МАОУ "Усть-Кубинский центр образования"

Умный курятник

Ученик 10 класса:

Сакулин Иван Михайлович

Руководитель:

Андреев Николай Николаевич

с. Устье, 2023

**Оглавление**

[Вступление](#__RefHeading___1)

[Теоретическая часть](#__RefHeading___28)

[1.1. Выбор сред и инструментов разработки.](#__RefHeading___14)

[1.2. Arduino. ArduinoIDE. Atmega328p. ESP8266.](#__RefHeading___11)

[1.3. Контроль версий.](#__RefHeading___16)

[1.4. Android-приложение](#__RefHeading___17)

[1.5 work](#__RefHeading___39)

[Практическая часть](#__RefHeading___18)

[1. Начало.](#__RefHeading___5)

[2. Разработка первоначальной печатной платы.](#__RefHeading___29)

[3. Создание первоначального приложения и локального сервера на esp8266.](#__RefHeading___30)

[4. Первый функционал](#__RefHeading___31)

[5. Приложение](#__RefHeading___34)

[6. Наполнение](#__RefHeading___37)

[7. work](#__RefHeading___38)

[Заключение](#__RefHeading___6)

[Список литературы](#__RefHeading___7)

[Приложения](#__RefHeading___8)

# Вступление

/\*Заполню после ТЕХ КАРТЫ\*/

Актуальность проекта:

Цель проекта:

Задачи:

План работы:

# 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 следующей основе:

Картинку сюды: Андроид-приложение/аппаратный экран и энкодер – esp8266 – atmega328 – приводы, датчики.\*/

## 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 намного сложнее.

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.

## 1.5. 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.

## 1.4. 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

/\* найти(мой курятник), анализ, таблица сравнения\*/

# 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.

/\* В конечном итоге я сделал новую более совершенную разводку платы и заказал её. \*/

## 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. Программно 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\*/

# Заключение

Текст

# Список литературы

Текст

# Приложения

Ничего пока нет