValue - 64) >> 7) + 1; // Для 8 кнопок
    if(actualKeyValue!=8){
        if (millis() % 1000 == 0)
            digitalWrite(speaker,1);
        lcd.setCursor(0, 0);
        lcd.println("EROR");
        lcd.setCursor(0, 1);
        lcd.println("PLEASE CLICK OK");
        EROR();
    }
}

void loop() {

    int oldKeyValue_2=GetKeyValue(butten_2,oldKeyValue_2); // Получаем актуальное состояние кнопок с
    коррекцией дребезга
    if (oldKeyValue_2 > 0) { // Если значение больше 0, значит кнопка нажата
        switch (oldKeyValue_2){

            case 8:
                lcd.setCursor(0, 0);
                lcd.println("Real");
                REAL(real_1);
                break;

            case 7:
                SET(set_1);
                break;

            case 6:
                TIME(time_1);
                break;

```

```

case 5:
  lcd.setCursor(0, 0);
  lcd.println("SAVING");
  lcd.setCursor(0, 1);
  lcd.print(time.gettime("H:i:s - d.m"));
  Serial.println(time.gettime("H:i:s - d.m"));
  ECO(eco_1);
  break;

case 4:
  lcd.setCursor(0, 0);
  lcd.print(time.gettime("H:i:s - d.m"));
  Serial.println(time.gettime("H:i:s - d.m"));

float h = dht.readHumidity(); //Измеряем влажность
float t = dht.readTemperature(); //Измеряем температуру
if (isnan(h) || isnan(t)) { // Проверка. Если не удастся считать показания, выводится «Ошибка
  считывания», и программа завершает работу
  Serial.println("Ошибка считывания");//доб ввывод ошибки
  EROR();
}
lcd.setCursor(0, 1);
lcd.print(String(t)+"`C "+String(h)+"%");
float temp=1;
nrf_service_data buf;
buf.service_uuid = NRF_DEVICE_INFORMATION_SERVICE_UUID; =
buf.value = BTLE::to_nRF_Float(temp);
if (!btle.advertise(0x16, &buf, sizeof(buf))) {
  EROR();
}
btle.hopChannel();
break;

case 3:
  delay(2000);
  back==1?(lcd.noBacklight()):(lcd.backlight());
  back==1?back==0:back==1;
  Serial.println("of");
  break;
}
}
if (millis() % 1000 == 0)
  lcd.clear();

}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Код устройства управление искусственным освещением
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include <SPI.h>; // Библиотека для работы с шиной SPI
#include <nRF24L01.h>; // Файл конфигурации для библиотеки RF24
#include <RF24.h>; // Библиотека для работы с модулем NRF24L01

```

```

#define PIN_CE 9 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define PIN_CSN 10 // Номер пина Arduino, к которому подключен вывод CSN радиомодуля

RF24 radio(PIN_CE, PIN_CSN); // Создаём объект radio с указанием выводов CE и CSN

int myData[3]={1,1,50}; // Объявляем массив для приёма и хранения данных (до 32 байт
    включительно).
int ackData[2]={50,50}; // Объявляем массив для передачи данных в пакете подтверждения приёма
    (до 32 байт включительно).


int MicroInPin = A3; //micro
int SensorValue = 400;//value micro
int pinPhoto = A1;
int TRIAC =3; //пин управления мощностью
int relay=5;


int manual_power=A0; //ручная регулировка


int maxbr = 977; // Значение при максимальной освещенности
int minbr = 0.5; // Значение при минимальной освещенности.
int photo = 0;//время для фото резистора


int button_manual=4;
int button_of=6;


boolean clap=1;


void setup()
{
    pinMode(TRIAC, OUTPUT);
    pinMode(manual_power,INPUT);
    pinMode(relay,OUTPUT);
    pinMode(MicroInPin,INPUT);
    pinMode( pinPhoto, INPUT );
    pinMode(button_manual, INPUT );
    pinMode(button_of, INPUT );


    Serial.begin(9600);


    radio.begin();// Инициализируем работу nRF24L01+
    radio.setChannel (27); // Указываем канал передачи данных (от 0 до 125), 27 - значит передача
        данных осуществляется на частоте 2,427 ГГц.
    radio.setDataRate (RF24_1MBPS); // Указываем скорость передачи данных (RF24_250KBPS,
        RF24_1MBPS, RF24_2MBPS), RF24_1MBPS - 1Мбит/сек.
    radio.setPALevel (RF24_PA_MAX); // Указываем мощность передатчика (RF24_PA_MIN=-18dBm,
        RF24_PA_LOW=-12dBm, RF24_PA_HIGH=-6dBm, RF24_PA_MAX=0dBm).
    radio.enableAckPayload();// Указываем что в пакетах подтверждения приёма есть блок с
        пользовательскими данными.

```

```

radio.openReadingPipe (1, 0x1515151515LL); // Открываем 1 трубу с адресом 0x1515151515, для
приема данных.
radio.startListening (); // Включаем приемник, начинаем прослушивать открытые трубы.
radio.writeAckPayload (1, &ackData, sizeof(ackData) ); // Помещаем данные всего массива ackData в
буфер FIFO. Как только будут получены любые данные от передатчика на 1 трубе, то данные из
буфера FIFO будут отправлены этому передатчику вместе с пакетом подтверждения приёма его
данных.
}
// В модуле имеется 3 буфера FIFO, значит в них одновременно может находиться до трёх разных или
одинаковых данных для ответа по одной или разным трубам.
// После отправки данных из буфера FIFO к передатчику, соответствующий буфер очищается и
способен принять новые данные для отправки.

void zero_crosss_int() // function to be fired at the zero crossing to dim the light
{
  int dimming=analogRead(manual_power);
  // Firing angle calculation : 1 full 50Hz wave =1/50=20ms
  // Every zerocrossing thus: (50Hz)-> 10ms (1/2 Cycle)
  // (10000us) / 1023 = 9
  if(digitalRead(button_manual)!=1){
    if(radio.available()){ // Если в буфере приёма имеются принятые данные от передатчика, то ...
      radio.read( &myData, sizeof(myData) );
      dimming=myData[3];
      ackData[0]=myData[3];
      radio.writeAckPayload (1, &ackData, sizeof(ackData) );
    }
  }
  int dimtime = (9*dimming);
  delayMicroseconds(dimtime); // Off cycle
  Serial.println("button"+String());
  digitalWrite(TRIAC, HIGH); // triac firing
  delayMicroseconds(10); // triac On propogation delay (for 60Hz use 8.33)
  digitalWrite(TRIAC, LOW); // triac Off
}

void loop() {
  if(1==digitalRead( button_manual)){
    if(1==digitalRead( button_of)){
      digitalWrite(relay,1);
      if(analogRead(MicroInPin)>300){ // считываем значение с выхода АЦП на канале A0
        clap=!clap;
        digitalWrite(relay,clap); // если на выходе АЦП наблюдается пик напряжения (sensorValue больше
        300), то переключаем реле
      }
    }
    attachInterrupt(0, zero_crosss_int, FALLING);
  } else
    digitalWrite(relay,0);

  }else{
    if(radio.available()){ // Если в буфере приёма имеются принятые данные от передатчика, то ...

```

```
radio.read( &myData, sizeof(myData) ); // Читаем данные из буфера приёма в массив myData
указывая сколько всего байт может поместиться в массив.
if(C1){
    if(analogRead(MicroInPin)>300){ // считываем значение с выхода АЦП на канале A0
        clap=!clap;
        digitalWrite(relay,clap);// если на выходе АЦП наблюдается пик напряжения (sensorValue больше
300), то переключаем реле
    }
}
ackData[1]= analogRead( pinPhoto );
attachInterrupt(0, zero_crosss_int, FALLING);
}
else{
digitalWrite(relay,0);
}

}
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
устройство управление естественным освещением
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include <SPI.h>; // Библиотека для работы с шиной SPI
#include <nRF24L01.h>; // Файл конфигурации для библиотеки RF24
#include <RF24.h>; // Библиотека для работы с модулем NRF24L01
#define PIN_CE 9 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define PIN_CSN 10 // Номер пина Arduino, к которому подключен вывод CSN радиомодуля

RF24 radio(PIN_CE, PIN_CSN); // Создаём объект radio с указанием выводов CEи CSN
int   myData[3]={50,13,18};// Объявляем массив для приёма и хранения данных (до 32 байт
включительно).
int   ackData[2]={50,50};// Объявляем массив для передачи данных в пакете подтверждения приёма (до
32 байт включительно).

int forward =8;
int back=7;
int up=6;
int down=5;
int holl=3;
int pinPhoto=A1;


void setup()
{   pinMode(up, INPUT);
    pinMode(down, INPUT);
    pinMode(forward, OUTPUT);
    pinMode(back, OUTPUT);
    pinMode(holl,INPUT);
    radio.begin();// Иницилируем работу nRF24L01+
    radio.setChannel    (27); // Указываем канал передачи данных (от 0 до 125), 27 - значит передача
данных осуществляется на частоте 2,427 ГГц.
    radio.setDataRate    (RF24_1MBPS); // Указываем скорость передачи данных (RF24_250KBPS,
RF24_1MBPS, RF24_2MBPS), RF24_1MBPS - 1Мбит/сек.
```

```

radio.setPALevel (RF24_PA_MAX); // Указываем мощность передатчика (RF24_PA_MIN=-18dBm,
RF24_PA_LOW=-12dBm, RF24_PA_HIGH=-6dBm, RF24_PA_MAX=0dBm).
radio.enableAckPayload();// Указываем что в пакетах подтверждения приёма есть блок с
пользовательскими данными.
radio.openReadingPipe (1, 0x1616161616LL); // Открываем 1 трубу с адресом 0x1616161616, для
приёма данных.
radio.startListening (); // Включаем приемник, начинаем прослушивать открытые трубы.
radio.writeAckPayload (1, &ackData, sizeof(ackData) ); // Помещаем данные всего массива ackData в
буфер FIFO. Как только будут получены любые данные от передатчика на 1 трубе, то данные из
буфера FIFO будут отправлены этому передатчику вместе с пакетом подтверждения приёма его
данных.
}
// В модуле имеется 3 буфера FIFO, значит в них одновременно может находиться до трёх разных или
одинаковых данных для ответа по одной или разным трубам.
// После отправки данных из буфера FIFO к передатчику, соответствующий буфер очищается и способен
принять новые данные для отправки.

void position(int a, int mas[]){
if(mas[0]!=160&&mas[0]!=0){
a>0?digitalWrite(forward,1):digitalWrite(back,1);
delay(100);
(digitalRead(holl)==1&&a>0)?digitalWrite(forward,0):digitalWrite(back,0);
a>0?mas[0]=mas[0]+10:mas[0]=mas[0]-10;
}
}

void loop(){
if(1==digitalRead(up) || 1==digitalRead(down)){
digitalRead(up)==1?position(1,ackData):position(0,ackData);
}
if(radio.available()){ // Если в буфере приёма имеются принятые данные от передатчика, то ...
radio.read( &myData, sizeof(myData) ); // Читаем данные из буфера приёма в массив myData
указывая сколько всего байт может поместиться в массив.
int a=(ackData[0]-myData[0])/10;
if(a>=0){
if(a!=0){
for(int i=a;i>=0;i--){
position(1,ackData);
}
}
else{
for(int i=a;i<=0;i++){
position(0,ackData);
}
}
}
ackData[1]= analogRead( pinPhoto );
radio.writeAckPayload (1, &ackData, sizeof(ackData) ); // Помещаем данные всего массива ackData в
буфер FIFO для их отправки на следующее получение данных от передатчика на 1 трубе.
}
}
}

```

ПРИЛОЖЕНИЕ Б

(обязательное)

Отчет о проверке на заимствования

Отчет о проверке на заимствования представлен на рисунке А.1.



Рисунок А.1 – Отчет о проверке на заимствования

_____ Дейлид Р.В.