

Контроллер умного курятника

/* Таким способом обозначаются комментарии. В готовой версии документа они должны отсутствовать, если они всё же есть, пожалуйста, не воспринимайте их всерьёз и обратитесь, по возможности, за полной версией по ссылке <https://github.com/OneTwoZzzPlus/chicken> файлы «SmartChicken-SakulinIM»*/

Ученик 10 класса:

Сакулин Иван Михайлович

Руководитель:

Андреев Николай Николаевич

Оглавление

Вступление	3
Введение и актуальность	3
Цель проекта	3
Задачи:	3
Проект подойдёт для	3
Теория	4
1.1. Выбор сред и инструментов разработки	4
1.2. Arduino. ArduinoIDE. Atmega328p. ESP8266.	4
1.3. Android - приложение	5
1.4. Печатные платы	5
1.5. Контроль версий и разработка	5
Аналоги	6
Этапы реализации	7
1. Начало.	7
2. Разработка первоначальной печатной платы.	7
3. Создание первоначального приложения и локального сервера на esp8266.	7
4. Первый функционал	7
5. Приложение	7
6. Периферия	7
7. Таймеры. Реле	8
8. Корпус	8
9. Доработка и добавление функций	8
Испытания	8
Ресурсы	9
Обеспечение	9
Время	9
Микроэлектроника	9
Перспективы и направления развития	10
1. Контроллер	10
2. Приложение	10
3. Модули	10
Заключение	10
Список литературы	12
Приложения	13

Вступление

Введение и актуальность

Курицам нельзя просто так насыпать кучу корма, налить вёдра воды, включить свет и ходить собирать яйца. Для их содержания необходимы режим дня, постоянная подача воды и выдача корма порционно, так как куры могут кушать безостановочно, до ожирения или предела своих жизненных возможностей. Если делать всё вручную, то необходимо несколько раз за день заходить, чтобы сделать пару простых однообразных действий. Автоматика призвана избавиться от этого! Нужного единого решения в Интернете я не смог найти, поэтому решил избавиться от этой проблемы. Оценив свои возможности, я решил сделать проект контроллера умного курятника, с возможностью управления нагрузкой, светом и интерфейсами для подключения систем подачи воды и корма. Настройка и управление контроллером происходит через мобильное приложение.

Цель проекта

Разработка и тестирование контроллера курятника с возможностью развития.

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

Проект подойдёт для

- Конечных пользователей – владельцев небольших и средних частных курятников (Необходима некоторая доработка до достижения промышленного производства)
- Инженерам-любителям, тем кто хочет самостоятельно собрать такой контроллер, добавив свои фишки (Необходим мануал по сборке и настройке)

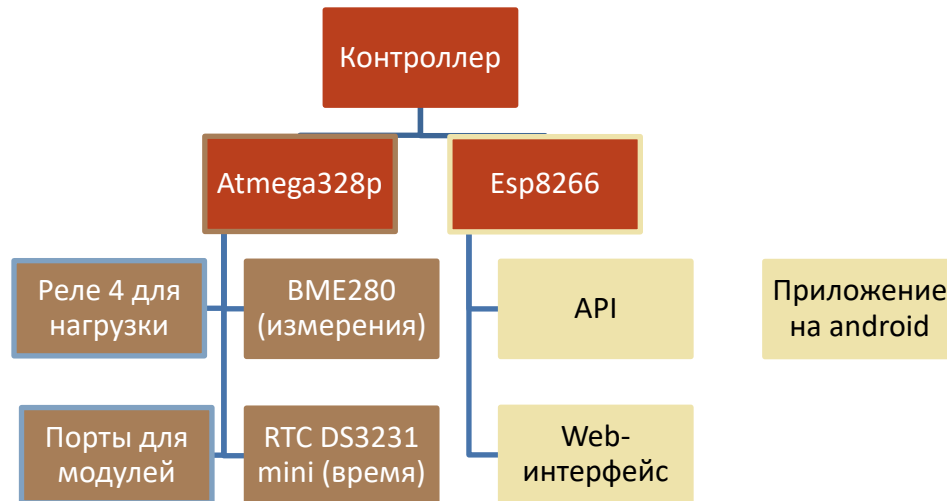
Теория

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 (1).

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

Android Studio — интегрированная среда разработки (IDE) для работы с платформой Android. Основана на программном обеспечении IntelliJ IDEA от компании JetBrains, — официальное средство разработки Android приложений (2).

1.4. Печатные платы

EasyEDA — кроссплатформенная веб-ориентированная среда автоматизации проектирования электроники, включающая в себя редактор принципиальных схем, редактор топологии печатных плат (3).

JLCPCB является крупнейшим предприятием по мелкосерийному производству печатных плат (4).

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 (5).

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 (6). 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. ИК.
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°C, 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	~1500р	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.

Испытания

/* 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

Обеспечение

Для реализации проекта я использовал микроэлектронику, которую можно приобрести в Китае (Таблица 2). Например, в магазине на Aliexpress. Печатные платы высокого качества за небольшую цену можно заказать на JLCPCB. Там же можно заказать 3д-модели высокого качества. Для разработки я также использовал ASP-программатор (для прошивки микроконтроллера), паяльную станцию, приспособления и расходные материалы к ней.

Время

- Проект разрабатывался на протяжении 3х месяцев.
- Расчётное время работы на проектом 200-400 часов.

Микроэлектроника

Таблица 2. Ресурсы микроэлектроники для прототипа

Название	Описание	Аналоги компонента	Стоимость, руб. 01.02.2023 ±50р
Atmega328p (DIP28)	8-разрядное устройство megaAVR, основанное на архитектуре RISC, улучшенной AVR. (D1)	Atmega328 (smd) и др. megaAVR.	250
ESP8266 (WeMos D1 mini)	Недорогой микрочип Wi-Fi со встроенным сетевым программным обеспечением TCP/IP и возможностью микроконтроллера. (D2)	ESP8266-12E/F	150
RTC DS3231 mini	Высокоточные часы реального времени (real-time clock, RTC) со встроенными термокомпенсированным кварцевым генератором (TCXO) и кварцевым резонатором. (D3)	DS1307, DS3231	100
BME280	Цифровой датчик влажности (нет в аналогах), температуры, давления (D4)	BMP280, DHT11, DHT22	300
Реле	Четырёхполосный релейный модуль с оптроном. Для нагрузки до 1кВт.	Отдельные реле	150
Прочее	Мелкие радиокомпоненты и провода.		50

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

Как мне кажется, основной цели проекта я добился: у меня есть контроллер и приложение, которые можно дорабатывать и улучшать.

Одну из версий контроллер я, надеюсь, вскоре установить в курятник на постоянной основе, а не только на время тестов, так как его использование очень удобно и сильно упрощает жизнь.

Основной репозиторий проекта на GitHub <https://github.com/OneTwoZzzPlus/chicken>.

Что есть в репозитории:

- программные коды;
- все, в т.ч. последняя рабочая, версии приложения;
- документация;
- фото и видео проекта.

Все актуальные ресурсы проекта и ссылки на них находятся в репозитории. Некоторые исходные программные коды могут быть закрыты.

Список литературы

Интернет-источники:

1. https://amperkot.ru/blog/arduino_projects_beginners/ (Что такое Arduino?)
2. https://ru.wikipedia.org/wiki/Android_Studio (Что такое Android Studio?)
3. <https://ru.wikipedia.org/wiki/EasyEDA> (Что такое EasyEDA?)
4. <https://www.instructables.com/Best-PCB-Manufacturer-for-Hobbyist-JLPCB-Review/> (Информация о JLCPCB)
5. <https://ru.wikipedia.org/wiki/GitHub> (Что такое GitHub?)
6. <https://мойкурятник.рф> (Аналог)

Datasheet компонентов:

- 1D. <https://static.chipdip.ru/lib/549/DOC001549488.pdf> (Atmega328P)
- 2D. https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf (ESP8266)
- 3D. <https://static.chipdip.ru/lib/248/DOC000248544.pdf> (RTC DS3231 mini)
- 4D. <https://static.chipdip.ru/lib/859/DOC003859720.pdf> (BME280)

GitHub <https://github.com/OneTwoZzzPlus/chicken>

Приложения

Приложение 1

/* фото тестов */

•