

федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт энергетики и электротехники  
(институт)

Кафедра «Промышленная электроника»

11.03.04 Электроника и наноэлектроника  
(код и наименование направления подготовки, специальности)

Промышленная электроника  
направленность (профиль)

## БАКАЛАВРСКАЯ РАБОТА

на тему Разработка будильника с информационными оповещениями и управлением жалюзи

Студент(ка)	<u>И.Д. Самарин</u> (И.О. Фамилия)	_____	(личная подпись)
Руководитель	<u>А.В. Прядилов</u> (И.О. Фамилия)	_____	(личная подпись)
Консультанты	<u>Н.В. Андрюхина</u> (И.О. Фамилия)	_____	(личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент А.А. Шевцов  
(ученая степень, звание, И.О. Фамилия) \_\_\_\_\_  
(личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2018 г.

## Аннотация

Тема выпускной квалификационной работы – «Разработка будильника с информационными оповещениями и управлением жалюзи». В этом проекте разрабатывается опытный образец будильника, обладающего дополнительными (помимо основных, стандартных) функциями: получение информационных пакетов с погодными условиями из сети Интернет при помощи технологии Wi-Fi; звуковое (речевое) оповещение, содержащее информацию о погоде вместо музыкального звонка (или следом за ним); автоматическое открывание жалюзи с целью скорейшего пробуждения человека. Работа содержит 4 части: первая посвящена раскрытию и пояснению темы работы и целей её проведения, в остальных детально рассматриваются схемотехника устройства, его программное обеспечение, конструкция и проверка требуемых характеристик.

В вводной части аргументируется актуальность проводимой работы, ставятся её цели и задачи. Здесь же приводится комплексное сравнение элементов разрабатываемой модели будильника с уже существующими решениями в области электроники, программирования и конструирования механических узлов. Доказывается оригинальность исполнения устройства и его функциональных характеристик.

В второй части ведётся полное рассмотрение электрической части будильника. Данная модель базируется на применении готовых модулей из области информационной и силовой электроники. Модули представляют собой законченные печатные узлы с выводами dip, расположенными по краям платы. В пунктах данной части приводятся описания модуля обработки информации, построенного на базе микроконтроллера ESP8266, модуля питания на базе линейного стабилизатора AMS1117, модуля воспроизведения информационных оповещений, записанных на SD-карту в удобном формате mp3, дисплея-индикатора, платы-расширения для управления электроприводом жалюзи. Описываются принципы

электрических соединений модулей и их обвязки в виде пассивных электронных компонентов.

Третья часть раскрывает логический аспект, закладываемый в алгоритмы работы устройства, комментируются части программного кода, написанного в Arduino IDE. Сложность написания скетча заключается в достаточно большом количестве модулей, связь с которыми должен поддерживать микроконтроллер.

В последней части приводятся разработка корпуса устройства, расположение внутри него печатных узлов. Акцентируется внимание на особенностях элементов конструкции и планах по их дальнейшему улучшению. Заключением этой части является анализ результатов тестирования будильника и введения его в эксплуатацию.

## Abstract

The title of the bachelor's thesis is "Development of an alarm clock with information alerts and jalousie control". The thesis is devoted to the development of a prototype alarm clock, which besides basic, standard functions has some additional ones, such as: obtaining data bursts with weather conditions from the Internet using Wi-Fi technology; sound (voice) notification, containing information about the weather instead of (or following) a musical call; automatic blinds opening with the aim of a speedy awakening of a person.

The aim of the work is to prove the expedience of creating and operating convenience of the developed alarm clock.

The premise of the work was the fact that tracking weather conditions is an integral part of a person's leisure, as they largely determine his state of health, his mood. It can even affect his actions and the daily routine. Weather conditions, as well as the time of the day, determine the illumination of the room.

The bachelor's thesis is divided into four logically connected parts.

In the introduction, we give a complex comparison of the elements of the developed alarm clock model with the already existing solutions. The originality of the device performance and its functional characteristics is proved.

The second part is a complete consideration of the electrical part of the alarm clock. Particular attention is paid to the usage of ready-made modules of industrial information and power electronics. This part of the project gives details about the information-processing module based on the ESP8266 microcontroller. The readers' attention is also drawn to other modules that ensure the operation of the alarm clock.

Next, in the third part, we explain parts of the code written in the Arduino IDE.

In the last part, attention is paid to the features of the structural elements and plans for their further improvement. We then analyse the results of testing the alarm clock and putting it into operation.

## Список обозначений и сокращений

СУ – система управления

JSON (англ. JavaScript Object Notation,) — текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми.

SMD (**SMD**-технология, SMT (англ. surface mount technology и от англ. surface mounted device — прибор, монтируемый на поверхность) — технология поверхностного монтажа печатных плат, также называют ТМП (технология монтажа на поверхность), компоненты для поверхностного монтажа также называют «чип-компонентами».

GPIO – Интерфейс ввода/вывода общего назначения (англ. *general-purpose input/output*,) – интерфейс для связи между компонентами компьютерной системы, к примеру микропроцессором и различными периферийными устройствами. Контакты GPIO могут выступать как в роли входа, так и в роли выхода – это, как правило, конфигурируется. GPIO контакты часто группируются в порты.

SNR – Отношение сигнал/шум (ОСШ; англ. *signal-to-noise ratio*, сокр. *SNR*) – безразмерная величина, равная отношению мощности полезного сигнала к мощности шума.

UART – Универсальный асинхронный приёмопередатчик (УАПП, англ. *Universal Asynchronous Receiver-Transmitter, UART*) – узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами.

NTP (англ. *Network Time Protocol* — протокол сетевого времени) — сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью. Протокол был разработан Дэвидом Л. Миллсом, профессором Делавэрского университета, в 1985 году. Версия на 2015 год — NTPv4.

## Содержание

Введение.....	7
1. Состояние вопроса.....	9
1.1 Формулирование актуальности, цели и задач проекта.....	9
1.2 Обзор существующих решений .....	11
2. Схемотехническая часть .....	21
2.1 Общие принципы построения системы.....	21
2.2 Узел обработки информации.....	25
2.3 Описание микроконтроллера ESP8266 .....	26
2.4 Дисплей для отображения текущих параметров погоды и времени...	31
2.5 Узел воспроизведения звуковых сообщений.....	33
2.6 Датчик прерывания луча.....	36
2.7 Назначение портов микроконтроллера ESP8266 .....	38
3. Программирование в среде Arduino IDE.....	39
4. Конструктивно-экспериментальная часть.....	57
4.1 Разработка и сборка конструкции.....	57
4.2 Отладка конструкции и экспериментальные исследования .....	60
Заключение .....	62
Список используемой литературы .....	63

## Введение

Элементы домашней автоматизации в последние годы всё более плотно входят в нашу жизнь: во многих современных устройствах бытовой техники можно найти функции интеллектуального управления: от включения и выключения приборов в определённое время до системного анализа воздействующих факторов и принятия определённых решений по дальнейшей работе устройства. Последовательность действий в результате таких решений называют сценариями.

В настоящее время энтузиасты всего мира в своих домашних лабораториях разрабатывают аппаратное и программное обеспечение для систем типа «Умный дом». В каждом конкретном случае алгоритмы и функции, которые должна поддерживать система, разные. Это связано с тем, что у разных пользователей возникает потребность автоматизации в разных сферах быта.

С 21 века мир стремительно растёт с помощью технологий. Люди требуют технологии более высокого качества как с аппаратными, так и программными аспектами. Мир движется в сторону поколения Интернета вещей (IoT), и к 2020 году почти 50 миллиардов устройств будут подключены к Интернету, и для того, чтобы выполнить требования, технологии должны быть оптимизированы, надёжны, распространяемы, и должны быть использованы для благого дела. [1]

Существуют решения, осуществляющие тотальный контроль всех систем дома. Они сложны в реализации, так как требуют больших подготовительных работ перед запуском, рассмотреть хотя бы сотни метров электропроводки, предназначенной для соединения главного микроконтроллера с периферией (датчики, исполнительные устройства, промежуточные узлы). Зачастую такая избыточность увеличивает себестоимость и временные затраты на проведение работ по монтажу,

настройке и отладке системы. Кроме того, готовые программные решения «под ключ» могут быть неоправданно дороги.

Именно поэтому пользователи вынуждены искать бюджетные решения, наилучшим образом подходящие под конкретные задачи, которые имеют место при данных условиях проживания, ритме жизни, нюансах географического положения жилища, в общем, потребностях человека. Стоит задуматься, что аппаратная периферия и алгоритмы работы таких систем тем сложнее, чем большее количество человек в рамках одной жилплощади ею пользуются. У каждого сожителя возникают свои пожелания к работе инфраструктуры. У разработчика возникает своеобразная система уравнений с логическим элементом И: все удовлетворены тогда и только тогда, когда каждый получает по потребностям.

Выбор темы настоящего проекта связан с желанием сконструировать нечто похожее на интеллектуальную систему JARVIS из фильма «Железный человек», в расшифровке «Just A Rather Very Intelligent System», (просто очень умная система) – искусственный интеллектуальный компьютер Тони Старка (главного героя). Он запрограммирован на разговор мужским голосом в британском акценте. JARVIS – домашняя вычислительная система Тони Старка, которая заботится обо всем, что связано с домом, от систем отопления и кондиционирования до анализа двигателя автомобиля Старка в гараже.[2]

В рамках данной работы будет рассмотрена разработка аппаратного и программного обеспечения устройства для реализации нескольких функций, взятых из фильма же, в пределах одной комнаты многокомнатной квартиры. Разумеется, можно всегда при наличии средств расширить зону действия системы.

## 1. Состояние вопроса

### 1.1 Формулирование актуальности, цели и задач проекта

Отслеживание погодных условий является неотъемлемым элементом досуга человека, поскольку они во многом определяют его самочувствие, настроение, то, в какой одежде он пойдёт на работу, могут даже повлиять на его действия и распорядок дня. Погодные условия, а также время суток определяют освещённость помещения.

Температура и влажность окружающей среды меняются каждый день, для людей сделать прогноз может показаться обычным делом, но он никогда не сбывается точно. Смена сезона является основной причиной изменений среды, поэтому температура и влажность колеблются с различными погодными условиями. Вариации в окружающей среде могут быть разрушительными и дорогостоящими для людей, если они не ожидают их и не следят за ними должным образом, чтобы планировать такие вариации в будущем. [1]

Обычно для регулирования проникновения солнечных лучей через окно применяют шторы или жалюзи. Перед окном может быть расположен рабочий стол, за которым проводятся операции, требующие определённого уровня освещённости для устранения напряжённости зрения человека. Удалённое управление жалюзи позволяет человеку не отвлекаться на рутинные операции регулировкой жалюзи.

Одна из функций разрабатываемого устройства – извещение человека о погодной обстановке в зрительной и звуковой форме, другая – увеличение светового потока в зависимости от текущего времени (пробуждение).

Решение вышеописанных задач благотворно скажется на состоянии человека, плодотворности его работы и эффективности отдыха, поэтому разработка данного устройства является актуальной задачей.

**Цель** данной работы: изготовление действующего прототипа устройства.

Следующие **задачи** поставлены и выполнены для достижения этой цели:

1. Обзор существующих решений
2. Выбор основных узлов схемы
3. Разработка электрической принципиальной схемы
4. Разработка расположения и электрических соединений узлов на макетной плате
5. Написание программы для микроконтроллера
6. Отладка программы
7. Разработка корпуса устройства
8. Определение степени соответствия результатов работы системы поставленным задачам

## 1.2 Обзор существующих решений

Рассмотрим похожие проекты, включающие функции, аналогичные тем, что должна выполнять разрабатываемая система.

Парсинг данных с интернет-сайтов. **Парсинг (Parsing)** – определение, которое принято в информатике по синтаксическому анализу. Кратко алгоритм действия парсера для всех случаев одинаков:

- выход в Сеть, доступ к коду веб-ресурса и его скачивание;
- считывание и обработка данных;
- представление полученных данных в удобном виде – файлы .txt, .sql, .xml, .html. Также парсингом пользуются программы автоматической проверки уникальности информации в текстовой форме. Они быстро сравнивают содержание многочисленных веб-страниц с заданным текстом.[3]

Парсингом также можно назвать получение данных, заключённых в определённую структуру устройствами на базе микроконтроллеров с возможностью выхода в Интернет посредством проводной или беспроводной (W-Fi) связи. В настоящей работе будет использоваться формат JSON, так как существуют готовые библиотеки для Си-подобных языков программирования для раскрытия данных из такой структуры.

Достаточно подробно процесс забора данных о погоде описан в [4]. Сервис, использующийся в статье, называется **OpenWeatherMap**. Он предоставляет большие возможности и бесплатен при несущественных ограничениях. Предполагается регистрация на сайте сервиса и получение индивидуального API-кода идентификации пользователя. Применяется в качестве аппаратного обеспечения модуль NodeMcu на базе микроконтроллера ESP-8266. Готовые библиотеки и скетчи позволяют повторить задумку.

Вывод данных на экран/табло. В статье [5] под названием «Контроллер управления вытяжным вентилятором с WiFi» рассмотрено создание

прототипа устройства, очень близкого по функциональности к настоящему проекту. Как и в данной работе, автору было необходимо, помимо всего прочего, выводить некоторые параметры системы, такие как информация с датчиков температуры и влажности, время, синхронизирующееся по NTP протоколу. Внимание было обращено на недорогие модули семисегментного четырёхразрядного индикатора, построенные на микросхеме TM1637 [6]. Выбор обосновывается его замечательными характеристиками: небольшие размеры платы (42x24мм), питание от 3,3 до 5 В, что особенно важно, когда ведущий контроллер работает на пониженном напряжении, возможность задавать яркость сегментов индикатора. Автор решил подвести модуль под свои нужды и повысить удобство считывания цифровой информации с дисплея; он обращает внимание читателя на функцию DisplaySpecialChar(), которая даёт возможность по битовой маске выводить какой-либо пользовательский символ на семисегментном индикаторе. Внешний вид устройства с отображением параметров в совокупности с пользовательскими символами изображён на рисунке 1.1



Рисунок 1.1 – Внешний вид Контроллера управления вытяжным вентилятором с WiFi. Отображение на дисплее величины влажности и знака «%». Отображение на дисплее величины температуры и знака «°»

Ещё один простой проект [7] доказывает, что взаимодействовать с модулем CATALEX (именно такое название указано на обратной стороне платы) может и микроконтроллер ESP8266 (рисунок 1.2). На дисплее отображаются часы и минуты. Время синхронизируется с NTP сервером.

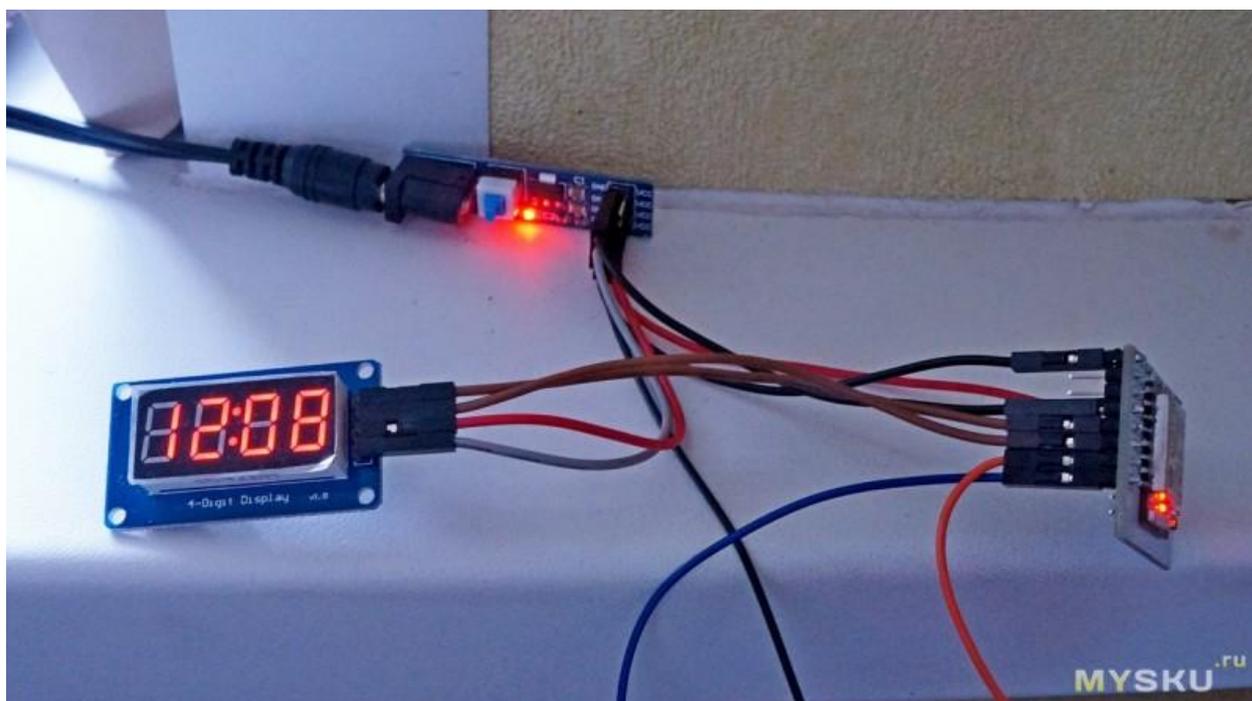


Рисунок 1.2 – Пример работы микроконтроллера ESP8266 и дисплея CATALEX.

Рассмотрим ещё проект, объединяющий функции не только отображения данных на дисплее, но и их озвучивания. В опубликованном обзоре модуля [8] рассматривается создание «говорящих» часов при использовании выше рассмотренного модуля дисплея и MP3 модуля DFPlayer Mini. Здесь приведены схематехника (весьма простая) и написание в среде Arduino программы для озвучивания текущего времени и информации с датчика освещённости, а также подпрограммы для корректного лексически правильного произношения чисел и единиц измерения. Необходимо отметить, что все используемые фразы заранее записаны на SD карту, и «универсальному голосовому синтезатору чисел» нужно лишь в определённой последовательности воспроизводить их, следуя правилам русского языка. Внешний вид системы представлен на рисунке 1.3.

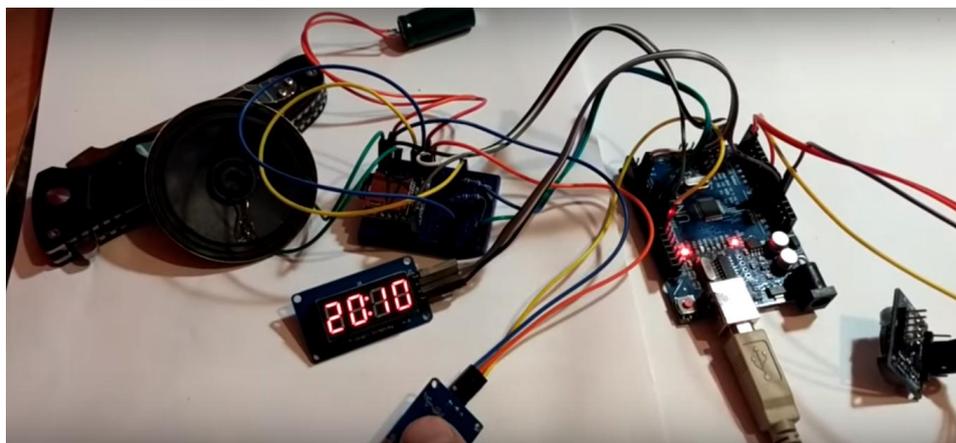


Рисунок 1.3 – Часы с функцией озвучивания времени

Теперь перейдём к функции управления жалюзи. Анализ показывает, что существует достаточно много решений промышленного производства. Они включают датчики освещённости, таймеры, даже датчики разбития окна. Могут быть запрограммированы на любые графики и расписания дня жителей.

Рассмотрим популярные системы.

Компания ТРИА предлагает интегрировать электроприводы в систему «Умный дом». Это позволяет удалённо контролировать жалюзи и другие устройства, имеющие электропривод для закрывания или открывания чего-либо.

Это делает жизнь в доме за городом, квартире, или коттедже не только удобной и комфортной.

Жалюзи могут открываться или закрываться либо отдельно, либо группами, например, только в одной комнате или во всей квартире. Это делает быт в большом доме легче.

Сенсорная панель позволяет закрыть все жалюзи в доме одним прикосновением, что избавляет жителей тратить энергию на рутинные дела.

На рисунках 1.4-1.6 представлены некоторые элементы интерфейса управления: сенсорные экраны с меню, клавишный выключатель.



Рисунок 1.4 – Внешний вид сенсорной панели, предлагаемой к установке компанией ТРИА



Рисунок 1.5 – Возможность управления жалюзи в системе



Рисунок 1.6 – В системе имеется возможность клавишного управления жалюзи

Не только ручным управлением ограничились разработчики: с помощью датчиков дождя и освещенности (рисунок 1.7) система может самостоятельно принимать решения.

Восход и сумерки заставляют жалюзи открываться и закрываться автоматически.



Рисунок 1.7 – Использование датчика света в автоматическом режиме управления жалюзи

Предлагается также управлять жалюзи и шторами по непростым алгоритмам, которые учитывают распорядок дня жильцов.

Устройство сочетает в себе современное оборудование зарубежных производителей, например, Creston, AMX и Control4, и контроллеры отечественного производства: главный управляющий контроллер Master Controller и контроллер данных (их ввода-вывода) CP-32 (рисунок 1.8).



Рисунок 1.8 – Применение контроллеров управления AMX, Creston и Control4 как оборудования СУ различными электроприводами и другими устройствами

Контроллеры, разработанные в компании ТРИА успешно выполняют многие функции управления разными устройствами, которые имеют электропривод. По сравнению с аналогами, выпускаемыми за рубежом, цены на их продукцию ниже. [9]

Стоит отметить лидера-разработчика подобных систем Gira. Одна из широкой линейки встраиваемых систем – Gira System 3000

Одним нажатием пользователь может сохранить удобное время для поднятия и опускания жалюзи в ежедневном режиме. Существует возможность выставления графика для каждого дня недели (рисунок 1.9).



Рисунок 1.9 – Внешний вид элементов управления и настройки системы Gira

Первые лучи солнца могут благотворно сказаться на непринуждённом пробуждении – эту систему можно настроить и на такие задачи, как динамичное изменение положения ламелей (пластин) в определённое время суток. [10]

Особое внимание хотелось бы уделить автору, опубликовавшему на сайте Geektimes статью «Автоматические жалюзи на Arduino» [11]. Многие непростые задачи схожи с теми, что решаются в настоящей работе. Автор задумывал создать систему, реагирующую на наружное естественное освещение и меняющую угол ламелей жалюзи. Кроме того, предполагалось использовать пульт дистанционного управления для ручного управления.

Из-за того, что стандартные горизонтальные пластинчатые жалюзи имеют в механизме поворота ламелей червячный вал, передающий усилие, или движение, только в одну сторону, пришлось его демонтировать, чтобы иметь возможность соединить механически оси двигателя и планки, на которую крепятся ламели.

Автор использовал небольшой двигатель постоянного тока с рабочим напряжением 5 В, драйвер L293D для коммутации обмоток, а в качестве обратной связи выступал потенциометр, установленный с двигателем на одну ось и впоследствии подключённый к АЦП микроконтроллера. Также в проекте применялся ИК приёмник TSOP31236 для дистанционного управления с пульта. На рисунке 1.10 представлена фотография механики конструкции.

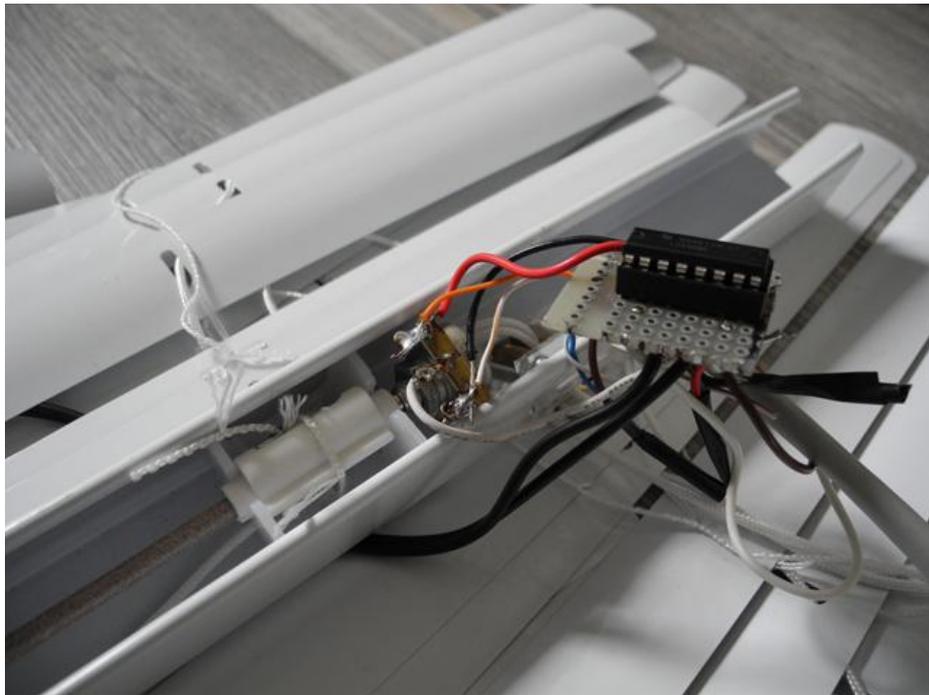


Рисунок 1.10 – Внешний вид конструкции

В роли датчика света выступал солнечный элемент из старого фонарика. Схема электрических соединений представлена на рисунке 1.11.

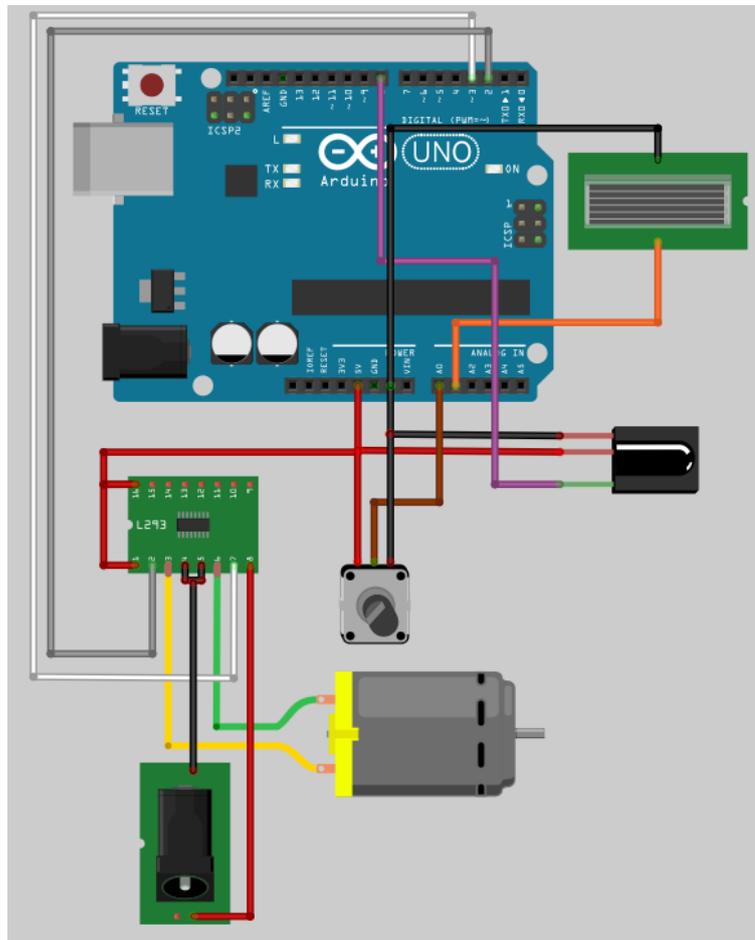


Рисунок 1.11 – Схема электрических соединений

## 2. Схемотехническая часть

### 1.3 Общие принципы построения системы

На рисунке 2.1 приведена функциональная схема, отображающая структуру и связи отдельных узлов разрабатываемой модели будильника.

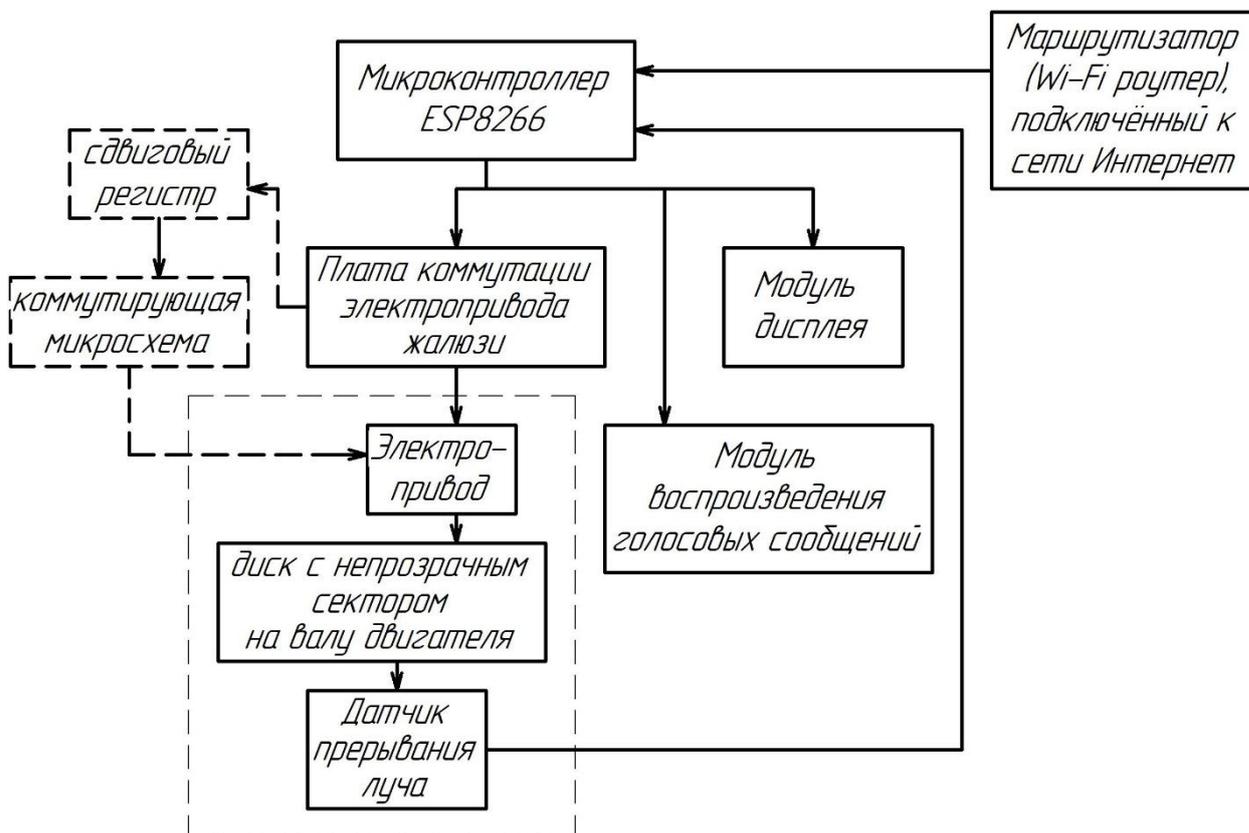


Рисунок 2.1 – Функциональная схема устройства

Согласно рисунку выше, основой системы, выполняющей функции обработчика информации, служит микроконтроллер ESP8266 версии 07. Модули на основе этого чипа имеют множество разных исполнений, многие выполняются в виде SMD платы, которая может быть механически смонтирована на основной печатной плате способом, пригодным для обычных SMD компонентов. Шаг контактных площадок у данной версии модуля отличается от распространённого «DIP», поэтому для электрического соединения с макетной печатной платой, на которой собран прототип устройства, был применён адаптер, осуществляющий переход от нестандартного шага контактов к стандартному (2,54 мм). Внешний вид

переходника с установленным на нём модулем ESP8266-07 показан на рисунке 2.2.

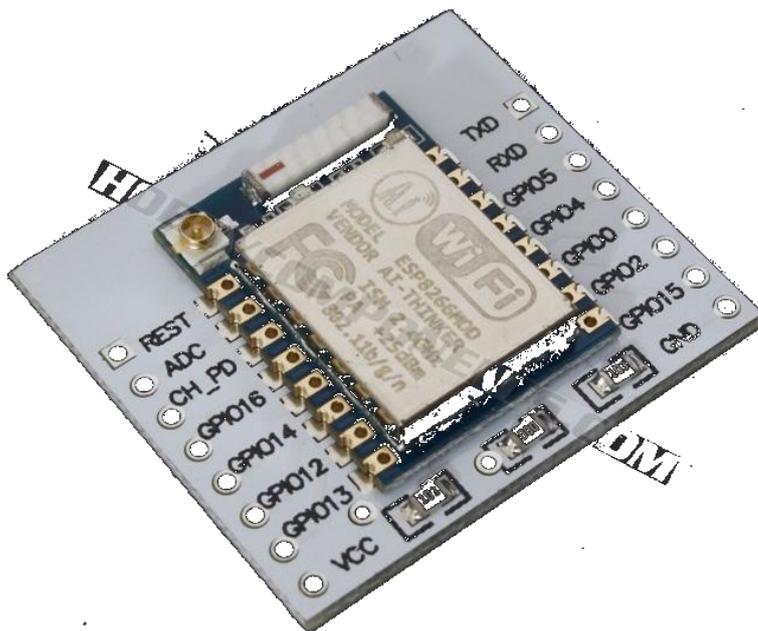


Рисунок 2.2 – Внешний вид адаптера с припаянным модулем микроконтроллера

Итак, центральный микроконтроллер управляет тремя модулями, выполненными в виде отдельных печатных узлов и имеющих по краям платы выводы для подключения. Подобные модули очень удобно фиксировать в слоты (или, как их ещё называют, сокеты – от англ. socket). Именно таким способом закреплены на главной макетной плате будильника *модуль воспроизведения голосовых сообщений*, собственно адаптер с *Wi-Fi модулем*, а также модуль стабилизатора напряжения.

Последний узел, не показанный на схеме (рисунок 2.1), будучи подключённым к источнику постоянного напряжения (5 В), осуществляет питание всех модулей, за исключением силовой части микросхемы L293D, расположенной на *плате коммутации электропривода жалюзи*. Повышенное напряжение (12 В) подаётся на соответствующие выводы микросхемы посредством клеммной колодки, установленной на плате, от другого, более мощного источника питания. Модуль стабилизатора представляет собой законченный печатный узел с входными и выходными двухпиновыми

разъёмами PLS (шаг 2,54 мм), основным элементом которого является линейный стабилизатор напряжения AMS1117-3.3. [12] Он имеет небольшую обвязку в виде конденсаторов и резисторов, построенную на основе оригинальной схемы из документации производителя. Плата оснащена светодиодом для индикации корректной работы преобразователя. Ниже следуют основные характеристики данного модуля.

- Входное напряжение: от 4,6 В до 12 В
- Максимальный ток: 750 мА
- Выходное напряжение: минимально – 3,234 В, максимально – 3,366 В при токе от 10 мА до 1 А

Внешний вид модуля показан на рисунке 2.3.

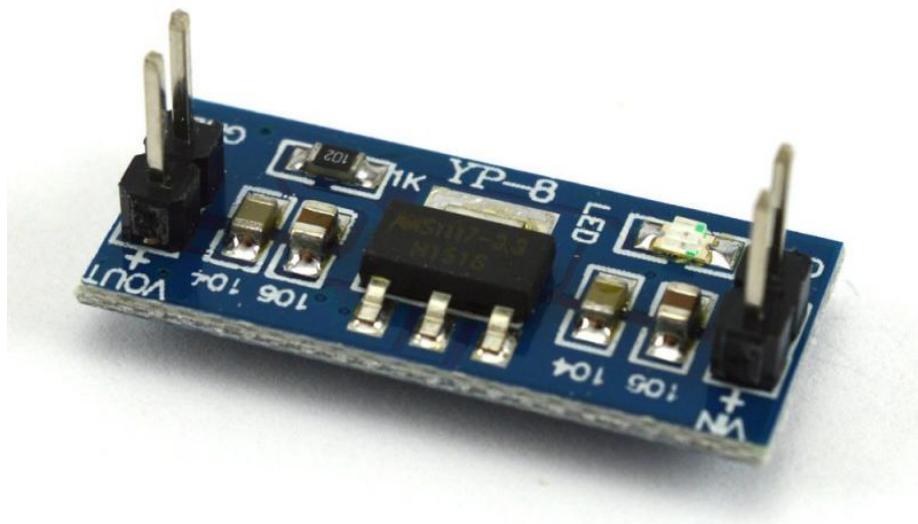


Рисунок 2.3 – Внешний вид модуля стабилизатора напряжения.

Плата коммутации электродвигателя Motor Shield (рисунок 2.4) конструктивно выполнена в виде платы расширения для платформы Arduino и механически прикреплена к одной из стенок корпуса винтами. Разработчики позаботились об удобстве её крепления и предусмотрели крепёжные отверстия.

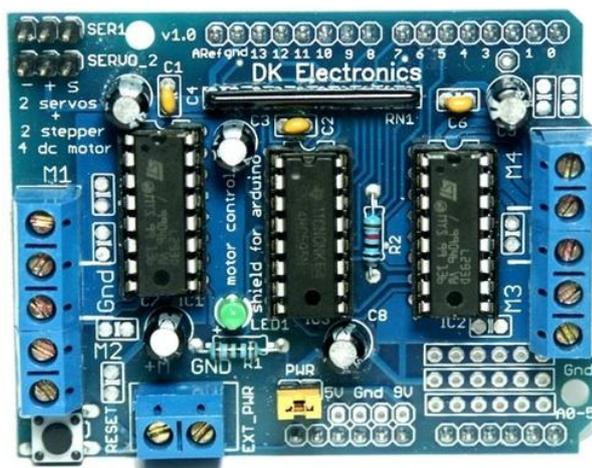


Рисунок 2.4 – Внешний вид платы расширения MotorShield

На данный момент раскрыта только небольшая часть большого потенциала данной платы, поскольку здесь имеется возможность управлять четырьмя моторами постоянного тока, регулировать скорость вращения двигателей, подключать напрямую к трёхпиновым разъёмам и управлять двумя сервоприводами.

Именно эта плата выбрана в качестве резерва для дальнейшего расширения, развития проекта.

Управление вышеупомянутыми силовыми микросхемами L293D осуществляется с помощью сдвигового регистра 74HC595. [13] Благодаря ему удаётся значительно сократить число используемых портов микроконтроллера до трёх. В нашем случае используются контакты, обозначенные на плате числами 4, 8 и 12. Назначение этих контактов известно из теории электронных промышленных устройств, – это линии:

- данных (data)
- тактовых импульсов (clock)
- разрешающего сигнала (latch).

К клеммной колодке, обозначенной на плате как M2, присоединены с помощью подводящих проводов клеммы электродвигателя.

Связь коммутирующей платы с главной, на которой установлен микроконтроллер, а также подвод питания (3,3 В для логической части),

осуществляются с помощью шлейфа из пяти проводов: 2 провода – питание и 3 провода – последовательный интерфейс.

#### 1.4 Узел обработки информации

Микроконтроллер ESP8266 не только очень требователен к напряжению питания, но и нуждается в подтяжке нескольких портов через резисторы определённого номинала к полюсам источника питания. Схемотехническое решение взято из рекомендаций [14]. На рисунке 2.5 приведена схема обвязки модуля для стабильной работы в течение продолжительного времени.

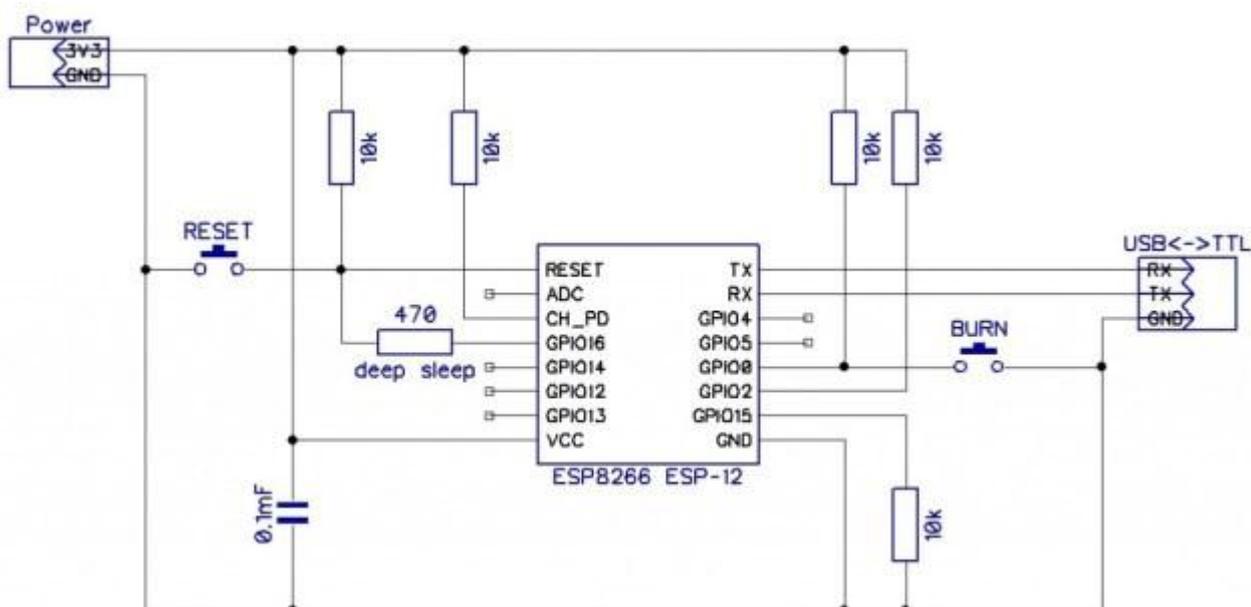


Рисунок 2.5 – Схема электрических соединений для подключения микроконтроллера ESP8266, обеспечивающая высокую стабильность в работе

Адаптер, о котором говорилось выше, уже содержит на себе подтягивающие резисторы, необходимые для успешного старта модуля. Это SMD резисторы, соответствующие приведённым на рисунке 2.5:

- Резистор номиналом 10 кОм, соединяющий контакт GPIO15 с общей точкой схемы
- Резистор номиналом 10 кОм, соединяющий контакт CH\_PD (CHIP ENABLE) с положительным полюсом источника питания (выпаян, вместо него используется резистор тем же номиналом,

расположенный на макетной плате, с последовательно включённым двухконтактным разъёмом в виде штырей. Разрыв цепи пропадает при установке джампера (перемычки) на разъём.

С учётом особенности адаптера (платы-переходника) были внесены поправки к разработке электрических соединений на макетной печатной плате. В частности, игнорировался первый из вышеуказанных резисторов. Остальные компоненты монтировались без изменений номиналов согласно рисунку 2.5.

### 1.5 Описание микроконтроллера ESP8266

Теперь рассмотрим более подробно особенности использования управляющего модуля и его технические характеристики.

Разрабатываемый будильник основан на модуле системы «Wi-Fi System-on-Chip» (SoC) под названием ESP8266, представляющем собой платформу, к которой достаточно удобна для прототипирования. [15] Плата показана на рисунке 2.6.

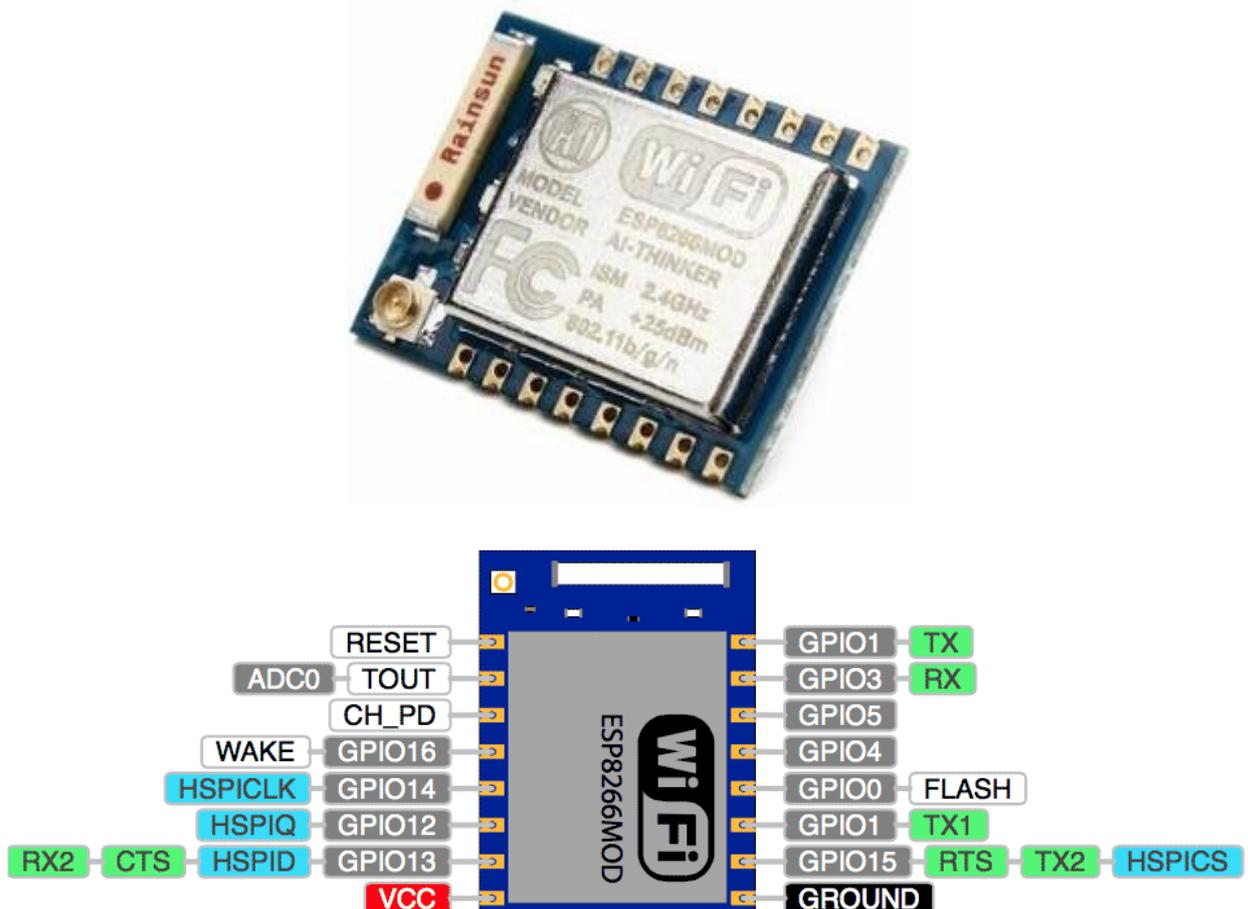


Рисунок 2.6 – Внешний вид радиомодуля Wi-Fi ESP8266 и его описание Выводов.

Это трансивер a2.4GHzWi-Fi, созданный Espressif Systems, со встроенным радиочастотным передатчиком, а также полностью программируемым 32-битным микроконтроллером, регистром для управления портами ввода / вывода и последовательным интерфейсом по протоколам связи SPI / I2C. Кроме того, возможность запуска загруженной прошивки и наличие управляемых цифровых выводов GPIO и аналогового порта АЦП позволяет связать ее с периферийными устройствами или внешними устройствами различной типологии (в рассматриваемом случае датчиками и модулями с последовательными протоколами связи). С помощью модуля ESP8266 может быть добавлен беспроводной доступ в Интернет в любое устройство на базе микроконтроллера с простой связью через доступный интерфейс UART. Имеется 9 контактов GPIO, каждому из которых может быть назначена функция микропрограммного обеспечения (например, вход прерывания), возможно использование внутреннего подтягивающего (к положительному полюсу/ общей точке) резистора. На рисунке 2.7 показана блок-схема модуля ESP8266 SoC Wi-Fi.

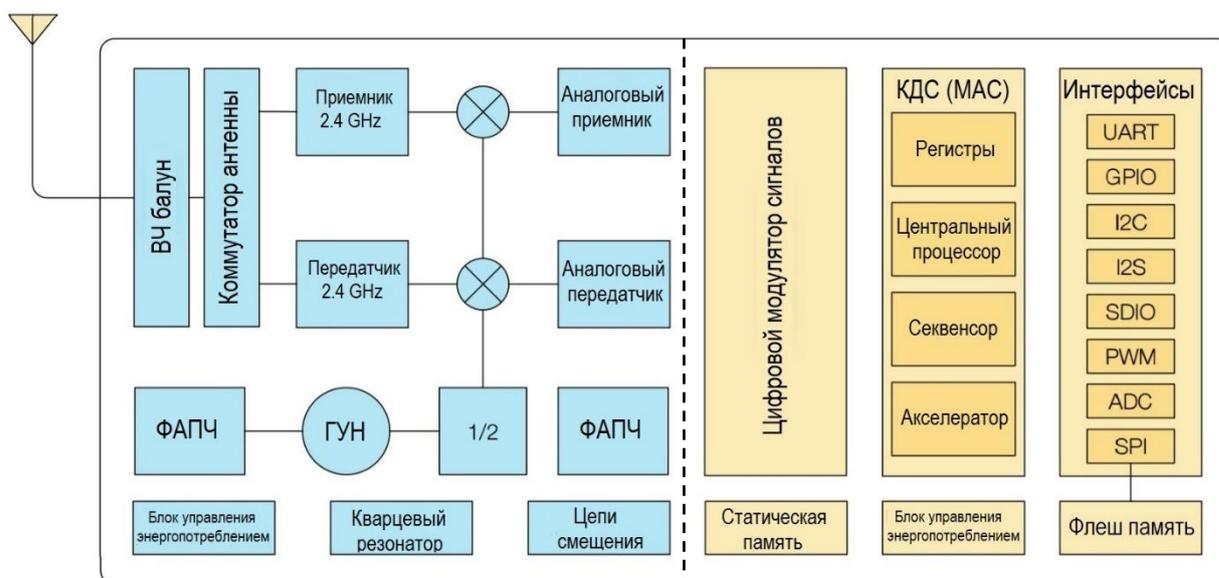


Рисунок 2.7 – Блок-схема ESP8266 с основными внутренними функциональными блоками.

Аналоговая часть схемы (рисунок 2.7, слева):

- ВЧ Балун — служит для согласования высокочастотных (ВЧ) каскадов ESP8266EX с антенной (от англ. balun — balanced-unbalanced)
- ФАПЧ — фазовая автоподстройка частоты (англ. PLL – phase-locked loop or phase lock loop)
- ГУН — генератор, управляемый напряжением (англ. VCO — voltage-controlled oscillator)
- Блок управления энергопотреблением (англ. PMU – power management unit)
- Цепи смещения — служат для определения производительности усилителя высокой частоты при изменении температуры

Цифровая часть схемы (рисунок 2.7, справа):

- Цифровой модулятор сигналов
- SRAM — статическая память с произвольным доступом (англ. SRAM) — static random access memory
- КДС — Контроллер доступа к среде (англ. MAC ) — Media Access Controller
- Флеш-память — (англ. flash memory)
- Секвенсор — аппаратное или программное устройство для записи и считывания данных в реальном времени через буфер оперативной памяти из флеш-памяти (от англ. *sequence* — «последовательность»)
- Интерфейсы ввода-вывода общего назначения (GPIO) [16]

Напряжение питания чипа ESP8266 – 3,3 В с максимально допустимым значением 3,6 В; каждый цифровой вход снабжен схемой защиты от перенапряжения, чтобы обеспечить интегральную схему от входных напряжений, превышающих 6 В. Наконец, чип позволяет использовать все самые современные протоколы шифрования, а также WEP (RC4), CCMP

(CBC-MAC, режим подсчета), TKIP (MIC, RC4) или WAPI (SMS4), WEP (RC4), CRC, WPA, WPA2 e WPS.

Встроенное программное обеспечение написано на C / C ++ и координирует входные и выходные устройства, сопряженные с контроллером ESP8266. Оно также запускает встроенные мини-серверы и веб-серверы для поддержки автономных операций с использованием двухрежимного (точка доступа и клиент) бортового Wi-Fi-модуля. Это должно расширить систему в случае прерывания соединения с облачными службами и обеспечить независимое использование внутри дома. [17] Возможности непосредственной обработки и хранения информации, имеющиеся у ESP8266, позволяют интегрировать его с датчиками и другими устройствами для конкретных применений с помощью собственных контактов GPIO с минимальными необходимыми подключениями. Благодаря высокой степени интеграции, включая Wi-Fi антенну с коммутатором balundevise, лишь используя преобразователи питания и USB-TTL конвертор для программирования, он требует минимальной внешней схемы, так что вся обвязка занимает минимальную область печатной платы.

Текущее энергопотребление ESP8266 в активном режиме достигает максимума 215 мА при максимальной мощности передачи по протоколу 802.11b (скорость передачи данных 11 Мбит / с) и снижается до 145 мА по протоколу 802.11g (скорость передачи данных равна 54 Мбит / с) [18].

#### Особенности ESP8266 [19]

- Поддержка интерфейса 802.11 b / g / n
- Интегрированный 32-разрядный микроконтроллер с низким энергопотреблением
- Встроенный 10-битный аналого-цифровой преобразователь
- Встроенный полный стек TCP / IP
- Поддержка станций, точек софт-доступа, станций + Мягкая точка доступа

- Встроенный переключатель TR, балун, LNA, усилитель мощности и соответствующая сеть
- Наличие внутренних PLL, регуляторов и блоков питания
- WiFi 2,4 ГГц с поддержкой WPA / WPA2
- Поддержка WiFi Direct
- Функция Smart Link поддерживает как устройства Android, так и iOS
- SDIO 2.0, (H) SPI, UART, I2C, I2S, ИК-пульт дистанционного управления, PWM, GPIO
- STBC, 1 × 1 MIMO, 2 × 1 MIMO
- Агрегация A-MPDU и A-MSDU и защитный интервал 0,4 с
- Выход из сна для передачи пакетов менее чем за 2 мс
- Выходная мощность до +20 дБм в режиме 802.11b
- Потребляемая мощность в режиме ожидания меньше 1 мВт.
- Мощность глубокого сна меньше 10 мкА, а ток утечки питания меньше 5 мкА
- Сертифицировано FCC, CE, TELEC, Альянсом WiFi и SRRC

#### Технические характеристики ESP8266 [19]

- Рабочее напряжение: 3,0 ~ 3,6 В
- Средний рабочий ток: 80 мА
- Tensilica Xtensa LX106 32-битный RISC-процессор, работающий на частоте 80 МГц
- 16 контактов GPIO
- SPI
- I2C
- I2S
- UART - 2x TX и 1x RX
- 1x 10-битный АЦП
- Рабочая температура: -40 °С ~ 125 °С

- Диапазон частот: 2400 ~ 2483,5 МГц

## 1.6 Дисплей для отображения текущих параметров погоды и времени

Для контроля работы будильника, а также для отображения числовых значений отслеживаемых параметров и кратких словесных сообщений, в данном проекте применяется модуль дисплея CATALEX (о нём упоминалось в первой главе), который построен на следующих основных компонентах:

- Семисегментный индикатор 3642ВН с высотой символов 0,36". Цвет излучения сегментов: красный.
- Микросхема ТМ1637 для реализации автономного динамического отображения символов на семисегментном индикаторе. Цоколёвка данной микросхемы показана на рисунке 2.7.

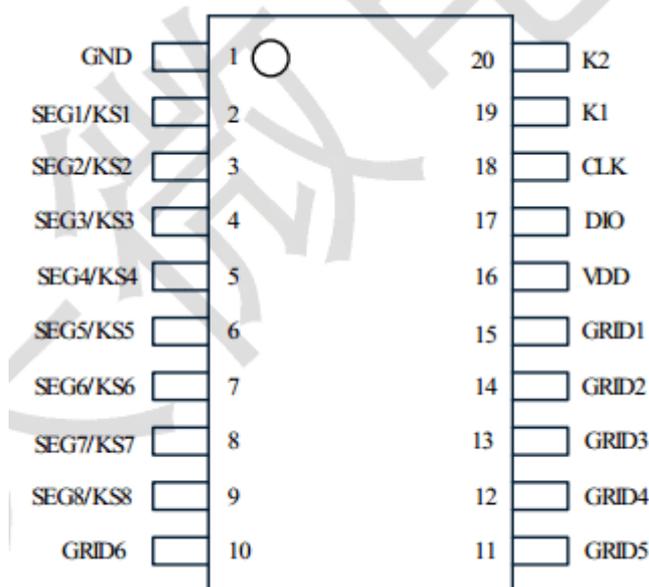


Рисунок 2.8 – Цоколёвка микросхемы ТМ1637

Эта микросхема была создана в Китае для работы в DVD плеерах для вывода времени, также она поддерживает считывание нажатий клавиш на матричной клавиатуре (до 16 кнопок), например, в микроволновых печах.

Особенности данного модуля [20]:

- Использование CMOS-процесса

- Режим отображения (6 знакомест по 8 сегментов в каждом), поддерживает индикаторы с общим анодом.
- Сканирование кнопок (8 × 2 бит), усовершенствованная схема распознавания кнопок с антиблокировкой
- Схема регулировки яркости (регулировка времени засветки символа)
- Двухпроводный последовательный интерфейс (CLK, DIO)
- Режим автономной генерации тактового сигнала для динамической индикации обеспечивается встроенным RC-генератором (450 кГц + 5%)
- Встроенная схема сброса питания
- Встроенная схема автоматического гашения
- Тип корпуса: DIP20 / SOP20

Внешний вид модуля дисплея показан на рисунке 2.9.



Рисунок 2.9 – Внешний вид модуля CATALEX

Связь модуля с контроллером ESP8266, а также подвод питания осуществляются посредством шлейфа из четырёх проводов, соединяющего контакты разъёма на модуле и гнезд на главной плате.



Итак, возвратимся к техническому аспекту разработки устройства. Здесь используется модуль DFPlayer mini, внешний вид которого изображена на рисунке 2.10.

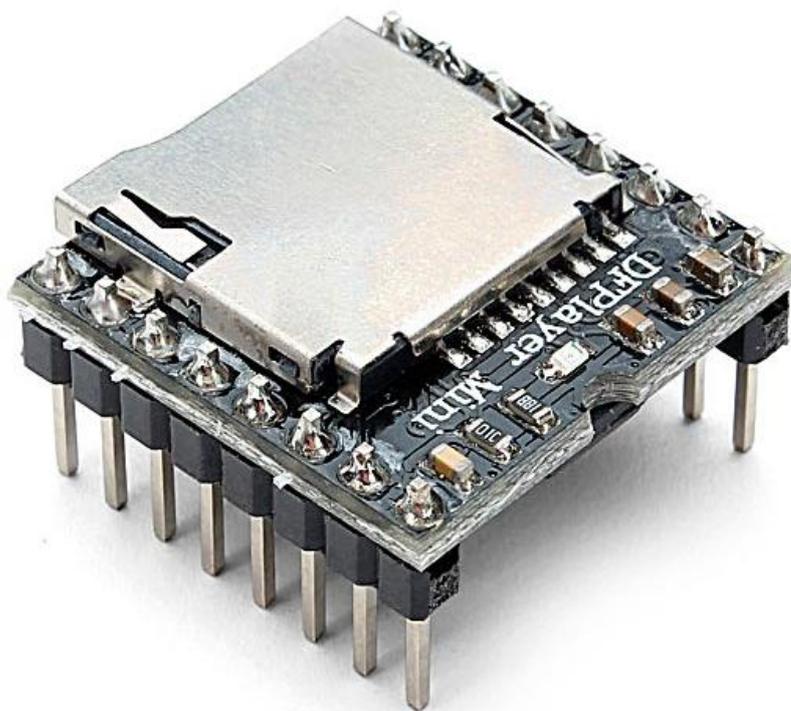


Рисунок 2.10 – Внешний вид модуля DFPlayer mini

Мини-MP3-плеер DFPlayer для Arduino - это небольшой и недорогой MP3-модуль с упрощенным выходом непосредственно на динамик. Модуль может использоваться как автономный модуль с подключенной батареей, динамиком и кнопками или использоваться в сочетании с микроконтроллером, поддерживающим интерфейс UART (линии RX / TX).

Особенности модуля [21]:

- поддерживаемые форматы звуковых файлов: mp3 с битрейтом до 320kbps, wav.
- поддерживаемые частоты дискретизации (кГц): 8 / 11,025 / 12/16 / 22.05 / 24/32 / 44.1 / 48
- 24-битный ЦАП, поддержка динамического диапазона 90 дБ, поддержка SNR 85 дБ

- полностью поддерживает файловую систему FAT16, FAT32, максимальную поддержку 32G TF-карты, поддержку 32G U-диска, 64M байт NORFLASH
- различные режимы управления, режим управления цифровыми кнопками (предыдущий/следующий трек, громкость выше/ниже), последовательный режим (UART), режим управления кнопками аналоговой клавиатуры.
- функция приоритета трека
- аудиофайлы могут быть отсортированы по папкам; поддерживается до 100 папок, каждая папка может содержать до 255 композиций
- регулировка громкости, 6-уровневый эквалайзер

Цоколёвка модуля воспроизведения звуковых файлов приведена на рисунке 2.11.

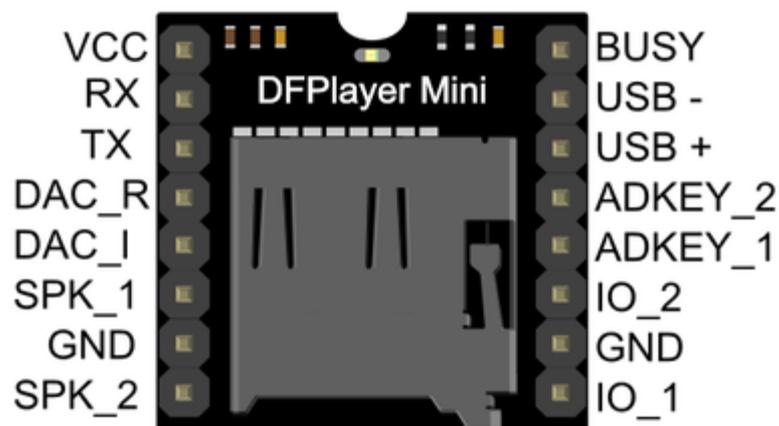


Рисунок 2.11 – Схема расположения выводов модуля DFPlayer mini

Типичная схема подключения плеера к микроконтроллеру по интерфейсу UART показана на рисунке 2.12. Заметим, что резистор здесь используется в качестве ограничительного, так как высокий логический уровень для контроллера DFPlayer mini равен пяти вольтам. В случае использования ESP8266, который оперирует более низким напряжением, необходимость использования резистора отпадает.

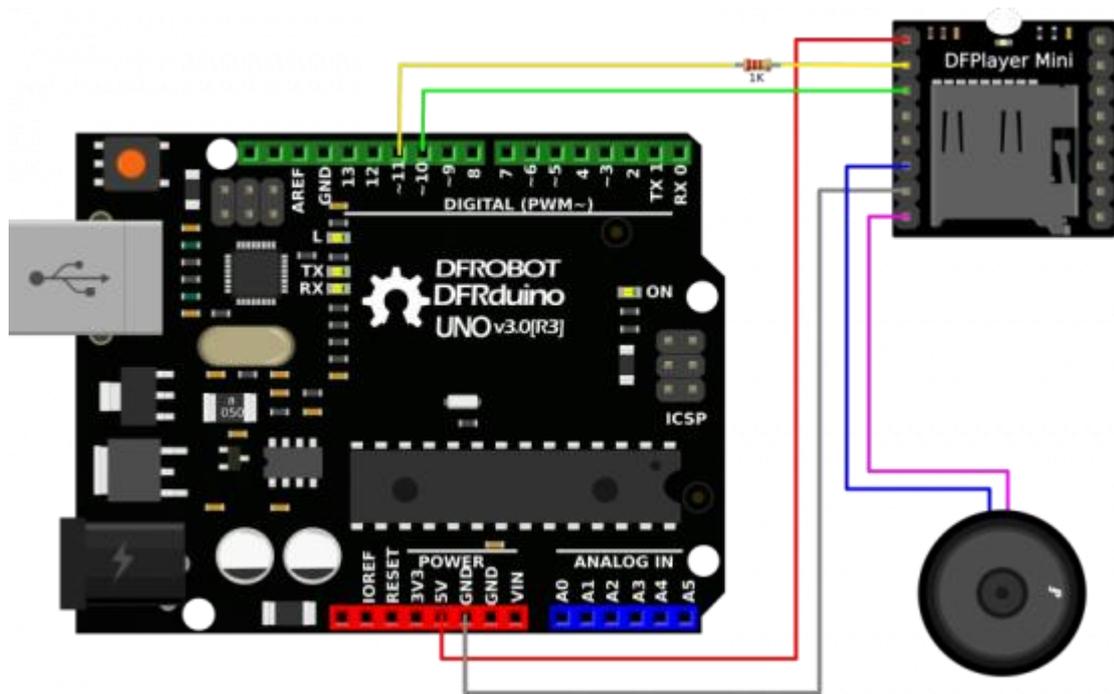


Рисунок 2.12 – Типичная схема подключения модуля плеера к микроконтроллеру

Ниже представлена таблица 1, в которой расшифровываются обозначения выводов модуля, используемых в проекте.

Таблица 1 – Назначение выводов модуля DFPlayer mini, используемых в настоящем проекте.

Вывод	Описание
VCC	Питание «+»
GND	Питание «-»
RX	UART приём
SPK1	Громкоговоритель «+»
SPK2	Громкоговоритель «-»
BUSY	Индикатор состояния («0» — ожидание, «1» — проигрывание)

Рассматриваемый модуль устанавливается в сокет, распаянный на плате. Связь с контроллером ESP8266, а также подвод питания (3,3 В) происходит с помощью проводов, соединяющих контакты сокетов на макетной печатной плате.

## 1.8 Датчик прерывания луча

Датчик прерывания используется для контроля и регистрации конечного положения ламелей жалюзи. У такого сенсора есть ещё несколько названий: щелевой фотопрерыватель, датчик скорости, датчик оборотов;

цифробуквенное название FC-03. Внешний вид модуля датчика приведён на рисунке 2.13.

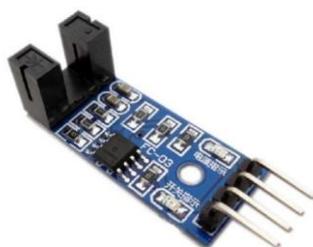


Рисунок 2.13 – Внешний вид модуля датчика прерывания

Как и на многих модулях, на данном выполнена обвязка контактов самого датчика, а также установлена микросхема двойного компаратора LM393, осуществляющая сравнение выходного аналогового сигнала фототранзистора с некоторым заданным значением. Напротив фотоприёмника расположен инфракрасный светодиод, который и является источником излучения. Корпус датчика сконструирован таким образом, что обеспечивает встречно-параллельное положение светодиода и фототранзистора, и одновременно позволяет поместить некий объект между ними.

Модуль выдаёт высокий или низкий уровень напряжения, в зависимости от того, прерван луч или нет. [22] Напряжение питания составляет от 3,3 до 5 В. Модуль оснащён SMD светодиодами, осуществляющими индикацию подачи питания и факт беспрепятственного прохождения луча.

Схема электрических соединений модуля представлена на рисунке 2.14.

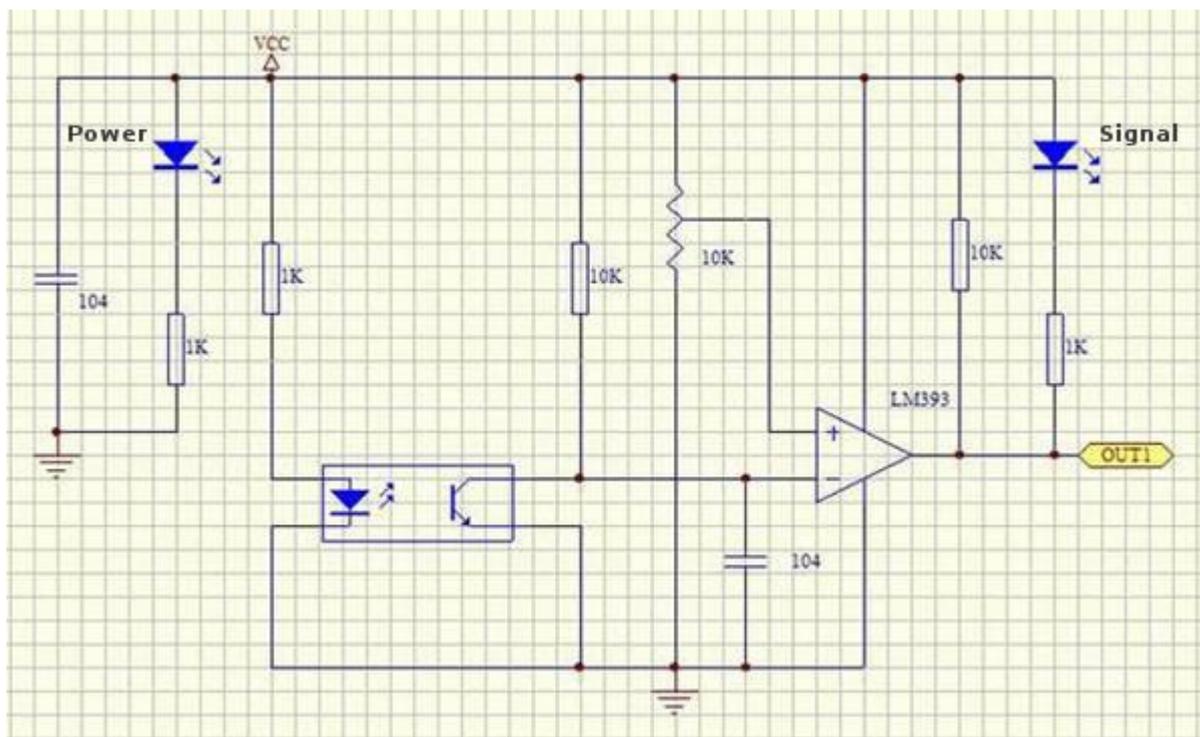


Рисунок 2.14 – Схема электрических соединений модуля FC-03

## 1.9 Назначение портов микроконтроллера ESP8266

Составим сводную таблицу задействованных портов микроконтроллера и назначенных им цифровых выводов описанных выше модулей (таблица 2).

Таблица 2 – Таблица соединений

№	Название модуля/датчика	Название вывода	Обозначение порта МК
1	Модуль воспроизведения звуковых файлов DFPlayer mini	RX (UART приём)	GPIO5
		BUSY (Сигнал «занят / свободен»)	GPIO16
2	Плата коммутации электропривода MotorShield	data (данные) <b>DS</b>	GPIO14
		latch (разрешающий сигнал) <b>STCP</b>	GPIO12
		clock (такт. импульсы) <b>SHCP</b>	GPIO13
3	Дисплей TM1637	DIO (данные)	GPIO0
		CLK (такт. импульсы)	GPIO2
4	Датчик прерывания луча	DO (цифровой выход)	GPIO3

Напряжение питания всех перечисленных в таблице 2 модулей одинаково, и составляет 3,3 В.

На рисунке 2.15 представлена полная принципиальная схема устройства.

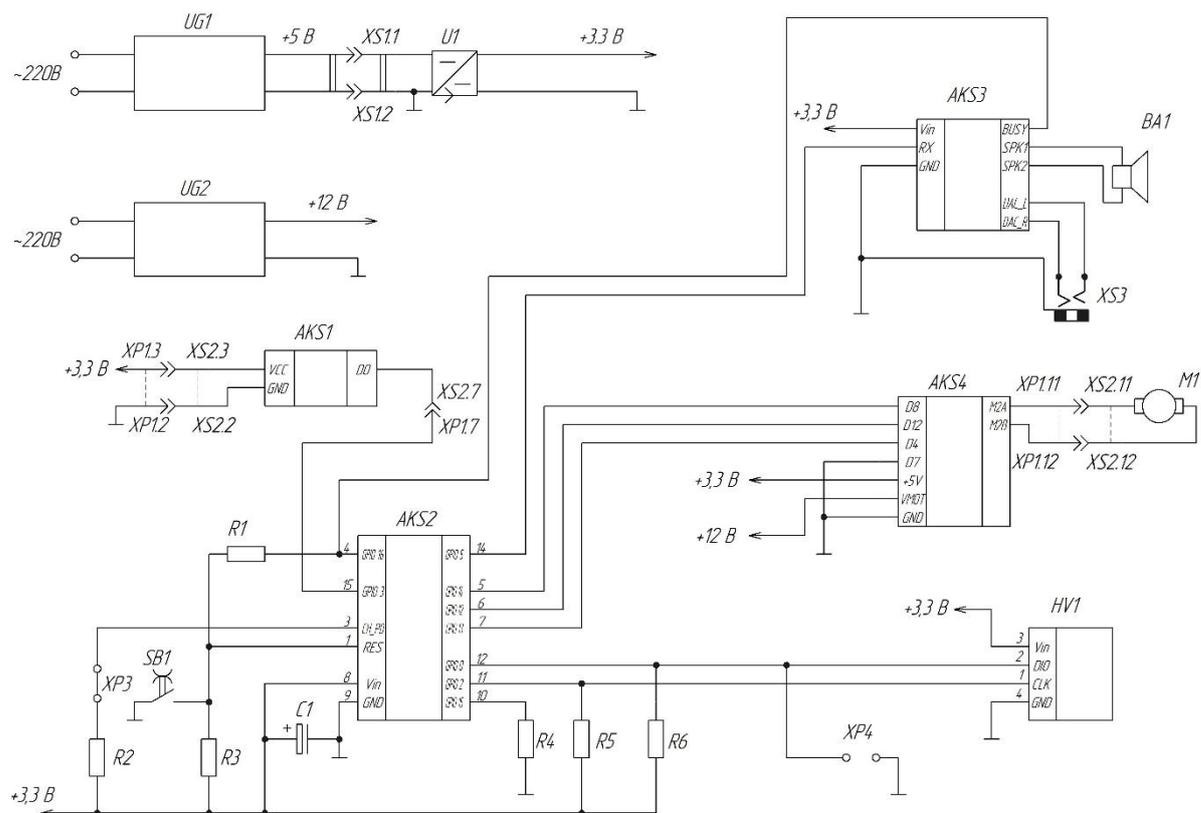


Рисунок 2.15 – Полная принципиальная схема устройства

### 3. Программирование в среде Arduino IDE

Программировать контроллер ESP8266 можно на нескольких языках. Среди них Forth, Lua, C++. Именно на СИ-подобном языке и выполнялось программирование, потому как в популярной среде Arduino IDE с множеством библиотек и примеров, немного затрагиваемой в курсах «Электроизмерительные приборы и датчики информации», «Информационная электроника» опытные программисты создали надстройку, позволяющую производить написание программных кодов, их компиляцию и загрузку скомпилированного бинарного файла в ESP8266 (с помощью USB-TTL конвертора), не покидая среду разработки.

Следует обратить внимание на структуру окна среды программирования. Как и в окнах многих других программ, вверху

располагается меню, ниже следует панель инструментов, доступ к функциям которых доступен также из пунктов меню. Далее следует панель вкладок. На рисунке 3.1 вкладка всего одна, но в настоящем проекте их насчитывается **более десяти**. Это разбиение служит для улучшения восприятия скетча. Таким же образом можно подключать библиотеки (файлы с расширением .h и .cpp), написанные пользователем и к которым необходимо постоянно обращаться и корректировать. На этом же уровне справа находится кнопка вызова меню управления вкладками, в котором представлены функции «Новая вкладка», «Переименовать», «Удалить», «Предыдущая вкладка», «Следующая вкладка». Здесь же перечислены названия всех вкладок, на случай, если пропадёт их видимость вследствие ограниченных размеров окна.

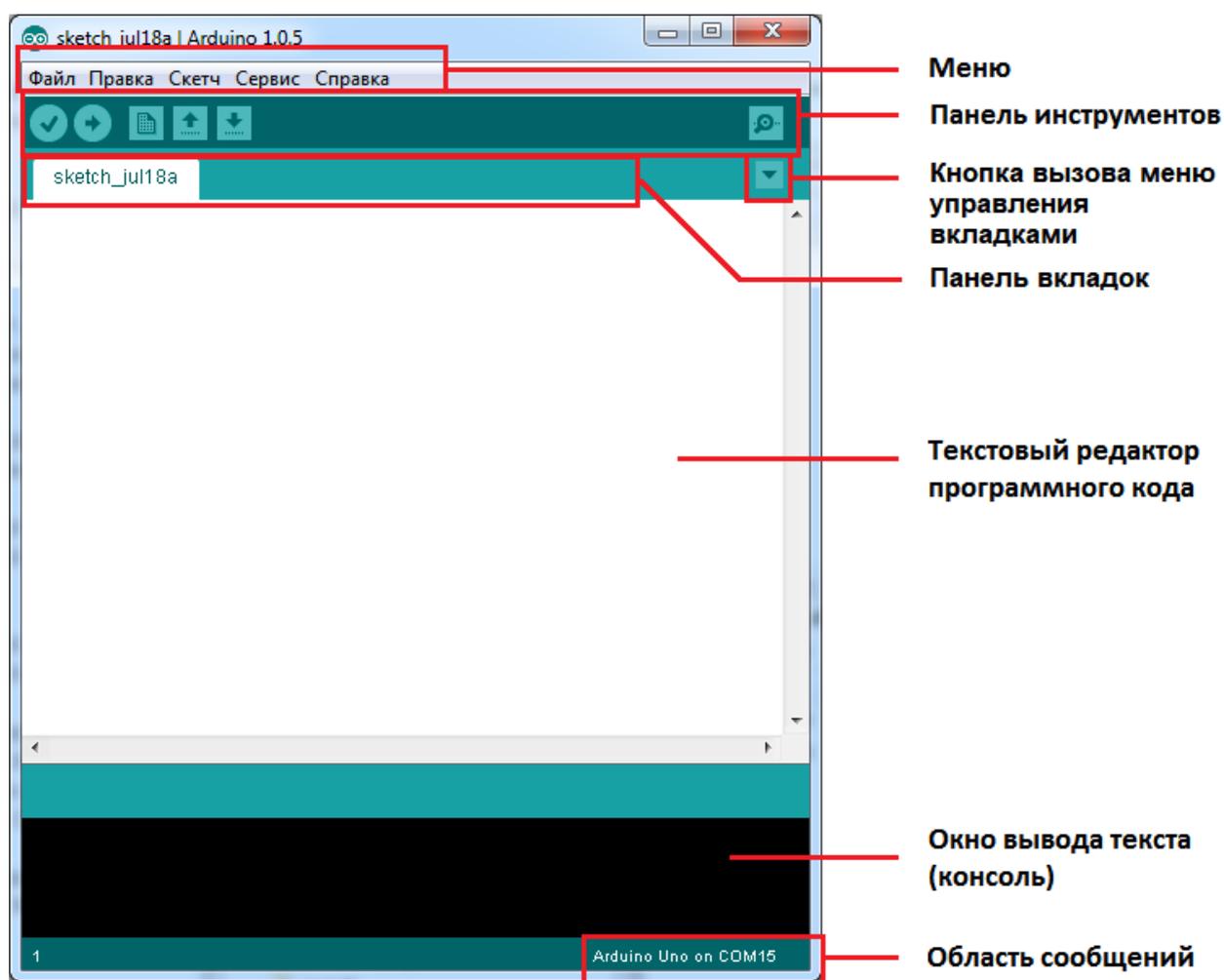


Рисунок 3.1 – Внешний вид окна редактора программного кода (Arduino IDE)

Затем следует поле для ввода текста программного кода, ниже располагается консоль для вывода сообщений о результатах компиляции скетчей. Внизу окна (справа) выводится информация о выбранной в меню плате.

Данное предисловие призвано обратить внимание на возможность создания нескольких вкладок при больших объёмах текста программного кода. Заметим, что вкладка, в которой среда обнаружила ключевые слова `void setup` и `void loop`, отображается первой в ряду вкладок, остальные расположены по имени в алфавитном порядке (*только латиница*). Название папки, в которой содержатся файлы проекта, представляющиеся в виде вкладок в среде, должно совпадать с именем первой основной вкладки.

На рисунке 3.2 изображён скриншот панели вкладок с частями программного кода для дальнейшей компиляции и загрузки на микроконтроллер. Вкладки с названием *Activ* и *SSDP* зарезервированы для дальнейшего расширения функционала устройства. Остальные вкладки используются, и код в них будет частично или полностью прокомментирован в этой главе.



Рисунок 3.2 – Вкладки с частями скетча для проекта

На следующих ниже рисунках 3.1-3.9 представлены скриншоты текста скетча. После скриншотов приведены комментарии.

```
EDWIN_1_2$ Activ DISPLAY JSON
1 // #include <Shift595.h>
2 extern "C" {
3 #include <sntp.h>
4 }
5 #include <SoftwareSerial.h>
6 #include <TM1637.h>
7 #include <DFPlayer_Mini_Mp3.h>
8 #include <ArduinoJson.h>
9
10 #include <ESP8266WiFi.h>
11 #include <WiFiClient.h>
12 #include <ESP8266WebServer.h>
13 #include <ESP8266mDNS.h>
14
15
16 #include <EEPROM.h>
17 #include "WC_EEPROM.h"
18 #include "WC_HTTP.h"
--
```

Рисунок 3.3 – Подключение библиотек

Как было сказано выше, язык Arduino очень удобен тем, что на нём написаны сотни библиотек, без которых скетчи были бы намного больше по объёму.

В данном проекте в скетч включены библиотеки (сверху вниз) для:

- работы с сервером точного времени (NTP)
- работы с программным последовательным портом (связь с модулем DFPlayer mini)
- управления модулем TM1637
- отправки команд MP3 плееру
- чтения информационных пакетов с сайта погоды OpenWeatherMap
- организации связи микроконтроллера с сетью Интернет по Wi-Fi
- забора и отправки данных на веб-сервер, который служит для установки пользовательских параметров и контроля системы
- возможности выбрать устройству имя в зоне .local
- доступа к базовым функциям по взаимодействию с энергонезависимой памятью EEPROM
- сохранения параметров, а также их чтения и записи в EEPROM

- отображения режимов работы, настройки конфигураций контроллера, а также непосредственного управления жалюзи во встроенном WEB-сервере

```
Alarm_clock$ Activ DISPLAY JSON MINUS
24
25 #define M2A 1 //M2A
26 #define M2B 4 //M2B
27
28 #define INT4pin 3//RX
29 #define MOT2pin 15//8
30 #define MOT3pin 4 //2
31
32 #define f1 1 // папка с числами
33 #define f2 2 // папка с Путиным
34 #define f3 3 // папка со смесью
35 #define f4 4 // папка с фразами
36 #define BusyState 16 // пин BUSY плеера
37 #define c19 19
38 #define c100 29
39 #define c1000 38
40 #define odna 76
41 #define dve 77
42
43 #define CLK 2 //GREEN
44 #define DIO 0 //YELLOW
45 TML637 CATALEX(CLK, DIO);
```

Рисунок 3.4 – Объявление констант

С помощью ключевого слова *#define* словам присваиваются числовые значения. Это константы для обозначения, например, номеров портов, к которым подключены те или иные датчики и модули. Их использование упрощает написание программного кода.

На строке 45 создаётся экземпляр *CATALEX* (дисплей), и указываются порты микроконтроллера для работы с ним.

```
Alarm_clock$ Activ DISPLAY JSON MINUS NTP SSDP Speaking
69 SoftwareSerial mp3Serial(SW_SERIAL_UNUSED_PIN, 5); // RX, TX/,
70 void setup() {
71   //Serial.begin(115200);
72   Serial.begin (115200, SERIAL_8N1, SERIAL_TX_ONLY);
73   Serial.printf("\n\nFree memory %d\n", ESP.getFreeHeap());
74   // Инициализация EEPROM
75   EC_begin();
76   EC_read();
77   pinMode(INT4pin, INPUT);
78   pinMode(BusyState, INPUT);
79   pinMode(dataPin, OUTPUT);
80   pinMode(latchPin, OUTPUT);
81   pinMode(clockPin, OUTPUT);
82   // Shifter.clearRegisters();
83   Serial.println("Ready");
84   //Инициализация модуля
85   CATALEX.init();
86   CATALEX.set(EC_Config.BRIGHT_DISPLAY);
87   // Подключаемся к WiFi
88   CATALEX.display(coed);
89   WiFi_begin();
90   delay(2000);
```

Рисунок 3.5 – Функция Setup

В 69-й строке создаётся экземпляр *mp3serial*. Первый аргумент в скобках показывает, что для связи с плеером задействуется только линия TX. Обратная связь от модуля будет поступать на микроконтроллер от специального вывода плеера *Busy pin* – «вывод занятости».

Далее запускается последовательная (аппаратная) передача на скорости 115200 бод (бит в секунду) – для отладки работы устройства с помощью последовательного порта (72 строка).

Определяются режимы портов (ввод/вывод) (77-81 строки). Происходит инициализация дисплея, устанавливается его яркость (85-86 строки).

В 89-й строке вызывается функция, создающая связь с Wi-Fi роутером.

```
Alarm_clock$ Activ DISPLAY JSON MINUS NTP SSDP Speaking WC_EEPROM.c
92 HTTP_begin();
93 char str[20];
94 delay (1000);
95 // Инициализация UDP клиента для NTP
96 sntp_set_timezone(EC_Config.TZ);
97 sntp_setservername(0, EC_Config.NTP_SERVER1);
98 sntp_init();|
99 delay(10);
100 Serial.println();
101 Serial.println("Setting up software serial");
102 mp3Serial.begin (9600);
103 Serial.println("Setting up mp3 player");
104 mp3_set_serial (mp3Serial); //set Serial for DFPlayer-mini mp3 module
105 delay(100); //wait lms for mp3 module to set volume
106 mp3_set_volume (EC_Config.VOLUME);
107 clrREG(0x0);
108 // Дебаг
109 Serial.println();
110 Privetstvie();
111 startUP();
112 }
```

Рисунок 3.6 – Конечные функции настройки системы. Функция clrREG()

На 92-й строке начинает свою работу веб-сервер.

Функции на строках 96-98 необходимы для настройки связи с сервером точного времени.

101-я строка открывает соединение с модулем плеера по UART-интерфейсу, на строке 106 задаётся громкость звучания динамика, соединённого с плеером.

В конце настройки (void setup) звучит приветствие (функция *Privetstvie(void)*, 129 строка): мужской голос говорит: «Приветствую Вас». Затем функция *startUP(void)* запускает процесс автоматической проверки положения ламелей жалюзи. Изначально системе неизвестно, в каком из двух положений они находятся: открыты или закрыты. Эта функция написана в отдельной вкладке и будет прокомментирована позже.

Далее по тексту программы объявляются около сорока глобальных переменных, использующиеся сразу в нескольких вкладках. Большинство из них отведены для хранения параметров погоды и времени.

```
Alarm_clock  Activ  DISPLAY  JSON  MINUS  NTF
160 void loop() {
161   Alarm();
162   static bool mas;
163   if (mas == 0) {
164     if (getTime() == 0) {
165       CATALEX.clearDisplay();
166       CATALEX.display(coed);
167       delay(3000);
168     }
169     else mas = 1;
170   }
171   if (change == true)
172   { CATALEX.set(EC_Config.BRIGHT_DISPLAY);
173     mp3_set_volume (EC_Config.VOLUME);
174     change = false;
175   }
176   }
177   if (flag_btn == true)
178   { Serial.println("turning_jalousie");
179     turning_jalousie();
180     flag_btn = false;
181   }
```

Рисунок 3.7 – Начало цикла *void loop*

В самом начале бесконечного цикла вызывается пожалуй главная функция *Alarm(void)*, которая проверяет надобность включения звонка, состоящего из мелодии и словесного прогноза погоды, и проигрывает его.

Далее проверяется необходимость подключения к серверу точного времени и запроса пакета данных о текущем времени. Попытки соединения будут выполняться до тех пор, пока флаг *mas* не будет установлен.

На 171-й строке представлена проверка флага *change*: если он устанавливается вследствие сохранения настроек в веб-интерфейсе, происходит обновление яркости свечения дисплея параметров и громкости звучания плеера; флаг сбрасывается.

Далее происходит проверка другого флага *flag\_btn*: он устанавливается при нажатии кнопки «ЖАЛЮЗИ» на главной странице веб-сервера. Если это произошло, запускается процесс смены положения ламелей жалюзи.

```
Alarm_clock$ Activ DISPLAY JSON MINUS NTP SSDP Speaking WC
182 unsigned long currentMillis = millis();
183 if (currentMillis - previousMillis >= EC_Config.TIMEOUT_DISPLAY)
184 { clrREG(byteToSend);
185   Serial.println(byteToSend, BIN);
186   previousMillis = currentMillis;
187   updDISP++;
188
189   if (updDISP > 7) {
190     updDISP = 1;
191   }
192   displayText();
193
194
195 }
196 if (updDISP == 1) {
197
198   if (updateTime() == 0) {
199     // Serial.println("ekran=0");
200     displayText();
201   }
202 }
```

Рисунок 3.8 – Основной цикл. Продолжение

На 183-й строке происходит проверка времени смены отображаемых на дисплее параметров: каждые несколько секунд (также задаётся в браузере) в цикле меняются показания дисплея, всего их 8. Текст на строках 196-202 создаёт эффект хода часов – мигание двоеточия каждую секунду в период показа времени. Приходится полностью обновлять экран каждые полсекунды.

```

Alarm_clock$ Activ DISPLAY JSON MINUS NTP SSDP Speaking
202 }
203 if (currentMillis >= updCnt) { // получаем данные погоды
204     updCnt = currentMillis + EC_Config.TIMEOUT_NTP;
205     //mas = 0;
206     Serial.println("Getting data ...");
207     CATALEX.point(POINT_OFF);
208     CATALEX.display(LOAD);
209     jsonGet();
210     if (currentMillis >= updNTP)// получаем данные времени
211     { updNTP = currentMillis + 10 * EC_Config.TIMEOUT_NTP;
212         getTime();
213     }
214     updDISP = 0;
215 }
216
217 HTTP_loop();
218 }

```

Рисунок 3.9 – Конец функции *void loop*

Строка 203: каждые *n* минут получаем свежий пакет данных с сайта OpenWeatherMap (строка 209, вызов функции *jsonGet(void)*). Примерно таким же способом выявляем потребность в точном времени, только делаем это гораздо реже (в 10 раз, строка 211). Строка 217 поддерживает активность веб-сервера.

Это была основная вкладка, теперь рассмотрим функции, которые вызывались отсюда. Рассмотрим вкладку DISPLAY (рисунки 3.10-3.18).

```

Alarm_clock Activ DISPLAY JSON MINUS NTP
1 int8_t DispMSG[4] = {0x00, 0x00, 0x00, 0x00};
2 int8_t N[] = {17, 17, 17, 12};
3 int8_t NE[] = {17, 12, 16, 8};
4 int8_t E[] = {17, 17, 17, 8};
5 int8_t SE[] = {46, 0, 16, 8};
6 int8_t S[] = {17, 17, 46, 0};
7 int8_t SW[] = {46, 0, 16, 3};
8 int8_t W[] = {17, 3, 10, 39};
9 int8_t NW[] = {17, 12, 16, 3};
10
11 int8_t asno[] = {47, 12, 19, 0};
12 int8_t mobil[] = {25, 0, 6, 38};
13 int8_t rvan[] = {27, 8, 10, 19};
14 int8_t obl[] = {17, 0, 6, 38};
15 int8_t livn[] = {38, 36, 8, 19};
16 int8_t dshzd[] = {34, 0, 35, 34};
17 int8_t groz[] = {33, 27, 0, 3};
18 int8_t sneg[] = {12, 19, 14, 33};
19 int8_t tuma[] = {40, 32, 25, 10};
20
21 int8_t priv[] = {39, 27, 36, 8};

```

Рисунок 3.10 – Маски отображения слов на семисегментном индикаторе.

Для отображения погодных условий, а также направления ветра понадобятся маски – массивы номеров символов (всего 4 знака), которые также хранятся в виде однобайтовых масок в библиотеке TM1637. Библиотека была модифицирована, в неё добавлены маски русских букв. Маски букв английского алфавита и цифр показаны на рисунке 3.11.



Рисунок 3.11 – Маски букв английского алфавита и цифр

```

Alarm_clock  Activ  DISPLAY  JSON  MINUS  NTP  SSDP  Sp
22
23 void displayText() {
24     switch (updDISP) {
25         case 1: {
26             CATALEX.clearDisplay();
27             if (dots) CATALEX.point(POINT_ON);
28             else CATALEX.point(POINT_OFF);
29             DispMSG[0] = h / 10;
30             DispMSG[1] = h % 10;
31             DispMSG[2] = m / 10;
32             DispMSG[3] = m % 10;
33             CATALEX.display(DispMSG); break;
34         }
35         case 2: {
36             CATALEX.clearDisplay();
37             CATALEX.point(POINT_OFF);
38             DispMSG[0] = temptrue(0); //round(tempC) / 10;
39             DispMSG[1] = temptrue(1); //round(tempC) % 10;
40             DispMSG[2] = 48;
41             DispMSG[3] = 12;

```

Рисунок 3.12 – Функция *displayText(void)*

Функция, представленная на рисунке 3.12, состоит из оператора switch-case. Он отслеживает переменную *updDisp*, которая представляет собой

счётчик от 1 до 7. В любом случае, если пришло время обновления экрана, дисплей очищается стандартной функцией `.clearDisplay()`. Рассмотрим случай, когда `updDisp=1`.

Строка 27: флаг `dots` каждые полсекунды меняет своё значение. Если он установлен, включается двоеточие, иначе (строка 28) выключается. Далее заполняется массив: десятки часов, единицы часов, десятки минут, единицы минут. Затем массив записывается в память дисплея, отображается время.

Ниже приведен список отображаемых по очереди параметров:

1. Время
2. Температура
3. Давление
4. Влажность
5. Скорость ветра
6. Направление ветра
7. Характер погоды (ясно, гроза и т.п.)

На вкладке JSON располагается функция `jsonGet(void)`, осуществляющая парсинг информационных пакетов с данными о погоде. Здесь происходит соединение с сервисом, запрос данных, расшифровка полученного пакета в формате JSON.

```
Alarm_clock  Activ  DISPLAY  JSON  MINUS  NTP  SSDP  Speaking
1 int temptrue(int tt)
2 {
3     if (tempC > -10 && tempC < 0) {
4         if (tt == 0) return 16;
5         if (tt == 1) return (abs((round(tempC)) % 10));
6     }
7     if (tempC <= -10 || tempC >= 0) {
8         if (tt == 0) return interesting(abs((round(tempC)) / 10));
9         if (tt == 1) return (abs((round(tempC)) % 10));
10    }
11 }
12
13 int interesting(int xint)
14 {
15     if (xint == 0) xint = 17;
16     return (xint);
17 }
18 //=====
```

Рисунок 3.13 – Вкладка MINUS

На вкладке MINUS располагаются функции для корректного отображения отрицательных температур на дисплее, где число знаков ограничено. Так, отобразить «-16°C» не получится: просто не хватит места, в этой записи больше четырёх символов. Поэтому было решено отображать двузначные отрицательные значения без знака минус. Функция *interesting(int xint)* заменяет нули в начале однозначных чисел пробелом, например: « 9°C», а не «09°C»

Вкладка NTP содержит функции, аналогичные тем, что размещены на вкладке JSON. Только запрашиваются и обрабатываются здесь данные о времени [23].

Вкладка Speaking предназначена для синтеза голосовых сообщений во время утреннего звонка будильника. Идея состоит в том, что сначала называется параметр, например, «Давление», затем читается его числовое значение по правилам русского языка, например, «семьсот шестьдесят девять»; далее называется единица измерения: «миллиметров ртутного

столба». [7] Таким образом описывается вся погодная обстановка. Вложенность функций в этой вкладке выглядит следующим образом:

```
void speakText()  
    |  
    | void ozv(int papka, int myfile)  
    |  
    | voidcedig(char cc[])  
    |  
    | | void ozv(int papka, int myfile)
```

Функция `speakText()` отвечает за последовательное проговаривание отдельных имён существительных и числительных: чередуются функции `void ozv(int papka, int myfile)` и `voidcedig(char cc[])`. Первая проигрывает отдельный звуковой файл из конкретной папки. Вторая «читает» имя числительное по правилам русского языка; она также использует простейшую функцию проговаривания слова.

При желании можно добавить проигрывание музыкального трека в начале, перед сообщением о погоде.

Правее располагаются вкладки библиотек `WC_EEPROM.cpp/h` для работы с энергонезависимой памятью и `WC_HTTP.cpp/h`, содержащая в основном код на языке HTML для загрузки страниц в браузер при их вызове. Эти библиотеки скачаны из [5] и слегка доработаны, оптимизированы под настоящий проект. На рисунке 3.14 представлен внешний вид главной страницы в браузере.

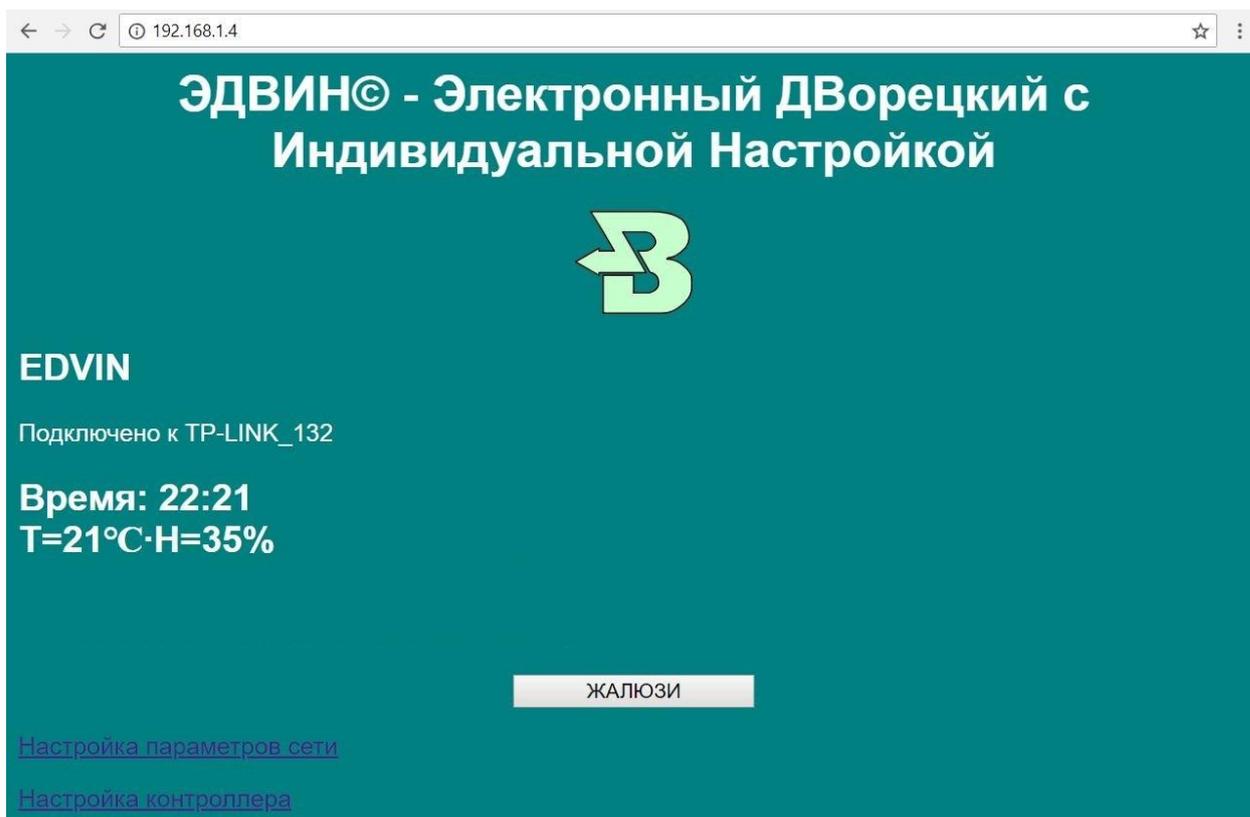


Рисунок 3.14 – Вид главной страницы веб-интерфейса будильника

На рисунке 3.15 изображён вид страницы настроек сети для связи будильника с сервисами.

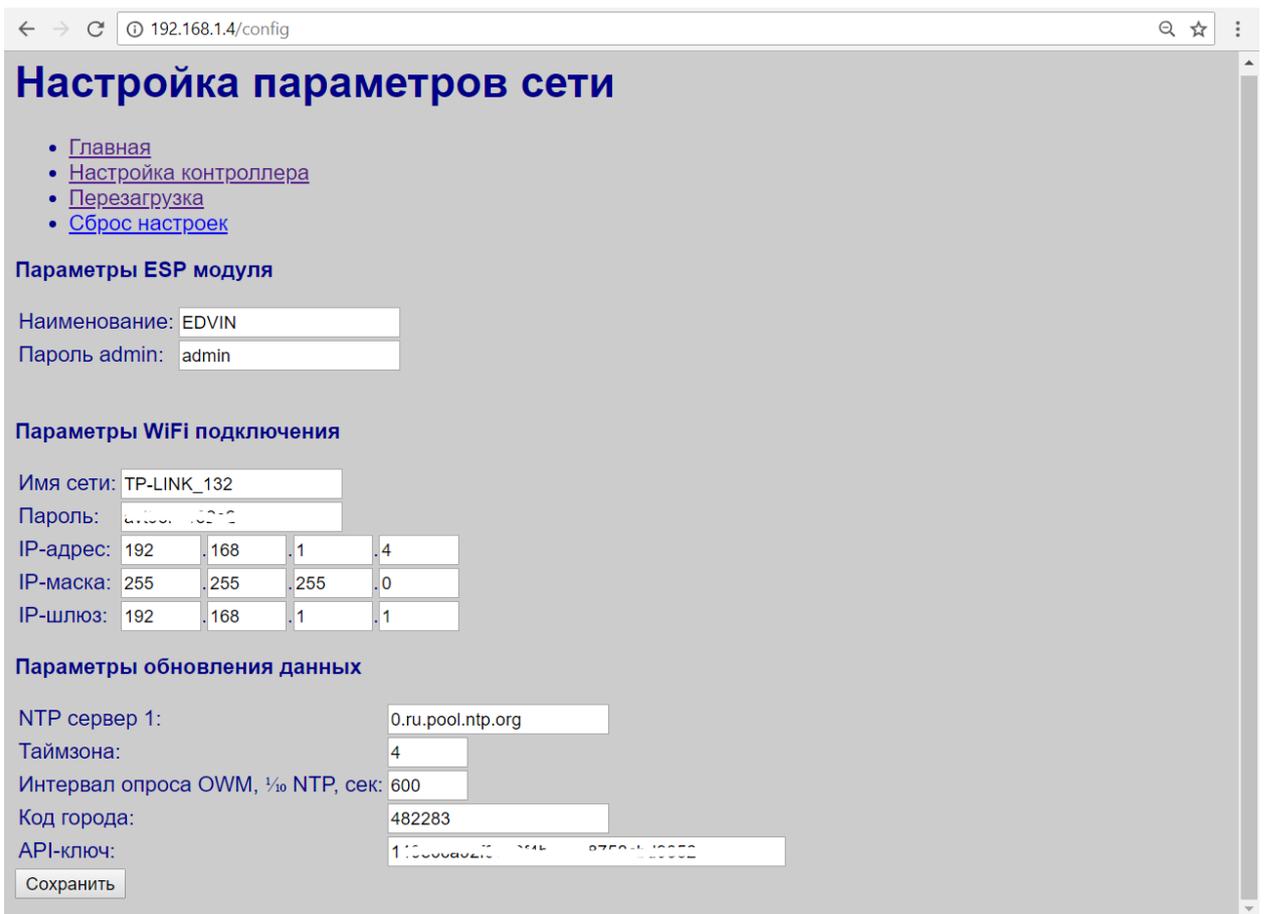


Рисунок 3.15 – Вид веб-страницы сетевых настроек

Рассмотрим следующую вкладку Weather (рисунок 3.16).

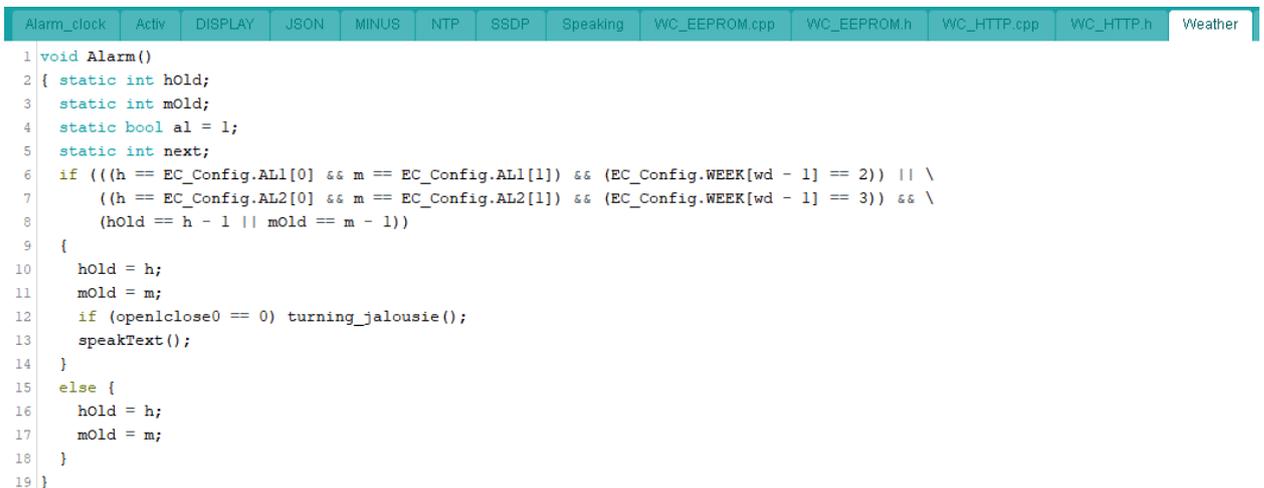
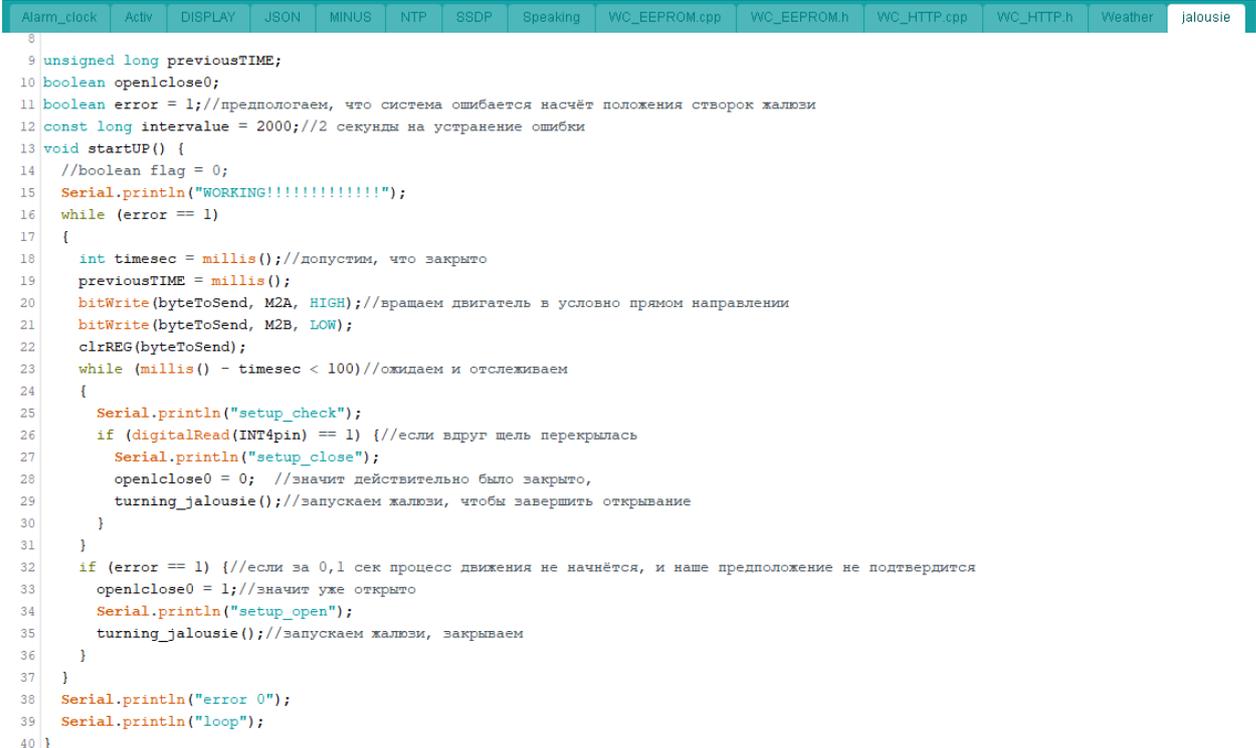


Рисунок 3.16 – Вкладка Weather

Функция *Alarm()*, которая вызывалась в строке 161 текста на рисунке 3.7, постоянно отслеживает текущее время, и если наступает момент звонка, и если он в этот день вообще установлен, открываются жалюзи (если они закрыты) запускается функция проговаривания текста (строка 12).

Прокомментируем не менее важную, последнюю вкладку *jalousie*, отвечающую за калибровку и дальнейшее функционирование жалюзи.

На рисунке 3.17 приведён скриншот окна с текстом функции *startUP()*, которая вызывалась в конце первоначально настройки системы (рисунок 3.6, строка 111).



```
8
9 unsigned long previousTIME;
10 boolean openIclose0;
11 boolean error = 1; //предполагаем, что система ошибается насчёт положения створок жалюзи
12 const long intervalue = 2000; //2 секунды на устранение ошибки
13 void startUP() {
14     //boolean flag = 0;
15     Serial.println("WORKING!!!!!!!!!!!!!!");
16     while (error == 1)
17     {
18         int timesec = millis(); //допустим, что закрыто
19         previousTIME = millis();
20         bitWrite(byteToSend, M2A, HIGH); //вращаем двигатель в условно прямом направлении
21         bitWrite(byteToSend, M2B, LOW);
22         clrREG(byteToSend);
23         while (millis() - timesec < 100) //ждем и отслеживаем
24         {
25             Serial.println("setup_check");
26             if (digitalRead(INT4pin) == 1) { //если вдруг шель перекрылась
27                 Serial.println("setup_close");
28                 openIclose0 = 0; //значит действительно было закрыто,
29                 turning_jalousie(); //запускаем жалюзи, чтобы завершить открытие
30             }
31         }
32         if (error == 1) { //если за 0,1 сек процесс движения не начнётся, и наше предположение не подтвердится
33             openIclose0 = 1; //значит уже открыто
34             Serial.println("setup_open");
35             turning_jalousie(); //запускаем жалюзи, закрываем
36         }
37     }
38     Serial.println("error 0");
39     Serial.println("loop");
40 }
```

Рисунок 3.17 – Функция *startUP()*

Многие строки здесь прокомментированы. Однако кратко опишем суть этой подпрограммы. Для полного понимания приведём схему узла с датчиком прерывания луча (рисунок 3.18).

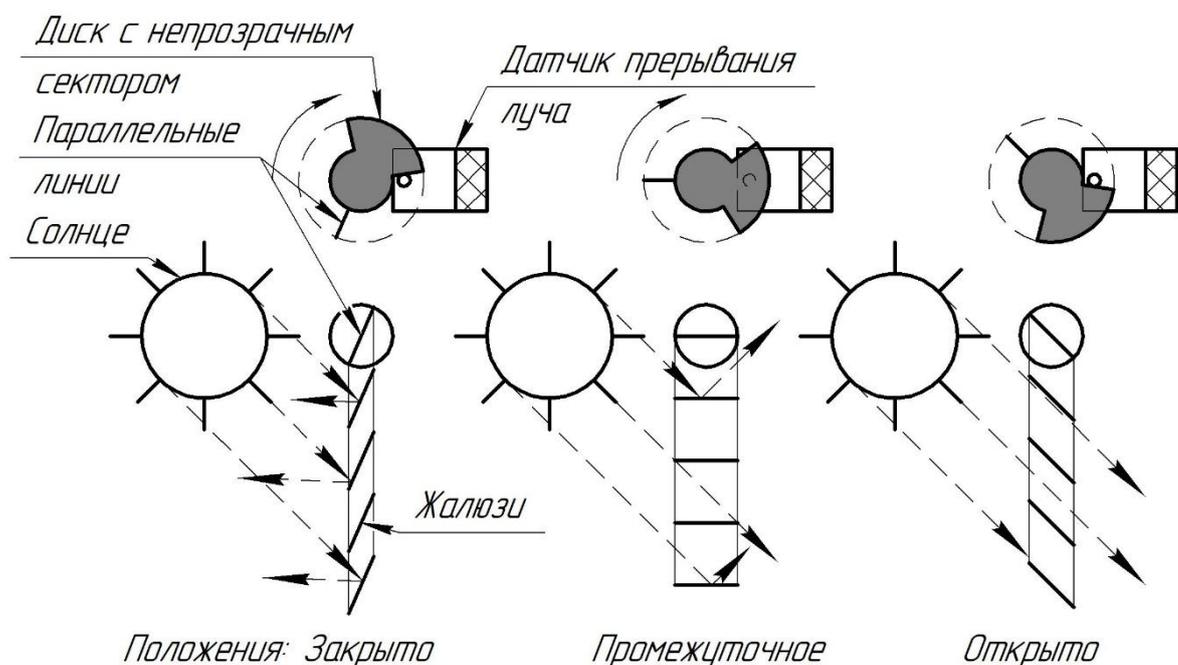


Рисунок 3.18 – Схема работы узла поворота ламелей жалюзи

На схеме изображён узел поворота ламелей жалюзи в трёх основных положениях: когда жалюзи закрыты, в промежуточном, и когда открыты. Сверху в каждом случае изображена сцепка «диск-датчик», снизу – система «Солнце-ламели». Для сопоставления верхней и нижней частей схемы, в неё введены опорные линии, которые для каждого из трёх случаев остаются строго параллельными.

Итак, возвратимся к программе. Изначально предполагаем, что жалюзи закрыты. Солнечные лучи почти не проникают в помещение. Инфракрасный луч датчика не прерывается. Но даже если он и прерывается, подключаем двигатель к источнику питания с такой полярностью, чтобы он вращался по часовой стрелке. Система переходит в промежуточное положение. Действие передаётся функции *turning\_jalousie()*, которая и завершит смену положения (строка 29 на рисунке 3.17).

Солнечных лучей проникает больше, частично это отражённые от ламелей лучи, частично – прямые. Двигатель продолжает вращать ламели по часовой стрелке до тех пор, пока инфракрасный луч снова не будет беспрепятственно проходить сквозь щель датчика (положение «открыто»).

Однако, если система изначально будет в положении «открыто», система этого знать не будет и всё равно создаст условно положительное направление вращения. Фактически движения не будет, так как механизм сделать это не позволит. Через 100 миллисекунд переменной *openIclose0* присвоится значение 1, что значит открыто. Функция *turning\_jalousie()* закроет жалюзи. Она объявлена на этой же вкладке и абсолютно симметрична: в зависимости от флага *openIclose0* при её вызове открываются или закрываются жалюзи, проходя все вышеперечисленные этапы.

#### 4. Конструктивно-экспериментальная часть

##### 1.10 Разработка и сборка конструкции

В первую очередь перед изготовлением корпуса будильника была подготовлена его трёхмерная модель в программном пакете Компас. Дизайн разработан с учётом размеров уже имеющихся компонентов.

Покажем внешние виды основных узлов будильника. На рисунке 4.1 показана трёхмерная модель главной платы устройства, созданная в программном пакете DipTrace. Узел, выполненный на макетной печатной плате в реальности выглядит аналогично, виден на рисунке 4.3.

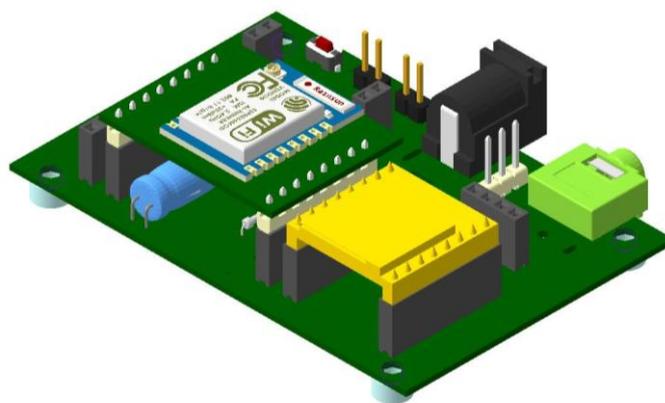


Рисунок 4.1 – Внешний вид главной платы (3D модель)

На рисунке 4.2 изображён внешний вид смоделированного будильника.

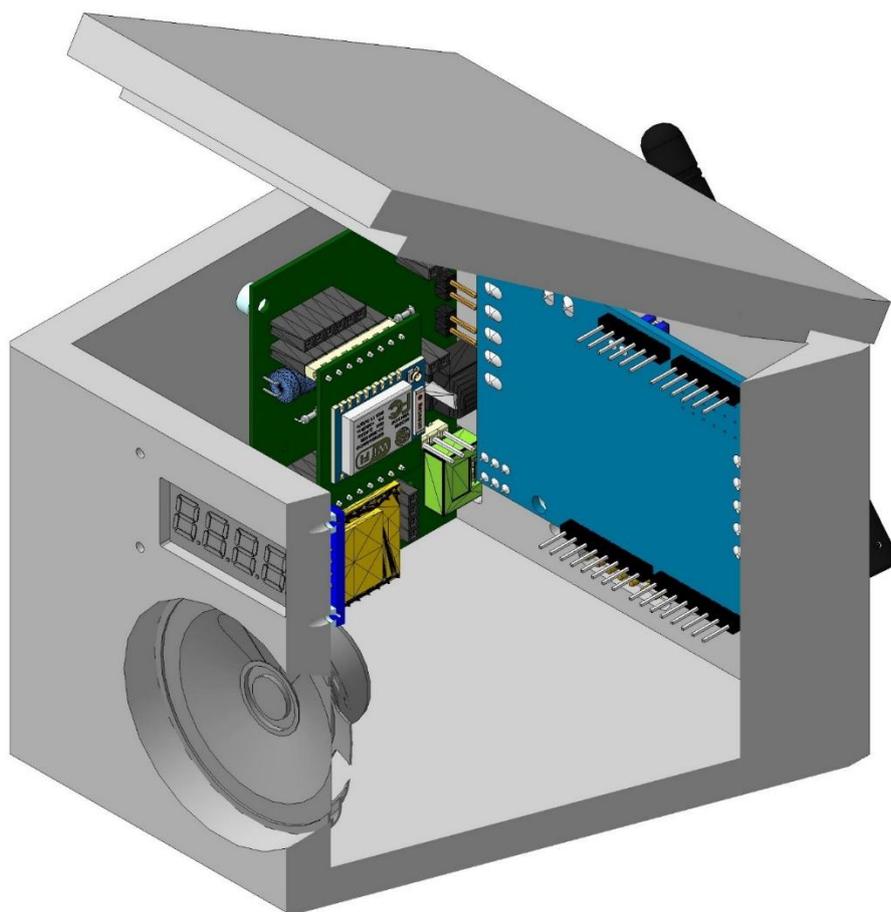


Рисунок 4.2 – Трёхмерная модель собранного устройства (в сечении)

На рисунке 4.3 изображён только что изготовленный корпус будильника с главной платой и модулем CATALEX. После предварительной примерки корпус выкрашен в серый цвет и в него установлена вся остальная периферия.



Рисунок 4.3 – Начало сборки устройства

Внешний вид полностью собранного работающего устройства показан на рисунке 4.4. Примечание: на дисплее показана температура: «18°C».



Рисунок 4.4 – Внешний вид собранного будильника

Узел поворота ламелей жалюзи (рисунок 4.5) соединяется с будильником при помощи патч-корда. Также на корпусе предусмотрен разъём Db15 (так много выводов необходимо для дальнейшего расширения функционала устройства). [24]

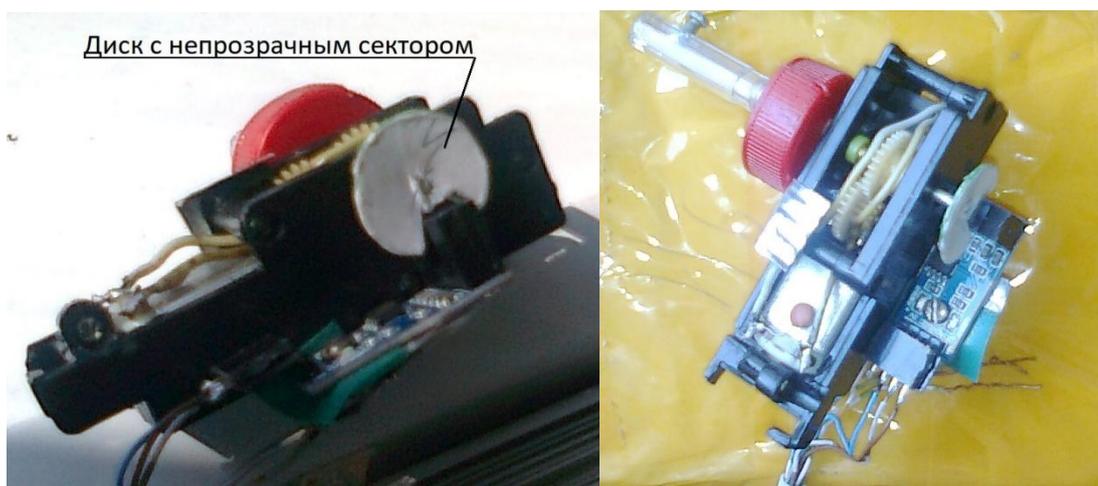


Рисунок 4.5 – Внешний вид узла «Мотор-диск с непрозрачным сектором-датчик прерывания»

На рисунках 4.6 и 4.7 изображены модифицированные жалюзи с положением ламелей «открыто» и «закрыто» соответственно.

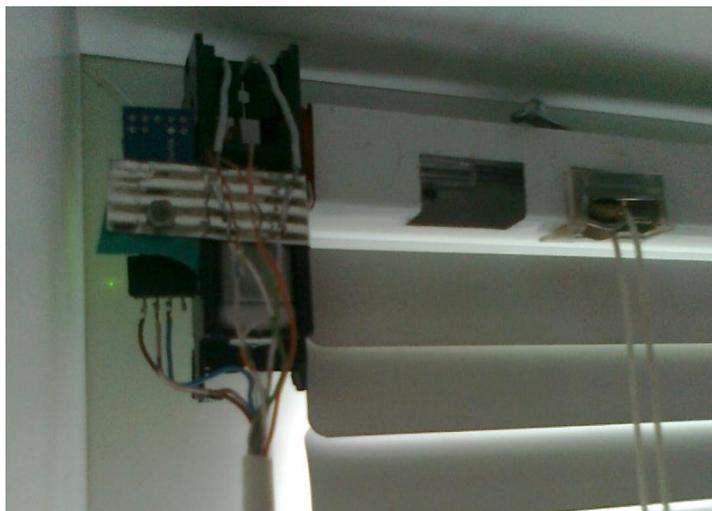


Рисунок 4.6 – Модифицированные жалюзи с положением ламелей «открыто»



Рисунок 4.7 – Модифицированные жалюзи с положением ламелей «закрыто»

### 1.11 Отладка конструкции и экспериментальные исследования

Приведём последовательность действий по включению будильника:

- Подключить к сети 220 В вилку кабеля, неотделимого от корпуса (питание силовой части; внутри корпуса размещается блок питания с выходным напряжением 12 В и током до 3А)
- Включить штекер кабеля к разъёму питания постоянного тока, расположенного на задней стенке корпуса будильника (2,5x5,5)

- Подключить к сети блок питания (выход 5 В, 1 А, для логической части)

Подразумевается, что к разъёму Db15 уже подключен штекер с кабелем, ведущим к узлу поворота ламелей жалюзи.

После вышеуказанных действий устройство попытается подключиться к точке доступа, имя и пароль к которой записаны в память EEPROM микроконтроллера. Если попытка завершится неудачей, то он создаёт свою точку доступа. По адресу 192.168.4.1 можно поменять параметры сети.

Далее запускается процесс инициализации жалюзи: ламели должны провернуться.

При подключении к сети Интернет на дисплее отображается ключевое слово «ГРУЗ». В это время будильник запрашивает и загружает необходимые информационные пакеты времени и погоды. Устройство полностью готово к работе.

Будильник достаточно автономен. Его работоспособность проверялась в течении недели на момент подготовки настоящей работы. Функциональность устройства подтверждена:

- В цикле отображаются параметры времени и погоды.
- Веб-сервер активен, веб-страницы в браузере отображаются корректно.
- Данные в веб-интерфейсе сохраняются корректно, проверено на изменении яркости дисплея.
- При нажатии кнопки ЖАЛЮЗИ на главной веб-странице с адресом 192.168.1.4 ламели меняют своё положение.
- Будильник звучит и жалюзи открываются в правильное время и в нужные дни недели. Звуковые сообщения составляются корректно.

Исходя из вышеперечисленных пунктов, разработанная модель будильника в целом работает корректно.

## Заключение

В настоящей работе были проанализированы некоторые устройства с функциями, подобными тем, что реализуются в данной модели. Составлены функциональная и принципиальная схемы электронного устройства. Для него разработаны макетный печатный узел, корпус. Подготовлен и введён в эксплуатацию действующий прототип. Также будильник всячески протестирован на предмет корректного исполнения предъявленных требований.

## Список используемой литературы

1. Kodali, R. K. Low Cost Ambient Monitoring using ESP8266 [Text] Ravi Kishore Kodali and Kopulwar Shishir Mahesh Department of Electronics and Communication Engineering National Institute of Technology, Warangal, 2017. – PP. 779-780 [Электронный ресурс]. – URL : [https://www.researchgate.net/publication/316732922\\_Low\\_cost\\_ambient\\_monitoring\\_using\\_ESP8266](https://www.researchgate.net/publication/316732922_Low_cost_ambient_monitoring_using_ESP8266) (дата обращения 8.09.2017)
2. J.A.R.V.I.S. [Электронный ресурс] // Marvel Movies is a FANDOM Movies Community: информационно-справочный портал, 2018. URL: <http://marvel-movies.wikia.com/wiki/J.A.R.V.I.S.> (дата обращения 27.01.2018)
3. Парсинг. Что это и где используется. [Электронный ресурс] // iPipe.ru: информационно-справочный портал, © 2018. URL: <https://www.ipipe.ru/info/parsing.html> (дата обращения 28.01.2018)
4. esp8266 и парсинг погоды с OpenWeatherMap [Электронный ресурс] // iT4iT CLUB: форум, 2018. URL: <https://it4it.club/topic/40-esp8266-i-parsing-pogodyi-s-openweathermap/> (дата обращения 28.01.2018)
5. Контроллер управления вытяжным вентилятором с WiFi [Электронный ресурс] //geektimes.com: научно-популярный журнал, © 2006 – 2018 «ТМ» URL: <https://geektimes.ru/post/270142/> (дата обращения 04.02.2018)
6. Datasheet\_TM1637.pdf [Электронный ресурс] //mcielectronics.cl: сайт компании-производителя, © 2017. URL: [http://www.mcielectronics.cl/website\\_MCI/static/documents/Datasheet\\_TM1637.pdf](http://www.mcielectronics.cl/website_MCI/static/documents/Datasheet_TM1637.pdf) (дата обращения 04.12.2017)
7. Дисплей на TM1637, часы с синхронизацией по NTP и программирование "народных" WiFi в среде Ардуино [Электронный ресурс] // mysku.ru: обзоры товаров интернет-магазинов, 2018. URL: <https://mysku.ru/blog/aliexpress/33970.html> (дата обращения 04.02.18)

8. Учим разговаривать DIY с помощью MP3 модуля [Электронный ресурс] // mysku.ru: обзоры товаров интернет-магазинов, 2018. URL: <https://mysku.ru/blog/aliexpress/50345.html> (дата обращения 08.08.2018)
9. Управление жалюзи, шторами, рольставнями, воротами, экранами кинотеатров и приводами других устройств [Электронный ресурс] // ТРИА Комплекс инженерных систем: рекламный сайт, 2018. URL: [http://www.tria-komm.ru/clever\\_house/control-blinds/](http://www.tria-komm.ru/clever_house/control-blinds/) (дата обращения 28.01.2018)
10. Система управления жалюзи Gira System 3000 [Электронный ресурс] // gira.ru: сайт компании, © 2006—2018 ООО «ГИЛЭНД» URL: <http://www.gira.ru/products/novinki/1404/> (дата обращения 28.01.2108)
11. Автоматические жалюзи на Arduino [Электронный ресурс] // geektimes.com: научно-популярный журнал, © 2006 – 2018 «ТМ» URL: <https://geektimes.ru/post/258112/> (дата обращения 28.01.2018)
12. AMS1117 3.3V – МОДУЛЬ СТАБИЛИЗАТОРА НАПРЯЖЕНИЯ [Электронный ресурс] // Вольтик: интернет-магазин, 2018. URL: <https://voltiq.ru/shop/voltage-regulator-3-3-v-ams1117/> (дата обращения 19.04.2018)
13. Adafruit motor shield подключение к Arduino [Электронный ресурс] // zelectro.cc: магазин и сообщество Zelectro, 2018. © 2013-2018. URL: [http://zelectro.cc/Adafruit\\_motor\\_shield](http://zelectro.cc/Adafruit_motor_shield) (дата обращения 10.12.2017)
14. ESP8266 — подключение и обновление прошивки [Электронный ресурс] // esp8266.ru: Сообщество разработчиков, © 2014-2016. URL: <https://esp8266.ru/esp8266-podkluchenie-obnovlenie-proshivki/> (дата обращения 10.08.2017)
15. Ms. Malathi, M. Home automation [Text] Ms.M.Malathi Assistant Professor: Dept. of I.T., Sri Krishna College of Technology, Coimbatore, India, 2017. – PP. 1-3 [Электронный ресурс]. – URL : [https://www.researchgate.net/publication/316926871\\_HOME\\_AUTOMATION\\_ON\\_ESP8266](https://www.researchgate.net/publication/316926871_HOME_AUTOMATION_ON_ESP8266) (дата обращения 8.09.2017)

- 16.МОДУЛИ НА БАЗЕ ESP8266 [Электронный ресурс] //mirrobo.ru: информационно-справочный портал, Академия робототехники, © 2018. URL: <https://mirrobo.ru/micro/moduli-na-baze-esp8266/> (дата обращения 11.03.2018)
- 17.Abdulrahman, T. A. Design, Specification and Implementation of a Distributed Home Automation System [Text] / T. A. Abdulrahman, O. H. Isiwekpeni, N. T. Surajudeen-Bakinde, A. O. Otuoze Department of Electrical & Electronics Engineering, University of Ilorin, P.M.B. 1515, Nigeria, IoT NAT 2016. – PP. 475-476 [Электронный ресурс]. – URL: [https://www.researchgate.net/publication/306047899\\_Design\\_Specification\\_and\\_Implementation\\_of\\_a\\_Distributed\\_Home\\_Automation\\_System](https://www.researchgate.net/publication/306047899_Design_Specification_and_Implementation_of_a_Distributed_Home_Automation_System)
- 18.Visconti, P Solar powered wireless monitoring system of environmental conditions for early flood prediction or optimized irrigation in agriculture [Text] / Paolo Visconti, Patrizio Primiceri and Cosimo Orlando Department of Innovation Engineering, University of Salento, Street for Monteroni - Lecce, Italy, 2016. – PP. 4624-4626 [Электронный ресурс]. – URL: [https://www.researchgate.net/publication/304929498\\_Solar\\_powered\\_wireless\\_monitoring\\_system\\_of\\_environmental\\_conditions\\_for\\_early\\_flood\\_prediction\\_or\\_optimized\\_irrigation\\_in\\_agriculture](https://www.researchgate.net/publication/304929498_Solar_powered_wireless_monitoring_system_of_environmental_conditions_for_early_flood_prediction_or_optimized_irrigation_in_agriculture) (дата обращения 8.09.2017)
- 19.WiFi SoC (System on Chip) [Электронный ресурс] //electrosome.com: портал технической организации, © 2018. URL: <https://electrosome.com/esp8266/>. (дата обращения 11.03.2018)
- 20.TM1637 Datasheet – IC for 4-digit 7-segment display – Titan Micro [Электронный ресурс] //datasheetcafe.com: электронный справочник, 2018. URL: <http://www.datasheetcafe.com/tm1637-datasheet-pdf/> (дата обращения 12.09.2017)
- 21.DFPlayer Mini SKU: DFR0299 [Электронный ресурс] //dfrobot.com: информационный ресурс ведущего в мире поставщика оборудования для робототехники, © 2008- 2018. URL:

[https://www.dfrobot.com/wiki/index.php/DFPlayer\\_Mini\\_SKU:DFR0299](https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299)

(дата обращения 10.08.2017)

- 22.МОДУЛИ СЧИТЫВАНИЯ УГЛОВОЙ СКОРОСТИ (ЭНКОДЕРЫ) ДЛЯ ARDUINO [Электронный ресурс] //mirrobo.ru: информационно-справочный портал, Академия робототехники, © 2018. URL: <https://mirrobo.ru/micro/schityvaniya-uglovoj-skorosti-jenkodery-dlja-arduino/> (дата обращения 5.09.2017)
- 23.TM1637 [Электронный ресурс] //youtube.com: видеохостинг, 2018 URL: <https://www.youtube.com/watch?v=zby4-YHot6w> (дата обращения 25.12.20)
- 24.Kuppusamy, P. Smart Home Automation using Sensors and Internet of Things [Text] / Dr. P. Kuppusamy IT Research Lab, Department of Computer Science & Engineering, Gnanamani College of Technology, Namakkal, India, 8, August 2016. – PP. 2644-2645 [Электронный ресурс]. – URL: [https://www.researchgate.net/publication/309855780\\_Smart\\_Home\\_Automation\\_Using\\_IoT](https://www.researchgate.net/publication/309855780_Smart_Home_Automation_Using_IoT)