МАОУ "Усть-Кубинский центр образования"

Умный курятник

Ученик 10 класса:

Сакулин Иван Михайлович

Руководитель:

Андреев Николай Николаевич

с. Устье, 2023

Оглавление

[Оглавление 2](#_Toc126105267)

[ЛОЛОЛО 4](#_Toc126105268)

[Вступление 4](#_Toc126105269)

[Теория 4](#_Toc126105270)

[1.1. Выбор сред и инструментов разработки. 4](#_Toc126105271)

[1.2. Arduino. ArduinoIDE. Atmega328p. ESP8266. 4](#_Toc126105272)

[1.3. Android-приложение 5](#_Toc126105273)

[1.4. Печатные платы 5](#_Toc126105274)

[1.5. Контроль версий и разработка 5](#_Toc126105275)

[Аналоги 5](#_Toc126105276)

[Этапы реализации 6](#_Toc126105277)

[1. Начало. 6](#_Toc126105278)

[2. Разработка первоначальной печатной платы. 6](#_Toc126105279)

[3. Создание первоначального приложения и локального сервера на esp8266. 6](#_Toc126105280)

[4. Первый функционал 6](#_Toc126105281)

[5. Приложение 6](#_Toc126105282)

[6. Периферия 6](#_Toc126105283)

[7. Таймеры. Реле 7](#_Toc126105284)

[8. Корпус 7](#_Toc126105285)

[9. Доработка и добавление функций 7](#_Toc126105286)

[10. Тесты 7](#_Toc126105287)

[Ресурсы 7](#_Toc126105288)

[Перспективы и направления развития 7](#_Toc126105289)

[1. Контроллер 7](#_Toc126105290)

[2. Приложение 8](#_Toc126105291)

[3. Модули 8](#_Toc126105292)

[Заключение 8](#_Toc126105293)

[Список литературы 8](#_Toc126105294)

[Приложения 9](#_Toc126105295)

# Вступление

/\* НУЖЕН ИВАН-ПИСАТЕЛЬ \*/

Актуальность проекта:

Цель проекта: Разработка и тестирование контроллера и нескольких частей автоматического/умного курятника.

Задачи:

* Изучение аналогов и литературы
* Разработка и сборка контроллера, корпуса к нему
* Разработка печатной платы на базе микроконтроллера atmega328p
* Разработка андроид-приложения
* Встраивание системы в обычный курятник и тестирование
* /\* Создание поилки, кормушки с приводом \*/

/\* Таким способом обозначаются комментарии. В готовой версии документа они должны отсутствовать, если они всё же есть, пожалуйста, не воспринимайте их всерьёз и обратитесь за полной версией по электронной почте [ivansakulin5505@gmail.com](mailto:ivansakulin5505@gmail.com) \*/

# Теория

## 1.1. Выбор сред и инструментов разработки.

Очень важные части в реализации проекта: подобрать достаточно удобные инструменты и расходные материалы, с помощью которых можно реализовать необходимые функции, создать основу и развить её.

В процессе работы я смог прийти к оптимальной схеме (Рисунок 1).

Рисунок 1. Схема проекта

## 1.2. Arduino. ArduinoIDE. Atmega328p. ESP8266.

Arduino — это небольшая управляющая плата с собственным процессором и памятью. В процессор Ардуино можно загрузить программу (скетч), которая будет управлять устройствами по заданному алгоритму, в т.ч. используя датчики.

Скетчи удобно разрабатывать в Arduino IDE на языке C++. Это самая мощная среда разработки, использовать что-то другое для меня нецелесообразно.

Для макетирования и сборки первых прототипов такая плата подходит идеально, что не всегда скажешь о реализации: на них много светодиодов, стабилизаторов и прочего "обвеса", который может не использоваться, но всё равно будет потреблять энергию. Поэтому в серьёзных проектах лучше использовать чип AVR, который имеет в себе процессор и память, самый популярный из них atmega328p – его я использую, только с необходимым "обвесом".

В проекте atmega328p занимается только частью управления курятником и обработкой датчиков.

Связь основного контроллера с пользователем с целью настройки и проверки происходит посредством более мощного чипа с намного меньшим количеством пинов (выводы и вводы для внешнего воздействия), большим потреблением энергии, но с wi-fi и большей скоростью работы – esp8266. Этот чип не из семейства AVR, но также поддерживается в Arduino IDE.

## 1.3. Android-приложение

Приложения для телефона я делал с помощью Android Studio на языке kotlin. Большая часть времени ушла на создание внутреннего сервиса запросов. Во включенном состоянии он параллельно основному потоку, но связанно делает запросы и отправляет команды пользователя из графического интерфейса на контроллер и обратно.

## 1.4. Печатные платы

EasyEDA — кросс-платформенная веб-ориентированная среда автоматизации проектирования электроники включающая в себя редактор принципиальных схем, редактор топологии печатных плат, SPICE-симулятор, облачное хранилище данных, систему управления проектами, а также средства заказа изготовления печатных плат.

JLCPCB (Shenzhen JIALICHUANG Electronic Technology Development Co., Ltd.), является крупнейшим предприятием по производству прототипов печатных плат в Китае и высокотехнологичным производителем, специализирующимся на быстром прототипе печатных плат и мелкосерийном производстве печатных плат.

Проект не нуждается в крупномасштабном серийном производстве. Но высокого качества малых партий при реализации добиться возможно.

## 1.5. Контроль версий и разработка

Проект получился весьма объёмный по количеству кода и прочих файлов, поэтому я прибег к GitHub – это удобный инструмент для контроля версий, с его помощью можно без лишних усилий откатить проект, если вдруг что-то важное или неизвестное перестанет работать.

Всю разработку я проводил так, чтобы в любой момент времени ко мне мог присоединиться кто-нибудь, и ему было просто выделить определённую часть, которая не связана с остальными, но обособлена от них.

# Аналоги

Ещё на моменте начала разработки я искал в Интернете возможные способы автоматизации курятника, но полноценных контроллеров, увы, найти не мог – тогда мне и пришла в голову мысль сделать свой. Но на последних этапах реализации мне на глаза попался хороший аналог проекта (https://мойкурятник.рф). Проанализировав внимательно их продукт, я выделил основные особенности и составил таблицу сравнения (Таблица 1).

Таблица 1. Сравнительная таблица аналогов

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Проект** | **«мойкурятник.рф»** |
| Состояние | Прототип 1.0, работает. | Готовое устройство 4.2. |
| Питание | 5В или 6-24В постоянного тока | 220В переменного тока |
| Настройка | Подключение по wi-fi.  Приложение для android, web-запросы. | Подключение по wi-fi. Приложение для android на 9 устройств, web-интерфейс. IK. |
| Измерения состояния окружающей среды | Датчик температуры (-40..+85°C), влажности (0-100%) и давления (30-110 кПа) BME280. Погрешности: | Датчик температуры (0..50°C) и влажности (20..90%) DHT-11.  Погрешности 2°С, 5%.  (DHT-22: 0-100% и -40..+85°C – в наличии не обнаружен). |
| Энергонезависимое время | Присутствует, RTC DS3231 mini. | Присутствует, нет описания. |
| Крепление, корпус | На стену, пластмасса. | На стену, защита ip45 |
| Нагрузка и модули | 4 канала реле с нагрузкой до 1кВт 220В переменного или 12V постоянного тока, 4 разъёма под другие устройства: подключение кормушки, поилки, вентиляции и т.д. предусмотрено, но модули ещё в разработке. | Уличный свет, внутренний свет, лампа обогрева, вентилятор 220В переменного тока. Прочие модули не предусмотрены. |
| Стоимость | /\*РАСЧЁТ\*/ | 4400р – контроллер (4060р без корпуса) |

# Этапы реализации

## 1. Начало.

В практической части я описал выполнение каждого пункта плана.

Началось всё, конечно, с поиска проблемы. После долгих раздумий, я пришёл к выводу: неплохо бы было сделать свой курятник чуть более автономным и попытаться сделать стабильную версию, чтобы в будущем предложить её соседям и родственникам. Я здраво оценивал свои силы и осознавал, что на это времени не хватит, поэтому в рамках проекта ограничил свою цель до размеров своего курятника.

## 2. Разработка первоначальной печатной платы.

Первое что я сделал – развёл плату в EasyEDA, опираясь на те части, которые должны присутствовать в проекте. Заказал её на JLCPCB. А в процессе разработки отмечал все недостатки и ошибки в ней, которые устранял в более совершенной разводке платы.

## 3. Создание первоначального приложения и локального сервера на esp8266.

Я реализовал простейший интерфейс с парой кнопок и нестабильный поток в приложении. На esp8266 основательно написал следующее: wi-fi точка доступа, сервер, универсальная структура обработчиков http-запросов, serial-соединение с основным чипом, который на этом этапе умел только мигать лампочкой по команде, сразу реализовал debug (отладку).

## 4. Первый функционал

Приводы и механизмы – это самое сложное, поэтому для начала я подключил к контроллеру модуль реального времени и датчик температуры, влажности и давления (BME280). Затем – прописал передачу этих значений на телефон для проверки

## 5. Приложение

Чтобы написать хорошее, стабильное приложение мне пришлось переписывать всю структуру несколько раз, пока я не добился идеального результата: «activity» – то, с чего всё начинается и запускается, «service» для общения с контроллером и несколько «fragment'ов» – страниц с функционалом.

## 6. Периферия

Дальнейшие действия - создание реальных устройств, таких как поилка, кормушка с приводами, свет, обогрев, и их включение в созданную основу.

## 7. Таймеры. Реле

Безусловно, одно из важных свойств контроллера курятника – работа по времени. Это необходимо как для управления освещением, так и для подачи корма. Чтобы считать время, я выбрал модуль «RTC DS3231 mini». А чтобы реализовать универсальные таймеры, написал алгоритм для неопределённого количества таймеров и реализовал их изменение посредством http запросов.

Для управления нагрузкой переменного тока (свет и прочая нагрузка до 1кВт) и постоянного <12V-10A я выбрал блок из четырёх твердотельных реле, работающих раздельно. В программе реализовал возможность работать как по таймеру, так и по датчику температуры.

## 8. Корпус

Далее - создал модели корпуса и распечатал их на 3д-принтере. Корпус состоит из планки с направляющими, на которую крепится вся электроника, и верхней крышки.

## 9. Доработка и добавление функций

Далее поочерёдно проводил доработку и тесты. Попутно разрабатывая систему подачи корма и воды, разные новые функции. В первую я добивался, чтобы backend (в моём случае контроллер) работал стабильно, а уже после реализовывал интерфейс приложения.

## 10. Тесты

/\* Описать мой курятничек))) Цитата отчима: «Надоело каждый день свет ходить включать и выключать и воду им всё время таскать надо. Покормить курок и яйца собрать – вот это я люблю» \*/

# Ресурсы

Для реализации проекта я использовал микроэлектронику, которую можно приобрести в Китае. (Например, в магазине на Aliexpress). Печатные платы высокого качества за небольшую цену можно заказать на JLCPCB. Там же можно заказать 3д-модели высокого качества.

/\*Таблица микроэлектроники

Необходимо и выборно(опционально)

Цены jlcpcb, его аналоги\*/

# Перспективы и направления развития

## 1. Контроллер

Прототип работает, но на будущее я разрабатываю единую полноценную плату с учтёнными особенностями питания: регулировка яркости ламп, питание от 220В переменного тока, «мосфеты» (МОП-транзисторы) и увеличение количества выходов; переход на smd компоненты.

В прошивке: новые режимы работы, управление внешними модулями (кормушка, поилка и т.д.).

В ESP8266 реализация подключения, не только в локальной сети, но и через Интернет: приложение или telegram бот.

## 2. Приложение

Подключение через Интернет, добавление новых функций и режимов работы. Увеличение функционала и упрощение интерфейса.

## 3. Модули

На момент написания у меня уже есть наработки в конструкциях кормушки и поилки. Например, винт кормушки (идею я взял у Alexgyver), поплавки и датчики для измерения уровня воды, клапан воды. Найдена некоторая информация по вентиляции, уборке отходов жизнедеятельности, сборщик яиц с заморозкой для хранения.

# Заключение

/\* НУЖЕН ИВАН-ПИСАТЕЛЬ \*/

# Список литературы

Текст

# Приложения

Приложение 1

Приложение 2